Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Dokumentácia k manažmentu projektu

Tímový projekt

Tím č. 19

Vypracoval: Jakub Perdek, Viktor Matovič
Vedúci projektu: Ing. Pavol Helebrandt Phd.

Obsah

1 Big Picture 3

1.1 Úvod 3

1.2 Role členov tímu 3

1.3 Podiel práce jednotlivých členov tímu 4

2 Aplikácie manažmentov 7

2.1 Úloha členov jednotlivých rolí 7

2.2 Nástroje pre aplikovanie scrumu 8

2.3 Komunikácia 8

Stretnutia v virtuálnych miestnostiach 8

Microsoft Teams 9

Wiki stránka 10

Facebook 10

Email 10

2.4 Manažment verzií 11

Workflow 11

Repozitáre 11

Nástroje gitu 11

Odovzdávanie kódu do repozitára 12

Tvorba žiadosti pre kontrolu kódu 13

Číslovanie verzií 13

2.5 Spravovanie backlogu 13

Prehliadka stavu projektu v Azure DevOps 14

Pravidlá prehliadky stavu projektu v Azure DevOps 14

Backlog v Azure DevOps 15

Ohodnocovanie náročnosti šprintu v story pointoch 15

2.6 Revízie kódu 16

2.7 Schvaľovací proces výstupov činností 17

2.8 Metodika tvorby dokumentácie 18

Technická dokumentácia 18

Zápisnice zo stretnutí 18

Zápisnice zo stretnutí 18

Dokumentácia retrospektívy šprintov 18

Ostatné dokumenty 19

3 Sumarizácia šprintov 20

3.1 Prvý šprint 20

Poznámky z priebehu prvého šprintu 20

Pokrok dosiahnutý na prvom šprinte 22

Výpis úloh z prvého šprintu 24

Retrospektíva prvého šprintu 25

Priebeh stretnutí 26

3.2 Druhý šprint 27

Pokrok dosiahnutý na druhom šprinte 27

Export úloh z druhého šprintu 29

Retrospektíva z druhého šprintu 29

3.3 Tretí šprint 31

Pokrok dosiahnutý na treťom šprinte 31

Export úloh z tretieho šprintu 34

Retrospektíva tretieho šprintu 35

3.4 Štvrtý šprint 36

Pokrok dosiahnutý na štvrtom šprinte 37

Export úloh zo štvrtého šprintu 40

Retrospektíva štvrtého šprintu 40

3.5 Piaty šprint 42

Pokrok dosiahnutý na piatom šprinte 42

Export úloh z piateho šprintu 45

Retrospektíva z piateho šprintu 46

3.6 Šiesty šprint 47

Pokrok dosiahnutý na šiestom šprinte 47

Export úloh zo šiesteho šprintu 51

Retrospektíva šiesteho šprintu 51

3.7 Siedmy šprint 53

Pokrok dosiahnutý na siedmom šprinte 53

Export úloh zo siedmeho šprintu 57

Retrospektíva siedmeho šprintu 58

3.8 Ôsmy šprint 60

Pokrok dosiahnutý na ôsmom šprinte 60

Export úloh z ôsmeho šprintu 64

Retrospektíva ôsmeho šprintu 64

3.9 Deviaty šprint 66

Pokrok dosiahnutý na deviatom šprinte 66

Export úloh z deviateho šprintu 70

Retrospektíva z deviateho šprintu 70

3.10 Desiaty šprint 72

Pokrok dosiahnutý na desiatom šprinte 73

Export úloh z desiateho šprintu 76

Retrospektíva z desiateho šprintu 77

3.11 Jedenásty šprint 78

Pokrok dosiahnutý na jedenástom šprinte 79

Export úloh z jedenásteho šprintu 83

Retrospektíva z jedenásteho šprintu 84

4 Globálna retrospektíva 86

4.1 Zimný semester 86

4.2 Letný semester 87

1 Big Picture

1.1 Úvod

V nasledujúcich častiach práca poskytuje pohľad do vnútra tímu č. 19, do dodržiavania metodík, vykonávania procesov a v neposlednom rade do tvorby artefaktov v procese vývoja. Aj keď je podstatou manažmentu práce dodržiavanie metodiky Scrum, počas vykonávania jednotlivých úloh je možné pozorovať používanie nástrojov typické aj pre inak manažované softvérové projekty.

1.2 Role členov tímu

Člen tímu	Scrum Rola v tíme	Úloha v tíme
Pavol Helebrandt	Product Owner	zastupuje zákazníka, spolutvorca Product Backlogu
Jakub Perdek	Biznis analytik	komunikácia medzi Product Ownerom, Scrum Masterom a Scrum tímom, spracovávanie dokumentácie k riadeniu projektu a inžinierskemu projektu
Miroslav Balga	Project Manager	reportovanie progresu Product Ownerovi, spracovávanie dokumentácie k inžinierskemu dielu
Viktor Matovič	člen Scrum tímu	nezadávať úlohy, riešiť konfliktné situácie, pomáhať členom tímu dosahovať ciele, komunikácia
Nikola Karakaš	člen Scrum tímu	spolupracovať počas šprintov, spolutvorca Sprint Backlogu
Abd Saleh	člen Scrum tímu	spolupracovať počas šprintov, spolutvorca Sprint Backlogu
Peter Spusta	Scrum Master	spolupracovať počas šprintov, spolutvorca Sprint Backlogu

Tabuľka 1: Role členov tímu

1.3 Podiel práce jednotlivých členov tímu

Člen tímu	Práca na funkcionalite	Percentuálny podiel (orientačné)
Jakub Perdek	Whois aplikácia Whois dokumentácia Kali nástroje tutoriál Návrhy scenárov Webová stránka tímového projektu +pravidelná aktualizácia Dokumentácia inžinierskeho diela Prevažná časť dokumentácie riadenia Metodika verziovania Metodika revízie kódov Metodika dokumentovania Metodika komunikácie Security – eshop frontend -šablóny košíka, titulná stránka, údaje o doručení, platobné informácie, finálna stranka Diagram nasadenia Tvorba progresu a retrospektívy pre šprint 3	28
Miroslav Balga		0
Viktor Matovič	Časť dokumentácie riadenia Metodika spravovania backlogu Security -eshop backend tvorba časti funkcionality Tvorba progresu pre šprint 1	20
Nikola Karakaš	Dokumentácia k Security -eshopu -diagramy, popis, používateľské rozhranie Tvorba progresu pre šprint 2	16
Abd Saleh	Security -eshop frontend Šablóny pre prihlásenie a registráciu Revízia a refaktoring kódu na frontende Pravidelné nasadenie stránky	17
Peter Spusta	Backend Security – eshop -tvorba webových služieb -ošetrenie problému s corsom -získanie prístupu k databáze Poznámky zo sprint review Vedenie diskusií v sprint review	19

Tabuľka 2: Podiel práce jednotlivých členov počas 2 a 3. šprintu

Člen tímu	Práca na funkcionalite	Percentuálny podiel (orientačné)
Jakub Perdek	Šablóna pre pridanie produktu Šablóna pre spravovanie používateľov Frontend pre šablónu určenú na spravovanie používateľov Spolupráca na backende pre spravovanie používateľov Presun prihlasovania do SQL databázy -tvorba backendu pre presun prihlasovania do SQL databázy -tvorba frontendu pre presun prihlasovania do SQL databázy -tvorba osobitného emailu pre možnosť posielať správy Update webovej stránky Tvorba progresu pre šprint 4 Tvorba retrospektívy pre šprint 4 Získanie hostingu pre SQL databázu a jej konfigurácia Pokusy rozbehnúť časti KYPO lokálne – chýbajúce zavislosti Update dokumentácie eshopu a jej revízia	29
Miroslav Balga		0
Viktor Matovič	Pomoc s tvorbou mapovania novej tabuľky	14
Nikola Karakaš	Review kódu na frontende Dokumentácia k Security -eshopu -dokumentácia spravovania používateľov -dokumentácia pridania priduktu -popis SQL injekcie	21,5
Abd Saleh	Rozbehnutie časti KYPO v lokálnom prostredí Riešenie problému s HTTPs -ešte nedokončené	19,5
Peter Spusta	Vloženie Hibernatu do projektu a rozpracovanie základných metód pre šablónu používateľov	16

Tabuľka 3: Podiel práce jednotlivých členov počas 4. šprintu

Člen tímu	Práca na funkcionalite	Percentuálny podiel (orientačné)
Jakub Perdek	Tvorba reportu o pokroku na šprinte Príprava reportu s retrospektívou – vypĺňali všetci Tvorba rolí pre eshop -na backende -na frontende Tvorba admin rozhrania pre zmenu rolí Tvorba informatívnej spätnej väzby pre používateľa Tvorba používateľskej príručky so scenármi Dopĺňanie dokumentu riadenia Tvorba JavaDoc pre kód pracujúci s relačnou databázou na backende Tvorba zobrazenia zaplatených produktov používateľovi pre možnosť ich stiahnutia	20
Miroslav Balga		0
Viktor Matovič	Tvorba CSRF ochrany na backende	20
Nikola Karakaš	Review kódu na frontende Časť dokumentácie JavaDoc na backende Dokumentácia k Security -eshopu -admin rozhranie -víťazný token	20
Abd Saleh	Revízia kódu, prípadný refaktoring	20
Peter Spusta	Dokončenie objednávky vrátením produktov s možnosťou ich stiahnutia pokiaľ je objednávka zaplatená	20

Tabuľka 4: Podiel práce jednotlivých členov počas 5. šprintu

2 Aplikácie manažmentov

Manažment sme aplikovali na základe dohodnutých metodík. Cieľom každej z metodík je zefektívniť proces vývoja softvéru.

2.1 Úloha členov jednotlivých rolí

Softvérový projekt bude a je riadený v 2 týždňových pravidelných intervaloch, nazývanými šprint. Product Owner realizuje komunikáciu so zákazníkom a v manažmente projektu ho zastupuje smerom k vývojovému tímu. V komunikácii medzi vývojovým tímom a zákazníkom je prostredníkom Biznis Analytik a Projektový manažér. Na udržiavanie želanej kultúry, výkonnosti tímu a efektívnej komunikácie má Scrum tím k dispozícii Scrum Mastera. Scrum vývojový tím vykonáva úlohy manažované a zaznamenávané v Sprint Backlogu, ktorého obsah sa určuje na základe Product Backlogu. Spolutvorcami Product Backlogu (Spring Backlogu aj Product Backlogu) sú členovia Scrum tímu a Product Owner. Scrum Master nemá možnosť určovať prácu členom Scrum tímu.

Vzhľadom na súčasnú pandemickú situáciu nie je možné realizovať osobné stretnutia členov tímu. Kooperácia členov tímu je realizovaná pomocou nástrojov s webovým rozhraním. Biznis analytik kontinuálne spolu s ostatnými členmi tímu skúma platformu Kypo a po získaní dôležitých informácií tieto informácie podáva ostatným členom projektu. V prípade že člen tímu alebo viacero členov tímu majú problém s realizovaným úlohy na ktorej pridelení sa tím vopred dohodol, Scrum Master mu pomôže a danú úlohu s ním konzultuje. Scrum Master sa taktiež kontinuálne vzdeláva v technikách a metodike Scrum, zlepšuje svoje zručnosti. Scrum Master má však nepriamo zakázané prideľovať úlohy členom tímu, nerozhoduje teda ani o technických záležitostiach a pri rozhodnutiach ktoré je potrebné spraviť pri realizácii jednotlivých úloh členov tímového projektu. V záujme efektívneho zavádzania a implementovania zmien pre zákazníka (FIIT STU) interpretovaných Product Ownerom počas softvérového projektu budeme používať agilný prístup - metodiku Scrum. Role jednotlivých členov Scrum tímu sú uvedené v tabuľke nižšie:

2.2 Nástroje pre aplikovanie scrumu

Na odbremenenie Scrum tímu od papierovej dokumentácie, na efektívnu komunikáciu na diaľku a na sprehľadnenie a vizualizáciu aktuálnych a plánovaných činností v šprintoch používame nasledujúce manažérske nástroje:

Softvérový nástroj	**Použitie / úloha**
fakultný Microsoft Teams	komunikácia členov tímu, denné standupy, retrospektívy po každom šprinte, sprint review (prezentácia výsledkov šprintov Product Ownerovi)
Azure DevOps	vizualizácia aktuálneho stavu projektu, spravovanie produkt a sprint backlogu

Tabuľka 5: Hlavné nástroje pre aplikovanie Scrumu

Na každý typ úlohy ktorej stav je manažovaný v Azure DevOps je možné priradiť len jedného vykonávateľa (obmedzenie prostredia). V prípade že na druhu činnosti sa zúčastňuje viac ako jeden vykonávateľ, táto činnosť bude uvádzaná v prostredí Azure DevOps a v dokumentácii k softvérovému projektu.

2.3 Komunikácia

Komunikácia je v tíme veľmi dôležitá. Používame preto rôzne nástroje pre komunikáciu. V nasledujúcich podkapitolách uvádzame dôležité časti z metodiky komunikácie.

Stretnutia v virtuálnych miestnostiach

Počas pandémie nie je možné osobne sa stretnúť minimálne so všetkými členmi. Osobné stretnutie by pomohlo v komunikácii aj tým, že by ju urýchlilo. Ďalšou výhodou je priamy rozhovor, a neraz aj názorná demonštrácia kreslením na tabuľu a podobne. Museli sme hľadať možnosti vo virtuálnom priestore. Sú nimi Microsoft Teams, Google Meets, Facebook, Slack ale aj emailová komunikácia v akademickom informačnom systéme našej univerzity.

Na týchto stretnutiach diskutujeme o problémoch, ale hlavne rizikách ktoré jednotlivé úlohy zahŕňajú. Ich časové a technologické obmedzenia sú najväčším zdrojom nami vyhodnocovaného rizika. Zamýšľame sa aj o smerovaní projektu, keďže našou úlohou je navrhnúť používateľsky príjemný scenár, na ktorom sa naučí techniky informačnej

bezpečnosti. Požiadavky na vytvorený scenár neobsahujú detaily jeho obsahu, preto je potrebné tento obsah navrhnúť.

Trojhodinové stretnutia absolvujeme s vedúcim, zvyšok času určeného na projekt v nami dohodnuté intervaly pri tvorbe obsahu funkcionality projektu.

Harmonogram stretnutí

Utorok: **8:00 – 11:00** **- aj s product ownerom**

Štvrtok: **20:00 – 00:00 – väčšinou časť rozdelená**

alebo presunutá na víkend

Microsoft Teams

Formálne používaný komunikačný kanál pre stretnutia a ich nahrávanie. Zároveň necháva zaznamenané komentáre, ku ktorým je možné sa neskôr vrátiť. Podporuje aj tvorbu viacerých miestností pre rôzne témy komunikácie. Založili sme tu aj vlastnú Wiki stránku, do ktorej dávame vytvorené dokumenty a programy.

Nami vytvorené miestnosti:

Všeobecné (General)

- Pre stretnutia tímu

Scenáre (Scenarios)

- Pre návrh bezpečnostných scenárov

TP Konverzácie (Tp Conversations)

- Pre informácie k tímovému projektu od product ownera

Vývoj webovej stránky tímu (Website Development)

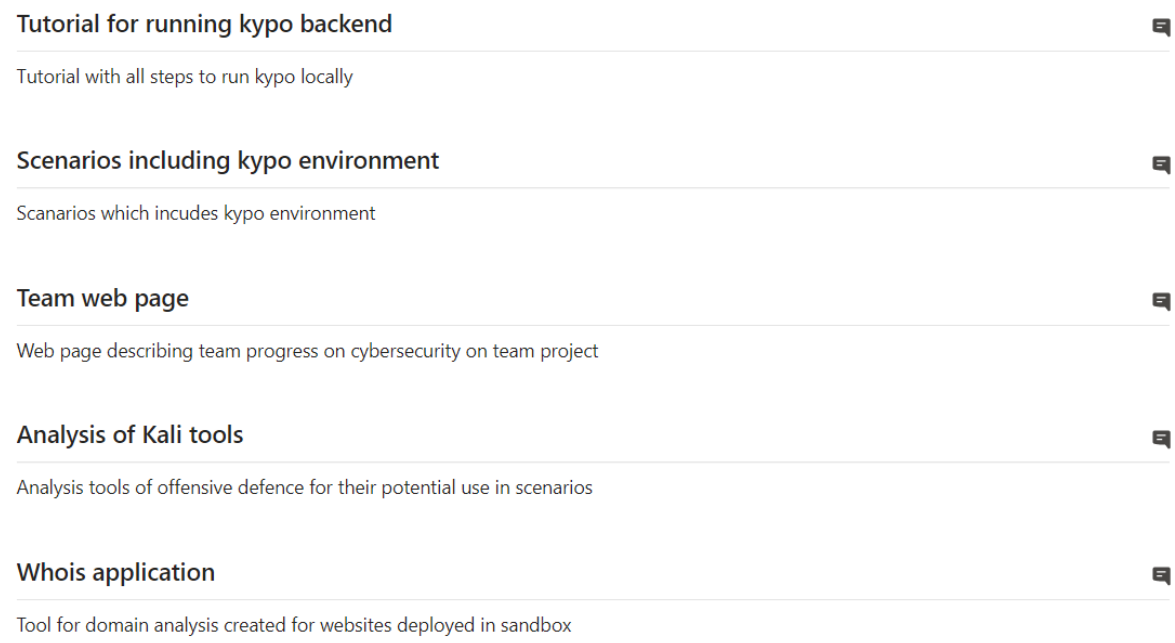
- Pre komunikáciu o tvorbe, aktualizácii a nasadení stránky

Bezpečnosť a penetračné testovanie (Security)

- Pre vývoj manuálov a diskusiu o penetračnom testovaní a používaní nástrojov pre penetračné testovanie

Wiki stránka

Stránka s všetkými vytvorenými analýzami a aplikáciami. Obsahuje aj stručný popis pridaných častí. Okrem tejto stránky sú dokumenty, hlavne z oblasti manažmentu a technická dokumentácia zverejňované na webovom sídle tímu.



Obrázok 1: Wiki stránka

Facebook

Najčastejšia neformálna komunikácia je prostredníctvom sociálnych sietí. Rýchlejšie sa načíta oproti Microsoft Teams. Zároveň rýchle chatovanie pomáha pri snahe o rýchlu orientáciu alebo riešenie problémov. Zároveň je touto formou možné vytvoriť hlasovanie a hlasovať o termíne stretnutia alebo o konkrétnom rozhodnutí. Nevýhodou je nemožnosť nahrať niektoré súbory do chatu a aj zmes osobných dojmov a emócií zneprehľadňujúca riešené problémy. Pri integrácii frontendu s backendom bol hojne používaný.

Email

Ako tím sme určili dve primárne mailové adresy, na ktoré sme pripravený reagovať. V rámci tímu by nemal byť problém rýchleho zdieľania informácií medzi členmi. Každý člen informuje ostatných o mailoch s tematikou tímového projektu. Pred zavedením chatovania to bol jediný spôsob komunikácie. Kontakt je určený ako komunikačný prostriedok s verejnosťou. Vedúci k súkromným emailovým adresám členov prístup nemá.

2.4 Manažment verzií

Pri tímovej práci na tvorbe kódu a výsledných aplikáciách je potrebné uplatniť pravidlá verziovania. V nasledujúcich podkapitolách uvádzame dôležité časti z metodiky verziovania. Počas tímového projektu manažujeme jednak zdrojový kód, dostupnosť a obsah prezentačnej webovej stránky webového tímu a jednak zdrojový kód predmetu tímového projektu samotný.

Workflow

Nástroj Azure DevOps obsahuje tri stavy To Do, Doing a Done. V budúcnosti pridáme stavy Review request and Review done.

Repozitáre

Využívame niekoľko repozitárov pre rôzne aplikácie potrebné pre projekt. Výstupom má byť prostredie pre kybernetickú obranu určené pre študentov univerzity preto naše repozitáre sú verejné. Súkromným je repozitár so serverom, pretože obsahuje prístupové údaje do databázy. Pri vývoji sme pracovali na jednoduchších aplikáciách samostatne, preto sme využívali jednu master vetvu. Následne sa spravil review celej aplikácie. V budúcnosti pri väčších aplikáciách a rozšíreniach budeme vytvárať nové vetvy s jednotlivými features, vytvárať pull requesty pre review a spájať ich po kontrole.

Backend pre bezpečnostný eshop: https://github.com/Peter-Spusta/Cyran-Server
Frontend pre bezpečnostný eshop: https://github.com/jperdek/security-eshop
Whois aplikácia pre analýzu web aplikácií: https://github.com/jperdek/whois-lookup
Funkcionalita webovej stránky tímu: https://github.com/jperdek/CYRAN-web-page

Nástroje gitu

Pri práci s gitom uvádzame prehľad najpoužívanejších operácií.

Tvorba novej vetvy:

```
git checkout -b <názov vetvy>
```

Prepnutie sa do druhej vetvy:

git checkout <názov vetvy>

Zobrazenie aktívnych vetiev v repozitári:

git branch

Aktualizovanie mapovania jednotlivých vetiev:

git fetch

Odovzdávanie kódu do repozitára

Použitie postupnosti príkazov pre git:

Pridanie súborov do lokálneho úložiska:

git add .

Vytvorenie commitu:

git commit .

alebo aj so správou pre commit

git commit -m “Sprava pre commit”

Pridanie commitnutých súborov z lokálneho úložiska do globálneho:

git push

Ďalšie užitočné príkazy:

Discardnutie vykonaných zmien:

git checkout -- .

Úprava predchádzajúceho commit:

git commit -amend

Zistenie či sú súbory pridané do commitu:

git status

Zistenie zmien vykonaných v poslednom commite:

git show

Zobrazenie zoznamu commitov:

git log

Zobrazenie zmien, ktoré nie sú súčasťou commit:

git diff

V prípade malého projektu na ktorom sa podieľa jeden člen tímu sme umožnili vkladať kód do hlavnej vetvy. V prípade väčších projektov s viacerými účastníkmi sa predpokladá dodržiavanie nasledujúcich pravidiel pre tvorbu žiadosti pre kontrolu kódu.

Tvorba žiadosti pre kontrolu kódu

Každý člen tímu pracujúci na osobitnej features používa výhradne novú vetvu. Po tvorbe konkrétnej funkcionality vytvorí pull request a kontaktuje kompetentnú osobu pre review kódu. Po jeho kontrole a pozitívnych výsledkoch môže byť vytvorená nová vetva s funkcionalitou spojená s hlavnou. Rovnako by mali byť oboznámené všetky osoby v tíme.

Číslovanie verzií

Verzie číslujeme v tvare <*major*>.<*minor*>.<*patch*>. Číslo je vkladané do vetvy pre release, v ktorej je funkčná aplikácia vhodná pre použitie v rámci vytvorenej funkcionality.

Konvencia číslovania verzií:

- Major
  - Hlavná funkcionalita a podstatné zlepšenia
  - V pred vydanej fáze má hodnotu 0
  - Číslo prvej produkčnej verzie je 1.0.0
- Minor
  - Väčšie zmeny v aplikácií
  - Pridanie ďalších zlepšení a features
- Patch
  - Malá oprava funkcionality

2.5 Spravovanie backlogu

V nasledujúcej časti uvádzame časť metodiky spravovania backlogu. Backlog pre tímový projekt 1 udržiavame v Azure DevOps a na nasledujúcom odkaze: https://dev.azure.com/FiitCyran. Manažment backlogu, teda aj jeho priebežný review je povinný pre každého člena tímu. Vykonáva sa priebežne. Každý člen tímu je povinný

vykonať prehliadku Kanban Boardovej časti Boards aspoň raz za šprint. Do časti obsahujúcej Kanban tabulu sa používateľ dostane po kliknutí v ľavom kontextovom paneli.

Prehliadka stavu projektu v Azure DevOps

Člen tímu by si pri prehliadke Kanban tabule položiť nasledujúce otázky:

- **Otázka ohľadom obsahu**
    - Pribudli v tabuli aktivity o ktorej neviem, ktoré neboli dohodnuté na začiatku šprintu?
- **Otázka ohľadom pokroku**
    - Vzhľadom na čas (blíži sa koniec šprintu)ktoré z položiek typu Epic, Issue a Task sú príliš dlhý čas v stave rozpracovania? Vedel by som vykonávateľovi danej úlohy alebo činnosti pomôcť alebo poradiť? Pribudli v tabuli aktivity o ktorej neviem, ktoré neboli dohodnuté na začiatku šprintu?

- **Otázka ohľadom obsahu**
    - Mám všetky úlohy za ktoré som v šprinte zodpovedný ukončené? Označil som Task alebo Issue značkou Dokončené ale ešte som neprezentoval výsledok svojej činnosti?

V prípade odpovede áno na jednu alebo viacero z vyššie položených otázok je členovi tímu odporúčané kontaktovať Scrum Mastera, člena tímu zodpovedného za konkrétnu úlohu/úlohy pomocou tímového nástroja na komunikáciu – Microsoft Teams.

Pravidlá prehliadky stavu projektu v Azure DevOps

- V prípade nezrovnalostí medzi dohodnutými činnosťami na začiatku šprintu a obsahom Kanban tabule je členovi tímu odporúčané kontaktovať Biznis analytika.
- V prípade, že člen tímu dokončil činnosť za ktorú bol v šprinte zodpovedný jeho povinnosťou je
    - túto skutočnosť oznámiť Scrum Masterovi tímu.

    - vytvoriť v skupinovej konverzácii v Teams hlasovanie o presnom čase konania prezentácie výsledku.
    - Ak sa blíži termín stretnutia kvôli dennému standup-u, zodpovedný riešiteľ ani Scrum Master tieto činnosti realizovať nemusia. Na nasledujúcom stretnutí sa však musí vyčleniť dostatočný priestor na prezentáciu výsledkov.
- Výsledok ukončovanej činnosti bude podriadený schvaľovaciemu procesu podľa metodiky Definition of Done.

Backlog v Azure DevOps

Produktový backlog a šprint Backlog sa v časti Backlog v Azure DevOps zobrazuje ako jedna tabuľa. V ľavej časti záložky Backlogs je možné ukázať aktuálne naplánované činnosti/ úlohy pre prebiehajúci alebo naplánovaný šprint. V záujme dodržania princípov Scrumu sa členom tímu neodporúča plánovať aktivity na viac ako jeden šprint dopredu. Po kliknutí na odkaz s názvom šprintu sa členovi tímu zobrazí zoznam aktivít naplánovaných na aktuálne prebiehajúci šprint. Členovia tímu sú zodpovední za spravovanie informácií poskytnutých v položkách na stránke k im prideleným úlohám. Členom tímu je odporúčané svoje položky komentovať. Po zaradení člena tímu na vykonávanie konkrétnej úlohy je člen tímu zodpovedný za vloženie popisu do položky oznamujúcej predom dohodnutú náročnosť v činnosti v story pointoch.

Ohodnocovanie náročnosti šprintu v story pointoch

Náročnosť činnosti budeme hodnotiť v story pointoch podľa nasledujúceho pravidla:

Náročnosť činnosti	Typ úlohy
10 – 15 story pointov	Epic
5 – 9 story pointov	Issue
1 – 4 story pointov	Task

Tabuľka 6: Ohodnotenie náročnosti šprintu v story pointoch

V predchádzajúcej tabuľke sú uvedené hodnoty len orientačné, ale zato odporúčané. V prípade, že sa pre jeden šprint plánuje vykonať typ úlohy ktorého náročnosť zaberá min. 70% náročnosti v story pointoch naplánovaných pre všetky činnosti začínajúceho šprintu je potrebné tento typ činnosti rozčleniť na menšie. Pri plánovaní činností v začínajúcom šprinte je potrebné predom zohľadniť nielen náročnosť realizovaných úloh, ale aj schopnosť tímu dodávať inkrementy počas jednotlivých šprintov (iterácií). Úlohy manažované v šprint alebo product Backlogu kategorizujeme hierarchicky. V prípade, že úloha môže byť podľa svojho kontextu pričlenená k inej (vzťah parent - child) je odporúčané členov poddruženej úlohy túto úlohu spojiť s jej vlastníkom (parentom).

2.6 Revízie kódu

V tíme sa pravidelne informuje o dosiahnutom pokroku na projekte. Pri tvorbe aplikácií väčšieho rozsahu skladajúcich sa z viacerých komponentov je potrebná revízia kódu. Kód by mal zostať prehľadný a nemal by obsahovať chyby. Revízia kódu by to mala zabezpečiť. V nasledujúcej časti uvádzame časť metodiky zaoberajúcej sa revíziami kódu.

Kontrola kódu prebieha niekoľkými spôsobmi:

- ***Pri práci vo dvojici***
    - jednotliví členovia tímu si prezerajú kód navzájom. Pokiaľ nájdu nejakú chybu alebo problém navzájom sa informujú a chybu opravia. Zároveň sa snažia písať kód rovnakým štýlom, tak ako keby ho písal jeden.
- ***Kontrola na strane autora***
    - autor kladie dôraz na správne odsadenie kódu
    - používa výstižné, opisné, názvy premenných
    - dodržuje zásady konkrétneho programovacieho jazyka
    - snaží sa ošetriť potencionálne neošetrené časti kódu
- ***Kontrola na strane reviewera***
    - kontroluje dodržané konvencie
    - prejde celý kód a snaží sa ho pochopiť a analyzovať
    - snaží sa odhaliť chyby v kóde
    - manuálne otestuje niektoré vstupy

Autor postupuje pri kontrole nasledujúcim spôsobom:

- Pokiaľ je projekt väčšieho rozsahu a nepracuje na ňom sám vytvorí pull request.
- Dohodne sa s kompetentným členom tímu (napríklad v prípade backendu tým, kto má na starosti backend) pre kontrolu vykonanej práce.
- Čaká na vykonanie kontroly ďalším kontrolórom kódu s ktorým sa dohodol.
- Podľa výsledkov kontroly vykoná jednu z akcií:
  - Ak kód úspešne prešiel kontrolou označí svoju úlohu v Azure DevOps ako splnenú, tým že jej nastaví stav Done
  - Ak kód neprešiel kontrolou, je autor povinný si ho opraviť, prípadne požiadať ďalšieho člena tímu o pomoc. Pokiaľ už nezostáva v šprinte čas zváži sa presunutie nedokončenej úlohy do ďalšieho šprintu. Na opravenom kóde by mal byť znovu vykonaná revízia kódu.

Reviewer postupuje pri revízii nasledujúcim spôsobom:

- Informuje sa o prípadnej potrebnej revízii z Azure DevOps.
- Nastaví stav úlohy na Doing review.
- Vykoná revíziu s dôrazom na posúdenie kvalitatívnych znakov kódu a nález potencionálnych chýb.
- Podľa výsledkov revízie:
  - Ak revízia dopadla úspešne povolí prípadný pull request.
  - Ak revízia dopadla neúspešne prípadne urobí záznam o chybách, a čo najskôr kontaktuje autora kódu.

Oboznámi autora kódu a následne aj celý tím s výsledkami vykonanej revízie.

2.7 Schvaľovací proces výstupov činností

Schvaľovací proces výstupov činností členov tímu počas jednotlivých šprintov sa riadi dvoj-stupňovým procesom schvaľovania. Členovia tímu si navzájom počas behu jednotlivých šprintov prezentujú výsledky svojej činnosti. Pri takejto prezentácii je potrebná účasť každého člena tímu. Pri ukončovaní šprintu sa výsledky vykonaných činností prezentujú Product Ownerovi.

2.8 Metodika tvorby dokumentácie

Metodika tvorby dokumentácie je určená pre tvorbu kvalitnej a zrozumiteľnej dokumentácie. Nasledujúce podkapitoly sú prebraté z metodiky tvorby dokumentácie tímu CYRAN.

Technická dokumentácia

Dokumentáciu vypracováva člen tímu, ktorý sa rozhodol ju spracovať. Podmienkou je, aby ju po dokončení posúdil, prípadne aj doplnil, autor kódu. Dokumentáciu by si mal prečítať každý člen tímu a poskytnúť jej autorovi spätnú väzbu. Pri revízii kódu sa kontroluje aj dokumentácia a okomentovanie kódu. Pokiaľ je projektov viac, pre každý sa vytvorí samostatný dokument. Zjednotenie týchto dokumentov sa uvedie v dokumente inžinierskeho diela.

Zápisnice zo stretnutí

Po každom stretnutí sa spíšu poznámky z myšlienok, problémov, chýb a ďalších informácií, ktoré členovia tímu a product owner povedali. Dôraz je kladený na biznis procesy, ale aj prípadné riziká a odkonzultované rozhodnutia. Z týchto zápisníc sa zhotovia úlohy, ktoré sa budú v šprintoch realizovať. Dokumenty pravidelne vkladáme na stránku tímu.

Zápisnice zo stretnutí

Spôsob akým kvalitne a efektívne dosiahneme svoje ciele by mal byť spísaný v metodikách. Cieľmi môže byť zlepšenie komunikácie alebo lepšia revízia kódu nepripúšťajúca nekvalitný kód. Metodiky rovnako zverejňujeme na našu stránku pod menom konkrétnej metodiky.

Dokumentácia retrospektívy šprintov

Po skončení šprintu diskutujeme o jeho priebehu, problémoch a zlepšeniach organizácie jednotlivých úloh a ich vykonávania. Základnými otázkami, na ktoré musí v priebehu retrospektívy zodpovedať každý účastník sú:

- Čo sa nám podarilo vykonať?
- Čo sa nám nepodarilo vykonať?
- Aké problémy sme identifikovali alebo máme?
- Čo by sme v nasledujúcom šprinte zlepšili?

Prípadne začne aj diskusia k identifikovaným problémom alebo návrhom na zlepšenie. V prípade, že sa niekto nemôže zúčastniť informuje ostatných členov tímu. Výsledky z retrospektívy by mali byť pochopené každým členom tímu, ktorý by sa nimi mal zároveň aj riadiť.

Priebeh retrospektívy je nasledovný:

- Scrum master sa opýta prvú z otázok vybratého člena
- Menovaný člen na ňu odpovie
- Jeden z členov zapisuje rozhovor pre jeho dokumentáciu
- Po skončení môžu ostatní niečo dodať alebo sa opýtať
- Scrum master sa opýta otázku ďalšieho člena tímu, až kým sa neopýta všetkých. Potom sa opýta ďalšiu otázku. Týmto spôsobom pokračuje pre všetky menované otázky.

Po skončení sa z retrospektívy vytvorí dokument. Retrospektívu rovnako zverejňujeme na stránke tímu.

Ostatné dokumenty

V priebehu vypracovávania scenárov alebo dokumentovania môžu vzniknúť artefakty. Sú nimi napríklad manuály k nástrojom pre Kali Linux alebo tutoriály pre inštaláciu KYPO. Tieto artefakty musia byť zverejnené na wiki stránke tímu. Členovia tímu ich môžu zverejniť aj na svojej stránke tímu v sekcii download.

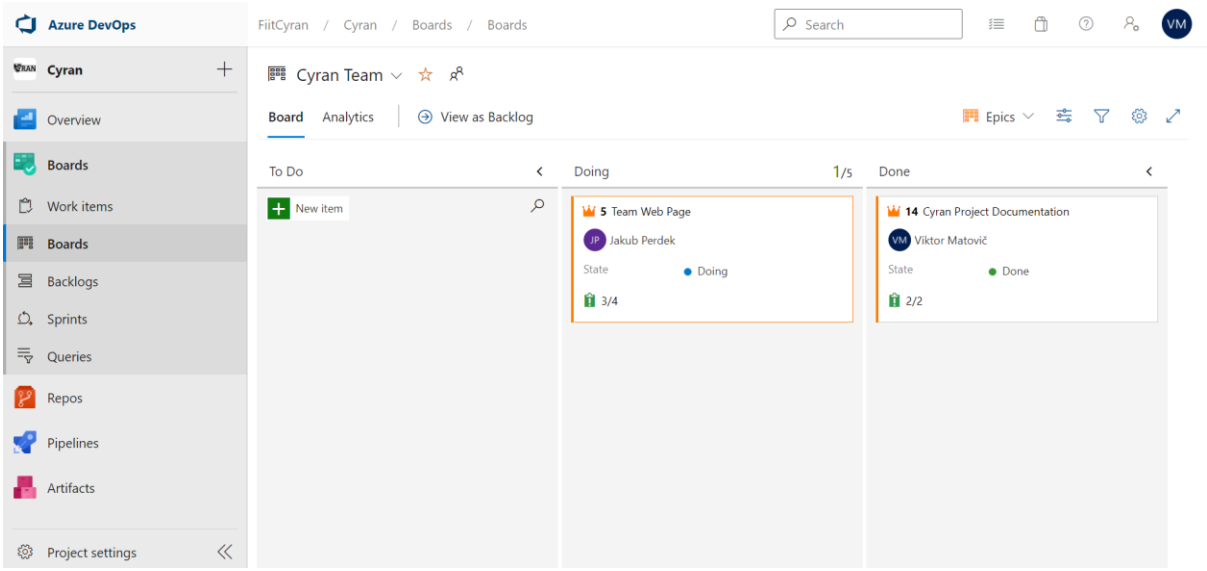
3 Sumarizácia šprintov

3.1 Prvý šprint

Prvý šprint podľa harmonogramu tímového projektu začal 12. Októbra v zimnom semestri. Počas prvotných stretnutí tímu s Product Ownerom bolo ešte pred začatím prvého šprintu rozhodnuté o potrebe analyzovať tému, konzultovať náš prínos do cyber-range platformy Kypo a potrebe dokumentovať progres a rozhodnutia vykonávané v rámci tímu.

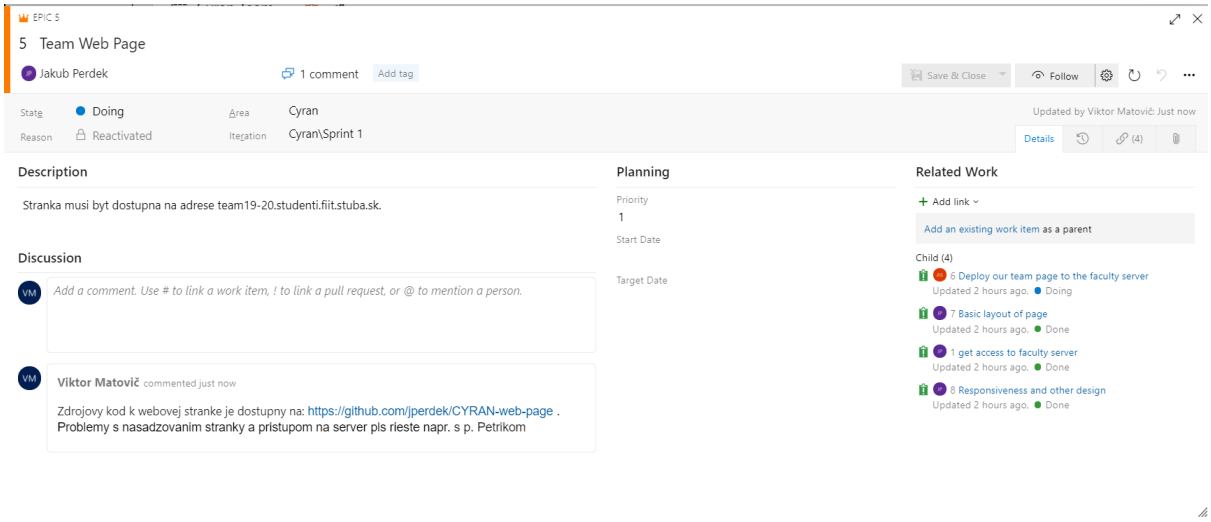
Poznámky z priebehu prvého šprintu

Z požiadaviek pre prvý šprint vznikli nasledujúce top-level úlohy, ktorých stav dokumentujeme v Azure DevOps:



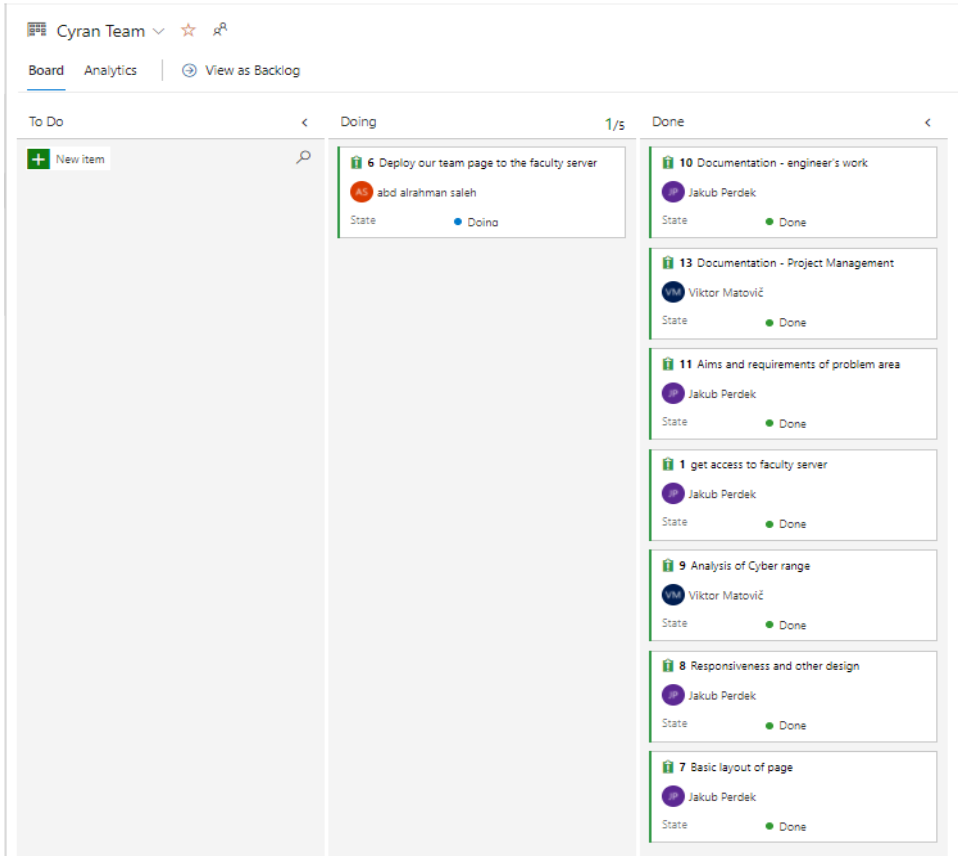
Obrázok 2: Vznik najprioritnejších úloh pre prvý šprint

Na predchádzajúcom obrázku sú zobrazené dva Epicy, spracovanie dokumentácie k tímovému projektu a príprava a nasadenie tímovej prezentačnej webovej stránky. Nasadzovaniu webovej stránky predchádzalo jej vytvorenie a odprezentovanie ostatným členom tímu. Tieto činnosti sú zobrazené napojené na Epic: Team Web Page.



Obrázok 3: Epic s webovou stránkou tímu

Stav ostatných dohodnutých úloh je podľa metodiky *Metodika spravovania backlogu* manažovaný v Product Backlogu. Na nasledujúcom obrázku je možné vidieť činnosti ktoré sa doteraz v tíme realizovali. Výskum predmetnej doménovej oblasti, vytvorenie a nasadzovanie webovej prezentačnej stránky tímu, dokumentovanie inžinierskeho diela a manažmentu softvérového projektu:



Obrázok 4: Backlog z prieehu prvého šprintu

Pokrok dosiahnutý na prvom šprinte

Scrum tímu č. 19 sa podarilo za posledný šprint úspešne dokončiť 10 úloh a 2 epicy. Práca na nich bola distribuovaná medzi troch členov tímu, ktorí sa na týchto úlohách podieľali.

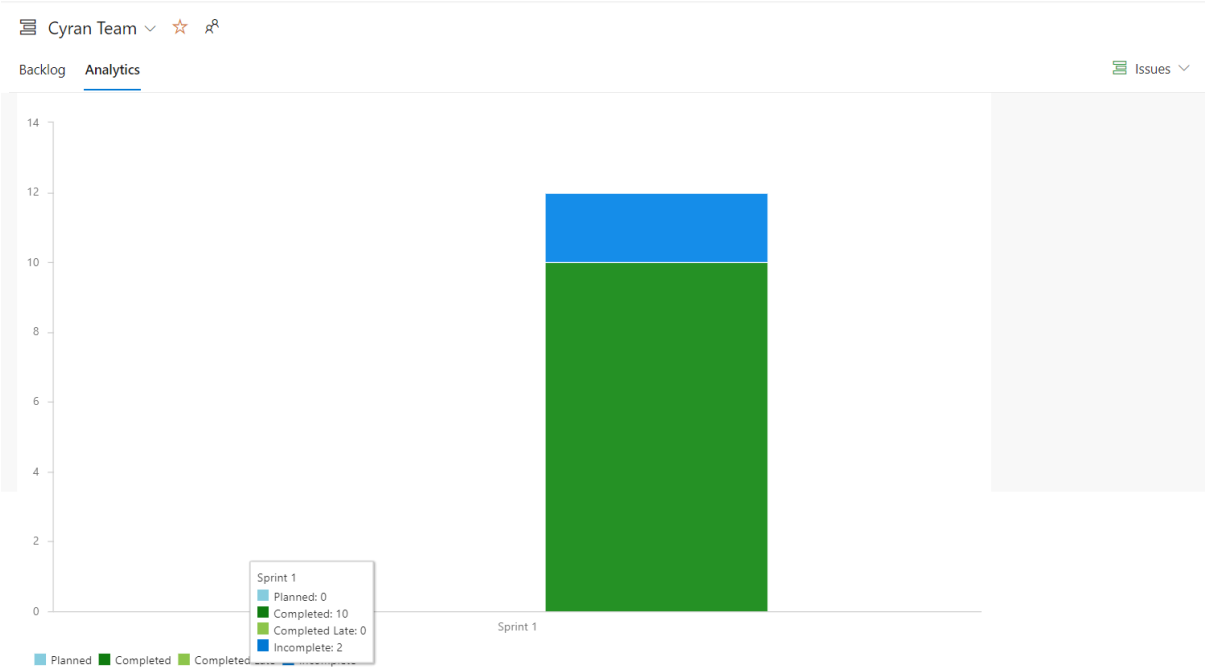
Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (piatok 11. 6)	Šprint
Vytvorenie a dotváranie prezentačnej webovej stránky tímu	Jakub Perdek	dokončené	šprint č. 1
Nasadzovanie	Abd Saleh	prvotné nasadenie dokončené	šprint č. 1
Administrácia servera pre nasadenie prezentačnej webovej stránky tímu	Jakub Perdek, Abd Saleh	vykonávanie administrácie, zmien a update-ov podľa potreby	šprint č. 1 a 2
Dokumentácia k riadeniu projektu vytvorenie metodík, dokumentácia, reportovanie retrospektívy a progresu	Viktor Matovič	aktualizované podľa potreby a časového miľníka	šprint č. 1 a č.2
analýza Kypo, konceptu, vybraných útokov (OWASP)	celý Scrum tím č. 19	kontinuálne vykonávaná úloha	šprint č. 1 a č. 2
vytvorenie example aplikácie pre jej ďalšie použitie počas útoku v rámci prostredia Kypo	Jakub Perdek	dokončené	šprint č. 2
Analýza nástrojov v Kali	Jakub Perdek	dokončené	šprint č. 1
Modelovanie scenárov útokov a obrany	Jakub Perdek	dokončené	šprint č. 1

Tabuľka 7: Úlohy na prvom šprinte

V prípade že niektorému členovi tímu nebola priradená úloha explicitne, ale táto úloha bola priradená všetkým členom tímu táto skutočnosť bola daným členom tímu oznámená počas

Scrum stand-up stretnutia podľa potreby, Sprint review stretnutia alebo Scrum retrospektívy. Prezentácia výsledkov úloh bola realizovaná podľa tímovej metodiky komunikácie.

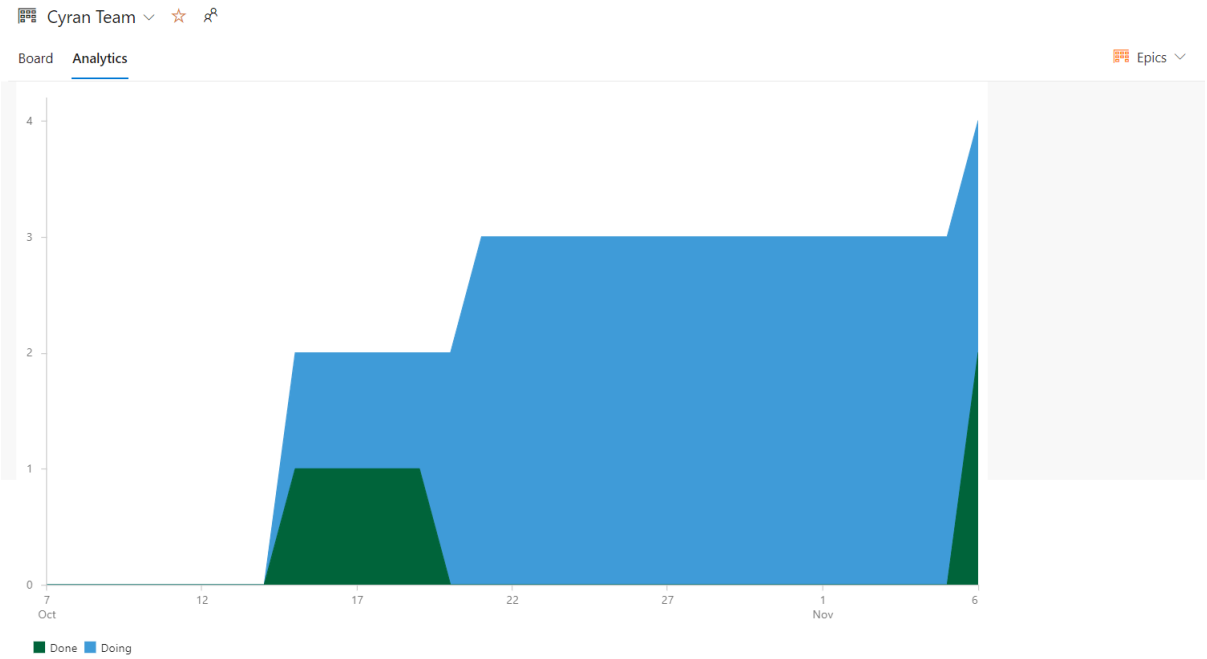
Nasledujúci diagram ukazuje výkonnosť tímu. O ňu sa v súčasnosti starajú len vybraní členovia tímu no ukazuje presah dokončených úloh nad nedokončenými za posledný a prvý šprint. Táto metrika bude užitočná po dokončení viacerých šprintov, pretože po analýze grafu bude viditeľné ako v tíme spolupracujeme (nespolupracujúci tím bude mať v tomto pomere veľké rozdiely).



Obrázok 5: Velocity tímu v šprinte 1

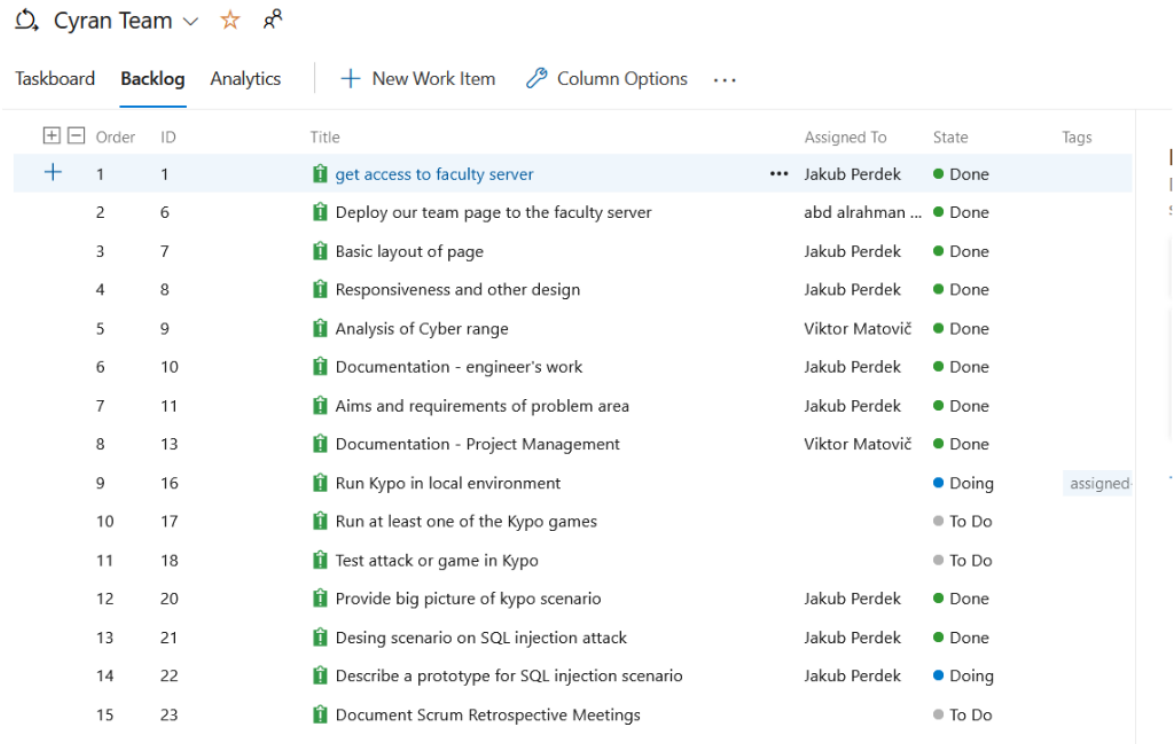
Na nasledujúcom obrázku je vidno že tím má stále väčšinu epicov rozpracovaných, než dokončených. Nedokončené úlohy je preto nútený posúvať na riešenie do ďalších šprintov. O obrázok ďalej je vidno stav a pomer dokončených úloh (taskov) oproti nedokončeným. Tento druhý diagram však už epicy nezobrazuje.

Kumulatívny tok: Diagram pre prvý šprint a začiatok druhého (epicy)



Obrázok 6: Kumulatívny tok: Diagram pre prvý šprint a začiatok druhého (issues - tasky)

Výpis úloh z prvého šprintu



Order	ID	Title	Assigned To	State	Tags
1	1	get access to faculty server	Jakub Perdek	Done	
2	6	Deploy our team page to the faculty server	abd alrahman ...	Done	
3	7	Basic layout of page	Jakub Perdek	Done	
4	8	Responsiveness and other design	Jakub Perdek	Done	
5	9	Analysis of Cyber range	Viktor Matovič	Done	
6	10	Documentation - engineer's work	Jakub Perdek	Done	
7	11	Aims and requirements of problem area	Jakub Perdek	Done	
8	13	Documentation - Project Management	Viktor Matovič	Done	
9	16	Run Kypo in local environment		Doing	assigned
10	17	Run at least one of the Kypo games		To Do	
11	18	Test attack or game in Kypo		To Do	
12	20	Provide big picture of kypo scenario	Jakub Perdek	Done	
13	21	Desing scenario on SQL injection attack	Jakub Perdek	Done	
14	22	Describe a prototype for SQL injection scenario	Jakub Perdek	Doing	
15	23	Document Scrum Retrospective Meetings		To Do	

Obrázok 7: Export úloh z prvého šprintu

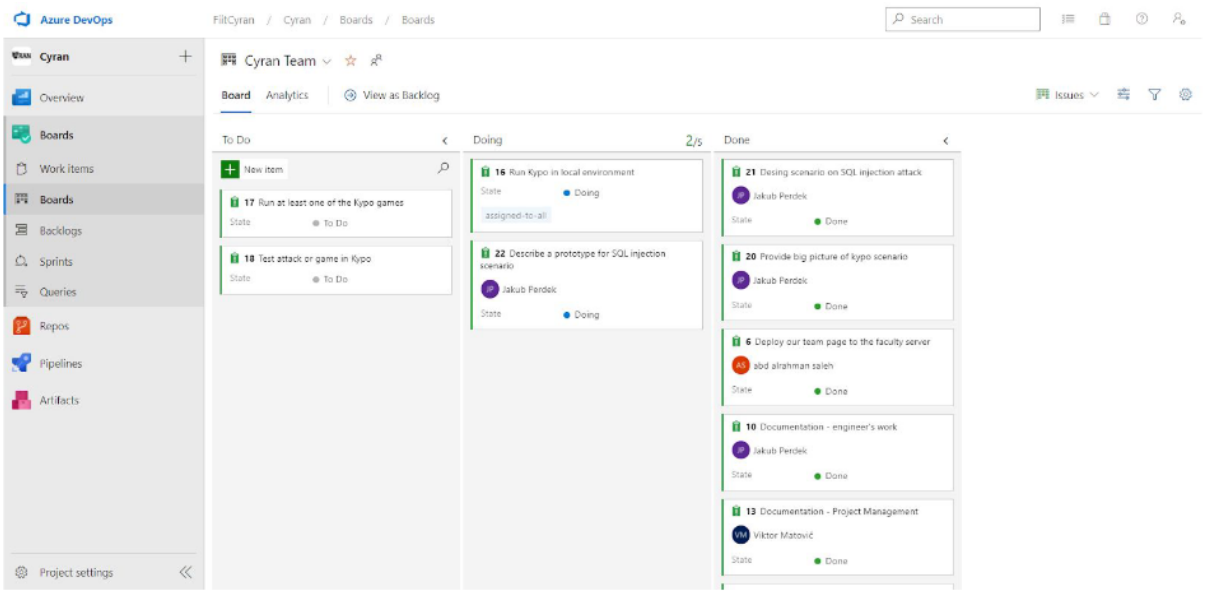
Retrospektíva prvého šprintu

Scrum tím č. 19 sa ku koncu druhého šprintu stretáva na vyhodnotenie predošlých aktivít počas Scrum retrospektívy. Jedná sa o jeden zo základných prvkov a mechanizmov používaných v metodike Scrum. Predchádzajúci šprint trval obvyklú a odporúčanú dobu: 2 kalendárdne týždne.

Dátum a čas konania	Pondelok 2. Novembra, od (cca) 20 :0 0 - 2 1:28 ho d.
Miesto konania	konferenčný hovor v General channel v Microsoft Teams
Retrospektíva za šprint:	12. Októbra - 26 . Októbra
Účastníci	Jakub Perdek, Peter Spusta, Viktor Matovič, Nikola Karakaš, Abd Saleh, Miroslav Balga
Spracovateľ	Viktor Matovič, Jakub Perdek

Tabuľka 8: Informácie o retrospektíve prvého šprintu

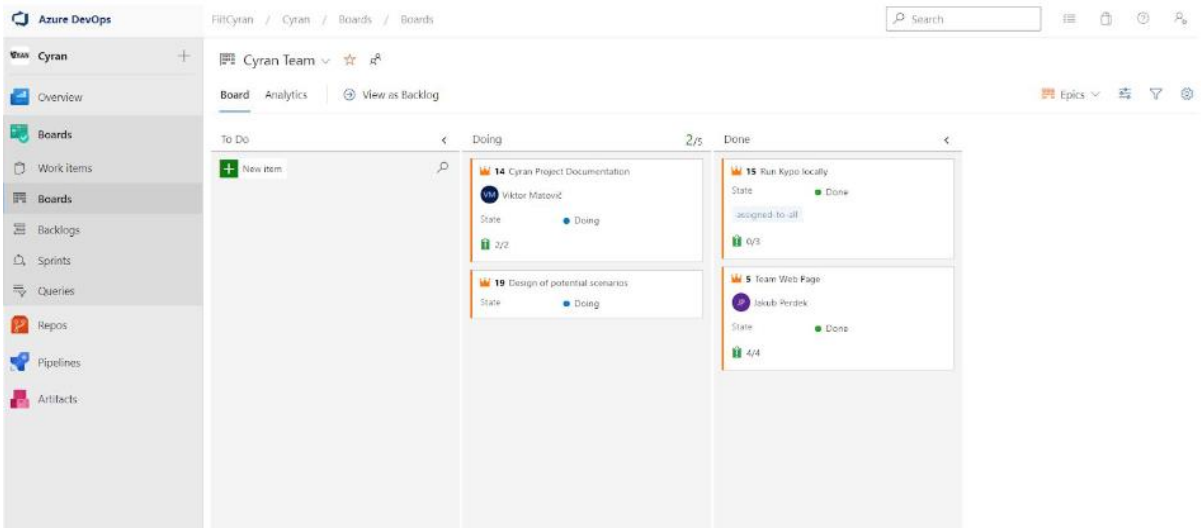
Počas prvého šprintu nebola dokončená jedna úloha. Ňou je TaskNo.16Run Kypo in local environment.ÚlohaTaskNo.22:



Obrázok 8: Nedokončená úloha spustenia Kypa

Scrum tímsa stále zabeháva, práca na téme je obmedzená pre nedostatok informácií a prístupu ku jednotlivým súčastiam rámca Kypo. Z týchto skutočností vyplýva tiež nízka

hodnota Velocity. Počas prvého šprintu sa podarilo dokončiť 2 konceptuálne najvyššie postavené položky Scrum Backlogu: vytvorenie a nasadenie prezentačnej webovej stránky tímu a spustenie jednej súčasti rámca Kypo každým členom tímu.



Obrázok 9: Splnenie dvoch najvyššie postavených položiek

Priebeh stretnutí

Stretnutie začína stručnou rekapituláciou činností, ktore sa podarilo a nepodarilo dokončiť. Členovia tímu prítomní na stretnutí sa dohadujú naorganizácii stretnutia. Dohodli sa, že každý prítomný člen tímu na stretnutí dostane slovo a oboznám iostatných s tým, s čím mal problém, čo mu chýbalo, čo by chcel zmeniť a s čím bol naopak spokojný. Miroslav tvrdí , že nemá závažné problémy s rozbehávaním súčasti Kypa ku ktorej máme jediný prístup. Viktor sa pridáva a hovorí, že už Sandbox Creator rozbehaný má, okrem toho študoval ďalšie pomocné materiály ku rámcu Kypo, ktoré sa však týkajú len samotného konceptu útokov a obrany. Nikola má rozbehaný Kypo Sandbox Creator tiež, okrem toho počas predchádzajúceho šprintu čítal články a materiály ku Kypo. Peter hovorí, že s nasadením Kypo Sandbox Creator lokálne má menšie problémy kvôli závislosti a konfigurácii Pythonu. Nastáva debata ohľadom vytvárania a používania používateľov a ich účtov pre Kypo. Jakub hovorí Petrovi, že administrátor musí byť prítomný pri každom používaní Kypa. Viktor sa pýta aký systém Kypo používa na manažment a prideľovanie rolí a oprávnení používateľom Kypa. Tak ako boli opísané problémy s ktorými sa jednotliví členovia tímu stretli tak taktíto členovia tímu komunikujú návrhy na zlepšenie. Jedným z nich je používanie techniky Scrumpoker (s pomocou webovej aplikácie) na odhadovanie náročnosti úloh pre ďalšie šprinty.

3.2 Druhý šprint

Druhý šprint bol zameraný na tvorbu nástrojov aplikácií pre penetračné testovanie webových aplikácií. Je ním Whois aplikácia umožňujúca vyhľadať záznam podľa domény a začiatok návrhu eshopu už aj s možnosťou prihlásenia. Prvý vytvorený scenár bol založený na prelamovaní slabých hesiel.

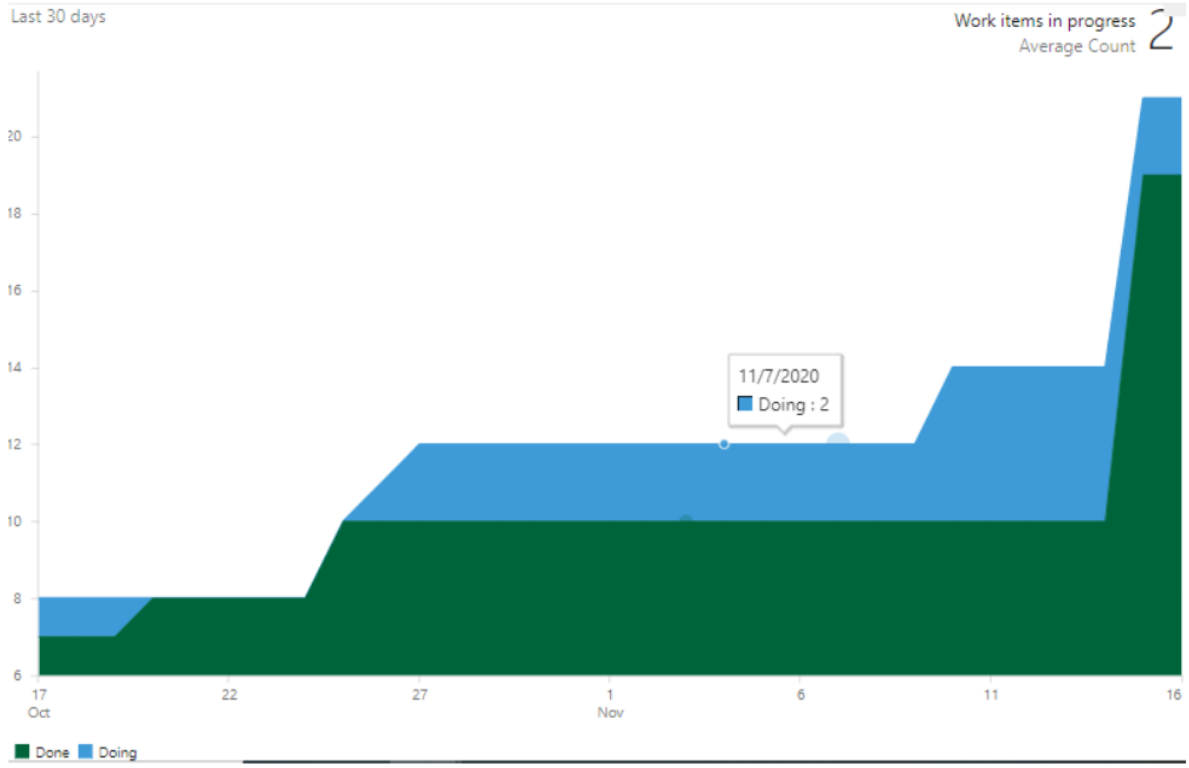
Pokrok dosiahnutý na druhom šprinte

Scrum tímu číslo 19 sa podarilo splniť všetky naplánované úlohy v šprinte číslo dva. Tím splnil celkovo 8 plánovaných úloh. V polovici šprintu sme narazili na ťažkosti, pretože sme stratili jedného člena tímu, takže teraz je nás celkovo 5.Hlavnými cieľmi v tomto šprinte bolo vytvoriť základnú verziu webovej stránky, ktorá by slúžila na zneužitie zraniteľnosti. Tím vytvoril webovú stránku elektronického obchodu, ktorá bude slúžiť ako súčasť určitých scenárov uskutočňovania kybernetických útokov. Úlohy vývoja tejto webovej stránky boli rozdelené medzi 5 členov tímu. 2 členovia za frontend, 2 členovia za backend a jeden člen za technickú dokumentáciu. Rozdelenie úloh bolo dobrovoľné. Každý mal možnosť zvoliť si, ktorú úlohu chce vykonať.Počas šprintu číslo 2 mal tím viac stretnutí pomocou komunikačnej platformy MS Teams. Členovia tímu boli tiež neustále v kontakte prostredníctvom súkromných rozhovorov.Tím bol schopný dokončiť zadané úlohy a vytvoril jednu funkčnú webovú stránku elektronického obchodu.Jeden člen tiež vykonal ďalšiu rolu a vytvoril funkčný web, ktorý slúži na zhromažďovanie ďalších informácií o webe, ktoré budú cieľom kybernetických útokov. Tento člen tímu sám prispel k vytvoreniu tejto webovej stránky.

Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (pondelok 16. 11.)	Šprint
Backend services for testing app	Viktor Matovič Peter Spusta	dokončené	šprint č. 2
Eshop - paying methods template	Jakub Perdek	dokončené	šprint č. 2
Eshop - register and login templates	Abd Saleh	dokončené	šprint č. 2
Eshop- shopping cart template	Jakub Perdek	dokončené	šprint č. 2
Eshop - delivery template	Jakub Perdek	dokončené	šprint č. 2
E shop - documentation	Nikola Karakas	dokončené	šprint č. 2
Whois application	Jakub Perdek	dokončené	šprint č. 2
Whois documentation	Jakub Perdek	dokončené	šprint č. 2

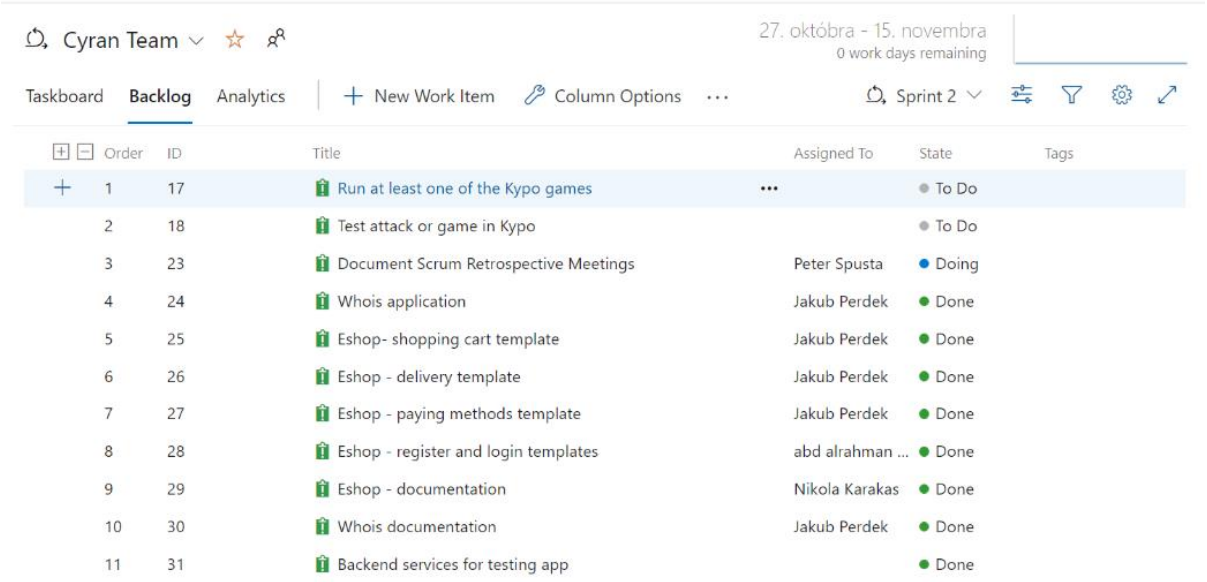
Tabuľka 9: Úlohy na druhom šprinte

Nasledujúci diagram ukazuje výkonnosť tímu. Táto metrika bude užitočná po dokončení viacerých šprintov, pretože po analýze grafu bude viditeľné ako v tíme spolupracujeme (nespolupracujúci tím bude mať v tomto pomere veľké rozdiely.



Obrázok 10: Výkonnosť tímu v druhom šprinte

Export úloh z druhého šprintu



Order	ID	Title	Assigned To	State	Tags
1	17	Run at least one of the Kypo games		To Do	
2	18	Test attack or game in Kypo		To Do	
3	23	Document Scrum Retrospective Meetings	Peter Spusta	Doing	
4	24	Whois application	Jakub Perdek	Done	
5	25	Eshop- shopping cart template	Jakub Perdek	Done	
6	26	Eshop - delivery template	Jakub Perdek	Done	
7	27	Eshop - paying methods template	Jakub Perdek	Done	
8	28	Eshop - register and login templates	abd alrahman ...	Done	
9	29	Eshop - documentation	Nikola Karakas	Done	
10	30	Whois documentation	Jakub Perdek	Done	
11	31	Backend services for testing app		Done	

Obrázok 11: Export úloh z druhého šprintu

Retrospektíva z druhého šprintu

Scrum tím č. 19 sa ku koncu druhého šprintu stretáva na vyhodnotenie predošlých aktivít počas Scrum retrospektívy. Jedná sa o jeden zo základných prvkov a mechanizmov používaných v metodike Scrum. Predchádzajúci šprint trval obvyklú a odporúčanú dobu: 2 kalendárdne týždne. V dôsledku čakanie na prístup ku kypo hrám, sme jeden týždeň medzi šprintami vynechali, respektíve sme sa venovali analýze.

Dátum a čas konania	Nedeľa 15 . Novembra, od (cca) 18 :0 0 - 2 0 :36 ho d.
Miesto konania	konferenčný hovor v General channel v Microsoft Teams
Retrospektíva za šprint:	2. Novembra - 16 . Novembra
Účastníci	Jakub Perdek, Peter Spusta, Viktor Matovič, Nikola Karakaš, Abd Saleh
Spracovateľ	Viktor Matovič, Jakub Perdek

Tabuľka 10: Informácie o retrospektíve druhého šprintu

Počas druhého šprintu sa podarilo vytvoriť aplikáciu Whois, ktorá má slúžiť v prípravnej fáze pre zber informácií. Zároveň má edukačný charakter, pretože je možné vkladať do obsahu aj informácie o potencionálnych hrozbách a viesť tak používateľa k získaniu informácií o nich. Tvorba bola nevyhnutná, pretože nasadené webové stránky v sandboxe, alebo len krátkodobo nasadené pravdepodobne nebudú vyhľadateľné štandardnými whois službami.

Zvyšná časť šprintu bola určená pre tvorbu scenárov. Konkrétne bola vytvorená webová aplikácia umožňujúca prihlásenie a registráciu používateľa. V rámci tohto šprintu boli navrhnuté ďalšie šablóny, aby mohla byť využívaná ako eshop. Neobsahuje ošetrenie hesiel, preto používateľ sa môže pokúsiť o slovníkový útok. Princíp tejto aplikácie spočíva v možnosti nastaviť slabé miesta v konfiguračnom súbore. Jednotlivý hráči potom majú za úlohu tieto miesta odhaliť. Vytvorili sme tak jednoduchý scenár prieniku do účtov používateľov. V nasledujúcich šprintoch budeme pokračovať na ďalších scenároch.

Priebeh stretnutí

Na stretnutí sa riešila podstatná otázka ohľadne prístupov k príkladom a používateľskému rozhraniu pre KYPO. Už mali byť pridelené, ale ešte stale nie sú k dispozícii. Navrhlo sa preto zhotoviť tie navrhnuté scenáre, ktoré možno nasadiť na ľubovoľný stroj v topológii hry. Webová stránka a penetračné testovanie na nej bola voľba, ktorú tím uskutočnil.

Jakub vytvoril prototyp pre Sql injekcie, ktorý by bol vhodnou súčasťou scenáru. Abd Saleh navrhol použiť Juice app. Analýza ukázala, že uvedená webová aplikácia je plná zraniteľností. Pravdepodobne neosahovala konfiguračný súbor. V rámci dohodnutého stretnutia sme sa pokúsili rozhodnúť medzi dvomi navrhnutými možnosťami. Použiť uvedenú aplikáciu alebo vytvoriť vlastnú. Problémom aplikácie bolo jej možné zneužitie pre vyriešenie scenára na základe inej chyby. Aplikácia sa nedala nakonfigurovať vypnutím nežiadaných slabých miest, a z toho dôvodu nemôže plnohodnotne byť využívaná ako učebná pomôcka pri cielene zadanej úlohe. Ďalej sme identifikovali, že používateľ môže už mať s touto aplikáciou skúsenosti, čo znamenalo aj slabší zážitok z hry. Menej dôležitým bol aj dizajn stránky, ktorý by sme chceli vylepšiť. Skupinovo sme sa zhodli na webovom riešení. Penetrovať uvedené slabé miesta by bolo rovnako časovo náročné pre ich identifikáciu.

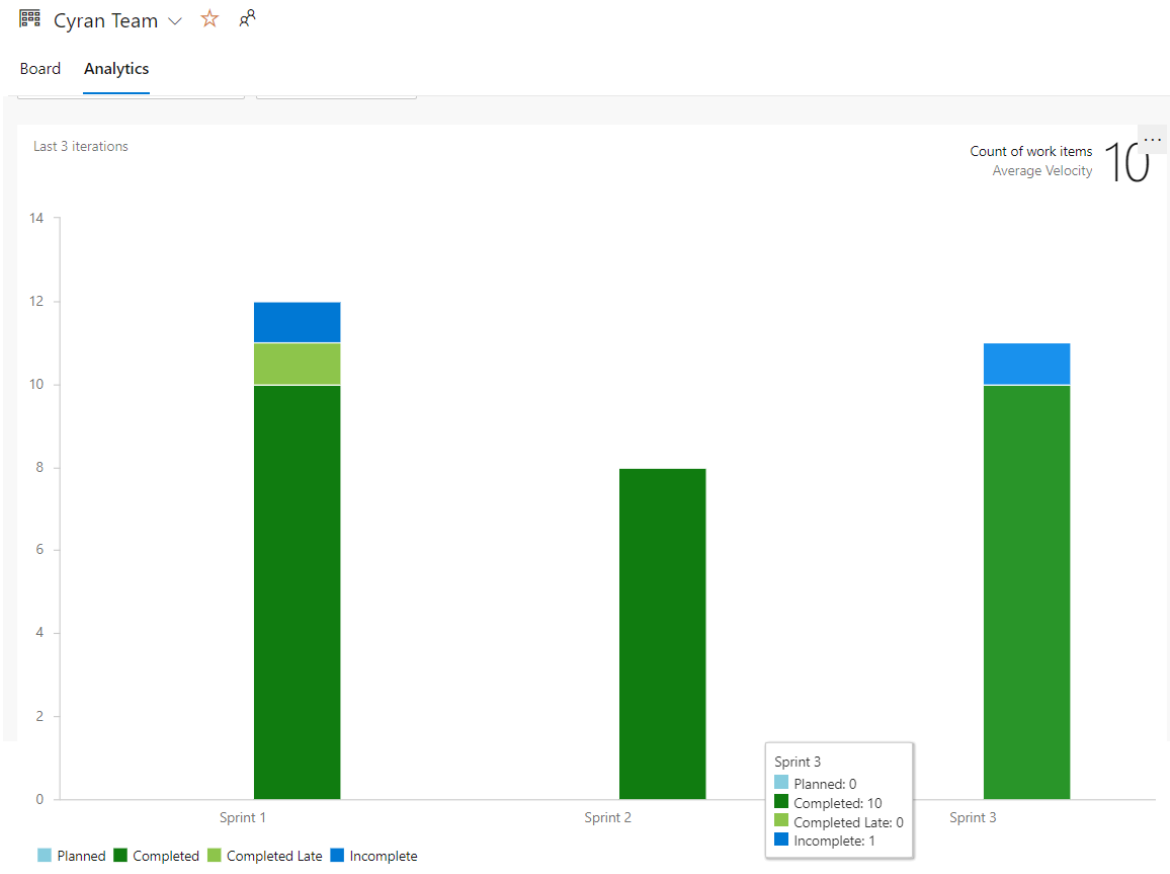
Počas ďalších dní sme vytvorili šablóny a služby, ktoré bude aplikácia využívať. Na poslednom stretnutí sme sa opäť vyjadrili k problémom a zlepšeniam. Abd Saleh navrhol skôr začať pracovať na prácach na šprinte. Jakub navrhol lepšiu komunikáciu a skoršiu odozvu na hlasovania pri plánovaní stretnutí.

3.3 Tretí šprint

Tretí šprint bol zameraný na tvorbu funkcionality eshopu zahŕňajúcu spracovanie objednávky. Boli navrhnuté šablóny pre dokončenie objednávky, načítanie ponuky produktov, tvorba funkcionality košíka, responzívnosť aplikácie a ďalšie.

Pokrok dosiahnutý na treťom šprinte

Tímu pokračoval v tvorbe webovej aplikácie. Podarilo sa dokončiť funkcionalitu košíka a integrovať základné služby pre načítanie a vloženie produktov do databázy. Pri načítaní eshopu sa tak zobrazia niektoré produkty ako ponuka. Bola vytvorená aj funkcionalita košíka pri ktorom sa položky načítajú do local storage. Cena sa automaticky prepočítava pri zmene množstva. Zmenený stav sa opäť uloží do local storage. Adresu kupujúceho s informáciami o lokalite a spôsobe doručenia rovnako ukladáme do local storage. Po výbere platobnej metódy na základe nich pripravíme objednávku. Boli vytvorené aj obrazovky pre získanie kúpených produktov.



Obrázok 12: Velocity tímu v šprinte 3

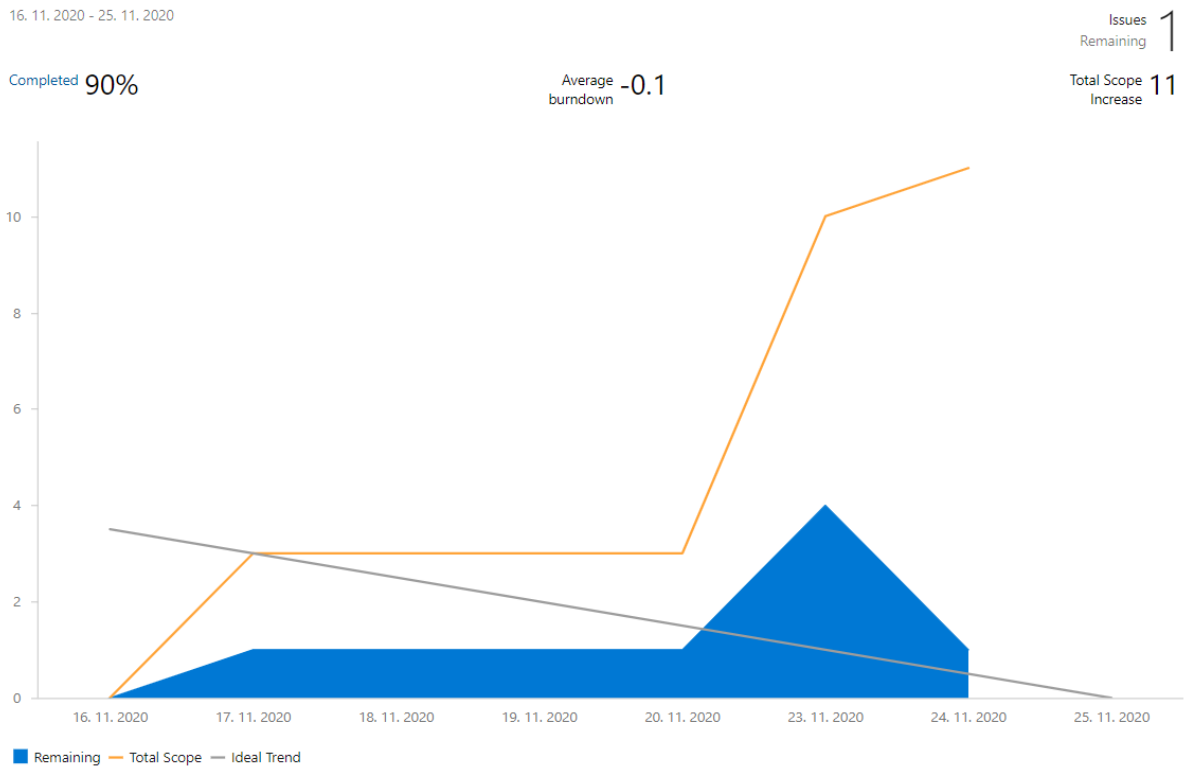
Väčší dôraz bol zameraný na tvorbu metodík, pretože so vzrastajúcim množstvom kódu bude potrebné zaviesť aj manažment revízií a verzií. Rovnako sme zdokumentovali náš spôsob komunikácie, vedenia backlogu a dokumentovania. Spojili sme potrebné dokumenty do jedného väčšieho.

Velocity sme v tomto šprinte mali dobrú, pretože boli zadané úlohy pre dokumentovanie a pokračovalo sa v zabehnutej tvorbe funkcionality eshopu z minulého šprintu. So základnou funkcionalitou tvorby objednávky bolo možné realizovať scenár ukradnutia produktov zaslaním chybných informácií na backend pomocou nástroja umožňujúceho obísť funkcionalitu frontendu. Zároveň sme dáta nechali prístupné vo verejnom adresári, čo pravdepodobne v budúcnosti chceme zmeniť, a poskytnúť len ako možnosť nastaviteľnú v konfigurácii. Splnenie všetkých úloh bolo istým spôsobom nevyhnutné, pretože dokumentácia bola nutnou podmienkou pri odovzdávaní.

Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (pondelok 16. 11.)	Šprint
Provide backend methods for finalize order	Viktor Matovič	dokončené	šprint č. 3
Create finished order template	Jakub Perdek	dokončené	šprint č. 3
Refactoring and making some web pages responsive	Abd Saleh	dokončené	šprint č. 3
Vurnerable order creation as scenario on frontend	Abd Saleh	dokončené	šprint č. 3
Create funcional shopping cart with functional services	Jakub Perdek	dokončené	šprint č. 3
Integrate frontend product management with backend in security app	Jakub Perdek	dokončené	šprint č. 3
Provide methods for managing product in backend	Viktor Matovič	dokončené	šprint č. 3
Deep documentation of eshop and revision of old one	Nikola Karakaš	dokončené	šprint č. 3
Create methodics	Jakub Perdek	dokončené	šprint č. 3
Create code review methodics	Jakub Perdek	dokončené	šprint č. 3
Create communication methodics	Jakub Perdek	dokončené	šprint č. 3
Create version management methodics	Jakub Perdek	dokončené	šprint č. 3
Create methodics of documentation	Jakub Perdek	dokončené	šprint č. 3
Finalize technical and management documentation	Jakub Perdek	dokončené	šprint č. 3

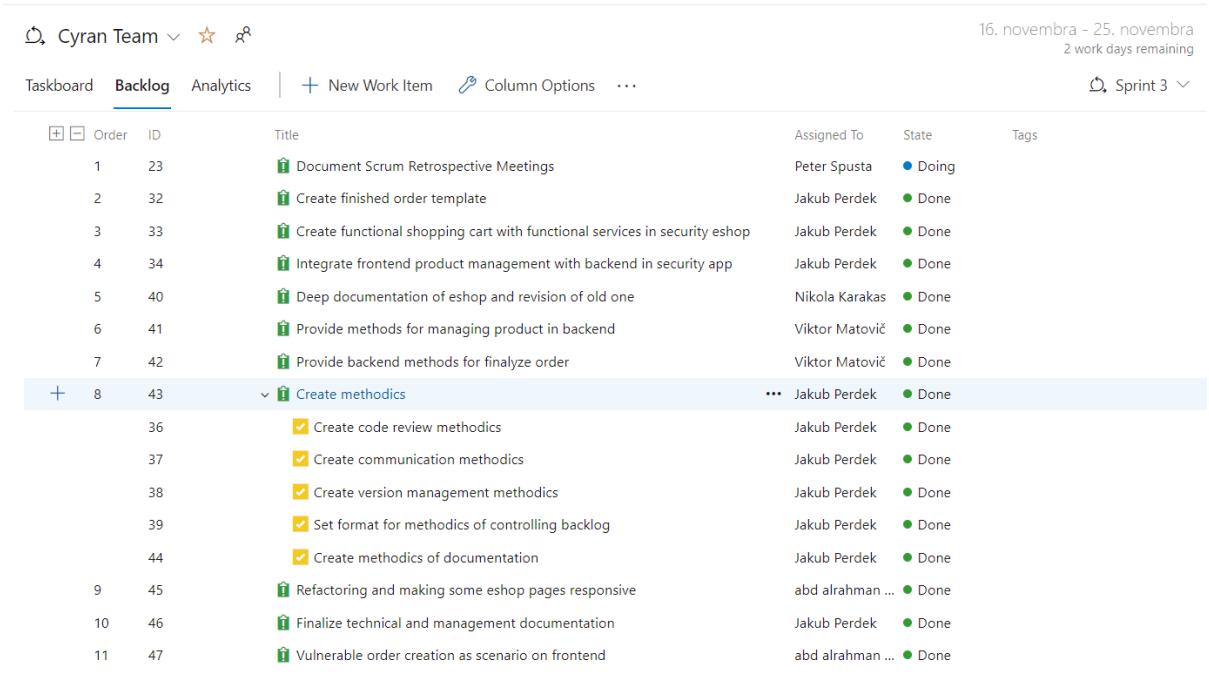
Tabuľka 11: Úlohy na treťom šprinte

Napriek meškaniu tvorby funkcionality na backende a neskorému začatiu šprintu sa podarilo scenáre dokončiť.



Obrázok 13: Výkonnosť tímu na treťom šprinte

Export úloh z tretieho šprintu



Order	ID	Title	Assigned To	State	Tags
1	23	Document Scrum Retrospective Meetings	Peter Spusta	Doing	
2	32	Create finished order template	Jakub Perdek	Done	
3	33	Create functional shopping cart with functional services in security eshop	Jakub Perdek	Done	
4	34	Integrate frontend product management with backend in security app	Jakub Perdek	Done	
5	40	Deep documentation of eshop and revision of old one	Nikola Karakas	Done	
6	41	Provide methods for managing product in backend	Viktor Matovič	Done	
7	42	Provide backend methods for finalyze order	Viktor Matovič	Done	
8	43	Create methodics	Jakub Perdek	Done	
	36	Create code review methodics	Jakub Perdek	Done	
	37	Create communication methodics	Jakub Perdek	Done	
	38	Create version management methodics	Jakub Perdek	Done	
	39	Set format for methodics of controlling backlog	Jakub Perdek	Done	
	44	Create methodics of documentation	Jakub Perdek	Done	
9	45	Refactoring and making some eshop pages responsive	abd alrahman ...	Done	
10	46	Finalize technical and management documentation	Jakub Perdek	Done	
11	47	Vulnerable order creation as scenario on frontend	abd alrahman ...	Done	

Obrázok 14: Export úloh z tretieho šprintu

Retrospektíva tretieho šprintu

Scrum tím čislo 19 sa stretol pre vyhodnotenie šprintu v ďalšej z jeho retrospektív. Zaoberal sa pokrokom na scenároch a obsahom odpovedí na preddefinované otázky.

Dátum a čas konania	Utorok 24. Novembra, od (cca) 11 :0 0 - 12 :36 ho d.
Miesto konania	konferenčný hovor v General channel v Microsoft Teams
Retrospektíva za šprint:	16. Novembra - 25 . Novembra
Účastníci	Jakub Perdek, Peter Spusta, Viktor Matovič, Nikola Karakaš, Abd Saleh
Spracovateľ	Jakub Perdek

Tabuľka 12: Informácie o retrospektíve tretieho šprintu

Priebeh stretnutí

Účastníkom boli položené nasledujúce otázky s nasledujúcimi odpoveďami:

- ***Čo sa nám podarilo vykonať?***

  Viktor: Pracoval na backende, chce poskytnúť ďalšiu funkcionalitu.

  Saleh: Pracoval na frontende, robil revíziu a refaktoring vytvoreného kódu. Šablóny, ktoré neboli responzívne urobil responzívnymi.

  Jakub: Pracoval na frontende a dokumentácii. Vytvoril šablónu pre dokončenie objednávky s možnosťou stiahnuť zakúpene súbory. Vytvoril funkcionalitu košíka s možnosťou pridávať a odoberať prvky. Ďalej zozbieral všetky dokumenty a napísal štyri metodiky k riadeniu pre ich zlúčenie do dokumentu o manažmente projektu. Vytvoril tiež diagram nasadenia a skompletizoval dokument inžinierske dielo.

  Nikola: Doplnil dokumentáciu k technickej časti eshopu.

  Peter: Po vytvorení databázy pripravil niektoré REST služby. Umožnil používať Cors hlavičky pre ladenie aplikácie.

- ***Čo sa nám nepodarilo vykonať?***

  Viktor: Viac času a funkcionality by mal venovať backendu.

Saleh: Potrebuje funkcionalitu z backendu pre tvorbu ďalšej funkcionality. Napríklad uloženie informácií o platobnej karte.
Jakub: Frontend by mohol byť používateľsky príťažlivejší formou rôznych správ pre používateľa. Dokumentácia je ale dôležitejšia.
Peter: Viac služieb, by chcel vytvoriť na backende.

- ***Aké problémy sme identifikovali alebo máme?***
  Viktor: Málo času má venovať sa backendu a TP vôbec.
  Saleh: Problém s chýbajúcou funkcionalitou na backende.
  Jakub: Word blbne a nedá sa v ňom nastaviť hierarchia nadpisov. Viacerí odpovedajú a komunikujú neskoro.
  Peter: Lepšia komunikácia v tíme.
  Nikola: Nemá problémy
- ***Čo by sme v nasledujúcom šprinte zlepšili?***
  Lepšia komunikácia a skoršie riešenie problémov je odpoveď od väčšiny z nás.
  Nikola si myslí, že to čo by sa malo zlepšiť nezáleži na tíme, ale na dodaní prístupov z MUNI.

Záver

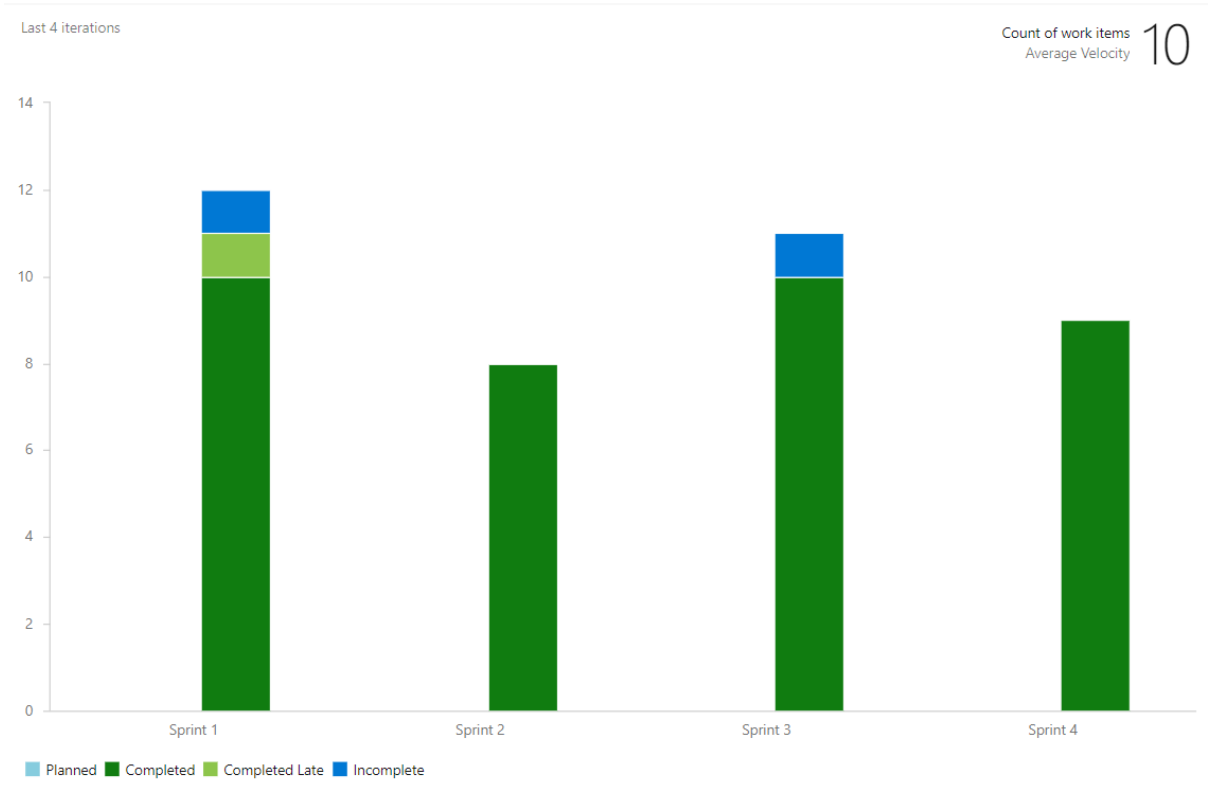
Účastníci by mali častejšie overovať poštu, a v prípade ich zaneprázdnenia dopredu informovať ostatných členov tímu.

3.4 Štvrtý šprint

Štvrtý šprint tímu 19 CYRAN začal 26. Októbra v zimnom semestri a skončil 8 decembra. Počas tohto šprintu sme sa zamerali na útok pomocou SQL injekcie a prepojenie vytvoreného scenára s ďalším scenárom. Pokiaľ budú scenáre komplexnejšie malo by to priniesť aj väčší používateľský zážitok pri objavovaní zraniteľností v aplikácii.

Pokrok dosiahnutý na štvrtom šprinte

Tím pokračoval v tvorbe webovej aplikácie. Boli vytvorené šablóny pre manažovanie eshopu pre pracovníka v obchode. Jedna z nich ponúka vytvorenie produktu s jeho názvom, cenou, množstvom a popisom. Druhá, kľúčová šablóna, je určená na vyhľadanie príslušného registrovaného používateľa, a pre prípadnú zmenu používateľovho mena alebo emailu. Jediný používateľ, ktorý by nemal byť zobrazený je samotný admin s jeho tajnou emailovou adresou. Umožnili sme ale realizovať SQL injekciu pre získanie aj tohto účtu a možnosť zmeny tejto emailovej adresy. Následne by útočník mal byť schopný nechať si vygenerovať nové heslo a dostať sa do účtu admina. Vytvorením SQL injekcie sme zapracovali ďalší scenár v našej webovej aplikácii. Spojením niektorých predchádzajúcich scenárov bude možné predstaviť komplexný scenár. Pre jeho realizáciu je ešte potrebné vytvoriť časť funkcionality v eshope. Pri tvorbe scenáru sme vytvorili novú relačnú databázu pre dáta z prihlásenia a využili JPA a Hibernate v Jave. Pre SQL injekcie sme vytvorili native query. Okrem spomínaných injekcií sme vytvorili aj kód, ktorý je odolný voči injekciám.



Obrázok 15: Velocity tímu v šprinte 4

V tomto šprinte sme realizovali úlohy zobrazené v tabuľke 12 a 13.

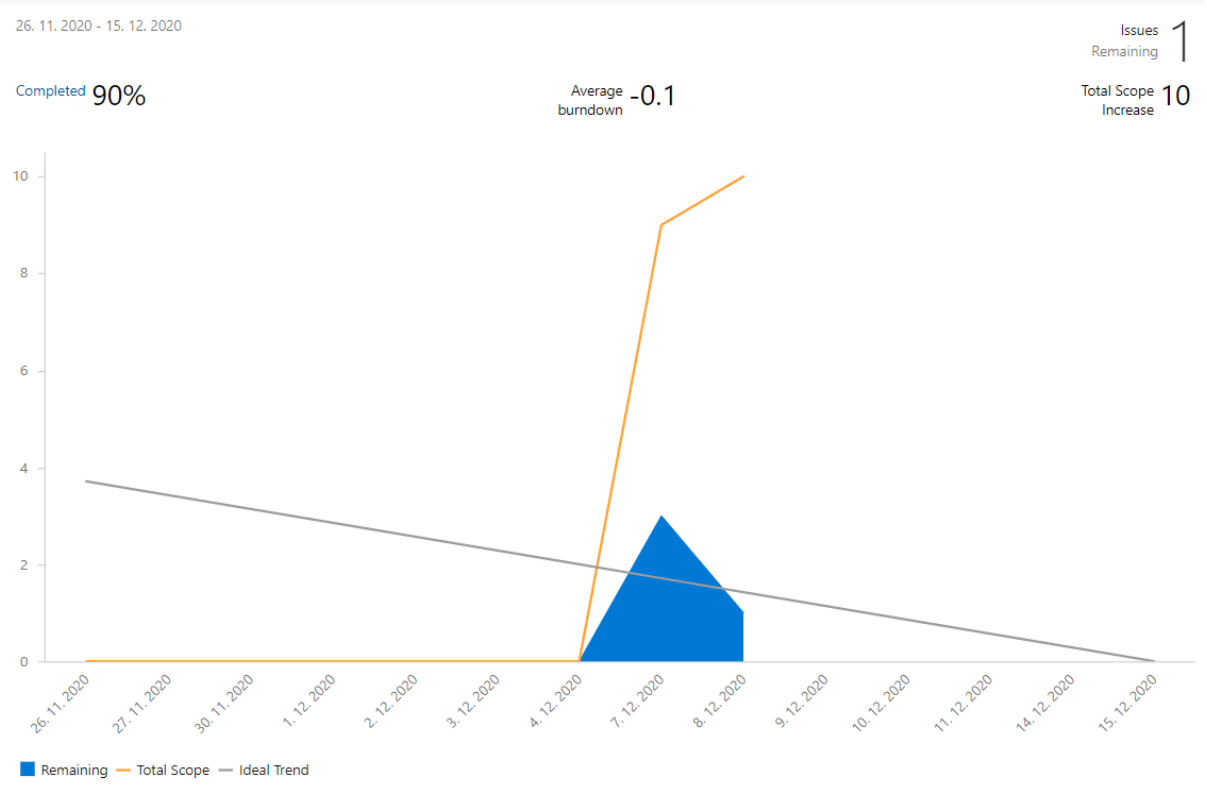
Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (pondelok 8. 12.)	Šprint
Create backend for user management	Jakub Perdek Peter Spusta	dokončené	šprint č. 4
Find and integrate database for user management and SQL injection attack	Jakub Perdek	dokončené	šprint č. 4
Create template for managing users	Jakub Perdek	dokončené	šprint č. 4
Create insert product template	Jakub Perdek	dokončené	šprint č. 4
Integrate shop management functionality on backend with frontend template	Jakub Perdek	dokončené	šprint č. 4
Documentation of eshop management	Nikola Karakaš	dokončené	šprint č. 4
Move authentication to relational SQL database	Jakub Perdek	dokončené	šprint č. 4
Make our web page more secure using secure protocol https	Abd Alrahman Saleh	dokončené	šprint č. 4
Create password regeneration and resend it to email	Jakub Perdek	dokončené	šprint č. 4
Provide backend for password resend to email	Jakub Perdek	dokončené	šprint č. 4
Provide frontend for password regeneration to email	Jakub Perdek	dokončené	šprint č. 4
Create email for eshop usage with configuration on backend	Jakub Perdek	dokončené	šprint č. 4
Create separated priviledges for admin and shop assistant in eshop		nezačaté	šprint č. 4

Tabuľka 13: Prvá časť úloh zo štvrtého šprintu

Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (pondelok 7. 11.)	Šprint
Create sprint review and retrospective	Jakub Perdek	dokončené	šprint č. 4
Run kypo parts in local environment	Abd Alrahman Saleh	dokončené	šprint č. 4

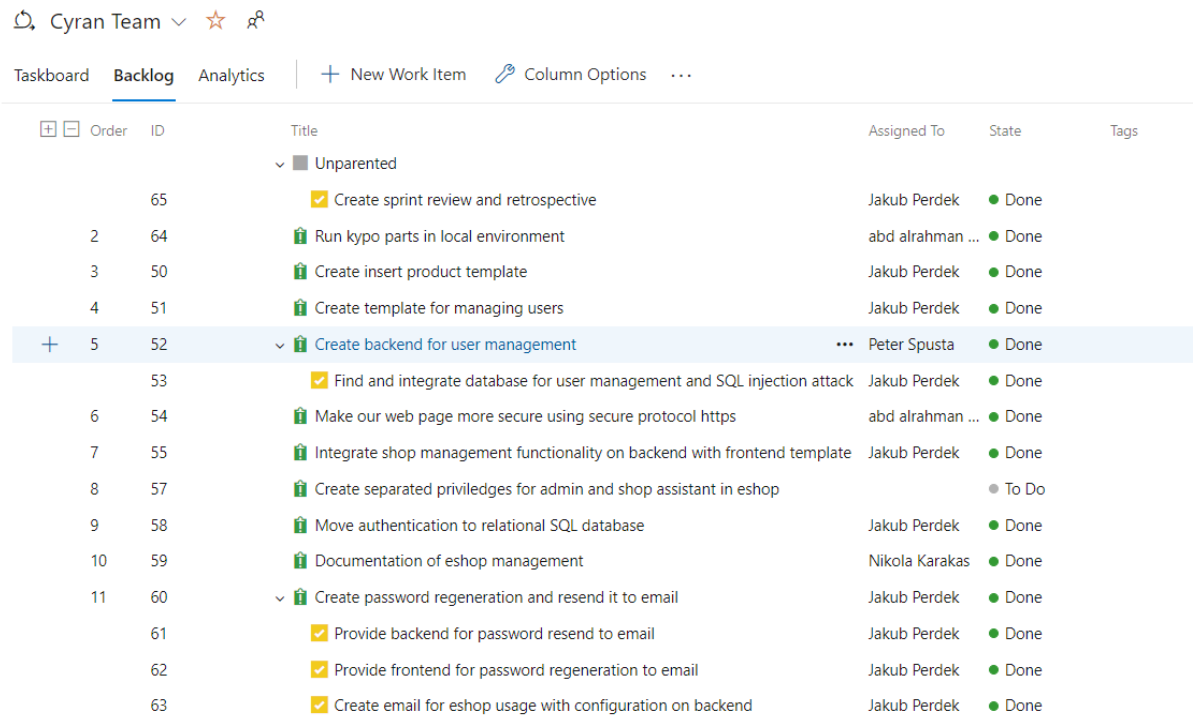
Tabuľka 14: Druhá časť úloh zo štvrtého šprintu

Okrem zhotovenia scenáru sa podarilo aj spustiť časť KYPO v lokálnom prostredí. Velocity pre tento šprint je zobrazená na obrázku 1 a je porovnateľná s predchádzajúcimi velocity ostatných šprintov. Stránka tímového projektu je vďaka v tomto šprinte zavedenému protokolu https bezpečnejšia. Tím väčšinu úloh dokončil až pred koncom šprintu, čo môžete vidieť na obrázku 2. Šprint hodnotíme úspešne, keďže bol vytvorený ďalší scenár a eshop bol rozšírený o ďalšie rozhrania. Zároveň sme tento scenár prepojili s predchádzajúcim scenárom zameraným na prelamovanie hesiel. Veríme, že zhotovená funkcionalita prinesie používateľovi čo najväčší zážitok z hry,



Obrázok 16: Výkonnosť tímu na štvrtom šprinte

Export úloh zo štvrtého šprintu



Order	ID	Title	Assigned To	State	Tags
		Unparented			
	65	Create sprint review and retrospective	Jakub Perdek	Done	
2	64	Run kypo parts in local environment	abd alrahman ...	Done	
3	50	Create insert product template	Jakub Perdek	Done	
4	51	Create template for managing users	Jakub Perdek	Done	
5	52	Create backend for user management	Peter Spusta	Done	
	53	Find and integrate database for user management and SQL injection attack	Jakub Perdek	Done	
6	54	Make our web page more secure using secure protocol https	abd alrahman ...	Done	
7	55	Integrate shop management functionality on backend with frontend template	Jakub Perdek	Done	
8	57	Create separated priviledges for admin and shop assistant in eshop		To Do	
9	58	Move authentication to relational SQL database	Jakub Perdek	Done	
10	59	Documentation of eshop management	Nikola Karakas	Done	
11	60	Create password regeneration and resend it to email	Jakub Perdek	Done	
	61	Provide backend for password resend to email	Jakub Perdek	Done	
	62	Provide frontend for password regeneration to email	Jakub Perdek	Done	
	63	Create email for eshop usage with configuration on backend	Jakub Perdek	Done	

Obrázok 17: Export úloh zo štvrtého šprintu

Retrospektíva štvrtého šprintu

Scrum tím čislo 19 sa stretol pre vyhodnotenie šprintu v ďalšej z jeho retrospektív. Zaoberal sa pokrokom na scenároch a obsahom odpovedí na preddefinované otázky.

Dátum a čas konania	Utorok 8. Decembra, od (cca) 10 :0 0 - 11 :42 ho d.
Miesto konania	konferenčný hovor v General channel v Microsoft Teams
Retrospektíva za šprint:	26. Novembra - 8. Decembra
Účastníci	Jakub Perdek, Peter Spusta, Viktor Matovič, Nikola Karakaš, Abd Saleh
Spracovateľ	Jakub Perdek

Tabuľka 15: Informácie o retrospektíve štvrtého šprintu

Priebeh stretnutí

Účastníkom boli položené nasledujúce otázky s nasledujúcimi odpoveďami:

- ***Čo sa nám podarilo vykonať?***

  Viktor: Pomohol s mapovaním na backende.

  Saleh: Dokázal spustiť časť KYPO – kypo trainings aj s backendom. Zaviedol protokol https na webovú stránku tímu.

  Jakub: Našiel a nakonfiguroval databázu. Vytvoril REST metódy pre SQL injekcie a dokončil kód na backende pre scenár. Integroval tento kód s ním vytvorenými šablónami pre manažovanie zákazníkov v obchode. Vytvoril a urobil funkčnou aj šablónu na pridávanie produktu do eshopu. Umožnil aj používateľovi preposlať email pri zabudnutí hesla na emailovú adresu zaregistrovaného zákazníka.

  Nikola: Spravil review na backende a dokumentáciu.

  Peter: Pomohol s backendom. Inicializoval ORM mapovanie pre tabuľku používateľov.

- ***Čo sa nám nepodarilo vykonať?***

  Viktor: Chcel pomôcť na backende, ale nemal čas.

  Saleh: Plánované zavedenie protokolu https na webovú stránku tímu sa mu napokon podarilo urobiť.

  Jakub: Stále nie sú prístupné niektoré závislosti pre Kypo.

  Peter: Nemal viac času na tvorbe ďalšej funkcionality na Backende.

  Nikola: Nemal problémy.

- ***Aké problémy sme identifikovali alebo máme?***

  Viktor: Má málo času.

  Saleh: Bolo by lepšie pracovať nie remotne.

  Jakub: Od začiatku tvorby nemáme session a bolo by dobré dokončiť časť s vydaním produktu po zaplatení, pretože produkty stále nie sú načítané do výslednej šablóny. Rovnako organizovanie sprint review je náročné pre rôzne harmonogramy jednotlivých členov tímu.

  Peter: Rovnako má málo času, pretože má 4 odovzdania tento týždeň.

  Nikola: Zlá koordinácia v tíme.

- ***Čo by sme v nasledujúcom šprinte zlepšili?***
  Vymedzili by sme viac času pre jednotlivé úlohy a prácu na šprinte. Komunikácia je často zdĺhavá a väčšinou len vymenujeme, čo sme spravili alebo opravujeme chyby. Je potrebné viac komunikovať nevyhnutné a podstatné záležitosti.

Záver

Komunikovať to podstatné a vymedziť primeraný čas riešeniu jednotlivých úloh.

3.5 Piaty šprint

Posledný šprint zimného semestra začal 9. decembra a skončil 18. decembra. V tomto šprinte boli dokončované a prepájané scenáre. Zlepšovali sme aj kvalitu kódu a vytvárali dokumentáciu na rôznej úrovni. Počiatočný zámer priniesť používateľovi čo najväčší používateľský zážitok sa mohol ešte viac naplniť, pretože pozornosť bola venovaná aj zlepšeniu dizajnu používateľských rozhraní.

Pokrok dosiahnutý na piatom šprinte

Tím sa v piatom šprinte sústredil na finalizáciu a prepojenie zhotovených scenárov. Snaha bola nasmerovaná aj na odstránenie slabých miest, ktoré nie sú súčasťou scenárov. Bola preto vytvorená obrana pred CSRF útokom.

Kritickým pre prepojenie scenárov bola funkcionalita manažmentu rolí. Doplnili sme preto tabuľku s možnými roľami a vytvorili rozhranie pre správcu/admina, aby ich mohol meniť. Roly boli vytvorené tri. Jedna pre používateľa s najnižšími právami. Potom roľa pracovníka v obchode, ktorý môže pridávať produkty a meniť email a meno používateľov s výnimkou admina. Správca/admin má neobmedzený prístup do všetkých vytvorených rozhraní a môže modifikovať role. Pridali sme tu aj informačnú stránku s predpripraveným tokenom pre infiltrovaného používateľa, ktorému sa konečne podarilo „dobyť“ túto stránku.

V tomto šprinte sme realizovali úlohy zobrazené v tabuľke 1 a 2.

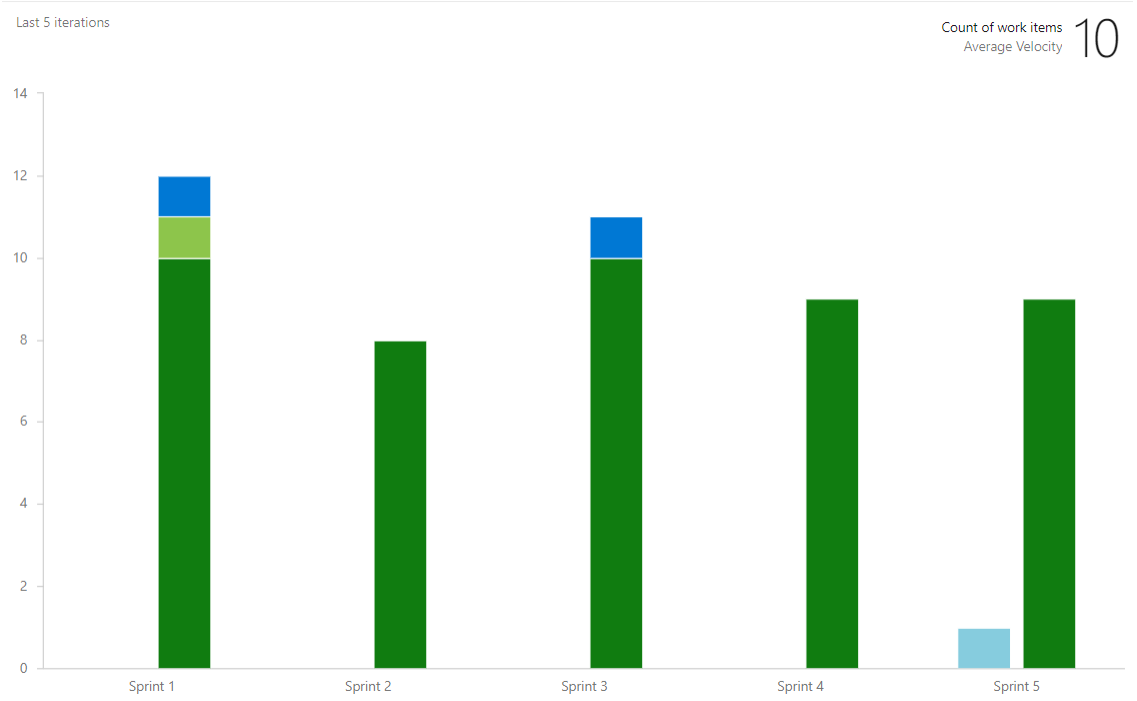
Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (piatok 18. 12.)	Šprint
Create backend for CSRF attack prevention	Viktor Matovič	dokončené	šprint č. 5
Create separated priviledges for admin and shop assistant in eshop	Jakub Perdek	dokončené	šprint č. 5
Create backend methods and prapere DB for role management	Jakub Perdek	dokončené	šprint č. 5
Create role management in frontend	Jakub Perdek	dokončené	šprint č. 5
Integrate shop management functionality on backend with frontend template	Jakub Perdek	dokončené	šprint č. 5
Create admin management board for managing roles in eshop	Jakub Perdek	dokončené	šprint č. 5
Create winner token accessible on admin board	Jakub Perdek	dokončené	šprint č. 5
Finalization of order management (download bought files, email confirmation)	Peter Spusta	dokončené	šprint č. 5
Create backend for sending bought products in payed order	Peter Spusta	dokončené	šprint č. 5
Create sprint progress	Jakub Perdek	dokončené	šprint č. 5
Create informative feedback to customer on frontend	Jakub Perdek	dokončené	šprint č. 5

Tabuľka 16: Prvá časť úloh z piateho šprintu

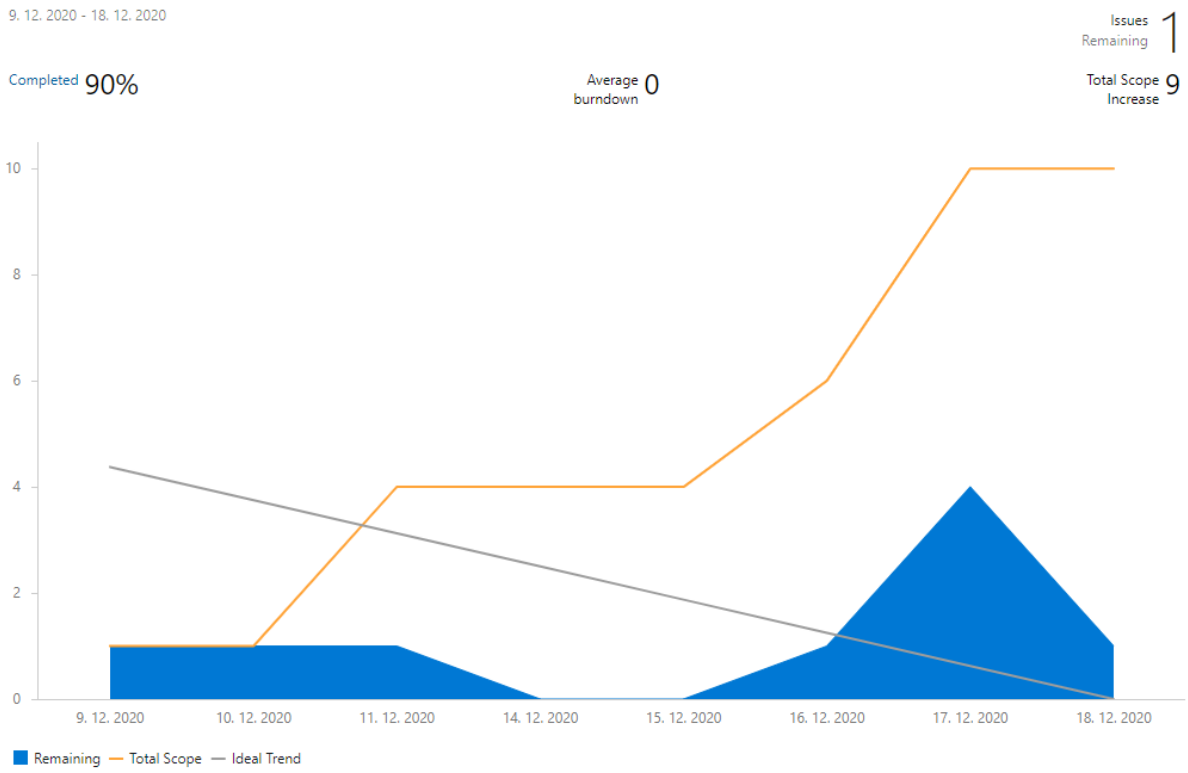
Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (piatok18. 12.)	Šprint
Create Javadoc documentation of backend	Jakub Perdek Nikola Karakaš	dokončené	šprint č. 5
Create guide for users with scenarios	Jakub Perdek	dokončené	šprint č. 5
Unit tests for backend HTTP requests	Viktor Matovič	dokončené	šprint č. 5
Create form validation on frontend	Abd Alrahman Saleh	dokončené	šprint č. 5
Refactoring code on frontend	Abd Alrahman Saleh	nezačaté	šprint č. 5

Tabuľka 17: Druhá časť úloh z piateho šprintu

Ďalšou vykonanou prácou bolo zlepšovanie dizajnu na frontende a dokumentovanie doposiaľ vytvorenej funkcionality ako aj kódu samotného. Bola vytvorená aj dokumentácia s opisom implementovaných scenárov. Okrem tejto dokumentácie sme vypracovali aj JavaDoc dokumentáciu metód na backende. Zlepšenie UX pozostávalo z implementácie spätnej väzby pre používateľa pri rôznych úkonoch na stránke. Napríklad pri pridaní produktu do košíka.



Obrázok 18: Velocity tímu v šprinte 5



Obrázok 19: Výkonnosť tímu na piatom šprinte

Export úloh z piateho šprintu

Cyran Team

Taskboard | Backlog | Analytics | New Work Item | Column Options | ...

Order	ID	Title	Assigned To	State	Tags
		Unparented			
	80	Create sprint progress	Jakub Perdek	Done	
2	57	Create separated priviledges for admin and shop assistant in eshop	Jakub Perdek	Done	
	66	Create backend methods and prapere DB for role management	Jakub Perdek	Done	
	67	Create role management in frontend	Jakub Perdek	Done	
3	68	Create backend for CSRF attack prevention	Viktor Matovič	Done	
4	69	Create admin management board for managing roles in eshop	Jakub Perdek	Done	
	70	Create winner token accessible on admin board	Jakub Perdek	Done	
5	71	Finalization of order management (download bought files, redirrects)		Done	
	72	Create backend for sending bought products in payed order	Peter Spusta	Done	
	73	Insert bought products to associated template	Jakub Perdek	Done	
6	74	Create informative feedback to customer on frontend	Jakub Perdek	Done	
7	75	Create Javadoc documentation of backend		Done	
8	76	Unit tests for backend HTTP requests	Viktor Matovič	Done	
9	77	Create form validation on frontend	abd alrahman ...	Done	
10	78	Refactoring code on frontend	abd alrahman ...	To Do	
11	79	Create guide for users with scenarios	Jakub Perdek	Done	

Obrázok 20: Export úloh z piateho šprintu

Retrospektíva z piateho šprintu

Scrum tím čislo 19 sa stretol pre vyhodnotenie šprintu v poslednej z jeho retrospektív za zimný semester. Zaoberal sa výsledkami prepájania scenárov do komplexných celkov a obsahom odpovedí na preddefinované otázky.

Dátum a čas konania	Utorok 17. Decembra, od (cca) 20 :0 0 - 21:12ho d.
Miesto konania	konferenčný hovor v General channel v Microsoft Teams
Retrospektíva za šprint:	9. Decembra - 18. Decembra
Účastníci	Jakub Perdek, Peter Spusta, Viktor Matovič, Nikola Karakaš, Abd Saleh
Spracovateľ	Jakub Perdek

Tabuľka 18: Informácie o retrospektíve piateho šprintu

Priebeh stretnutí

Účastníkom boli položené nasledujúce otázky s nasledujúcimi odpoveďami:

- ***Čo sa nám podarilo vykonať?***

  Viktor: Vytvoril funkcionalitu na obranu pred CSRF útokom pre naše formuláre. Pracoval na funkcionalite jednotkových testov.

  Saleh: Spravil refaktoring kódu a validáciu formulárov.

  Jakub: Vytvoril šablónu pre manažovanie rolí používateľov pre používateľa s oprávnením admin. Umožnil aby bolo možné zmeniť role jednotlivým registrovaným používateľom cez toto rozhranie. Vytvoril finálnu šablónu poskytujúcu víťazný kód infiltrovanému používateľovi.

  Nikola: Vypracoval dokumentáciu pre eshop a vygeneroval JavaDoc.

  Peter: Umožnil odoslanie kúpených produktov používateľovi na frontend a spravoval databázu.

- ***Čo sa nám nepodarilo vykonať?***

  Viktor: Lepšie keby mal viac času pracovať viac na úlohách tímového projektu.

  Saleh: Nemal veľa času.

Jakub: Nedokončil polovicu JavaDoc dokumentácie.
Peter: Neurobil reštruktualizáciu kódu.
Nikola: Nič čo by mal naplánované sa mu nepodarilo nevykonal.

- ***Aké problémy sme identifikovali alebo máme?***
  Viktor: Znovu má málo času, mal náročný týždeň.
  Saleh: Málo času.
  Jakub: Mali by sme viac pracovať na funkcionalite a scenároch.
  Peter: Problémy s notebookom a málo času.
  Nikola: Nemal problémy. Rovnako mal málo času.

- ***Čo by sme v nasledujúcom šprinte zlepšili?***
  Zlepšenie práce na KYPO a viac kontaktovať univerzitných študentov majúcich projekty s KYPO. Nájsť niekoho kto zoberie zodpovednosť za dodanie KYPO. Konečne dostať Projekt Backlog.

Záver

Usilovať sa aby bola zabezpečená KYPO funkcionalita a projektový Backlog.

3.6 Šiesty šprint

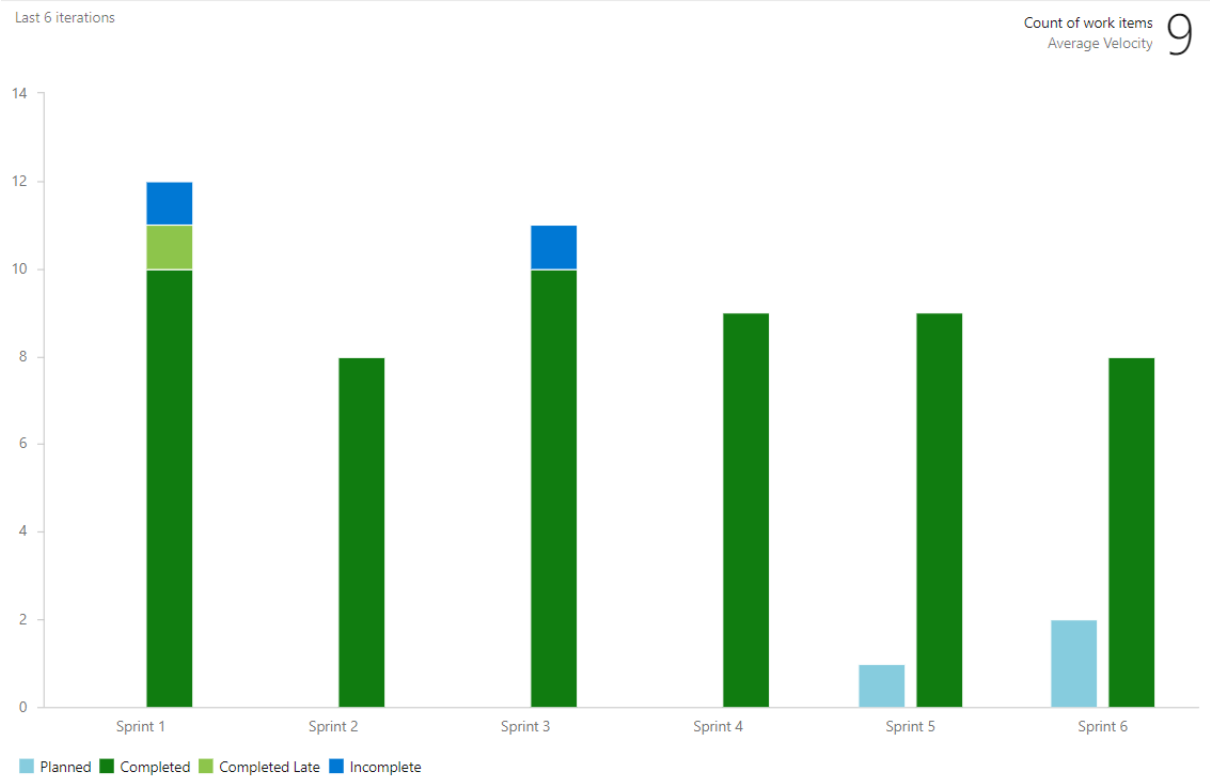
Šiesty šprint tímu 19 CYRAN začal 22. februára v letnom semestri a skončil 7. marca. Počas šprintu sme sa zaoberali prípravami pre používateľský prieskum. Vytvárali sme preto dotazníky, lepší prehľad metodík scenárov, ale aj samotnú kontajnerizáciu a možnosti nasadenia na cloud rovnako ako do lokálneho prostredia.

Pokrok dosiahnutý na šiestom šprinte

Predmetom šiesteho šprintu bola príprava pre vyhodnotenie zhotovenej aplikácie. Ešte stále sme nezískali prístup ku KYPU a OpenStack-u, ktoré sme chceli prepojiť s našou aplikáciou. Hľadali sme preto ďalšie možnosti nasadenia aplikácie spolu so spôsobom

zozbierania a vyhodnotenia spätnej väzby zhotovenej aplikácie.

Dozvedeli sme sa, že dostupné prostriedky na nasadenie v školskom prostredí sú nepostačujúce, preto sme začali analyzovať ďalšie možné spôsoby. Saleh analyzoval možnosti cloudov. Zistil, že nasadenie by bolo možné na študentský účet, ale len obmedzený čas. Rozhodli sme sa pripraviť aplikáciu pre lokálne nasadenie. Jakub zhotovil docker pipelinu pre frontent, backend a databázu. Okrem toho bol potrebný samostatný Docker súbor pre každú časť. Bolo potrebné zabezpečiť komunikáciu medzi kontajnerom s databázou a backendom, prípadne backendu s frontendom.

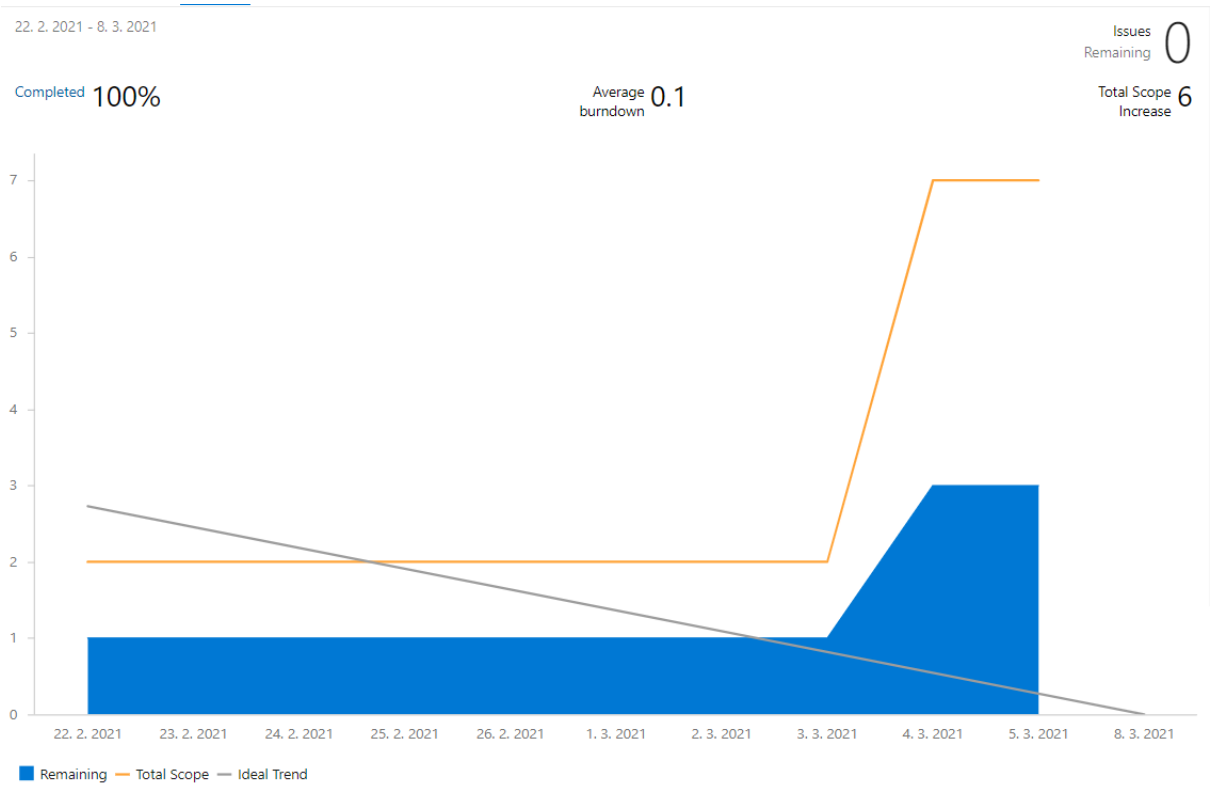


Obrázok 21: Velocity tímu v šprinte 6

Okrem toho bolo potrebné predpripraviť dáta v databáze pre scenáre. Inicializáciu databázy vykonáva Hibernate, s výnimkou vloženia ďalších dát pri inicializácii. Viktor preto vytvoril inicializačný skript, aby bolo možné ihneď po stiahnutí aplikáciu plnohodnotne využívať.

Následne boli kontajnery po vykonaní buildu uploadnuté na Docker Hub pre ich sprístupnenie budúcim používateľom. Po stiahnutí sa už kontajnery nebuildujú a môžu sa ihneď používať. V niektorých prípadoch sa ešte pri štarte spustí inicializačný skript pre databázu.

Okrem hlavnej aplikácie sme vytvorili kontajnere prostredníctvom Dockera aj pre Whois aplikáciu. V tomto prípade bolo potrebné vyriešiť inicializáciu databázy a jej komunikáciu s NodeJS serverom. Výsledný build bol opäť uploadnutý na Docker Hub.



Obrázok 22: Výkonnosť tímu v šiestom šprinte

Velocity tímu dosahuje priemerné hodnoty, ale vzhľadom na typy úloh späté s kontajnerizáciou a integrovaním s jednotlivými obrazmi sme vykonali náročnú a dôležitú časť v projekte. Riešili sme rôzne chyby od písania inicializačných skriptov pre databázu, integráciu databázy s backendom až po ladenie problémov, ktoré sme odhalili ladením aplikácie. Velocity tímu počas šiesteho šprintu je na obrázku 1. Začali sme aj prípravy opisu scenárov. Nikola vypracoval opisy metodík v scenároch a snažil sa ich priblížiť používateľovi. Peter zhotovil dotazník pre spätnú väzbu od používateľov. Zaujímala nás náročnosť scenárov, dĺžka ich hrania a v neposlednom rade aj návod na zlepšenie. Prípadné ďalšie úvahy analyzujeme a dopracujeme v ďalších šprintoch.

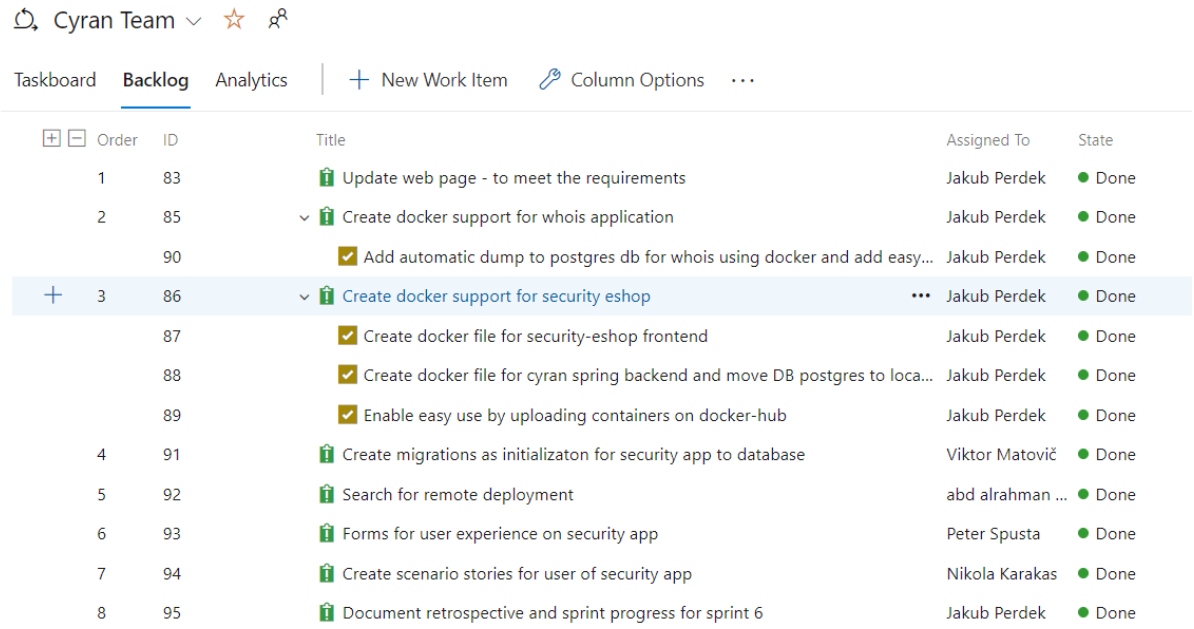
Okrem samotnej prípravy na používateľský prieskum a ďalšie vylepšenia aplikácie sme vylepšili aj naše webové sídlo o ďalšie efekty. Celkovo sme splnili všetky zadané úlohy počas dvoch týždňov šprintu. Výkonnosť tímu v šiestom šprinte je zobrazená na obrázku 22.

V tomto šprinte sme realizovali úlohy zobrazené v tabuľke 1 a 2.

Pridelená úloha	**Zodpovedný riešiteľ**	**Aktuálny stav rozpracovania (nedeľa 8. 3.)**	**Šprint**
Create migrations as initializaton for security app to database	Viktor Matovič	dokončené	šprint č. 6
Update web page - to meet the requirements	Jakub Perdek Abd Alrahman Saleh	dokončené	šprint č. 6
Create docker support for whois application	Jakub Perdek	dokončené	šprint č. 6
Add automatic dump to postgres db for whois using docker and add easy setup using docker hub	Jakub Perdek	dokončené	šprint č. 6
Create docker support for security eshop	Jakub Perdek	dokončené	šprint č. 6
Create docker file for security-eshop frontend	Jakub Perdek	dokončené	šprint č. 6
Create docker file for cyran spring backend and move DB postgres to local usage	Jakub Perdek	dokončené	šprint č. 6
Enable easy use by uploading containers on docker-hub	Jakub Perdek	dokončené	šprint č. 6
Search for remote deployment	Abd Alrahman Saleh	dokončené	šprint č. 6
Forms for user experience on security app	Peter Spusta	dokončené	šprint č. 6
Create scenario stories for user of security app	Nikola Karakaš	dokončené	šprint č. 6
Document retrospective and sprint progress for sprint 6	Jakub Perdek	dokončené	šprint č. 6

Tabuľka 19: Úlohy zo šiesteho šprintu

Export úloh zo šiesteho šprintu



Order	ID	Title	Assigned To	State
1	83	Update web page - to meet the requirements	Jakub Perdek	Done
2	85	Create docker support for whois application	Jakub Perdek	Done
	90	Add automatic dump to postgres db for whois using docker and add easy...	Jakub Perdek	Done
3	86	Create docker support for security eshop	Jakub Perdek	Done
	87	Create docker file for security-eshop frontend	Jakub Perdek	Done
	88	Create docker file for cyran spring backend and move DB postgres to loca...	Jakub Perdek	Done
	89	Enable easy use by uploading containers on docker-hub	Jakub Perdek	Done
4	91	Create migrations as initializaton for security app to database	Viktor Matovič	Done
5	92	Search for remote deployment	abd alrahman ...	Done
6	93	Forms for user experience on security app	Peter Spusta	Done
7	94	Create scenario stories for user of security app	Nikola Karakas	Done
8	95	Document retrospective and sprint progress for sprint 6	Jakub Perdek	Done

Obrázok 23: Export úloh zo šiesteho šprintu

Retrospektíva šiesteho šprintu

V novom semestri scrum tím 19 zahájil stretnutie pre vykonanie retrospektívy po dvojtýždňovom šprinte. Zaoberal sa výsledkami prípravy pre používateľský prieskum zahŕňajúci nasadenie aplikácií a tvorbu formulárov ako aj ďalších zlepšení.

Dátum a čas konania	Nedeľa 7. Marca, od (cca) 18 :0 0 - 19:34ho d.
Miesto konania	konferenčný hovor v General channel v Microsoft Teams
Retrospektíva za šprint:	22. Februára - 7. Marca
Účastníci	Jakub Perdek, Peter Spusta, Viktor Matovič, Nikola Karakaš, Abd Saleh
Spracovateľ	Jakub Perdek

Tabuľka 20: Informácie o retrospektíve šiesteho šprintu

Priebeh stretnutí

Účastníkom boli položené nasledujúce otázky s nasledujúcimi odpoveďami:

- ***Čo sa nám podarilo vykonať?***

  Viktor: Vytvoril inicializáciu databázy na backende pre bezproblémový chod scenárov pri použití novej databázy.

  Saleh: Updatol stránku sídla a analyzoval možnosti nasadenia v cloude.

  Jakub: Pripravil docker pipeliny a zabezpečil lokálne nasadenie. Následne s prístupnil riešenie na Docker Hube pre jednoduchšie spustenie a spravovanie závislostí.

  Nikola: Pracoval na scenároch pre používateľa.

  Peter: Vypracoval formuláre pre používateľský prieskum.

- ***Čo sa nám nepodarilo vykonať?***

  Viktor: Chcel by dômyselnejšiu konfiguráciu pre backend.

  Saleh: Zamýšľal aplikáciu aj nasadiť.

  Jakub: Chcel zlepšiť funkcionalitu aplikácie a začať pracovať na ďalších scenároch a odstránení chýb.

  Peter: Chcel analyzovať a zapracovať vylepšenie pre aplikáciu obsahujúce funkcionalitu KYPA. Na to však je potrebné vybudovať niečo KYPU podobné.

  Nikola: Hádanie hesla nie je veľmi dobré aj napriek tomu, že heslo je jednoduché. Pôvodne to mal byť slovníkový útok, jeho síce možné zrealizovať ale bezplatný Burpsuit ho prostredníctvom anguláru nepodporuje – odchytáva len http žiadosť na backend, na ktorý potom dokáže posielať úroky, aspoň podľa mojej analýzy. (odpoveď od Jakuba).

- ***Aké problémy sme identifikovali alebo máme?***

  Viktor: Odladenie inicializácie databázy veľmi trvá.

  Saleh: Prekrývajúce úlohy spôsobili zasahovanie do časti jeho úlohy.

  Jakub: Aplikácia môže mať viac bezpečnostných slabín ako má scenárov.

  Peter: Hľadá záujemcov o scrum mastera. Nakoniec sa ale tejto roly ujíma.

  Nikola: Potreboval veľa komunikovať o častiach s aplikáciou. Už sa dozvedel podrobnosti.

- ***Čo by sme v nasledujúcom šprinte zlepšili?***

  Komunikáciu detailných krokov pre jednotlivé úlohy.
  Zabezpečili neprelínanie úloh alebo lepšie zabezpečili spoluprácu na ich plnení.

Záver

Je potrebné podrobnejšie si opísať ako konkrétnu úlohu budeme riešiť, aby nedošlo k vzájomným zásahom do práce ostatných, a mohlo sa efektívne paralelne pracovať na funkcionalite. Pri identifikácii prekrývajúcich sa častí radšej buď zvoliť spoluprácu alebo sa dohodnúť kto bude riešiť inú úlohu.

3.7 Siedmy šprint

Šprint číslo 7 tímu 19 CYRAN začal 8. marca v letnom semestri a skončil 7. marca. Obsahom šprintu bolo testovanie zhotovených Docker obrazov. Keďže sme identifikovali niekoľko chýb a závislostí pri builde od operačného systému a problémom s príkazom pull na Ubuntu spravili sme revíziu Docker kontajnerov a následné ich ďalšie testovanie. Odhalené chyby sme sa buď snažili vyriešiť alebo pripraviť ich riešenie pri samotnom testovaní.

Pokrok dosiahnutý na siedmom šprinte

Siedmy šprint sme zamerali na zlepšovanie aplikácie a jej ďalšie testovanie spolu s prípravou na lokálne nasadenie u používateľov. Cieľom bolo spraviť aplikáciu atraktívnejšiu a vypracovať automatické, prípadne semi-automatické spôsoby, ktorými si používateľ môže pomôcť pri prelamovaní hesiel.

Prírastkom sú ďalšie dve služby zamerané na použitie Bcrypt spolu s popisom ich významu. Na ich tvorbe pracoval Jakub. Ich význam spočíva v možnosti vygenerovať hash v rôznom čase a tým získať predstavu ako tento algoritmus funguje. Používateľ podľa zvolenej hodnoty môže očakávať získanie výsledkov od niekoľkých minút až približne do jednej hodiny. Zároveň sa môže dozvedieť o výhodách soli a spomenutom Bcrypte.

Druhou službou môže overiť, či jeho hash reprezentuje vložený text. Význam služieb spočíva s použitím BurpSuitu, kde postačuje zistiť hashe zakódované Bcryptom používateľov

a následne znovu použiť BurpSuite na tejto druhej zlužbe so zoznamom týchto hesiel pre ich porovnanie. Celý postup sme otestovali a spísali v používateľskej príručke.

Testovanie pre lokálne nasadenie bolo nevyhnutnou súčasťou šprintu. Nikola testoval aplikáciu na svojom Windowse. Identifikovanými problémami lokálneho nasadenia bola záťaž Dockera, spustená databáza Postgres a problém s novým zariadením pri odosielaní emailu. Ako riešenie sme museli vypnúť lokálnu databázu a pre email sme hľadali spôsoby ako vypnúť overenie zariadenia, prípadne použiť menej zabezpečený email alebo emailový server. V prípade emailu sme nenašli vhodnú službu a na G-maily sa nastavenie nepodarilo uskutočniť. Emailový server, spustený v Dockeri, pre úspešné odoslanie správy požadoval prístup do emailovej adresy odosielateľa. Pre lokálne nasadenie bude preto potrebné namiesto odosielania emailu správu zobraziť používateľovi priamo, alebo týchto používateľov explicitne v G-maily povoliť.

Testovanie na linuxe dopadlo neúspešne. Saleh identifikoval problém s príkazom docker-compose pull, ktorý nestiahol uvedené obrazy. Zároveň ostatní členovia mu pomáhali pri builde v uvedenom prostredí. Identifikovali sme nepostačujúci identifikátor host.docker.internal pre linux, pretože ten údajne funguje iba pre operačné systémy Mac a Windows. Pri operačných systémoch typu linux, ako je naše testované Ubuntu, je potrebné použiť konkrétnu IP adresu. Po jej použití bola aplikácia funkčná. Zároveň došlo k spojeniu frontendu a backendu pre ľahšiu tvorbu buildov v osobitnom repozitári.
V tomto šprinte sme realizovali úlohy zobrazené v tabuľkách 21 a 22.

Nakoniec Saleh zaviedol spoločnú sieť pridaním adaptérov pre bezproblémové použitie a build aj na rôznych operačných systémoch. Jakub ešte dodatočne toto riešenie otestoval na Windowse. O úpravu dotazníka pri zapracovaní požiadaviek produkt ownera sa postaral Peter. Pre každý scenár doň zapracoval otázky pre získanie čo najväčšieho počtu informácií od používateľov.

V šprinte sa podarilo vypracovať množstvo funkcionality a otestovať lokálne nasadenie pre rôzne operačné systémy. Zlepšenie sme realizovali pridaním už spomenutého semi-automatického spôsobu slovníkového útoku, rozšírením dotazníka, prepracovaním používateľskej príručky a ďalších menších úprav. Snažili sme sa vyriešiť identifikované problémy, aby pri lokálnom použití nevznikli zásadné problémy. Saleh ešte dodatočne obnovil certifikát pre používanie bezpečného protokolu HTTPS na našom webovom sídle. Naša efektivita bola v porovnaní s ostatnými šprintmi vysoká.

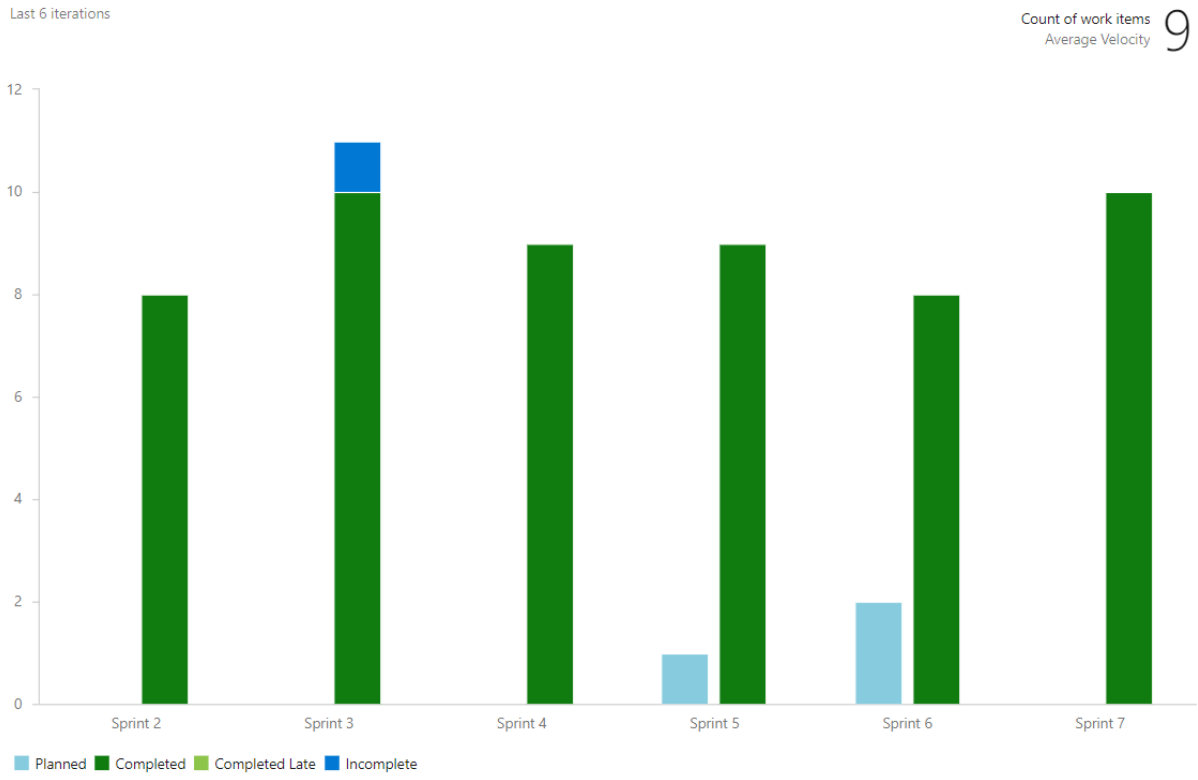
Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (nedeľa 21. 3.)	Šprint
Create BCrypt services and tutorial template	Jakub Perdek	dokončené	šprint č. 7
Create BCrypt encryptor service	Jakub Perdek	dokončené	šprint č. 7
Create BCrypt validator service	Jakub Perdek	dokončené	šprint č. 7
Present information about salt and Bcrypt as intro to BCrypt services	Jakub Perdek	dokončené	šprint č. 7
Document dictionary attack and add it to user guide	Jakub Perdek	dokončené	šprint č. 7
Run and test application using docker on Windows	Nikola Karakaš	dokončené	šprint č. 7
Test docker images on Linux	Abd Alrahman Saleh	dokončené	šprint č. 7
Find email which not check device or observe docker containers for smtp servers	Jakub Perdek	dokončené	šprint č. 7
Enhance user forms to gain their experiences from using application	Peter Spusta	dokončené	šprint č. 7
Create build for easier deployment on linux machines	Abd Alrahman Saleh Jakub Perdek Viktor Matovič	dokončené	šprint č. 7
Apply the same network rules for docker images	Abd Alrahman Saleh	dokončené	šprint č. 7
Create sprint retrospective and sprint progress for sprint 7	Jakub Perdek	dokončené	šprint č. 7
Test scenarios on new version of docker-compose version of our app on Windows	Jakub Perdek	dokončené	šprint č. 7

Tabuľka 21: Úlohy zo siedmeho šprintu časť 1

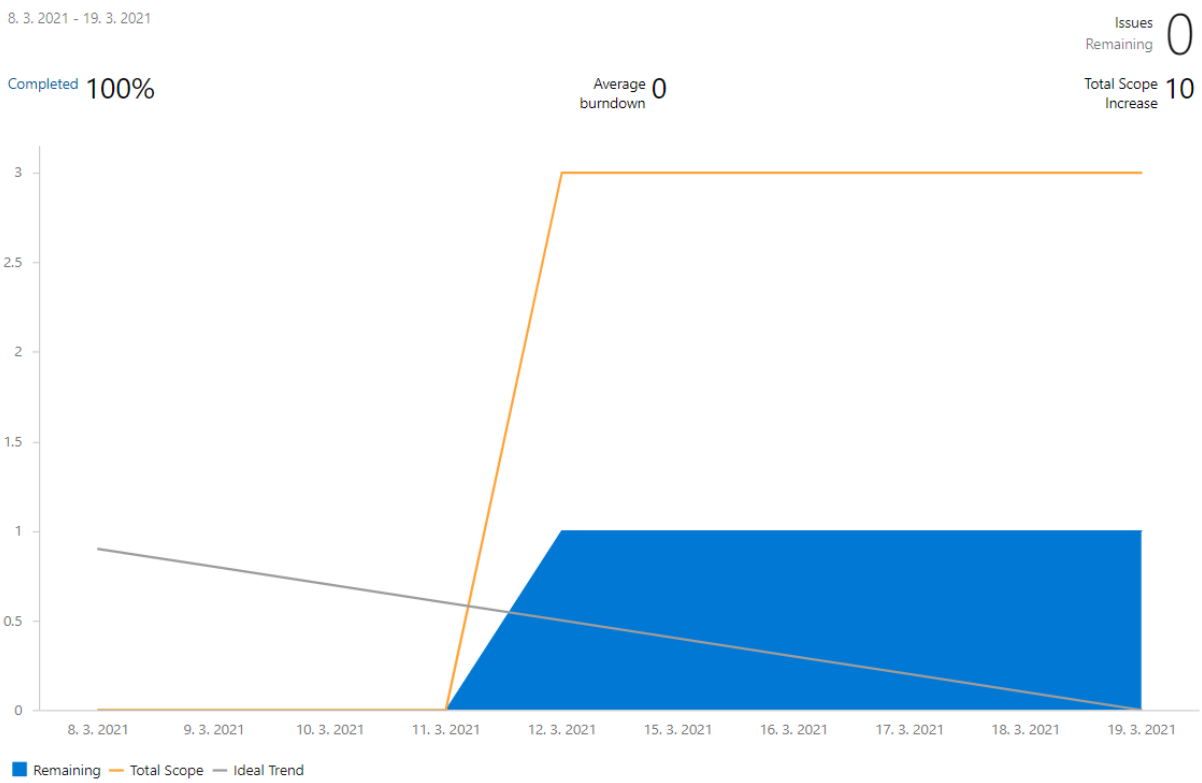
Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (nedeľa 21. 3.)	Šprint
Regenerate website certificate	Abd Alrahman Saleh	dokončené	šprint č. 7

Tabuľka 22: Úlohy zo siedmeho šprintu časť 2

Velocity je zobrazená na obrázku 1. Veľa úloh spočívalo v testovaní funkčnosti scenárov na rôznych operačných systémoch a riešení už len dodatočne identifikovaných problémov s Dockerom. Celková výkonnosť v siedmom šprinte je zobrazená na obrázku 2. Vykonané úlohy sú v tabuľke 21 a 22.



Obrázok 24: Velocity tímu v siedmom šprinte



Obrázok 25: Výkonnosť tímu v siedmom šprinte

Export úloh zo siedmeho šprintu

Cyran Team

Taskboard | Backlog | Analytics | New Work Item | Column Options | ...

Order	ID	Title	Assigned To	State	Tags
1	97	Create BCrypt services and tutorial template	Jakub Perdek	Done	
	98	Create BCrypt encryptor service	Jakub Perdek	Done	
	99	Create BCrypt validator service	Jakub Perdek	Done	
	100	Present information about salt and Bcrypt as intro to BCrypt services	Jakub Perdek	Done	
2	101	Document dictionary attack and add it to user guide	Jakub Perdek	Done	
3	102	Run and test application using docker on Windows	Nikola Karakas	Done	
4	103	Test docker images on Linux	abd alrahman ...	Done	
5	104	Find email which not check device or observe docker containers for smtp ser...	Jakub Perdek	Done	
6	105	Enhance user forms to gain their experiences from using application	Peter Spusta	Done	
7	106	Create build for easier deployment on linux machines		Done	
	108	Apply the same network rules for docker images	abd alrahman ...	Done	
8	107	Create sprint retrospective and sprint progress for sprint 7	Jakub Perdek	Done	
9	109	Test scenarios on new version of docker-compose version of our app on Win...	Jakub Perdek	Done	
10	111	Regenerate website certificate	abd alrahman ...	Done	

Obrázok 26: Export úloh zo siedmeho šprintu

Retrospektíva siedmeho šprintu

Už po druhý raz v novom semestri scrum tím 19 zahájil stretnutie pre vykonanie retrospektívy po dvojtýždňovom šprinte. Zaoberal sa výsledkami testovania lokálneho nasadenia Docker kontajnerov pre používateľský prieskum, zlepšenia podoby formulárov a celkový pokrok na zlepšovaní možností aplikácií s dôrazom na koncového používateľa.

Dátum a čas konania	Nedeľa 21. Marca, od (cca) 18 :0 0 - 19:34ho d.
Miesto konania	konferenčný hovor v General channel v Microsoft Teams
Retrospektíva za šprint:	8. Marca - 21. Marca
Účastníci	Jakub Perdek, Peter Spusta, Viktor Matovič, Nikola Karakaš, Abd Saleh
Spracovateľ	Jakub Perdek

Tabuľka 23: Informácie o retrospektíve siedmeho šprintu

Priebeh stretnutí

Účastníkom boli položené nasledujúce otázky s nasledujúcimi odpoveďami:

- ***Čo sa nám podarilo vykonať?***

  Viktor: Pomáhal pri builde docker kontajnerov na Ubuntu.

  Saleh: Testoval kontajnere na Ubuntu a inicializoval osobitný build na linuxe. Pripravil novú verziu Dokera pre všetky operačné systémy. Pracoval na refaktoringu a spolupracoval s ostatnými. Uploadoval aktualizovaný obsah na naše webové sídlo a aktualizoval HTTPS protokol na tomto sídle.

  Jakub: Pripravil scenár pre možnosť semi-automatickej detekcie. Následne preň aktualizoval obsah v používateľskej príručke. Rovnako asistoval na builde pre linux a odstraňovaní problémov vzniknutých po testovaní. Otestoval build s pridaním bridgu a rovnakej siete pre všetky kontajnere na Windowse.

  Nikola: Testoval aplikáciu na svojom zariadení používajúce Windows. Identifikoval niekoľko problémov.

Peter: Zlepšoval dotazníky pre používateľa, aby poskytli čo najviac informácií od používateľom pre každý scenár.

- ***Čo sa nám nepodarilo vykonať?***
  Viktor: Nestihol vylepšiť časti aplikácie.
  Saleh: Všetko, čo si naplánoval sa podarilo.
  Jakub: Viac vylepšiť UX na frontende a použiť SMTP server na odosielanie správ bez nutnosti prihlasovania do hosťovského emailového účtu.
  Peter: Urobil všetko čo mal, ale mali by sme zapracovať nejakú automatickú metódu ako napríklad nejaký skript na zapracovanie evaluácie od používateľov.
  Nikola: Nemal problémy. Urobil všetko, čo si naplánoval.

- ***Aké problémy sme identifikovali alebo máme?***
  Viktor: Nevie, že by mal problémy.
  Saleh: Nemá problémy.
  Jakub: Email pre lokálne nasadenie bude potrebné obísť.
  Peter: Nemá žiadne problémy ani nejaké neidentifikoval.
  Nikola: Ako spomenul, identifikoval problémy s aplikáciou pri používaní Docker kontajnerov ako napríklad nefunkčný email a nutnosť vypnúť lokálne bežiacu postgres databázu.

- ***Čo by sme v nasledujúcom šprinte zlepšili?***
  Viac sa stretávať, ale väčšina udalostí vyhovuje.
  Viac komunikovať o problémoch.
  Byť efektívni pri plnení úloh, tak aby sa ich stihlo čo najviac.

Záver

Vďaka spolupráci sa podarilo vytvoriť lokálne nasadené riešenie a otestovať ho na rôznych operačných systémoch, čo by malo pomôcť pri úspešnom testovaní aplikácie. Podarilo sa stihnúť skoro všetky podstatné úlohy.

3.8 Ôsmy šprint

Zlepšovanie UX a zavedenie logovania sú dôležitými pri nasadení a používateľskom testovaní aplikácie. Práve preto sme sa im tesne pred začiatkom tohto testovania v ôsmom šprinte venovali. Šprint začal 22. marca a skončil 4. apríla. Pretestovali sme aplikácie a pripravili nástroje pre vyhodnocovanie logov. Začali sme aj s prácami na RBAC modeli a session, ktoré sú pri webových aplikáciách dôležité.

Pokrok dosiahnutý na ôsmom šprinte

V tomto šprinte sme zapracovali riešenia ďalších identifikovaných problémov. Súčasťou šprintu bolo pridanie lokálneho emailového servera, zlepšenie použiteľnosti, refaktorizácia a oprava chýb, pridanie whois aplikácie do spoločného docker súboru pre plnohodnotný používateľský zážitok a dôležitou časťou bolo zavedenie logovača a jeho konfigurácia pre možnosť získať logy z používateľského testovania o aktivite používateľov.

Chýbajúci emailový server pri lokálnom nasadení sme sa rozhodli vyriešiť inak ako priamym zobrazením hesla. Vďaka existujúcemu testovaciemu kontajneru s emailovým serverom sme mohli všetky odoslané emaily z aplikácie zobraziť priamo v ňom. Kontajner sme pridali do docker-copose súboru a nakonfigurovali pripojenie na rovnakú sieť pomocou Bridgu a IP adresu pomocou premennej prostredia. Tú sme nakonfigurovali z docker súboru a odovzdali samotnému konfiguračnému súboru pre Spring, ktorý ju sprístupnil časti odosielajúcej emaily. Zároveň sme oddelili lokálne nasadenie od štandardného nasadenia na serveri pomocou ďalších premenných. Aplikáciu je možné použiť viacerými spôsobmi. Pri lokálnom nasadení boli vytvorené filtre konkrétne pre dve emailové adresy, ktoré zabraňujú aby boli odosielané správy od admina a asistenta. Tí boli vo východzom nastavení pridaní do aplikácie.

Zamýšľali sme sa aj na pridaní viacerých asistentov so slabým heslom, aby sme hráčom umožnili ľahšie zistiť aké účty sú prítomné v systéme.

V tomto šprinte sme realizovali úlohy zobrazené v tabuľkách 24 a 25.

Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (nedeľa 04. 04.)	Šprint
Add whois to dockerfile and fix issues with backup insertion server failures	Jakub Perdek	dokončené	šprint č. 8
Add fake email server to local docker deploy	Jakub Perdek	dokončené	šprint č. 8
Add logging to frontend connected with sentry	Jakub Perdek	dokončené	šprint č. 8
Create free developer account on sentry	Jakub Perdek	dokončené	šprint č. 8
Create script to obtain logs from sentry	Jakub Perdek	dokončené	šprint č. 8
Document sentry in engineer's work	Jakub Perdek	dokončené	šprint č. 8
Implement role based access control	Viktor Matovič	rozpracované	šprint č. 8
UX fixing and improvements	Jakub Perdek	dokončené	šprint č. 8
Disable admin and shop assistant email for local deployment	Jakub Perdek	dokončené	šprint č. 8
Review docker updated containers and applications	Abd Alrahman Saleh	dokončené	šprint č. 8
Create session for frontend and backend	Peter Spusta	rozpracované	šprint č. 8
Create progress and retrospective for sprint 8	Jakub Perdek	dokončené	šprint č. 8
Test functionality and UX for all scenarios using docker on local deployment	Nikola Karakaš	dokončené	šprint č. 8

Tabuľka 24: Úlohy z ôsmeho šprintu časť 1

Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (nedeľa 04. 04.)	Šprint
Create guide for local deployment using docker	Nikola Karakaš	dokončené	šprint č. 8

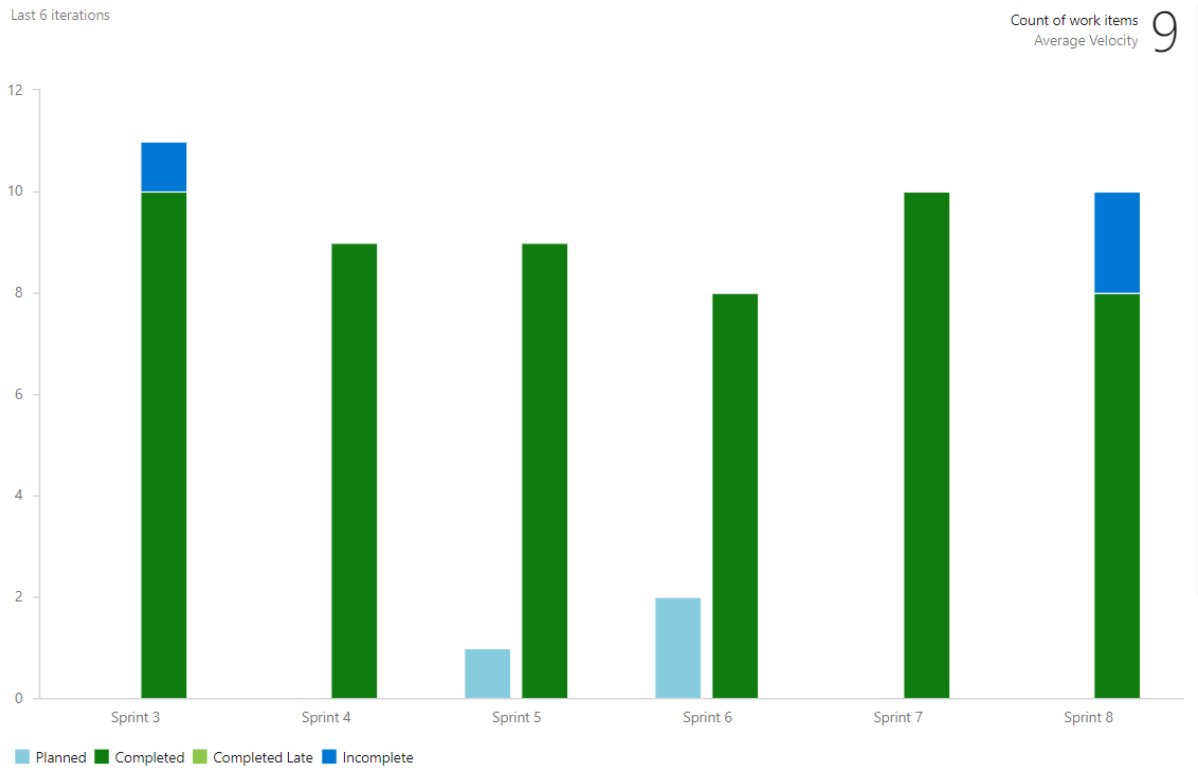
Tabuľka 25: Úlohy z ôsmeho šprintu časť 2

Podstatnou časťou šprintu bola aj fixácia problémov s použiteľnosťou aplikácie a ich zlepšenie, keďže v minulých šprintoch sme sa venovali zlepšovaniu scenárov, tvorbe dotazníkov a kontajnerizácii. Prevažne každá zmena v aplikácii už má svoju responzívnu spätnú väzbu. V menšej miere došlo aj k refaktorizácii aplikácie.

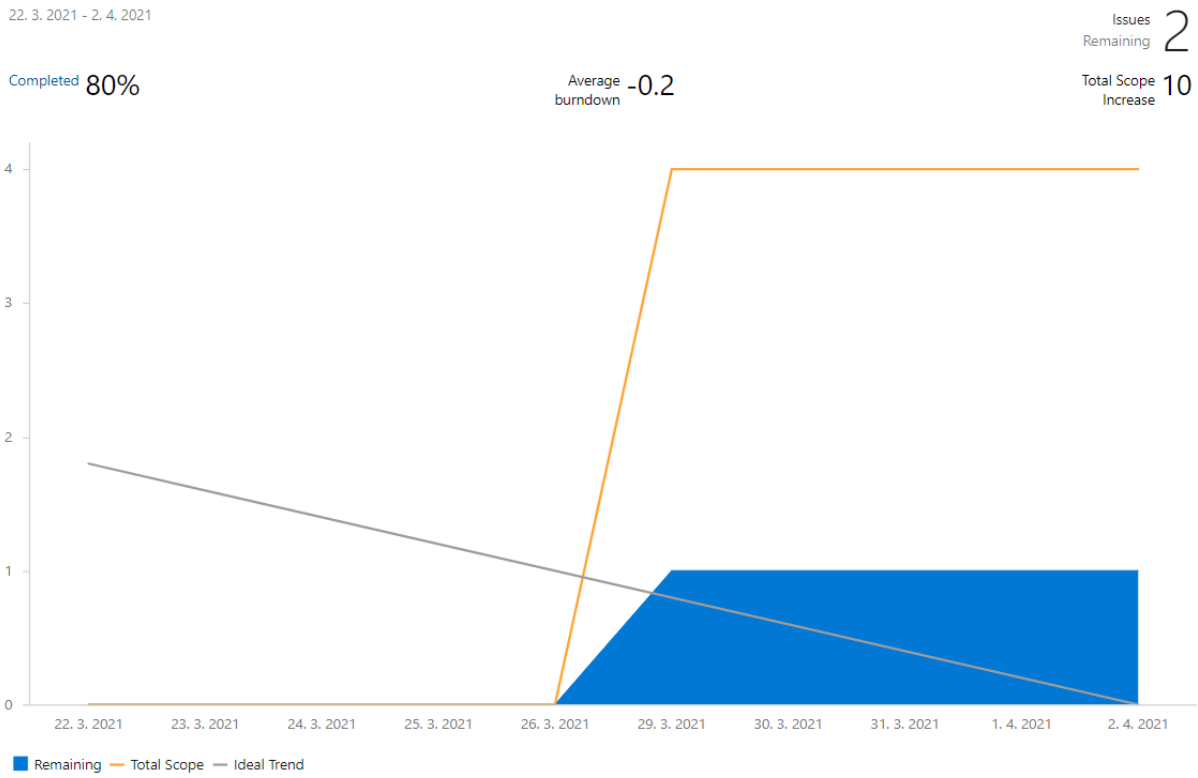
Dôležitým prírastkom v šprinte je tvorba logov, ktoré budú použité na rôzne účely. Prvým je monitorovanie pokroku používateľov a reálne overenie dosiahnutých výsledkov. Ďalším je testovanie schopnosti útočníka vyhnúť sa zalogovaniu jeho aktivity pri dosiahnutí nejakého cieľa ako je napríklad neodoslanie logu pri získaní produktov bez platby. Okrem toho sledovanie chýb vyskytujúcich sa v aplikácii tiež môže byť prínosné. Logy vieme získať zo servera pomocou skriptu v csv formáte. Možno nad nimi realizovať ďalšie analýzy. Použitý je developérsky účet na Sentry, ktorý by mal stačiť na testovanie niekoľkými používateľmi.

V šprinte sa nepodarilo dokončiť RBAC model a session pre odstránenie potencionálnych bezpečnostných slabín v aplikácii. Práce na tejto funkcionalite boli zahájené, ale vzhľadom na komplexnosť úloh ich prípadne presunieme do nasledujúceho šprintu.

Nikola otestoval aplikáciu po zmenách v tomto šprinte. Komunikoval dôležité problémy a chyby zavedené zmenami v kóde. Tie sa podarilo v krátkom čase odstrániť. Z hľadiska použiteľnosti a funkčnosti by aplikácia mala byť pripravená na používateľské testovanie. Zaznamenali sme štandardný výkon v šprinte, ktorý sa ale nepodarilo vzhľadom na náročnosť úloh na backende vylepšiť o tieto úlohy. Velocity v šprinte 8 môžete vidieť na obrázku 27. Výkonnosť tímu v šprinte 8 zobrazuje obrázok 28.

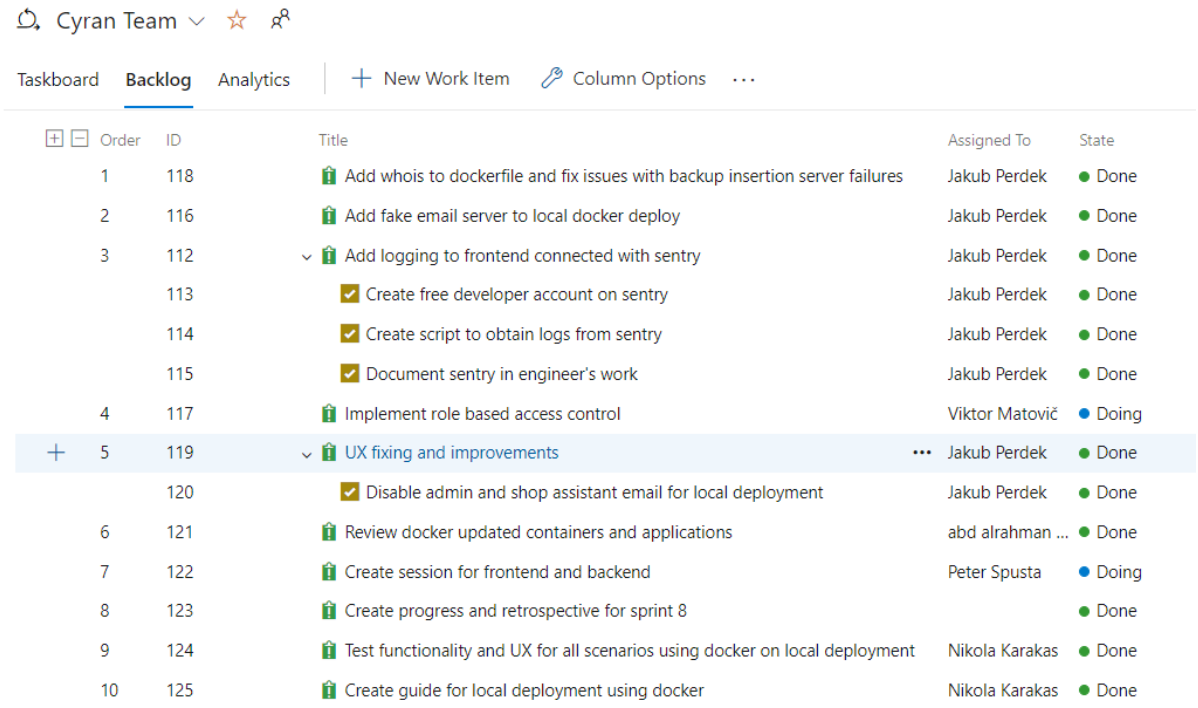


Obrázok 27: Velocity tímu v ôsmom šprinte



Obrázok 28: Výkonnosť tímu v ôsmom šprinte

Export úloh z ôsmeho šprintu



Order	ID	Title	Assigned To	State
1	118	Add whois to dockerfile and fix issues with backup insertion server failures	Jakub Perdek	Done
2	116	Add fake email server to local docker deploy	Jakub Perdek	Done
3	112	Add logging to frontend connected with sentry	Jakub Perdek	Done
	113	Create free developer account on sentry	Jakub Perdek	Done
	114	Create script to obtain logs from sentry	Jakub Perdek	Done
	115	Document sentry in engineer's work	Jakub Perdek	Done
4	117	Implement role based access control	Viktor Matovič	Doing
5	119	UX fixing and improvements	Jakub Perdek	Done
	120	Disable admin and shop assistant email for local deployment	Jakub Perdek	Done
6	121	Review docker updated containers and applications	abd alrahman ...	Done
7	122	Create session for frontend and backend	Peter Spusta	Doing
8	123	Create progress and retrospective for sprint 8		Done
9	124	Test functionality and UX for all scenarios using docker on local deployment	Nikola Karakas	Done
10	125	Create guide for local deployment using docker	Nikola Karakas	Done

Obrázok 29: Export úloh z ôsmeho šprintu

Retrospektíva ôsmeho šprintu

Dvojtýždňový šprint začal 22. marca a končí 4. Apríla 2021. Pre tím 19 je to tretí šprint letného semestra. Šprint je opäť plný vylepšení a prípravy nástrojov pre získanie čo najväčšieho množstva informácií z používateľského prieskumu.

Dátum a čas konania	Pondelok 5. Apríla, od (cca) 20 :1 0 – 20:42 hod.
Miesto konania	konferenčný hovor v General channel v Microsoft Teams
Retrospektíva za šprint:	22. Marca - 4. Apríl
Účastníci	Jakub Perdek, Peter Spusta, Viktor Matovič, Nikola Karakaš, Abd Saleh
Spracovateľ	Jakub Perdek

Tabuľka 26: Informácie o retrospektíve ôsmeho šprintu

Priebeh stretnutí

Účastníkom boli položené nasledujúce otázky s nasledujúcimi odpoveďami:

- ***Čo sa nám podarilo vykonať?***

  Viktor: Stále pracuje na RBAC modely. Stále zostáva ešte veľa funkcionality na dokončenia.

  Saleh: Robil revízie pre pridané časti do docker-compose súboru. Bola nimi whois aplikácia a webový server.

  Jakub: Pridal do docker kontajnera emailový server. Pre plný používateľský zážitok bola pridná whois aplikácia do docker-compose. Zároveň založil účet na Sentry a umožnil logovanie chýb a informácii z aplikácie hlavne o pokrokoch používateľa. Vypracoval spôsob získania logov zo Sentry pre ich ďalšie vyhodnotenie. Opravil chyby v UX a zaviedol responzívnu spätnú väzbu pre používateľa naprieč celou aplikáciou.

  Nikola: Robil testovanie všetkých komponentov v aplikácii pre docker lokálne nasadenie. Našiel chyby a komunikoval ich. Pripravil dokument pre používateľa pre lepšie zorientovanie sa v aplikácii.

  Peter: Stále pracuje na session.

- ***Čo sa nám nepodarilo vykonať?***

  Viktor: Pripravil kód na RBAC, ale nedokončil ho.

  Saleh: Stále pracuje na refaktoringu a fixnutí SSL certifikátu pre webové sídlo projektu.

  Jakub: Viac zlepšení bezpečnostných problémov aplikácie a dôkladnejšiu refaktorizáciu.

  Peter: Nedokončil session.

  Nikola: Všetko čo si naplánoval aj stihol.

- ***Aké problémy sme identifikovali alebo máme?***

  Viktor: Nemá žiadne problémy, ale chcel by pracovať viac na úlohách.

  Saleh: Má problémy s SSL certifikátom.

  Jakub: Aplikácia má nedostatky v riešení ďalších bezpečnostných problémov ako je overovanie rolí iba na frontende.

Peter: Mal problémy s počítačom, čo mu spôsobilo meškanie. Rovnako potrebuje viac času.

Nikola: Nemá problémy.

- ***Čo by sme v nasledujúcom šprinte zlepšili?***

  Mali by sme dokončiť všetky rozpracované úlohy pre nasmerovanie používateľa k riešeniu predpripravených úloh zo scenárov, tak aby nemohol využiť inú chybu v systéme.
  Uplatniť ďalšie metodiky z MUNI.

Záver

Zameranie sa na zlepšenie zážitku používateľa v rôznych formách ako bolo zlepšenie UX, pridanie logovania a príprava rôznych foriem lokálneho nasadenia aplikácií aj pre menej výkonné stanice by mala byť dobrým krokom urobeným v tomto šprinte (Používateľ si môže vybrať či chce aj Whois aplikáciu). Zakomponovanie ďalších metodík je podstatné pre projekt a mal by naň byť v budúcnosti kladený dôraz.

3.9 Deviaty šprint

Šprint výnimočný používateľským testovaním bol práve ten deviaty. Používatelia si mohli skúsiť jednotlivé scenáre a ohodnotiť použiteľnosť aplikácií. Očarení boli vzhľadom, originalitou úloh a spracovaním, ale kritizovali rôzne technické problémy a ich chýbajúci popis v príručke. Zároveň by chceli viac scenárov s väčšou zložitosťou. Niektoré z týchto požiadaviek sme sa rozhodli uspokojiť už v tomto šprinte.

Pokrok dosiahnutý na deviatom šprinte

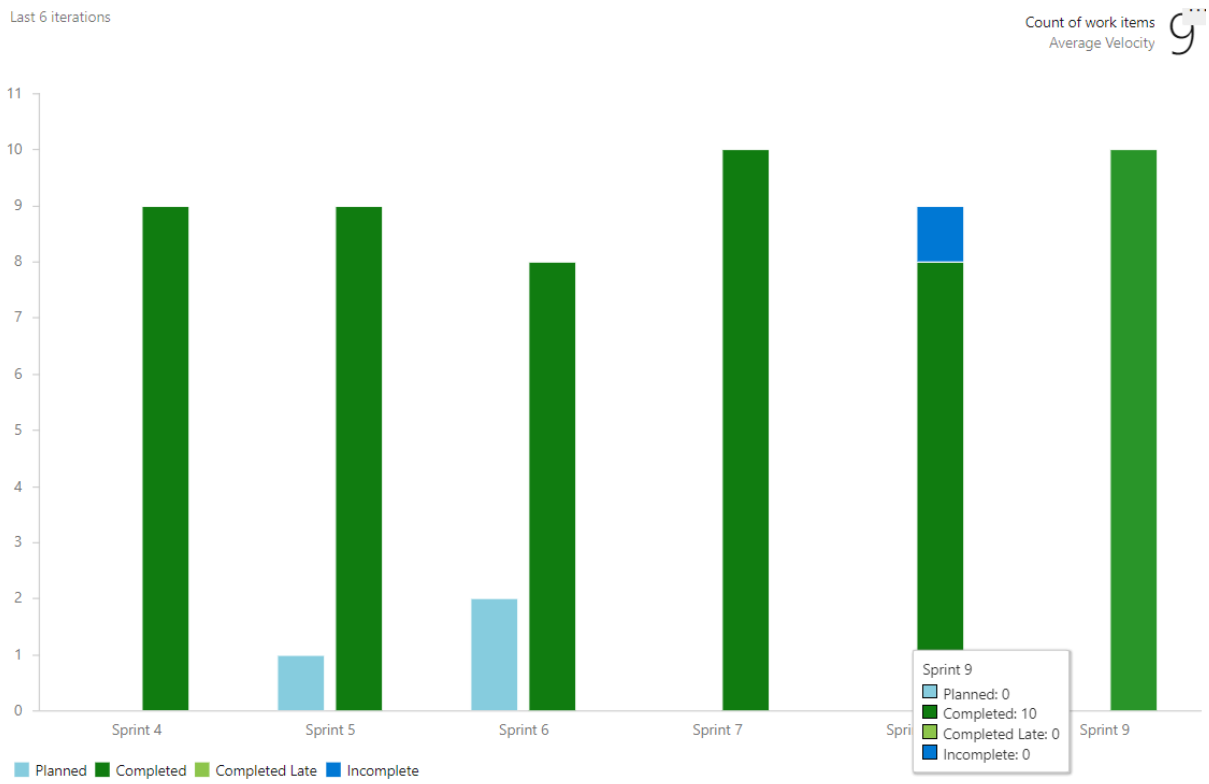
Na konci deviateho šprintu sa začalo používateľské testovanie. V priebehu tohto testovania sme identifikovali množstvo chýb pomocou vlastného logera s názvom Sentry a na základe zozbieranej spätnej väzby od používateľov. Používatelia siahali po používateľskej príručke a identifikovali nedostatky, ktoré sa ani po našom tímovom testovaní nepodarilo identifikovať. Očakávania boli rôzne. Aj napriek použitiu príručky používatelia očakávali viac logických problémov väčšiu zložitosť a viac scenárov. Ďalej by uvítali zahrnúť do

používateľskej príručky aj konfiguráciu k BurpSuite nástroju. Aplikácia sa im aj napriek problémom páčila a posmeľujú k jej vylepšeniu.

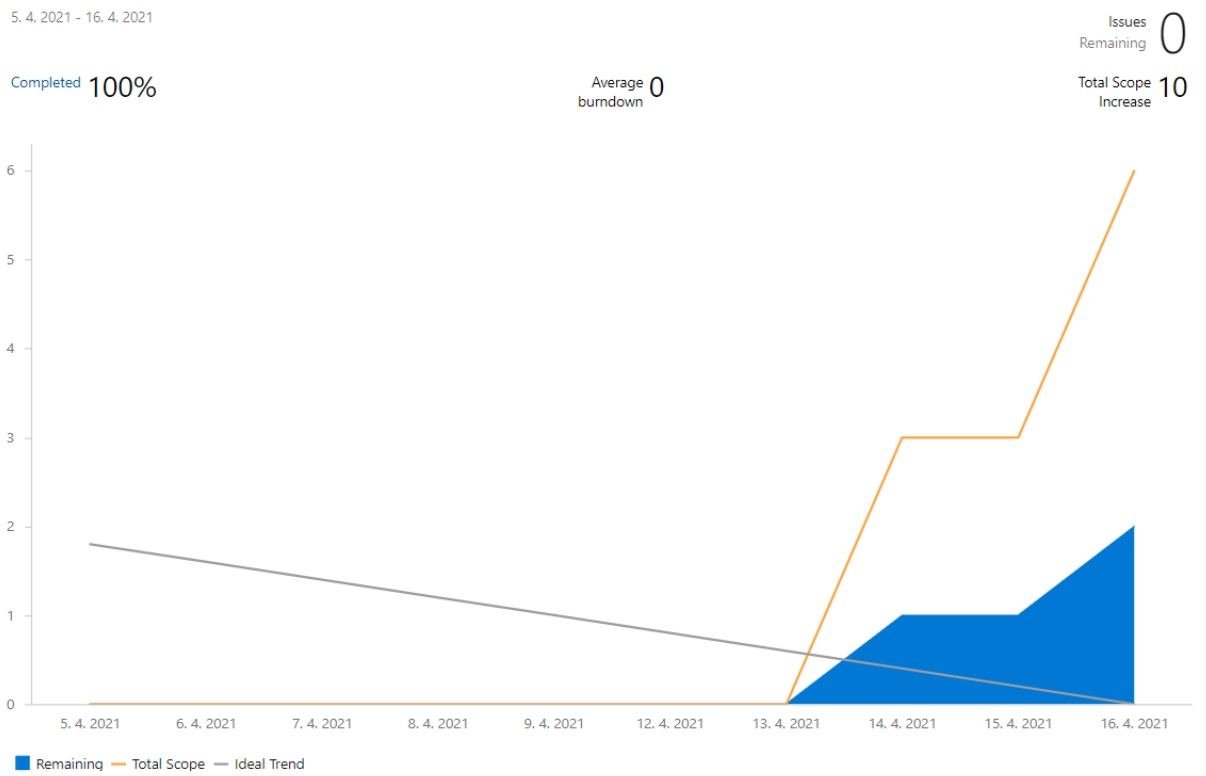
Zabezpečiť plynulý chod pri testovaní bolo nevyhnutné. Logy sme pravidelne sledovali v Sentry a rovnako aj mailovú komunikáciu na ktorú sme pohotovo reagovali. Okrem samotných docker obrazov sme nakoniec sprístupnili aj pôvodný repozitár po jeho vyžiadaní. Väčšina emailov zahŕňala spätnú väzbu. Tú sme spracovali do excel dokumentu rozdelením na časť s identifikovanými chybami, ďalej na časť s pocitmi používateľov, návrhmi na vylepšenie a chválenú funkcionalitu. Podľa logov a spätnej väzby používatelia väčšinu scenárov úspešne dokončili.

V šprinte sme pokračovali na vylepšeniach aplikácií. Konkrétne sme analyzovali možné vloženie referencie security eshopu do whois záznamov pre našu Whois aplikáciu. Lokálne nasadenie bolo znovu problematické. Docker compose na základe zistení neumožnil vygenerovať jedinečný identifikátor pre novú doménu, ale sa dal vložiť len ako premenná, buď pri volaní docker-compose alebo z .env súboru, prípadne iným spôsobom. Druhý problém bol závažnejší. Lokálne nasadenie pri zmene domény vyžaduje nastavenie domény v host súbore konkrétneho používateľa. Používateľ by tím vedel akú doménu bude vyhľadávať a bol by ešte viac zaťažený inštaláciou. Ďalším riešením je reverse proxy, ale aj to vyžaduje pluginy do prehliadača. Doména preto zostáva localhost a vo whois aplikácii bude potrebné zťažiť prístup k nej.

Vloženie záznamu o security eshope vyžadovalo aj zverejnenie nejakých zraniteľností. Boli preto vytvorené ďalšie tri tabuľky umožňujúce mapovať zraniteľnosti s ľahkým pridaním ďalšej a zmeny stupnice pre ich mieru nebezpečnosti. Po tvorbe záznamu Jakub pre ne pridal 2 záznamy o zraniteľnostiach v podobe nickov dvoch používateľov a upozornenie na únik citlivých údajov. Zároveň používateľ by mal byť motivovaný agregovať si doménu s najväčším počtom zraniteľností pomocou zložitej SQL injekcie, ktorú bude možné realizovať v hlavnom vyhľadávacom okne. Funkcionalita ale umožňuje vrátenie len jedného záznamu, čo bude používateľ musieť pri SQL injekcii dodržať. Prístupná pre tento scenár bude aj schéma databázy, ktorou sa aplikácia na jednej obrazovke aj pochváli. Navrhli sme a vytvorili tak ďalší scenár a reagovali tak na požiadavku používateľov žiadajúcich scenáre nútiace rozmýšľať a vynájsť sa pri ich riešení. Vymyslenú doménu budeme ešte musieť v texte whois záznamu namapovať na aktuálny security eshop informáciou o zmene domény na localhost.



Obrázok 30: Velocity tímu v šprinte 9



Obrázok 31: Výkonnosť tímu v deviatom šprinte

V tomto šprinte sme realizovali úlohy zobrazené v tabuľke 1.

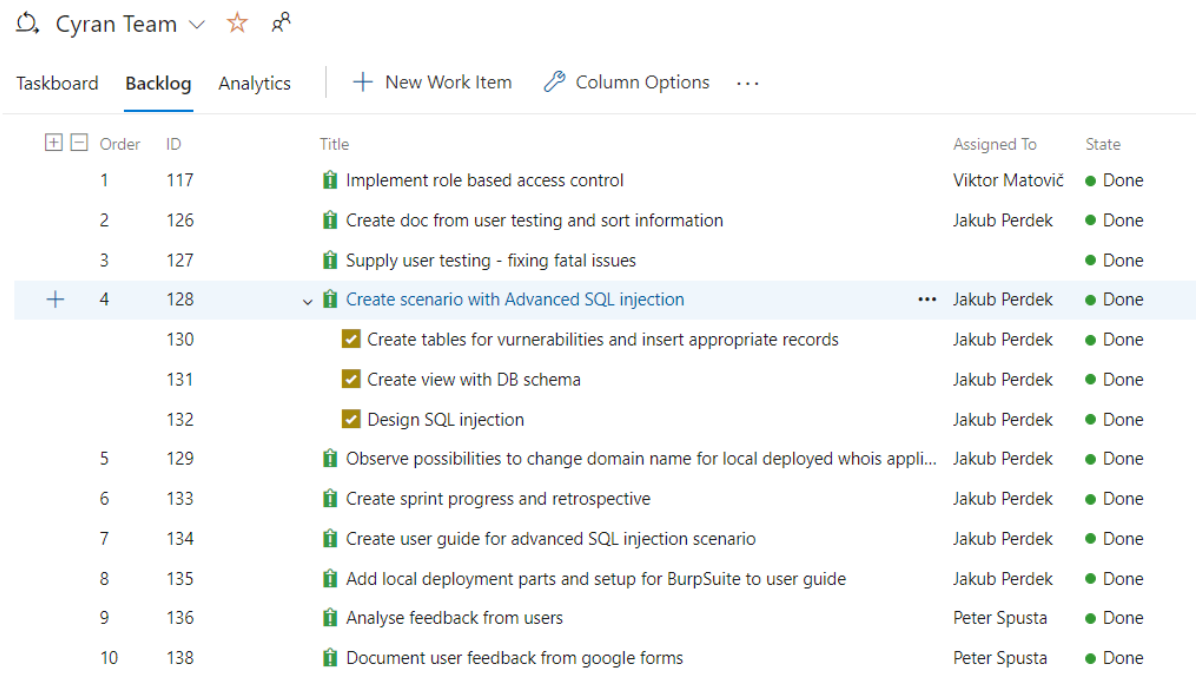
Pridelená úloha	**Zodpovedný riešiteľ**	**Aktuálny stav rozpracovania (nedeľa 18. 04.)**	**Šprint**
Implement role based access control	Viktor Matovič	dokončené	šprint č. 9
Create doc from user testing and sort information	Jakub Perdek	dokončené	šprint č. 9
Supply user testing - fixing fatal issues	Jakub Perdek Abd Alrahman Saleh	dokončené	šprint č. 9
Create scenario with Advanced SQL injection	Jakub Perdek	dokončené	šprint č. 9
Create tables for vurnerabilities and insert appropriate records	Jakub Perdek	dokončené	šprint č. 9
Create view with DB schema	Jakub Perdek	dokončené	šprint č. 9
Design SQL injection	Jakub Perdek	dokončené	šprint č. 9
Observe possibilities to change domain name for local deployed whois application	Jakub Perdek	dokončené	šprint č. 9
Create sprint progress and retrospective	Jakub Perdek	dokončené	šprint č. 9
Create user guide for advanced SQL injection scenario	Jakub Perdek	dokončené	šprint č. 9
Add local deployment parts and setup for BurpSuite to user guide	Jakub Perdek	dokončené	šprint č. 9
Analyse feedback from users	Peter Spusta	dokončené	šprint č. 9
Document user feedback from google forms	Peter Spusta	dokončené	šprint č. 9

Tabuľka 27: Úlohy z deviateho šprintu

Peter analyzoval spätnú väzbu z google formulárov a aj ju spracoval do inžinierskeho diela. Zároveň sa venoval aj rozpracovanej session, ktorú sme ale do šprintu nezahrnuli. Pravdepodobne ju dokončí v nasledujúcom šprinte. Podarilo sa dokončiť aj funkcionalitu na backende implementujúcu riadenie prístupu na základe rolí, ktoré používateľ má. Jej tvorcom bol Viktor Matovič. Nikola sa snažil otestovať aplikáciu pomocou OWASP nástroja, ale pre nemožnosť zmeniť port z 8080 sa mu to nepodarilo.

V šprinte sme boli výkonní aj napriek tomu, že sme čakali na prvú spätnú väzbu od používateľov. Tú sme nielen zdokumentovali ale následne z nej aj vyriešili veľké množstvo problémov. Naša velocity bola preto jedna z najlepších doposiaľ dosiahnutých. Zobrazuje ju obrázok 1. Výkonnosť v šprinte zobrazuje obrázok 2.

Export úloh z deviateho šprintu



Order	ID	Title	Assigned To	State
1	117	Implement role based access control	Viktor Matovič	Done
2	126	Create doc from user testing and sort information	Jakub Perdek	Done
3	127	Supply user testing - fixing fatal issues		Done
4	128	Create scenario with Advanced SQL injection	Jakub Perdek	Done
	130	Create tables for vurnerabilities and insert appropriate records	Jakub Perdek	Done
	131	Create view with DB schema	Jakub Perdek	Done
	132	Design SQL injection	Jakub Perdek	Done
5	129	Observe possibilities to change domain name for local deployed whois appli...	Jakub Perdek	Done
6	133	Create sprint progress and retrospective	Jakub Perdek	Done
7	134	Create user guide for advanced SQL injection scenario	Jakub Perdek	Done
8	135	Add local deployment parts and setup for BurpSuite to user guide	Jakub Perdek	Done
9	136	Analyse feedback from users	Peter Spusta	Done
10	138	Document user feedback from google forms	Peter Spusta	Done

Obrázok 32: Export úloh z deviateho šprintu

Retrospektíva z deviateho šprintu

Dvojtýždňový šprint začal 5. apríla a končí 18. apríla 2021. Pre tím 19 je to štvrtý šprint letného semestra v ktorom prebehlo používateľské testovanie.

Dátum a čas konania	Nedeľa 18. Apríla, od (cca) 20 :1 0 – 21:05 hod.
Miesto konania	konferenčný hovor v General channel v Microsoft Teams
Retrospektíva za šprint:	5. Marca - 18. Apríl
Účastníci	Jakub Perdek, Peter Spusta, Viktor Matovič, Nikola Karakaš, Abd Saleh
Spracovateľ	Jakub Perdek

Tabuľka 28: Informácie o retrospektíve deviateho šprintu

Priebeh stretnutí

Účastníkom boli položené nasledujúce otázky s nasledujúcimi odpoveďami:

- ***Čo sa nám podarilo vykonať?***

  Viktor: Dokončil RBAC model a prešiel si spätnú väzbu z dokumentov.

  Saleh: Upravil prístupové práva na repozitár a jeho obsah pre rýchly a používateľsky prívetivý setup docker kontajnerov. Asistoval pri riešení problémov s používateľmi.

  Jakub: Vytvoril pomocný repozitár s inštrukciami pre rozbehnutie docker obrazov a asistoval pri riešení problémov s používateľmi. Vytvoril scenár pre zložitú SQL injekciu, ktorý zahŕňal tvorbu DB schémy, jej zverejnenie na samostatnej stránke a doplnenie funkcionality pre zobrazovanie zraniteľností . Zároveň doplnil používateľskú príručku o tento scenár. Zozbieral spätnú väzbu od používateľov a doplnil konfiguráciu/riešenie problému s Burpsuite do príručky.

  Nikola: Snažil sa použiť OWASP nástroj.

  Peter: Evaluoval spätnú väzbu od používateľov a vytvorené výsledky zaznamenal do dokumentácie.

- ***Čo sa nám nepodarilo vykonať?***

  Viktor: Nestihol k novej službe pre RBAC model urobiť testy.

  Saleh: Nestihol refaktorovať aplikáciu.

  Jakub: Nestihol refaktorovať frontend a backend, hlavne urobiť aplikáciu viac konfigurovateľnou napríklad pre ľahšiu zmenu portu.

Peter: Rozrobil prácu na session, stále ešte nejakú funkcionalitu je potrebné dokončiť.

Nikola: Neotestoval aplikáciu pomocou OWASP nástroja.

- ***Aké problémy sme identifikovali alebo máme?***

  Viktor: Mal technické problémy s JPA.

  Saleh: Prístup k repozitáru nebol používateľom udelený, ale dostali mail s inštrukciami. Musel repozitár preto sprístupniť.

  Jakub: Používateľská príručka neobsahovala popis konfigurácie BurpSuite a emailová komunikácia pre lokálne nasadenie rovnako nebola popísaná.

  Peter: Jedna z úloh používateľom nefungovala. Bola to tá využívajúca remotnú Google databázu.

  Nikola: Identifikoval konfiguračné problémy. Nefungoval mu OWASP nástroj. Viktor poradil, že by si mohol stiahnuť Sonar Community edition a cez XPATH môže vytvoriť pravidlo podľa seba na identifikáciu SQL injection.

- ***Čo by sme v nasledujúcom šprinte zlepšili?***

  Test aplikácie na inom než našich vlastných počítačoch, aby sme dokázali napríklad identifikovať problém s prístupom z iného zariadenia.
  Vytvoriť ďalšie scenáre pre aplikáciu.

Záver

Lepšie testovať aplikácie na druhých počítačoch, respektíve so zapojením druhých používateľov. Rýchlo zapracovať čo najviac vylepšení a zlepšovaní ako v tomto šprinte.

3.10 Desiaty šprint

V predposlednom šprinte sme sa zamerali na zlepšovanie kvality kódu, analýzu logov zo Sentry a návrh ďalších scenárov. Frontend vôbec neobsahoval komentáre, a preto okomentovať prácu približne z desiatich šprintov predstavovalo veľa práce. Analyzovali sme aj použitie OWASP ZAP nástroja a jeho následnú demonštráciu pri riešení scenárov v našich aplikáciách. Kód aplikácií by po tomto šprinte mal byť zdokumentovaný a refaktorovaný.

Pokrok dosiahnutý na desiatom šprinte

V desiatom šprinte sa tím 19 sústredil na zlepšovanie kvality kódu, jeho zdokumentovanie, ďalej na analýzu a vizualizáciu logov po používateľskom testovaní a v neposlednom rade aj na tvorbu jednotkových testov. Pokračovali sme aj na vylepšeniach očakávaných od jednotlivých používateľov po používateľskom prieskume.

Peter dokumentoval pridanú funkcionalitu na backende. Jakub zdokumentoval všetky komponenty a služby na frontende. Frontend sa nestihol zdokumentovať v predchádzajúcom semestri, preto bola práca na dokumentácii zdĺhavejšia. V prípade backendu sme dokumentovali štandardne v JavaDoc. Pre frontend sme použili plugin do Visual Studio Code Comments in Typescript[1], ktorý automaticky vygeneroval šablónu do ktorej sme doplnili jednotlivý popis metódy s jej parametrami a návratovou hodnotou. Komentovali sa služby a komponenty. Rúry alebo iná funkcionalita nebola vytvorená, preto ju nebolo potrebné popisovať.

Nikola sa zatiaľ pokúšal navrhnúť scenár s použitím OWASP ZAP nástroja. Jeho pôvodným zámerom bolo, aby tento nástroj používateľ použil manuálne cieleným spôsobom. Tento nástroj ale využíval HUD, ktorý predpokladal zabezpečené HTTPS spojenie. Bez neho dochádzalo k problémom, ktoré znemožňovali prijímanie odoziev z backendu frontendom. Jeho zablokovanie umožnilo nástroj OWASP ZAP používať, ale znemožnilo jeho kľúčovú funkcionalitu. Následne boli analyzované automatické vymoženosti nástroja. Nástroj napríklad automaticky dokáže odhaliť zabudnuté komentáre vývojárov, čo môže byť využiteľné ako dodatočná pomôcka pri jednom z budúcich scenárov. Náš kód na frontende sa ale automaticky minifikuje, pričom dochádza k odstráneniu komentárov.

Saleh spustil príkaz na update docker obrazov vo svojom docker hub repozitári a pravidelne overuje funkčnosť aplikácie, ktorá je v hlavnej vetve. Podieľal sa rovnako na review kódu pri spájaní komitov do hlavnej vetvy. Za cieľ si dal rovnako refaktoring.

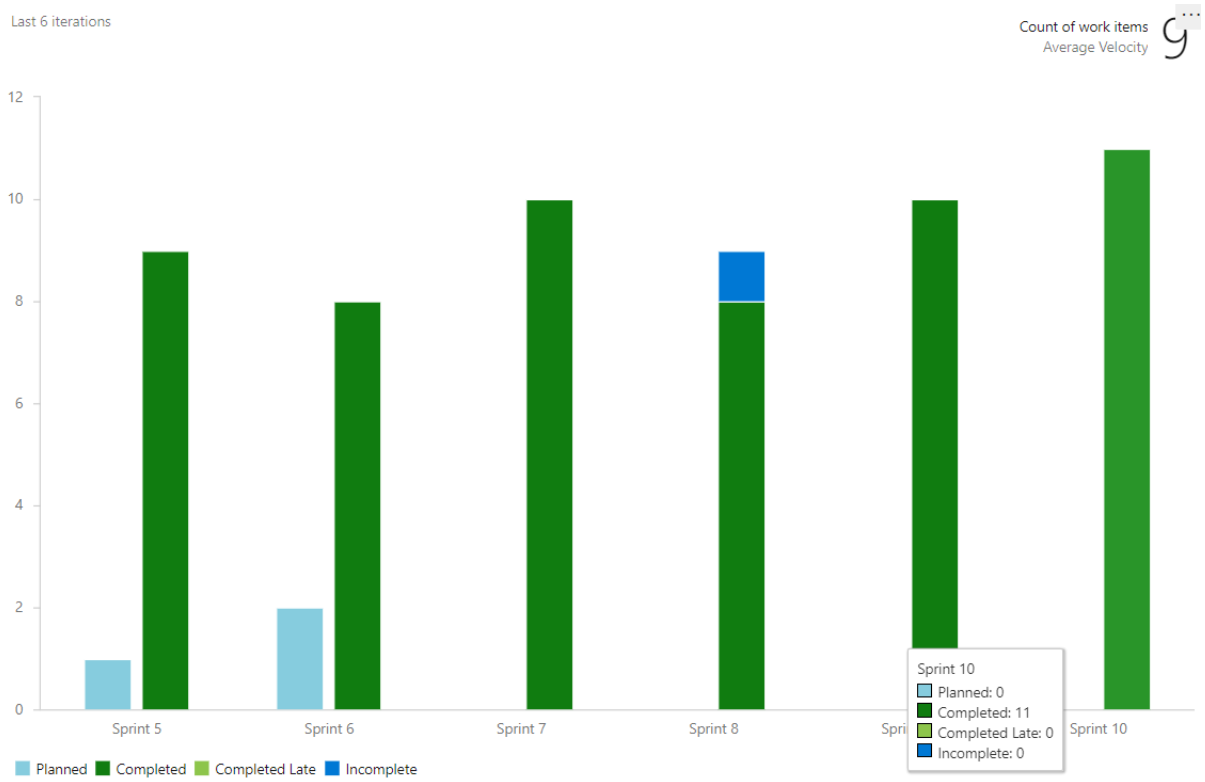
Okrem prípravy nových scenárov a zlepšovaní aplikácie bolo potrebné vyhodnotiť a vizualizovať aj logy nazbierané pomocou Sentry logovača. Jakub napísal skripty v R vytvárajúce grafy z jednotlivých údajov. Informácie obsahovali typ správ, typ lokácií z URL stránky frontendu, počet udalostí, počet unikátnych používateľov a prípadné komentáre reviewerov konkrétnych správ, ktoré ale neboli prítomné. Číselné hodnoty mali len počet udalostí a počet unikátnych používateľov medzi ktorými sme identifikovali lineárnu závislosť.

[1] https://marketplace.visualstudio.com/items?itemName=salbert.comment-ts

Z logov bola identifikovaná len jedna častá chyba v prehliadači. Zvyšné logy tvorili správy o prelomení nejakej obrany v eshope. Najčastejšie boli posielané z prihlasovacieho komponentu aplikácie.

Viktor vytvoril jednotkové testy pre pridanú funkcionalitu v minulom šprinte. Tie testovali role používateľov v requestoch dopytujúcich sa po konkrétnej službe.

Efektívnosťou sme prekonali efektivitu v predchádzajúcich šprintoch. Dokončili sme celkovo 11 úloh. Väčšina úloh sa zameriavala na zlepšenie kvality kódu alebo na pokrytie kódu jednotkovými testami. Zvyšné úlohy boli analytické. Napríklad analýza OWASP ZAP nástroja alebo analýza a vizualizácia logov. Zistili sme, že je možné pridať pomôcky pre útočníkov vo forme zabudnutých komentárov za predpokladu, že kód nebude minifikovaný a tento nástroj to úspešne detekuje. Zároveň zaujímavé zistenie zo samotného používateľského testovania bolo, že používatelia po prelomení hesla asistenta v eshope istú chvíľu nevedeli ako ďalej, lebo sa viac krát prihlasovali pod tento účet. Zároveň táto ako aj ďalšia ich aktivita bola zalogovaná. Velocity bola preto 11. Zobrazuje ju obrázok 1. Výkonnosť tímu v šprinte číslo 10 môžete vidieť na obrázku číslo 2.

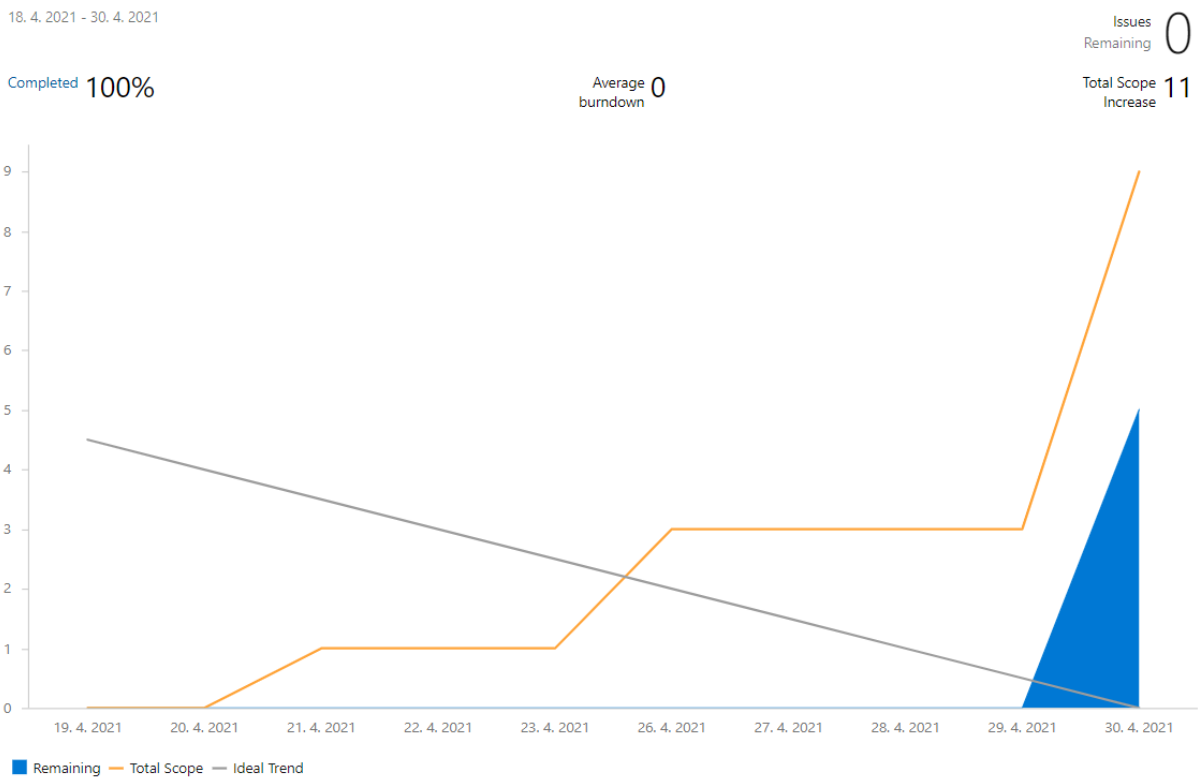


Obrázok 33: Velocity tímu v šprinte 10

V tomto šprinte sme realizovali úlohy zobrazené v tabuľke 1.

Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (nedeľa 02. 05.)	Šprint
Comment new code on backend	Peter Spusta	dokončené	šprint č. 10
Refactoring code on frontend with test of functionality	Abd Alrahman Saleh	dokončené	šprint č. 10
Make application more portable	Jakub Perdek	dokončené	šprint č. 10
Move logic to separated services on frontend	Jakub Perdek	dokončené	šprint č. 10
Create configuration for easy url/port change for app on frontend	Jakub Perdek	dokončené	šprint č. 10
Test changes on scenarios locally - without docker	Jakub Perdek	dokončené	šprint č. 10
Analyze and visualize logs from sentry	Jakub Perdek	dokončené	šprint č. 10
Document analysis and visualization of logs from Sentry	Jakub Perdek	dokončené	šprint č. 10
Review changes and update images	Abd Alrahman Saleh	dokončené	šprint č. 10
Analyze OWASP ZAP automatic scan/attack as scenario	Nikola Karakaš	dokončené	šprint č. 10
Comment components and services on frontend	Jakub Perdek	dokončené	šprint č. 10
Create sprint progress and retrospective	Jakub Perdek	dokončené	šprint č. 10
Analyze OWASP ZAP manual attack using HUD	Nikola Karakaš	dokončené	šprint č. 10
Unit testing on backend	Viktor Matovič	dokončené	šprint č. 10

Tabuľka 29: Úlohy z desiateho šprintu



Obrázok 34: Výkonnosť tímu v desiatom šprinte

Export úloh z desiateho šprintu

Cyran Team

Taskboard | Backlog | Analytics | New Work Item | Column Options | ...

Order	ID	Title	Assigned To	State
1	151	Unit testing on backend	Viktor Matovič	Done
2	149	Comment new code on backend	Peter Spusta	Done
3	78	Refactoring code on frontend with test of functionality	abd alrahman ...	Done
4	139	Make application more portable	Jakub Perdek	Done
	140	Move logic to separated services on frontend	Jakub Perdek	Done
	141	Create configuration for easy url/port change for app on frontend	Jakub Perdek	Done
	142	Test changes on scenarios locally - without docker	Jakub Perdek	Done
5	143	Analyze and visualize logs from sentry	Jakub Perdek	Done
6	144	Document analysis and visualization of logs from Sentry	Jakub Perdek	Done
7	145	Review changes and update images	abd alrahman ...	Done
8	146	Analyze OWASP ZAP manual attack using HUD	Nikola Karakas	Done
9	147	Comment components and services on frontend	Jakub Perdek	Done
10	148	Create sprint progress and retrospective	Jakub Perdek	Done
11	150	Analyze OWASP ZAP automatic scan/attack as scenario	Nikola Karakas	Done

Obrázok 35: Export úloh z desiateho šprintu

Retrospektíva z desiateho šprintu

Dvojtýždňový šprint začal 19. apríla a končí 2. mája 2021. Predposledný šprint v letnom semestri bol zameraný na skvalitňovanie vlastností aplikácie a zapracovanie zlepšení od používateľov.

Dátum a čas konania	Nedeľa 02. mája, od (cca) 20:1 9 – 21:15 hod.
Miesto konania	konferenčný hovor v General channel v Microsoft Teams
Retrospektíva za šprint:	19. apríla - 02. mája
Účastníci	Jakub Perdek, Peter Spusta, Viktor Matovič, Nikola Karakaš, Abd Saleh
Spracovateľ	Jakub Perdek

Tabuľka 30: Informácie o retrospektíve desiateho šprintu

Priebeh stretnutí

Účastníkom boli položené nasledujúce otázky s nasledujúcimi odpoveďami:

- ***Čo sa nám podarilo vykonať?***

  Viktor: Písal jednotkové testy pre pridanú službu na backende.

  Saleh: Robil review pred mergom do master vetvy a testuje aplikáciu s docker obrazmi.

  Jakub: Vytvoril dokumentáciu pre frontend. Analyzoval logy zo Sentry a urobil aj ich vizualizáciu. Celé to následne spísal do dokumentácie. Refaktoroval aplikáciu, aby bolo možné jednoducho zmeniť port.

  Nikola: Analyzoval OWASP ZAP nástroj. Navrhol nový scenár s jeho využitím. Narazil aj na HUD nástroj rovnako v OWASP ZAP pre manuálne testovanie, ktorý ale nie je možné použiť.

  Peter: Komentoval kód na backende a robil ďalšiu dokumentáciu backendu.

- ***Čo sa nám nepodarilo vykonať?***

  Viktor: Nepodarilo sa mu zverejniť testy, ale do konca šprintu to stihne.

  Saleh: Neotestoval aplikáciu pretože ešte pribudnú zmeny po šprinte.

Jakub: Nepodarilo sa mu dostať k riešeniu ďalších problémov používateľov.
Peter: Odložil vylepšenie google forms, ktoré neboli na tento šprint naplánované.
Nikola: Čo mal naplánované urobil. Bol by rád keby sa navrhnutý scenár realizoval.

- ***Aké problémy sme identifikovali alebo máme?***
  Viktor: Nemá žiadne problémy.
  Saleh: Rovnako nemá problémy.
  Jakub: Komentovanie kódu zabralo veľmi veľmi veľa času.
  Peter: Nemá dostatok času na dokončenie implementácie session. Má časté výpadky internetu.
  Nikola: Nefungoval HUD v OWASP ZAP nástroji a teda realizácia jedného zo zaujímavých scenárov nebola možná.

- ***Čo by sme v nasledujúcom šprinte zlepšili?***
  Aby sme boli na volaniach všetci v rovnaký čas.

Záver

Úspešne pokračujeme vo vylepšovaní aplikácií. Zlepšená bola hlavne kvalita kódu a boli implementované niektoré požiadavky používateľov. Organizácia práce na projekte začína byť ovplyvnená blížiacim sa skúškovým obdobím a termínmi odovzdávania projektov k predmetom.

3.11 Jedenásty šprint

Posledný šprint tímového projektu bol venovaný finalizácii projektu ako aj tvorbe akčnej prezentácie. Vyriešili sme posledné problémy identifikované v používateľskom testovaní a zdokumentovali zvyšok pridaného kódu. Okrem dokumentácie sme kód aj refaktorovali. Na záver sme pripravili finálne Docker obrazy a aplikáciu aj pretestovali. Testovanie sme uskutočnili lokálne bez a s Dockerom.

Pokrok dosiahnutý na jedenástom šprinte

Posledný šprint letného semestra a tímového projektu bol orientovaný na dokončovacie práce a uspokojenie posledných požiadaviek používateľa. Nezabudli sme ani na prezentáciu produktu ako celku tvorbou akčnej prezentácie. Finalizácia projektu znovu zahŕňala zostavenie a otestovanie novej verzie, ktorá má byť funkčná a zahŕňať hlavne zmeny po používateľskom testovaní. Šprint má iba jeden týždeň, preto sme už nové scenáre nerealizovali.

Jakub spravil migrácie z pôvodnej Firebase databázy do lokálnej postgres databázy s použitím objektovo relačného mapovača Hibernate pre scenár s ukradnutím produktu. Migrácia bola dôležitá kvôli problémom s lokálnym nasadením využívajúcim Docker. Prihlásenie z druhých zariadení je pravdepodobne blokované pre druhých používateľov a veľmi dlho trvá, načo sa používatelia v prieskume sťažovali. Problém sme identifikoval až pri používateľskom prieskume, pretože všetci z tímu už mali prístup k Firebase databáze, a preto neidentifikovali žiadne problémy. Lokálna databáza je pripravená a bolo potrebné napísať príslušné triedy, ktoré sa mapujú na tabuľku. Následne sme prispôsobili konkrétne služby tejto funkcionalite. Pôvodnú funkcionalitu sme nechali v pôvodnom stave, aby bolo produkt možné nasadiť aj pôvodným spôsobom. Pre vzniknutý kód Jakub rovnako doplnil JavaDoc anotácie a komentáre, preto by kód mal aj naďalej zostať zdokumentovaný.

Ďalšou súčasťou bola tvorba akčnej prezentácie. Jakub vytvoril základnú kostru aj s animáciami, ktorá odráža princíp projektu, viaceré rozhodnutia a možné rozšírenie o Openstack Kypo. Obsahom prvej časti boli informácie čo produkt je a čo sú jeho silné stránky. V ďalšej časti prezentácie sme sa zamerali na prezentovanie hlavnej zápletky a jednotlivých úloh, ktoré používatelia budú musieť vyriešiť. Nevynechali sme prieskumnú analýzu, SQL injekcie, Bcrypt šifrovanie, prelamovanie hesiel, slovníkový útok, získavanie privilégií, čím sme používateľom ukázali prechod rôznymi fázami penetračného testovania až po finálne získanie vlajky. V predposlednej časti sme spomenuli náš návrh zahŕňajúci nasadenie na KYPO uzloch a podpora monitoringu a rôznych pomôcok pre používateľov. Hra by s ním bola interaktívnejšia a použitie Whois aplikácie zaujímavejšie a opodstatnenejšie. OpenStack KYPO ale stále nemáme a dlho sme verili, že aspoň na záver projektu bude k dispozícii. Na záver sme prezentovali aj myšlienku o konfigurovateľnosti aplikácie, ktorá by tak mala výhodu pred známym JuiceShopom a lepšie uplatnenie pre voľbu scenárov a ich prispôsobeniu používateľom.

Dôraz bol kladený aj na zlepšenie kvality kódu. Viktor sa snažil kód na backende lepšie refaktorovať použitím návrhových vzorov. Kvalita kódu by sa preto mala výrazne zlepšiť.

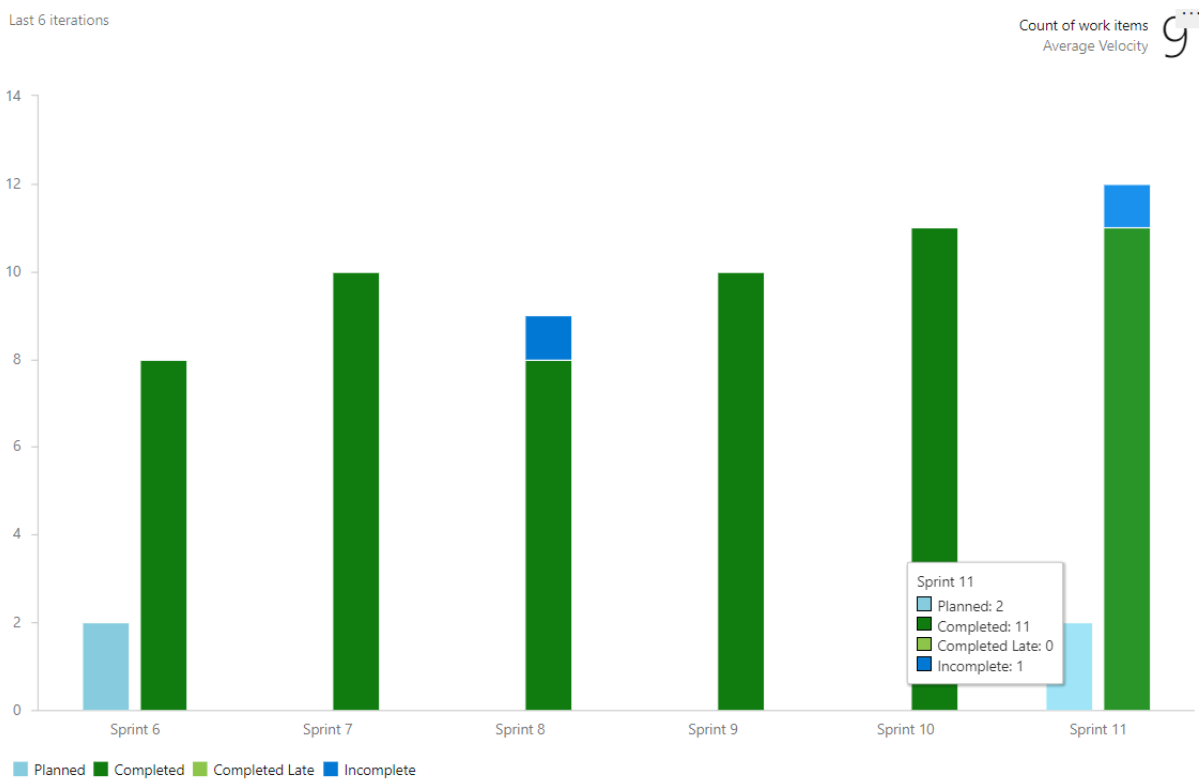
Jakub pre pridané časti pridal JavaDoc dokumentáciu. Na ostatné časti kódu už bola dokumentácia vytvorená, hlavne Petrom v minulom šprinte a Nikolom na konci zimného semestra. Nakoniec Peter vygeneroval výsledný JavaDoc pre backend, pričom ešte opravil niekoľko chýb v tejto dokumentácii.

Riešenie sme následne znovu otestovali hneď niekoľkými spôsobmi. Pokiaľ boli k dispozícii jednotkové testy tak sme ich vyhodnotili. Testovali sme bez ale aj s použitím Docker obrazov, nakoľko niektoré nastavenia sa odlišujú. Napríklad pri Dockeri musí byť zabezpečená aj komunikácia kontajnerov alebo emailový klient je pri lokálnom nasadení testovacia aplikácia. Nikola aplikáciu otestoval použitím Dockera na Windowse, Jakub zasa lokálne bez Dockera. Netreba zabudnúť ani na Salehove review kódu pred mergom do hlavnej vetvy.

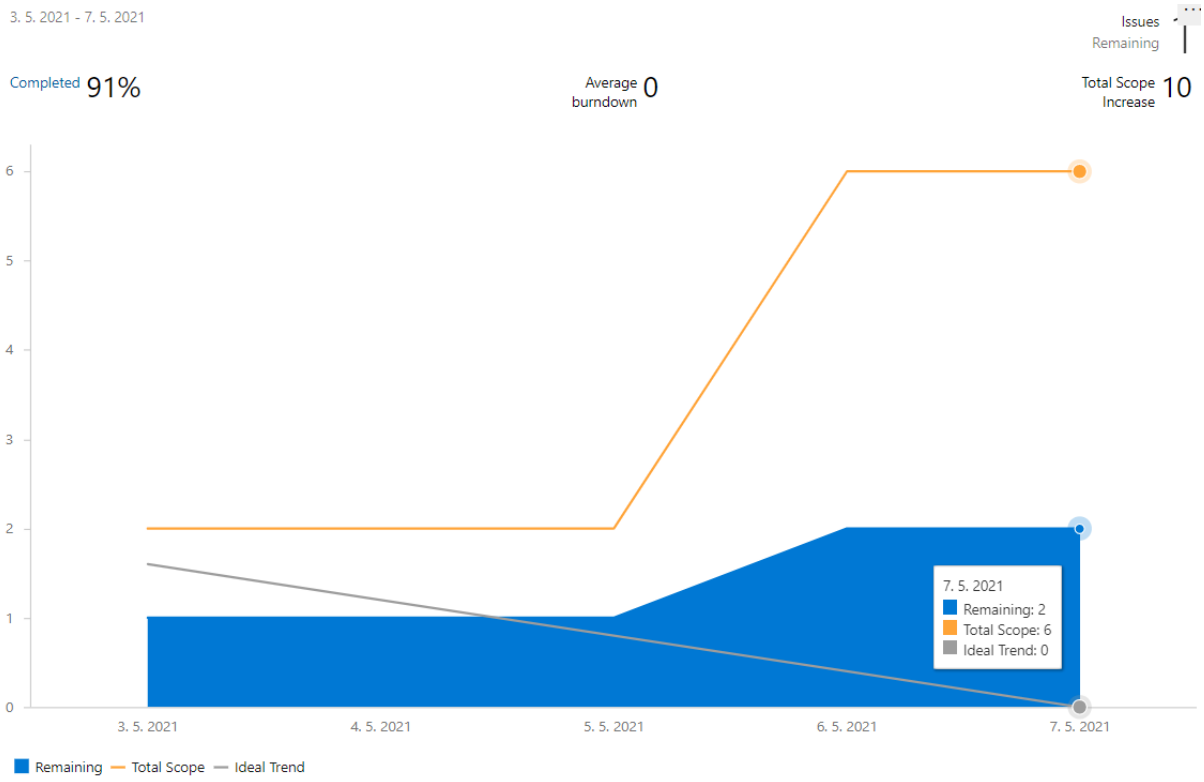
Zaoberali sme sa aj úlohou testovať aplikáciu iného tímu ako aj vytvoriť report z používateľského testovania podľa príslušnej šablóny. Peter kontaktoval niekoľko tímov. Niektoré sa vôbec neozvali a ostatné si nechceli svoje aplikácie dať otestovať. Jakub zatiaľ zostavil report z názorov používateľov z používateľského prieskumu a sám odoslal niekoľko šablón jednotlivým používateľom z prieskumu. Vyplnil tak report z používateľského prieskumu pre konkrétnu šablónu.

Aj napriek dĺžke posledného šprintu, ktorá bola iba týždeň, sa podarilo vyriešiť aj zvyšné používateľské problémy a vyrovnať sa velocity minulého šprintu. Väčšina úloh bola orientovaná ďalšie otestovanie celej funkcionality s dôrazom na tú pridanú. Migrácia celej funkcionality do lokálnej relačnej Postgres databázy umožnila chod nefunkčného scenára pri lokálnom nasadení s použitím Dockeru, čo bolo dôležité. Velocity v poslednom jedenástom šprinte je zobrazená na obrázku 36. Výkonnosť v šprinte je zobrazená na obrázku 37.

Posledný finálny šprint sa podarilo dokončiť a tesne pred ukončením sme dostali nápad na vylepšenie, možno už pre pokračujúci tím. Bolo by v ňom možné pridať honeypot, ďalšiu aplikáciu, umožňujúcu zistiť návyky útočníka. Útočník by mal detekovať, že to nie je plnohodnotná aplikácia a vyhnúť sa mu. Za vlámanie sa do takejto aplikácie by mu mohli byť strhnuté body a scenáre tak spraviť menej priamočiare a zaujímavejšie.



Obrázok 36: Velocity tímu v šprinte 11



Obrázok 37: Výkonnosť tímu v jedenástom šprinte

V tomto šprinte sme realizovali úlohy zobrazené v tabuľkách 31 a 32.

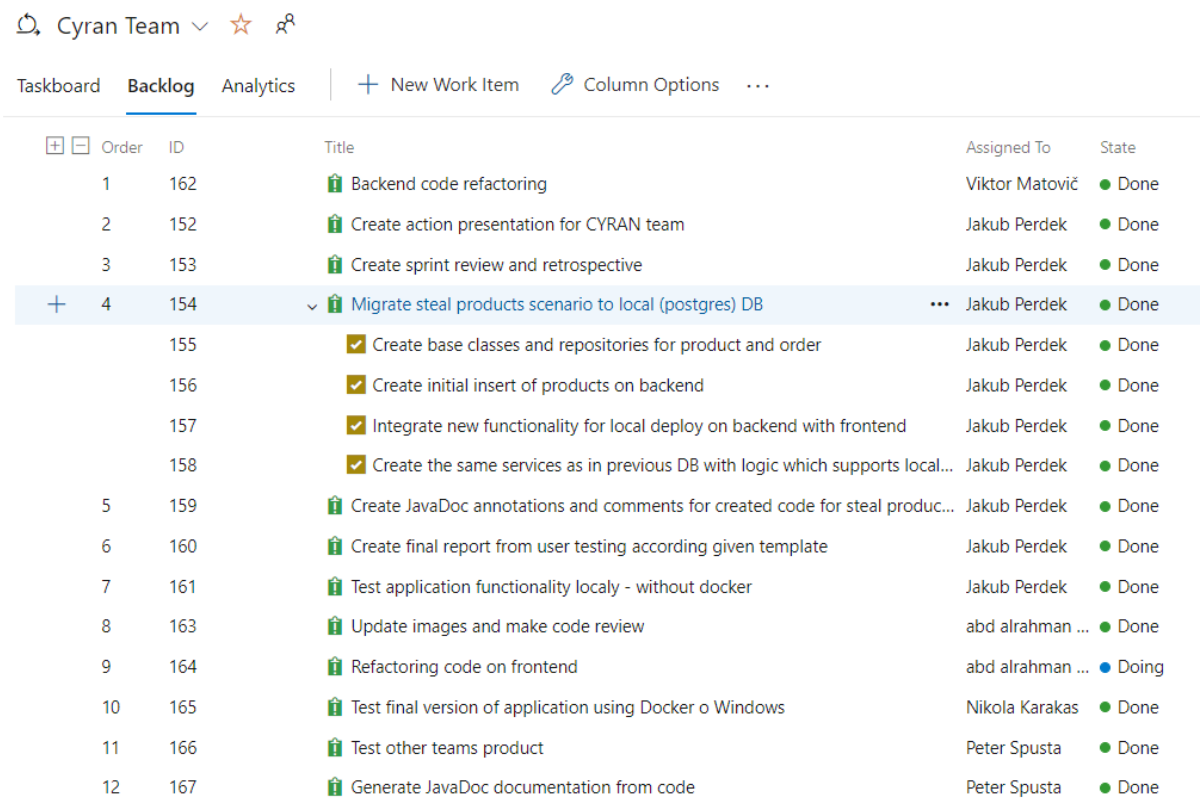
Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (nedeľa 09. 05.)	Šprint
Create action presentation for CYRAN team	Jakub Perdek	dokončené	šprint č. 11
Migrate steal products scenario to local (postgres) DB	Jakub Perdek	dokončené	šprint č. 11
Create base classes and repositories for product and order	Jakub Perdek	dokončené	šprint č. 11
Create initial insert of products on backend	Jakub Perdek	dokončené	šprint č. 11
Integrate new functionality for local deploy on backend with frontend	Jakub Perdek	dokončené	šprint č. 11
Create the same services as in previous DB with logic which supports local DB	Jakub Perdek	dokončené	šprint č. 11
Create JavaDoc annotations and comments for created code for steal product scenario	Jakub Perdek	dokončené	šprint č. 11
Test application functionality localy - without docker	Jakub Perdek	dokončené	šprint č. 11
Create final report from user testing according given template	Jakub Perdek	dokončené	šprint č. 11
Create sprint review and retrospective	Jakub Perdek	dokončené	šprint č. 11
Backend code refactoring	Viktor Matovič	dokončené	šprint č. 11

Tabuľka 31: Úlohy z jedenásteho šprintu - časť 1

Pridelená úloha	Zodpovedný riešiteľ	Aktuálny stav rozpracovania (nedeľa 09. 05.)	Šprint
Update images and make code review	Abd Alrahman Saleh	dokončené	šprint č. 11
Refactoring code on frontend	Abd Alrahman Saleh	rozpracované	šprint č. 11
Test final version of application using Docker o Windows	Nikola Karakaš	dokončené	šprint č. 11
Test other teams product	Peter Spusta	dokončené	šprint č. 11
Generate JavaDoc documentation from code	Peter Spusta	dokončené	šprint č. 11

Tabuľka 32: Úlohy z jedenásteho šprintu - časť 2

Export úloh z jedenásteho šprintu



Order	ID	Title	Assigned To	State
1	162	Backend code refactoring	Viktor Matovič	Done
2	152	Create action presentation for CYRAN team	Jakub Perdek	Done
3	153	Create sprint review and retrospective	Jakub Perdek	Done
4	154	Migrate steal products scenario to local (postgres) DB	Jakub Perdek	Done
	155	Create base classes and repositories for product and order	Jakub Perdek	Done
	156	Create initial insert of products on backend	Jakub Perdek	Done
	157	Integrate new functionality for local deploy on backend with frontend	Jakub Perdek	Done
	158	Create the same services as in previous DB with logic which supports local...	Jakub Perdek	Done
5	159	Create JavaDoc annotations and comments for created code for steal produc...	Jakub Perdek	Done
6	160	Create final report from user testing according given template	Jakub Perdek	Done
7	161	Test application functionality localy - without docker	Jakub Perdek	Done
8	163	Update images and make code review	abd alrahman ...	Done
9	164	Refactoring code on frontend	abd alrahman ...	Doing
10	165	Test final version of application using Docker o Windows	Nikola Karakas	Done
11	166	Test other teams product	Peter Spusta	Done
12	167	Generate JavaDoc documentation from code	Peter Spusta	Done

Obrázok 38: Export úloh z jedenásteho šprintu

Retrospektíva z jedenásteho šprintu

Dvojtýždňový šprint začal 03. mája a končí 09. mája 2021. Posledný šprint letného semestra a tímového projektu sme venovali finalizácii posledných prác na projekte.

Dátum a čas konania	Nedeľa 09. mája, od (cca) 20:05 – 21:25 hod.
Miesto konania	konferenčný hovor v General channel v Microsoft Teams
Retrospektíva za šprint:	03. mája- 09. mája
Účastníci	Jakub Perdek, Peter Spusta, Viktor Matovič, Nikola Karakaš, Abd Saleh
Spracovateľ	Jakub Perdek

Tabuľka 33: Informácie o retrospektíve z jedenásteho šprintu

Priebeh stretnutí

Účastníkom boli položené nasledujúce otázky s nasledujúcimi odpoveďami:

- ***Čo sa nám podarilo vykonať?***

  Viktor: Snažil sa refaktorovať kód. Aplikoval návrhové vzory.

  Saleh: Updatoval docker image do finálneho stavu a rovnako urobil review pri zlučovaní zmien.

  Jakub: Spravil analógiu pre tvorbu objednávky pre lokálne Docker nasadenie, keďže zo zariadení iných používateľov bola autentifikácia do Google FireBase blokovaná. Zároveň okomentoval tento kód a vytvoril akčnú prezentáciu produktu. Vytvoril report z názorov získaných počas používateľského prieskumu.

  Nikola: Testoval finálnu verziu s použitím dockeru.

  Peter: Zabezpečil testovania iného produktu iného tímu a vygeneroval JavaDoc dokumentáciu pre backend.

- ***Čo sa nám nepodarilo vykonať?***

  Viktor: Nepodarilo sa mu zlúčiť ešte zmeny z refaktoringu.

  Saleh: Nepodarilo sa mu spraviť refaktoring kódu na frontende, ktorý zamýšľal.

Jakub: Tesne pred ukončením šprintu sa dozvedel o honeypotoch, lákadlách útočníkov, ktoré by mohli byť pridané ako Docker obrazy a následne využité pre otestovanie skúseností používateľov ich detekovať. V prípade neodhalenia honeypotu by mohli byť za to penalizovaní pokiaľ sa nechajú nalákať. Nepodarilo sa mu vzhľadom na koniec šprintu túto funkcionalitu zakomponovať do projektu.
Peter: Nenašiel žiadny ochotný tím, ktorý by ponúkol ich projekt na testovanie.
Nikola: Všetko sa mu podarilo.

- ***Aké problémy sme identifikovali alebo máme?***
  Viktor: Neidentifikoval žiadne problémy.
  Saleh: Rovnako nemal problémy.
  Jakub: Viac vedľajších scenárov by sa zišlo.
  Peter: Nemá žiadne problémy.
  Nikola: Rovnako nemá žiadne problémy.

- ***Čo by sme v nasledujúcom šprinte zlepšili?***
  Zvoliť pevný čas stretnutí a dobre sa na ne pripraviť.
  Navrhovať ďalšie scenáre a lepšie plánovať ich realizáciu.
  Mať denné standupy.

Záver

Veríme, že sme nevytvorili prototyp na zahodenie, a že na tomto projekte bude môcť pokračovať druhý tím. Zároveň použitím aplikácie by sa mala zdokonaliť výučba, a prípadne sa nasadí s použitím KYPO ako bolo zamýšľané. Prínosom tímového projektu je aj to, že sme sa naučili rozbehať scrum v tíme, čo všetci hodnotíme pozitívne.

4 Globálna retrospektíva

4.1 Zimný semester

V zimnom semestri sme ako tím nadobudli skúsenosti s používaním Scrumu a nástrojov pomáhajúcich pri jeho realizácii. Uvádzame preto zhrnutie týchto skúseností pre zimný semester.

Z čoho sme sa poučili a čo sme sa naučili

- Nečakať kým niekto bude mať čas implementovať nejakú funkcionalitu
- Robiť review je podstatné
- Naučili sme sa riešiť problémy s chýbajúcimi zdrojmi – KYPO
- Pracovať v Scrum tíme
- Pracovať v šprintoch, na denných standupoch, robiť backlog
- Duda nám pomohol s metodikou Skrumu

Z čoho sa poučíme

- Venovať viac času tímovému projektu
- Nenechávať si všetko na poslednú chvíľu
- Robiť viac jednotkových testov
- Viac dokumentovať kód – Javadoc a ďalšie formy
- Každý by mal byť vždy prítomný na stretnutiach
- Pýtať si produktový backlog
- Používať ďalšie nástroje na spravovanie backlogu

V čom zostaneme poučení a čo naďalej praktizovať

- Pri komunikovaní každej nevyhnutnej veci
- Aktívne sa zaujímame o dianie na projekte
- Pomoc iným členom tímu zrýchli prácu na projekte
- Zostaneme používať Azure DevOps
- Nespoliehať sa na personál Muni pre KYPO

4.2 Letný semester

Letný semester bol semestrom, v ktorom sme už mali základné informácie s používaním Scrumu. Zároveň semester bol výnimočný používateľským testovaním, z ktorého spätnej väzby sme sa inšpirovali a mohli následne zlepšovať zhotovené riešenie.

Z čoho sme sa poučili a čo sme sa naučili

- Integrovať viaceré docker kontajnere v tíme
- Prispôsobiť aplikáciu pre lokálne nasadenie po tom čo kolegovia identifikovali nefunkčnosť pôvodného návrhu
- Efektívnejšie pracovať s technológiami
- Získanie poznatkov o nových zraniteľnostiach a obrane voči nim
- Zlepšenie procesu dokumentácie v scrum projekte
- Osvojili sme si párové programovanie

Z čoho sa poučíme

- Mali by sme striktnejšie dodržiavať vlastné metodiky ako definition of done
- Dokončiť aspoň niekoľko taskov počas šprintov každý
- Nenechávať si písanie dokumentácie kódu na koniec
- V budúcnosti by sme mohli vyskúšať BDD
- Nenechávať si dokončenie úloh na koniec šprintu lebo sa nestihnú
- Mať testovacie prostredie
- Spraviť beta testing pred používateľským testovaním

V čom zostaneme poučení a čo naďalej praktizovať

- Lepšie manažovať kód – git-flow
- Priebežne robiť continuous integration a continuous deployment aplikácií
- Testovať na rôznych operačných systémoch aj v prípade Dockera
- Udržovať verzie pre lokálne aj globálne nasadenie

Príloha A: Motivačný dokument: Tím 19

1. Predstavenie tímu - členovia tímu

Peter Spusta
Abd alrahman saleh
Viktor Matovič
Jakub Perdek
Nikola Karakaš
Miroslav Balga

Členovia nášho tímu prišli s odporúčanými technológiami pre projekty z kapitoly 2 do styku v akademickom prostredí ako aj v prostredí praxe. Svoje skúsenosti nadobudli pri tvorbe informačných systémov ako aj webových stránok / prezentácií pre komerčné subjekty. Nehľadiac na záber projektov pri ktorých nadobúdali svoje skúsenosti sa v tíme integrovali softvéroví špecialisti na rozličné a zároveň moderné serverové a klientske riešenia. Každý člen tímu preukázal schopnosť kolaboratívne pracovať a riešiť tímové úlohy, schopnosť navrhnúť, konštruovať, vytvoriť a otestovať riešenie produktu na ktorého implementácii sa podieľal. Vzhľadom na doterajšie výsledky prezentované navzájom je každý člen tímu schopný prevziať zodpovednosť za dodanie samostatného a komplexného softvérového produktu. V nasledujúcej tabuľke uvádzame vybrané nástroje a technológie v ktorých členovia tímu preukázali svoje doterajšie praktické skúsenosti:

FRONT-END	BACK-END	Tools / Middleware
Javascript Typescript Angular 2+ Css Scss - React Native - React - HTML 5	Laravel Django Java EE Node JS Postgress(db) MS(db)	Docker Bash, R Java C/C++ Python (Scikit-learn), Keras (Tensorflow, Theano), Sci-kit learn

Building an information system isn't the only thing we're looking for but having an expandable system where it's gonna make it easier to add new services based on the university needs, a user-friendly system which will make it inserting for the students to use it.

Ofcourse building such a system is not going to easy, a plenty of services are upon us, but with the great team we have we're prepared, we're greatly motivated to build a system not just for our own benefit, but to make it on production for our faculty, we will provide most of the services which is needed.

Our team is very well prepared for building it with the newest technologies, such angular 2+ and nodejs, providing a very well documented project which will make it easier to be expanded later.

From our view, and based on real interviews with employees in our faculty, it will be our first move towards a stable system, easy to use and integrate with other websites such as google calendar to assign the semester schedule there.

There are two main categories of coding, scripting and programming which we're considering to use based on our practical nad very well background experience, as well as a very well done projects :

Client Side Scripting / Coding:

- HTML5 (HyperText Markup Language)
- CSS (Cascading Style Sheets), SCSS
- TypeScript, JavaScript
- angular 9

Server Side Scripting / Coding:

- Nodejs 12.8.4
- Postgres or MSS for database
- Docker
- Python

Sme pripravení prijať túto výzvu.

2. Motivácia k spracovaniu tém

V nasledujúcich odrážkach sa čitateľovi snažíme poskytnúť komplexný a prehľadný náhľad na doteraz preukázané schopnosti členov tímu, ktoré si chcú pri vybraných témach nižšie doplniť získaním nových vedomostí a osvojením si konkrétnych techník používaných pri práci s technológiami, ktoré tieto projekty vyžadujú:

A. Podporný informačný systém pre študijné oddelenie (19)
B. Automatické rozpoznávanie spektier (8)
C. FIFé Medzinárodná výstava mačiek (18)

2.1. Podporný informačný systém pre študijné oddelenie

Pre zhotovenie informačného systému pre študijné oddelenie by sme vedeli ponúknuť naše zručnosti v oblasti webových technológií a návrhu informačných systémov. Systém vnímame potrebu vytvoriť použitím agilnej metodológie (pre SDLC), teda opakovaným zhotovovaním prototypov na rôznej úrovni deskriptívnosti s odkomunikovaním dôležitých čŕt systému. Prototypy by sme upravovali podľa získaných a upravovaných požiadaviek. Neoddeliteľnou súčasťou práce na projekte je aj modelovanie biznis procesov na základe ktorých by sme boli schopní vyhodnotiť potrebu webových formulárov, ale aj vyhodnotiť nastavenia prvkov používateľského rozhrania. Na základe získanej spätnej väzby pre prototypy by sme dopĺňali formuláre a spresňujúce komponenty, ktoré by sme naštýlovali podľa potrieb a požiadaviek zákazníka. Disponujeme ľuďmi so znalosťami CSS. V prípade potreby vieme využiť skúsenosti členov pri tvorbe štýlovania rozhraní s pomocou SCSS. Dôraz by sme kládli na responzívnosť a prístupnosť webovej aplikácie pre mobilné zariadenia. Celý systém podrobne zdokumentujeme v rôznych formách a podobách. Formou biznis procesov, prototypov, ale aj hotových šablón. Neoddeliteľnú súčasť tvorí vývoj s použitím jazyka Javascript, pri ktorom a vzhľadom na ekosystém tvorby aplikácii v tomto jazyku (npm) vidíme príležitosť ho využiť pre vývoj v celom softvérovom projekte. V tíme máme ľudí so znalosťami aj ďalších kompilovaných a interpretovaných jazykov a rámcov, pokiaľ by bolo nutné naprogramovať aplikáciu v nejakom inom jazyku. Členovia tímu disponujú dostatočnou znalosťou pri práci s databázami, relačnými aj objektovými, modernými a často používanými riešeniami poskytujúcimi úložisko údajov. Jazyk EcmaScript aj s jeho ďalšími časťami sme schopní s pomocou dodatočných nástrojov a doplnkov webových rámcov minifikovať, a v optimalizovanom formáte pripraviť pre nasadenie v produkčnom prostredí. Riešenie by sme

preto mohli exportovať aj ako docker image, aby ho bolo jednoduchšie nasadiť napríklad na AWS.

Problematika je nám ako študentom z väčšej miery známa, pretože na oddelení niektoré rôzne problémy opakovane riešime. Veríme, že náš návrh, vývoj systému až po nasadenie by viedli k výslednému plnohodnotnému informačnému systému a dokázali by pomôcť pri riešení problémov na študijnom oddelení. V aplikácii vnímame ako podstatný dobrý vyhľadávací systém, umožňujúci orientovať sa vo veľkom množstve otázok a problémov. V analýze by sme sa preto venovali prípadnému použitiu NOSQL databázy a technikami pri vyhľadávaní ako napríklad vhodnej voľbe indexov a indexovania obsahu. Obsahom spomínaných prototypov by mohol byť prehľad študentov s niektorými nevybavenými povinnosťami, rovnako detail informácií o študijných záležitostiach každého študenta, ktorý by bol zobrazený po špecifickej žiadosti od autorizovanej študijnej referentky. Študenti by mohli vyhľadávať a prezerať si rôzne odpovede a problémy ostatných. Časté otázky by boli umiestnené do FAQ. Prototypy by mali byť dostatočne prehľadné, mali by obsahovať špecifické informácie a navigačné prvky z tejto domény, ale aj jednoduché, keďže už existujú rôzne systémy pre komunikáciu študentov, akým je napríklad Askalot, na ktorom často riešia problémy spojené so študijným oddelením. Vnímame preto šablóny a ich štýlovanie za dôležitý prvok pre čo najväčšiu zrozumiteľnosť a čo najväčší používateľský zážitok. Kľúčovým môže byť preto overenie spätnej väzby od študentov, ktorú by sme v rámci riešenia chceli zrealizovať.

Motiváciou je aj vývoj podporných učebných nástrojov niektorými z nás. Sú nimi snaha vizualizovať Karnaughovu mapu, konštrukcia fraktálov alebo aj efektívne generovanie náhodných bludísk s dôrazom na ich náhodnosť.

2.2. Automatické rozpoznávanie spektier

Teoretické základy ako predpoklad na uchádzanie sa o túto tému sme získali po absolvovaní predmetov Umelá Inteligencia, Objavovanie znalostí a Vyhľadávanie informácií. Počas práce na seminárnych zadaniach v rámci predmetov Objavovanie znalostí a Vyhľadávanie informácií sme si osvojili techniky spracovania veľkého množstva dát, v štruktúrovanej alebo neštruktúrovanej podobe z heterogénneho prostredia Webu.

S jazykom Python, v ktorom sú často implementované nástroje na prehliadanie a zbieranie dát z Webu (Web Scrypers) sme sa naučili pracovať na realizácii expertných úloh, spočívajúcich v spracovaní, klasifikácii, vizualizácii a v neposlednom rade interpretácii

informácií abstrahovaním zo získanej dátovej množiny. V rámci riešenia by sme vedeli aplikovať a následne porovnať rôzne algoritmy realizované pomocou strojového učenia najmä v Scikit-learn a neurónových sieťach s využitím frameworku Keras. Zaujímame sa aj o problematiku lineárnej regresie a ďalších algoritmov ako SVM alebo Naivný Bayes, ktoré by sme rovnako implementovali a vizualizovali v jazyku R. Cieľom by bolo porovnať rôzne metriky ako F1 a správnosť, ale aj voľba algoritmov, ktoré sú dobre interpretovateľné.

Nakoľko sa od spracovateľov projektu očakáva realizovať podobné úlohy, nás, ako možných riešiteľov motivuje možnosť pracovať s rozhraním a výstupom ojedinele používaného (expertmi doménovej a aplikačnej oblasti) zariadenia, označovaného ako IMS spektrometer. S požadovaným expertným systémom (alebo ako súčasť riešenia) sme sa počas štúdia Umelej inteligencie mohli oboznámiť, realizácia riešenia pre túto tému nám môže poskytnúť príležitosť takýto systém aj vytvoriť. O tému taktiež prejavujeme záujem v dôsledku faktu, že takúto úlohu je možné realizovať len po osvojení si teoretickej základne danej domény. Realizáciu tejto úlohy berieme ako výzvu.

2.3. FIFé Medzinárodná výstava mačiek

Tému tohto projektu sme vybrali ako jednu z najlepších pre náš tímový projekt a to najmä z hľadiska znalostí a vedomostí nášho tímu.

Pre zhotovenie informačného systému pre študijné oddelenie by sme vedeli ponúknuť naše zručnosti v oblasti webových technológií a návrhu informačných systémov. Systém by sme vyvíjali opakovaným zhotovovaním prototypov na rôznej úrovni deskriptívnosti s odkomunikovaním dôležitých čŕt systému, a to aj pre lepšiu spätnú väzbu. Následne by sme upravili prototypy podľa požiadaviek. Neoddeliteľnou súčasťou je aj modelovanie biznis procesov, na základe ktorých by sme boli schopní vyhodnotiť potrebu formulárov. Na základe prototypov by sme napokon vytvorili formuláre a komponenty, ktoré by sme naštýlovali podľa potrieb.

Bolo by pre nás výzvou navrhnúť dizajn a realizovať požiadavky aplikácie, ktorá je využívaná pre výstavy mačiek a obsahuje iba základné užívateľské rozhranie podobné tomu textovému. Ako študenti FIIT mame všetci skúsenosti s vývojom softvéru od výberu vhodných technológií, cez návrh, až po implementáciu a nasadenie softvéru. Viacerí z nás majú aj pracovné skúsenosti s vývojom aplikácií a všetci sa radi učíme nové veci. Preto si myslíme že táto téma by bola pre nás vhodná a umožnila by nám ďalej rozvíjať naše schopnosti.

Tento projekt by nás mohol posunúť od implementácie imaginárnych nápadov k realizácii skutočného a užitočného projektu, pracujúceho so skutočnými údajmi, ako aj k implementácii podľa mnohých odporúčaných štandardov, z ktorých by sme sa mohli veľa naučiť.

Veríme že využitím znalostí nášho tímu vieme vytvoriť skvelý projekt a získané znalosti nám v budúcnosti otvoria nové príležitosti pre prácu s mobilnú aplikáciu pre zariadenia Android aj IOS.

3. Preferencie projektov

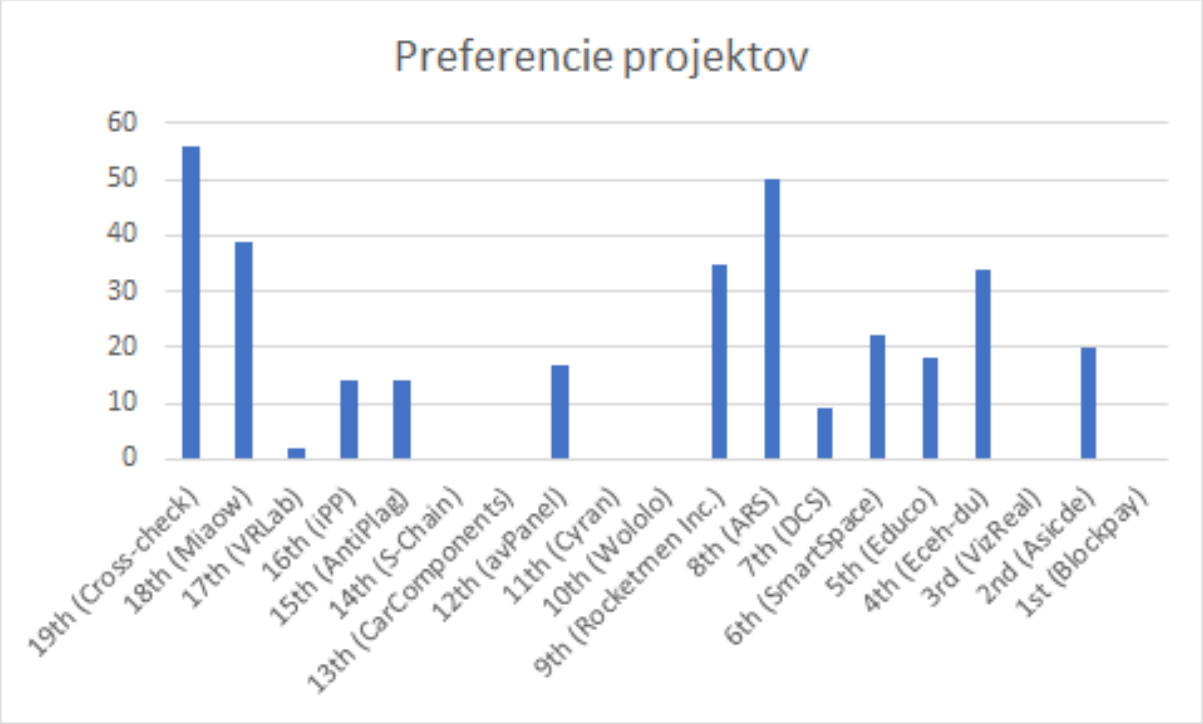
Po konzultáciách v rámci tímu, spoznávaní a získaní informácií o svojich doterajších skúsenostiach sme sa rozhodli uchádzať o témy Tímového projektu v tomto poradí:

1. miesto: (najviac želané): Téma č. 19, podporný informačný systém
2. miesto: Téma č. 8, rozpoznávanie spektier
3. miesto: Téma č. 18, inteligentný informačný systém pre výstavy
4. miesto: Téma č. 9, monitorovanie zdravotného stavu
5. miesto: Téma č. 4, databáza otázok a odpovedí
6. miesto: Téma č. 6, transformácia priestorov pre prácu
7. miesto: Téma č. 2, webové IDE pre ASIC
8. miesto: Téma č. 5, orchestračný portál
9. miesto: Téma č. 12, analýza dát pre autonómne vozidlo
10. miesto: Téma č. 15, vyhľadávač podobností textu
11. miesto: Téma č. 16, informačný systém pre verejné obstarávanie
12. miesto: Téma č. 11, testovanie kybernetickej ochrany

4. Hlasovacia tabuľka

Priorita **Meno**	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
Nikola	8	7	19	9	6	18	12	15	16	4
Viktor	19	8	2	18	5	4	6	9	16	12
Peter	9	18	19	4	2	8	12	5	17	16
Jakub	19	8	5	18	4	15	6	9	16	12
Saleh	19	8	18	12	15	4	6	9	16	17
Miro	19	9	8	4	2	16	6	18	5	17

Táto tabuľka zobrazuje preferencie jednotlivých členov tímu.



Tento graf zobrazuje preferencie nášho tímu pre všetky projekty

5. Rozvrh voľných hodín pre konzultácie

Deň v týždni / Účastník	Pondelok	Utorok	Streda	Štvrtok	Piatok
Nikola	8.00 AM - 16:00 PM	8.00 AM - 14:00 PM	8:00 AM- 15:00 PM	10:00 AM - 12:00 AM 16:00 PM - 22 :00 PM	Celý deň
Viktor	8.00 A.M- 3:50 P.M, 6:00 P.M - 7:50 P.M.	8:00 A.M- 1:50 P.M	10:00 A.M-2:50 P.M.	4:50 P.M.-7:50 P.M.	Celý deň
Peter	-------	7:00 P.M.	---------	6:00 P.M.	2:00 P.M.
Jakub	8:00 A. M. - 11: 50 A.M, 2:00 P. M - 3:50 PM, 6:00 P. M - 7:50 P. M.	8:00 A.M. - 1:50 P.M.	8:00 A. M - 9:50 A. M, 0:00 P. M. - 2:50 P. M.	10:00 A. M:- 11:50 A. M, 2:00 P.M - 9:00 P.M	8:00 A. M. - 11:00 A. M.
Saleh	9:00 -> 12:00 14:00 -> 17:00	8:00 -> 13:00	10:00 -> 12:30	10:00->14:00 16:00->23:00	----------
Miro	6:00 P.M - 9:00 P.M	7:00 P.M - 9:00 P.M	4:00 P.M - 7:00 P.M	4:00 P.M - 9:00 P.M	4:00 P.M - 9:00 P.M

Rozpis voľného času pre stretnutia:

2 + 2 pre tím:

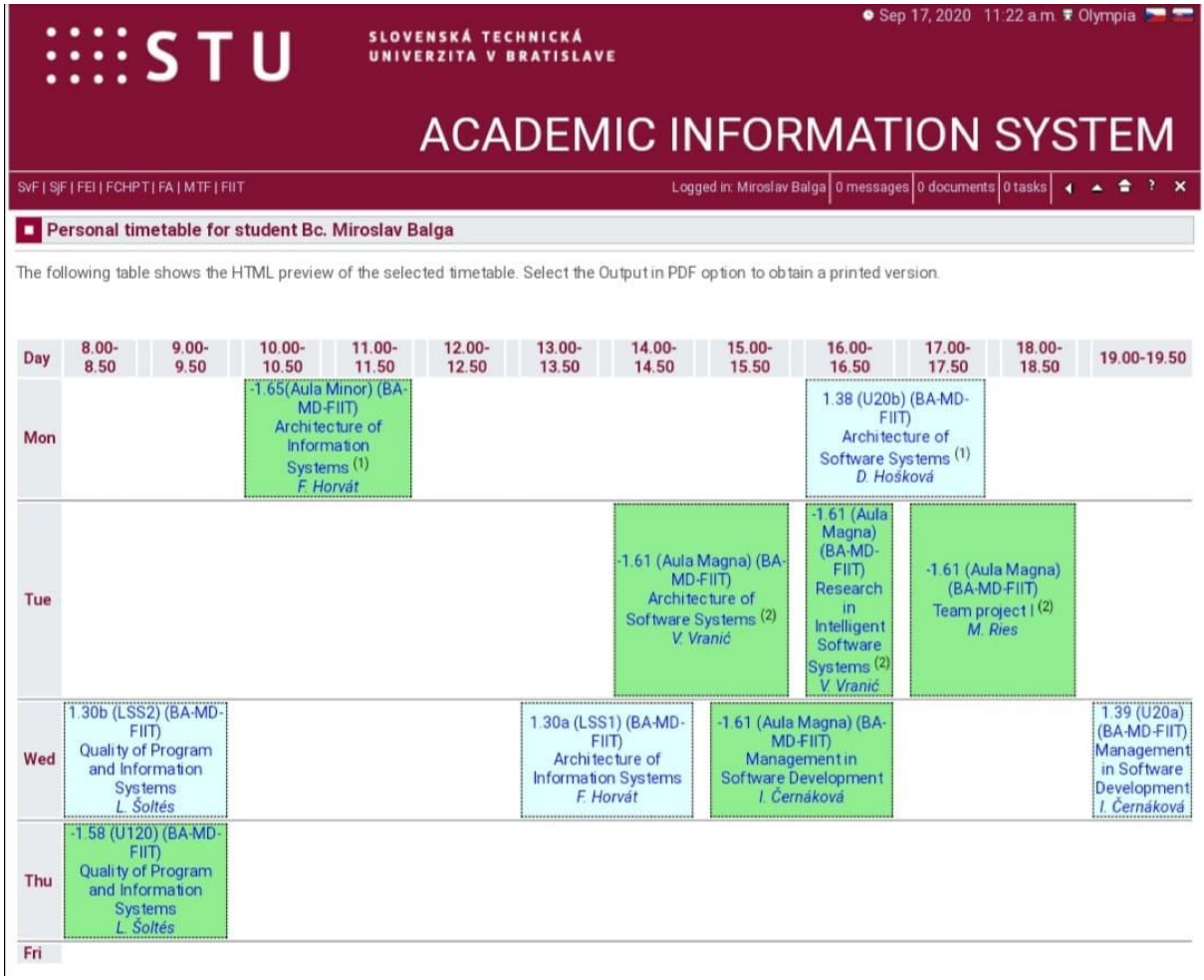
1- každý štvrtok od 20:00 do 00:00

2- v pondelok od 19:00 do 21:00 + každý štvrtok od 20:00 do 22:00

3 hodiny s vedúcim:

1-každý utorok od 08:00 do 11:00

2-Štvrtok od 18:00 do 21:00



Sep 17, 2020 11:22 a.m. Olympia

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

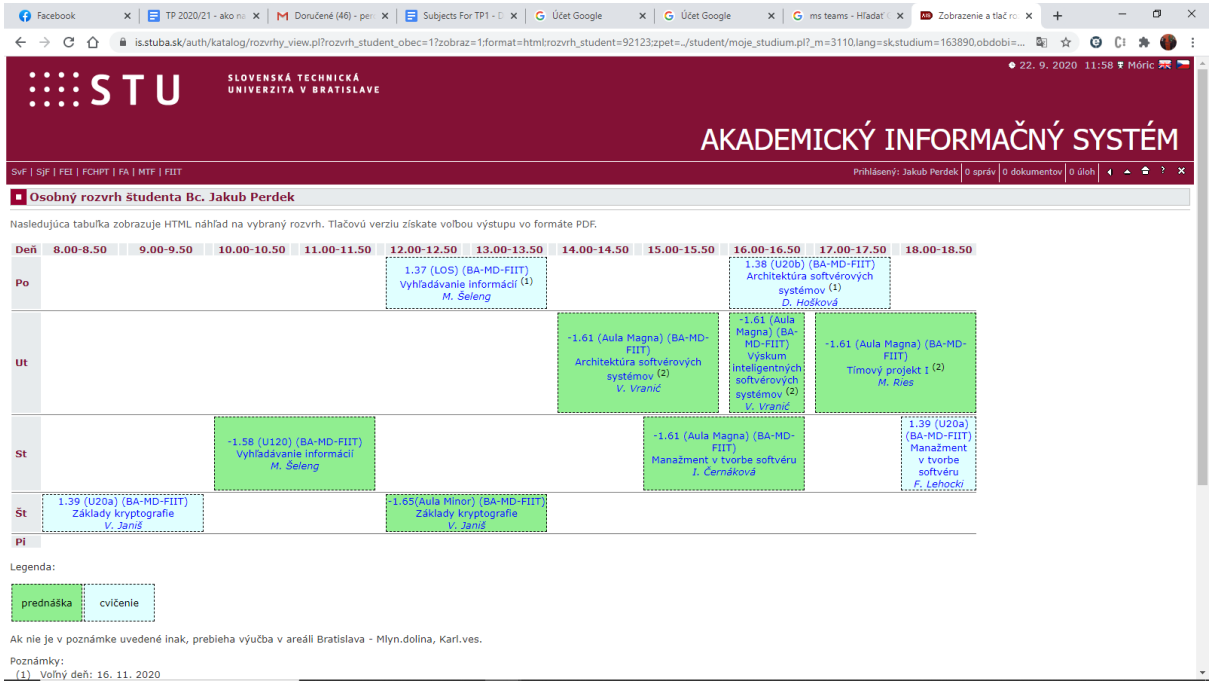
ACADEMIC INFORMATION SYSTEM

SvF | SjF | FEI | FCHPT | FA | MTF | FIIT — Logged in: Miroslav Balga | 0 messages | 0 documents | 0 tasks

Personal timetable for student Bc. Miroslav Balga

The following table shows the HTML preview of the selected timetable. Select the Output in PDF option to obtain a printed version.

Day	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50
Mon			-1.65(Aula Minor) (BA-MD-FIIT) Architecture of Information Systems (1) *F. Horvát*						1.38 (U20b) (BA-MD-FIIT) Architecture of Software Systems (1) *D. Hošková*			
Tue							-1.61 (Aula Magna) (BA-MD-FIIT) Architecture of Software Systems (2) *V. Vranić*		-1.61 (Aula Magna) (BA-MD-FIIT) Research in Intelligent Software Systems (2) *V. Vranić*	-1.61 (Aula Magna) (BA-MD-FIIT) Team project I (2) *M. Ries*		
Wed	1.30b (LSS2) (BA-MD-FIIT) Quality of Program and Information Systems *L. Šoltés*					1.30a (LSS1) (BA-MD-FIIT) Architecture of Information Systems *F. Horvát*		-1.61 (Aula Magna) (BA-MD-FIIT) Management in Software Development *I. Černáková*				1.39 (U20a) (BA-MD-FIIT) Management in Software Development *I. Černáková*
Thu	-1.58 (U120) (BA-MD-FIIT) Quality of Program and Information Systems *L. Šoltés*											
Fri												



is.stuba.sk/auth/katalog/rozvrhy_view.pl?rozvrh_student_obec=1?zobraz=1;format=html;rozvrh_student=92123;zpet=../student/moje_studium.pl?_m=3110,lang=sk,studium=163890,obdobi=...

22. 9. 2020 11:58 Móric

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

AKADEMICKÝ INFORMAČNÝ SYSTÉM

SvF | SjF | FEI | FCHPT | FA | MTF | FIIT — Prihlásený: Jakub Perdek | 0 správ | 0 dokumentov | 0 úloh

Osobný rozvrh študenta Bc. Jakub Perdek

Nasledujúca tabuľka zobrazuje HTML náhľad na vybraný rozvrh. Tlačovú verziu získate voľbou výstupu vo formáte PDF.

Deň	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50
Po					1.37 (LOS) (BA-MD-FIIT) Vyhľadávanie informácií (1) *M. Šeleng*				1.38 (U20b) (BA-MD-FIIT) Architektúra softvérových systémov (1) *D. Hošková*		
Ut							-1.61 (Aula Magna) (BA-MD-FIIT) Architektúra softvérových systémov (2) *V. Vranić*		-1.61 (Aula Magna) (BA-MD-FIIT) Výskum inteligentných softvérových systémov (2) *V. Vranić*	-1.61 (Aula Magna) (BA-MD-FIIT) Tímový projekt I (2) *M. Ries*	
St			-1.58 (U120) (BA-MD-FIIT) Vyhľadávanie informácií *M. Šeleng*					-1.61 (Aula Magna) (BA-MD-FIIT) Manažment v tvorbe softvéru *I. Černáková*			1.39 (U20a) (BA-MD-FIIT) Manažment v tvorbe softvéru *F. Lehocki*
Št	1.39 (U20a) (BA-MD-FIIT) Základy kryptografie *V. Janiš*				-1.65(Aula Minor) (BA-MD-FIIT) Základy kryptografie *V. Janiš*						
Pi											

Legenda:

prednáška | cvičenie

Ak nie je v poznámke uvedené inak, prebieha výučba v areáli Bratislava - Mlyn.dolina, Karl.ves.

Poznámky:
(1) Voľný deň: 16. 11. 2020

Osobný rozvrh študenta Bc. Peter Spusta

Nasledujúca tabuľka zobrazuje HTML náhľad na vybraný rozvrh. Tlačovú verziu získate voľbou výstupu vo formáte PDF.

Deň	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50
Po					1.39 (U20a) (BA-MD-FIIT) Systémové myslenie v IT (1,2) *R. Kazička*		1.38 (U20b) (BA-MD-FIIT) Architektúra softvérových systémov (1) *D. Hošková*		-1.58 (U120) (BA-MD-FIIT) Nové médiá v spoločnosti (1) *A. Hrčková*		-1.58 (U120) (BA-MD-FIIT) Nové médiá v spoločnosti (1) *A. Hrčková*	
Ut							-1.61 (Aula Magna) (BA-MD-FIIT) Architektúra softvérových systémov (3) *V. Vranić*		-1.61 (Aula Magna) (BA-MD-FIIT) Výskum inteligentných softvérových systémov (3) *V. Vranić*	-1.61 (Aula Magna) (BA-MD-FIIT) Tímový projekt I (3) *M. Ries*		
St								-1.61 (Aula Magna) (BA-MD-FIIT) Manažment v tvorbe softvéru *I. Černáková*		1.31a (BA-MD-FIIT) Systémové myslenie v IT (2,4) *R. Kazička*		1.39 (U20a) (BA-MD-FIIT) Manažment v tvorbe softvéru *F. Lehocki*
Št							1.31a (BA-MD-FIIT) Návrh a vývoj počítačových hier *D. Dolhá*		3.08 (zasUISI) (BA-MD-FIIT) Návrh a vývoj počítačových hier (5) *M. Ferko*			
Pi												

Osobný rozvrh študenta Bc. Abd Alrahman Saleh

Nasledujúca tabuľka zobrazuje HTML náhľad na vybraný rozvrh. Tlačovú verziu získate voľbou výstupu vo formáte PDF.

Deň	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50
Po					1.38 (U20b) (BA-MD-FIIT) Bezpečnosť informačných technológií (2) *M. Pikula*					1.19 (PU3) (BA-MD-FIIT) Počítačové a komunikačné siete (1,2) *K. Košťál*	
						-1.61 (Aula Magna) (BA-MD-FIIT) Počítačové a komunikačné siete (2) *I. Kotuliak*					
Ut								1.37 (LOS) (BA-MD-FIIT) Výskum v informačnej bezpečnosti (3) *I. Kotuliak*		-1.61 (Aula Magna) (BA-MD-FIIT) Tímový projekt I (3) *M. Ries*	
St	1.38 (U20b) (BA-MD-FIIT) Bezpečnosť informačných technológií *M. Pikula*						1.40 (U40) (BA-MD-FIIT) Penetračné testovanie *I. Kotuliak*			-1.40 (PU1) (BA-MD-FIIT) Penetračné testovanie *I. Kotuliak*	
Št							-1.65(Aula Minor) (BA-MD-FIIT) Manažment informačnej bezpečnosti *I. Kotuliak*				
Pi											

Personal timetable for student Bc. Viktor Matovič

The following table shows the HTML preview of the selected timetable. Select the Output in PDF option to obtain a printed version.

Day	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50
Mon									1.38 (U20b) (BA-MD-FIIT) Architecture of Software Systems (1) *D. Hošková*			
Tue							-1.61 (Aula Magna) (BA-MD-FIIT) Architecture of Software Systems (2) *V. Vranić*		-1.61 (Aula Magna) (BA-MD-FIIT) Research in Intelligent Software Systems (2) *V. Vranić*	-1.61 (Aula Magna) (BA-MD-FIIT) Team project I (2) *M. Ries*		
Wed	1.30b (LSS2) (BA-MD-FIIT) Quality of Program and Information Systems *J. Petrík*							-1.61 (Aula Magna) (BA-MD-FIIT) Management in Software Development *I. Černáková*				1.39 (U20a) (BA-MD-FIIT) Management in Software Development *F. Lehocki*
Thu	-1.58 (U120) (BA-MD-FIIT) Quality of Program and Information Systems *L. Šoltés*				1.39 (U20a) (BA-MD-FIIT) Aspect-Oriented Software Development *V. Vranić*		1.39 (U20a) (BA-MD-FIIT) Aspect-Oriented Software Development *V. Vranić*					
Fri												

Key:

Personal timetable for student Bc. Nikola Karakaš

The following table shows the HTML preview of the selected timetable. Select the Output in PDF option to obtain a printed version.

Day	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50
Mon									1.37 (LOS) (BA-MD-FIIT) Innovative entrepreneurship in ICT (1) *M. Zajko*		1.37 (LOS) (BA-MD-FIIT) Innovative entrepreneurship in ICT (1) *M. Zajko*	
Tue							-1.61 (Aula Magna) (BA-MD-FIIT) Architecture of Software Systems (2) *V. Vranić*		-1.61 (Aula Magna) (BA-MD-FIIT) Research in Intelligent Software Systems (2) *V. Vranić*	-1.61 (Aula Magna) (BA-MD-FIIT) Team project I (2) *M. Ries*		
Wed								-1.61 (Aula Magna) (BA-MD-FIIT) Management in Software Development *I. Černáková*			1.39 (U20a) (BA-MD-FIIT) Management in Software Development *F. Lehocki*	
Thu	1.39 (U20a) (BA-MD-FIIT) Introduction to Cryptography *V. Janiš*				-1.65(Aula Minor) (BA-MD-FIIT) Introduction to Cryptography *V. Janiš*		-1.40 (PU1) (BA-MD-FIIT) Architecture of Software Systems *L. Graf*					
Fri												

Miroslav Balga:

Deň	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50
Po			-1.65(Aula Minor) (BA-MD-FIIT) Architektúra informačných systémov (1) *F. Horvát*						1.38 (U20b) (BA-MD-FIIT) Architektúra softvérových systémov (1) *D. Hošková*			
Ut							-1.61 (Aula Magna) (BA-MD-FIIT) Architektúra softvérových systémov (2) *V. Vranić*		-1.61 (Aula Magna) (BA-MD-FIIT) Výskum inteligentných softvérových systémov (2) *V. Vranić*	-1.61 (Aula Magna) (BA-MD-FIIT) Tímový projekt I (2) *M. Ries*		
St	1.30b (LSS2) (BA-MD-FIIT) Kvalita programových a informačných systémov *J. Petrík*					1.30a (LSS1) (BA-MD-FIIT) Architektúra informačných systémov *B. Bindas*		-1.61 (Aula Magna) (BA-MD-FIIT) Manažment v tvorbe softvéru *I. Černáková*				1.39 (U20a) (BA-MD-FIIT) Manažment v tvorbe softvéru *F. Lehocki*
Št	-1.58 (U120) (BA-MD-FIIT) Kvalita programových a informačných systémov *L. Šoltés*											
Pi												

6. Mailový kontakt na tím

Pre kontaktovanie tímu použite mailovú adresu:

1- perdek.jakub@gmail.com
2-xperdek@stuba.sk
3- xsaleh@stuba.sk
4- nikolakarakas95@gmail.com
5- balgamiroslav@gmail.com

Príloha B: Export úloh

B-1. Export úloh prvého šprintu

Cyran Team

Taskboard | Backlog | Analytics | New Work Item | Column Options

Order	ID	Title	Assigned To	State	Tags
1	1	get access to faculty server	Jakub Perdek	Done	
2	6	Deploy our team page to the faculty server	abd alrahman ...	Done	
3	7	Basic layout of page	Jakub Perdek	Done	
4	8	Responsiveness and other design	Jakub Perdek	Done	
5	9	Analysis of Cyber range	Viktor Matovič	Done	
6	10	Documentation - engineer's work	Jakub Perdek	Done	
7	11	Aims and requirements of problem area	Jakub Perdek	Done	
8	13	Documentation - Project Management	Viktor Matovič	Done	
9	16	Run Kypo in local environment		Doing	assigned
10	17	Run at least one of the Kypo games		To Do	
11	18	Test attack or game in Kypo		To Do	
12	20	Provide big picture of kypo scenario	Jakub Perdek	Done	
13	21	Desing scenario on SQL injection attack	Jakub Perdek	Done	
14	22	Describe a prototype for SQL injection scenario	Jakub Perdek	Doing	
15	23	Document Scrum Retrospective Meetings		To Do	

Obrázok 1: Export úloh prvého šprintu

B-2. Export úloh druhého šprintu

Cyran Team

27. októbra - 15. novembra
0 work days remaining

Taskboard | Backlog | Analytics | New Work Item | Column Options | Sprint 2

Order	ID	Title	Assigned To	State	Tags
1	17	Run at least one of the Kypo games		To Do	
2	18	Test attack or game in Kypo		To Do	
3	23	Document Scrum Retrospective Meetings	Peter Spusta	Doing	
4	24	Whois application	Jakub Perdek	Done	
5	25	Eshop- shopping cart template	Jakub Perdek	Done	
6	26	Eshop - delivery template	Jakub Perdek	Done	
7	27	Eshop - paying methods template	Jakub Perdek	Done	
8	28	Eshop - register and login templates	abd alrahman ...	Done	
9	29	Eshop - documentation	Nikola Karakas	Done	
10	30	Whois documentation	Jakub Perdek	Done	
11	31	Backend services for testing app		Done	

Obrázok 2: Export úloh druhého šprintu

B-3. Export úloh tretieho šprintu

Cyran Team

16. novembra - 25. novembra
2 work days remaining

Taskboard | Backlog | Analytics | + New Work Item | Column Options | ··· | Sprint 3

Order	ID	Title	Assigned To	State	Tags
1	23	Document Scrum Retrospective Meetings	Peter Spusta	Doing	
2	32	Create finished order template	Jakub Perdek	Done	
3	33	Create functional shopping cart with functional services in security eshop	Jakub Perdek	Done	
4	34	Integrate frontend product management with backend in security app	Jakub Perdek	Done	
5	40	Deep documentation of eshop and revision of old one	Nikola Karakas	Done	
6	41	Provide methods for managing product in backend	Viktor Matovič	Done	
7	42	Provide backend methods for finalyze order	Viktor Matovič	Done	
8	43	Create methodics	Jakub Perdek	Done	
	36	Create code review methodics	Jakub Perdek	Done	
	37	Create communication methodics	Jakub Perdek	Done	
	38	Create version management methodics	Jakub Perdek	Done	
	39	Set format for methodics of controlling backlog	Jakub Perdek	Done	
	44	Create methodics of documentation	Jakub Perdek	Done	
9	45	Refactoring and making some eshop pages responsive	abd alrahman ...	Done	
10	46	Finalize technical and management documentation	Jakub Perdek	Done	
11	47	Vulnerable order creation as scenario on frontend	abd alrahman ...	Done	

Figure 3: Export úloh z tretieho šprintu

B-4. Export úloh štvrtého šprintu

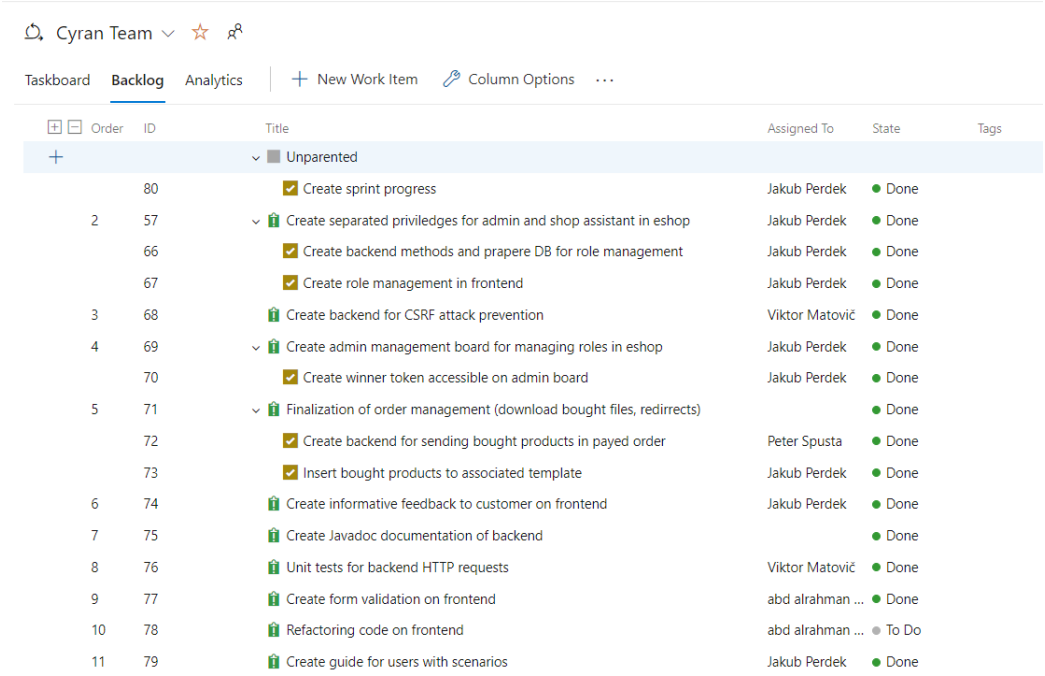
Cyran Team

Taskboard | Backlog | Analytics | + New Work Item | Column Options | ···

Order	ID	Title	Assigned To	State	Tags
		Unparented			
	65	Create sprint review and retrospective	Jakub Perdek	Done	
2	64	Run kypo parts in local environment	abd alrahman ...	Done	
3	50	Create insert product template	Jakub Perdek	Done	
4	51	Create template for managing users	Jakub Perdek	Done	
5	52	Create backend for user management	Peter Spusta	Done	
	53	Find and integrate database for user management and SQL injection attack	Jakub Perdek	Done	
6	54	Make our web page more secure using secure protocol https	abd alrahman ...	Done	
7	55	Integrate shop management functionality on backend with frontend template	Jakub Perdek	Done	
8	57	Create separated priviledges for admin and shop assistant in eshop		To Do	
9	58	Move authentication to relational SQL database	Jakub Perdek	Done	
10	59	Documentation of eshop management	Nikola Karakas	Done	
11	60	Create password regeneration and resend it to email	Jakub Perdek	Done	
	61	Provide backend for password resend to email	Jakub Perdek	Done	
	62	Provide frontend for password regeneration to email	Jakub Perdek	Done	
	63	Create email for eshop usage with configuration on backend	Jakub Perdek	Done	

Figure 4: Export úloh z tretieho šprintu

B-5. Export úloh z piateho šprintu



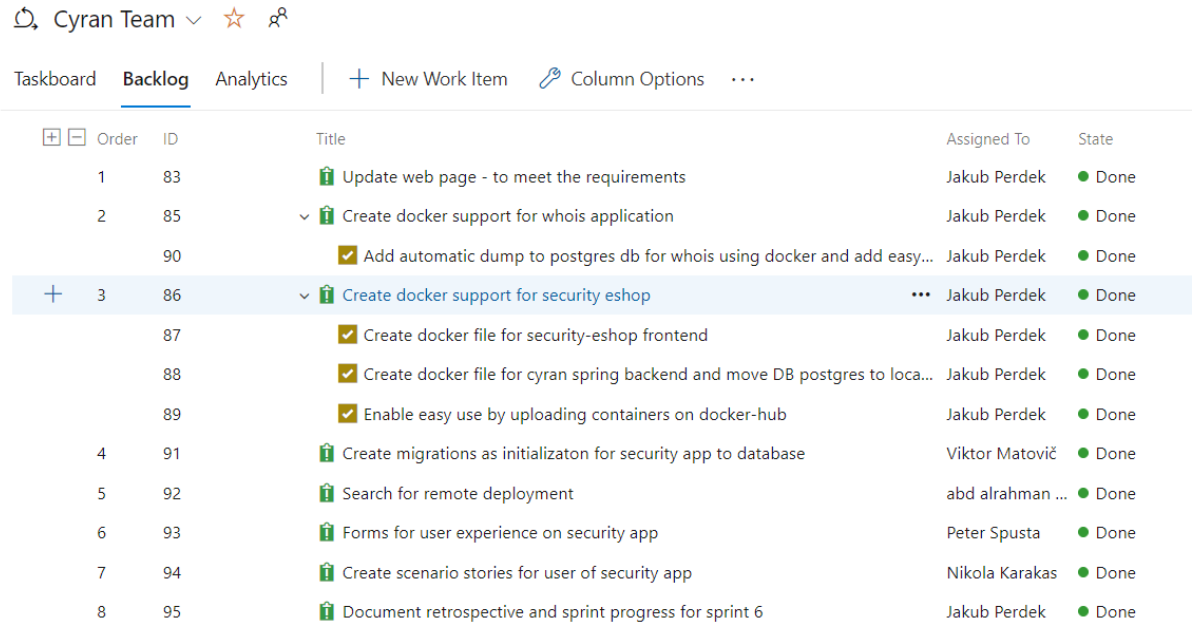
Cyran Team

Taskboard | Backlog | Analytics | + New Work Item | Column Options | ...

Order	ID	Title	Assigned To	State	Tags
		Unparented			
	80	Create sprint progress	Jakub Perdek	Done	
2	57	Create separated priviledges for admin and shop assistant in eshop	Jakub Perdek	Done	
	66	Create backend methods and prapere DB for role management	Jakub Perdek	Done	
	67	Create role management in frontend	Jakub Perdek	Done	
3	68	Create backend for CSRF attack prevention	Viktor Matovič	Done	
4	69	Create admin management board for managing roles in eshop	Jakub Perdek	Done	
	70	Create winner token accessible on admin board	Jakub Perdek	Done	
5	71	Finalization of order management (download bought files, redirrects)		Done	
	72	Create backend for sending bought products in payed order	Peter Spusta	Done	
	73	Insert bought products to associated template	Jakub Perdek	Done	
6	74	Create informative feedback to customer on frontend	Jakub Perdek	Done	
7	75	Create Javadoc documentation of backend		Done	
8	76	Unit tests for backend HTTP requests	Viktor Matovič	Done	
9	77	Create form validation on frontend	abd alrahman ...	Done	
10	78	Refactoring code on frontend	abd alrahman ...	To Do	
11	79	Create guide for users with scenarios	Jakub Perdek	Done	

Figure 5: Export úloh z piateho šprintu

B-6. Export úloh zo šiesteho šprintu



Cyran Team

Taskboard | Backlog | Analytics | + New Work Item | Column Options | ...

Order	ID	Title	Assigned To	State
1	83	Update web page - to meet the requirements	Jakub Perdek	Done
2	85	Create docker support for whois application	Jakub Perdek	Done
	90	Add automatic dump to postgres db for whois using docker and add easy...	Jakub Perdek	Done
3	86	Create docker support for security eshop	Jakub Perdek	Done
	87	Create docker file for security-eshop frontend	Jakub Perdek	Done
	88	Create docker file for cyran spring backend and move DB postgres to loca...	Jakub Perdek	Done
	89	Enable easy use by uploading containers on docker-hub	Jakub Perdek	Done
4	91	Create migrations as initializaton for security app to database	Viktor Matovič	Done
5	92	Search for remote deployment	abd alrahman ...	Done
6	93	Forms for user experience on security app	Peter Spusta	Done
7	94	Create scenario stories for user of security app	Nikola Karakas	Done
8	95	Document retrospective and sprint progress for sprint 6	Jakub Perdek	Done

Figure 6: Export úloh šiesteho šprintu

B-7. Export úloh zo siedmeho šprintu

Cyran Team

Taskboard | Backlog | Analytics | New Work Item | Column Options | ...

Order	ID	Title	Assigned To	State	Tags
1	97	Create BCrypt services and tutorial template	Jakub Perdek	Done	
	98	Create BCrypt encryptor service	Jakub Perdek	Done	
	99	Create BCrypt validator service	Jakub Perdek	Done	
	100	Present information about salt and Bcrypt as intro to BCrypt services	Jakub Perdek	Done	
2	101	Document dictionary attack and add it to user guide	Jakub Perdek	Done	
3	102	Run and test application using docker on Windows	Nikola Karakas	Done	
4	103	Test docker images on Linux	abd alrahman ...	Done	
5	104	Find email which not check device or observe docker containers for smtp ser...	Jakub Perdek	Done	
6	105	Enhance user forms to gain their experiences from using application	Peter Spusta	Done	
7	106	Create build for easier deployment on linux machines		Done	
	108	Apply the same network rules for docker images	abd alrahman ...	Done	
8	107	Create sprint retrospective and sprint progress for sprint 7	Jakub Perdek	Done	
9	109	Test scenarios on new version of docker-compose version of our app on Win...	Jakub Perdek	Done	
10	111	Regenerate website certificate	abd alrahman ...	Done	

Figure 7: Export úloh siedmeho šprintu

B-8. Export úloh z ôsmeho šprintu

Cyran Team

Taskboard | Backlog | Analytics | New Work Item | Column Options | ...

Order	ID	Title	Assigned To	State
1	118	Add whois to dockerfile and fix issues with backup insertion server failures	Jakub Perdek	Done
2	116	Add fake email server to local docker deploy	Jakub Perdek	Done
3	112	Add logging to frontend connected with sentry	Jakub Perdek	Done
	113	Create free developer account on sentry	Jakub Perdek	Done
	114	Create script to obtain logs from sentry	Jakub Perdek	Done
	115	Document sentry in engineer's work	Jakub Perdek	Done
4	117	Implement role based access control	Viktor Matovič	Doing
5	119	UX fixing and improvements	Jakub Perdek	Done
	120	Disable admin and shop assistant email for local deployment	Jakub Perdek	Done
6	121	Review docker updated containers and applications	abd alrahman ...	Done
7	122	Create session for frontend and backend	Peter Spusta	Doing
8	123	Create progress and retrospective for sprint 8		Done
9	124	Test functionality and UX for all scenarios using docker on local deployment	Nikola Karakas	Done
10	125	Create guide for local deployment using docker	Nikola Karakas	Done

Figure 8: Export úloh z ôsmeho šprintu

B-9. Export úloh z deviateho šprintu

Cyran Team

Taskboard | Backlog | Analytics | New Work Item | Column Options

Order	ID	Title	Assigned To	State
1	117	Implement role based access control	Viktor Matovič	Done
2	126	Create doc from user testing and sort information	Jakub Perdek	Done
3	127	Supply user testing - fixing fatal issues		Done
4	128	Create scenario with Advanced SQL injection	Jakub Perdek	Done
	130	Create tables for vurnerabilities and insert appropriate records	Jakub Perdek	Done
	131	Create view with DB schema	Jakub Perdek	Done
	132	Design SQL injection	Jakub Perdek	Done
5	129	Observe possibilities to change domain name for local deployed whois appli...	Jakub Perdek	Done
6	133	Create sprint progress and retrospective	Jakub Perdek	Done
7	134	Create user guide for advanced SQL injection scenario	Jakub Perdek	Done
8	135	Add local deployment parts and setup for BurpSuite to user guide	Jakub Perdek	Done
9	136	Analyse feedback from users	Peter Spusta	Done
10	138	Document user feedback from google forms	Peter Spusta	Done

Figure 9: Export úloh z deviateho šprintu

B-10. Export úloh z desiateho šprintu

Cyran Team

Taskboard | Backlog | Analytics | New Work Item | Column Options

Order	ID	Title	Assigned To	State
1	151	Unit testing on backend	Viktor Matovič	Done
2	149	Comment new code on backend	Peter Spusta	Done
3	78	Refactoring code on frontend with test of functionality	abd alrahman ...	Done
4	139	Make application more portable	Jakub Perdek	Done
	140	Move logic to separated services on frontend	Jakub Perdek	Done
	141	Create configuration for easy url/port change for app on frontend	Jakub Perdek	Done
	142	Test changes on scenarios locally - without docker	Jakub Perdek	Done
5	143	Analyze and visualize logs from sentry	Jakub Perdek	Done
6	144	Document analysis and visualization of logs from Sentry	Jakub Perdek	Done
7	145	Review changes and update images	abd alrahman ...	Done
8	146	Analyze OWASP ZAP manual attack using HUD	Nikola Karakas	Done
9	147	Comment components and services on frontend	Jakub Perdek	Done
10	148	Create sprint progress and retrospective	Jakub Perdek	Done
11	150	Analyze OWASP ZAP automatic scan/attack as scenario	Nikola Karakas	Done

Figure 10: Export úloh z desiateho šprintu

B-11. Export úloh z jedenásteho šprintu

Cyran Team

Taskboard | Backlog | Analytics | New Work Item | Column Options | ...

Order	ID	Title	Assigned To	State
1	162	Backend code refactoring	Viktor Matovič	Done
2	152	Create action presentation for CYRAN team	Jakub Perdek	Done
3	153	Create sprint review and retrospective	Jakub Perdek	Done
4	154	Migrate steal products scenario to local (postgres) DB	Jakub Perdek	Done
	155	Create base classes and repositories for product and order	Jakub Perdek	Done
	156	Create initial insert of products on backend	Jakub Perdek	Done
	157	Integrate new functionality for local deploy on backend with frontend	Jakub Perdek	Done
	158	Create the same services as in previous DB with logic which supports local...	Jakub Perdek	Done
5	159	Create JavaDoc annotations and comments for created code for steal produc...	Jakub Perdek	Done
6	160	Create final report from user testing according given template	Jakub Perdek	Done
7	161	Test application functionality localy - without docker	Jakub Perdek	Done
8	163	Update images and make code review	abd alrahman ...	Done
9	164	Refactoring code on frontend	abd alrahman ...	Doing
10	165	Test final version of application using Docker o Windows	Nikola Karakas	Done
11	166	Test other teams product	Peter Spusta	Done
12	167	Generate JavaDoc documentation from code	Peter Spusta	Done

Figure 11: Export úloh z jedenásteho šprintu

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Technická Dokumentácia

Tímový projekt

Tím č. 19

Vypracoval: Jakub Perdek
Vedúci projektu: Ing. Pavol Helebrandt Phd.

Obsah

1 Požiadavky riešenia 3

1.1 Scenáre 3

1.2 Nasadenie 3

1.3 Nefunkcionálne požiadavky 4

2 Big Picture 5

2.1 Úvod 5

2.2 Ciele 5

2.3 Ohraničenia 6

2.4 Globálne ciele na zimný semester 6

2.5 Globálne ciele na letný semester 7

2.6 Celkový pohľad na systém 8

3 Technická dokumentácia 9

3.1 Whois aplikácia pre vyhľadanie domény 9

Vyhľadanie domény 10

Informácie o vyhľadanej doméne 10

BCrypt a overenie jeho vlastností 13

Schéma ku databáze 15

Zhodnotenie k whois aplikácii 16

3.2 Cieľová stránka e-shopu 17

Používateľské rozhranie a dizajn stránky 17

Domovská stránka 17

Prihlásenie a registrácia 18

Nákupný košík 19

Informácie o doručení 20

Informácie o platbe 21

Správa rolí používateľov 25

Server a riadiaca časť systému 26

Databáza 27

Databázový model 28

3.3 Scenáre s použitím e-shopu 30

Prelamovanie slabých hesiel – slovníkový útok 30

Ukradnutie produktu odoslaním falošnej informácie 30

Ukradnutie produktu prístupom do priečinka 30

SQL injekcia pre zmenu emailovej adresy admina 30

SQL injekcia pre získanie informácií z whois 32

3.4 Logovanie v aplikácii 33

Analýza dát zo Sentry 34

Analýza dát z Google forms 40

3.5 Posudky na prototyp tímu č. 19 ____ 43
Posudok na základe používateľského testovania ____ 43
Posudky jednotlivých testerov z bezpečnostného semináru ____ 44

1 Požiadavky riešenia

Podľa zadania a následných konzultácií s product ownerom boli identifikované nasledovné požiadavky riešenia:

- Navrhnúť simulačné prostredie spolu s vybranými scenármi pre testovanie kybernetickej ochrany
- Použiť platformu (simulačného prostredia) pre realizáciu tohto prostredia (Odporúčanie použiť KYPO)
- Tvorba simulačného prostredia na jednom fyzickom PC pomocou viacerých virtuálnych strojov

1.1 Scenáre

- Otestovať už existujúce scenáre
- Navrhnúť 2-3 vlastné scenáre vhodné do výučby na FIIT
- Implementovať navrhnuté prostredie a scenáre na prostriedkoch FIIT
- Otestovať navrhnuté prostredie a scenáre na prostriedkoch FIIT
- Scenáre by mali slúžiť na podporu a zlepšenie výučby predmetov informačnej a sieťovej bezpečnosti.
- Identifikácia vhodných typov scenárov pre zapracovanie do problematiky
- Identifikácia vhodných typov problémov pre zapracovanie do scenárov
- Scenáre by mali zaujať hráča
- Zakomponovanie špeciálnych vlastností virtuálnych systémov s dôrazom na ich vplyv na existujúce a aj nové zraniteľnosti a detekcie (resp. prevencie prienikov zneužívajúcich tieto zraniteľnosti)
- Obsahom scenárov by malo byť zabezpečenie rôznych systémov ako aj rôzne prieniky do nich

1.2 Nasadenie

- Nasadenie výsledného riešenia pomocou virtuálnych strojov
- Nasadenie simulačného prostredia v prostredí OpenStack

- Nasadenie výsledného riešenia s minimalizáciou manuálnych úkonov a zásahov zo strany pedagóga

1.3 Nefunkcionálne požiadavky

- Riešenie by malo byť dynamicky škálovateľné podľa aktuálnych potrieb a dostupných prostriedkov

2 Big Picture

2.1 Úvod

Cyran projekt je zameraný na možnosť zlepšenia a testovania svojich schopností v simulovanej realite kyberpriestoru. Účastníci riešia rôzne úlohy a snažia sa odvrátiť útoky alebo sa infiltrovať do počítača cudzej osoby, prípadne podniknúť inú formu útoku. Cieľom je nájsť potencionálnu zraniteľnosť systému pre tím, ktorý sa obraňuje, prípadne získať informáciu v najčastejšie v podobe textového reťazca od brániaceho sa tímu.

2.2 Ciele

V rámci projektu je naším hlavným cieľom zostrojiť aplikáciu využívajúcu platformu KYPO, ktorá by používateľom umožnila vzdelať a súperiť v oblasti kybernetickej ochrany formou vytvorených hier. Každá hra bude založená na originálnom scenári pre otestovanie a prípadne aj naučenie používateľa rôznym technikám, na ktoré bude orientovaný. Ďalšími vedľajšími cieľmi, ktoré poslúžia pre realizáciu hlavného cieľa alebo napĺňajú novú funkcionalitu, ktorá podporuje požiadavky riešenia sú:

- Použitie platformy KYPO pri realizácii aplikácie ako aktualizovaného prostredia
- Analýza problematiky kybernetickej bezpečnosti
- Návrh scenárov zameraných na špeciálne situácie akými sú chyby v systéme alebo oboznámenie sa z rôznymi nástrojmi
    - Tieto scenáre budú mať edukatívny charakter
    - Nápovedy by mali slúžiť pre ponorenie používateľa do problému
    - Herný systém by mal identifikovať schopnosti a úroveň hráča pre lepší herný zážitok a poučenie z hry
    - Akcie používateľa by sa mali zaznamenávať pre identifikáciu rôznych návykov
    - Overenie na základe dotazníkov a rozhovorov by malo slúžiť na hľadanie vhodného scenáru pre konkrétnu problematiku

    - Analýza novo nájdených zraniteľností
    - Automatizácia procesov vyhodnocovania priebehu hry
        - Rozhodnutie ktoré schváli koordinátor
        - Automatické rozhodovanie
- Tvorba docker image-ov pre jednoduché nasadenie aplikácie
- Nasadenie aplikácii na OpenStack ako želaného miesta
- Dôraz pri návrhu a implementácii na objektové prístupy, architektúru s podpory interoperability a rozšíriteľnosti riešenia

2.3 Ohraničenia

Ohraničenia, ktoré náš systém bude mať budú počet realizovaných scenárov a overenia s konkrétnymi študentmi pre dĺžku trvania projektu.

2.4 Globálne ciele na zimný semester

Pre nedostupnosť KYPO platformy sme realizovali webovú aplikáciu ako samostatný celok fungujúci aj mimo platformy KYPO. Po získaní prístupu k platforme aplikáciu hodláme nasadiť na jeden z uzlov do OpenStacku.

Globálne ciele na zimný semester sú

- Použitie platformy KYPO pri realizácii aplikácie ako aktualizovaného prostredia
- Analýza problematiky kybernetickej bezpečnosti
- Návrh scenárov zameraných na špeciálne situácie akými sú chyby v systéme alebo oboznámenie sa z rôznymi nástrojmi
    - Tieto scenáre budú mať edukatívny charakter
    - Nápovedy by mali slúžiť pre ponorenie používateľa do problému
    - Herný systém by mal identifikovať schopnosti a úroveň hráča pre lepší herný zážitok a poučenie z hry

        - Akcie používateľa by sa mali zaznamenávať pre identifikáciu rôznych návykov
        - Overenie na základe dotazníkov a rozhovorov by malo slúžiť na hľadanie vhodného scenáru pre konkrétnu problematiku
        - Analýza novo nájdených zraniteľností
        - Automatizácia procesov vyhodnocovania priebehu hry
            - Rozhodnutie ktoré schváli koordinátor
            - Automatické rozhodovanie
- Tvorba docker image-ov pre jednoduché nasadenie aplikácie
- Dôraz pri návrhu a implementácii na objektové prístupy, architektúru s podpory interoperability a rozšíriteľnosti riešenia

2.5 Globálne ciele na letný semester

Globálne ciele na letný semester sú

- Použitie platformy KYPO pri realizácii aplikácie ako aktualizovaného prostredia
- Kontajnerizácia a nasadenie vytvorených aplikácií v minulom semestri
- Zlepšenie vytvorenej webovej aplikácie
    - Vylepšenie dizajnu, hrateľnosti, realizácie konfigurovateľných chýb v aplikácii
- Návrh scenárov zameraných na špeciálne situácie akými sú chyby v systéme alebo oboznámenie sa z rôznymi nástrojmi
    - Tieto scenáre budú mať edukatívny charakter
    - Nápovedy by mali slúžiť pre ponorenie používateľa do problému
    - Herný systém by mal identifikovať schopnosti a úroveň hráča pre lepší herný zážitok a poučenie z hry
    - Akcie používateľa by sa mali zaznamenávať pre identifikáciu rôznych návykov
    - Overenie na základe dotazníkov a rozhovorov by malo slúžiť na hľadanie vhodného scenáru pre konkrétnu problematiku
    - Analýza novo nájdených zraniteľností

- Automatizácia procesov vyhodnocovania priebehu hry
    - Rozhodnutie ktoré schváli koordinátor
    - Automatické rozhodovanie
- Tvorba docker image-ov pre jednoduché nasadenie aplikácie
- Dôraz pri návrhu a implementácii na objektové prístupy, architektúru s podpory interoperability a rozšíriteľnosti riešenia

2.6 Celkový pohľad na systém

Diagram nasadenia

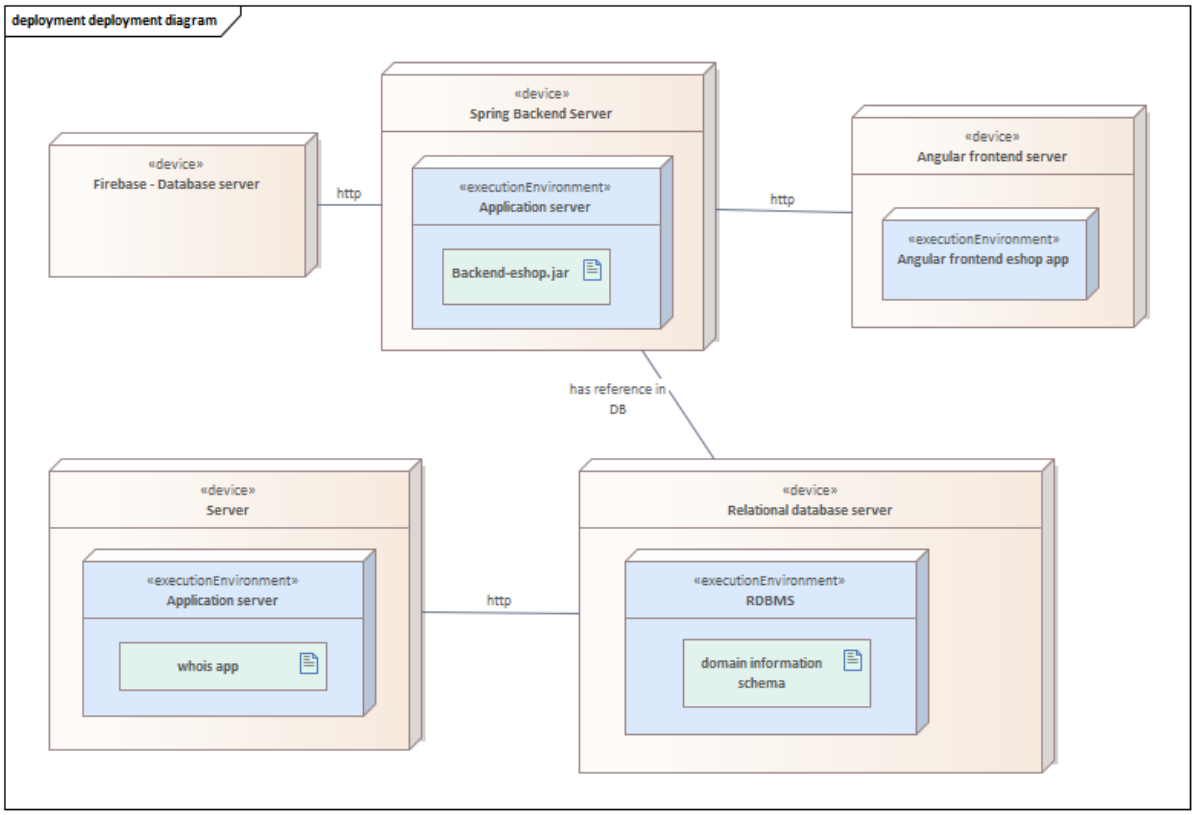


Diagram 1: Fyzické rozvrhnutie systému

3 Technická dokumentácia

K aplikáciám bola vytvorená ich technická dokumentácia. Uvádzame tu dokumentáciu k backendu a frontendu eshopu. Zdokumentovaná je aj Whois aplikácia. V dokumentácii uvádzame používateľské rozhrania, použité služby a funkcionalitu konkrétnej aplikácie.

3.1 Whois aplikácia pre vyhľadanie domény

Aplikácia slúži na vyhľadávanie informácií v databáze o konkrétnej doméne. Databáza je získaná z internetu a bude doplnená o ďalšie domény zahrnuté v scenároch. Dodatočne k informáciám o konkrétnej doméne môžu byť pridané aj potencionálne hrozby. Reprezentuje nástroj, na základe ktorého môže používateľ vyhľadať informácie o nájdených hrozbách a použiť ich pre potencionálny útok alebo obranu konkrétnej aplikácie. Zároveň sa predpokladá, že získa zručnosti pri práci s takýmto nástrojom. Navrhnutý dizajn má približovať meniacu sa sieť internetových prepojení.

Obrázok 1: Okno vyhľadávača

Obrázok 2: Navigácia vyhľadávača

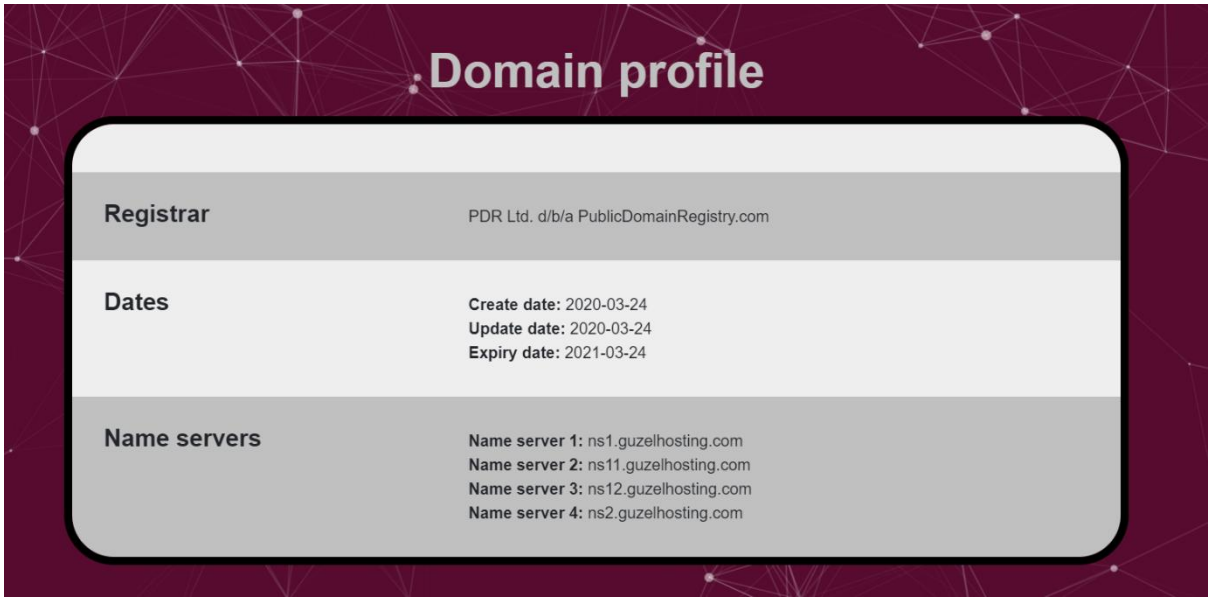
Vyhľadanie domény

Používateľ po načítaní stránky vloží názov domény do okna v strede obrazovky a stlačí tlačidlo Search. Formulár je zobrazený na Obrázku 1. Reťazec je hľadaný v uprostred doménových mien. Výsledok môže obsahovať tento reťazec kdekoľvek v názve domény. Vrátený je len jeden výsledok, preto by dopyt mal byť čo najpresnejší. Hlavnú stránku tvorí lista v hlavičke obsahujúce logo vľavo a menu tlačidlá na vpravo. Lišta je zobrazená na Obrázku 2. Päta stránky informuje o možnostiach tohto webu. Na jej samom spodku sa nachádzajú informácie o tvorcoch stránky. Päta je zobrazená na Obrázku 3.

Obrázok 3: Päta vyhľadávača

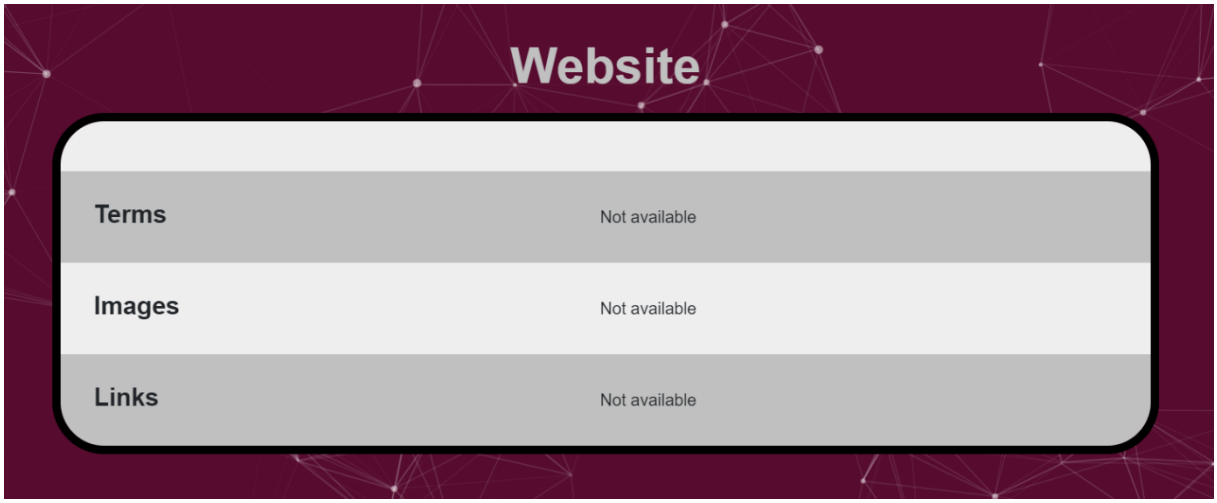
Informácie o vyhľadanej doméne

Pokiaľ bolo vyhľadanie úspešné zobrazia sa dostupné informácie o konkrétnej doméne. Zahŕňajú informácie o registračnej doméne, dátumoch vzniku, úpravy a doby platnosti. V základnom popise sú uvedené aj menné servery. Doménový profil je zobrazený na Obrázku 4.



Obrázok 4: Profil domény

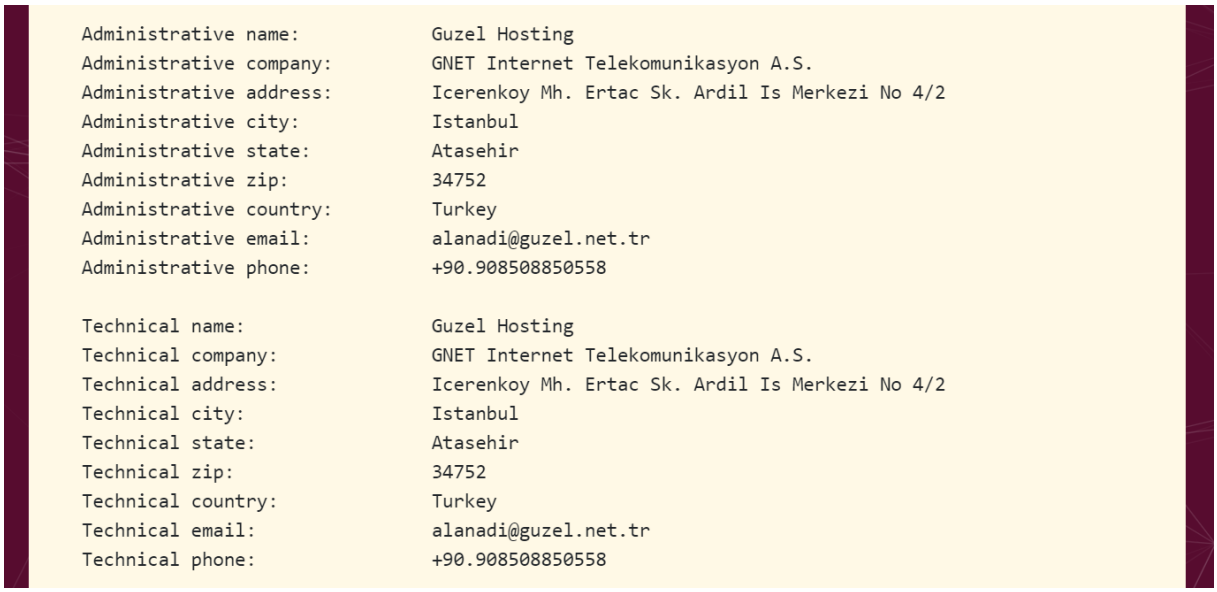
Základné zozbierané informácie o stránke je možné uviesť a neskôr získať z časti pre informácie o stránke. Tvorí ju základná štatistika o výskyte termov, obrázkov a odkazov na stránke. V našom riešení tieto informácie neuvádzame ani nezbierame, ale v budúcnosti môže byť riešenie rozšírené o preliezač webu, ktorý získa tieto informácie. Táto časť je zobrazená na Obrázku 5.



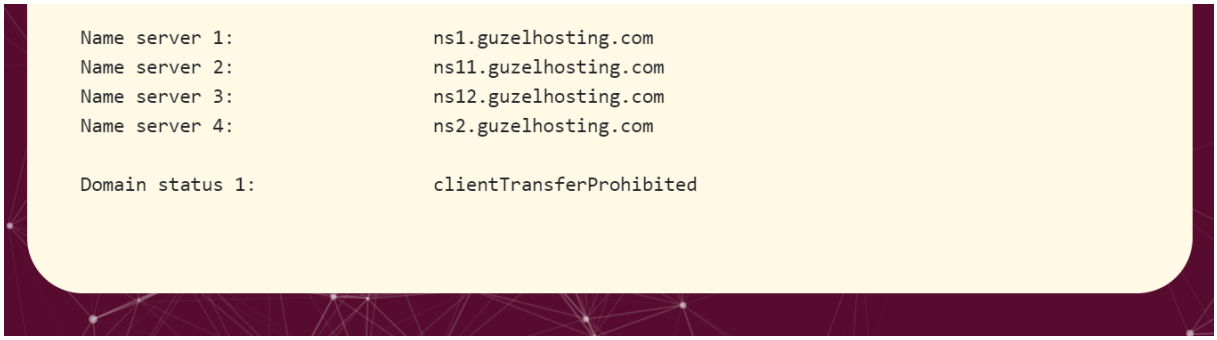
Obrázok 5: Informácie o stránke

Podrobnejšie informácie sme vložili do samostatného okna. Zobrazujeme tu všetky dostupné informácie z databázy pre konkrétnu doménu. Obsahom sú mailové adresy, telefónne čísla, adresy a ďalšie informácie o administratíve, platbách, prípadne o technickom stave pokiaľ sú k dispozícii. Pokiaľ niektorá informácia nebola nájdená alebo chýba v databáze, potom sa vo výslednom výpise nezobrazí. Ukážky výpisu pre doménu cukurovabims.com sú zobrazené na Obrázkoch 6 až 8.

Obrázok 6: Podrobnejšie informácie

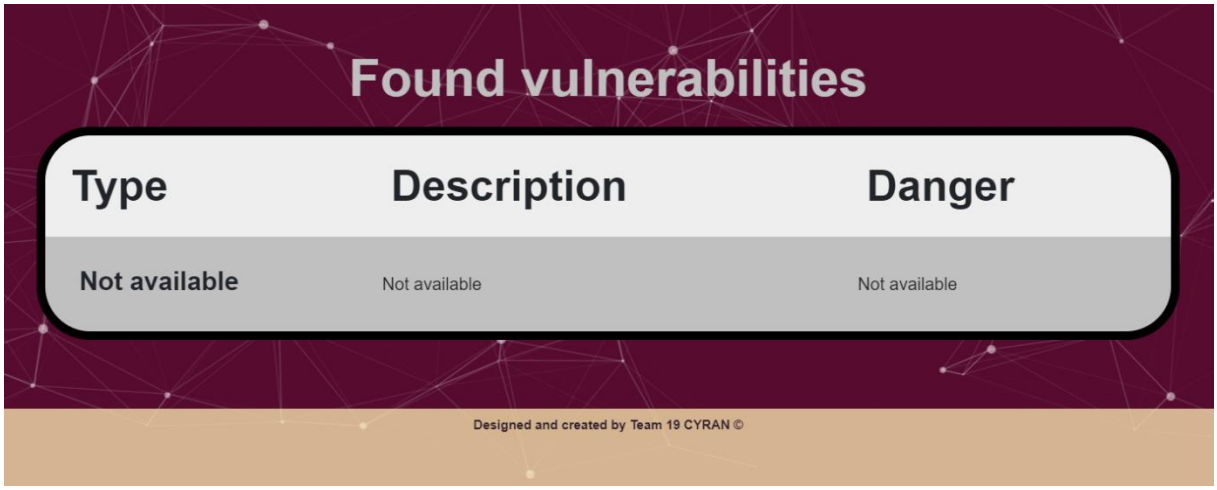


Obrázok 7: Podrobnejšie informácie pokračovanie 1



Obrázok 8: Podrobnejšie informácie pokračovanie 2

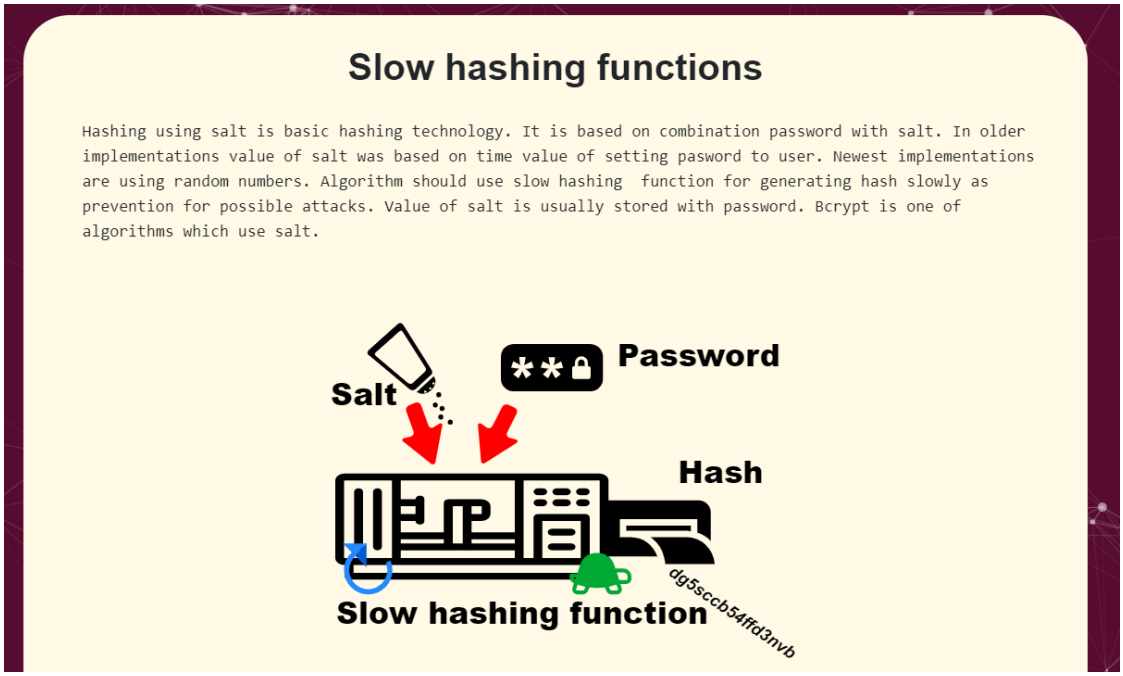
Podstatným informačným obsahom pre penetračného testera alebo útočníka sú informácie o zraniteľnostiach. Vytvorili sme pre ne samostatnú tabuľku. V prípade scenára je možné poskytnúť používateľovi informáciu o zraniteľnostiach domény, na základe čoho by mal byť schopný dohľadať doplňujúce informácie a urobiť vhodnú akciu. Databáza whois ale informácie o zraniteľnostiach neobsahuje.



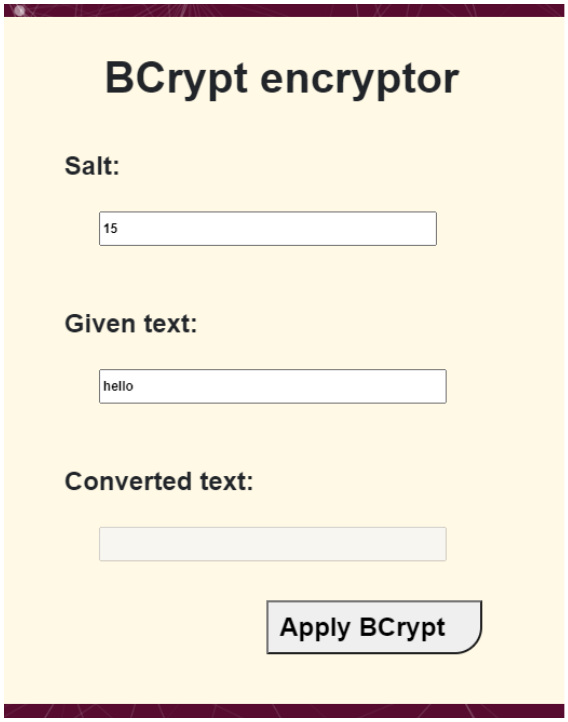
Obrázok 9: Nájdené hrozby

BCrypt a overenie jeho vlastností

Doplnok umožňujúci dozvedieť sa niečo o fungovaní BCrypt algoritmu a reálne si vyskúšať vygenerovať a následne overiť heslo by malo byť pre používateľov, ktorý sa s ním nestretnú praktické. Malo by to byť hlavne kvôli dlhšej dobe čakania na výsledok pri vyššej hodnote soli, keďže sa jedná o pomalú hashovú funkciu. Funkcionalita vznikla ako podporná časť semi-automatického prelamovania hesiel v security eshope. Útočník si službou môže pomôcť pri overovaní zhody hashu s hádanými reťazcami. Úvodný text môžete vidieť na obrázku 10. Formuláre pre šifrovanie a dešifrovanie sú zobrazené na obrázku 11.



Obrázok 10: Pomalé hashové funkcie



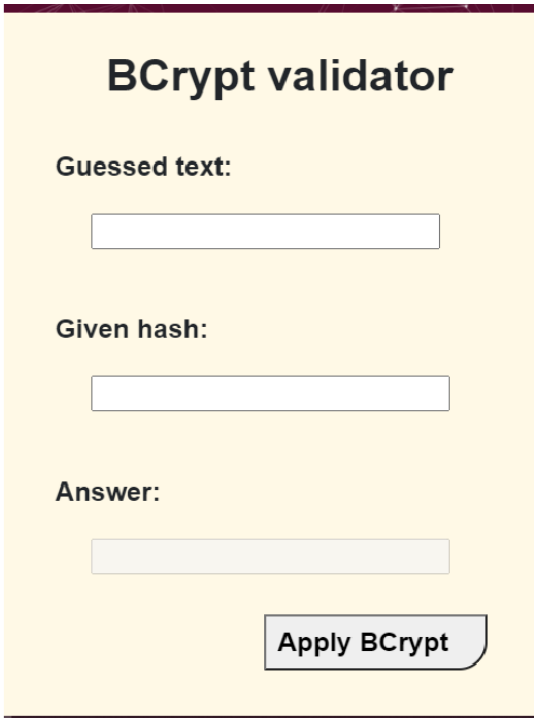
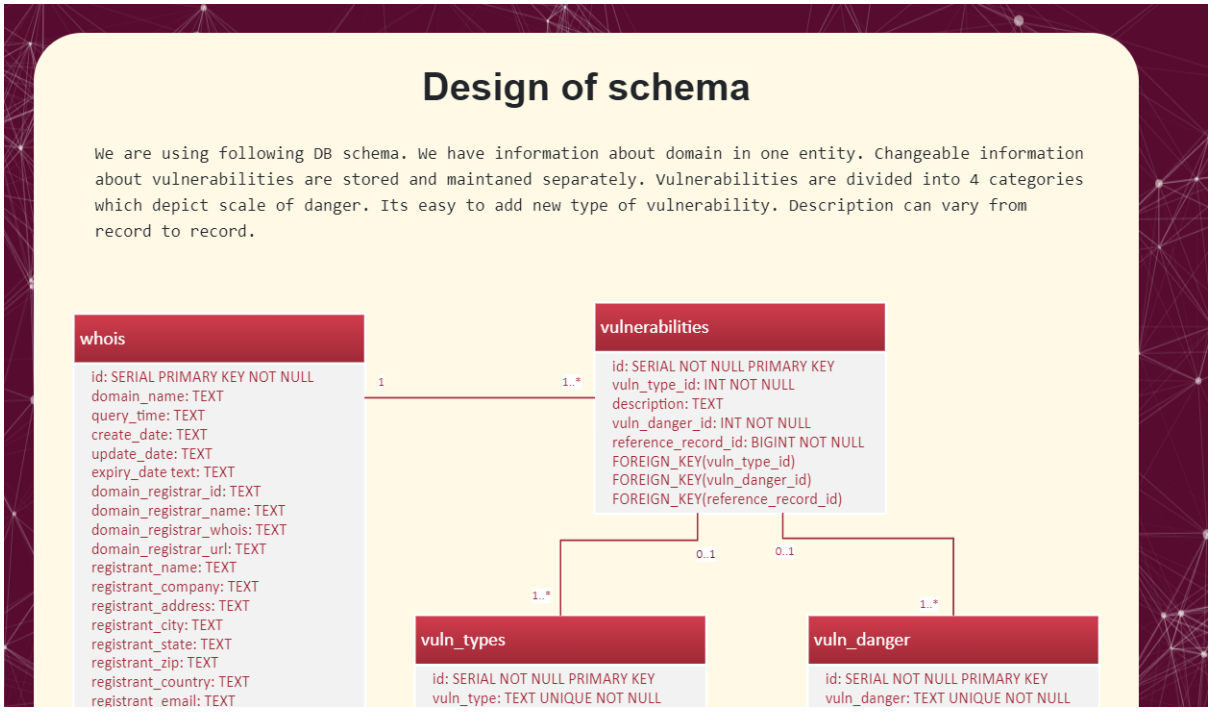


Schéma ku databáze

Predpokladáme, že whois aplikácia chce prezentovať svoju schému ukladania dát. Po doplnení častí zo zraniteľnosťami obsahuje 4 tabuľky. Pôvodná tabuľka obsahuje podstatné informácie o doméne. Ostatné slúžia na popis zraniteľností, ktoré sa môžu často meniť ako aj pribúdať nové typy hrozieb. Zároveň sme chceli aj kategorizovať úroveň nebezpečnosti hrozieb. Schéma okrem toho má používateľovi slúžiť na možnosť uplatniť pokročilú SQL injekciu, keďže pokročilé vyhľadanie whois aplikácia zatiaľ neobsahuje a špecifické potreby používateľa rovnako nie sú zahrnuté. Pri budúcom zlepšovaní aplikácie by funkcionalita mohla byť prístupná menej skúseným používateľom a pre tých skúsenejších ponechaná možnosť SQL injekcie pri získavaní dát. Obrazovku so zobrazenou schémou môžete vidieť na obrázku 12.



Obrázok 11: Schéma ku databáze

Zhodnotenie k whois aplikácii

Vyhľadanie a zber informácií je podstatnou časťou penetračného testovania. Vytvorili sme preto aplikáciu pre vyhľadanie informácií o konkrétnej doméne. V rámci bezpečnostných scenárov by do databázy ktorú aplikácia využíva mali byť pridané informácie o doménach bežiacich v sandboxe, respektíve o webových objektoch bezpečnostných scenárov. Predpokladáme, že bežne dostupné whois servery tieto informácie nebudú mať, a to hlavne z dôvodu dostupnosti nami pridaných webových lokalít. Pridanie vlastných zraniteľností do informácií o doméne by malo vylepšiť hrateľnosť scenárov a podnietiť používateľa vyhľadať si informácie o nich. Rovnako pri vypnutí niektorých zraniteľností je zhotovené riešenie flexibilné, keďže je potrebné len zmeniť hodnotu uloženú v databáze.

3.2 Cieľová stránka e-shopu

Tento dokument popisuje základné komponenty webovej stránky, ktoré budú súčasťou scenára. Táto webová stránka bude cieľom kybernetických útokov.
Webová stránka elektronického obchodu je navrhnutá ako klasický webový obchod, kde má používateľ môže:

- prihlásiť sa
- registrovať sa
- vyhľadať produkty
- pridať produkty do košíka
- vybrať dodávateľa a miesto dodania
- vybrať spôsob platby
- zaplatiť online

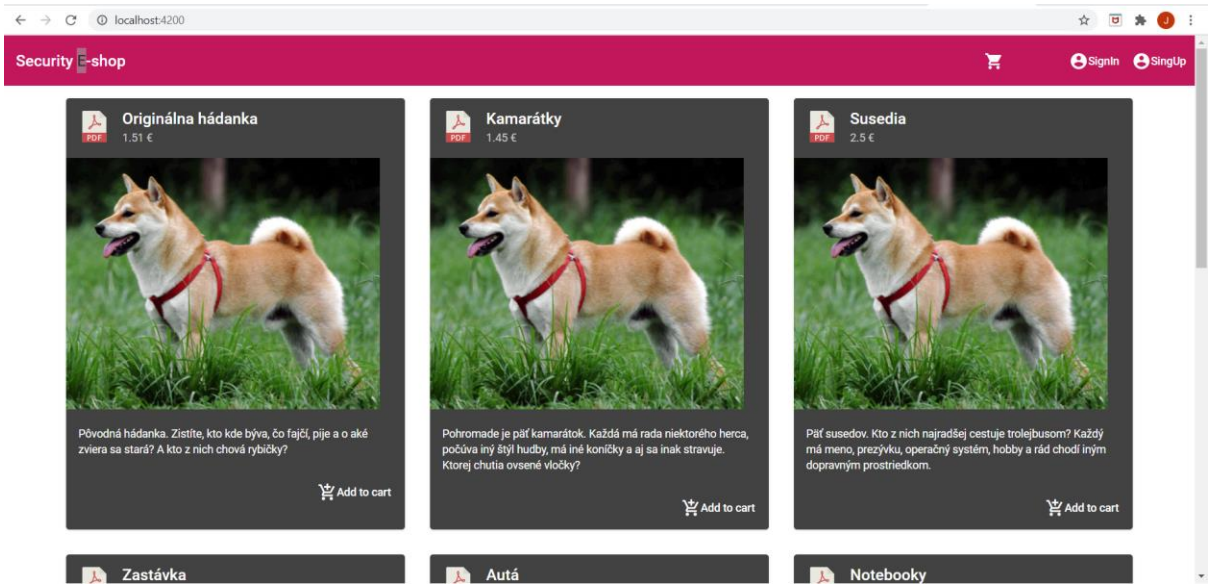
Stránka je koncipovaná ako fiktívny cieľ s cieľom využiť jej nedostatky a uskutočniť rôzne typy kybernetických útokov. Lokalita ako celok bude veľmi dynamická, aby sa v neskorších scenároch mohla technológia webu prispôsobiť povahe útoku, napríklad zmenám v databáze alebo funkčnosti alebo backendu samotnému.

Používateľské rozhranie a dizajn stránky

Ako technológia pre frontend bol použitý Angulár. Webové sídlo sa skladá z 3 hlavných stránok. Prvou stránkou je domovská stránka, ktorá je hlavnou prezentáciou webu elektronického obchodu.

Domovská stránka

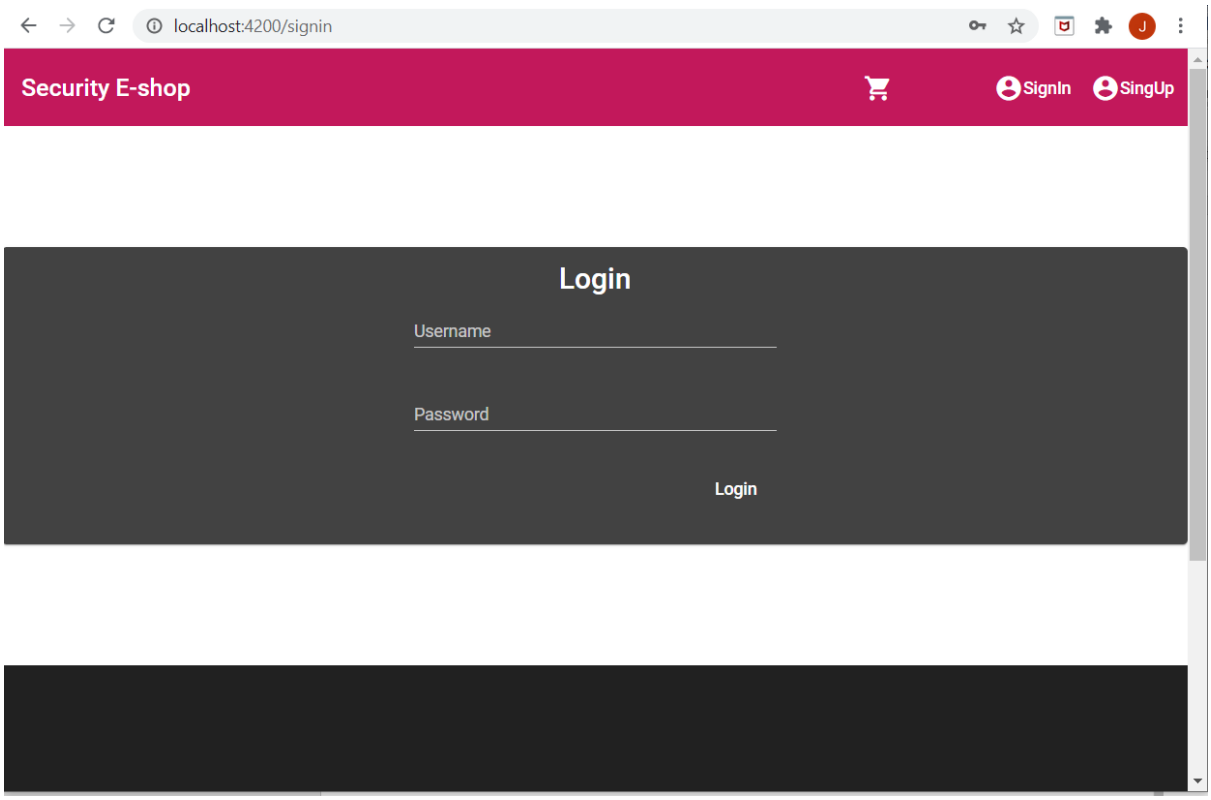
V zobrazení domovskej stránky môže používateľ prehľadávať produkty bez predchádzajúceho prihlásenia alebo registrácie. Odtiaľ si môže zvoliť, či prejde registráciou / prihlásením, alebo podrobnejším vyhľadávaním produktu.



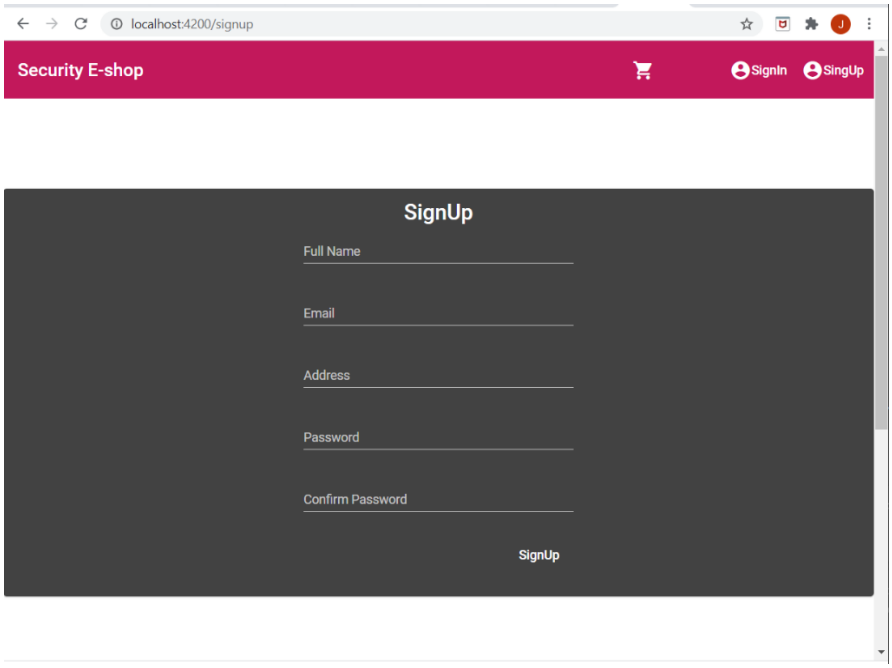
Obrázok 12 Zobrazenie domovskej stránky

Prihlásenie a registrácia

Z domovskej stránky sa môže používateľ prejsť na stránku s prihlasovaním alebo registráciou.



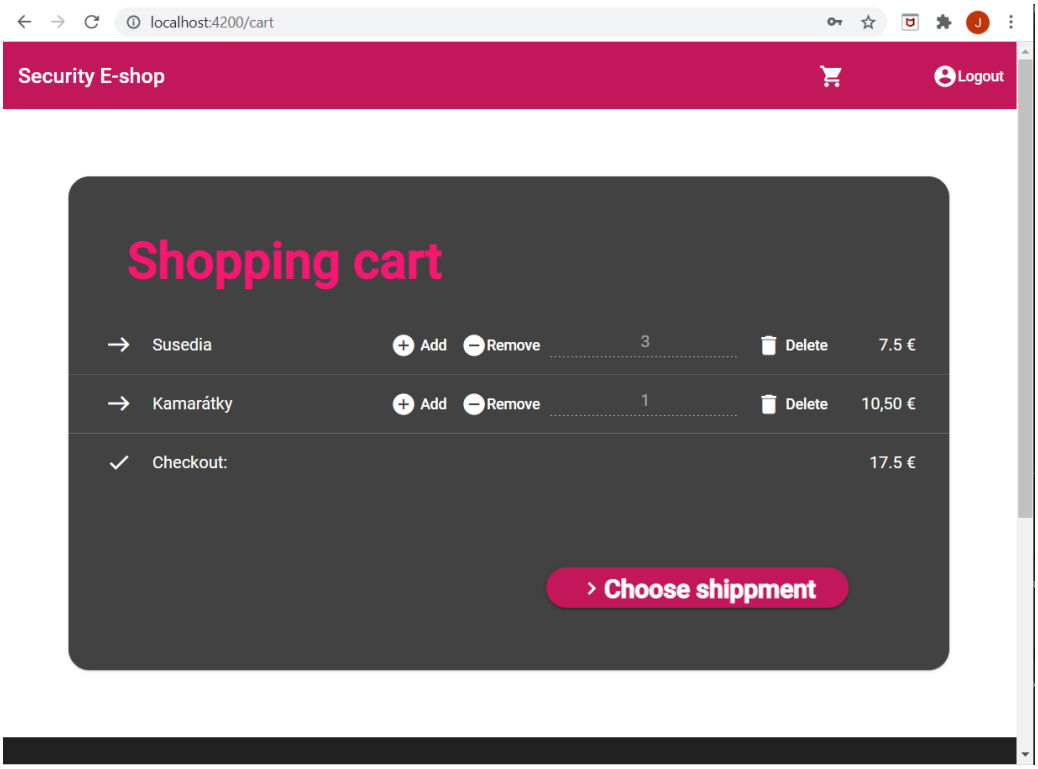
Obrázok 13: Formulár na prihlásenie



Obrázok 14 Formulár na registráciu

Nákupný košík

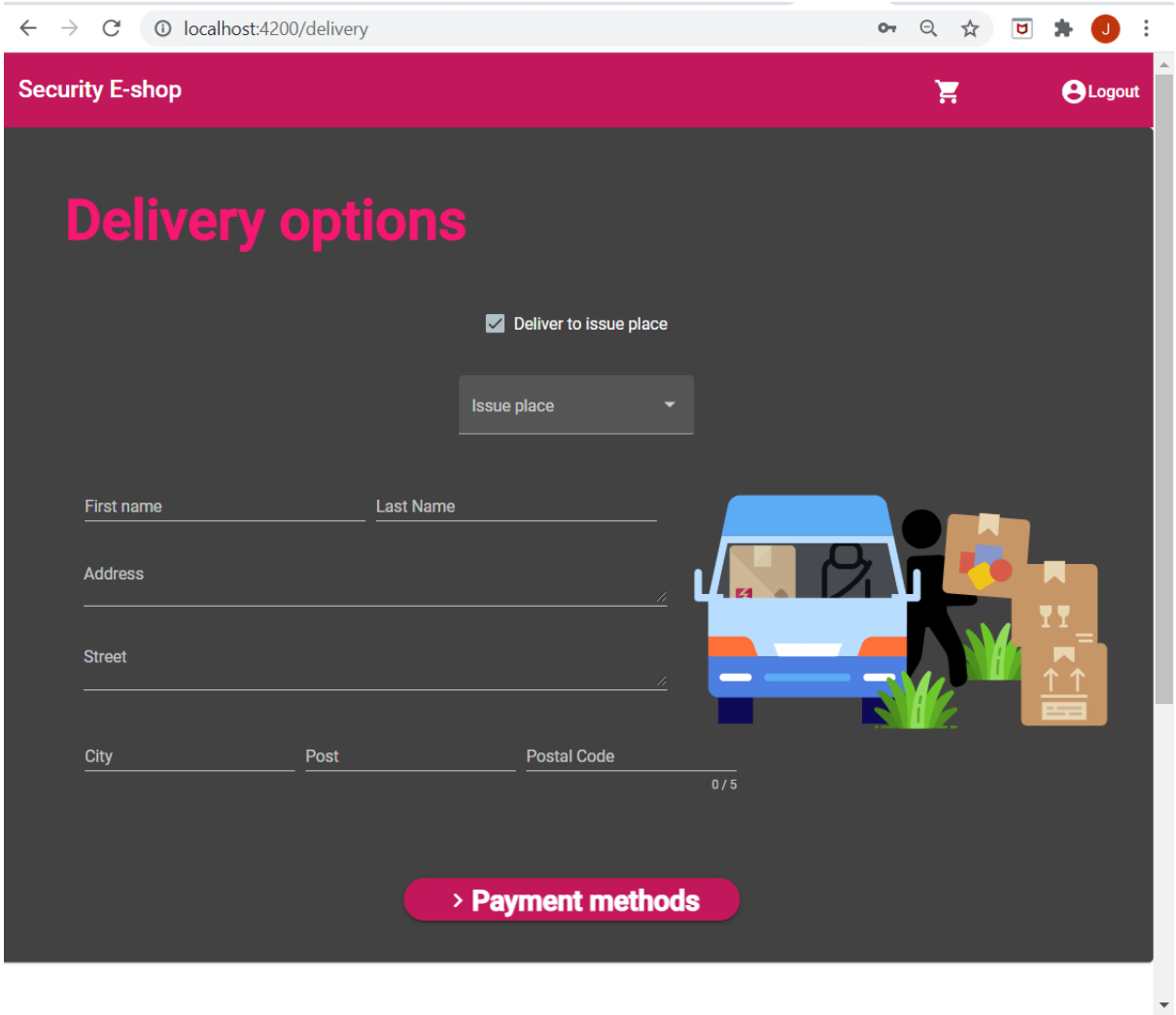
Zobrazenie nákupu začína presmerovaním na zobrazenie nákupného košíka. Tu si používateľ vyberie požadované množstvo vybraných produktov, a prechádza na výber spôsobu doručenia.



Obrázok 15 Zobrazenie nákupného košíka

Informácie o doručení

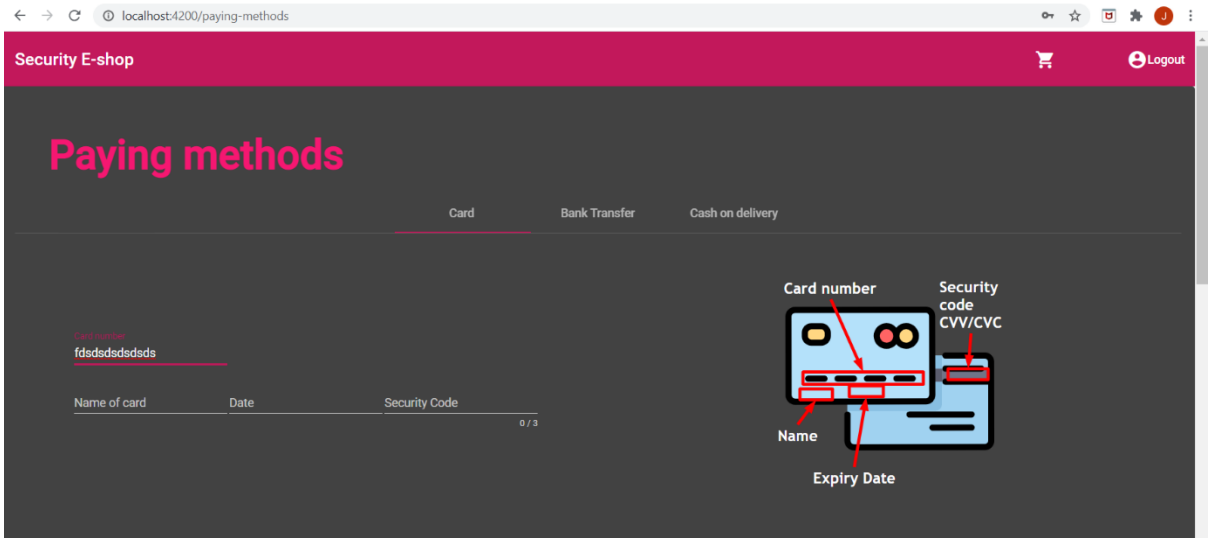
Do formuláru na Obrázku 17 používateľ vloží informácií o príjemcovi objednávky.



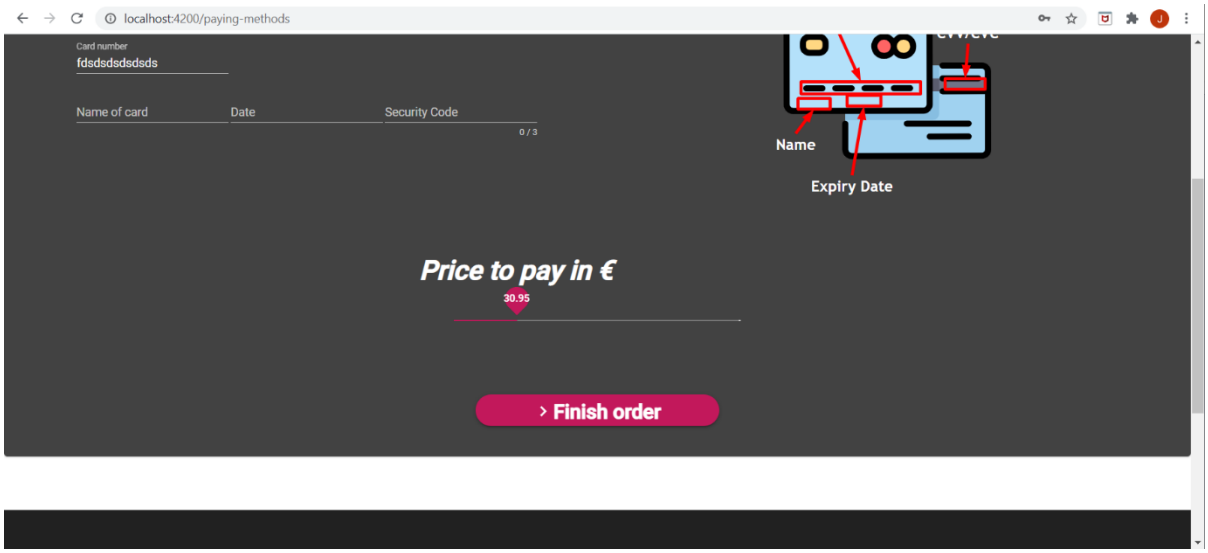
Obrázok 16: Formulár na zadanie informácií o príjemcovi objednávky

Informácie o platbe

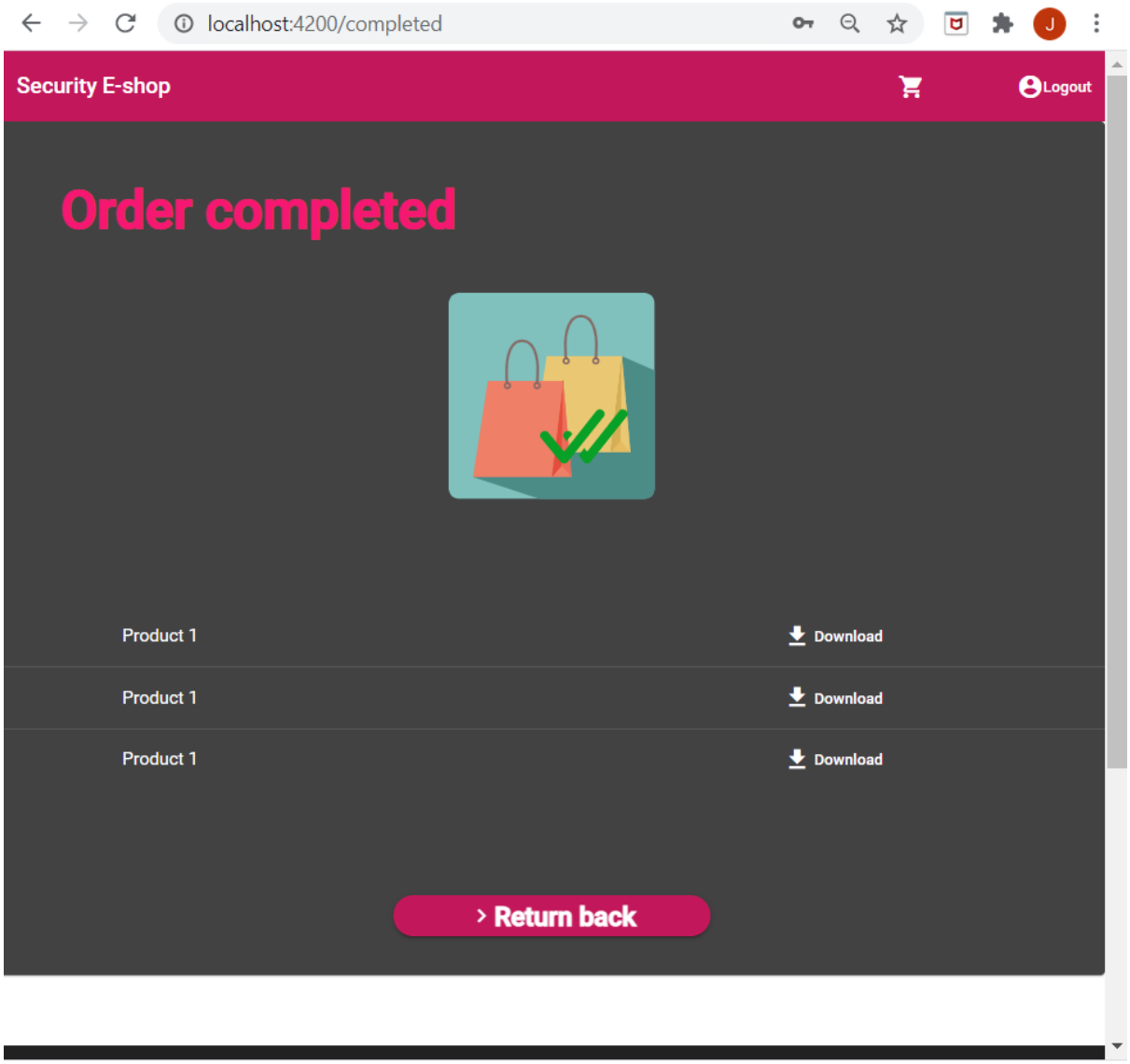
Proces elektronického nákupu končí výberom spôsobu platby a zadaním platobných údajov. Môže si vybrať medzi platbou kartou online, bankovým prevodom alebo poslaním na dobierku. Pri platbe kartou online sa používateľovi zobrazí formulár pre zadanie informácií o platobnej karte Následne klikne na tlačidlo pre dokončenie objednávky, a zobrazí sa mu správa o úspešnej alebo neúspešnej transakcii.



Obrázok 17 Formulár na zadanie informácií o platbe



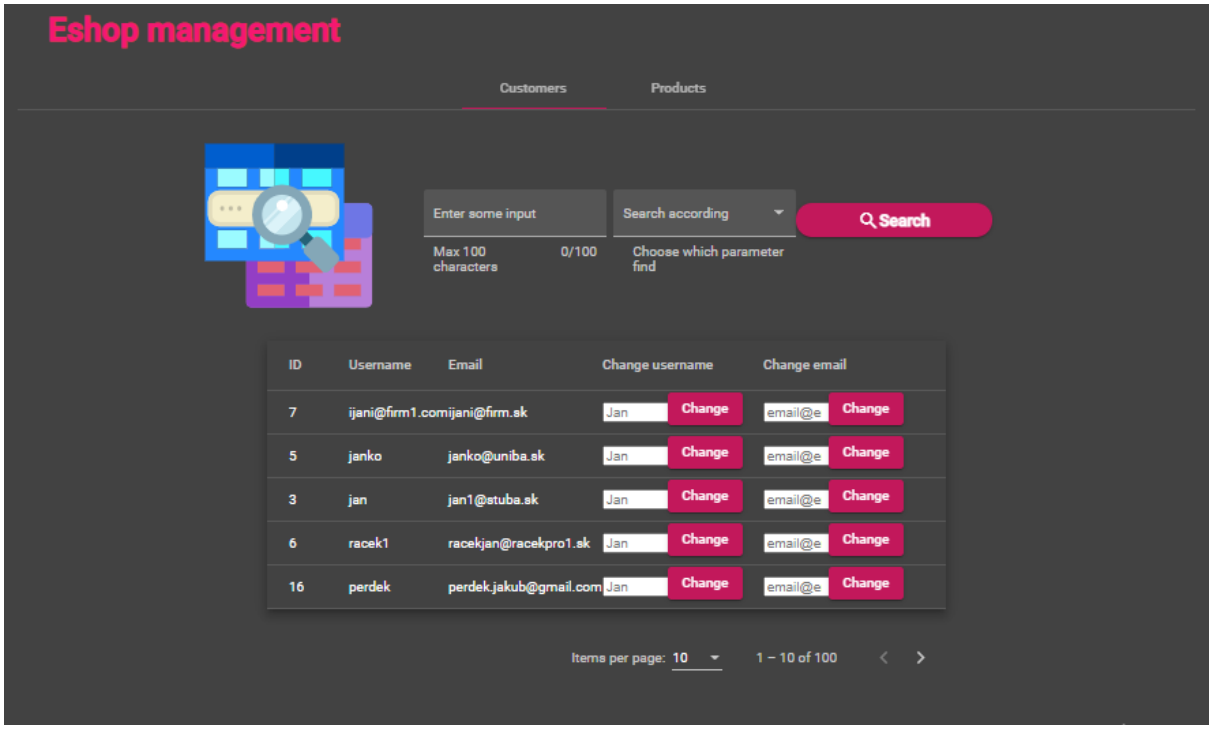
Obrázok 18: Možnosť podporiť e-shop



Obrázok 19: Možnosť stiahnuť zakúpený tovar

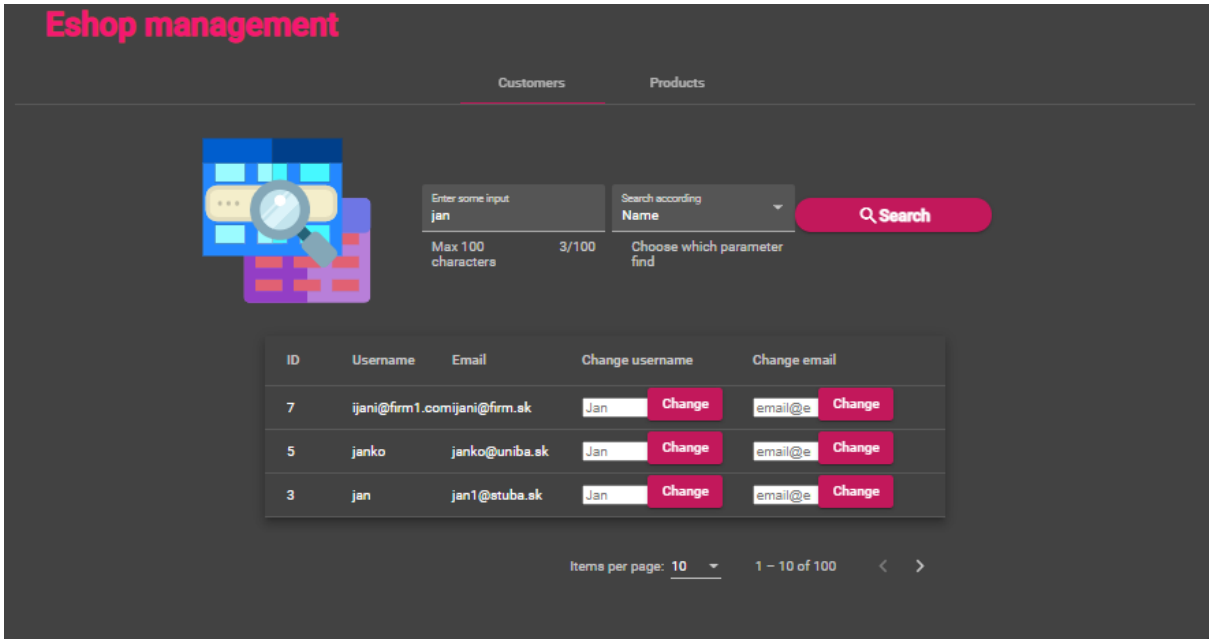
Správa používateľov a produktov

Používateľ s oprávneniami pracovníka obchodu bude mať oprávnenie nad ostatnými používateľmi a produktmi. V tomto rozhraní má možnosti upravovať zákaznícke účty. Vyhľadávať môže podľa dvoch atribútov: meno a e-mail. Po kliknutí na tlačidlo Search (hľadať) budú vygenerovaní všetci používatelia, ktorí vyhovujú dopytu.



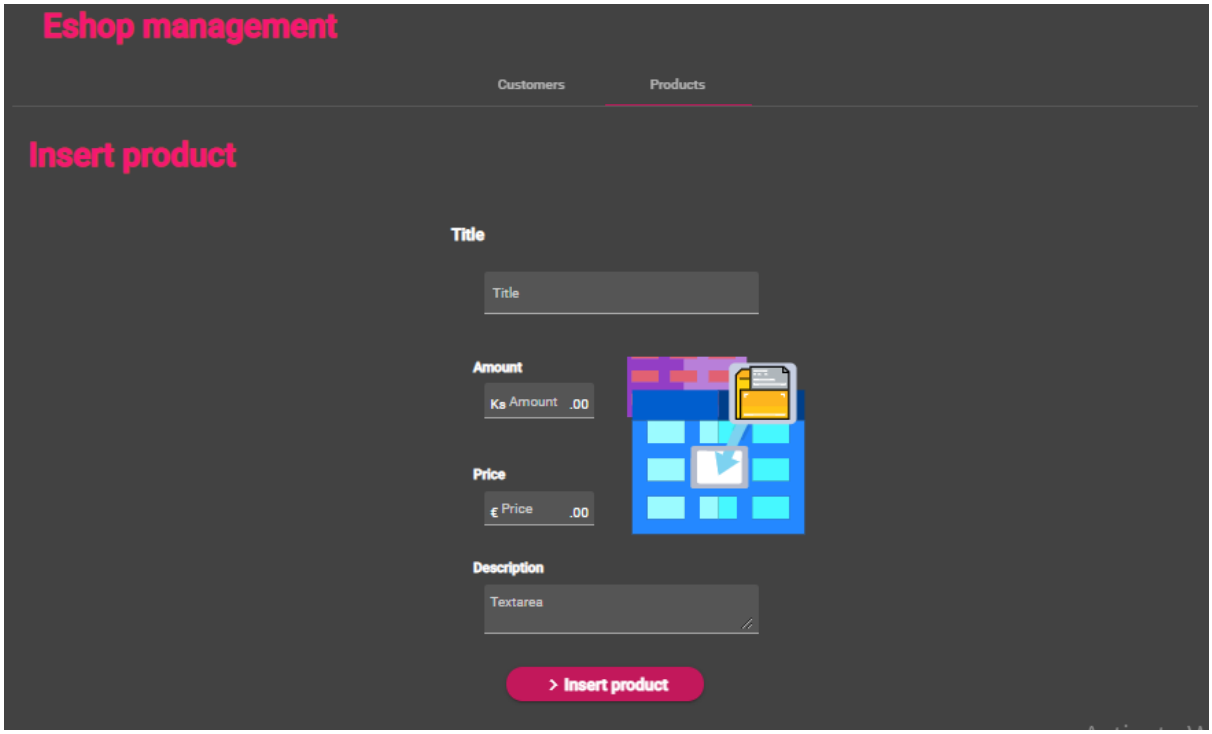
Obrázok 20: Správa používateľov

V ďalších krokoch môže správca zmeniť ich mená alebo e-mailové adresy. Kliknutím na tlačidlo Change (zmeniť) vykonáte a potvrdíte, že sa vykonáva.



Obrázok 21: Vyhľadávanie používateľa

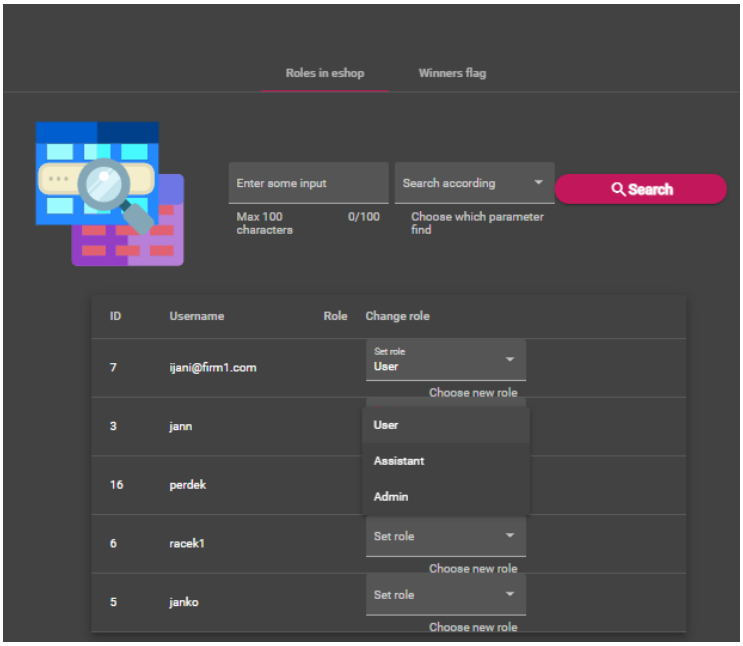
Podľa rozhrania na obrázku 22 môže používateľ s vyššími oprávneniami pridávať nové produkty do databázy obchodu. Po zadaní všetkých informácií o produkte klikne na tlačidlo Insert product (vložiť produkt). Nový produkt bude pridaný do databázy obchodov.



Obrázok 22: Správa produktov

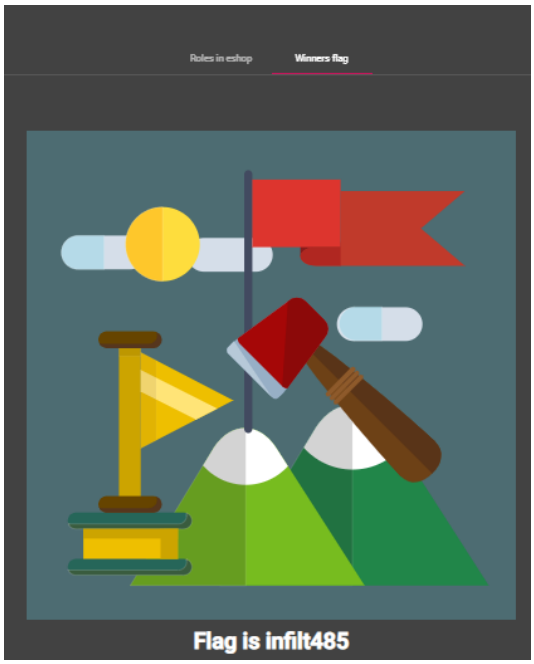
Správa rolí používateľov

V ďalšom okne má používateľ s administrátorskými právami možnosť spravovať roly ostatných používateľov. Kliknutím na jeden z účtov a potom na pole Zmeniť úlohu môže správca priradiť rolu ďalšiemu používateľovi: bežný používateľ, asistent alebo správca.



Obrázok 23 Možnosť zmeny užívateľských rolí

V tejto časti systému sa nachádza aj flag, ktorý by sa útočník v systéme mal pokúsiť získať. Ak sa útočníkovi podarilo prebiť do časti pre zmenu rolí používateľa, uvidí flag z obrázku nižšie a jeho štítok.



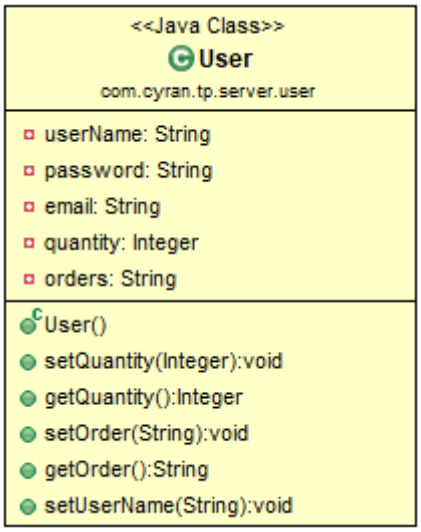
Obrázok 24 Flag ktorý je potrebné získať

Server a riadiaca časť systému

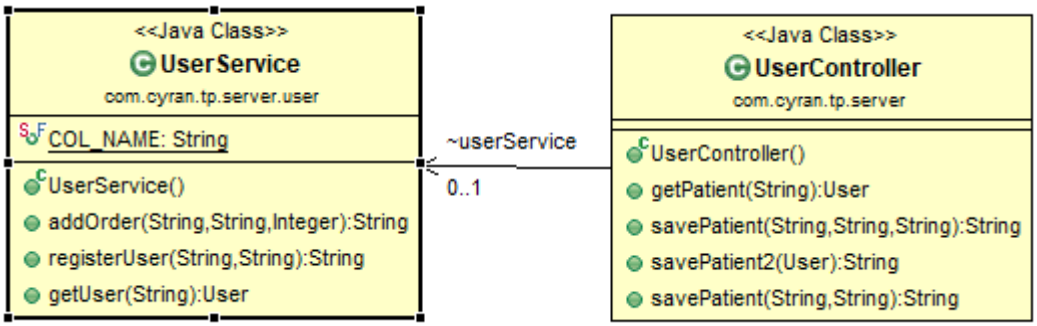
Pre riadiacu časť systému bol zvolený programovací jazyk Java, pričom nad ním je využívaný rámec Spring. Závislosti FireStore sa priamo pridávajú do projektu pomocou správcu závislostí Maven.

Na ďalšom diagrame tried môžeme vidieť hlavné triedy, z ktorých každá predstavuje jednu zo základných entít databázy.

Obrázok 25 Diagram základných tried



Obrázok 26 Trieda User entity



Obrázok 27 Diagram tried obsluhujúcich User entitu

Metódy na diagrame triedy sú pomerne priame a popisujú funkcie slúžiace entite Používateľa. V tomto okamihu poskytuje back-end funkčnosť registrácie a prihlásenia, ako aj objednávania produktov.

Databáza

Ako prvú možnosť implementácie databázy, webový obchod používa flexibilnú databázu NoSql od spoločnosti Google, FireStore. Firestore je optimalizovaný na ukladanie veľkých zbierok malých dokumentov. FireStore je ľahko škálovateľná cloudová databáza založená na dokumentoch.

Databázový model

Štruktúru databázy tvoria 3 primárne modely:

- model používateľa (Users)
- model produktu (Products)
- model objednávky (Orders)

Users

Model používateľa predstavuje registrovaného používateľa, ktorý si úspešne vytvoril účet na webovej stránke. Používateľský model má nasledujúce atribúty:

- userId – jedinečné ID používateľa, na základe ktorého sa vykonáva identifikácia používateľa
- userName – jedinečné užívateľské meno k účtu
- email – e-mailová adresa používateľa
- password – heslo na prístup k používateľskému účtu
- orders – atribút, ktorý odkazuje na model objednávky, tj. hovorí o objednávkach vykonaných z používateľského účtu

Products

Model produktov predstavuje entitu všetkých produktov, ktoré e-shop ponúka. Skladá sa z nasledujúcich atribútov:

- productId - jedinečné identifikačné číslo produktu
- productName - názov produktu
- price - cena produktu
- description - krátky popis produktu
- quantity - číslo, ktoré predstavuje množstvo dostupných produktov
- url - adresa URL, kde sa nachádza obrázok produktuOrders

Orders

Modul Objednávky predstavuje kolekciu všetkých objednávok zadaných v e-shope. Skladá sa z nasledujúcich atribútov:

- orderId - jedinečné číslo objednávky, na základe ktorého je identifikovaná
- creditCard - informácie o kreditnej karte, z ktorej bola platba vykonaná
- shipmentAddress - adresa, na ktorú má byť objednávka doručená
- userName - meno používateľa, ktorý zadal objednávku
- cartInfo - obsahuje presnejšie informácie o objednávke a skladá sa z 2 atribútov:
    - finalPrice - konečná cena objednávky
    - výrobok - odkaz na model výrobku. Obsahuje zoznam objednaných produktov v rámci jednej objednávky

Rozhrania API servera

Nasledujúca tabuľka popisuje rozhrania, ktoré možno použiť na vytvorenie databázových požiadaviek.

Operation	HTTP method	path	returns
Get Single User	GET	/getUser	JSON of User
Register a User	POST	/register	userId
Get a Single Product	GET	/getProduct	JSON of Product
Create a Product	POST	/create/product	productId
Update a Product	POST	/update/product	productId
Create a Order	POST	/create/order	orderId

Tabuľka 1: Rozhrania API servera

Model Users (v postgres SQL databáze)

Model používateľa predstavuje registrovaného používateľa, ktorý si úspešne vytvoril účet na webovej stránke. Tabuľka bola vytvorená pre možnosť použiť SQL útoky. Databáza využíva hosting na https://www.elephantsql.com/. Používateľský model má nasledujúce atribúty:

- id – jedinečné ID používateľa, na základe ktorého sa vykonáva identifikácia používateľa
- name – jedinečné užívateľské meno k účtu
- email – e-mailová adresa používateľa
- password – heslo na prístup k používateľskému účtu

3.3 Scenáre s použitím e-shopu

Vytvorený eshop umožňuje realizáciu niekoľkých scenárov za predpokladu, že budú splnené pre nich určené požiadavky.

- Prelamovanie slabých hesiel – slovníkový útok
- Ukradnutie produktu bez zaplatenia zmenením odoslaných informácií na backend
- Ukradnutie produktu prístupom do adresára s produktami
- SQL injekcia pre zmenu emailu admina
- SQL injekcia pre získanie záznamu z Whois s najväčším počtom zraniteľností a získanie bližšieho popisu k nim – záznam o security eshope

Prelamovanie slabých hesiel – slovníkový útok

Útočník použije nástroj na prelamovanie slabých hesiel, pričom použije ľubovoľný nástroj pre to určený. Môže využiť aj dostupné slovníky. Pre uplatniteľnosť scenára nesmie aplikácia určovať požiadavky na silu hesla a zároveň musí byť slabé heslo prítomné v systéme.

Ukradnutie produktu odoslaním falošnej informácie

Útočník použije nástroj burpsuite alebo iný nástroj ktorý mu umožní zmeniť obsah http requestu na server. Nastaví nulovú hodnotu. Server nesmie kontrolovať vstupu. Kontrola vstupov by mala byť len na používateľskom rozhraní.

Ukradnutie produktu prístupom do priečinka

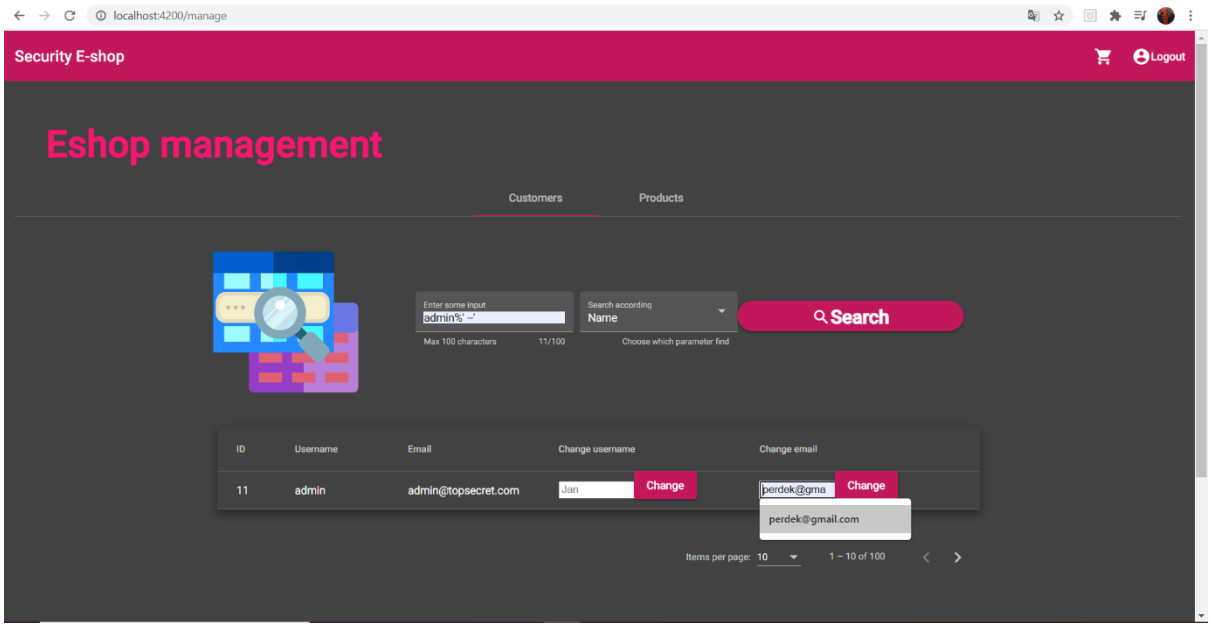
Útočník prehľadá možné adresy kde by sa súbory mohli nachádzať a stiahne potrebné súbory z nich. Je potrebné aby tieto adresáre boli pre útočníka prístupné.

SQL injekcia pre zmenu emailovej adresy admina

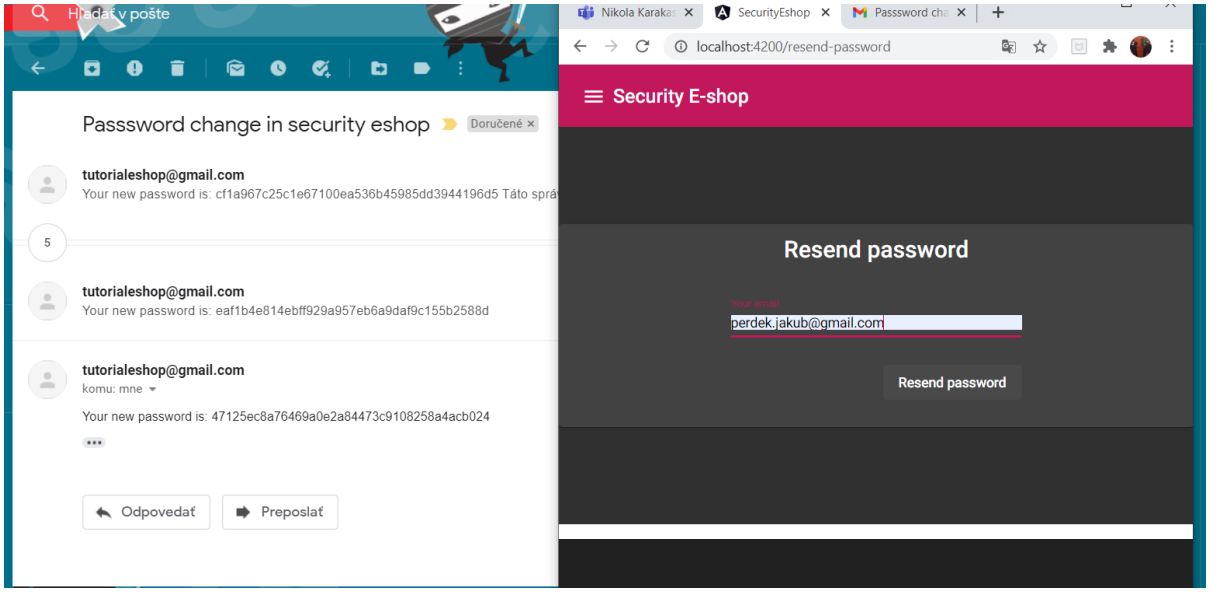
Pri vypisovaní všetkých používateľov v časti systému určenej pre pracovníka eshopu bude účet s oprávneniami správcu vynechaný. SQL dopyt, ktorý vypíše všetkých používateľov, je nasledovný:

SELECT name, email FROM users WHERE name LIKE '%a%' AND name != 'admin'.

Útočník sa pokúša vytvoriť SQL injekciu tak, aby získal informácie o účte s oprávneniami správcu. To je možné vykonať pridaním nasledujúceho dotazu: *admin%' --'* do pola za vyhľadávanie používateľov podľa mena.



Obrázok 28: Aplikovanie SQL injekcie



Obrázok 29: Generovanie a poslanie hesla na email pri jeho strate

Následne útočník v roli predavača zmení email používateľa na nejaký, ku ktorému má prístup. Potom sa odhlási a nechá si vygenerovať nové heslo pre zmenený email. Na zadaný email mu bude doručené zmenené heslo, ktoré použije pri prihlasovaní. Na základe tohto útoku útočník získal privilégiá admina. Tento útok môže realizovať pracovník obchodu, ale primárne je určený v spojení s útokom prelamovania hesiel, v ktorom útočník sa na základe

slabého hesla dostane do role pracovníka v obchode. Pracovník v obchode má nižšie práva ako samotný admin. Obrázky 29 a 30 popisujú uvedený implementovaný útok.

SQL injekcia pre získanie informácií z whois

Vytvorili sme aj komplexnú injekciu, ktorá vyžaduje väčšie úsilie. Vo whois aplikácii nie je realizovaná a pravdepodobne vzhľadom na účel aplikácie ani nebude sprístupnená funkcionalita pre agregačné funkcie umožňujúce napríklad získať doménu s najväčším počtom zraniteľností a podobne. Útočník na základe pokročilej SQL injekcie nechá vyhľadať záznam najväčším počtom zraniteľností. Pri SQL dopyte používame funkciou one, takže bude musieť použiť LIMIT 1 v príkaze. Rovnako jediný vstup z ktorého sa dá urobiť injekcia je ohraničený znakmi % a '. Pre získanie ostatných dát musí ukončiť prvý príkaz a začať písať druhý s agregačnou funkciou, a to tak že vráti len jeden výsledok. Príklad takéhoto príkazu môže vyzerať nasledovne:

```
a%' OR 1=1;
SELECT * FROM whois, (
        SELECT COUNT(vulnerabilities.reference_record_id) AS count, whois.id AS ww
        FROM whois
        LEFT JOIN vulnerabilities ON whois.id = vulnerabilities.reference_record_id
        GROUP BY whois.id
        ORDER BY count DESC
        LIMIT 1
) ww
WHERE ww = whois.id
LIMIT 1 --'
```

3.4 Logovanie v aplikácii

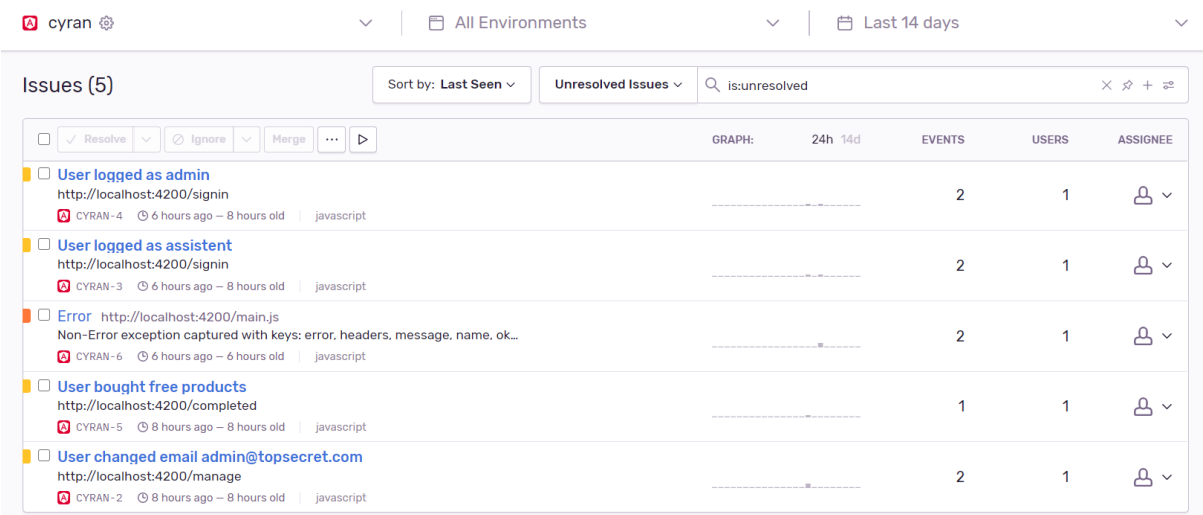
Do aplikácie sme zapracovali aj logovanie s použitím služieb Sentry. Logovanie má veľký význam, hlavne v rámci používateľského prieskumu. Očakávame identifikáciu potencionálnych chýb, ale hlavne možnosť reálne zachytiť pokrok používateľa pri realizácii scenárov. Význam logovania je:

- Pre možnosť zachytiť pokrok používateľa
- Pre odhalenie chýb, ktoré vzniknú pri používaní aplikácie
- Pre možnosť porovnať výsledné hodnoty z prieskumu s reálnym používaním aplikácie
- Pre možnosť sťažiť útočníkovi infiltráciu – za odhalenie môže byť aplikovaný bodový postih

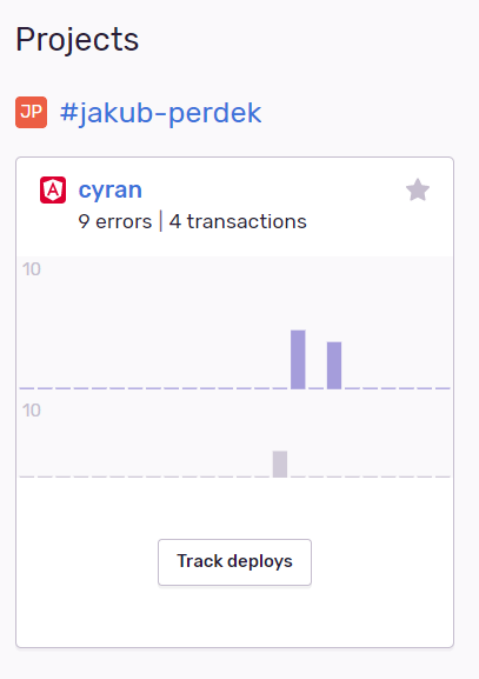
Developérsky plán v Sentry umožňuje v čase písania dokumentácie prijať až 5000 chybových, varovných prípadne iných správ a následne za pomoci knižnice tretej strany napísanej v jazyku python je možné tieto logy exportovať do csv súboru. Knižnica má názov sentry2csv.

Vytvorili sme varovné správy umožňujúce identifikovať, či útočník získal prístup k účtu user alebo admin. Ďalej, či bola zmenená emailová adresa v účte pre používateľa admin. Okrem toho sme ešte v rámci kúpy produktu vytvorili správu informujúcu, že používateľ obišiel možnosť zaplatiť za produkty, aj napriek tomu, že BurpSuite tieto súbory ihneď identifikuje a prístup priamo k nim takúto správu nevyvolá.

Okrem toho logujeme aj prípadné chyby. Ručný test potvrdil, že v prípade výpadku databázy tieto správy sú naďalej logované.



Obrázok 30: Zaznamenané logy



Obrázok 31: Projekt v sentry.io

Správy sú odosielané z aplikácie a následne zachytávané a zobrazované v projekte ako je zobrazené na obrázku 31. Vytvorený projekt cyran v rámci služieb sentry.io zobrazuje obrázok 32. Výsledný súbor so získanými logmi je zobrazený na obrázku 33.

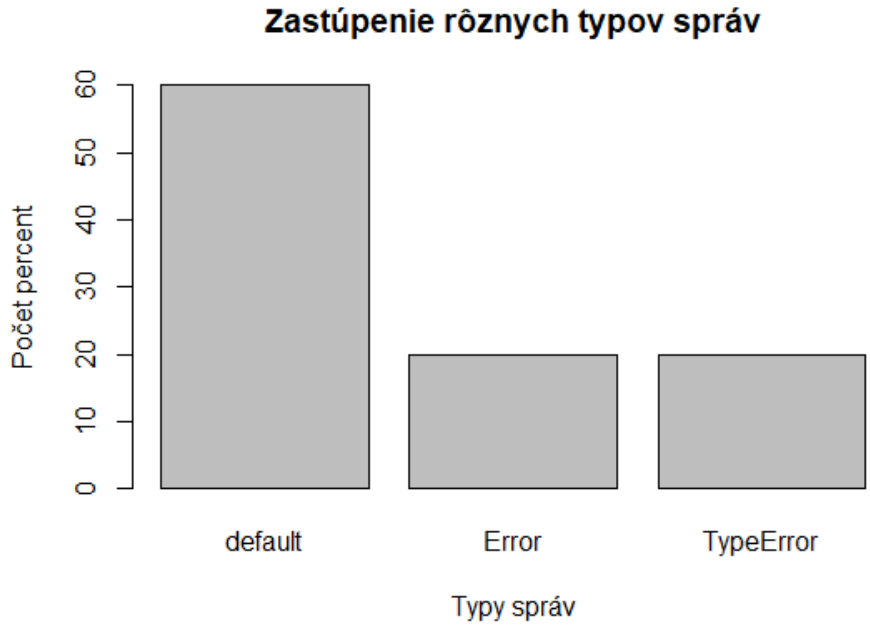
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Error,Location,Details,Events,Users,Notes,Link																
2																	
3	default,http://localhost:4200/signin,User logged as assistent,2,1,,https://sentry.io/organizations/jakub-perdek/issues/2300743017/																
4																	
5	Error,http://localhost:4200/main.js,"Non-Error exception captured with keys: error, headers, message, name, ok...",2,1,,https://sentry.io/organizations/jakub-perdek/issues/2300848250/																
6																	
7	default,http://localhost:4200/completed,User bought free products,1,1,,https://sentry.io/organizations/jakub-perdek/issues/2300754646/																
8																	
9	default,http://localhost:4200/signin,User logged as admin,1,1,,https://sentry.io/organizations/jakub-perdek/issues/2300745018/																
10																	
11	default,http://localhost:4200/manage,User changed email admin@topsecret.com,2,1,,https://sentry.io/organizations/jakub-perdek/issues/2300710095/																
12																	

Obrázok 32: Získané logy vo formáte csv

Analýza dát zo Sentry

Po skončení používateľského testovania sme pomocou skriptu získali reporty od používateľov, ktoré sa zaznamenali do Sentry. Výsledné údaje obsahovali niekoľko stĺpcov. Prvým bol typ záznamu. Mohla to byť bežná varovná správa označovaná ako východzia alebo rôzne druhy chýb. Nasledovala lokácia webovej stránky na ktorej používateľ vyvolal udalosť z ktorej vznikol report. Tretím v poradí boli detailné informácie popisujúce report. Pri varovnej

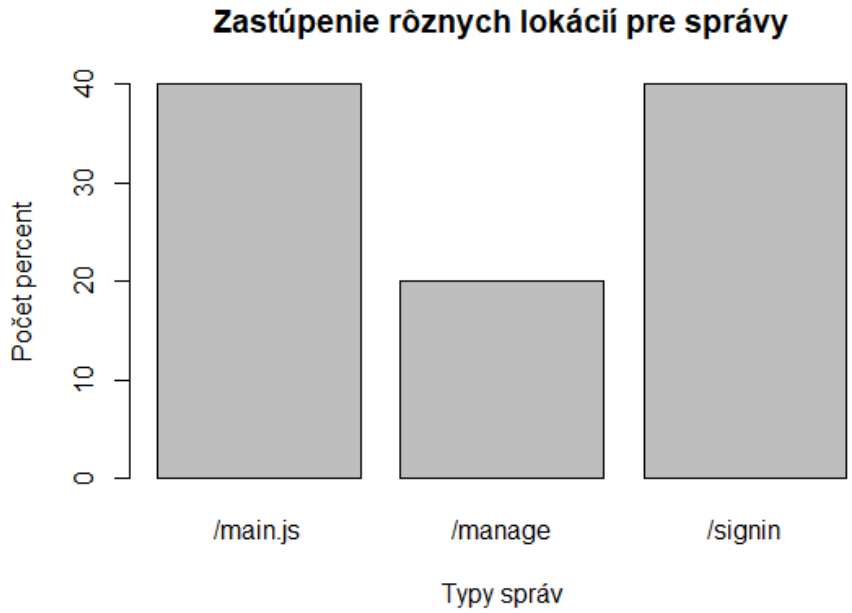
správe to bol obsah tejto správy. Napríklad obsahom bolo, že sa používateľ prihlásil ako admin. Ďalšie dva stĺpce boli číselné. Prvý z nich obsahoval informáciu o počte udalostí s konkrétnym typom záznamu. Druhý identikoval počet jedinečných používateľov, od ktorých sa konkrétny report vygeneroval. Súbor obsahoval ešte stĺpce ako poznámky alebo link na stránku zo Sentry, ale tie z hľadiska vyhodnotenia nemali význam, keďže poznámky neboli zaznamenané žiadne a link je z hľadiska spracovania zbytočný. Uvedené stĺpce sme podrobili analýze a následne sme ich aj vizualizovali.



Obrázok 33: Zastúpenie rôznych typov správ

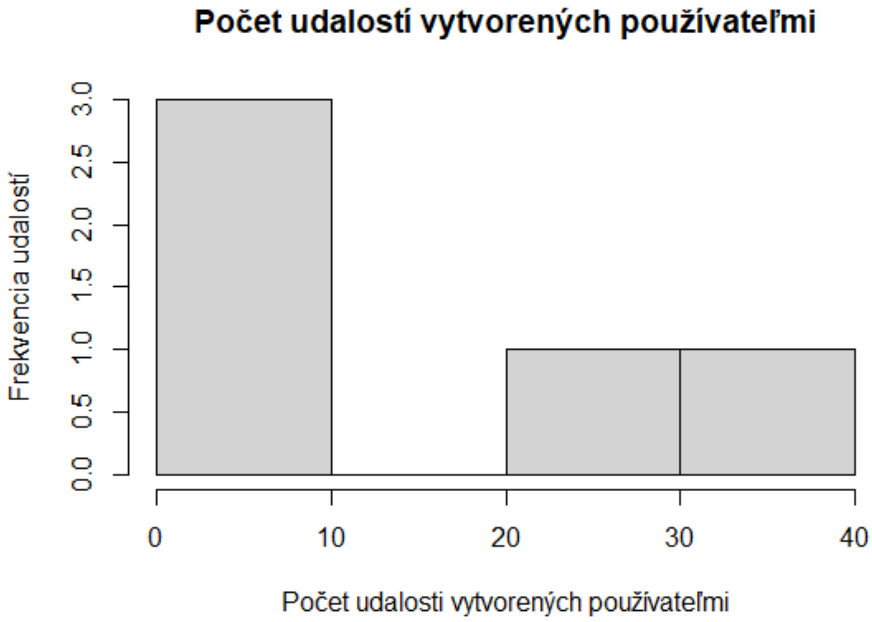
Informácie o akciách používateľa tvorili 60 percent. Zvyšných štyridsať percent tvorili chyby vzniknuté pri používaní aplikácie. Boli zapríčinené chybou v prehliadači používateľa. Polovica percent z týchto chýb boli typové chyby.

Reporty boli generované z rôznych stránok frontendu. Najviac udalostí bolo generovaných po prihlásení používateľa a z hlavného javascriptovského súboru, ktorých bolo až 40 percent. Tie druhé chyby vznikli v javascriptovom súbore. Udalosti pri prihlásení tvorili dva základné súčasti scenárov ako vniknutie do účtu asistenta v obchode a nakoniec aj admina. Zvyšných dvadsať percent pochádzalo s manažérskeho prostredia asistenta v eshope, kde bola generovaná udalosť zmena emailu administrátora pomocou SQL injekcie.



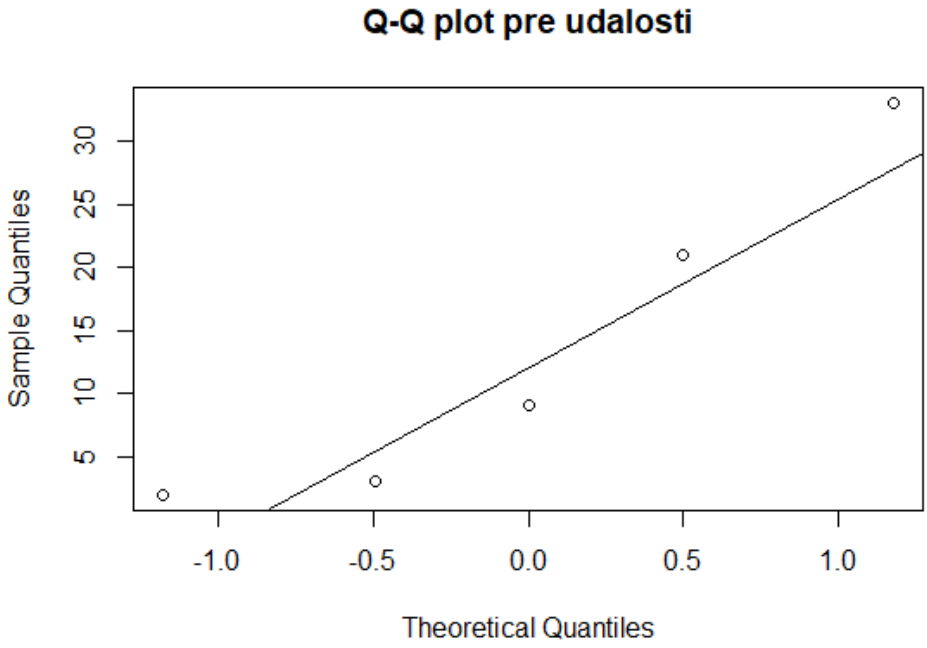
Obrázok 34: Zastúpenie rôznych lokácií

Ďalej sme vyhodnotili frekvencie pri jednotlivých počtoch udalostí. Konkrétny report sa vygeneroval v 3 prípadoch maximálne v 10 kusoch. Pri veľkosti od 20 do 30 kusov to bolo len pri jednom type reportu. Rovnako jeden typ reportu mal frekvenciu od 30 do 40 kusov. Používatelia pri riešení na tomto úseku pravdepodobne nevedeli ako pokračovať. Najviac frekventovanou udalosťou bol jeden druh chyby a hneď po nej to bolo prihlásenie do účtu asistenta v eshope.



Obrázok 35: Počet udalostí vygenerovaných používateľmi

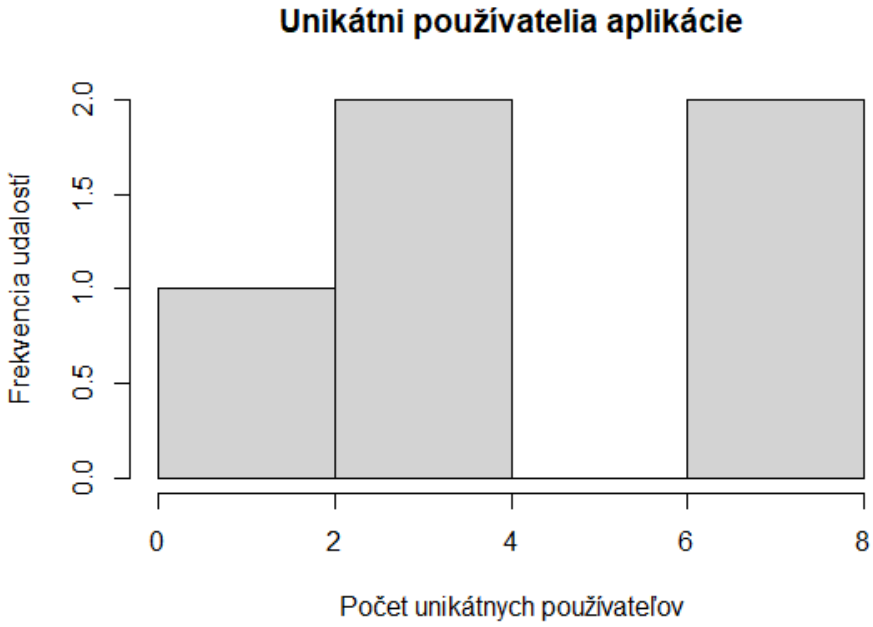
Pri udalostiach nás zaujímalo aj ich rozdelenie. Vytvorili sme preto QQplot a na základe okometrickej metódy sme usúdili, že je to normálne rozdelenie. Následne sme na finálne potvrdenie tohto predpokladu použili Shapirov-Wilkov test normálnosti. V tomto teste vyšlo W = 0.96307, p-value = 0.7982, pričom p bolo vysoké. Dáta sú normálne ak p > 0.05. V tomto teste vyšlo 0.7982 >> 0.05, čo potvrdzuje, že dáta sú normálne. Netreba zabudnúť, že test bol realizovaný iba na piatich reportovacích správach.



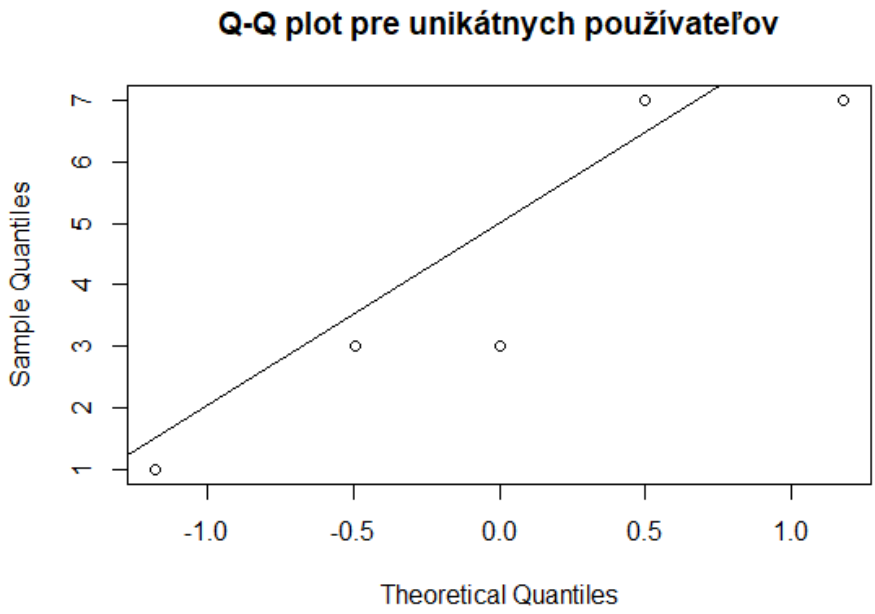
Obrázok 36: QQ plot pre udalosti

Analyzovali sme aj unikátnych používateľov. Zároveň informácia o nich je veľmi dôležitou kvôli prehľadu o počte zúčastnených hlavne vzhľadom na počet generovaných udalostí. V dvoch prípadoch sa vygenerovali 2 udalosti. V prvom prípade pre 2 až 4 používateľov a v druhom pre 6 až 8 používateľov. Jedna udalosť sa vygenerovala 0 až 2 používateľom. Opäť aj v tomto prípade chybová udalosť a vniknutie do účtu asistenta boli časté udalosti.

Aj v prípade unikátnych používateľov sme testovali či dáta sú normálne. Na základe okometrickej metódy boli niektoré body viditeľne vzdialené od priamky v diagrame. Testovanie Shapiro-Wilksovou metódou však aj v tomto prípade potvrdilo normálnosť dát. Hodnoty testu boli W = 0.89495, p-value = 0.4064. Hodnota bola menšia ako v predchádzajúcom prípade ale stále mnohonásobne väčšia ak 0.05. Analyzovaných reportov aj v tomto prípade bolo len 5.



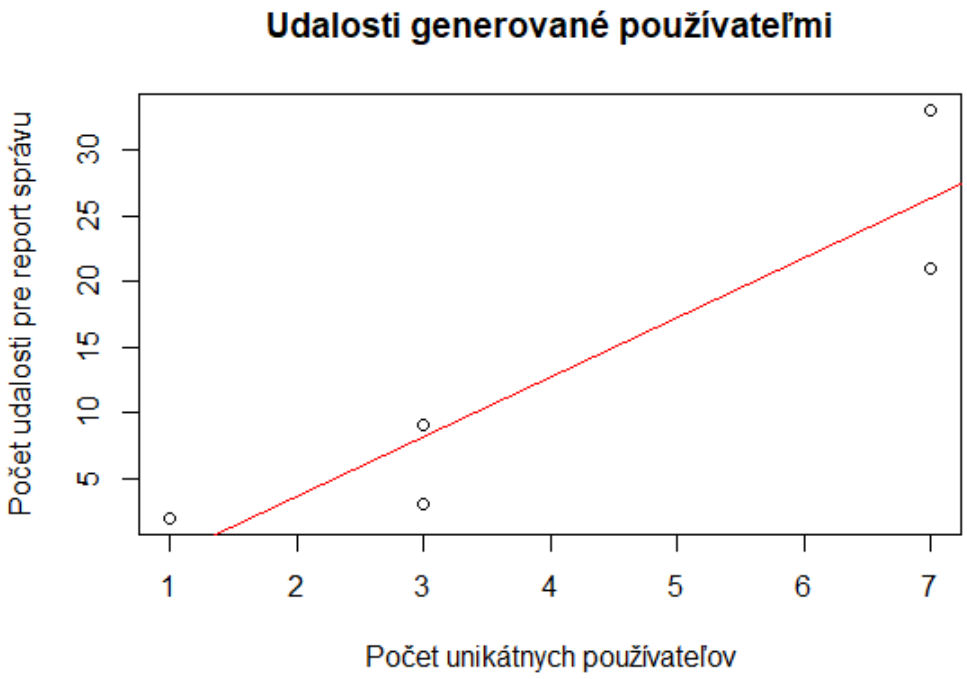
Obrázok 37: Unikátni používatelia aplikácie



Obrázok 38: QQ plot pre unikátnych používateľov

Finálnym vyhodnotením bola závislosť unikátnych používateľov od udalostí. Zostrojili sme lineárny model a preň aj graf. Body od zostrojenej priamky boli mierne vzdialené. Pri analýze modelu sa aj potvrdilo, že vzdialenosť bodov je badateľná. Hodnota p vyšla 0.0274, čo je menej ako 0.05 a postačuje to na zamietnutie nulovej hypotézy, ale reálne by táto hodnota mala byť niekoľkonásobne menšia. Determinačný koeficient vyšiel 0.844. Tento výsledok

potvrdil, že model dobre zachytáva vlastnosti dát. Táto hodnota ale zvykne kolísať pri väčšom množstve dát. V našom prípade bolo analyzovaných len 5 reportov.



Obrázok 39: Udalosti generované používateľmi

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -5.417      5.428  -0.998   0.3918
Users          4.528      1.122   4.035   0.0274 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.022 on 3 degrees of freedom
Multiple R-squared:  0.8444,    Adjusted R-squared:  0.7926
F-statistic: 16.28 on 1 and 3 DF,  p-value: 0.02737
```

Obrázok 40: Výsledky pre vyhodnotenie lineárnosti modelu

Základné údaje o používateľoch:

Priemer je: 4.2
Maximálna hodnota je: 7
Minimálna hodnota je: 1
Rozsah je: 6
Variancia je: 7.2
Modus je: 7
Medián je: 3

Základné údaje o udalostiach:

Priemer je: 13.6
Maximálna hodnota je: 33
Minimálna hodnota je: 2
Rozsah je: 31
Variancia je: 174.8
Modus je: 2
Medián je: 9

Analýza dát z Google forms

Spätnú väzbu používatelia zadali aj do google forms dotazníkov.

Počet meraní: 6
Úlohy:

1. Prelomenie hesla pre používateľa user.
2. Získanie administrátorských údajov pomocou SQL injekcií.
3. Získanie vlajky.
4. Odoslanie prvého requestu pri pokuse o ukradnutie produktu.
5. Zmenenie hodnôt v druhom requeste na 0.
6. Získanie súboru pomocou URL adresy.

Úspešnosť účastníkov:

0. Celý kurz prešlo 50% účastníkov.
1. Heslo používateľa user prelomilo 100% účastníkov.
2. Administrátorské údaje pomocou SQL injekcií získalo 83.3% účastníkov.
3. Vlajku získalo 100% účastníkov
4. Prvý request pri pokuse o ukradnutie produktu odoslalo 100% účastníkov.
5. V druhom requeste zmenilo hodnoty na 0 33.3% účastníkov.
6. Súbory pomocou zmeny URL získalo 33.3% účastníkov.

Vyhodnotenie:

Môžeme vidieť že najväčšie problémy mali účastníci s úlohami 5 a 6, ktoré nesplnilo až 67.7 % z nich.

Náročnosti úloh:

	Počet hlasov				
Úloha	**Veľmi ľahké**	**Ľahké**	**Stredne ťažké**	**Ťažké**	**Veľmi ťažké**
7. Prelomenie hesla pre používateľa user.	2	2	2	0	0
8. Získanie adminových údajov pomocou SQL injekcií.	1	3	1	1	0
9. Získanie vlajky.	2	2	2	0	0
10. Odoslanie prvého requestu pri pokuse o ukradnutie produktu.	1	3	1	1	0
11. Zmenenie hodnôt v druhom requeste na 0.	3	0	2	0	1
12. Získanie súboru pomocou URL adresy.	2	0	3	0	1

Obrázok 41: Náročnosti úloh

Vyhodnotenie:

Z tabuľky môžeme vidieť že používatelia hodnotili, ako najnáročnejšie, úlohy 5 a 6 a ako najjednoduchšie úlohy 1 a 3.

Trvanie úloh

	Počet hlasov					
Úloha	**1-5 min.**	**5-10 min.**	**10-15 min.**	**15-20 min.**	**20-30 min.**	**30-45 in.**
1. Prelomenie hesla pre používateľa user.	1	3	0	1	0	1
2. Získanie adminových údajov pomocou SQL injekcií.	1	2	2	1	0	0
3. Získanie vlajky.	2	4	0	0	0	0
4. Odoslanie prvého requestu pri pokuse o ukradnutie produktu.	1	4	1	0	0	0
5. Zmenenie hodnôt v druhom requeste na 0.	4	1	1	0	0	0
6. Získanie súboru pomocou URL adresy.	3	3	0	0	0	0

Obrázok 42: Trvanie úloh

Vyhodnotenie:

Z tabuľky môžeme vidieť, že najkratšie trvali úlohy 3 a 6 a jednému používateľovi trvala úloha 1 od 30 do 45 min.

Celkové trvanie kurzu:

20, 35, 40, 50, 65, 95 minút.

Poznámky od používateľov:

1. Ulohy skor nefungovali (teda posledne dve), odoslanie objednavky nepreslo, dostavam CORS missing allow header, pretoze ten request nikde neprejde po 5 minutach cakania. Inak boli ulohy vpohode, ale trochu asi jednoduche. Urcite by som uvital viac uloh, pripadne mozno nie tak jednoznacne, ze tam pridat trochu zakernych veci :) Zvysok feedbacku posielam na xperdek@stuba.sk.
2. Všetko bolo pekne spracované a išlo to pomerne plynulo. Úlohy mierne sťažila dokumentácia, ktorá by mohla byť o niečo viac jednoznačnejšia. Za veľký problém

považujem setup Burp Suitu a určite by som niekde zakomponoval poznámku, že v prípade, ak nám nejde aj po vypnutí interceptu pripojiť sa na akúkoľvek stránku, tak stačí zmeniť port napríklad 8085. S týmto som strávil peknych pár minút, ale našťastie mi Jakub Pedrek a Nikola Karakaš boli ochotní pomôcť. Taktiež je vhdoné niekedy v dokumentácii pripomenúť, na ktorom lokálnom porte sa nachádza emailová schránka, keďže sa mi to v priebehu riešenia "podarilo úspešne" zabudnúť. Inak všetko hodnotím veľmi dobre a práca tímu sa mi naozaj páči.

3. lebo bolo treba spristupnit port 8080 aj ked sa pisalo na dockeri, ze not accessible, adresy na localhost a nie na 192.168....
4. Adresy sú fixnuté na localhost. Ja som spustil docker na inej IP a z druheho PC som sa na to chcel prihlásiť, ale nešlo to. SqlMap nefungoval, backend vracal 500-ky.

Vyhodnotemie:

Z poznámok môžeme usúdiť že:

Problémy z úlohami 5 a 6, ktoré sme si všimli z predchádzajúcich hodnotní môže byť spôsobené technickými problémami alebo zlým pochopením úloh. Dokumentácia, hlavne príručka pre používateľov, by mohla byť viac prepracovaná a detailnejšia. Mohla by obsahovať informáciu na ktorom porte sa nachádza emailová schránka.

Celkové vyhodnotenie:

Najväčší problém, ktorý sme z tejto spätnej väzby zaznamenali bol že niektorí používatelia mali problém dokončiť úlohy 5 a 6 kvôli technickým problémom špecifikovaným v poznámkach od používateľov. Ďalej môžeme z hodnotenia používateľov vidieť že úlohy 1 a 3 boli priveľmi ľahké. Používatelia by ocenili väčšie množstvo úloh.

3.5 Posudky na prototyp tímu č. 19

Na základe spätnej väzby z používateľského testovania sme vypracovali posudok podľa určeného protokolu. Rovnako sme zaslali posudky priamo na vypracovanie používateľom z používateľského testovania, ktorí boli účastníci bezpečnostného semináru. Niektorí nám neodpovedali. Od dvoch používateľov sa podarilo získať vyplnený protokol.

Posudok na základe používateľského testovania

1. Študenti bezpečnostného semináru: Posudok na prototyp tímu č. 19: CYRAN

1.1. Úvod

Testovanie aplikácie vykonali študenti bezpečnostného semináru a každý z nich pridal spätnú väzbu a svoje postrehy. Projekt sa im veľmi páčil svojou originalitou ale rovnako aj kritizovali objavené nedostatky. Hodnotili, respektíve sa vyjadrovali ku kvalite výstupu, jednotlivým scenárom, kvalite používateľskej príručky a úrovni náročnosti, prípadne aj návrhu scenárov

1.2. Hodnotenie prototypu

Študenti hodnotili jednotlivé scenáre, ktoré sa zároveň snažili prejsť. Rôzni používatelia mali rôznu úroveň schopností a znalostí z penetračného testovania. Niektorí zhrnuli iba identifikované chyby, ktoré ich pri riešení spomalili. Boli nimi nefunkčnosť vedľajšieho scenáru kvôli problémom s autentifikáciou do FireBase databázy, chýbajúce detaily v používateľskej príručke a podobne. Používatelia žiadali upraviť aj otázky v Google Forms pre možnosť lepšej spätnej väzby. Svoje postrehy zhrnuli v svojich emailoch.

Jeden používateľ sa vyjadril o priamočiarosti úloh a dobrým zakomponovaním šifrovaním hesla a potrebou vynájsť sa pri jeho prelamovaní. Zároveň by chcel viac podobných úloh nútiacich rozmýšľať. Sám poskytol aj nejaké návrhy pre uskutočnenie. Očakával aj nejakú základnú ochranu na frontende proti SQL injekciám.

Ďalší používateľ mal problém s opísanými chýbajúcimi detailami v používateľskej príručke, ale po kontaktovaní členov tímu sa podarilo všetko vyriešiť. K samotnému riešeniu sa vyjadril, že je to dosť dobré.

1.3. Hodnotenie práce tímu

Študenti sa nezaoberali prácou tímu, ale jednotliví členovia im počas testovania v prípade problémov asistovali poskytovaním rád a ďalšej pomoci. Najčastejšie problémy boli z BurpSuitom a chýbajúcou informáciou o lokálnom emailovom klientovi v používateľskej príručke, aj napriek tomu, že inštalačná príručka tento detail obsahovala.

1.4. Zhodnotenie

Aplikácia sa študentom páčila a pozitívne ju hodnotili. Zároveň poukazovali na nefunkčnosť jedného zo scenárov a nekompletnosť setupu BurpSuite nástroja spolu s chýbajúcou informáciou o lokálnom emailovom klientovi, ktorá bola spomenutá v inštalačnej príručke. Úlohy im prišli intuitívne, priamočiare, s originalitou zapracovania ale chceli by ich viac a s väčším sťažením riešenia týchto úloh. Väčšina študentov, ktorá produkt testovala dokončila hlavný scenár.

Posudky jednotlivých testerov z bezpečnostného seminára

2. Účastník 1: Posudok na prototyp tímu č. 19: CYRAN

2.1. Hodnotenie prototypu

Celkovo si myslím, že tento projekt má potenciál, študenti dokázali navrhnúť celkom slušný prototyp, ktorý viac menej fungoval, aj keď bolo odhalených pár chýb, ktoré ale nebude zložité opraviť. Základný prototyp je dosť dobrý, aj keď ja by som uvítal viac rôznorodých úloh, ktoré treba v tejto hre splniť. Možno by sa tam hodili viaceré úrovne obťažnosti. Inak bol prototyp solídny a dobre pripravený. Inštalácia bola jednoduchá a dokumentácia obsahovala zväčša dostatočné inštrukcie na splnenie potrebných krokov a úloh.

2.2. Zhodnotenie

Výsledok práce tímu je veľmi dobrý a spĺňa ciele, ktoré boli v tomto projekte určené. Implementácia je na veľmi dobrej úrovni, ale v niektorých častiach je úroveň nižšia, prípadne niektoré časti nie sú plne dokončené.

Pár nedostatkov informácií bolo už pri inštalácii projektu na vlastnom stroji, v tejto oblasti je určite možné vylepšiť dokumentáciu a pridať aj návody na riešenie bežných problémov s nasadením. Pri niektorých úlohách boli inštrukcie nedostatočné, a bolo potrebné investigovať problémy, ktoré bránili v splnení úlohy. Jedna úloha nefungovala vôbec, zdalo sa, že backend aplikácie nemá implementované potrebné API volanie.

Úlohy sa mi páčili, a niektoré by bolo lepšie zťažiť, a pridať ešte pár ďalších, aby hra trvala trochu dlhšie. Prípadne by som ocenil viac kreativity, aby sa z úloh študent naučil zaujímavé techniky, ktoré sa využívajú pri exploitácii webových služieb.

3. Účastník 2 (tím č. 15): Posudok na prototyp tímu č. 19: CYRAN

3.1. Hodnotenie prototypu

Tím Y sa zaoberal projektom zameraným najmä na vytvorenie interaktívnej hry poukazujúcej na riziká a nebezpečenstvá spojené s manažovaním a tvorbou e-shopu. Pri testovaní som postupoval podľa dodanej príručky. Pripomienky k nej sme dodal už v čase testovania, t.j. Apríl 2021. V konečnom dôsledku sa mi podarilo všetko správne a bez väčších problémov nainštalovať. Musím zhodnotiť, že predvedený produkt na mňa pôsobil zaujímavo a jeho edukatívna forma priniesla svoje ovocie v podobe nadobudnutia nových znalostí. Pozitívne hodnotím najmä prístup, akým tvorcovia vysvetľujú danú problematiku. Jediným negatívom spojeným s projektom boli časte gramatické chyby, ktoré sa časom určite podarilo/podarí odstrániť.

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Sprievodca lokálnou inštaláciou aplikácií

Tímový projekt

Tím č. 19

Vypracoval: Nikola Karakaš
Vedúci projektu: Ing. Pavol Helebrandt Phd.

Sprievodca

Táto používateľská príručka je zameraná na zoznámenie používateľov so základnými krokmi spôsobu spustenia aplikácie. Obsahuje skrátený návod na stiahnutie a inštaláciu aplikácie. Tu sú tiež všetky užitočné adresy, ktoré musí používateľ poznať, aby mohol úspešne získať prístup k všetkým samostatným častiam aplikácie (webová stránka, aplikácia whois, e-mailový klient).

1. Prerequisites

Docker - Aplikácia je kontajnerovaná, a preto je potrebné mať nainštalovaný Docker, ktorý umožní virtualizáciu aplikácie na úrovni OS so všetkými potrebnými doplnkami na jej spustenie. Program Docker si môžete stiahnuť z oficiálnej webovej stránky https://www.docker.com/ a jeho použitie je bezplatné. Pri inštalácii Dockeru je tiež potrebné nainštalovať subsystém Widnows pre Linux (WSL 2), ktorý umožňuje spustenie kontajnerov Linux v systéme Windows.

Github pre Destkop - voliteľne. Aplikácia Cyran sa nachádza na stránkach github a je potrebné ju odtiaľ stiahnuť. Aplikáciu je možné stiahnuť z githubu priamym stiahnutím zip priečinka alebo klonovaním adresára github do lokálneho počítača.

2. Download

Aplikácia je k dispozícii na stránkach github na odkaze https://github.com/Abdo-Saleh/eshop-security. Aplikáciu je možné stiahnuť dvoma spôsobmi: a) priamym stiahnutím priečinka ZIP z github a b)príkazom git bash git clone https://github.com/Abdo-Saleh/eshop-security.git.

3. Installation

Pri štarte aplikácie je potrebné urobiť nasledovné kroky:

a) Otvorte príkazový riadok
b) Prejdite do priečinka, kde sa nachádza stiahnutá aplikácia
c) spustiť príkaz "docker-compose -f docker-compose-local-with-mail-whois.yml up –build"

4. Interaction with the application

Po úspešnom vytvorení aplikácie k nej môžeme pristupovať prostredníctvom internetového prehliadača. Aplikácia je nasadená na nasledujúcich portoch, ktoré zadáme v paneli s adresou:

a) localhost: 4200 - na tomto porte je cieľový e-shop
b) localhost: 5001 - whois aplikácia, ktorá pomáha pri prelomení hesiel
c) localhost: 8025 - e-mailovy klient, pomocou ktorého môžeme prijímať e-maily z e-shopu
d) localhost: 8080 - backend. Backend nie je viditeľný a nemusíme ho používať

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Používateľská príručka pre security e-shop

Tímový projekt

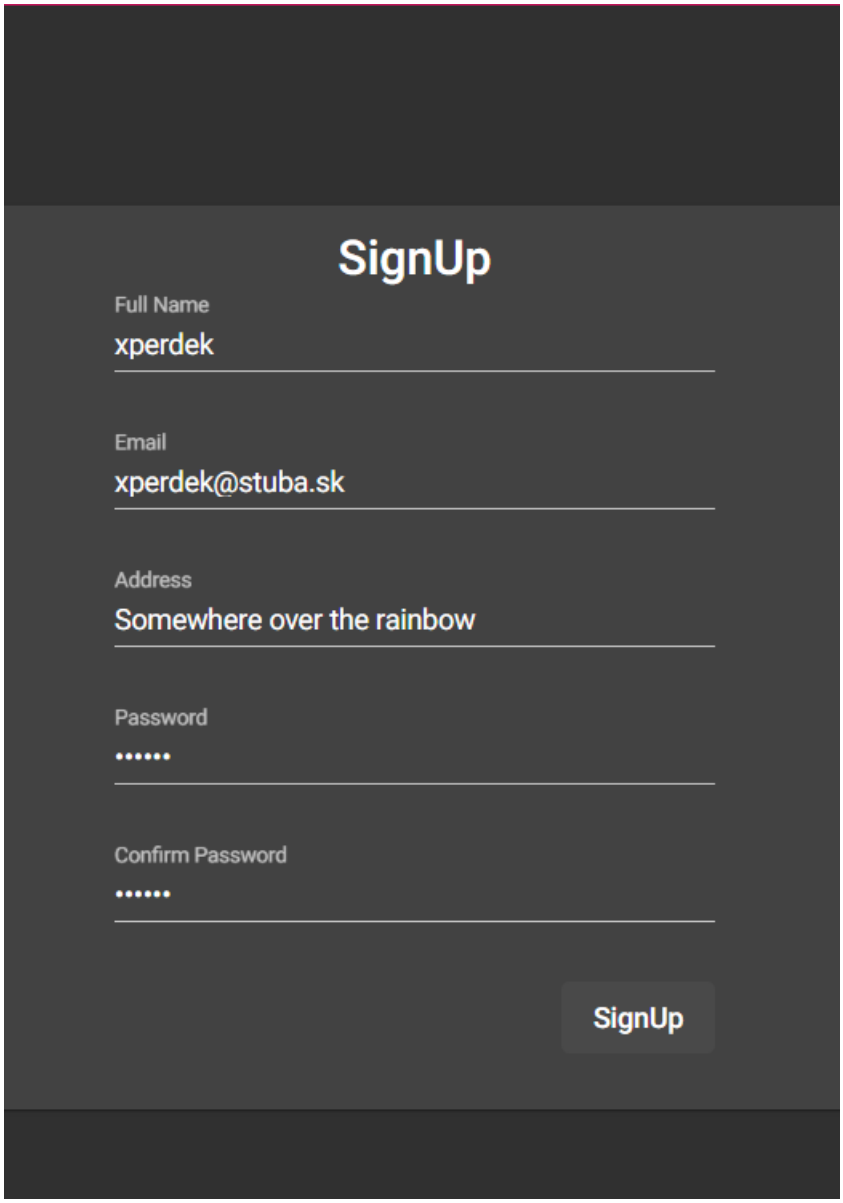
Tím č. 19

Vypracoval: Jakub Perdek
Vedúci projektu: Ing. Pavol Helebrandt Phd.

Registrácia a prihlásenie používateľa

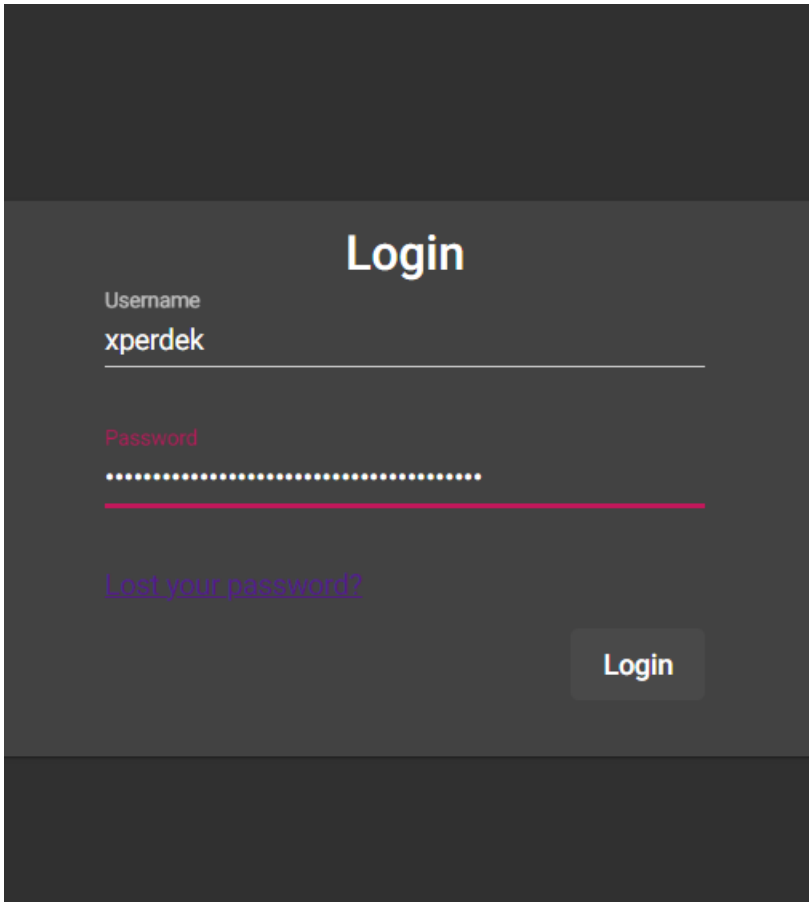
Na začiatku sa používateľ zaregistruje. Vyplní všetky položky registračného formulára. Zapamätá si meno a heslo a uvedie funkčný a jedinečný email. Následne použije meno a heslo pri prihlasovaní. Automaticky mu bude priradená roľa používateľa.

1. Zaregistrujte sa stlačením na tlačidlo SignUp v hornom rohu stránky.



Obrázok 1: Registrácia používateľa

2. Následne sa prihláste zadaním vášho používateľského mena a hesla.

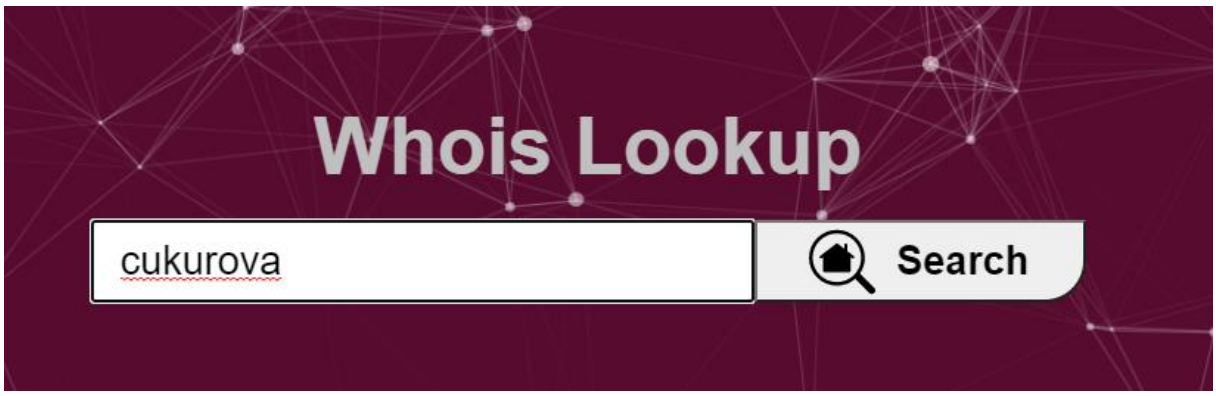


Obrázok 2: Prihlásenie používateľa

Získanie informácií o najzraniteľnejšej stránke

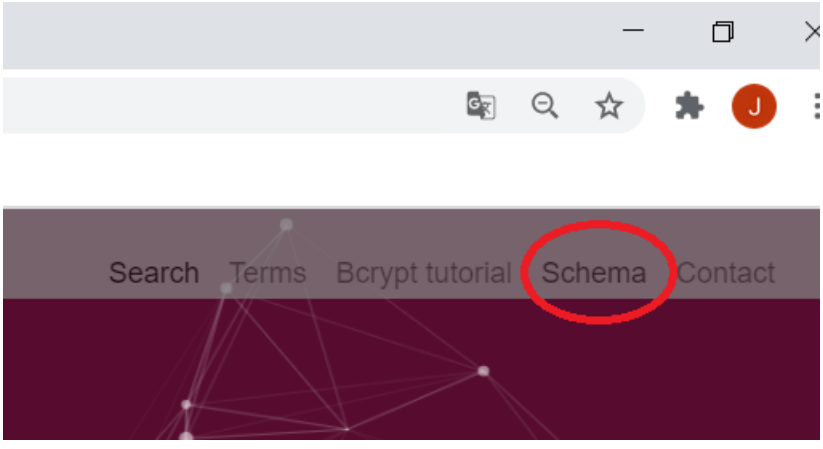
Tento scenár prezentuje pokročilú SQL injekciu. Samotné získané informácie v tomto scenári môžu uľahčiť realizáciu ďalších scenárov. Útočník sa rozhodne získať informácie z whois databázy s tým, že ho zaujíma doména s najväčším počtom zraniteľností. Vyhľadávanie mu ale vráti maximálne jednu stránku, ktorá sa najviac zhoduje s vyhľadávaným výrazom. Už asi tušíte, že potrebujete nejaký dopyt využívajúci agregačné funkcie. Našťastie Whois aplikácia poskytla schému z databázy, keďže chce prezentovať používané postupy. K potrebným informáciám sa môžete dostať na základe nasledovného postupu:

1. Otvorte whois aplikáciu dostupnú na localhost:5001
2. Zistiť ako funguje Whois môžete vyhľadaním zadaného reťazca s doménou.



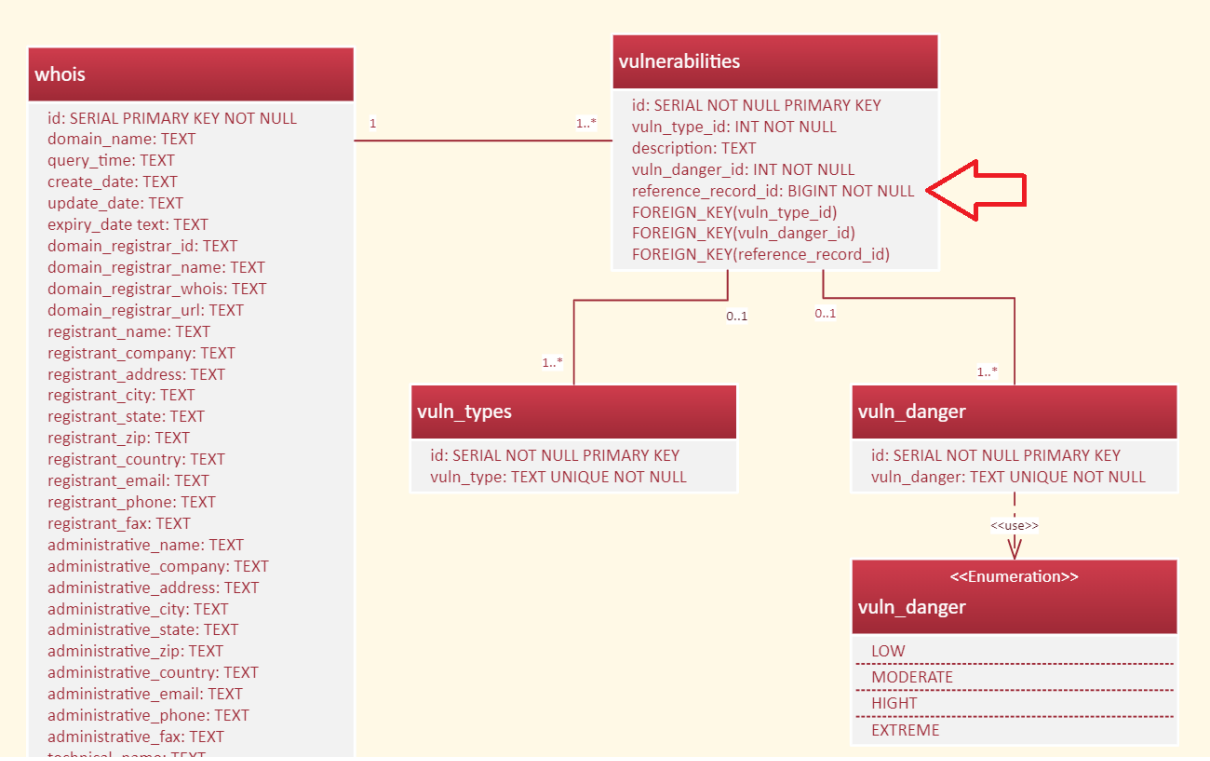
Obrázok 3: Overenie funkcionality whois

3. Následne sa presmerujte na stránku s dátovým modelom whois



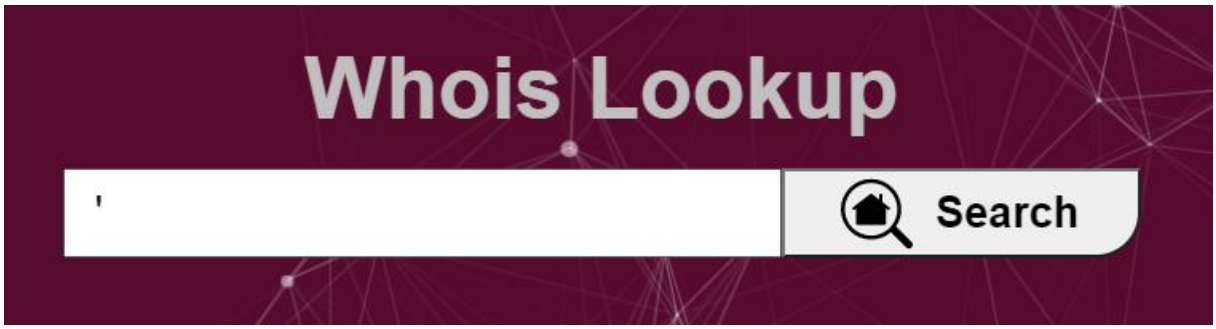
Obrázok 4: Presmerovanie sa na stránku s whois schémou

4. V menu zvoľte Schema. Dostali ste sa na stránku s databázovou schémou. Zo schémy môžete zistiť, že whois tabuľka so záznamami z ktorých sa vyhľadáva sú prepojené s tabuľkou zraniteľností pomocou cudzieho kľúča reference record id.



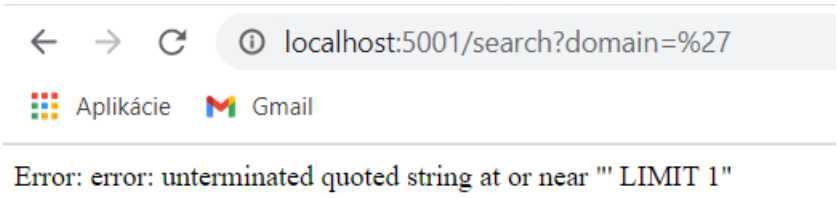
Obrázok 5: Databázová schéma aplikácie Whois

5. Teraz už viete, čo môžete pri písaní SQL injekcie využiť. Ešte je potrebné overiť, či injekcia bude fungovať. Presmerujte sa preto na úvodnú stránku s vyhľadávačom.
6. Zadajte do okna ‘ a potvrďte. V špeciálnych prípadoch môže byť ochrana vo formulároch na frontende. Pokiaľ by bola bolo by potrebné použiť na odosielanie requestov BurpSuite. Jeho použitie si ukážeme v ďalších scenároch.



Obrázok 6: Overenie, či SQL injekcia bude fungovať

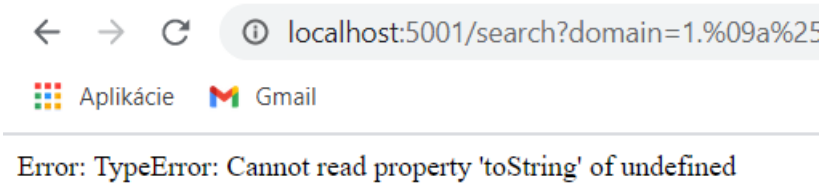
7. Zobrazila sa vám chybová hláška, na základe ktorej viete, že prípadná SQL injekcia bude úspešná.



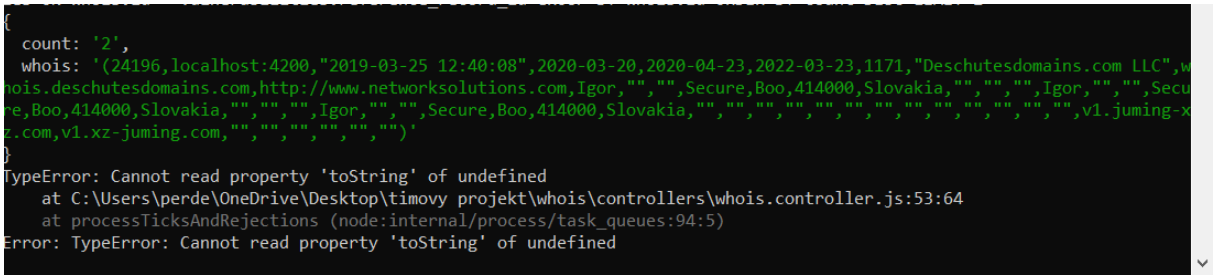
Obrázok 7: Chybové hlásenie zobrazujúce neošetrenú slabinu v systéme

8. Konečne môžete navrhnúť SQL injekciu. Najprv je potrebné zistiť ako funguje vyhľadávanie na základe reťazca. Keďže slovo je hľadané kdekoľvej v doméne potom možno usúdiť, že v postgrese je výraz ohraničený a vyzerá nasledovne: '%vyhladavana_domena%'
   Preto je potrebné najprv uzatvoriť predtým vyhľadávaný reťazec a zároveň zabezpečiť, aby podmienka pre akýkoľvek platila, napríklad použitím logického OR a výrazu, ktorý bude vždy pravdivý.
   Zatiaľ sme navrhli výraz: a%' OR 1=1
   Ten je potrebné ešte okomentovať a vložiť pred komentár ohraničenie pre vrátenie práve jedného výsledku, lebo v kóde sa vracia najviac jeden vyhľadaný a pravdepodobne sa volá funkcia one. Ak by bolo vrátených viac výsledkov skončí s chybou. Pridáme preto na koniec reťazec LIMIT 1 --'
   Zatiaľ máme: a%' OR 1=1 LIMIT 1 --'
9. Môžete skúsiť použiť výraz a%' OR 1=1 LIMIT 1 --' vo vyhľadávaní. Vidíte, že bez chyby vráti nejaký výsledok. Vy ale chcete aby bo, vrátený výsledok s najväčším počtom zraniteľností.
10. Do reťazca doplňte agregačný dopyt na základe ktorého bude možné získať potrebné výsledky. Vyžite informácie zo schémy Whois aplikácie. Keby sme mali prístup k database napísali by sme takýto SELECT:
    SELECT
    COUNT (vulnerabilities.reference_record_id) AS count, whois AS whois
    FROM whois
    LEFT JOIN vulnerabilities ON whois.id = vulnerabilities.reference_record_id
    GROUP BY whois.id
    ORDER BY count DESC
    LIMIT 1
    Týmto selectom na základe agregačnej funkcie COUNT spočítame záznamy pre cudzí kľúč záznamu zraniteľnosti odkazujúci na whois záznam. Čím je tento poćet vyšší, tým viac záznamov o zraniteľnostiach pre konkrétny whois záznam existuje. Netreba zabudnúť spojiť tabuľku so zraniteľnosťami a whois tabuľku so záznamami LEFT JOINOM. Opäť je potrebný výber práve jedného záznamu pomocou LIMIT 1. Chceme najvyššiu hodnotu preto zoradíme výsledky zostupne pomocou ORDER BY count DESC, kde count je agregovaný počet pre každú jedinečnú hodnotu cudzieho kľúča, respektíve identifikátor whois zánamu.

11. Navrhnutý agregačný dopyt skombinujte pridaním bodkočiarky za výraz 1=1 a jeho doplnením za túto bodkočiarku.
    V tomto kroku by sme mali mať:
    a%' OR 1=1; SELECT COUNT(vulnerabilities.reference_record_id) AS count, whois AS whois FROM whois LEFT JOIN vulnerabilities ON whois.id = vulnerabilities.reference_record_id GROUP BY whois.id ORDER BY count DESC LIMIT 1 --'
    Skúste tento výraz vložiť do vyhľadávacieho okno.
12. Dostali ste chybovú hlášku, kde sa program sťažuje, že niektorá hodnota je undefined. Pri simulovaní situácie s použitím podobnej node js aplikácie alebo po hlbšej úvahe by ste mohli zistiť, že hodnoty sa nevrátia ako slovník.



13. **Obrázok 8: Chyba pri vyskúšaní pripravenej injekcie**



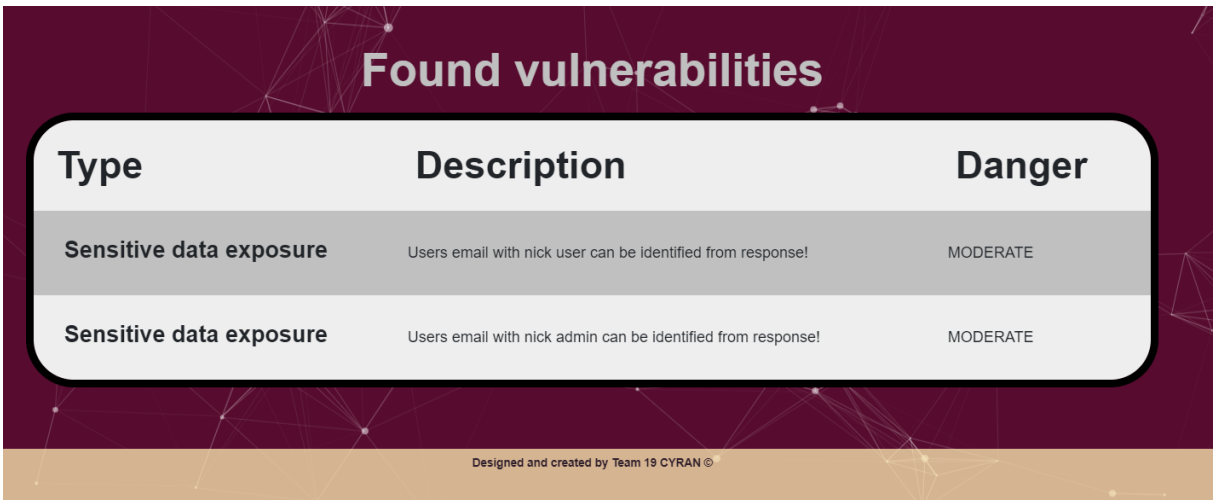
Obrázok 9: Hodnoty záznamu sa nevrátia ako slovník

Obrázok 10: Podoba dát keby boli slovníkom

14. Upravte preto príkaz tak, aby bol vrátený len id hľadaného záznamu teraz už bez ďalších JOIN operácií, tak aby vrátilo slovník. Výraz by mal fungovať. Postup je nasledovný.

V predchádzajúcom agregačnom dopyte zmeňte vrátený výsledok z whois záznamu len na id whois záznamu:
SELECT
COUNT (vulnerabilities.reference_record_id) AS count, whois**.id** AS **ww**
FROM whois
LEFT JOIN vulnerabilities ON whois.id = vulnerabilities.reference_record_id
GROUP BY whois.id
ORDER BY count DESC
LIMIT 1
Celý výraz po vložení vyzerá nasledovne:
a%' OR 1=1; SELECT COUNT(vulnerabilities.reference_record_id) AS count, whois**.id** AS **ww** FROM whois LEFT JOIN vulnerabilities ON whois.id = vulnerabilities.reference_record_id GROUP BY whois.id ORDER BY count DESC LIMIT 1 --'

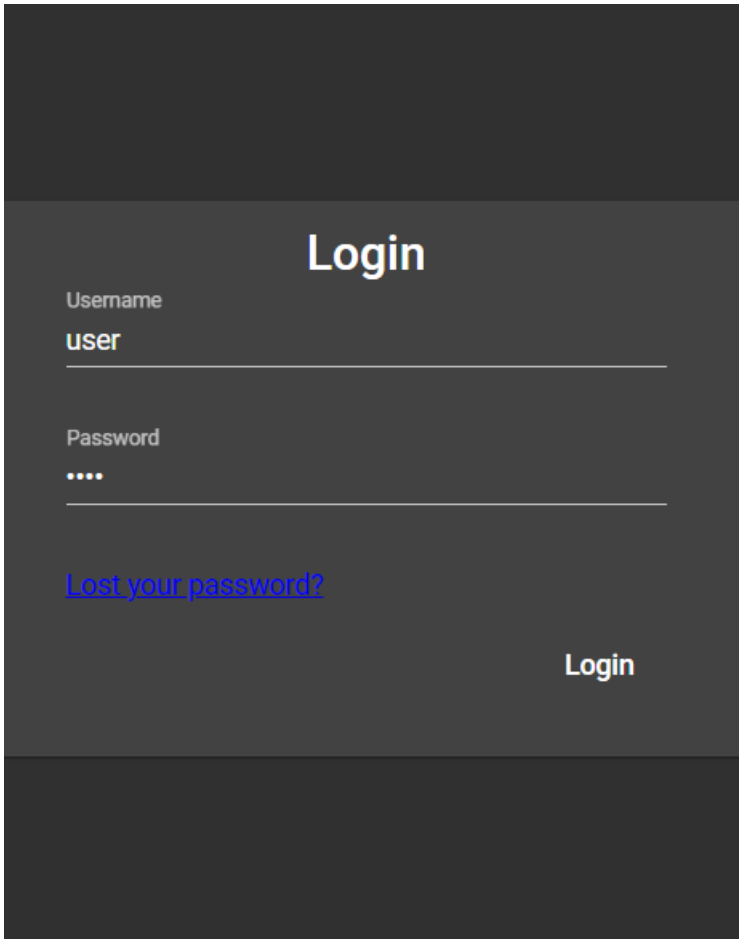
15. Pridajte ďalší SELECT do tohto agregovaného dopytu dopytujúceho sa do tabuľy whois po zázname na základe získaného id z dopytu:
a%' OR 1=1; **SELECT * FROM whois, (**SELECT COUNT(vulnerabilities.reference_record_id) AS count, whois.id AS ww FROM whois LEFT JOIN vulnerabilities ON whois.id = vulnerabilities.reference_record_id GROUP BY whois.id ORDER BY count DESC LIMIT 1**) ww WHERE ww = whois.id** LIMIT 1 --'
16. Následne ho overte pri vyhľadávaní. Získali ste záznam s dvomi zraniteľnosťami. Asi nie je prekvapením, že odkazujú na security eshop. Zo záznamov ste sa mohli dozvedieť mená dvoch významných používateľov eshopu, ktoré sa zídu v ďalších scenároch.



Obrázok 11: Získané zraniteľnosti pre doménu s najväčším počtom zraniteľností

Prelamovanie hesiel

Jeden z pracovníkov obchodu má nastavené uhádnuteľné slabé heslo. Princípom tohto scenára je zistiť toto heslo skúšaním rôznych hesiel pre používateľov pomocou ľubovoľného nástroja. Musí to ale realizovať prostredníctvom rozhrania pre Angulár. Stačí ak vyskúša jednoduché heslá ručne. Rovnako si môže zistiť hash hesla vytvorený bcryp-tom vrátený do Anguláru pre overenie. Ten môže získať sledovaním premávky. Následne by mohol skúšať známe heslá a porovnávať vytvorené hashe s hashmi vytvorenými pre reťazce na zozname. Túto časť môže realizovať aj offline. Meno a heslo sú rovnaké, a to user a user. Malo by ich preto byť jednoduché zistiť. Často sú na zozname najpoužívanejších hesiel.

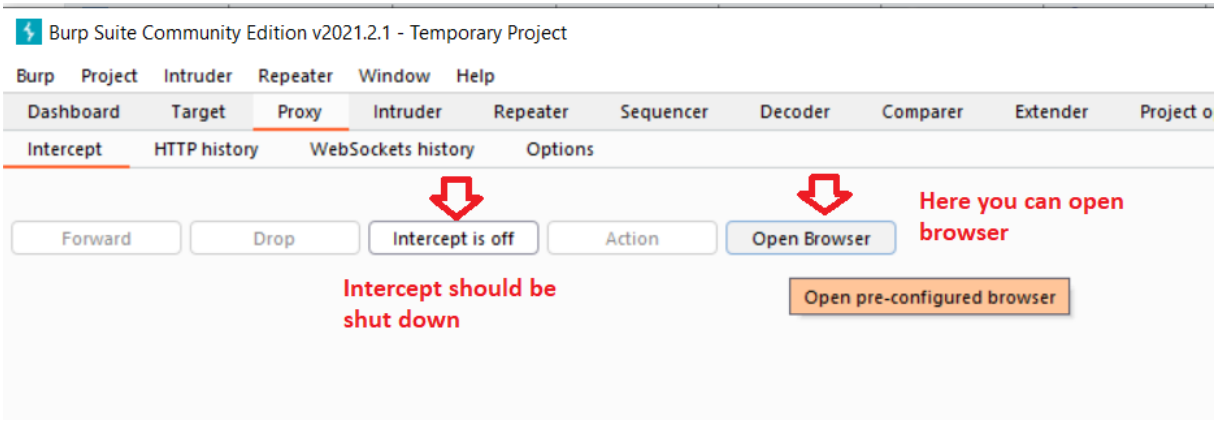


Obrázok 12: Aplikovanie jednoduchého hesla user

Prelamovanie hesiel slovníkovým útokom

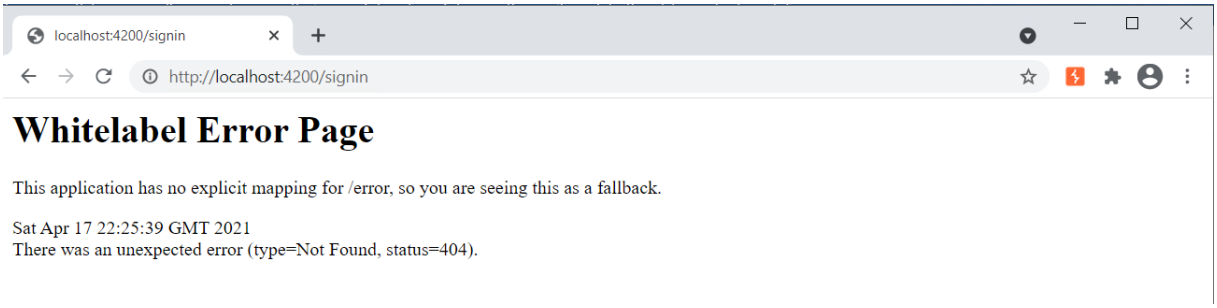
Útočník môže zrealizovať slovníkový útok na základe získaných informácií z login stránky. K užitočným informáciám sa dostanete na základe nasledujúceho postupu:

17. Po zapnutí burpsuitu a prejdite do kolónky proxy.
18. Vypnite intercept v rozkliknutom menu BurpSuite.
19. Otvorte si prehliadač kliknutím na open browser.



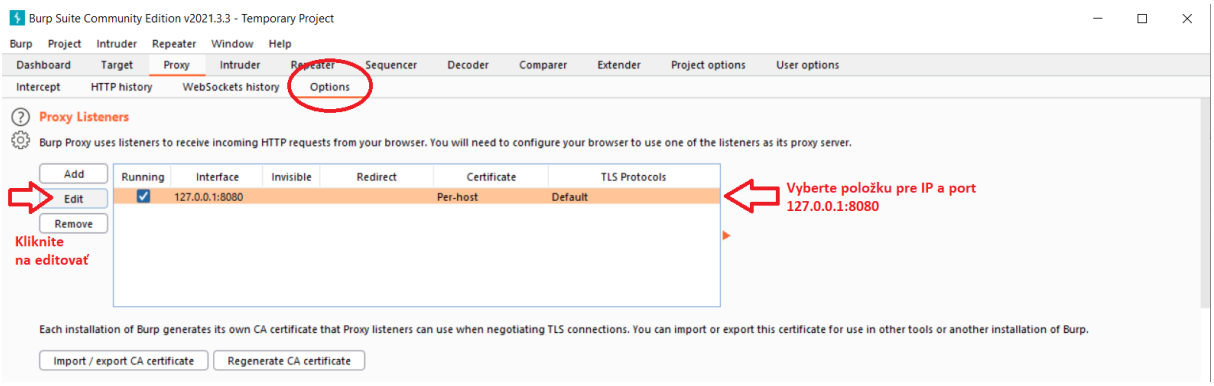
Obrázok 13: Otvorenie prehliadača a interceptor v BurpSuite

20. Prejdite na stránku http://localhost:4200/signin.
21. Ak sa vám zobrazí chyba ako na obrázku 5 postupujete podľa ďalších krokov. Ak vám všetko funguje pokračujte krokom 8.



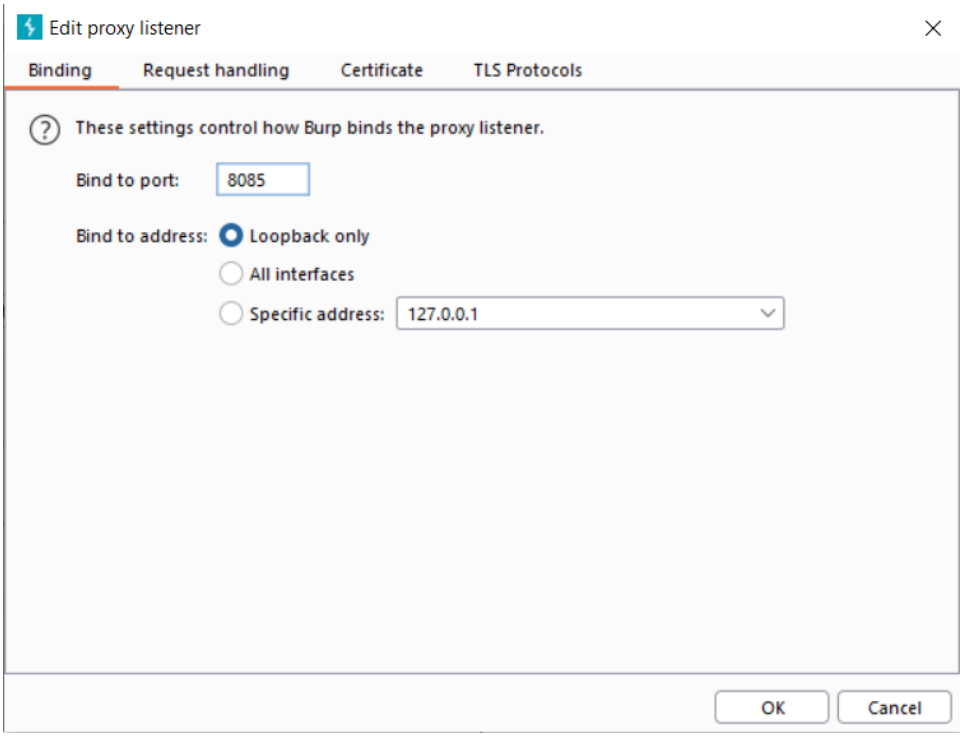
Obrázok 14: Chyba v zabudovanom prehliadači pre BurpSuite

22. V burpsuite si na lište vo vybranej kolónke proxy rozkliknite tab Options. V časti Listeners zvoľte záznam pre localhost s IP 127.0.0.1 a kliknite na editovať.



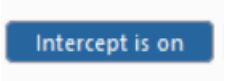
Obrázok 15: Zmena nastavenia proxy

23. Následne zmeňte port z 8080 napríklad na 8085. Overte či obsah v prehliadači funguje. Ak áno pokračujte nasledovným bodom.



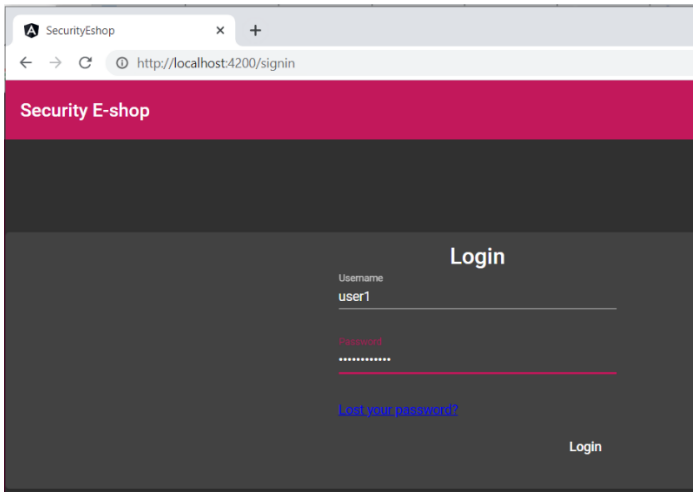
Obrázok 16: Zmena portu

24. Zapnite intercept na tej istej položke v menu BurpSuite.



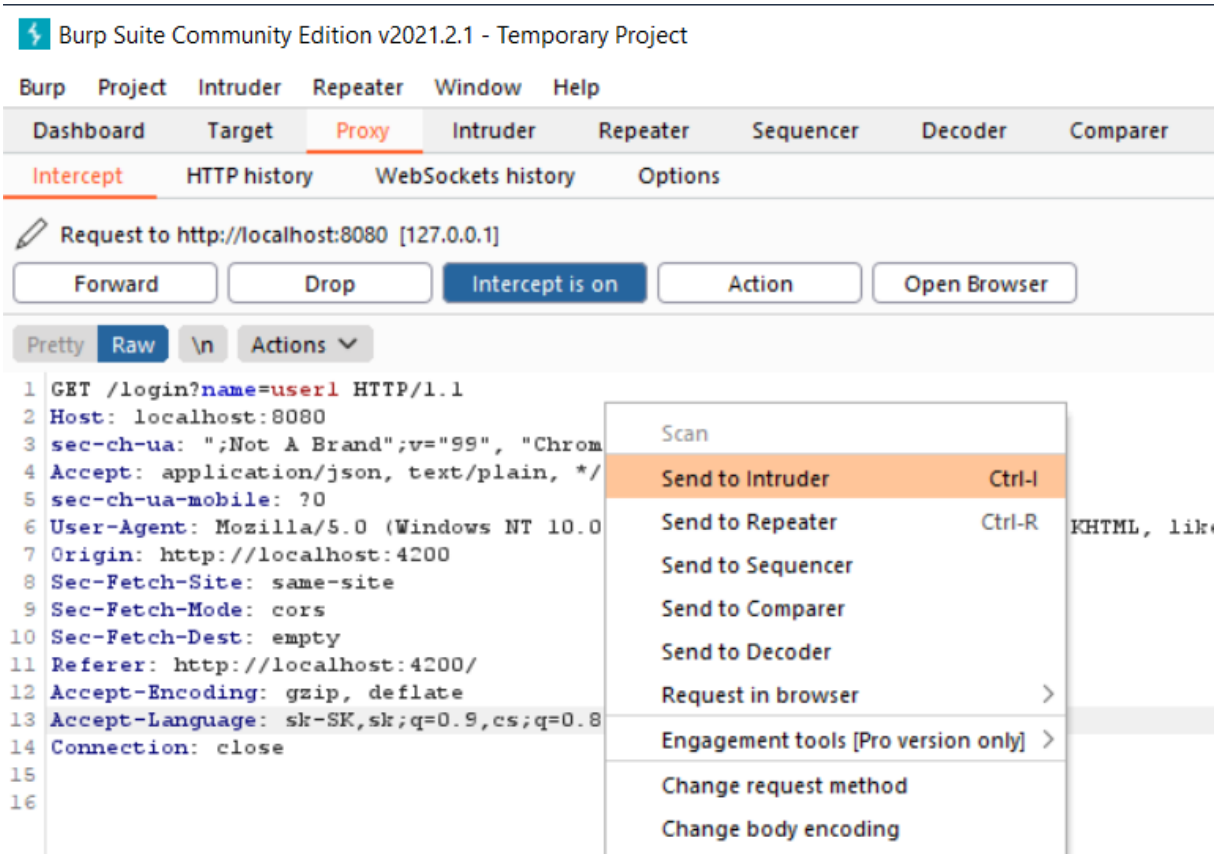
Obrázok 17: Zapnutie interceptora

25. Pokúste sa prihlásiť s ľubovoľným menom a heslom.



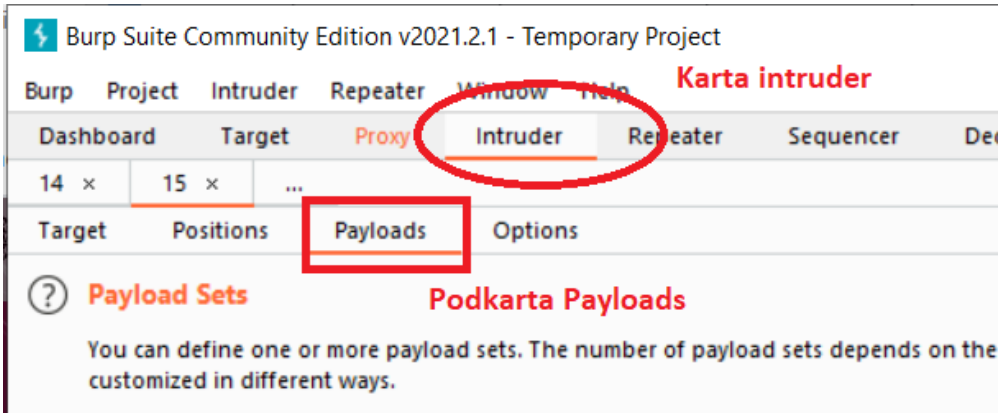
Obrázok 18: Pokus o prihlásenie v stavanom prehlaidači BurpSuitu

26. Následne sa prepnite do burpsuitu, kde sa zobrazí informácia o dopyte,
27. Zobrazte menu kliknutím ľavým tlačidlom myši do prostriedku informácií o dopyte.
28. Z menu vyberte položku “Send to Intruder”.



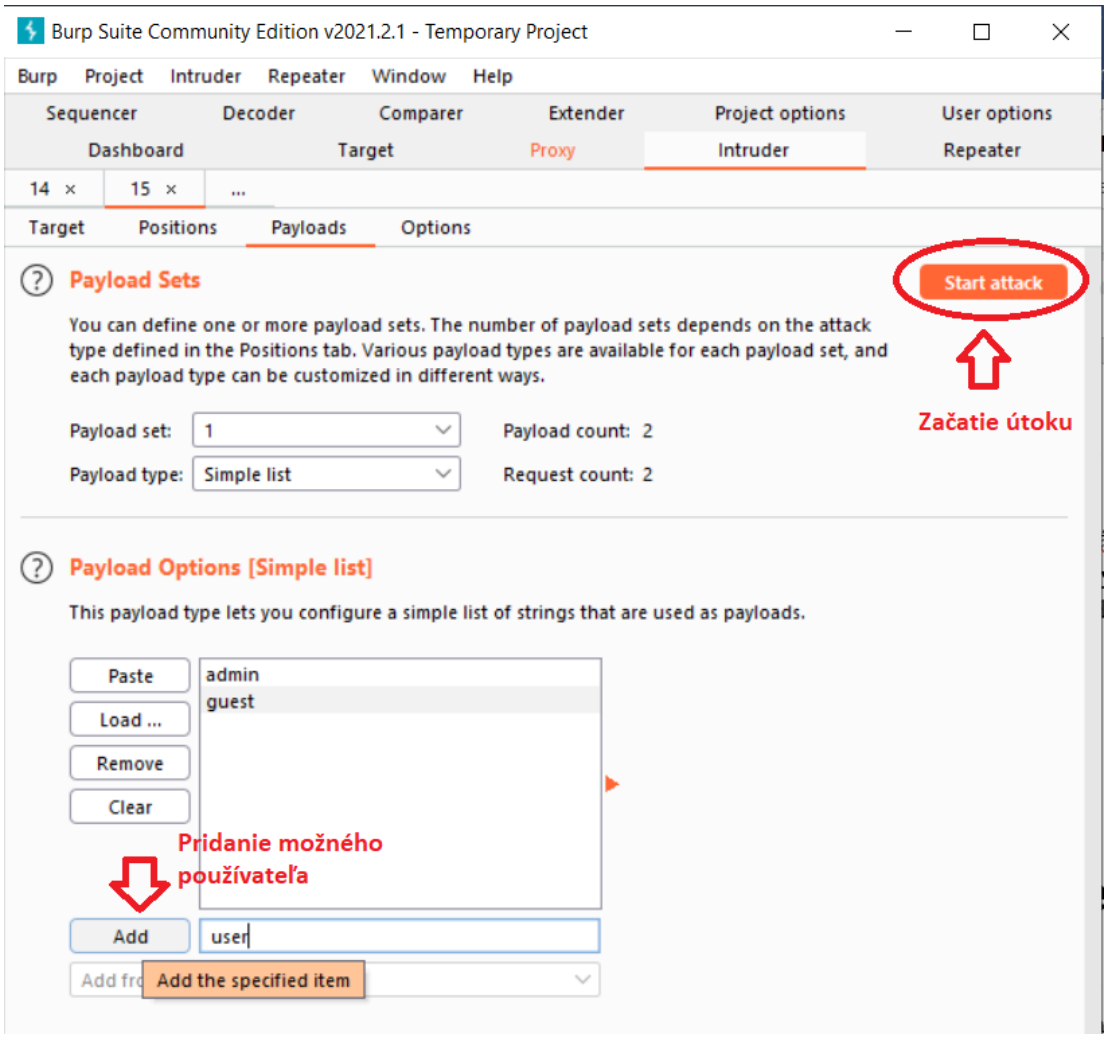
Obrázok 19: Odoslanie requestu do intrudera

29. Následne kliknite na otvorenú položku Intruder-a.
30. Rozkliknite podmenu Payloads v položke Intruder-a.



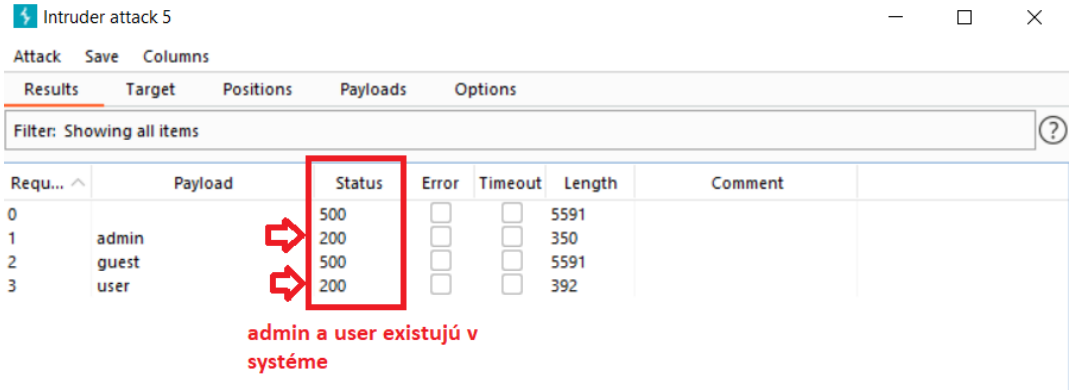
Obrázok 20: Presunutie sa na položku Payloads v Intruderovi

31. Vo vrchnej časti Payload Sets rozkliknutej karty v BurpSuite nechajte nastavené Payload set na 1 a Payload type na Simple list.
32. Nižšie v rozkliknutej karte nájdite časť Payload options a pomocou tlačítka Add pridajte niekoľko mien, ktoré by mohli byť potencionálni používatelia, pričom sa riadte častými názvami ako admin, user, guest a podobne.
33. Zvoľte položku "Start attack".



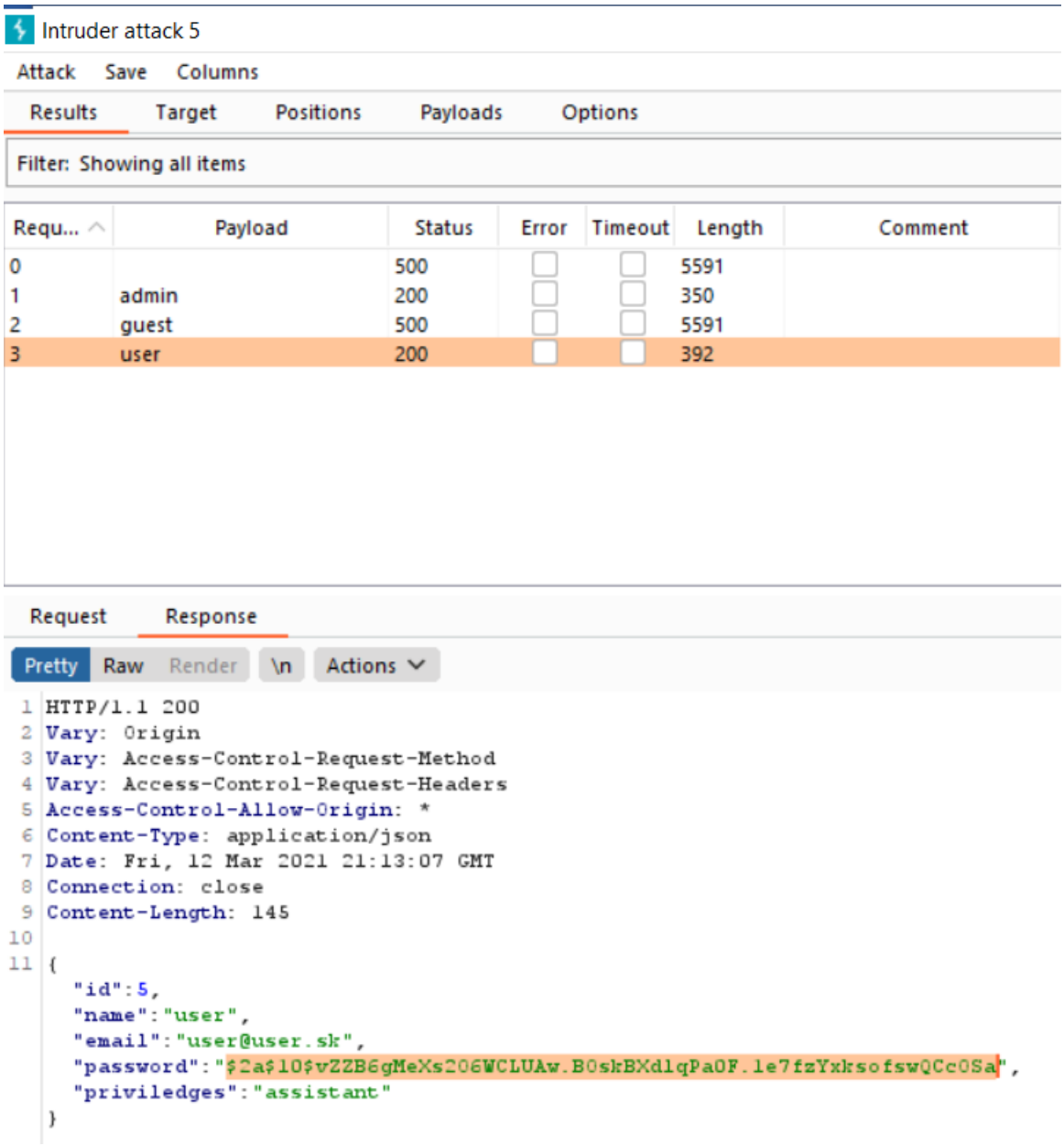
Obrázok 21: Zadanie zoznamu potencionálnych používateľov a začatie útoku

34. Otvorí sa okno, v ktorom podľa vráteného statusu môžte zistíť, ktorí používatelia existujú v systéme.
35. Kliknite na jeden z riadkov, ktorý má status 200.
36. Prepnite sa na kartu Response, v okne ktoré sa zobrazí nižšie.
37. Môžte zistiť, že aplikácia dostala heslo spolu s emalom a roľou používateľa. Pre admina zistíte, že jeho heslo nie je zahešované. Naopak pre používateľa zistíte, že jeho heslo je hash. Systém teda heslá šifruje, inak by sme sa prihlásili pomocou získaného hesla. Účet admina bude nejak zablokovaný. So získaných informácií zistíte, že používateľ user je v skutočnosti asistent. Skúsime preto v nasledujúcej časti zistiť jeho heslo.



Obrázok 22: Zistenie existujúcich používateľov v systéme

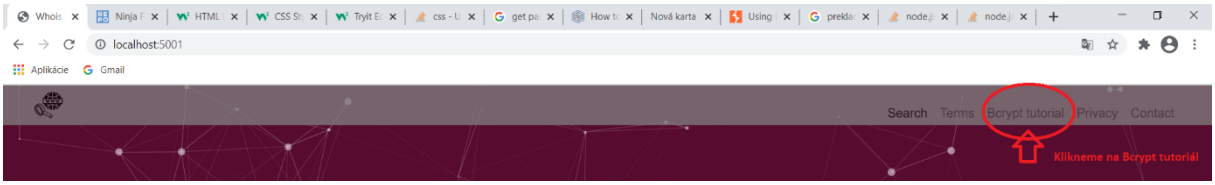
38. Skopírujte heslo usera, ktorý je asistent.



Obrázok 23: Získanie zašifrovaného hesla asistenta s používateľským menom user

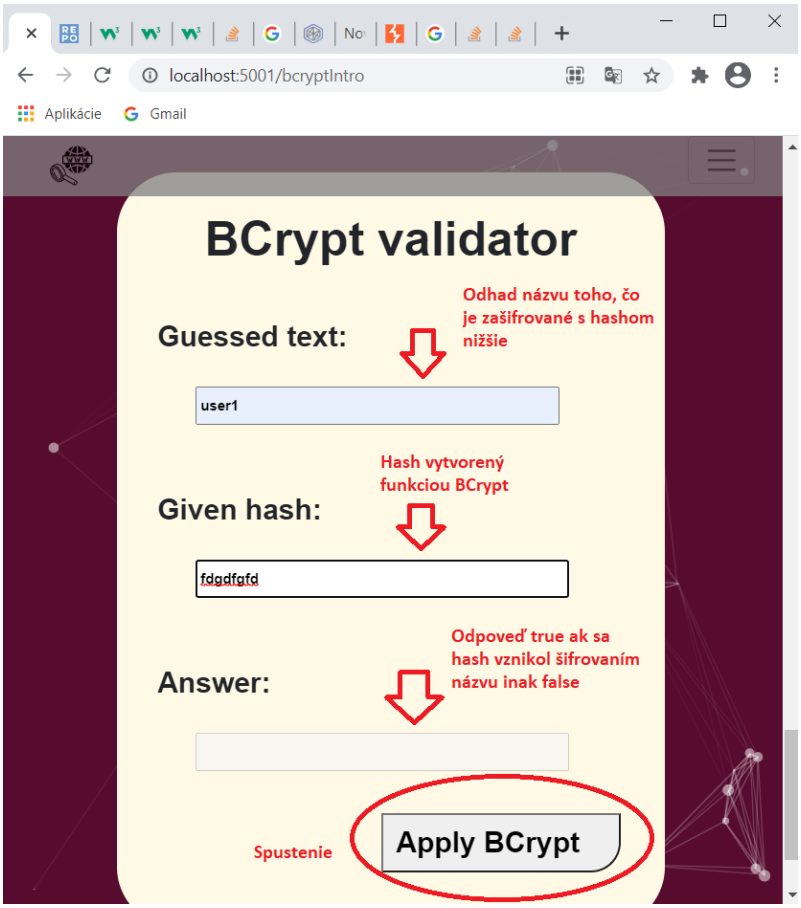
Získali ste heslo, ale je ho potrebné ešte prelomiť. Z Whois aplikácie, prezretím zdrojového kódu projektu, alebo vďaka nejakej nápovede by ste mali vedieť, že na šifrovanie bol použitý bcrypt v javascripte. Je teda potrebné overiť množinu možných hesiel voči tomuto hashu. Skúsite ich preto overiť použitím služby tutorial aplikácie vysvetľujúcej základy bcryptu. Postup je nasledovný:

1. Zapnite BurpSuite a znova sa prepneme do kolónky proxy.
2. Vypnite interceptor a zapnite prehliadač, ktorý má BurpSuite.
3. Prejdite na adresu http://localhost:5001/.
4. V ľavom hornom rohu kliknite na položku v menu s názvom “Bcrypt tutorial”.



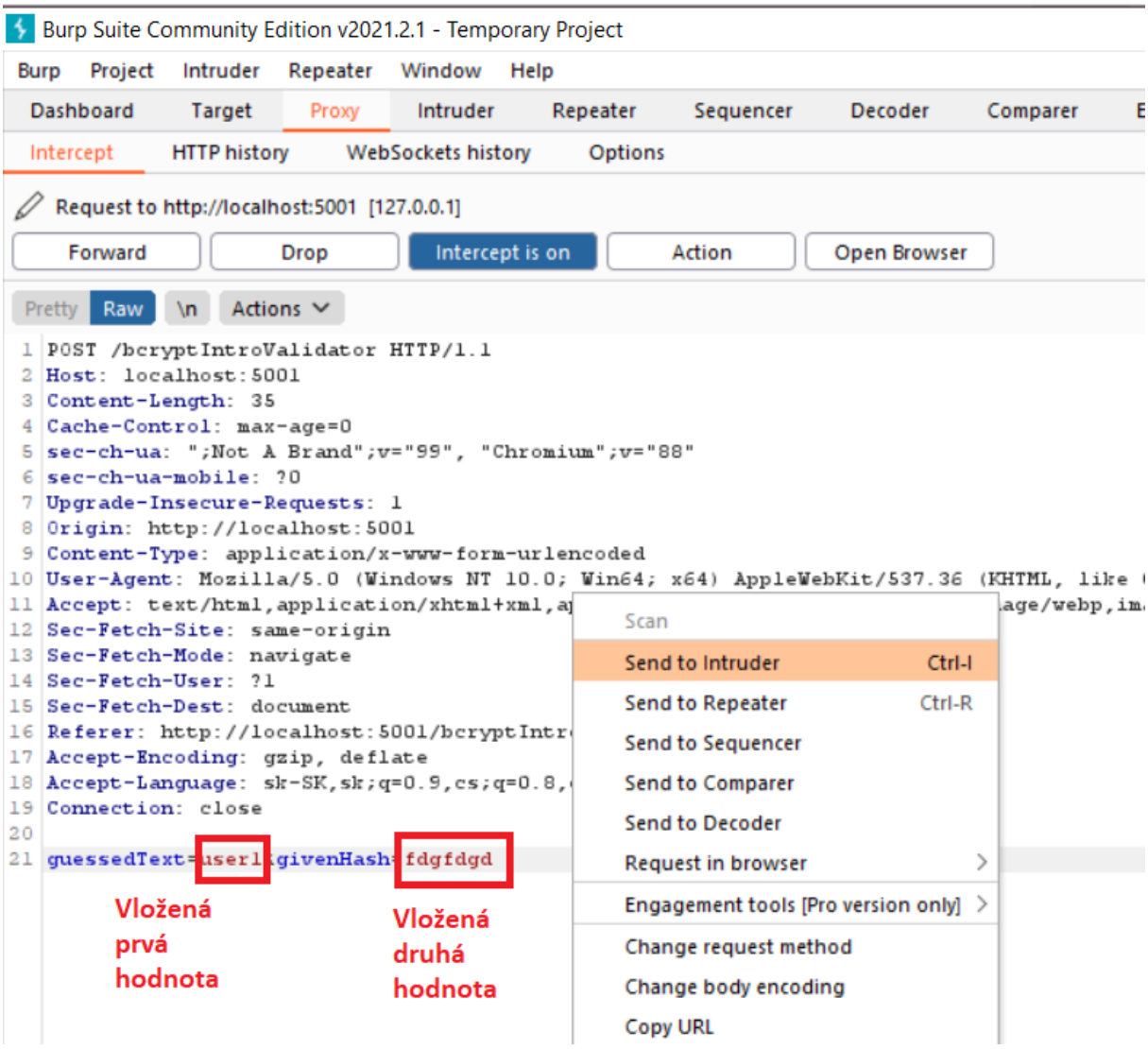
Obrázok 24: Vyhľadanie stránky s tutoriálom pre Bcrypt

5. Preskrolujte na službu s názvom BCrypt validator.



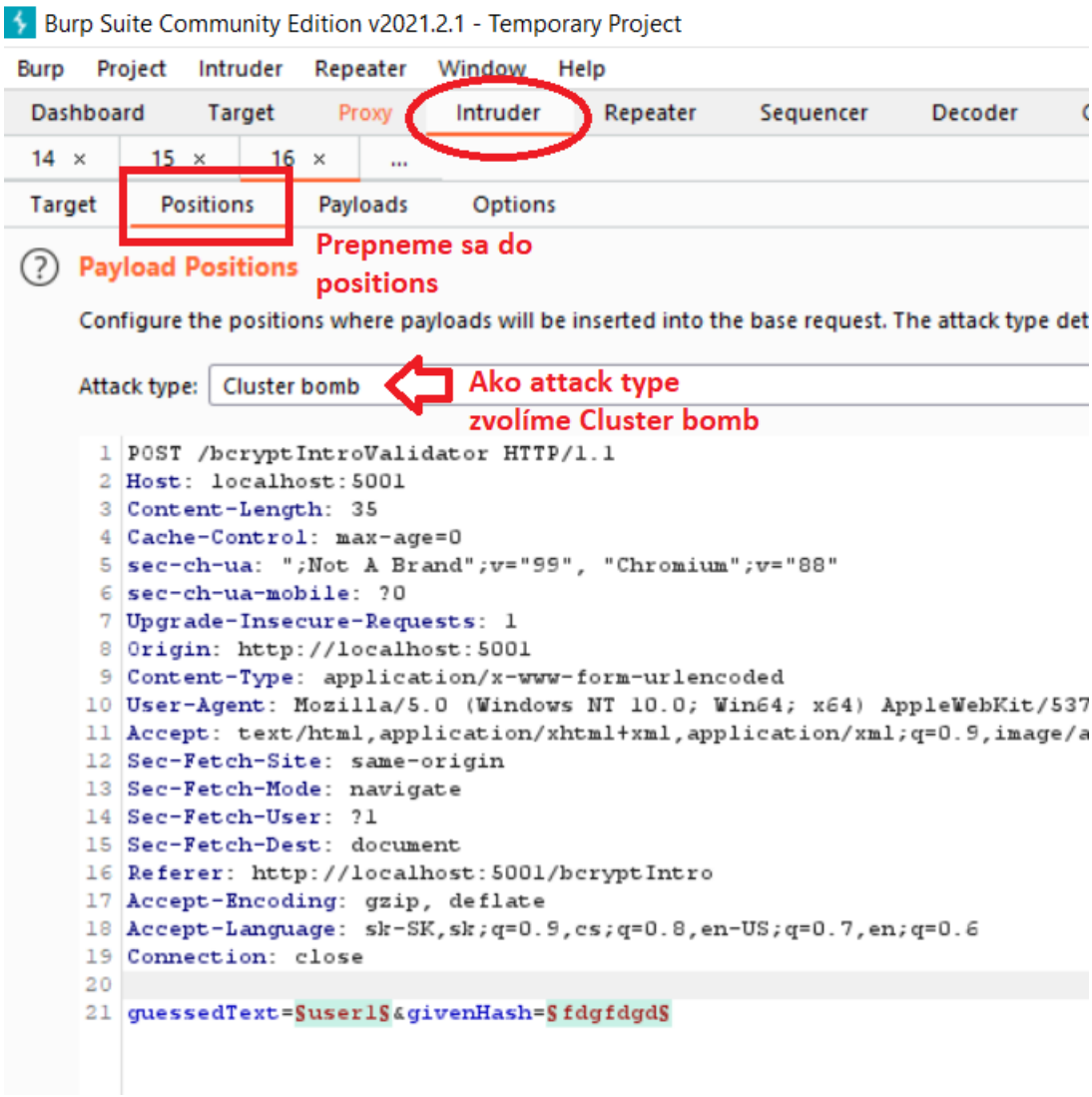
Obrázok 25: Vyvolanie služby pre zistenie či hash vznikol šifrovaním odhadovaného textu

6. Vložte nejaký text do polí Guesed text a Given hash.
7. Zapnite interceptor v BurpSuite.
8. Opäť kliknite ľavým tlačidlom doprostred a v menu vyberte položku “Send to intruder”. V okne ste si mohli všimnúť odosielané hodnoty.



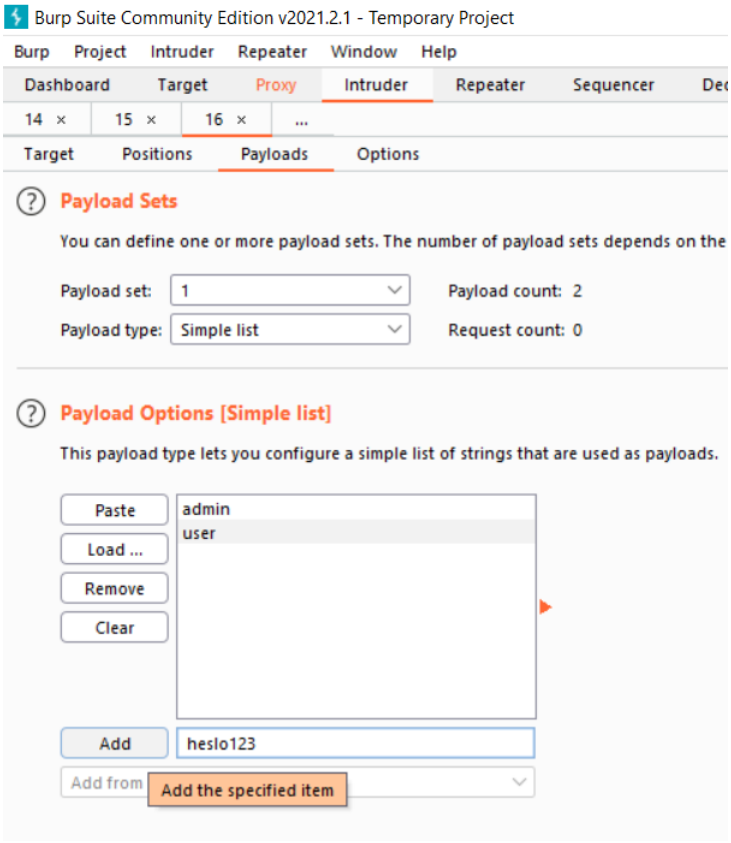
Obrázok 26: Zachytená odoslaná žiadosť na server a odoslanie do intrudera

9. V karte Intruder sa prepnite do podmenu Positions.
10. Následne prenastavte Attack type na Cluster bomb.

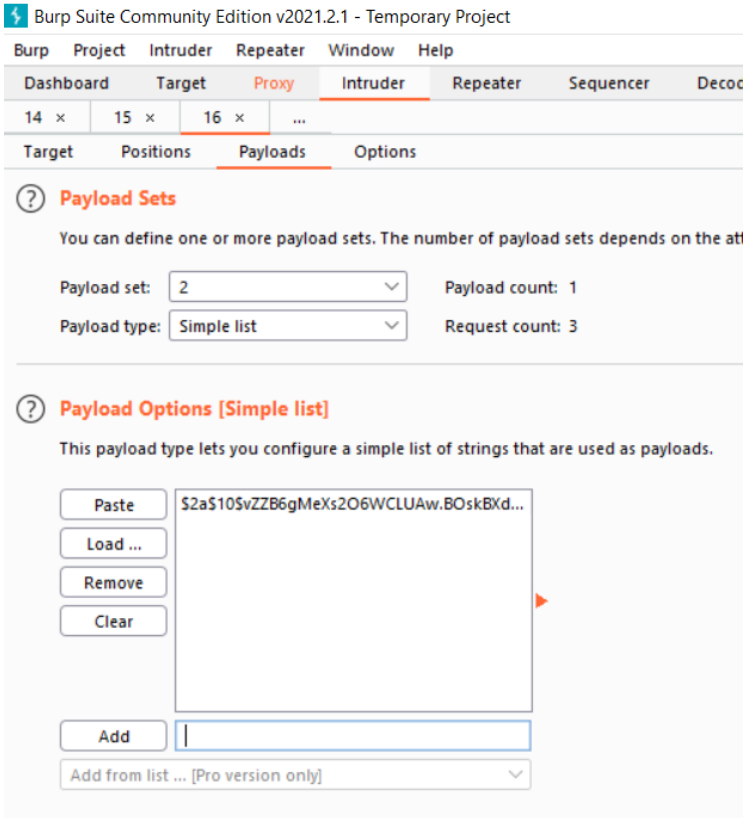


Obrázok 27: Nastavenie typu útoku na Cluster bomb

11. Následne sa prepnite do podmenu karty Intruder s názvom Payloads.
12. Nechajte opäť v prvej časti nastavený Payloads set na 1 a Payload type na “Simple list”. Môžte si všimnúť, že Payloads set je možné prenastaviť na 2. To je preto, že prvé je pre prvý parameter requestu, odhadovaný text a druhý je pre jednu z jeho šifrovaných podôb.
13. Pridajte nižšie v časti Payload options Vami odhadované heslá, opäť také, ktoré sú často používané. Napríklad najčastejšie také, ktoré sú zhodné aj s menom používateľa. Napríklad admin, user, heslo123 a podobne.
14. Prepnite Payloads set v hornej časti s názvom Payloads set na 2. Teraz nastavuje šifrovanú podobu nejakého textu.
15. Opäť v časti Payload options pridajte skopírovaný šifrovaný text používateľa user, ktorý je asistentom.
16. Kliknite na tlačidlo Start attack v pravom hornom rohu.
17. Zobrazil sa Vám zoznam s výsledkami. Keďže služba vracia hodnotu 500, a to v prípade, že hash nebol vytvorený šifrovaním zadaného odhadovaného textu, stačí pozrieť hodnotu výsledného statusu.

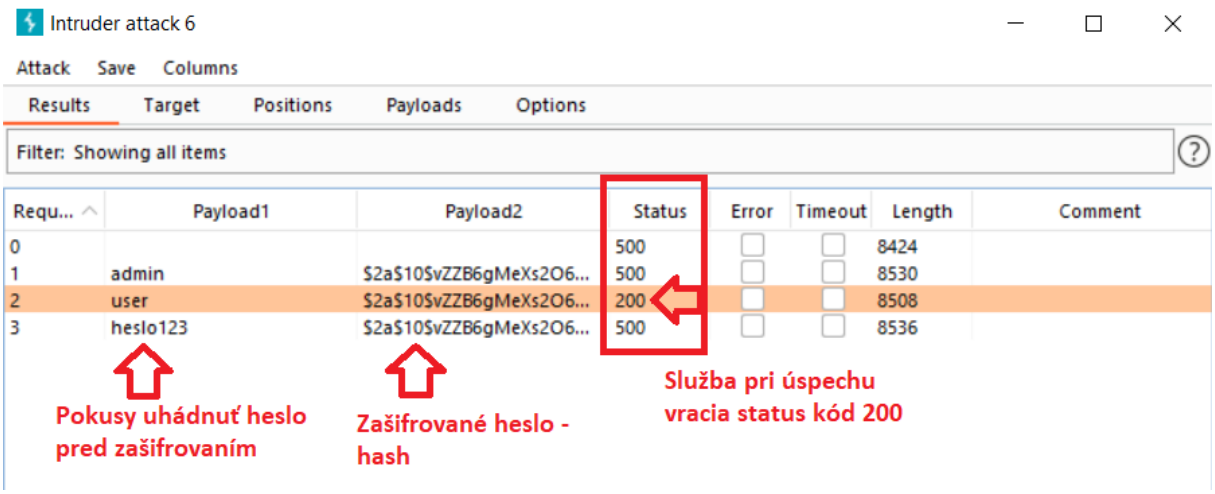


Obrázok 28: Zadanie potencionálnych odhadovaných hesiel – nešifrovaných



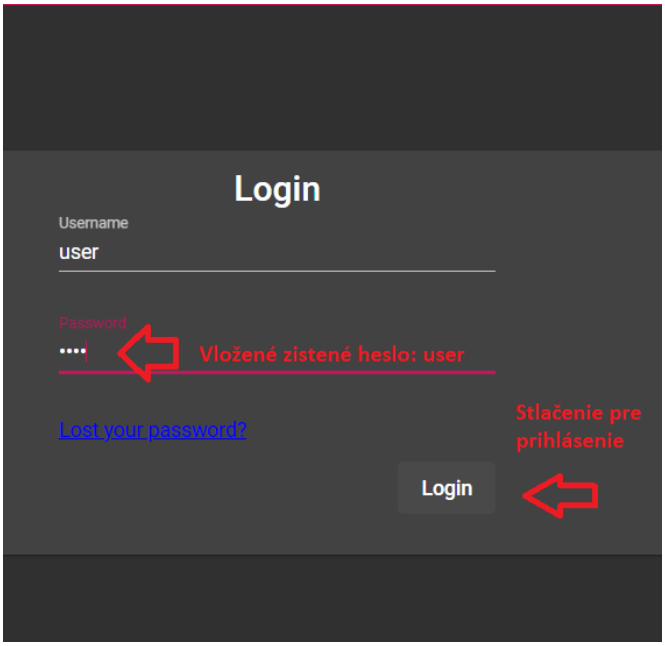
Obrázok 29: Pridanie šifrovanej podoby hesla pre druhý parameter

18. V tabuľke nájdite riadok/riadky s hodnotou status kódu 200. Pozrite sa na Payload číslo 1. Vidíte aké je heslo, ktoré po zašifrovaní môže nadobúdať hash v stĺpci Payload číslo 2. Skopírujte si heslo zo stĺpca Payload číslo 1.

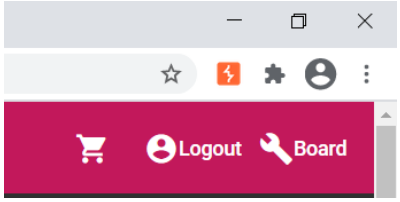


Obrázok 30: Získanie hesla pred zašifrovaním

19. Následne heslo spolu s používateľským menom overte prihlásením sa. Môžte si overiť, že používateľ má naozaj práva asistenta podľa položky Board v hornom menu.



Obrázok 31: Overenie získaného hesla pre používateľa user prihlásením

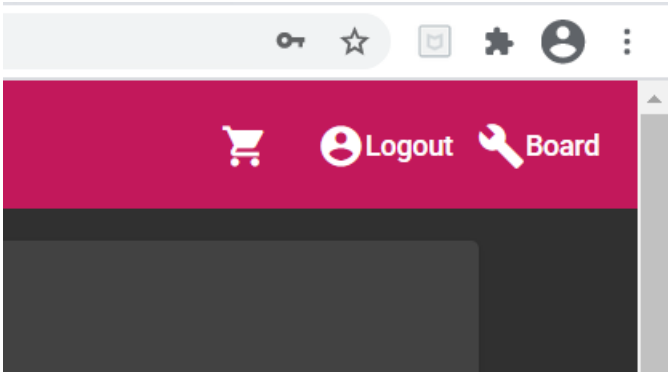


Obrázok 32: Overenie role asistenta

Použitie SQL injekcie

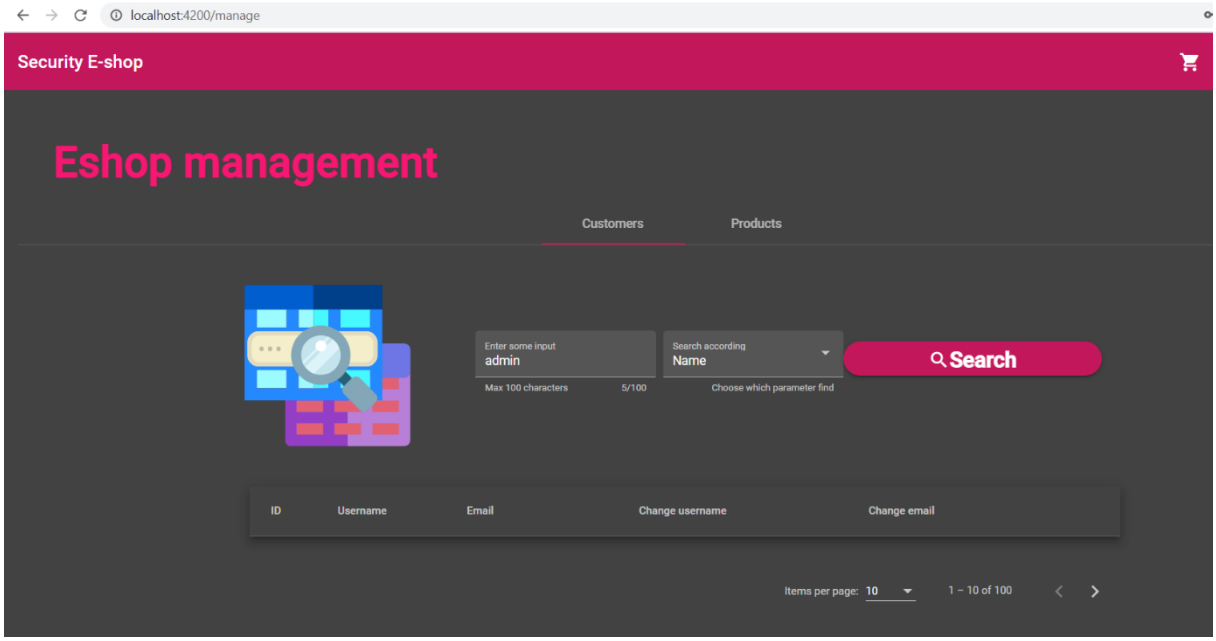
Útočník pri prelamovaní hesiel sa bol schopný dostať do role pracovníka v obchode. Následne má prístup k používateľským emailom a menám. Jeho úlohou bude ale vyhľadať admina, ktorý sa nezobrazuje. Použije SQL injekciu. V tejto časti ponúkame postup pri scenári aplikovania SQL injekcie.

1. Kliknite na tlačidlo Board v pravom hornom rohu potom, čo ste prihlásený ako pracovník v obchode.



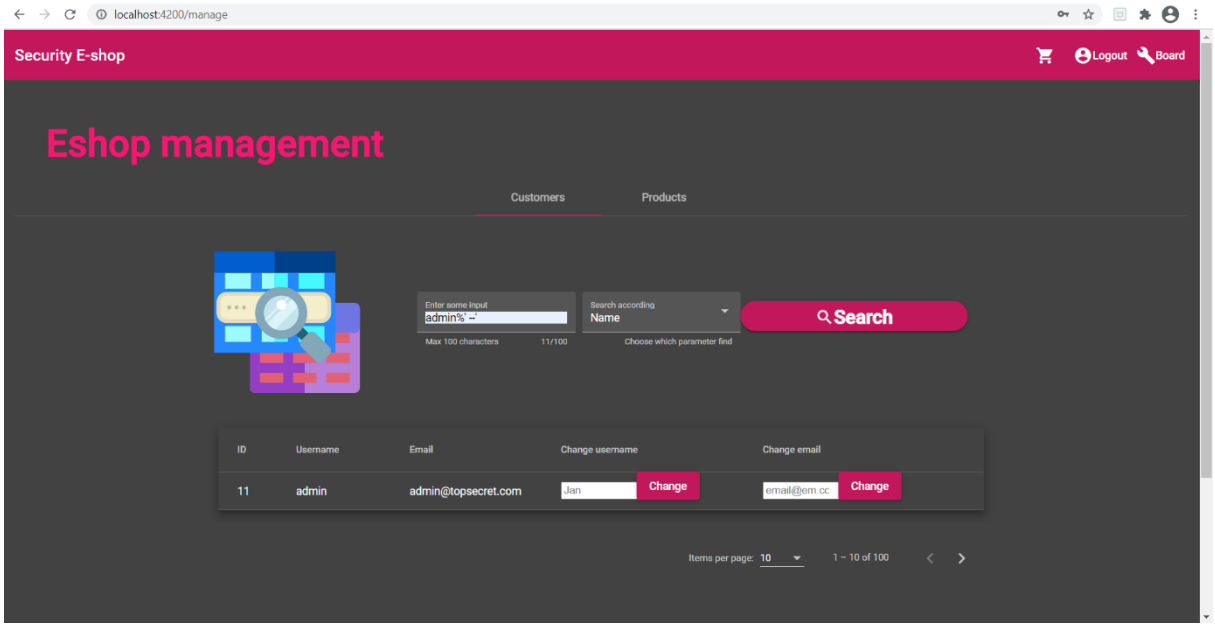
Obrázok 33: Pracovník v obchode má prístup k tabuli používateľov

2. V časti Customers sa pokúste vyhľadať používateľa s menom admin.



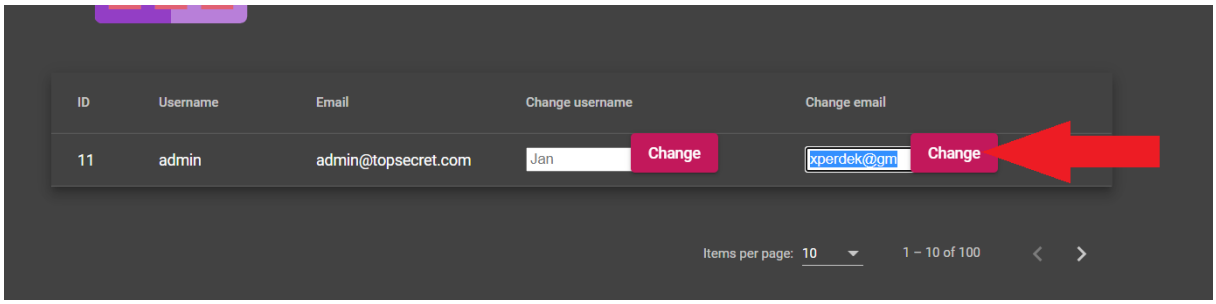
Obrázok 34: Pokus vyhľadať používateľa s menom admin

3. Skúste použiť SQL Injekciu pre používateľa admin, tým že necháte výraz admin vyhľadať a zároveň odignorovať zvyšnú časť výrazu.



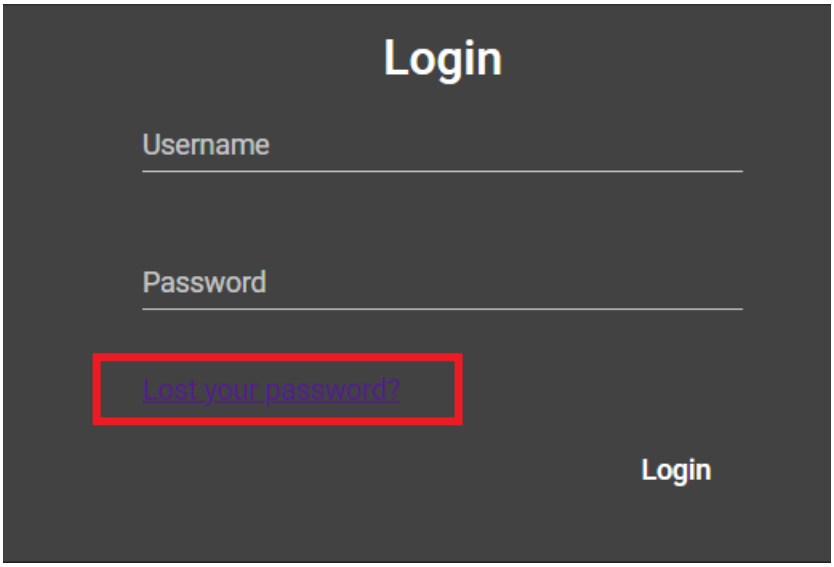
Obrázok 35: Použitie SQL Injekcie pre vyhľadanie používateľa s menom admin

4. Zmeňte email používateľa admin na svoj. Pre unikátnosť emailov nesmie byť tento email už predtým použitý.



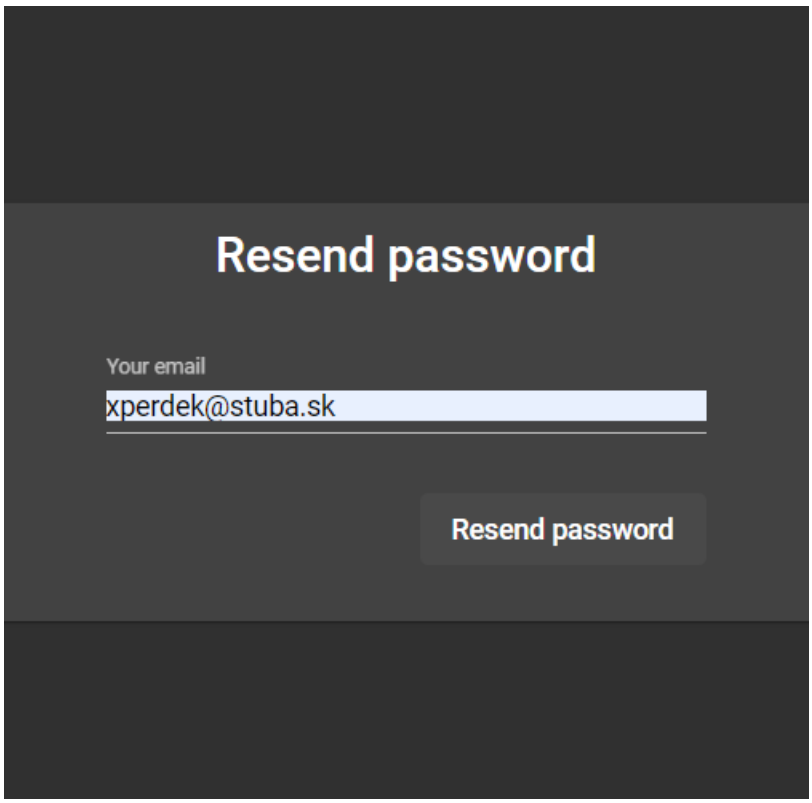
Obrázok 36: Zmena emailovej adresy používateľa admin na svoj vlastný

5. Odhláste sa kliknite na tlačidlo pre opätovné prihlásenie. Namiesto prihlásenia ale kliknite na odkaz Lost your password?



Obrázok 37: Prihlasovací formulár s odkazom na obnovu zabudnutého hesla

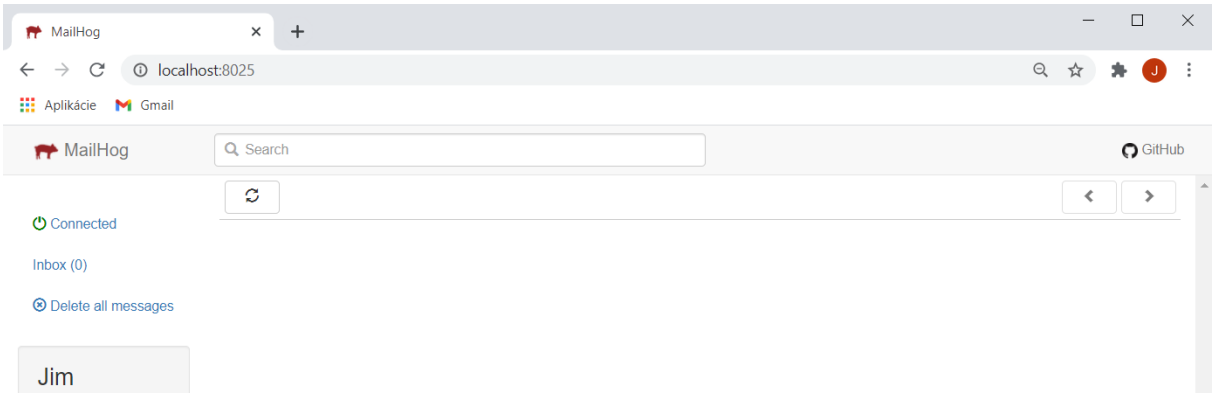
6. Na nasledujúcom formulári zadajte zmenený email a kliknite na tlačidlo Resend password.



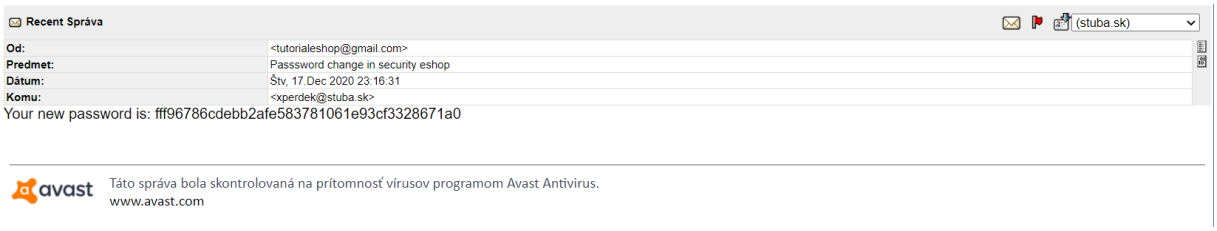
Obrázok 38: Formulár pre pregenerovanie nového hesla

7. Otvorte svojho emailového klienta a počkajte kým vám príde email z eshopu. Potom z neho získajte heslo. Ak používate aplikáciu nasadenú lokálne v prostredí docker, tak mail by mal byť doručený do MailHog aplikácie dostupnej na adrese localhost:8025 (kvôli zabezpečeniu emailového účtu nefunguje prihlásenie do mailu, keďže každý používateľ má iné zariadenie a Gmail prihlásenie zablokuje). Obsah mailu by mal byť identický ako v predchádzajúcom prípade.

Obrázok 39: Doručenie správy so zmeneným heslom

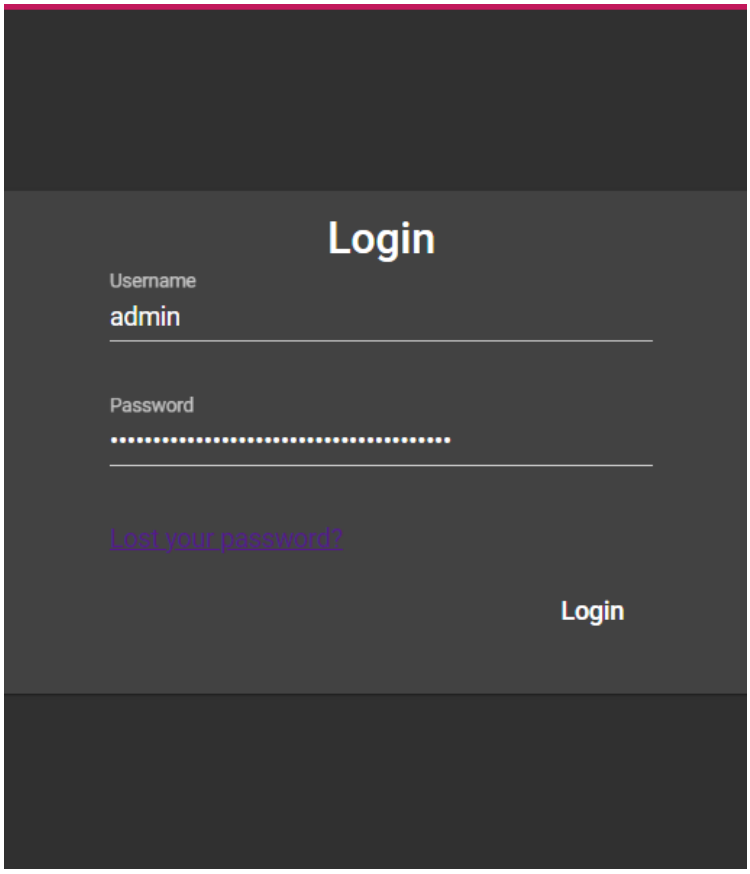


Obrázok 40: MailHog dostupný pre lokálne nasadenie



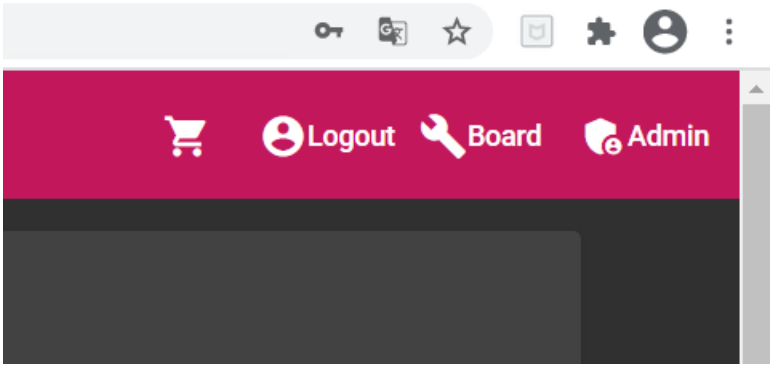
Obrázok 41: Zmenené heslo sa nachádza v správe

8. Prihláste sa pod menom admin a zadajte vygenerované heslo.

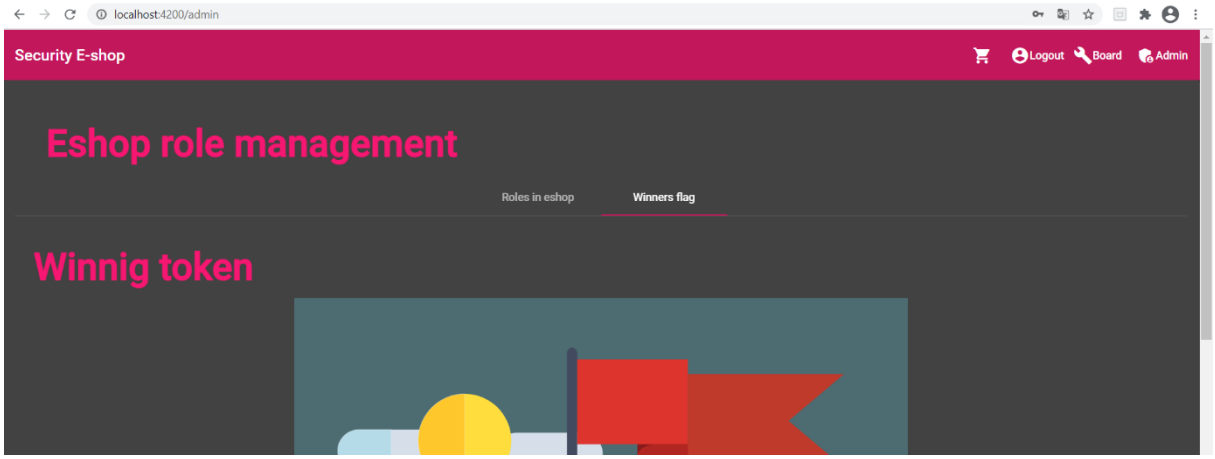


Obrázok 42: Vloženie zmenených údajov do formulára pre prihlásenie

9. Dostali ste sa do účtu, ktorý má najvyššie privilégium. Teraz môžete meniť privilégiá ostatných používateľov. Víťazný token/vlajku môžete nájsť v časti pre manažovanie rolí. Konečne je eshop dobytý!



Obrázok 43: Používateľ s privilégiom admin má vlastný ovládací panel

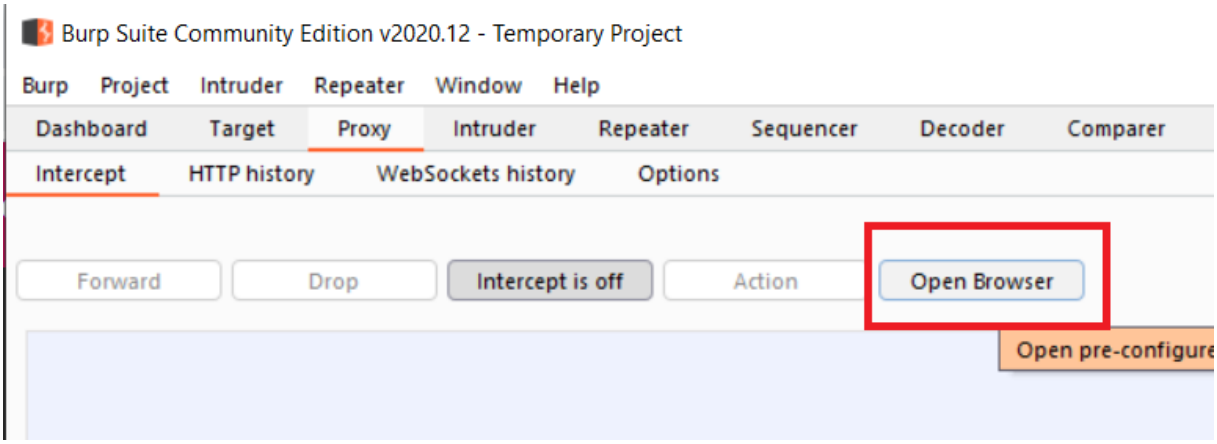


Obrázok 44: Prekliknutie sa na víťazný token

Ukradnutie produktu z eshopu

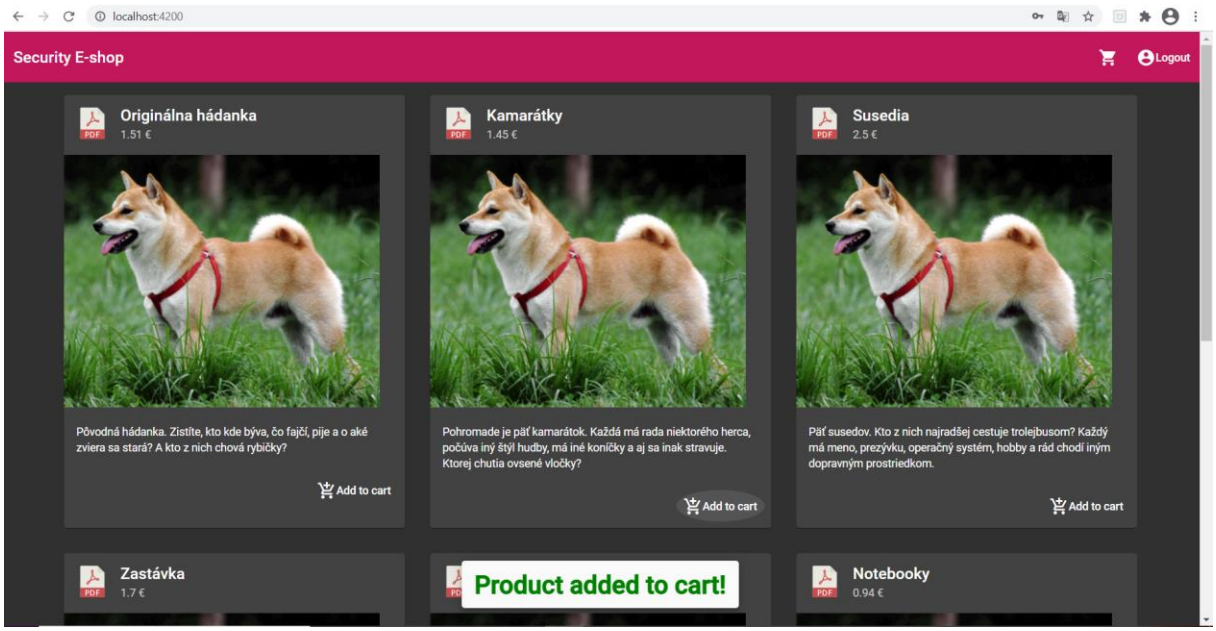
Útočník ukradne produkty z eshopu tým, že pošle vo formulári nulovú hodnotu. Najprv ale musí vytvoriť objednávku.

1. Otvorte program Burp Suite a prepnite sa na lištu Proxy. Následne otvorte prehliadač.



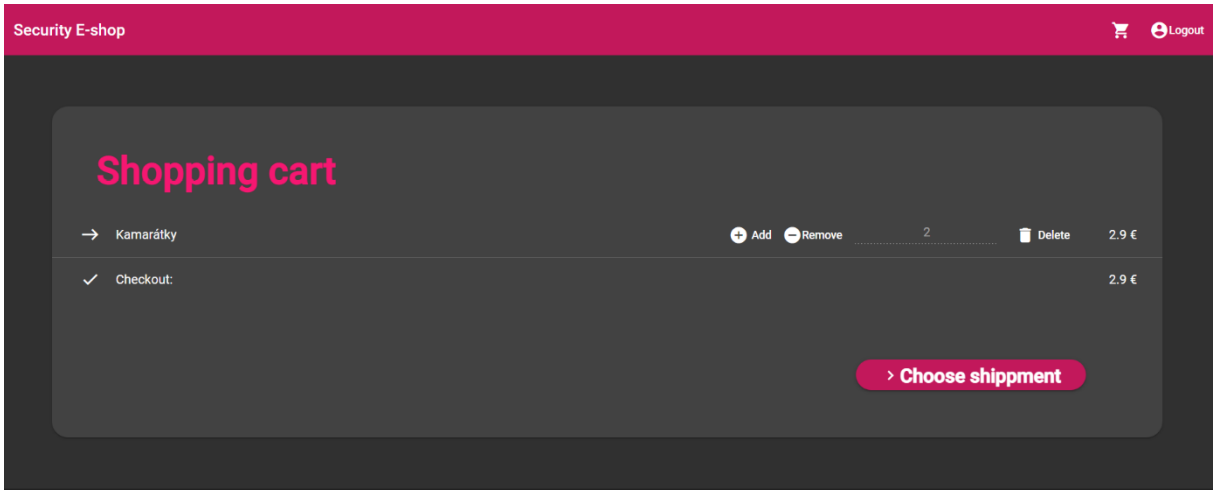
Obrázok 45: Zapnutie burpsuite a otvorenie vlastného prehliadača

2. Prihláste sa pod ľubovoľným používateľom a pridajte nejaký produkt do košíka.



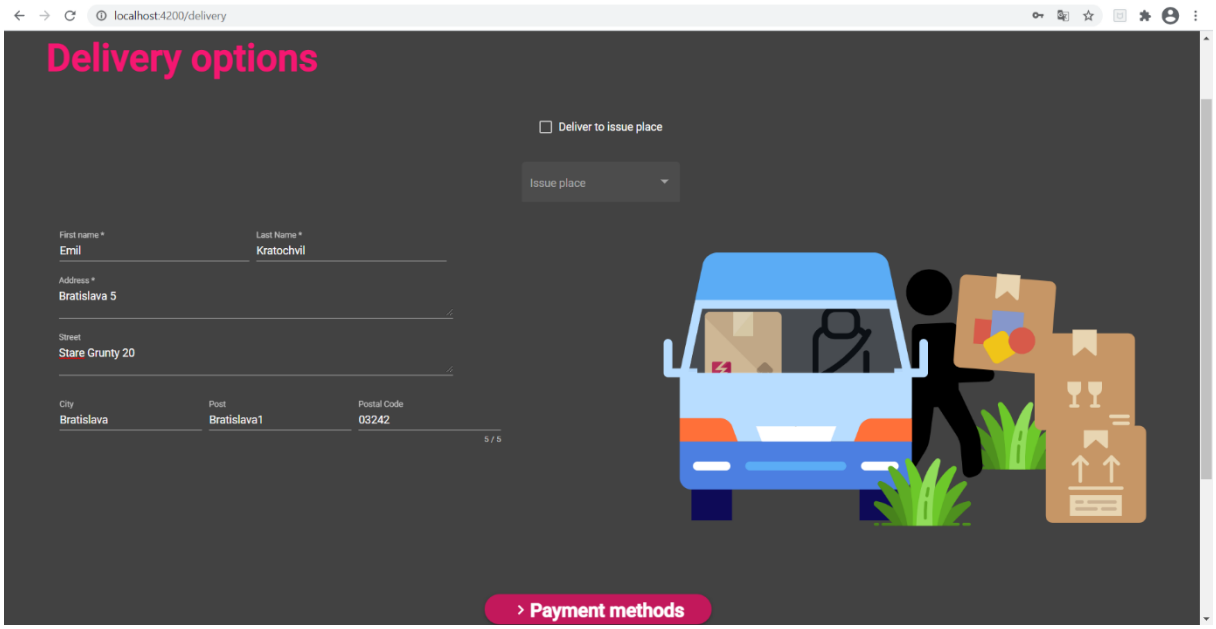
Obrázok 46: Pridanie produktu do košíka

3. Potvrďte produkty v košíku vybraním výberu spôsobu dodania stlačením na tlačidlo Choose shippment.



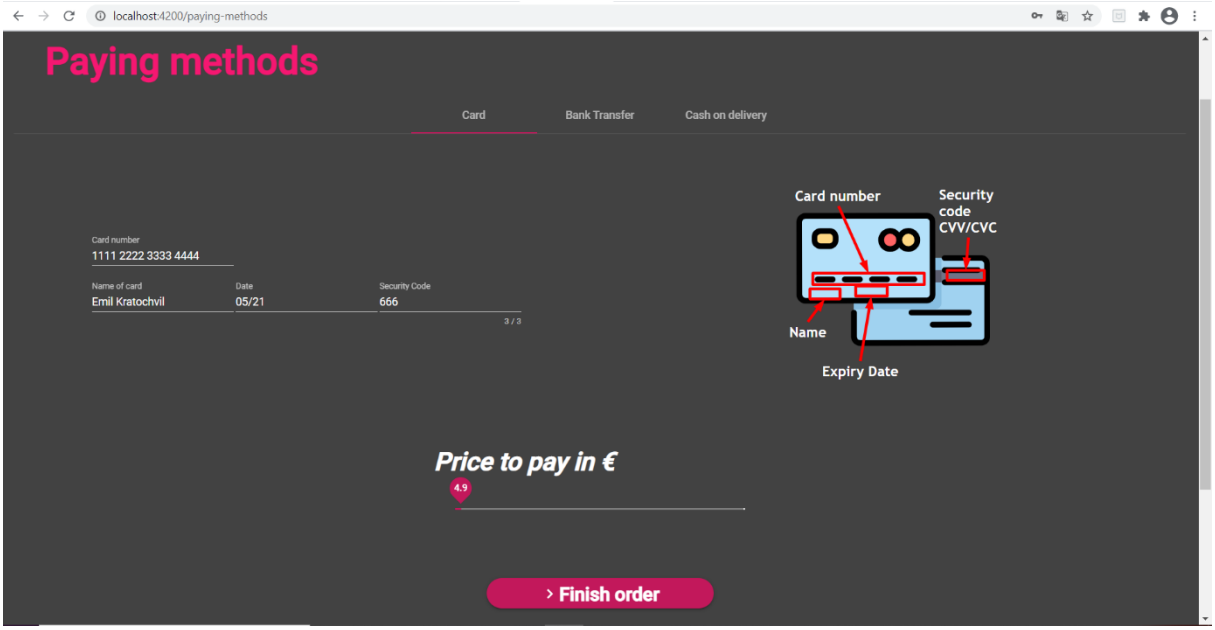
Obrázok 47: Potvrdenie produktov v košíku

4. Zadajte informácie o dodaní. Nejakú adresu a ďalšie potrebné údaje a potvrďte.



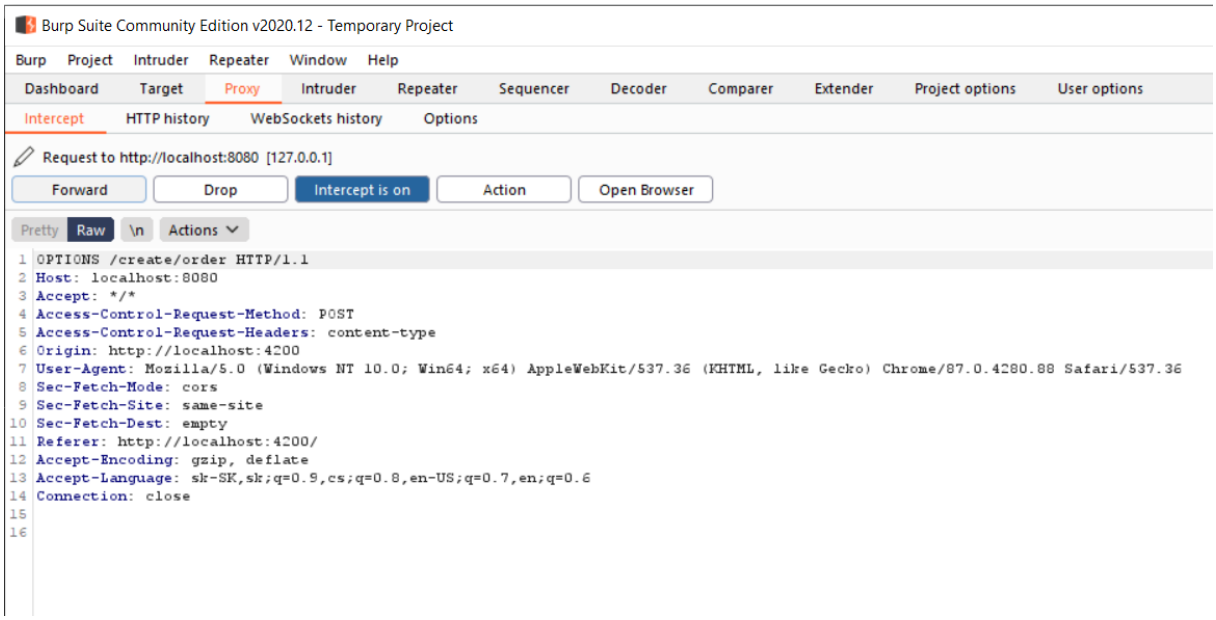
Obrázok 48: Určenie dodacej adresy

5. Vyberte nejakú platobnú metódu. Pred potvrdením nezabudnite v Burp Suite zapnúť intercept na on. Následne potvrďte.



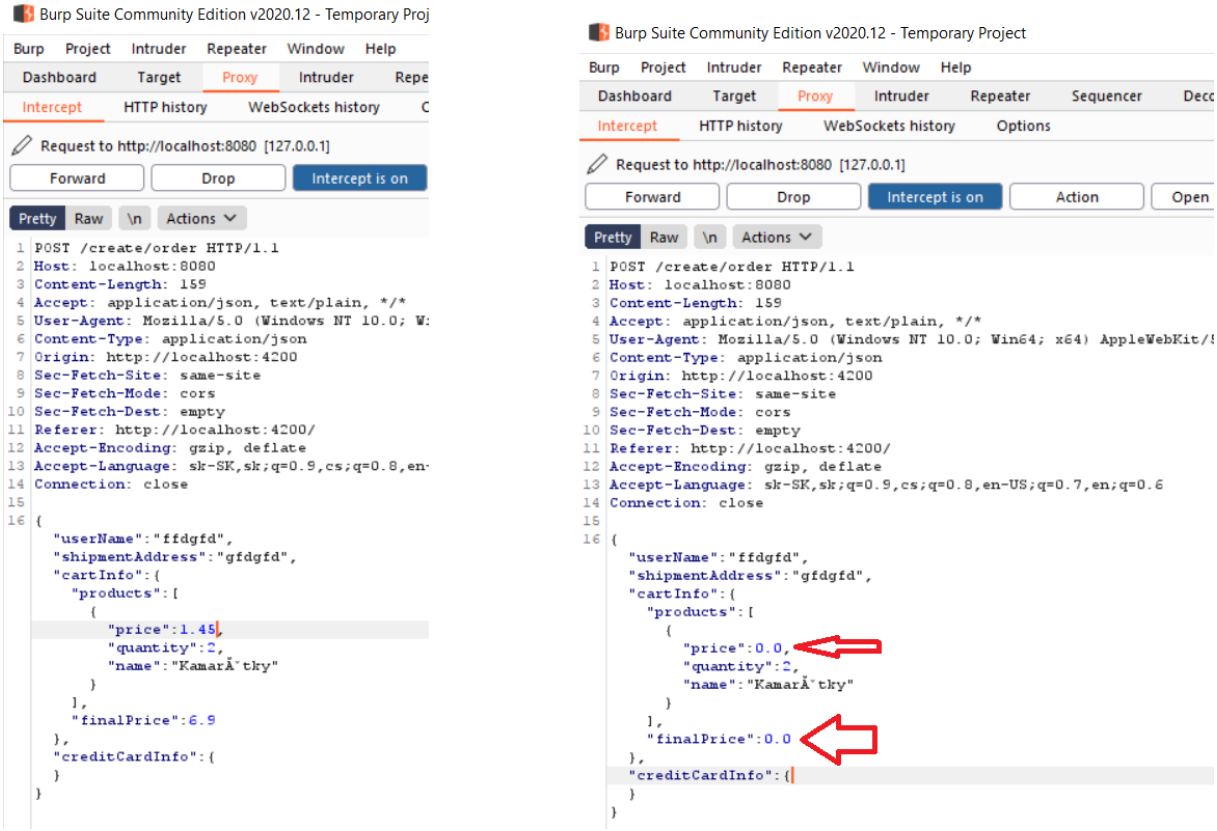
Obrázok 49: Zadanie informácií o platbe a potvrdenie

6. Prvý request prepošlite stlačením tlačidla forward.



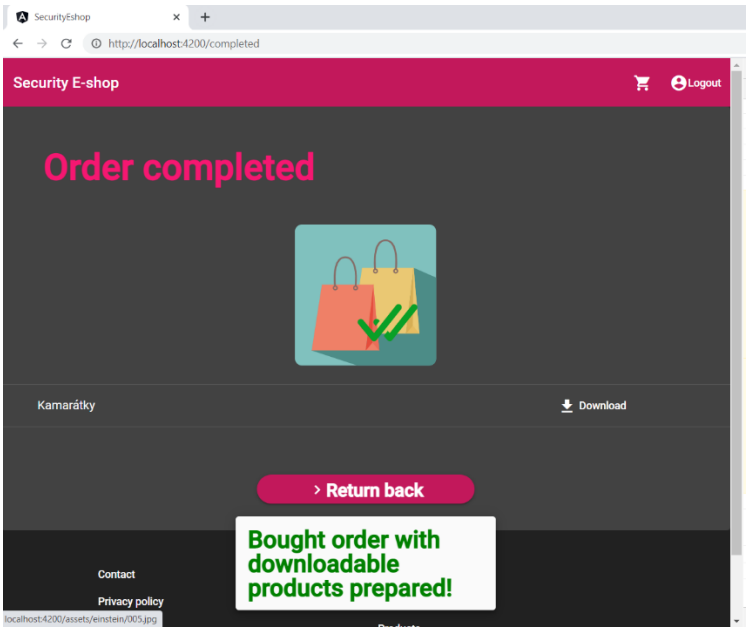
Obrázok 50: Ignorovanie prvého requestu

7. V druhom requeste zmeňte finalPrice na 0. Pre istotu zmeňte aj všetky ceny produktov na nulu. Následne stlačte forward.



Obrázok 51: Zmena informácií v druhom requeste

8. Objednávka bola úspešne uskutočnená. Teraz si môžete stiahnuť ukradnuté produkty.

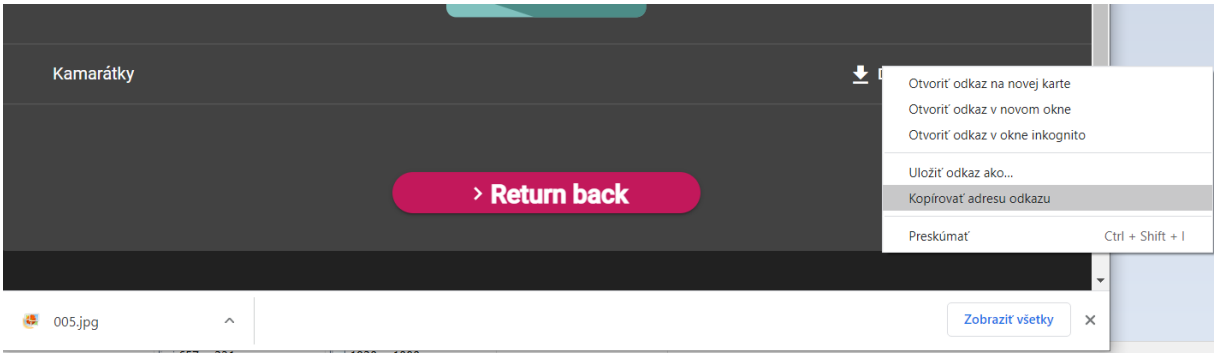


Obrázok 52: Stiahnutie ukradnutých produktov

Získanie prístupu k súborom

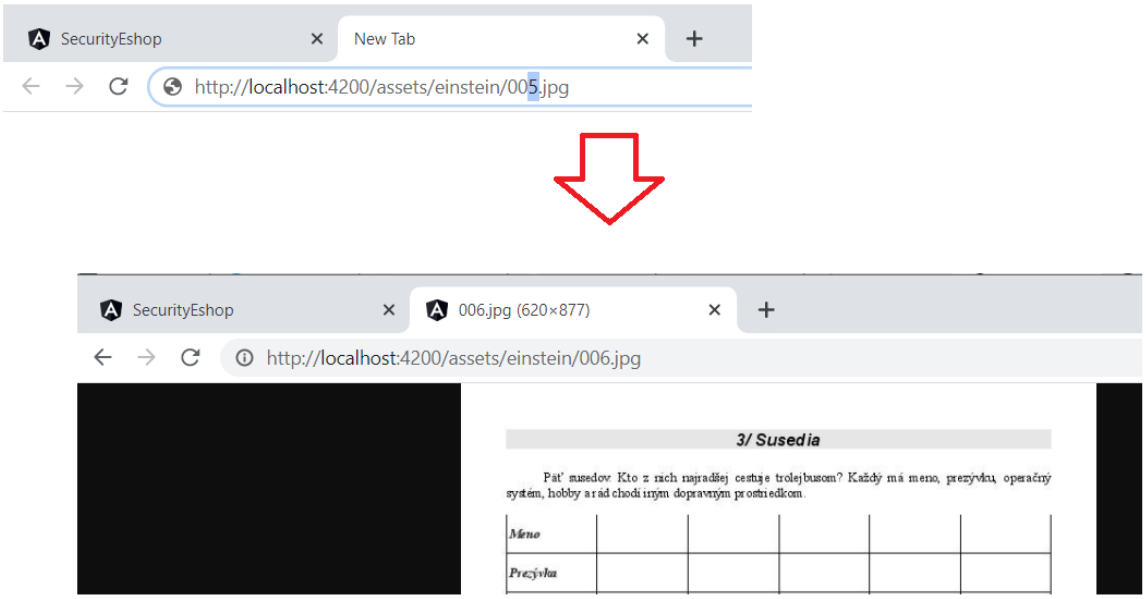
Útočník by mal vedieť, že ako technológia bol použitý Angulár. Na základe tejto informácie by mal byť schopný dostať sa k verejne uloženým súborom na stránke zadaním do prehliadača cestu k assets/images. Už je len potrebné zistiť presnú cestu. Pri minulom scenári s ukradnutím produktu si ale môže všimnúť, že produkty obsahujú cestu vedúcu na frontend a verejne dostupnú. Inkrementuje číslo nejakého súboru a získa ďalší zo súborov bez väčšej námahy. Následne môže stiahnuť obsah ponúkaných produktov aj bez nutnosti platby za ne.

1. Získajte odkaz z ukradnutého súboru.



Obrázok 53: Získanie odkazu na stiahnutý obsah

2. Použite podobný názov súboru pri zadaní do okna prehliadača.



Obrázok 54: Vyskúšanie podobnej adresy s inkrementovaným číslom obrázka