

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Míľnik I.

Tímový projekt I.

Bc. Jasmína Badová
Bc. Marek Benovič
Bc. Matej Janeček
Bc. Paula Kollárová
Bc. Eduard Krajčír
Bc. Ladislav Lelkes
Bc. Denis Piovarči

Študijný program: ISS

Vedúci: Ing. Lukáš Kohútka, PhD.

Kontakt na tím: tp.fiit.2021@gmail.com

Webové sídlo tímu: <https://team16-21.studenti.fiit.stuba.sk/>

ZS 2021

Obsah

Obsah	1
Úvod	5
Ciele pre zimný semester	6
Špecifikácia požiadaviek	7
Funkčné požiadavky	7
Nefunkčné požiadavky	7
Architektúra	8
Vývojové prostredie pre všetky súčasti projektu	8
Backend	10
Použité technológie	10
Moduly backendu	11
Frontend	14
Použité technológie	15
Služby	15
Middleware	16
Homepage	17
Správa používateľského účtu	17
Práca s repozitármi	18
Dashboard organizácie	18
Edukačný modul	19
Práca s triedami	19
Komunikačný modul	20
Prílohy	21
Príloha A - Frontend prototypy	21
Pricing	21
Features	23
About Us	24
Príloha B - Merania kvality softvéru ASICDE	25
Úvod	36
Role členov	37
Jasmína Badová	37
Paula Kollárová	38
Ladislav Lelkes	39
Matej Janeček	40
Eduard Krajčír	41
Marek Benovič	42
Denis Piovarči	43

Aplikácie manažmentov	44
Verziovanie zdrojového kódu	44
Manažment správy úloh	44
Komunikácia	44
Sumarizácia šprintov	45
Šprint 1	45
Šprint 2	47
Šprint 3	49
Globálna retrospektíva	51
Motivačný dokument	52
Metodiky	56
GitHub	56
Jira	58
Komunikácia	60
Export evidencie úloh	61
Šprint 1	61
Šprint 2	62
Šprint 3	63
Webové sídlo	65

Inžinierske dielo

Úvod

Vývojári číslicových systémov zväčša tieto systémy navrhujú v rámci jednoduchých textových editorov ako je napríklad Notepad++. Tieto editory však častokrát nie sú dostatočne robustné a vybavené na to, aby pokryli všetku požadovanú funkcionality, ktorá je vhodná alebo žiadúca pre týchto vývojárov. Editor, ktorý by predstavoval komplexnejšie a sofistikovanejšie vývojové prostredie by mohol byť riešením pre tento problém.

ASICDE je online editor, ktorý podporuje syntaktické zvýrazňovanie textu, automatické dopĺňanie textu, kontrolu syntaktických chýb a vývoj číslicového dizajnu integrovaných čipov, konkrétne pre programovací jazyk SystemVerilog. Toto vývojové prostredie by okrem editovania a vizualizácie kódu umožňovalo aj kolaboráciu a komunikáciu v rámci skupiny ľudí alebo organizácie a jej tímov. Vďaka týmto funkcionalitám by vývoj rozsiahlejších číslicových systémov mal byť jednoduchší, rýchlejší a efektívnejší. Systém bude taktiež poskytovať možnosti predaja a kúpy jednotlivých modulov, ktoré používatelia alebo organizácie sprístupnia k predaju v rámci obchodu s licenciami na IP jadrá.

Tento projekt sme prevzali po predošlých študentoch, ktorí sa tomuto projektu venovali v rámci diplomových a bakalárskych prác, a taktiež tímových projektov. Naimplementovali v ňom základné funkcionality editora, ako aj základné funkcie správy používateľov na backende. Kvôli tomu, že dané funkcionality neboli kompletne implementované a odladené, niekedy nespĺňali zadané konvencie a samozrejme, že vždy je možné systém zlepšovať, sme sa rozhodli v tomto projekte pokračovať. Našimi úlohami je ďalej vyvíjať tento systém, opraviť v ňom existujúce chyby a dokončiť neúplné funkcionality, vylepšovať ho, zvýšiť jeho bezpečnosť a pridávať do neho aj nové funkcionality.

Našou hlavnou motiváciou pre riešenie tohto projektu je práve jeho komplexnosť, zaujímavosť a vysoký potenciál byť užitočným pre mnohé firmy ako aj jednotlivcov. Ďalším dôvodom bola nedostupnosť iných dostatočne robustných nástrojov, ktoré na rozdiel od tohto systému boli často krát obmedzené, neintuitívne, ba dokonca nepoužiteľné v reálnom nasadení¹.

¹ Nakoľko pracujeme na už existujúcom projekte, na ktorom sa podieľali viacerí študenti, niektoré časti dokumentácie sme prebrali z minuloročných materiálov. Sem patrí opis a samotná architektúra softvéru, nakoľko sa tieto časti nemenili. Túto činnosť nám schválil náš vedúci p. Kohútka.

Ciele pre zimný semester

Pre zimný semester sme si stanovili nasledujúce ciele:

- oboznámenie sa s aplikáciou, prostredím a technológiami
- rozbehanie a spustenie už existujúceho projektu
- nasadenie aplikácie na server
- implementácia automatického buildenia, testovania a nasadenia
- identifikácia bugov súčasnej verzie
- úprava a doladenie nefunkčných funkcionalít
- úprava bugov
- navrhnutie prototypov pre jednotlivé podstránky
- implementácia podstránok (features, about us, pricing)
- refaktoring a optimalizácia kódu
- spísanie dokumentácií k projektu

Podrobný plán pre zimný semester sa nachádza na webovom sídle² tímu.

² <https://team16-21.studenti.fiit.stuba.sk/>

Špecifikácia požiadaviek

Súhrn funkčných a nefunkčných požiadaviek pre potreby projektu. Zoznam bol spravený po konzultácii s vedúcim projektu.

Funkčné požiadavky

Míľnik 1

- Používateľ si vie resetovať heslo v prípade že ho zabudol
- Používateľ je schopný pozvať ľudí na kolaboráciu pomocou mailu bez ohľadu na to, či je daný mail už registrovaný alebo nie
- Používatelia a organizácie vedia efektívne a jednoducho spravovať ich repozitáre a dokážu ich zdieľať
- Aplikácia umožní používateľom komunikovať v rámci aplikácie (chat)
- Zlepšenie edukačného modulu aplikácie aby bola vhodná pre praktické využitie na predmete a hodnotenie žiakov
- Používatelia a organizácie sú schopní zdieľať, predávať alebo prenajímať ich SystemVerilog projekty
- Aplikácia má viaceré licencie ktoré budú k dispozícii na zakúpenie cez platobnú bránu
 - k dispozícii sú licencie pre individuálnych zákazníkov a organizácie
 - každá z 2 vyššie spomínaných kategórii bude mať podkategórie "Basic", "Standard" a "Premium"
- Aplikácia bude obsahovať administrátorský prístup
- Aplikácia bude obsahovať modul na generovanie stavových diagramov s možnosťou ich uloženia

Nefunkčné požiadavky

- Odozva aplikácie musí byť menej ako 5 sekúnd
- Riešenie musí byť nezávislé od operačného systému alebo webového prehliadača
- Systém musí byť možné použiť viacerými používateľmi naraz
- Systém musí byť jednoducho škálovateľný
- Systém musí byť responzívny
- Systém musí byť modulárny
- Systém musí byť prehľadný
- Systém musí byť vizuálne prívetivý
- Systém musí spĺňať prísne podmienky dostupnosti
- Systém musí byť jednoducho konfigurovateľný
- API rozhranie spĺňa konvencie a štandardy REST
- Systém musí byť zabezpečený voči neautorizovaným operáciám
- API rozhranie neumožňuje získanie citlivých údajov určených pre interné spracovanie (heslá a iné metadáta)

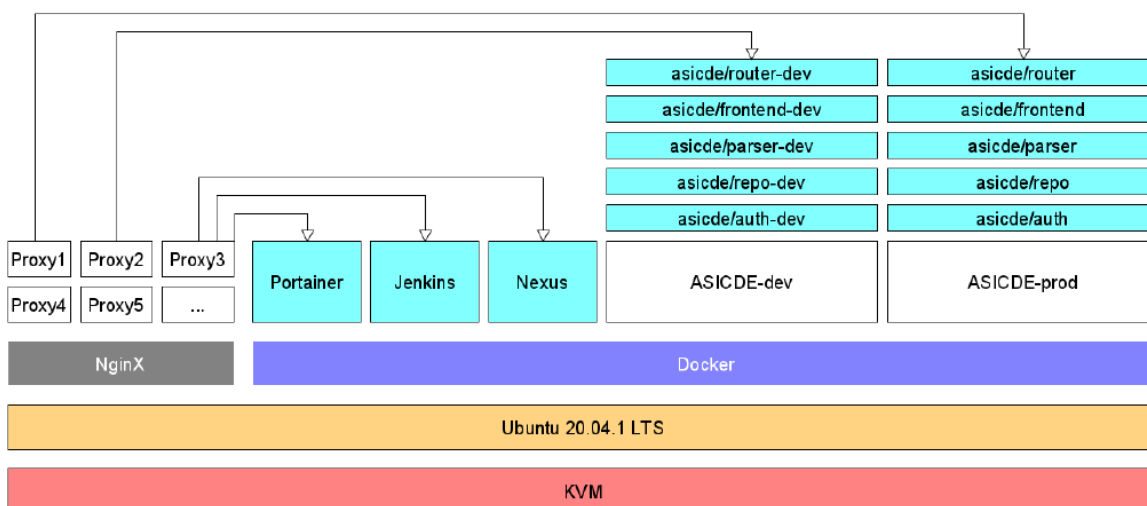
Architektúra

Našu aplikáciu vieme rozdeliť na dve hlavné časti:

- Backend
 - Základná časť, ktorá je implementovaná vo frameworku Spring (programovací jazyk Java)
 - Komunikačný a kolaboratívny modul sú implementované v jazyku TypeScript.
- Frontend - je implementovaná vo frameworku Angular (programovací jazyk JavaScript / TypeScript)

Vývojové prostredie pre všetky súčasti projektu

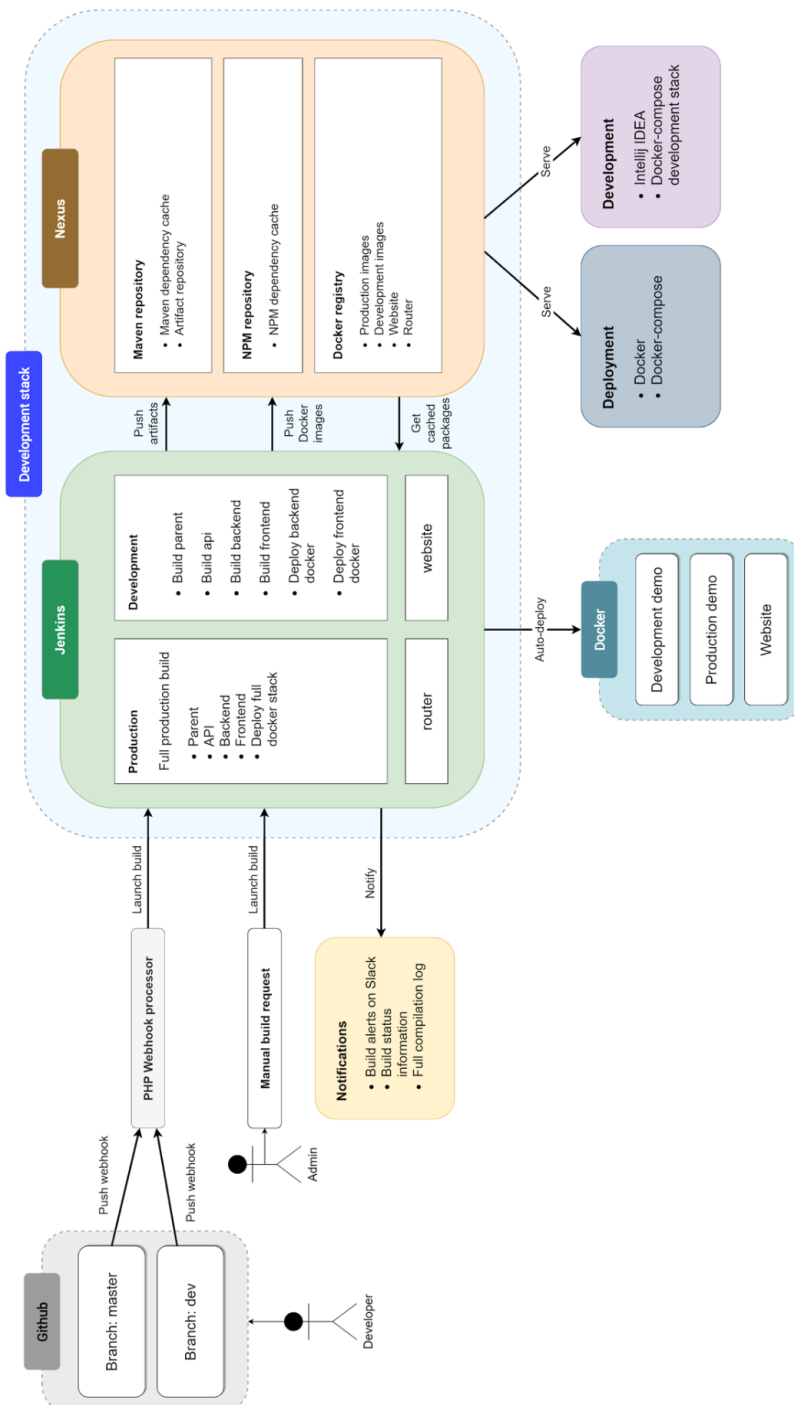
Kvôli komplexnosti projektu a jeho nasadeniu sme si vytvorili automatizované prostredie, ktoré nám uľahčí vývoj, umožní nám odsledovať všetky chyby pri kompilácii kódu ako aj nasadiť aplikáciu bez potreby manuálnej zásahy. Celý kód tohto projektu sa taktiež nachádza na kolaboračnej platforme GitHub.



Využili sme pre to virtuálny stroj, ktorý nám poskytla fakulta v rámci riešenia tohto tímového projektu. Virtuálny stroj je nainštalovaný s operačným systémom Ubuntu 20.04.1 LTS. Na tento stroj sme následne nainštalovali platformu Docker a webový server a proxy Nginx. Na platforme Docker sme nasadili jednotlivé komponenty potrebné pre automatizovaný vývoj a nasadenie ako aj samotné inštancie našej aplikácie. Webový server a proxy nám umožňujú kontrolovať prístup k aplikáciám ako aj smerovať rôzne webové adresy na služby, ktoré využívame. Služba Portainer nám umožňuje sledovať stav a spravovať jednotlivé služby nasadené v rámci platformy Docker.

Medzi služby potrebné pre vývoj patria Jenkins a Nexus. Jenkins je služba ktorá umožňuje spúšťanie automatizovaných úkonov. Využívame ju na spúšťanie kompilácie kódu hneď po jeho aktualizácii na platforme GitHub, kde je náš kód uchovávaný. Jenkins sleduje celý priebeh kompilácie a testovania jednotlivých modulov a odosiela notifikácie o úspešnosti

kompilácií na náš tímový Slack (náš komunikačný prostriedok). V prípade, že sa v kóde 8 vyskytnú chyby a kompilácia skončí s chybou, vieme na takúto situáciu promptne reagovať a dané chyby opraviť. Nexus je repozitár artefaktov, knižníc, balíkov a Dockerových obrazov. Služi ako server pre uchovanie rôznych knižníc a balíkov, ktoré využívame v našom projekte v pamäti cache, aby ich nebolo potrebné pri každej kompilácii sťahovať z internetu. Okrem toho slúži na uchovanie medzivýsledkov kompilácie a hotových modulov v rámci nášho projektu, čo nám umožňuje zdieľať tieto súbory medzi všetkými programátormi. Jenkins taktiež vytvára na konci úspešnej kompilácie kódu obrazy (balíky) pre platformu Docker, ktoré je následne možné nasadiť na akýkoľvek stroj, kde je Docker nainštalovaný, a tieto obrazy sú uložené taktiež na serveri Nexus.



Backend

Backend je súčasť nášho systému, ktorá je zodpovedná za obsluhovanie požiadaviek z používateľského rozhrania. Hlavnými úlohami backendu sú autentifikácia a autorizácia používateľov, správa súborov, správa používateľských účtov, organizácií, tímov a repozitárov.

Táto časť je implementovaná formou API (aplikačného programovacieho rozhrania), ktorého dokumentácia je súčasťou tohto dokumentu a zobrazuje všetky volania, ich vstupy, výstupy a chybové kódy, ktoré môžu požiadavky a odpovede nadobudnúť - Príloha A.

Keďže systém je navrhnutý tak, aby bol modulárny a škálovateľný, jednotlivé časti backendu sú oddelené tak, aby bola táto požiadavka splnená a jednotlivé moduly nemali medzi sebou náväznosti. Jednotlivé moduly systému a ich funkcionality sú opísané v sekcii Moduly backendu.

Použité technológie

V rámci backendu bol použitý programovací jazyk Java, nad ktorým je postavený framework Spring, ktorý slúži na implementovanie robustného API rozhrania. V projekte využívame aj správcu balíkov a verzií Maven. Vďaka tomuto nástroju dokážeme jednoducho pracovať s nadväzujúcimi knižnicami, ktoré sú potrebné v projekte, ako aj jednoducho zdefinovať spôsoby kompilácie našej aplikácie, tvorby artefaktov, ukladanie pomocných súborov do pamäte cache alebo formy testovania aplikácie.

Používané verzie nástrojov:

- Java - OpenJDK 13
- Maven - Apache Maven 3.6.3
- Spring - 2.2.5
- PostgreSQL 12

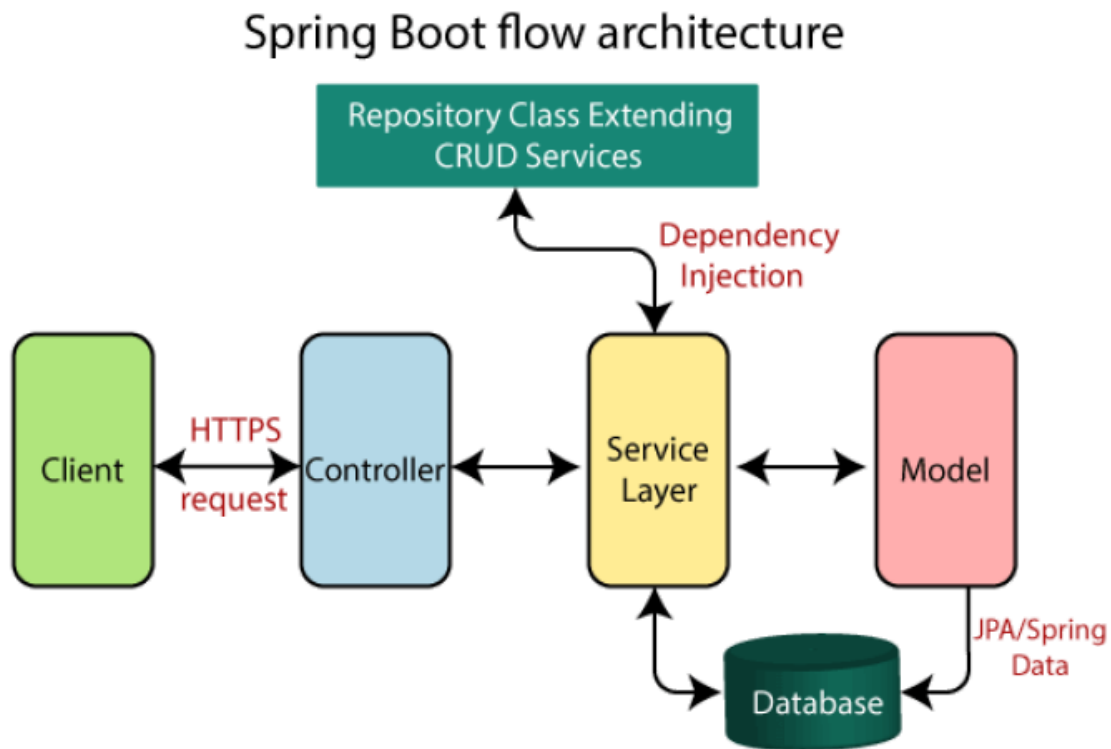
Používané verzie knižníc:

- lombok - 1.18.16
- postgresql - 42.2.5
- liquibase-core - 3.8.5
- modelmapper - 2.3.9
- jackson-jsog - 1.1.1
- java-jwt - 3.11.0

Na vývoj sme používali vývojové štúdio (IDE) IntelliJ Idea (Community edition), keďže tento nástroj poskytuje priamu podporu jazyka Java a poskytuje možnosť nainštalovať si rôzne doplnky pre prácu s frameworkom Spring ako aj knižnicami Lombok či Maven. Okrem užitočných doplnkov a šikovného prostredia poskytuje aj nástroje na odlaďovanie chýb.

Moduly backendu

V tejto časti opíšeme niektoré dôležitejšie moduly a ich úlohy. Všeobecná štruktúra modulu zodpovedá konvenciám vo frameworku Spring, kde sa každý modul skladá z troch častí (MVC): dátový model, kontrolér a zobrazenie. V realite ide o dátovú vrstvu, servisnú vrstvu a prezentačnú vrstvu.



Zdroj: <https://www.javatpoint.com/spring-boot-architecture>

core

Core modul je hlavný modul našej aplikácie. Tento modul obsahuje implementáciu nasledujúcich funkcionalít:

- Správa používateľov, autentizácia, autorizácia, vytváranie používateľských účtov
- Správa organizácií a tímov v rámci organizácie
- Vytváranie a spravovanie repozitárov
- Komunikačný (chat) modul - API
- Edukačný modul - správa tried, zadaní
- Správa projektových súborov

Tento modul je v prípade potreby možné použiť aj v inej implementácii webového IDE, nakoľko z pohľadu dependencií je minimálne zviazaný s ostatnými súčiastkami projektu. Modul obsahuje balíky rozdelené na základe použitia (konfiguračné súbory, perzistenčná vrstva, servisná vrstva, webová vrstva, zabezpečenie...).

Poskytované API týmto modulom je zabezpečené pomocou JWT (Json Web Token), a takmer ku všetkým dopytom na toto API je potrebný práve spomenutý JWT - tým je zaručená identifikácia a autorizácia používateľa, ktorý daný dopyt vykonal. Dokumentácia API je priložená v tejto práci.

Všeobecný tok spracovania dopytu je nasledujúci:

1. Prijatie dopytu na aplikačnej vrstve (Controller), validácia a extrakcia tela a parametrov dopytu
 - a. Ošetrovanie autorizácie - JWT a zároveň z pohľadu domény (PreAuthorize anotácia)
2. Volanie servisnej vrstvy (Service, resp. Service implementation)
 - a. Ošetrovanie korektnosti parametrov (nulové hodnoty a pod.)
3. Získanie dát z perzistenčnej vrstvy (spravované pomocou JPA/Hibernate) - v prípade nutnosti, ako napr. ukladanie objektu, získanie objektu resp. dát a podobne
4. Mapovanie objektu do API odpovede (mapovanie z pohľadu entity na Data Transfer Object (DTO))
5. Odoslanie odpovede pomocou HttpServlet-u

V prípade neočakávaného stavu alebo neúspešnej autentizácie/autorizácie po spracovaní dopytu ošetrojeme tieto situácie pomocou špecifických HTTP kódov a vlastnej chybovej hlášky, ktorá je obsiahnutá v odpovedi na daný dopyt.

Dátová vrstva pozostáva z PostgreSQL databázy. Databáza je rozdelená na viacero schém:

- public - základná schéma postgresQL databázy. V tejto schéme sa nachádzajú tabuľky potrebné pre verziovací systém Liquibase
- chat - schéma pozostávajúca z databázových tabuliek pre komunikačný modul
- core - schéma obsahujúca tabuľky zodpovedajúce entitám pre modul core

Ako už bolo spomenuté vyššie, na verziovanie databázy používame technológiu Liquibase, ktorá zabezpečuje vytváranie databázových objektov a ich úpravu pomocou štruktúrovanej schémy (v tomto projekte vo formáte XML), a tým môžeme abstrahovať od používania samotných SQL databázových dopytov - avšak v prípade nutnosti môžeme tiež vytvoriť vlastný SQL skript, ktorý bude vykonávaný ako každý iný verziovací súbor.

Konfigurácia core modulu je možná pomocou premenných prostredia, alebo pomocou premenných definovaných v súbore resources/application.properties. Všetky špecifické nastavenia pre tento projekt sú nasledovné:

parser

Parser modul rieši prácu s kódom ktorý používatelia upravujú v editore aplikácie. Jeho hlavnou úlohou je analýza a validácia tohto kódu.

commons-error-handling

Modul commons-error-handling implementuje vlastnú logiku mapovania chybových hlášok na odpovede obsiahnuté v REST odpovediach. Je knižnica obsahujúca definíciu

projektových výnimiek, ktoré sa používajú v každom module. Každá nízko úrovňová výnimka vyhodnená knižnicou tretej strany, je zaobalená do jednej z projektových výnimiek definovanej práve v tejto knižnici. Obsahuje tiež konfiguráciu REST API, ktorá každú projektovú výnimku preloží do HTTP odpovede s prislúchajúcim HTTP statusom.

commons-security

Je knižnica zodpovedná za zabezpečenie aplikácie. Obsahuje konfiguračné súbory, ktoré nastavujú a dopĺňajú základnú Spring Security vrstvu. Definuje úroveň zabezpečenia pre jednotlivé cesty REST služby, obsahuje v sebe Autentifikačného klienta, pomocou ktorého priamo komunikuje s autentifikačným modulom. Ak je potrebné, aby nový modul podporoval zabezpečenie, stačí importovať knižnicu Commons Security a načítať konfiguračné súbory do kontextu aplikácie. Tento modul momentálne nie je plne využívaný v našej práci, nakoľko v dôsledku rozsiahlych zmien sa presunula logika autentizácie používateľov do modulu core.

commons-web

Obsahuje v sebe konfiguračné súbory, ktoré nastavujú webovú vrstvu aplikácie. Importuje Spring Web knižnicu aj s vybranými konfiguračnými súbormi. Konfiguruje Spring Actuator, ktorý umožňuje sledovať základné údaje o aplikácii ako status a rôzne metriky počas behu. Tento modul by mala využiť každá súčasť systému, ktorá pracuje tiež na úrovni webových (REST) služieb. Okrem už spomenutých funkcionalít tento modul obsahuje tiež konfiguráciu, ktorá riadi spracovanie výnimiek (Exceptions), ktoré sa ďalej spracúvajú v komponente commons-security.

Komunikačný modul

Komunikačný modul je oddelený od hlavného modulu projektu. Je napísaný v jazyku TypeScript bez podporného frameworku. Tento modul slúži na sprostredkovanie rozhrania medzi samotnou implementáciou na frontende a hlavným modulom, ktorý slúži na správu a ukladanie komunikácie. Komunikačný backend disponuje websocket serverom, ktorý umožňuje komunikáciu v reálnom čase a následne komunikuje s API na hlavnom backende. Na vytvorenie tohto komunikačného kanálu je využitá knižnica Socket.io. Úlohou tohto modulu je vytvárať a manažovať chatovacie miestnosti, pripojených klientov, a distribuovať odosielané správy.

Kolaboratívny modul

Tento modul je prácou ďalšieho študenta a je zapojený do systému ako izolovaný modul podobne ako komunikačný modul. Jeho úlohou je sprostredkovať funkcionalitu zdieľaného editovania zdrojových kódov viacerými používateľmi súčasne. Pre komunikáciu so zvyškom systému je rovnako využívané API rozhranie hlavného modulu.

Frontend

Frontend je používateľské rozhranie pre tento systém. Sprístupňuje hlavnú stránku projektu ako aj rôzne funkcionality editora, organizácií, či chatu. Na pozadí komunikuje s API rozhraním na backende, vďaka ktorému môže registrovať, autentifikovať a autorizovať používateľov, zobrazovať im ich repozitáre, informácie o používateľskom účte, a spravovať organizácie či tímy. Hlavné funkcie editora, ako vizualizácia kódu, oprava syntaxe či validácia písaného kódu sa z väčšej časti realizuje priamo v klientskej aplikácii. Backend slúži hlavne na uchovanie súborov repozitárov a ďalších dát.

Samotný frontend sa skladá z viacerých modulov, ako napríklad auth - login, registrácia a autentifikácia používateľa, editor - práca s editorom, home - nastavenia používateľa a repozitárov, app-routing.module - prepája stránky na frontende, a ďalšie. Tieto moduly sú zväčša rozdelené do viacerých komponentov. Každý z komponentov sa skladá z implementácie layoutu v HTML, definovania štýlu v SCSS, testov napísaných v TypeScripte a samotného controlleru taktiež napísanom v TypeScripte. Okrem modulov obsahujúcich komponenty sú na frontende implementované viaceré services, čiže služby, ktoré slúžia na komunikáciu s backendom, alebo spracovanie dát.

V aktuálnom stave sú implementované nasledovné funkcionality:

- Webové rozhranie - využitie frameworku Angular a Typescript.
- Editor - založený na existujúcom Monaco editore, s pridanou podporou programovacieho jazyka SystemVerilog.
- Editor - podporuje syntaktické zvýrazňovanie textu, kontrolu syntaktických chýb a automatické dopĺňanie textu.
- Editor - je doplnený kolaboračným modulom pre spoluprácu viacerých používateľov v rámci jedného repozitáru
- Komunikácia s API rozhraním (backend).
- Homepage - úvodná stránka do systému. V prípade, že používateľ nie je prihlásený, je mu ponúknutá možnosť prihlásenia alebo registrácie.
- Autorizácia - prihlásenie existujúceho používateľa alebo registrácia nového používateľa.
- Autentifikácia - používateľ je autentifikovaný pomocou JWT (JSON web token).
- Správa používateľa - zahŕňa spravovanie osobných údajov.
- Správa repozitárov - zahŕňa zobrazenie a spravovanie všetkých osobných repozitárov.
- Úprava nastavení (osobných a repozitárov) je povolená až po úspešnom prihlásení používateľa.
- Organizácie - používateľ vie vytvoriť organizáciu. Používateľ, ktorý organizáciu vytvoril, je jej vlastníkom. Vlastník organizácie ju následne vie aj zmazať.
- Správa členov organizácie - Vlastník organizácie vie pridávať členov do organizácie, aj ich z organizácie odstrániť.
- Správa tímov - v rámci organizácie používateľ vie vytvoriť tímy. Do existujúceho tímu môžu byť priradení členovia alebo môžu byť z tímu odstránení. Členom tímu môžu byť zmenené ich práva.

Použité technológie

Klientská časť systému bola implementovaná v jazykoch JavaScript a TypeScript za pomoci framework AngularJS. V rámci tohto frameworku sme využili aj nasledovné dodatočné moduly, ktoré nám sprístupňujú dôležité funkcionality:

- angular/core - 9.1.6
- angular/common - 9.1.6
- angular/material - 9.2.3
- angular/router - 9.1.6
- ng-bootstrap/ng-bootstrap - 8.0.0
- syncfusion/ej2-angular-navigations - 18.1.45
- jszip - 3.4.0
- ngx-monaco-editor - 8.1.1
- socket.io-client - 4.0.1
- zip-js - 0.0.2

Na implementovanie samotného prostredia pre písanie kódu je namiesto implementovania vlastného editora použitý existujúci open-source editor Monaco. Tento editor disponuje funkcionalitami ako napríklad syntax highlighting, vyznačovanie chýb a upozornení a pridávanie nových jazykových modulov. Toto nám veľmi uľahčilo prácu a okrem toho je možné tento editor nezávisle od modifikácií, ktoré doňho budú nami implementované, aktualizovať.

Služby

Authentication service

Autentifikačná a autorizačná služba je kritickým komponentom celého systému. Služi na prvotné prihlásenie a registráciu používateľa ako aj na udržanie aktuálnej relácie používateľa v rámci aplikácie. Jej hlavnou úlohou je správa relácie používateľa a udržiavanie tejto relácie počas celej interakcie s aplikáciou. Táto relácia je ukladaná v rámci local storage v prehliadači a pri interakcii so systémom sa nepretržite obnovuje, aby používateľ nestratil prístup a nebol odhlásený počas tejto interakcie. Pri odhlásení taktiež korektne zruší používateľskú reláciu a odstráni citlivé dáta, ktoré sú uchovávané v prehliadači pre správny chod aplikácie.

User service

Táto služba slúži na správu používateľov. Sprostredkúva rozhranie medzi API na backende a samotnou aplikáciou.

Repository service

Služba správy repozitárov sprostredkúva dôležitú funkcionality spojenú s manažmentom a interakciou s repozitármi. Pri vybraní repozitáru si uchováva jeho dáta, a následne umožňuje

interakciu s ním. Poskytuje funkcionality pre úpravu repozitára, jeho nastavení, ako aj prácu s gitom a súbormi.

Organization service

Táto služba poskytuje všetkú funkcionality spojenú so správou a interakciou s organizáciami, tímami a ich repozitármi. Umožňuje spravovať nastavenia organizácie, pridávať a odoberať jej členov, či spravovať tímy v rámci danej organizácie.

Collab service

Kolaboračná služba je kritickým komponentom systému, ktorý je úzko naviazaný na prácu v editore a prácu so súbormi. Jeho hlavnou funkcionality je sprostredkovanie interakcie s repozitárom a jeho súbormi v reálnom čase, teda vytvára medzivrstvu spracúvajúcú zmeny v zdrojových kódach a tieto zmeny následne distribuuje všetkým pripojeným klientom, a zaručuje korektné ukladanie zmien všetkých klientov. Okrem kódu repozitára rovnako synchronizuje a sprostredkuje interakciu v reálnom čase s vizualizáciou logických členov. Ďalšou funkcionality je správa aktuálne pripojených klientov a zobrazovanie ich prítomnosti, pozície v editore, a ostatných interakcií.

Chat service

Komunikačná služba sprostredkováva funkcionality potrebné pre správu chatovacích miestností, ich používateľov a samotnej komunikácie, ktorá je v reálnom čase spracovávaná.

Education service

Táto služba poskytuje všetku funkcionality spojenú so správou a interakciou s používateľov a tried, žiakov a zadaní. Umožňuje spravovať nastavenia triedy, pridávať a odoberať jej členov, či spravovať zadaní v rámci danej triedy.

Middleware

JWT interceptor

Tento pomocný kód slúži na autorizovanie požiadaviek z aplikácie voči API rozhraniu na backende. Každá požiadavka (okrem požiadaviek na verejné časti API) musí byť autorizovaná. Na toto sú využívané JWT tokeny, ktoré sa pomocou tohto interceptora pridávajú do hlavičiek požiadaviek a autorizujú používateľské interakcie.

Error interceptor

Tento pomocný kód slúži na spracovanie chybových odpovedí z API rozhrania a transformovanie chýb do správ čitateľných a zrozumiteľných pre bežného používateľa aplikácie.

Homepage

Domovská stránka predstavuje prvý kontakt s používateľom. Na nej je používateľovi poskytnutá možnosť registrácie, pokiaľ ešte nemá založený účet. V prípade, že používateľ už je registrovaný, môže sa z domovskej stránky priamo prihlásiť. Pri zvolení jednej z dvoch možností, používateľ bude presmerovaný na príslušný formulár, kde vyplní buď svoje prihlasovacie údaje (v prípade loginu) alebo svoje osobné údaje potrebné na registráciu. Či už v prípade registrácie, alebo prihlásenia, používateľovi je v oboch prípadoch poskytnuté prejsť na opačnú možnosť (v registračnom formulári prejsť na login a z loginu možnosť prejsť na registráciu). V prípade, že používateľ je prihlásený, domovská stránka ho privíta a poskytne mu možnosť prejsť na prehľad svojich repozitárov. Domovská stránka a s ňou súvisiace registrácia a prihlásenie používateľa obsahuje nasledovné komponenty:

- LandingComponent: Komponent domovskej stránky, ktorý slúži na to, aby sa zistilo, či je aktuálne prihlásený používateľ alebo nie je. Podľa toho buď privíta prihláseného používateľa alebo mu poskytne možnosť prihlásenia a registrácie. Následne presmeruje používateľa na stránku jeho voľby.
- AuthContainerComponent: Komponent, ktorý slúži na zobrazenie, formátovanie a následne presmerovanie na stránky registrácie alebo prihlásenia.
- LoginComponent: Komponent slúžiaci na prihlásenie používateľa, na prijatie a kontrolu prihlasovacích údajov. V prípade, ak údaje boli nesprávne, respektíve používateľ s danými prihlasovacími údajmi neexistuje, bude vypísaná chyba oznamujúca neúspešné prihlásenie. V opačnom prípade používateľ bude úspešne prihlásený.
- RegisterComponent: Komponent zodpovedný za registráciu nového používateľa. V prípade, že boli zadané všetky potrebné údaje v správnom formáte, používateľ bude úspešne zaregistrovaný a systém ho presmeruje na prihlasovaciu stránku, kde sa môže prihlásiť. V opačnom prípade, ak údaje neboli všetky vyplnené alebo nespĺňajú stanovený formát, používateľ bude upozornený.

Správa používateľského účtu

UserEditationComponent slúži na to, aby prihlásený používateľ mal možnosť spravovať svoj účet (My account). V rámci tohto je mu poskytnutá možnosť zmeniť si svoje meno (krstné meno a priezvisko), mailová adresa, ako aj zmeniť si heslo. Používateľ vie svoj účet aj odstrániť zo systému. V prípade, že používateľ je vlastníkom organizácie, tak v rámci týchto nastavení mu je umožnené danú organizáciu odstrániť. Taktiež, ak je používateľ členom nejakej organizácie, v nastaveniach účtu má možnosť opustiť organizáciu.

Práca s repozitármi

Ďalším prvkom na frontende je prehľad repozitárov prihláseného používateľa. Rozloženie tejto obrazovky sme koncipovali podľa predom pripravených a konzultovaných návrhov (mockupy v prílohe). Vďaka tejto obrazovke má používateľ prístup ku svojim repozitárom, ktoré môže spravovať alebo priamo k nim pristupovať a editovať ich. V prehľade má k dispozícii tie najpodstatnejšie informácie o repozitároch ako je ich názov, popis repozitáru a pod.

- **Dashboard repozitárov:** MyReposComponent je komponent, ktorý slúži na základný prístup k repozitárom - zobrazenie repozitárov prihláseného používateľa, možnosť pridať nový repozitár, pridať repozitár medzi obľúbené, archivovať repozitár a zmazať repozitár. Umožňuje presmerovanie k nastaveniam zvoleného repozitára a otvorenie repozitára v editore.
- **Nastavenia repozitárov:** V rámci nastavení repozitára (RepositoryEditationComponent) používateľ vie upravovať zvolený repozitár - zmeniť mu názov a URL. Používateľ by mal vedieť vykonávať ďalšie akcie, ako sú odstránenie a archivovanie zvoleného repozitára, deaktivovať synchronizáciu s gitom, zdieľať repozitár alebo zmeniť vlastníka.
- **Editor:** EditorComponent je hlavný komponent editora. V rámci tohto komponentu sú implementované základné operácie pre prácu so súborami repozitára, navigáciu v rámci editora ako aj samotná inicializácia Monaco editora. Navigácia medzi otvorenými oknami editora a samotná implementácia editora sú oddelené do osobitných komponentov - EditorTabsComponent a EditorTileComponent.

Dashboard organizácie

Manažment organizácií je nová funkcionálna, ktorá bola pridaná do systému. Používateľovi je umožnené vytvoriť organizáciu, do ktorej môžu byť pozvaní členovia a môžu v nej byť vytvorené tímy a repozitáre. Správa tímov a ich členov, repozitárov a ich nastavení je zakomponovaná do časti organizácií. Používateľ má možnosť prehliadať jednotlivé repozitáre a priradovať ich jednotlivým tímom. V časti spravovanie tímov, vidí výpis členov tímu, môže meniť ich poverenia a priradovať jednotlivé repozitáre.

1. Repozitáre:
2. Tímy organizácie: V rámci organizácie môžu byť vytvárané tímy. Každý tím sa skladá z aspoň jedného člena (používateľ, ktorý tím vytvoril). Členov tímu vieme rozdeliť nasledovne:
 - a. Vlastník organizácie (Owner)
 - i. Vytvoril tím, automaticky je jeho vlastníkom
 - ii. Má právo zmazať tím
 - iii. Má právo meniť nastavenia tímu
 - iv. Má právo pridávať a odstraňovať členov do/z tímu
 - v. Má právo meniť práva členov tímu
 - vi. Vidí repozitár tímu

- vii. Má právo upravovať kód v rámci repozitáru
 - b. Administrátor (Admin)
 - i. Je spravcom tímu
 - ii. Má právo meniť nastavenia tímu
 - iii. Má právo pridávať a odstraňovať členov do/z tímu
 - iv. Má právo meniť práva členov tímu
 - v. Vidí repozitár tímu
 - vi. Má právo upravovať kód v rámci repozitáru
 - c. Member (read and write právo)
 - i. Vidí tím, do ktorého patrí
 - ii. Vidí repozitár tímu
 - iii. Má právo upravovať kód v rámci repozitáru
 - d. Viewer (read právo)
 - i. Vidí tím, do ktorého patrí
 - ii. Vidí repozitár tímu
- 3. Členovia organizácie: Členov organizácie delíme na:
 - a. Vlastník organizácie (Owner)
 - i. Vytvoril organizáciu, automaticky je jej vlastníkom
 - ii. Má právo zmazať organizáciu
 - iii. Má právo meniť nastavenia organizácie
 - iv. Má právo zmeniť vlastníka organizácie
 - v. Má právo pridávať a odstraňovať členov do/z organizácie
 - vi. Má právo vytvárať tímy v organizácii (stane sa vlastníkom tímu)
 - b. Člen organizácie (Member)
 - i. Vidí obsah organizácie
 - ii. Vidí tímy, ktorých je členom

Edukačný modul

Edukačný modul a jeho manažment je nová funkcionálna, ktorá bola pridaná do systému. Používateľovi je umožnené vytvoriť triedu, do ktorej môžu byť pozvaní členovia (žiaci alebo owneri) a môžu v nej byť vytvorené úlohy, ktoré žiaci musia vypracovať. Vypracované úlohy žiakov sa tiež nahrávajú do systému. Správa tried a ich členov, úloh a ich nastavení je zakomponovaná do časti edukačného modulu. Používateľ má možnosť prehliadať jednotlivé triedy, owner (učiteľ) má možnosť odstrániť triedy alebo upraviť triedy. V časti spravovanie tried, vidí výpis členov tímu a výpis všetkých úloh v triede.

Práca s triedami

Ďalším prvkom na frontende je prehľad tried prihláseného používateľa. Rozloženie tejto obrazovky sme koncipovali podľa dohodnutej a spísanej špecifikácie podľa vedúceho tímu. Vďaka tejto obrazovke má používateľ prístup ku svojím triedam, ktoré môže spravovať alebo priamo k nim pristupovať a editovať ich. V prehľade má k dispozícii tie najpodstatnejšie informácie o úlohách a členoch.

- **Dashboard tried:** ClassroomsComponent je komponent, ktorý slúži na základný prístup k triedam - zobrazenie tried prihláseného používateľa, možnosť pridať novú triedu alebo pripojiť sa už k existujúcej. Ak je používateľ ownerom niektorej z tried, tak ju môže odstrániť alebo zmeniť jej nastavenia, avšak to už sa rieši v nasledujúcom komponente.
- **Nastavenia tried:** V rámci nastavení tried (ClassroomSettingsComponent) používateľ vie upravovať zvolenú triedu - zmeniť názov, slug a heslo. Používateľ by mal vedieť vykonávať ďalšie akcie, ako sú zdieľať triedu alebo uzamknúť triedu.
- **Trieda:** ClassroomComponent je hlavný komponent dashboardu tried. V rámci tohto komponentu sú implementované základné operácie pre prácu so v triede. Medzi možnosti patrí vytváranie úloh, pridávanie používateľov. Okrem toho je tiež možné priamo odstrániť používateľov alebo úlohy z triedy. Taktiež sú vytvorené úlohy najskôr v stave neublikovanom. Používateľ môže rovno z tohoto komponentu vybrať úlohy, ktoré publikuje.
- **Nastavenie úloh:** V rámci nastavení úloh (AssignmentComponent) je zobrazená úloha vytvorená učiteľom spolu s úlohami, ktoré vypracovali jednotliví žiaci. Tento komponent je potrebné doimplementovať o funkcionality ako hodnotenie odovzdaných úloh, zmenu dátumu odovzdania, zmenu zadania úlohy. Zatiaľ je tento komponent iba akýmsi prototypom pripraveným pre ďalšiu prácu na ňom.

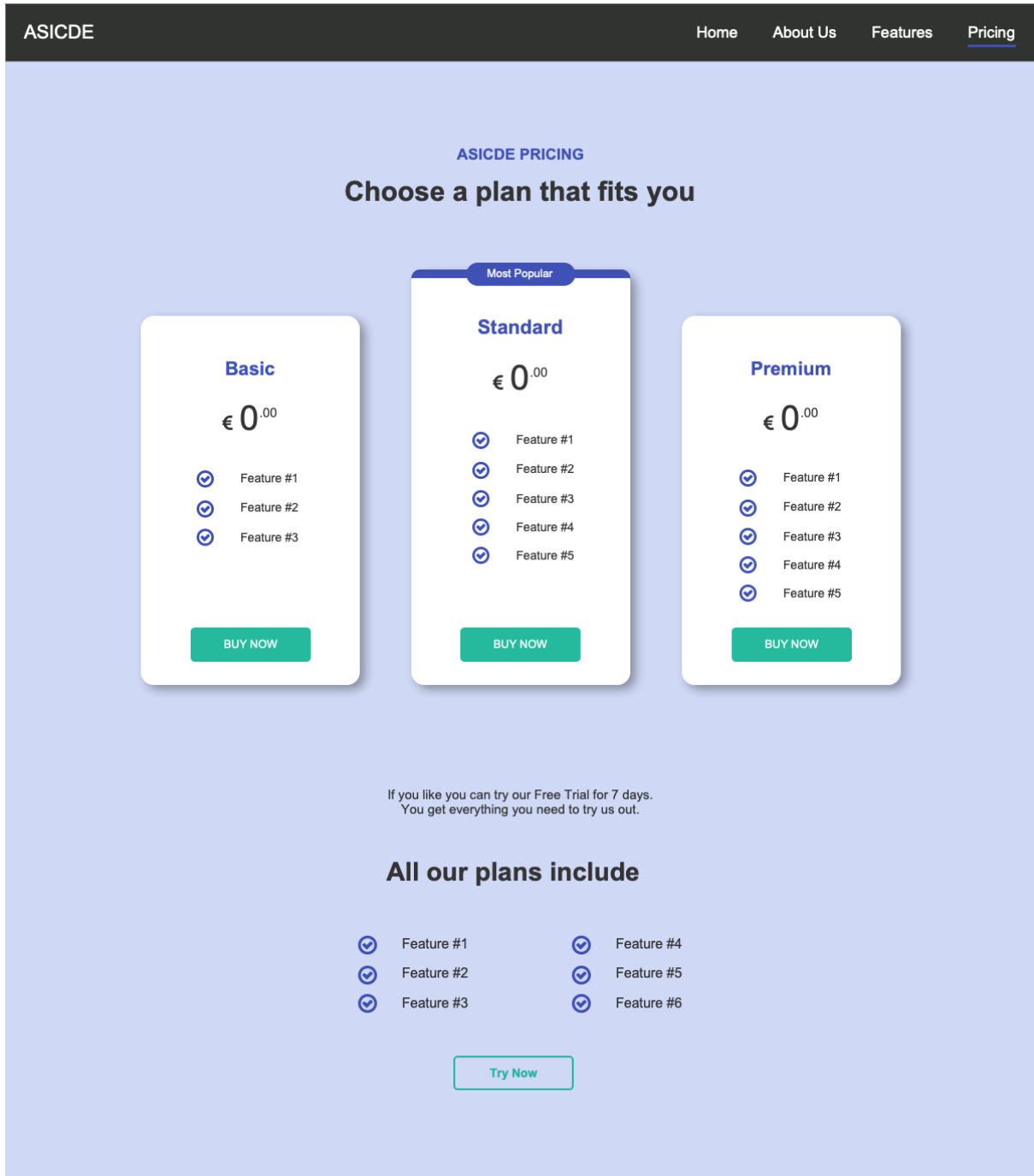
Komunikačný modul

Komunikačný modul je nová funkcionality, ktorá pridáva možnosť chatovania v podobe dedikovaných chatovacích miestností pre jednotlivé organizácie. Každá organizácia má by default vytvorenú chatovaciu miestnosť kde sú pridaní všetci členovia organizácie. Používateľ organizácie vidí na frontende zoznam všetkých chatovacích miestností v ktorých sa nachádza, spolu s ich názvom a dodatočnými informáciami ako napr. počet aktuálne pripojených užívateľov v miestnosti. Po kliknutí na chatovaciu miestnosť sa používateľ pripojí a notifikuje všetkých pripojených užívateľov o svojom pripojení do miestnosti, taktiež vidí chatovaciu históriu danej miestnosti aj v čase keď pripojený nebol. Používateľ potom môže ľubovoľne chatovať v miestnosti v reálnom čase. Momentálne nie je na frontende možnosť vytvoriť nové chatovacie miestnosti okrem predvolenej pre organizáciu, avšak funkcionality je na backende implementovaná s potenciálom pre jednoduché rozšírenie o nové funkcionality ako user management, chatroom management, súkromné správy a podobne.

Prílohy

Príloha A - Frontend prototypy

Pricing





ASICDE PRICING

Choose a plan that fits you

Basic€ 0^{.00}

- Feature #1
- Feature #2
- Feature #3

[BUY NOW](#)

Most Popular

Standard€ 0^{.00}

- Feature #1
- Feature #2
- Feature #3
- Feature #4
- Feature #5

[BUY NOW](#)**Premium**€ 0^{.00}

- Feature #1
- Feature #2
- Feature #3
- Feature #4
- Feature #5

[BUY NOW](#)

If you like you can try our Free Trial for 7 days.
You get everything you need to try us out.

All our plans include

- Feature #1
- Feature #2
- Feature #3
- Feature #4
- Feature #5
- Feature #6

[Try Now](#)


Features

ASICDE [Home](#) [About Us](#) [Features](#) [Pricing](#)

ASICDE / Features

App overview

- ✓ Feature #1
- ✓ Feature #2
- ✓ Feature #3
- ✓ Feature #4
- ✓ Feature #5




</> 🔗 💬

Feature #1 slogan

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo.


- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet.
- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet.



Feature #2 slogan

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo.


- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet.
- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit.



Feature #3 slogan

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo.


- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet.
- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet.



Feature #4 slogan

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo.


- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet.
- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit.



Feature #5 slogan

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet. Proin gravida dolor sit amet lacus accumsan et viverra justo commodo.

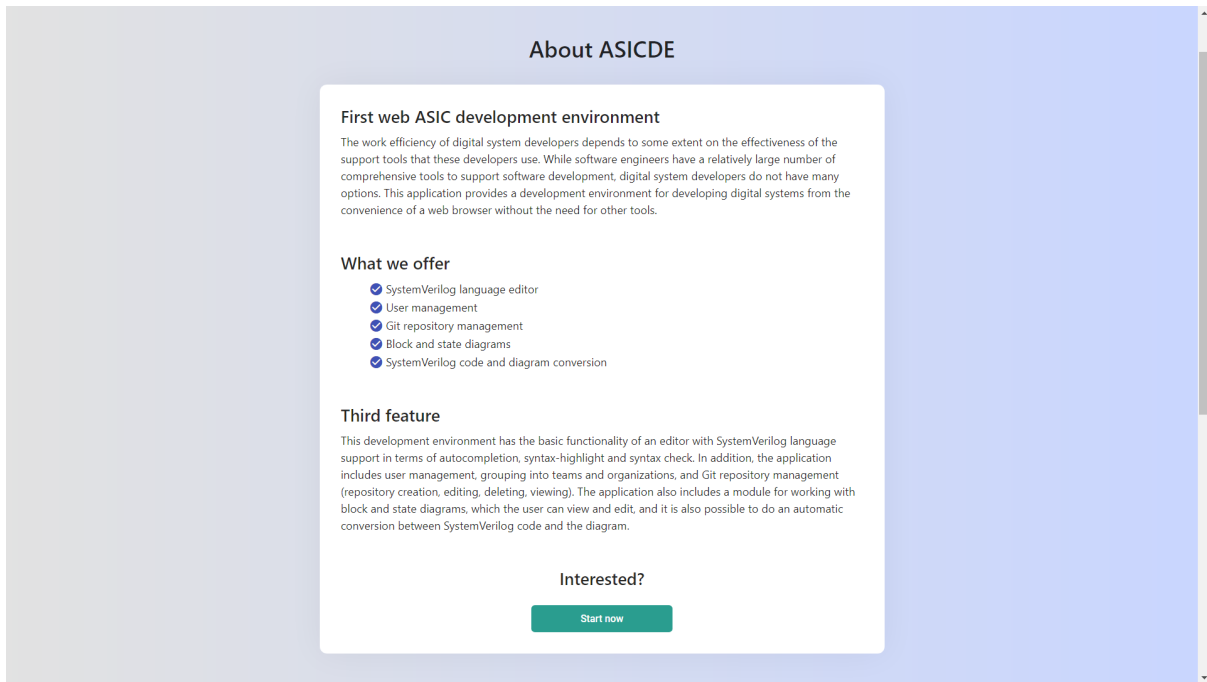
- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet.
- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit.
- ✓ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean euismod bibendum laoreet.



About Us

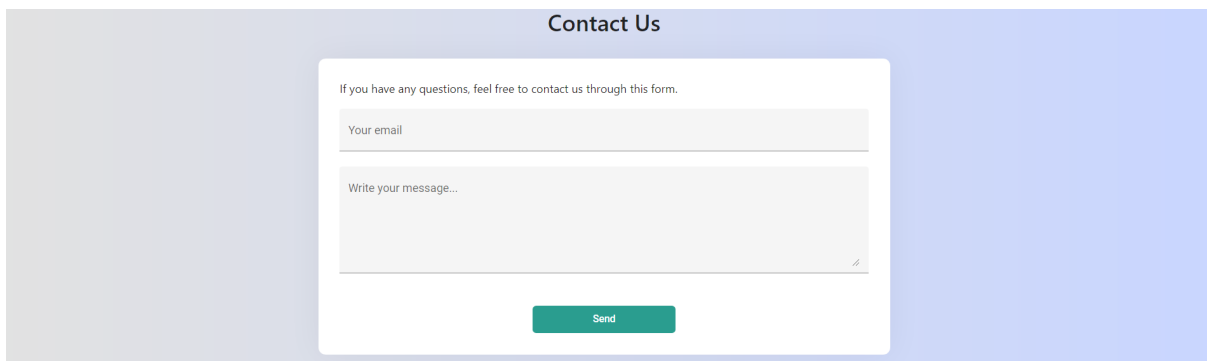
1) sekcia: About Us

Krátke predstavenie projektu pre potenciálneho používateľa spolu so zoznamom ponúkaných funkcií



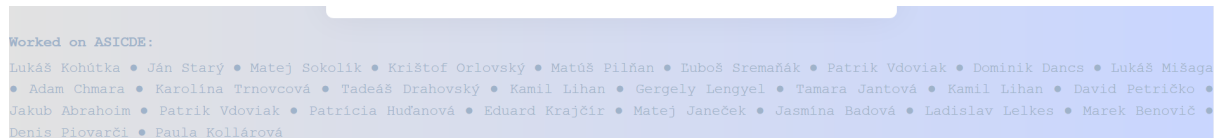
2) sekcia: Contact Us

Kontaktný formulár, pomocou ktorého nás bude môcť používateľ kontaktovať.



3) sekcia: Worked on ASICDE

Zoznam všetkých ľudí, ktorí sa podieľali na vývoji tohto projektu.



Príloha B - Merania kvality softvéru ASICDE

Úvod

Ako zdrojový kód na vykonanie metrík a následného refaktoringu som si vybral aplikáciu ASICDE z tímového projektu. Jedná sa o webovú aplikáciu, ktorá pomáha hardvérovým inžinierom navrhovať modely číslicových systémov.

Aplikácia pozostáva z viacerých modulov, ktoré sú úzko prepojené. Keďže tento projekt je vyvíjaný už niekoľko rokov skrz tímové projekty a záverečné práce, zdrojový kód je obsiahlejší. Z týchto dôvodov budem analyzovať a merať iba modul ASICDE-backend, ktorý obsahuje hlavnú časť backend logiky celej aplikácie. Na vývoj aplikácie je používaný Java framework SpringBoot. Merania a refaktoring robím nad vetvou dev z dôvodu horšej kvality kódu oproti master vetve. Výsledný zrefaktorovaný kód bude využitý aj v rámci predmetu KPAIS, preto dbám na skompilovateľnosť kódu.

Analýza

V rámci analýzy zmeriam metriky zdrojového kódu pomocou troch nástrojov:

- SourceMonitor
- SonarQube
- JArchitect

Pozn.: Vo viacerých metrikách vyskočili hraničné hodnoty pre SystemVerilog HDL parser. Jedná sa o libku, ktorú nebudem upravovať.

SourceMonitor

Analýza metrík pomocou nástroju SourceMonitor. Pred meraním som použil príkaz mvn clean, aby som prečistil kód od automaticky generovaného kódu, aby mi neovplyvnilo merania. Pre potreby analýzy som si vybral nasledovné metriky (v hranatých zátvorkách je interval optimálnych hodnôt na základe SourceMonitora).

1. Analýza LoC (počet fyzických riadkov)
2. Analýza Statements (počet logických riadkov)
3. Analýza % Comments [8-20]
4. Analýza Methods/Class [4-16]
5. Analýza Max Complexity [2-8]
6. Analýza Max Depth [3-7]

Analýza LoC:

Nie je presne zadefinované, aký je optimálny počet fyzických riadkov pre jednu triedu. Na základe internetových článkov sa odporúča pre OOP nemať viac ako 1000 riadkov v jednej triede. Najväčšia trieda je testovacia trieda ClassroomServiceImplTest. Trieda obsahuje jednotkové testy k triede ClassroomServiceImpl, ktorá je sama o sebe objemnejšia. V tomto momente nepovažujem za potrebné rozdelenie triedy na viaceré triedy.

File Name	Lines
parser\target\generated-sources\antlr4\org\asicde\asicde\parser\SysVerlogHDLParser.java	27,123
parser\target\generated-sources\antlr4\org\asicde\asicde\parser\SysVerlogHDLBaseListener.java	5,379
parser\target\generated-sources\antlr4\org\asicde\asicde\parser\SysVerlogHDLListener.java	4,460
parser\target\generated-sources\antlr4\org\asicde\asicde\parser\SysVerlogHDLLEXer.java	1,048
core\src\test\java\org\asicde\asicde\core\service\impl\ClassroomServiceImplTest.java	591
core\src\main\java\org\asicde\asicde\core\service\impl\ClassroomServiceImpl.java	403
parser\src\main\java\org\asicde\asicde\parser\parser\SystemVerlogParser.java	400
core\src\test\java\org\asicde\asicde\core\service\impl\UserServiceImplTest.java	378
core\src\test\java\org\asicde\asicde\core\service\impl\TeamServiceImplTest.java	315
core\src\main\java\org\asicde\asicde\core\service\impl\UserServiceImpl.java	295

Obrázok 1 - Analýza LoC

Analýza Statements:

Tak ako pri analýze LoC, nie je presne zadefinovaný optimálny počet statementov pre jednu triedu. Najvyššie zastúpenie statementov má pochopiteľne trieda ClassroomServiceImplTest, kde aktuálne nepovažujem za nutné rozdeľovať na viacero menších subtried.

File Name	Lines	Statements
parser\target\generated-sources\antlr4\org\asicde\asicde\parser\SysVerlogHDLParser.java	27,123	19,866
parser\target\generated-sources\antlr4\org\asicde\asicde\parser\SysVerlogHDLBaseListener.java	5,379	1,793
parser\target\generated-sources\antlr4\org\asicde\asicde\parser\SysVerlogHDLListener.java	4,460	893
core\src\test\java\org\asicde\asicde\core\service\impl\ClassroomServiceImplTest.java	591	434
parser\src\main\java\org\asicde\asicde\parser\parser\SystemVerlogParser.java	400	306
core\src\test\java\org\asicde\asicde\core\service\impl\UserServiceImplTest.java	378	263
core\src\main\java\org\asicde\asicde\core\service\impl\ClassroomServiceImpl.java	403	259
core\src\test\java\org\asicde\asicde\core\service\impl\TeamServiceImplTest.java	315	224
core\src\test\java\org\asicde\asicde\core\service\impl\RepoServiceImplTest.java	275	194
core\src\main\java\org\asicde\asicde\core\service\impl\UserServiceImpl.java	295	189

Obrázok 2 - Analýza Statements

Analýza % Comments:

Kiviath graf ukazuje optimálne % zastúpenie komentárov v triede v intervale 8-20%. Ako môžeme vidieť na Obr. 3 a 4, projekt obsahuje triedy s nadmerným percentuálnym zastúpením komentárov, ako aj triedy (väčšina), ktoré neobsahujú žiaden komentár.

Vysoké % komentárov (Obr. 3):

- Triedy v balíku /service/ a /enumeration/ sú interface-y, pričom k definícií metódy je vygenerovaný JavaDoc
- Triedy UserIT, ReposIT je testovacia trieda, kde sú zakomentované telá testovacích metód

Nízke % komentárov (Obr. 4):

- Takmer 50 tried v celom moduli backend obsahuje menej ako 8% komentárov

File Name	% Comments ▾
parser\target\generated-sources\antlr4\org\asicde\asicde\parser\SysVerilogHDLBaseListener.java	83.2
parser\target\generated-sources\antlr4\org\asicde\asicde\parser\SysVerilogHDLListener.java	79.9
core\src\main\java\org\asicde\asicde\core\service\StorageService.java	61.8
core\src\main\java\org\asicde\asicde\core\service\UserService.java	49.1
core\src\test\java\org\asicde\asicde\core\integration_test\UserIT.java	41.9
core\src\main\java\org\asicde\asicde\core\security\service\AuthenticationService.java	40.0
core\src\test\java\org\asicde\asicde\core\integration_test\ReposIT.java	39.4
core\src\main\java\org\asicde\asicde\core\persistence\entity\enumeration\OrganizationInviteStatus.java	30.0
core\src\main\java\org\asicde\asicde\core\persistence\entity\enumeration\UserTeamPermission.java	30.0
core\src\main\java\org\asicde\asicde\core\service\RepoValidationService.java	28.6
core\src\main\java\org\asicde\asicde\core\config\CoreGeneralConfig.java	23.1

Obrázok 3 - Analýza % Comments zostupne

File Name	% Comments ▲
core\src\test\java\org\asicde\asicde\core\api\model\form\AddTeacherToClassroomFormFactory.java	0.0
core\src\test\java\org\asicde\asicde\core\api\model\form\TeamCreateFormFactory.java	0.0
core\src\main\java\org\asicde\asicde\core\web\mapper\ChatMembersMapper.java	0.0
core\src\main\java\org\asicde\asicde\core\web\mapper\ChatMessageMapper.java	0.0
core\src\main\java\org\asicde\asicde\core\web\mapper\OrganizationInviteMapper.java	0.0
core\src\main\java\org\asicde\asicde\core\web\mapper\OrganizationMapper.java	0.0
parser\src\main\java\org\asicde\asicde\parser\parser\model\Connection.java	0.0
parser\src\main\java\org\asicde\asicde\parser\parser\model\ParsedCode.java	0.0
commons-error-handling\src\main\java\org\asicde\asicde\commons\error\handling\exception\AuthenticationException.java	0.0
commons-error-handling\src\main\java\org\asicde\asicde\commons\error\handling\exception\ReferentialIntegrityViolationException.java	0.0

Obrázok 4 - Analýza % Comments vzostupne

Analýza Methods/Class:

SourceMonitor definuje optimálny interval pomeru Methods/Class od 4 – 16. Na Obr. 5 a 6 je vidno, že projekt obsahuje zopár tried s väčším množstvom tried, ktoré majú vysoký počet metód. Avšak viac ako polovica tried obsahuje menej ako 4 metódy, ba niekoľko desiatok tried neobsahuje ani jednu metódu, viď Obr. 6.

Vysoké % Methods/Class (Obr. 5):

- Triedy, ktoré presahujú hranicu 16 metód, nie sú v tomto prípade kritické, keďže odchýlka od hornej hranice nie je príliš vysoká
- Testovacie triedy na triedy obsahujúce približne hornú hranicu metód taktiež obsahujú väčší počet jednotkových testov (testovacích metód)

Nízke % Methods/Class (Obr. 6):

- Triedy typu Entity či DTO vo frameworku SpringBoot definujú atribúty databázovej tabuľky, preto neobsahujú metódy
- Triedy so sufixom Form taktiež definujú atribúty formuláru, preto neobsahujú žiadne metódy
- Identifikoval som aj triedy v priečinkoch /model/, ktoré tiež iba definujú premenné
- Triedy, ktoré extendujú iné triedy pre dodržanie SpringBoot konvencie
- Utility triedy, ktoré slúžia ako namespace pre iné funkcie, Mapper Triedy, Exception triedy a pod.

- Identifikoval som aj zbytočné triedy, ktoré nemajú využitie a môžu byť zmazané (OrganizationInviteRepository – 0 metód)

File Name	Methods/Class ▾
parser\target\generated-sources\antlr4\org\asicde\asicde\parser\SysVerilogHDLBaseListener.java	894.00
parser\target\generated-sources\antlr4\org\asicde\asicde\parser\SysVerilogHDLListener.java	890.00
core\src\test\java\org\asicde\asicde\core\service\impl\ClassroomServiceImplTest.java	30.00
core\src\main\java\org\asicde\asicde\core\service\impl\ClassroomServiceImpl.java	23.00
core\src\main\java\org\asicde\asicde\core\service\impl\UserServiceImpl.java	21.00
core\src\test\java\org\asicde\asicde\core\service\impl\UserServiceImplTest.java	21.00
core\src\main\java\org\asicde\asicde\core\service\impl\TeamServiceImpl.java	17.00
core\src\main\java\org\asicde\asicde\core\web\ClassroomController.java	17.00

Obrázok 5 - Analýza Methods/Class zostupne

File Name	Methods/Class ▲
core\src\main\java\org\asicde\asicde\core\persistence\entity\Assignment.java	0.00
core\src\main\java\org\asicde\asicde\core\model\UserDto.java	0.00
core\src\main\java\org\asicde\asicde\core\persistence\entity\Classroom.java	0.00
core\src\main\java\org\asicde\asicde\core\persistence\entity\Organization.java	0.00
core\src\main\java\org\asicde\asicde\core\persistence\entity\Repo.java	0.00
core\src\main\java\org\asicde\asicde\core\persistence\entity\ChatMessage.java	0.00
core\src\main\java\org\asicde\asicde\core\persistence\entity\TeamMembership.java	0.00
core\src\main\java\org\asicde\asicde\core\persistence\entity\OrganizationInvite.java	0.00
core\src\main\java\org\asicde\asicde\core\persistence\entity\BlacklistedToken.java	0.00
core\src\main\java\org\asicde\asicde\core\model\AbstractAuditedDto.java	0.00

Obrázok 6 - Analýza Methods/Class vzostupne

Analýza Complexity:

SourceMonitor definuje optimálny interval pre komplexitu od 2 do 8. Iba trieda OrganizationServiceImpl má väčšiu komplexitu ako 8 (konkrétne 9), čo nie je veľká odchylka. Väčšina tried ale má komplexitu menšiu ako 2, čo je dolná hranica optimálneho intervalu. Dôvod je rovnaký ako nízke percento počtu metód na triedu, kedy SpringBoot obsahuje aj malé triedy, ktoré majú iba definičnú funkciu, preto majú nízku komplexitu.

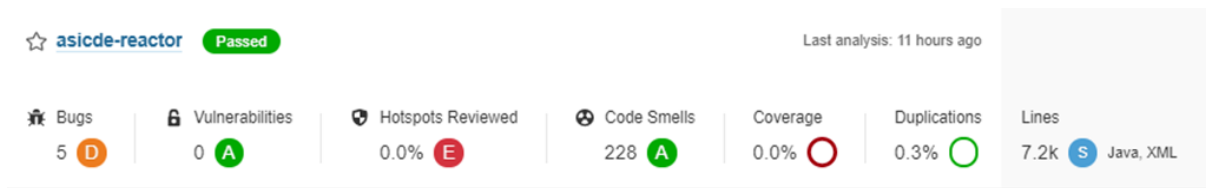
Analýza Max Depth:

SourceMonitor definuje optimálny interval pre maximálnu hĺbku 3-7. Najvyššia hĺbka v analyzovanom projekte je 5, čo nepredstavuje problém, avšak triedy ako entity, dto, form a iné, ktoré majú za úlohu definovať premenné či konštanty majú podpriemernú hodnotu maximálnej hĺbky, čo však aktuálne nepoukazuje na nekvalitný kód.

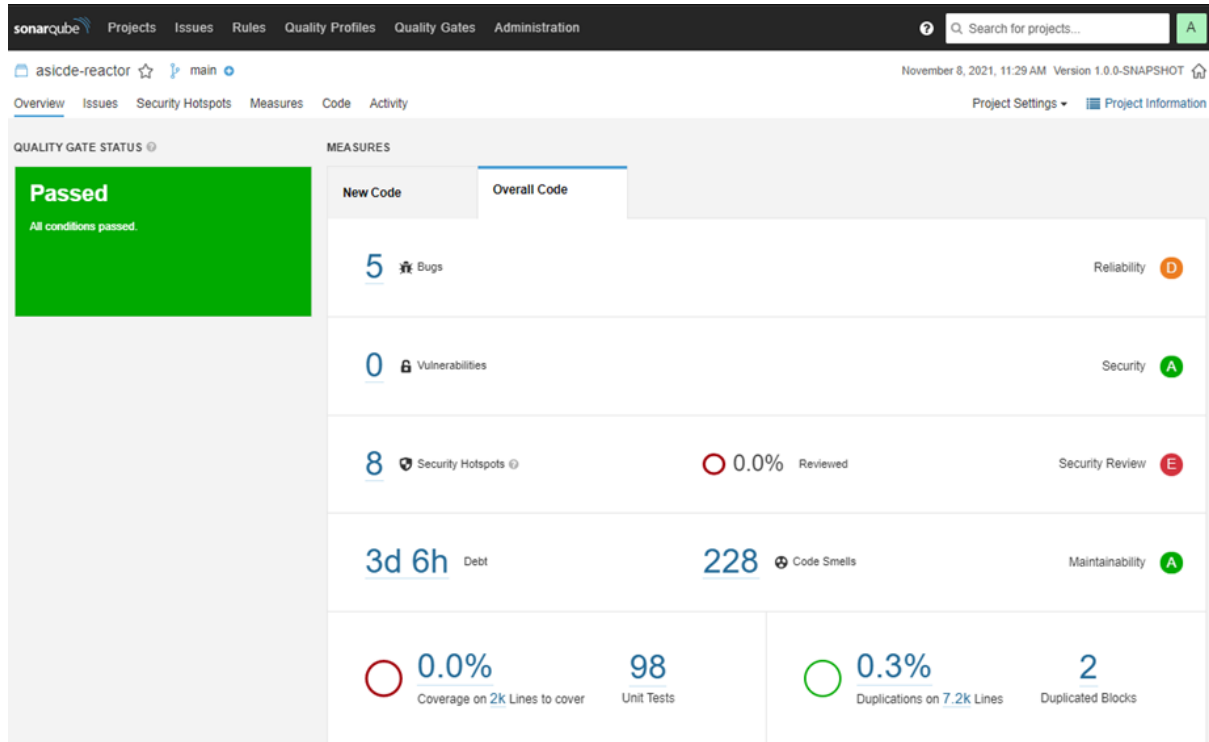
SonarQube

Spojzdnil som nástroj na meranie kvality kódu SonarQube nad modulom backend projektu ASICDE. Využil som community verziu, ktorá umožňuje meranie kvality len nad master vetvou, preto som vytvoril nový repozitár, kde som naklonoval dev vetvu ako master. V tomto bode bolo potrebné vytvoriť GitHub app, ktorá umožnila prepojenie repozitára s localhost verziou Sonaru. Na získanie metrík bolo potrebné spárovať projekt s tokenom zo SonarCube cez Maven.

Po prepojení Sonaru s repositárom, nástroj s default nastaveniami ohodnotil backend modul nasledovne:

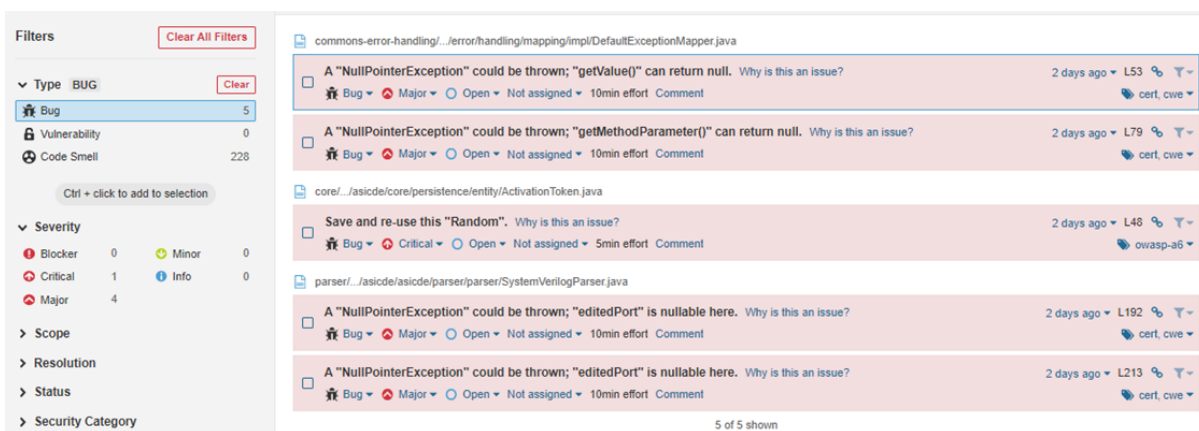


Obrázok 7 - SonarQube project



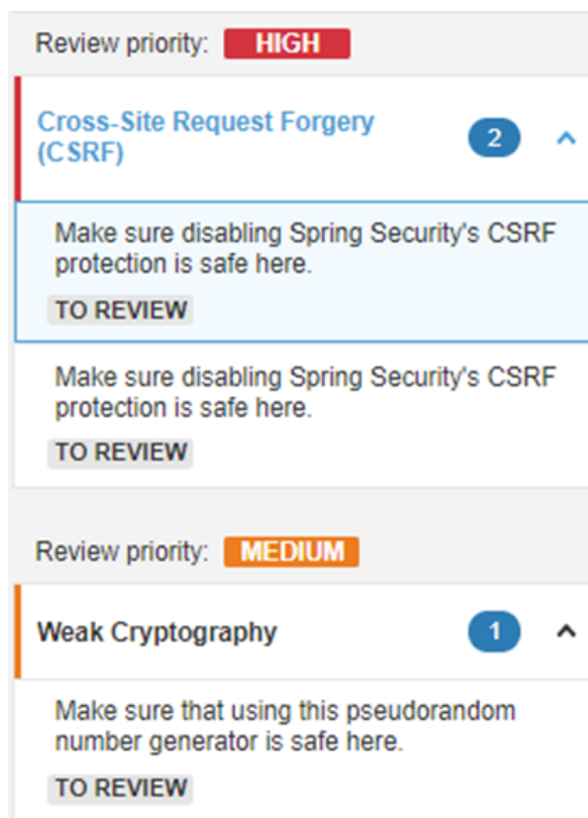
Obrázok 8 - SonarQube výsledky metrik

Reliability – Bugs:

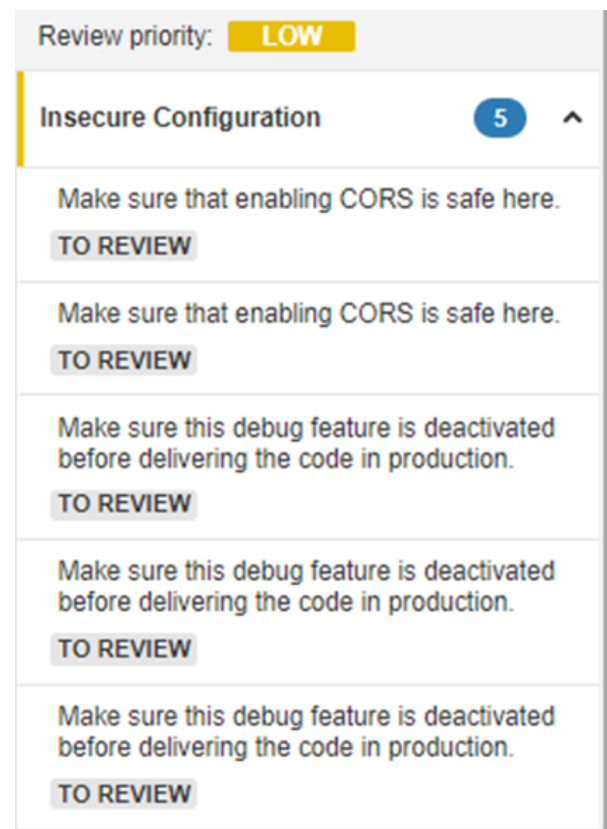


Obrázok 9 - SonarQube Reliability metric

Security HotSpots:



Obrázok 10 - Security Hotspots with high and medium priority



Obrázok 11 - Security Hotspots with low priority

Code Smells:

Sonar identifikoval 228 code smells o približnej práci na opravu 3 dní a 6 hodín

Tests:

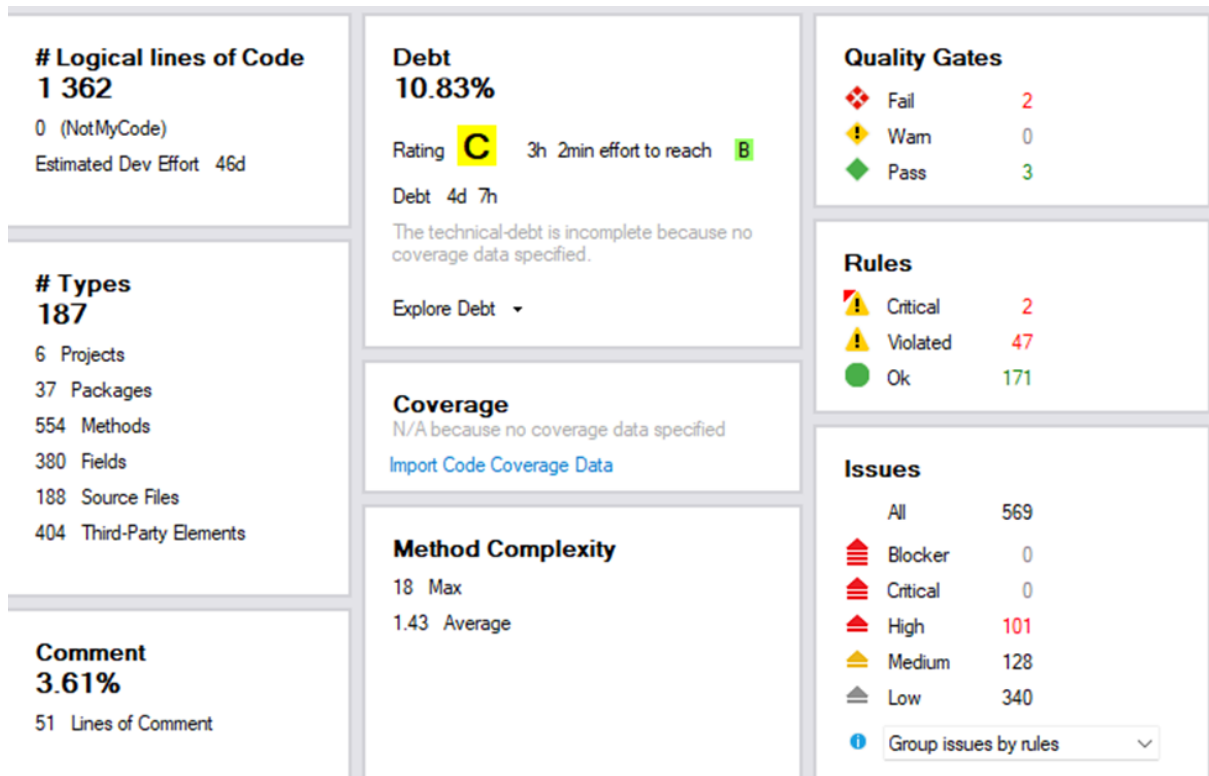
Projekt má veľmi slabé zastúpenie testov, čo výrazne znižuje jeho kvalitu.

JArchitect:

Zaregistroval som free trial verziu na 14 dní, inak platený nástroj. Veľmi ľahká integrácia k projektu cez mavenovský pom.xml. Ponúka širokú škálu metrík a quality atribútov.

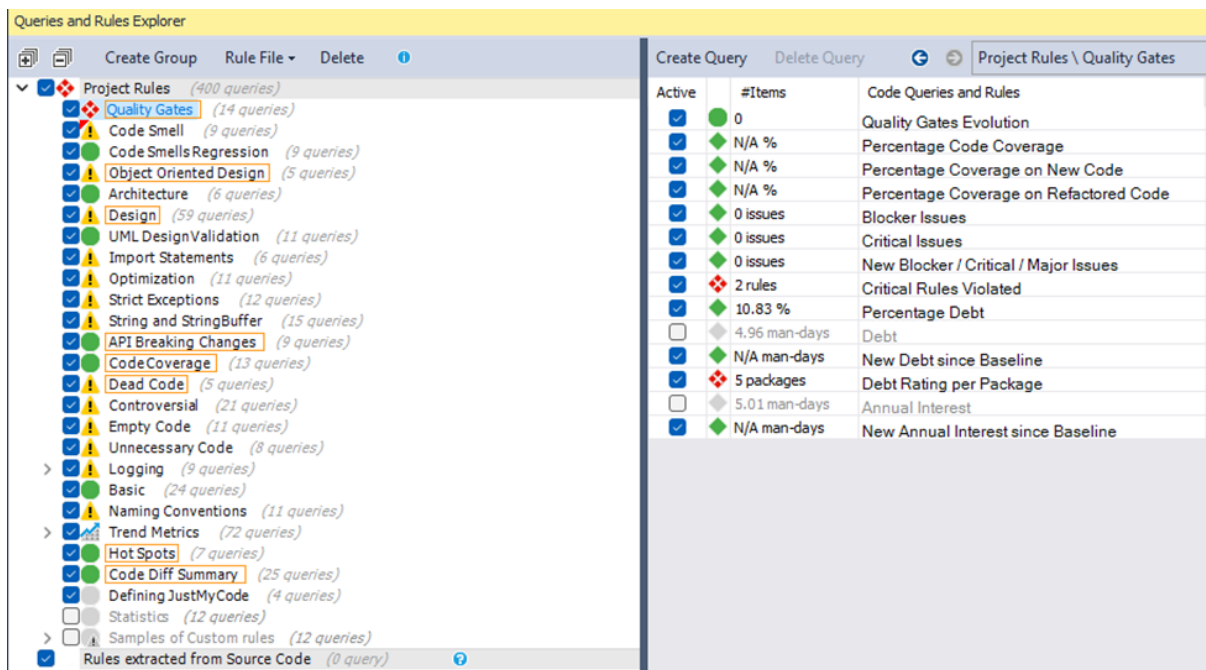
Analýza dashboardu:

- Iba 51 riadkov komentárov, celkovo 3,61% (neberie v úvahu JavaDoc!)
- Debt odhadovaný na takmer 11%, odhadovaný čas riešenia 4dni a 7 hodín
- Priemerná komplexita na úrovni 1,43, pričom spodná hranica zadaná SourceMonitorom je 2 čiže nie tak zlé ak si vezmem do úvahy veľa metód, ktoré definujú nejaké atribúty (dôsledok využitia SpringBootu)
- Narušenia quality gates, rules a veľmi veľa issues



Obrázok 12 - JArchitect dashboard

Explorer obsahuje rýchle a efektívne vyhľadávanie problémov na základe dôležitosti, kedy sa dokážem preklikať až k zdrojovému kódu problému:



Obrázok 13 - JArchitect Explorer

Ohodnotenie projektu na základe cyclomatic complexity (CC):



Obrázok 14 - CC graf pre projekt

Ohodnotenie projektu na úrovni metód na základe CC:



Obrázok 15 - CC na úrovni metód

Záver analýzy

- **SourceMonitor** nám poskytuje základe softvérové metriky. Analýza neodhalila výrazné odchýlené hodnoty. LoC a počet statements ukazovali vcelku fajn hodnoty, percentuálne zastúpenie komentárov (hlavne JavDoc) by sa malo výrazne vylepšiť, metrika methods per class z hornej hranice neobjavila veľké odchýlky od priemerných hodnôt, z dolnej hranice som identifikovali veľa odchýliek spôsobených štruktúrou súborom SpringBootu. Analýza metrick CC a max depth nám taktiež neodhalila z hornej hranice výrazne vychýľujúce sa hodnoty, z dolnej hranice sa objavili prípady z dôvodu spomenutého pri metrike methods per class.

- **SonarQube** objavil v našom kóde zopár kritických bugov, zopár bezpečnostných dier ale nič hrozného. Nekvalita kódu sa objavila hlavne pri code smelloch, ktorých sonar identifikoval 228 a pri malom počte testov.
- **JArchitect** obsahuje metriky, ktoré obsahuje aj SourceMonitor, taktiež identifikuje rôzne problémy, bugy a iné zdroje nekvalitného kódu. Z tohto nástroja som prioritne využíval vizualizácie pre rýchlejšiu a efektívnejšiu identifikáciu nekvalitných častí kódu za pomoci metrick analyzovaných pomocou SourceMonitora.

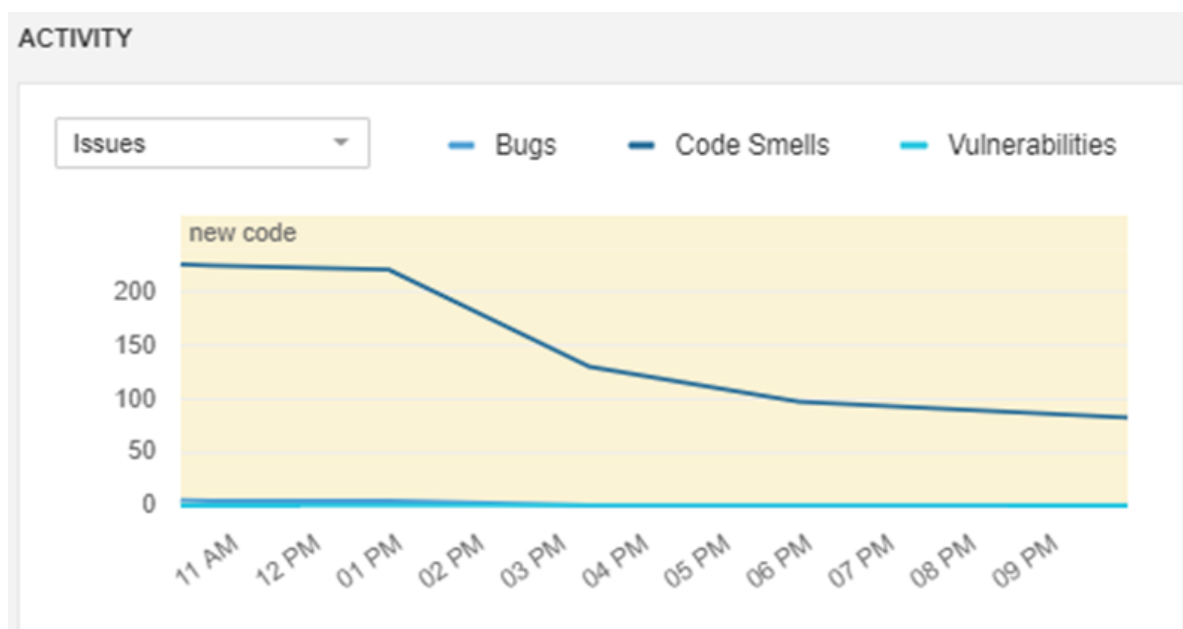
Návrh implementácie

Keďže nástroje SourceMonitor či JArchitect neobjavili výrazné odchýlky vo vybraných metrikách, rozhodol som sa skvalitniť kód prioritne na základe identifikovaných nekvalít kódu nástrojom SonarQube nasledovne:

- Opraviť identifikované bugy
- Ošetriť security hotspots
- Vyriešiť čo najviac code smellov na zlepšenie celkovej kvality kódu

Výsledky implementácie a porovnanie

- V analýze som identifikoval 5 bugov, z toho dva boli false positive, 2 sa týkali libky verilog parseru, 1 bol vyriešený
- V analýze som identifikoval 8 security hotspotov, z toho dva som aktuálne nevedel vyriešiť, preto som ich dočasne označil ako safe, zvyšných 6 bolo vyriešených
- V analýze som identifikoval 228 code smellov rôzneho druhu a pôvodu, z toho bolo vyriešených 146, 82 ostáva otvorených

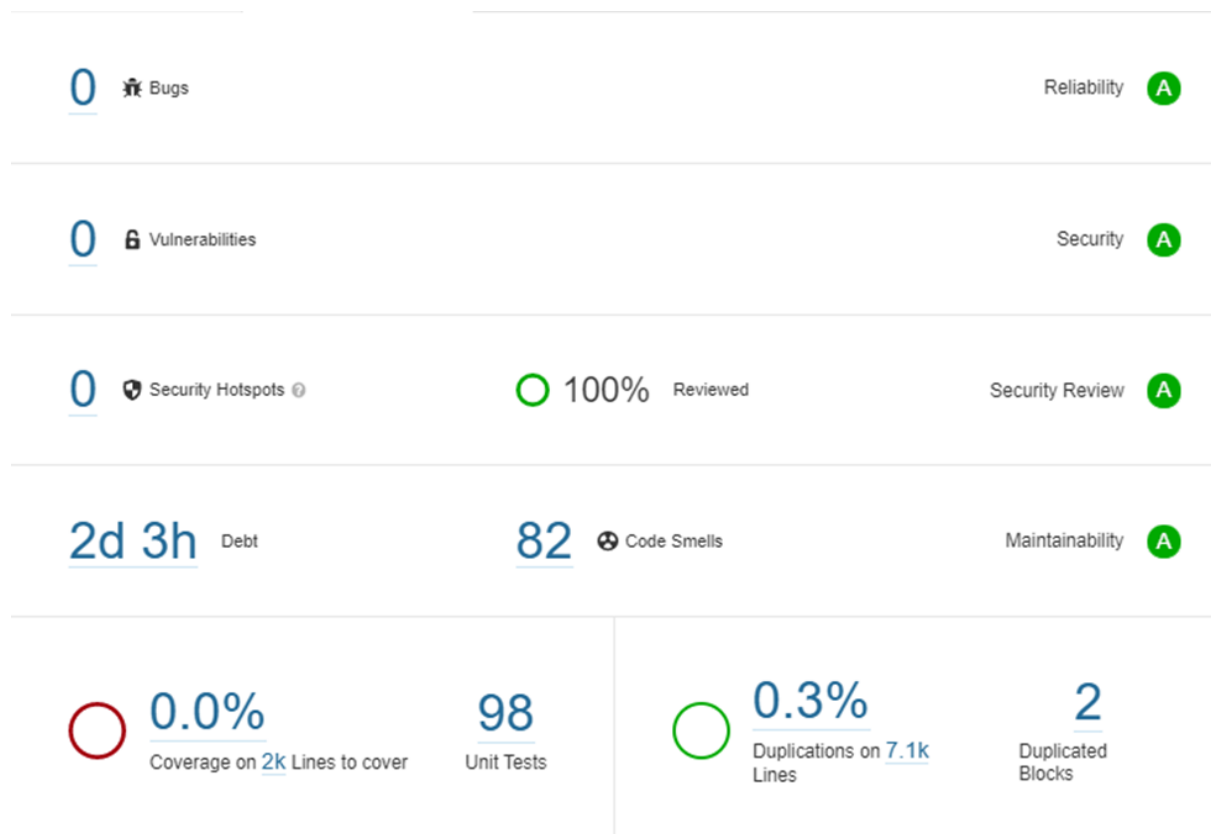


Obrázok 16 - SonarQube activity graph

Záver

Pre môj projekt sú dôležité nasledovné metriky:

- LoC, Statements – keďže projekt implementujú študenti v rámci TP a záverečných prác, je dôležité aby bol kód prehľadný a ľahko udržiavateľný, preto je dôležité zabezpečiť primeraný obsah jednotlivých tried či metód
- Methods/Class, CC, Max Depth – dôležité sledovať hlavne horné hranice prijateľného intervalu
- Bugs a Code Smells zo Sonaru – pre lepší rozvoj projektu



Obrázok 17 - SonarQube dashboard po refaktoringu

Riadenie projektu

Úvod

Keďže cieľom tohto predmetu je aj oboznámenie sa s technikami pri riadení softvérových projektov a nie len vytvorenie funkčného prototypu, v našom projekte využívame riadenie na základe metodiky Scrum. Podľa tejto metodiky sme riešili prácu na projekte v takzvaných šprintoch. Naše šprinty trvali 2 týždne, pričom zahájenie prvého šprintu sme museli posunúť o týždeň, nakoľko sme v prvej fáze semestra riešili spoznanie prostredia, prístup k virtuálnemu stroju a naštudovanie si dokumentácií. V každom šprinte sme určili ciele a úlohy, ktoré chceme splniť a rozdelili sme si ich medzi sebou tak, aby sme mohli pracovať čo najefektívnejšie.

Ešte pred začiatkom prvého šprintu sme vytvorili niekoľko metodík ktoré určujú postupy a spôsoby pri vykonaní jednotlivých úloh. Tieto metodiky ako je Jira, GitHub a komunikačné prostriedky, sú opísané v tomto dokumente.

Počas semestra sme mali dve stretnutia každý týždeň, z ktorých sme sa raz stretávali s vedúcim projektu a raz ako tím. Popritom sme neustále komunikovali pomocou komunikačných prostriedkov ohľadom progresu, nejasností a samotného stavu taskov. Stretnutia s vedúcim sme mali zaznamenávané a pritom sme mali zvoleného jedného zapisovateľa, ktorý samotný priebeh stretnutí zapisoval. Všetky dokumenty a materiály sa nachádzajú na webovom sídle tímu.

Tento dokument obsahuje opis riadenia tímového projektu, role jednotlivcov tímu, sumarizáciu šprintov, aplikácie manažmentov, jednotlivé metodiky a globálnu retrospektívu.

Role členov

Jasmína Badová



Frontend

- Prototypovanie Features podstránky
- Implementácia Features podstránky
- Refaktoring

Zapisovateľka

- Zapisovanie poznámok z konzultácií s vedúcim

Dokumentácia

- Vytvorenie štruktúry
- Prílohy/Frontend prototypy/Features
- Metodiky/Komunikácia

Paula Kollárová



Backend

- Úprava a doladenie funkcionality pozvania do organizácie
- Refaktoring

Dokumentácia

- Sumarizácia šprintov/Šprint 2
- API dokumentácia

Ladislav Leikes



Scrum Master

- Vytvorenie taskov
- Pridelenie taskov
- Vytvorenie planu
- Export úloh

Frontend

- Responzívny Homepage
- Prototypovanie Pricing podstránky
- Implementácia Pricing podstránky
- Úprava UI pre privátne repozitáre
- Refaktoring
- Spísanie bugov
- Sortovanie privátnych repozitárov
- Úprava UI v organizácii
- Pridanie tabov All/Favorite/Archived pre privátne repozitáre

Dokumentácia

- Ciele pre zimný semester
- Prílohy/Frontend prototypy/Pricing
- Role členov
- Aplikácie manažmentov/Manažment správy úloh
- Metodiky/Jira
- Export evidencie úloh

Matej Janeček



Server Admin

- Spojazdnenie tímovej web stránky na serveri
- Spojazdnenie ASICDE na serveri
- Umožnenie autodeployu
- Spravovanie serveru

Frontend

- Refaktoring
- Mergovanie branchí na GitHube
- Úprava repozitárov v organizácii
- Spísanie bugov

Dokumentácia

- Aplikácie manažmentov/Verziovanie zdrojového kódu
- Metodiky/GitHub

Eduard Krajčír



Web Admin

- Vytvorenie web stránky pre tím
- Spravovanie web stránky

Frontend

- Prototypovanie About Us podstránky
- Implementácia About Us podstránky
- Refaktoring

Dokumentácia

- Špecifikácia požiadaviek/Nefunkčné požiadavky
- Prílohy/Frontend prototypy/About Us
- Globálna retrospektíva

Marek Benovič



Backend

- Vytvorenie funkcionality pre zabudnuté heslo
- Refaktoring

Dokumentácia

- Špecifikácia požiadaviek/Funkčné požiadavky
- Sumarizácia šprintov/Šprint 1
- API dokumentácia

Denis Piovarči



Backend

- Umožnenie prihlásenia sa aj pomocou e-mailu
- Refaktoring
- Optimalizácia
- Umožnenie spustenia BE bez Dockera

Dokumentácia

- Sumarizácia šprintov/Šprint 3
- API dokumentácia
- Príloha B - Merania kvality softvéru ASICDE

Aplikácie manažmentov

Verziovanie zdrojového kódu

Aby sme mohli na našom projekte efektívne pracovať a sledovať zmeny v kóde, je potrebné používať verziovací softvér. V našom prípade sme dostali do správy už existujúce repozitáre v platforme Github s plným prístupom. Do týchto repozitárov mali prístup aj diplomanti. Preto sme sa rozhodli vytvoriť si vlastné branche, aby nedošlo ku kolíziám. Taktiež ostalo zopár nemergnutých vetiev po našich predchodcoch. Tie sme museli pred začiatkom našej práce mergnúť. Viac o tom ako sme pracovali s Gitom a Githubom je popísané v sekcii metodiky. Viac o danej metodike sa nachádza v časti *Metodiky - GitHub*.

Manažment správy úloh

Pre úspešné manažovanie úloh sme zvolili portál Jira od Atlassianu. Daný portál ponúka mnoho funkcionalít pre úspešné riadenie tímových projektov, ktoré praktizujú agilnú metódu, ale v našom prípade sme sa najmä zamerali na správu úloh a šprintov. Pomocou Jiry dokážeme pridávať úlohy, meniť ich stav, ohodnotiť ich a prideliť ich jednotlivým osobám. Stav jednotlivých úloh sa môžu na mieru preddefinovať, pričom my sme si zadefinovali 4 typy stavov. Viac o danej metodike sa nachádza v časti *Metodiky - Jira*.

Komunikácia

Komunikácia prebiehala na stretnutiach tímu s vedúcim projektu a medzi členmi tímu. V dôsledku pandemickej situácie sme využívali na stretnutia s vedúcim tímu platformu Microsoft Teams. Priebeh stretnutí pozostával z odprezentovania progresu aplikácie, úloh ktoré sme splnili od posledného stretnutia a taktiež úlohy, na ktorých sa aktuálne pracuje. Predstavili sme súčasný stav projektu a spísali sme prípadné pripomienky, ktoré sme premenili v tasky a následne zaradili do zoznamu úloh. Ďalším predmetom stretnutí bolo predstaviť, na čom plánujeme v ďalšom týždni pracovať. V prípade nejasností sme diskutovali a riešili problémy, ktoré vznikli počas práce na úlohách.

Vnútoraná tímová komunikácia bola zabezpečená prostredníctvom platformy Discord, kde sme v rámci tímového serveru, založili viacero kanálov pre efektívnejšiu komunikáciu. Discord sme využívali hlavne na stretnutia tímu, riešenie problémov, či informovanie o progresu a neodkladných informáciách. Okrem toho, v prípade potreby členovia tímu komunikovali medzi sebou v súkromných chatoch. Na spojenie s ľuďmi, ktorí doteraz pracovali na tomto projekte, sme používali skupiny vytvorené v aplikácii Microsoft Teams. Detailnejšie je táto časť opísaná v Metodike komunikácie a správy dát.

Sumarizácia šprintov

Popis jednotlivých šprintov a ich sumarizácia.

Šprint 1

Trvanie šprintu: 11.10. 2021 - 25.10. 2021

Vzhľadom na to, že rozvíjame už existujúcu aplikáciu, tak našim prvotným cieľom bolo rozbehnutie aplikácie a pochopenie jej častí. Aplikácia obsahovala dostatočné inštrukcie na rozbehnutie jej častí, avšak zvyšné časti aplikácie neboli poriadne zdokumentované. Pre túto skutočnosť bolo potrebné intenzívnejšie študovať kód. Okrem venovania sa samotnej aplikácii sme vytvorili stránku projektu a server, kde beží demoverzia aplikácie. Bolo potrebné spísať existujúce bugy a neexistujúce funkcionality aby sme vedeli na čom bude treba pracovať. Ďalej bola aplikácia vylepšená po FE stránke. Spravili sme návrh viacerých stránok (About us, Pricing) a spravili web responzívnym.

Retrospektíva

Čo sme robili dobre?

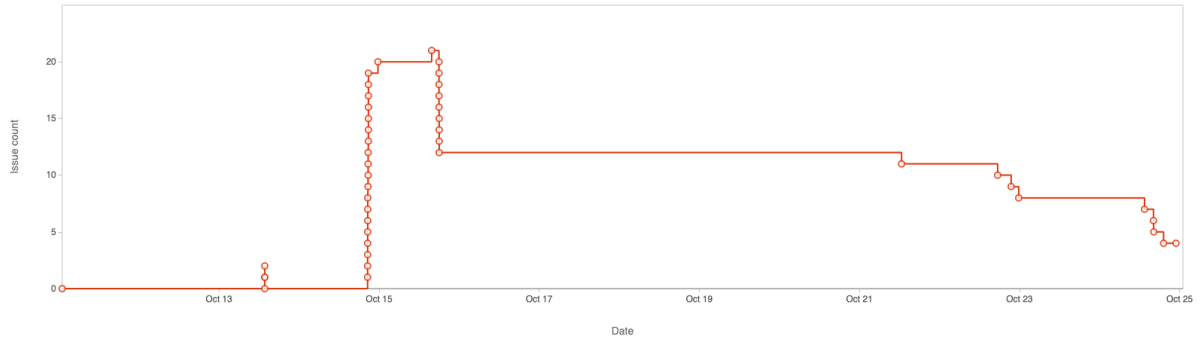
- Rozdelili sme tím na FE a BE tak, aby sme čo najlepšie využili schopnosti každého člena tímu
- Identifikovali sme problémy v aplikácii a pripravili sme ich na riešenie v ďalších šprintoch
- Vytvorili sme platformu na komunikáciu - Discord server
 - Vieme sa tu rýchlo dohodnúť a kontaktovať v prípade potreby (individuálny ľudia ale aj skupiny s určitou zodpovednosťou, napr. FE, BE...)

Čo by sme mohli zlepšiť?

- Väčšina taskov v šprinte bola zameraných na FE
- Rozbehnutie BE trvalo dlhšie vzhľadom na to, že nastali komplikácie pri operačnom systéme Windows
- Podcenenie existujúcej dokumentácie. Na prvý pohľad sa zdá obširna ale chýbajú v nej vitálne informácie
- Ujasnenie informácií ohľadom hodnotenia pomocou story points

Zhrnutie

Zoznámili sme sa s aplikáciou a identifikovali sme existujúce problémy. Progres bolo vidieť vďaka práci na FE (responzivita...). BE progres nebol zanedbateľný vzhľadom na to, že sme narazili na viacero problémov pri pokuse o jeho spustenie. Aj napriek komplikáciám sa nám podarilo splniť väčšinu nami stanovených úloh a zvyšné boli presunuté do ďalšieho šprintu..



Incomplete issues

[View in issue navigator](#)

Key :	Summary :	Issue type :	Epic :	Status :	Assignee :	Issue count
ASICDE2021-12	Spojzdzenie backendu na serveri	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-22	Umoznenie prihlasenia sa aj pomocou emailu	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-27	Umoznenie pouzitia klavesovych skratiek na navigovanie sa vo webovom prostredi	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-37	Vytvorit "forgot password" funkciu	<input checked="" type="checkbox"/> Task		DONE		1

Completed issues

[View in issue navigator](#)

Key :	Summary :	Issue type :	Epic :	Status :	Assignee :	Issue count
ASICDE2021-10	Vytvorenie web stranky pre tim	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-13	Aktualizacia timovej web stranky	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-14	Vytvorenie contentu na timovu web stranku	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-15	Spojzdzenie timovej web stranky na VM	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-16	Zapisovanie poznamok z konzultacii	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-17	Zapisovanie poznamok z konzultacii #2	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-18	Vytvorenie taskov a planu na Jira	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-33	Spojzdzenie ASICDE aplikacie na OS Windows	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-11	Lokalne spojzdzenie aplikacie	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-9	Spisanie bugov	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-19	Prestudovanie existujucich zdrojov a oboznamenie sa s prostredim	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-21	Spraviti homepage stranku na responzivnu	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-23	Navrhnut About us pod-stranku	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-24	Navrhnut Features pod-stranku	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-25	Navrhnut Pricing pod-stranku	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-26	Umozniť otvorenie rezpozitara aj pomocou kliknutia na kartu	<input checked="" type="checkbox"/> Task		DONE		1
ASICDE2021-34	Ohodnotenie jednotlivych taskov podla casovej narocnosti	<input checked="" type="checkbox"/> Task		DONE		1

Issues completed outside of sprint

Key :	Summary :	Issue type :	Epic :	Status :	Assignee :	Issue count
-------	-----------	--------------	--------	----------	------------	-------------



No issues have been completed outside of the sprint

Šprint 2

Trvanie šprintu: 25.10. 2021 - 08.11. 2021

V 2. šprinte sme okrem odstraňovania bugov opravovali funkcie, ktoré nefungovali, fungovali iba lokálne (teda miesto práce s databázou sa dáta zapisovali iba do lokálnych premenných bez permanentného uchovávanía) alebo nepracovali tak ako mali. Zároveň sme potrebné funkcie aj dopĺňali, nakoľko veľa FE elementov nevykonávalo žiadnu akciu, ktorá bola pre správne fungovanie nevyhnutá a mala byť už hotová. Tým pádom sa nielen dopĺňali funkcie v BE ale aj nové elementy na FE (napríklad akcie po zakliknutí indikátora), čím sme stratili veľa času, ktorý sme plánovali venovať implementácii novej funkcionality. Okrem tohto sme opravovali ďalšie problémy s responzivitou, implementovali sme na prvotné verzie niektorých podstránok (About us, Pricing, Features) a podarilo sa nám spustiť BE na serveri. Umožnili sme zároveň prihlásenie používateľa pomocou emailu, čo bola tiež jedna z nefunkčných akcií a rovnako sme opravili zobrazenie korektnej chybovej hlášky pri neakceptácii hesla a vytvorení funkcie obnovy zabudnutého hesla. Vylepšili sme spôsob navigácie vo webovom rozhraní pomocou klávesových skratiek a uľahčili sme návrat z vizualizačného modulu naspäť do repozitárov. Pokračovali sme aj v refaktoringu kódu.

Retrospektíva

Čo sme robili dobre?

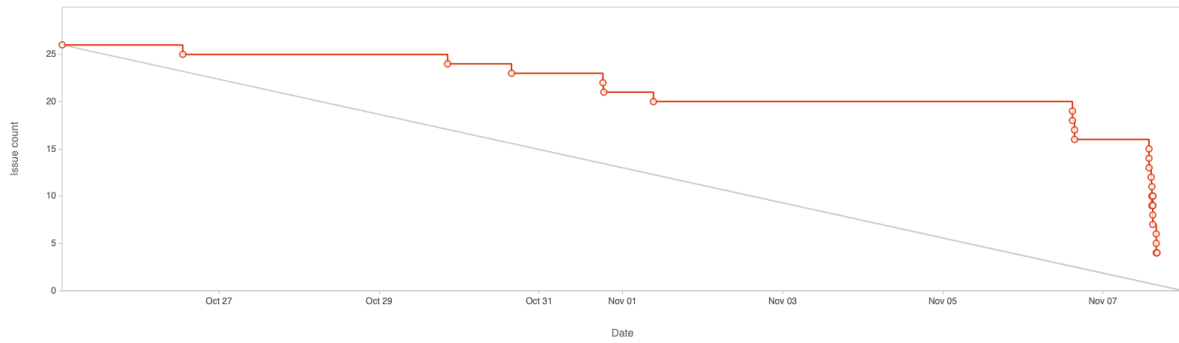
- Pomerne dobre sme si rozvrhli počet taskov vzhľadom na ich náročnosť
- Vďaka iniciatíve jedného z členov BE tímu nám bolo umožnené pracovať bez spúšťania Dockeru
- Vďaka iniciatíve jedného z členov FE tímu sme si dokázali určiť termíny stretnutí a vedeli sme sa tomuto termínu prispôbiť
- S pochopením sme konzultovali svoje progresy a problémy

Čo by sme mohli zlepšiť?

- Komunikáciu medzi tímom BE a FE kvôli prepájaniu FE elementov s BE funkciami
- Priebežnejšie dokumentovať každú zmenu (na malé zmeny sa ľahko zabúda)

Zhrnutie

Progres na FE je veľmi viditeľný nie len úpravou existujúcich elementov ale najmä pridaním nových elementov. Progres na BE sa uskutočnil takisto ale je menej viditeľný, nakoľko ešte neboli funkcie prepojené s FE, teda sa ich aplikácia na FE neprejavila (čiže nebola pre používateľov viditeľná). Väčšinu taskov sa nám podarilo počas šprintu úspešne dokončiť a ich ohodnotenie pomerne dobre sedelo s reálnou náročnosťou.



Incomplete issues

[View in issue navigator](#)

Key :	Summary :	Issue type :	Epic :	Status :	Assignee :	Issue count
ASICDE2021-8	Opravit pole mena v My Account	Task		IN PROGRESS		1
ASICDE2021-37	Vytvorit "forgot password" funkciu	Task		DONE		1
ASICDE2021-50	Praca na dokumentacii	Task		IN PROGRESS		1
ASICDE2021-56	Mergovanie branch-i	Task		DONE		1

Completed issues

[View in issue navigator](#)

Key :	Summary :	Issue type :	Epic :	Status :	Assignee :	Issue count
ASICDE2021-6	Pridat moznost sa vratit z vizualizacneho modulu do struktury repozitara	Task		DONE		1
ASICDE2021-12	Spojzdenie backendu na serveri	Task		DONE		1
ASICDE2021-22	Umozenie prihlasenia sa aj pomocou emailu	Task		DONE		1
ASICDE2021-27	Umozenie pouzitia klavesovych skratiek na navigovanie sa vo webovom prostredi	Task		DONE		1
ASICDE2021-35	Invite do organizacie - dat moznost vyberu ano/nie, nie ho tam hned pridať	Task		DONE		1
ASICDE2021-36	Invite do organizacie - pridať status "pending" "accepted" alebo "declined"	Task		DONE		1
ASICDE2021-38	Pri prihlasovaní sa, zobrazit adekvatnu chybovu hlasku ak portal neakceptuje heslo	Task		DONE		1
ASICDE2021-39	Aktualizovat plan timu	Task		DONE		1
ASICDE2021-42	Refaktoring front-end kodu	Task		DONE		1
ASICDE2021-43	Refaktoring back-end kodu	Task		DONE		1
ASICDE2021-44	Umozenie spustenia back-endu bez Dockera	Task		DONE		1
ASICDE2021-45	Spustenie back-endu na serveri	Task		DONE		1
ASICDE2021-46	Aktualizacia web stranky timu	Task		DONE		1
ASICDE2021-47	Implementacia prvej verzie pod-stranky About Us	Task		DONE		1
ASICDE2021-48	Implementacia prvej verzie pod-stranky Pricing	Task		DONE		1
ASICDE2021-49	Implementacia prvej verzie pod-stranky Features	Task		DONE		1
ASICDE2021-51	Exportovanie sprintu #1	Task		DONE		1
ASICDE2021-52	Pokracovanie v samostudiu ohľadom front-end technologií	Task		DONE		1
ASICDE2021-53	Pokracovanie v samostudiu ohľadom back-end technologií	Task		DONE		1
ASICDE2021-54	Spisanie dalsich bugov z front-endu	Task		DONE		1
ASICDE2021-55	Spisanie dalsich bugov z back-endu	Task		DONE		1
ASICDE2021-57	Opravit pretecenie containerov pri login/registracii	Task		DONE		1

Issues completed outside of sprint

[View in issue navigator](#)

Key :	Summary :	Issue type :	Epic :	Status :	Assignee :	Issue count
ASICDE2021-40	Vytvorenie a pridelenie taskov na dalsi sprint	Task		DONE		1
ASICDE2021-41	Ohodnotenie taskov pomocou planning pokeru	Task		DONE		1

Šprint 3

Trvanie šprintu: 8.11. 2021 - 22.11. 2021

V 3. šprinte sme pokračovali v upravovaní nedostatkov po minuloročnom tíme. Jednalo sa o refaktoring backend, ako aj frontend modulov s prípadnou optimalizáciou na miestach, kde bola potrebná. Na zlepšenie kvality kódu boli použité nástroje na meranie metrík ako SourceMonitor, SonarQube či JArchitect. Tak ako v predošlých šprintoch, opäť sme objavili viacero FE aj BE bugov, ktoré priebežne odstraňujeme. V tejto chvíli boli potrebné aj organizačné úpravy pri verziovaní kódu pomocou GitHub-u, kde sme museli vyriešiť konflikty medzi starými branchami a pomergovať do jednej plne funkčnej vetvy. Počas 3. šprintu opäť pribudli nové podstránky s informačným kontextom pre budúcich používateľov. Taktiež sa nám podarilo spojzdiť demo verziu projektu na školskom serveri.

Retrospektíva

Čo sme robili dobre?

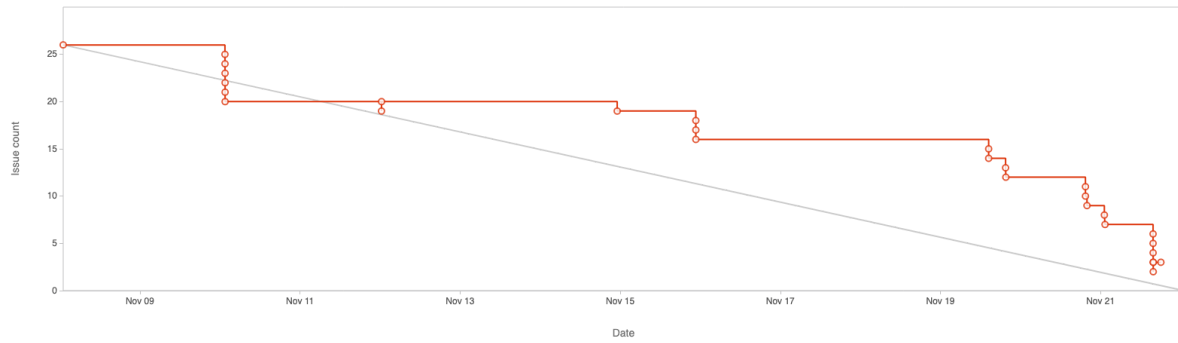
- Správne sme ohodnotili tasky pred spustením šprintu pomocou nového nástroju
- Stanovili sme si dobrý GitHub manažment
- Spustili sme systém code-reviews medzi sebou pre fázu testovania
- Postupne odstraňujeme bugy a zlepšujeme kvalitu kódu
- Pribudli informačné podstránky (FE tasky)
- Máme funkčné demo na školskom serveri
- Spísali sme dokumentáciu, ktorá odzrkadľuje prácu počas prvých troch šprintov

Čo by sme mohli zlepšiť?

- Zlepšiť komunikáciu medzi členmi BE tímu
- Viac sa venovať zdokonaľovaniu existujúcich funkcionalít namiesto pridávania nových
- Dopracovať nedotiahnuté funkcionality po minuloročnom tíme, kedy dáta sa spracúvajú len v typescriptoch ale nie sú ukladané v DB

Zhrnutie

Progres je viditeľný, kvalita kódu sa zlepšuje, bugy postupne odstraňujeme. Jednotlivé úlohy boli pred spustením šprintu relatívne dobre ohodnotené a rozdelené medzi členov tímu. Počas šprintu sa väčšina úloh dokončila, pričom ku každému bol vykonaný aj code-review na zabezpečenie dostatočnej kvality a funkčnosti pridaného kódu.



Incomplete issues

[View in issue navigator](#)

Key :	Summary :	Issue type :	Epic :	Status :	Assignee :	Story points
ASICDE2021-69	Umožnit zobrazenie repozitarov v Organizácii	Task		IN PROGRESS		7
ASICDE2021-86	Prepojenie BE s FE pri pozvanke do organizacie	Task		IN PROGRESS		10
ASICDE2021-89	Vytvorit modálne okno k forgot password funkcii + nastaviť tam routes dopyt	Task		IN PROGRESS		6

Completed issues

[View in issue navigator](#)

Key :	Summary :	Issue type :	Epic :	Status :	Assignee :	Story points
ASICDE2021-37	Vytvorit "forgot password" funkciu	Task		DONE		6
ASICDE2021-50	Praca na dokumentacii	Task		DONE		5
ASICDE2021-8	Opravit pole mena v My Account	Task		DONE		5
ASICDE2021-56	Mergovanie branch-i	Task		DONE		9
ASICDE2021-59	Sortovat repozitare podľa 'favorite' najprv a potom ostatne	Task		DONE		4
ASICDE2021-60	Opravit pridanie repozitarov do organizacie	Task		DONE		6
ASICDE2021-61	Opravit vymazanie chat roomky	Task		DONE		6
ASICDE2021-62	Opravit editaciu chat roomky	Task		DONE		7
ASICDE2021-64	Vylepsit UI v Organizacii v casti Teams	Task		DONE		6
ASICDE2021-67	Pridat v 'Repositories' sekcie 'Favorite' a 'Archived'	Task		DONE		7
ASICDE2021-70	Pridat v 'Pricing' navigaciu pre zoznam cien pre Individual a Organization members	Task		DONE		5
ASICDE2021-74	Aktualizacia web stranky timu	Task		DONE		2
ASICDE2021-79	Refaktoring back-end kodu	Task		DONE		8
ASICDE2021-76	Mergovanie front-end branch-i	Task		DONE		9
ASICDE2021-77	Pokracovanie v samostudiu ohladom back-end technologii	Task		DONE		1
ASICDE2021-78	Pokracovanie v samostudiu ohladom front-end technologii	Task		DONE		1
ASICDE2021-80	Refaktoring front-end kodu	Task		DONE		8
ASICDE2021-81	Spisanie dalsich bugov z front-endu	Task		DONE		3
ASICDE2021-82	Optimalizacia back-endu	Task		DONE		5
ASICDE2021-83	Optimalizacia front-endu	Task		DONE		5
ASICDE2021-84	Deploy dema na server	Task		DONE		5
ASICDE2021-88	Pri pozvani človeka do organizácie sa vypíše "pozvanie úspešné" aj keď je neúspešne...	Task		DONE		5
ASICDE2021-87	Vytvorenie contact us formulara v About Us	Task		DONE		5
ASICDE2021-93	Aktualizovat plan timu	Task		DONE		2

Issues completed outside of sprint

[View in issue navigator](#)

Key :	Summary :	Issue type :	Epic :	Status :	Assignee :	Story points
ASICDE2021-58	Umožnit ulozenie prihlasovacich udajov do webového prehliadača	Task		DONE		2
ASICDE2021-73	Ohodnotenie taskov pomocou planning pokeru	Task		DONE		3

Globálna retrospektíva

Počas zimného semestra sme sa venovali hlavne refaktoringu kódu, zlepšovaniu a opravovaniu chýb, keďže kód sme prevzali po minuloročných študentoch. Prvé šprinty boli preto zamerané primárne na oboznámenie sa s projektom - skúmanie vnútorného fungovania všetkých častí projektu, jeho architektúry, dátovej štruktúry a spôsobu, ako má byť nastavená kompilácia a spustenie kódu lokálne, ale aj na serveri. Okrem toho, nie každý člen nášho tímu mal skúsenosti s potrebnými technológiami. Aj z tohto dôvodu sme v rámci prvých šprintov kládli dôraz na to, aby každý nabral potrebné základné schopnosti v oblasti, ktorej sa chce venovať, aby mohol efektívne pracovať na projekte a jednotlivých úlohách. Postupne sme začali opravovať a vylepšovať existujúce časti projektu a dokončovať funkcionality, ktoré boli nedokončené alebo úplne nefunkčné.

Na začiatku semestra sme si vytvorili metodiky, ktoré sú bližšie opísané v časti - Metodiky. V rámci nich, sme mali definované určité pravidlá a formy, ktorými sme sa riadili po zbytok semestra, aby práca na projekte bola jednotná, prehľadná, konzistentná a efektívna. Počas práce na projekte sme narazili na viacero problémov, ktoré bolo potrebné vyriešiť. Preto jednou z najdôležitejších častí bola komunikácia, ktorá sa každým šprintom zdokonalovala. Keďže častokrát, hlavne v neskorších šprintoch, jednotlivé tasky boli medzi sebou poprepájané a mali vysokú mieru návaznosti, efektívna komunikácia bola kľúčom k ich úspešnému riešeniu.

Všeobecne môžeme zhrnúť, že počas doterajších troch šprintov sme na projekte pracovali úspešne. V každom šprinte sme dokázali úspešne splniť drvivú väčšinu zadefinovaných úloh, no taktiež v každom šprinte ostalo zopár úloh, ktoré sa presunuli do nasledujúceho šprintu. V takýchto prípadoch sa jednalo väčšinou o komplexné úlohy, ktoré si vyžadovali viac času. Toto bolo často spôsobené tým, že počas riešenia niektorej z úloh sa vyskytli neočakávané problémy, ktoré bolo potrebné riešiť prioritne. Celkovo, počas všetkých troch šprintov sa nám úspešne darilo napredovať a vyriešiť veľkú väčšinu existujúcich problémov, ako aj úspešne splniť úlohy definované zadávateľom projektu.

Motivačný dokument

Tím

Náš tím pozostáva z absolventov FIIT, ktorí sa stretli na viacerých predmetoch počas bakalárskeho štúdia. Ako skupina sme spolu v takomto zložení nikdy nepracovali, ale spolupráca nám zatiaľ ide bez problémov. Vďaka tomu sa naučíme spolupracovať s neznámymi ľuďmi z nášho odvetvia.

Viacero z nás absolvovalo nasledujúce voliteľné predmety počas bakalárskeho štúdia, ktoré by mohli nadmerne prispieť k nižšie zvoleným témam: UI - Umelá inteligencia, IAU - Inteligentná analýza údajov, PARALPR - Paralelné programovanie, SPAASM - Systémové programovanie a assembly, PIB - Princípy informačnej bezpečnosti, MSOFT - Modelovanie softvéru, TEAP - Tvorba efektívnych algoritmov a programov, FAPS - Forenzná analýza počítačových systémov, WTECH - Webové technológie, WPUB - Webové publikovanie, PPGSO - Princípy počítačovej grafiky a spracovania obrazu, SSIIT - Spoločenské súvislosti informatiky a informačných a komunikačných technológií, PAP - Projektovanie aplikačných počítačov, DM - Digitálny marketing.

Počas inžinierskeho štúdia sa nám podarilo zapísať si nasledujúce predmety: SMVE - Štatistické metódy vyhodnocovania experimentov, VD - Vizualizácia dát, NMVS - Nové médiá v spoločnosti, KVANTP - Kvantové počítanie, ZKGRA - Základy kryptografie, KPAIS - Kvalita programových a informačných systémov, VINF - Vyhľadávanie informácií, AASS - Aplikačné architektúry softvérových systémov, PRIPOC - Prírodou inšpirované počítanie, TSOFT - Testovanie softvéru, PDT - Pokročilé databázové štruktúry.

Tím má znalosti z programovania webových aplikácií, využívania webových technológií, analýzy dát a iných sfér informatiky. Medzi naše znalosti patria:

Java (SpringBoot), Python (Django, Flask), Databázové systémy (Oracle, Postgres, MSSQL, MySql, Sybase), Data Streaming (Kafka), Analytické zručnosti (UML, BPMN -> EA, Camunda), Git, Docker, Vagrant, PHP (Laravel, Nette, CodeIgniter, Wordpress, Symfony), Android, JavaScript (jQuery, GSAP, ScrollMagic, Vue.js, React+Redux), HTML, CSS (Bootstrap, Tailwind), C, C#.

Aj vo voľnom čase sa podielame na vývoji rôznych osobných projektoch, ako sú napríklad webové aplikácie za pomoci *Node JS, Express JS, EJS, Electron, Bootstrap, jQuery, CSS, HTML, SQL a NoSQL (MongoDB)*. Taktiež sme vytvárali aplikácie za použitia *C, Python, Java, C#*. Niektorí z nás už aj aktívne pracujú v rôznych firmách kde počas pracovného úväzku nadobudli rôzne zručnosti s *Java (SpringBoot), Data Streaming (Kafka), SQL, analýza softvéru (UML, BPMN), PHP a JavaScript*.

Motivácia

Educational Content Engineering Hub - Databáza otázok, odpovedí, úloh a riešení [ECEH-DU]

Niektoré predmety na našej fakulte sú vedené tak, že na prednáške dostaneme pekný a jednoducho urobený výklad, a na skúške alebo zápočte dostaneme jednoduchú otázku položenú zbytočne komplikovaným spôsobom, poprípade sme v prednáškach na danú otázku ani nedostali odpoveď (resp. sme sa ňou priamo nezaoberali). Takáto databáza by pomohla nielen vyplniť medzery v látke, ktorú je treba sa na skúšky učiť ale možno aj formou Q&A lepšie dané učivo vysvetliť. Takisto by sa študenti mohli pripraviť na spôsob, akým učiteľ otázku pokladá. Viacero z nás malo predmety, na ktorých sme už podobný koncept zachytili, aj keď slúžil iba na utvrdenie učiva a otázky na skúške boli iné. Preto, ak by takýto systém existoval, a hlavne by umožnil študentom sa aktívne zapájať, tak by to bolo z pohľadu študentov vítané. Aj z tohto dôvodu by sme sa chceli podieľať na zlepšení kvality vzdelávania na fakulte týmto spôsobom.

FIIT WIX

Nad vytvorením podobného nástroja už väčšina z nás premýšľala. Mohlo by to byť zaujímavé pre ľudí bez zručností z oblasti vytvárania webstránok, ktorí sa chcú prezentovať na internete. Takáto aplikácia by umožnila dizajnovanie frontendu bez zaberania väčšieho množstva času. Radi by sme sa naučili lepšie pracovať s Angularom a Django. Podobné aplikácie už síce existujú, ale niektorí ľudia majú stále problém s nimi pracovať. Bolo by skvelou príležitosťou dostať šancu vytvoriť podobný nástroj, ktorý by bol viac user-friendly. Na dosiahnutie jednoduchej a používateľsky prívetivej aplikácie by sme využili znalosti nadobudnuté počas štúdia na FIIT ako i skúsenosti z iných projektov.

Ion Mobility Spectrometry for Rapid HEMP Potency Testing

Téma nás zaujala vzhľadom na to, že je podobná zadaniam našich bakalárskych prác. Od témy očakávame prehĺbenie znalostí v oblasti machine learningu a zároveň využitie doteraz nadobudnutých dovedností v tejto oblasti (bakalárske práce, IAU...). Počas našich bakalárskych prác sme sa už oboznámili so spracovaním a analýzou rôznych datasetov, na ktoré sme využili viacero vyhodnocovacích metód. Neraz sme využili štatistické testovanie, ktorých výsledky sme následne interpretovali strojovým učením. Všetky tieto nadobudnuté zručnosti by sme vedeli plne využiť v danej problematike.

Príloha A - Zoradenie všetkých tém podľa priority

1. Educational Content Engineering Hub - Databáza otázok, odpovedí, úloh a riešení [ECEH-DU] *(téma č. 8)*
2. FIIT WIX *(téma č. 16)*
3. Ion Mobility Spectrometry for Rapid HEMP Potency Testing *(téma č. 15)*
4. Spektrometrické rozpoznávanie túh do pera *(téma č. 12)*
5. Adverse Media Screening *(téma č. 4)*
6. (Q)SAR analýza fototoxických látok *(téma č. 11)*
7. Document Wizard *(téma č. 17)*
8. Webové IDE pre ASIC [ASICDE] *(téma č. 18)*
9. Aplikačné riešenie pre elektronický volebný systém *(téma č. 20)*
10. IoT platforma na priemyselnú automatizáciu - malý pivovar *(téma č. 14)*
11. Monitorovanie a správa systému pre výrobný areál [LOMON] *(téma č. 9)*
12. Transformácia priestorov na bezpečné a inteligentné miesta na prácu [SmartSpace2] *(téma č. 2)*

Príloha B - Rozvrh tímu

	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	20:00 - 20:50
Pondelok		Prednáška Kvant.	Cvičenie VINF. (Benovič)			MÓŽEME ROBIŤ TP			Prednáška Riad.		Cvičenie AS (Plovárči) Cvičenie Riad. (Benovič)		
Utorok		Prednáška Sprac. Inf.	Cvičenie VINF. (Leikes, Kollárová, Janeček, Badová, Krajčír)			MÓŽEME ROBIŤ TP			Prednáška VISS		Prednáška TP1		Cvičenie Sprac. Inf. (Plovárči)
Streda		Prednáška Kvalita.	Prednáška VINF	Cvičenie Kvalita. (Plovárči, Kollárová)					Prednáška MTS	Cvičenie MTS (DP, LL, PK, MJ, EK, MB)			Cvičenie MTS (Badová)
Štvrtok		Prednáška Krypt.	Cvičenie Kvant. (Benovič) Cvičenie Krypt. (Kollárová)	Prednáška NIMYS	Cvičenie NIMYS (Leikes, Janeček, Badová)				Cvičenie AS (Benovič, Leikes, Kollárová, Janeček, Krajčír, Badová)				
Piatok		Práca vo firme		Cvičenie PDT (Krajčír)	Cestovanie domov			Cestovanie domov					

Prednáška

Práca vo firme

Cestovanie domov

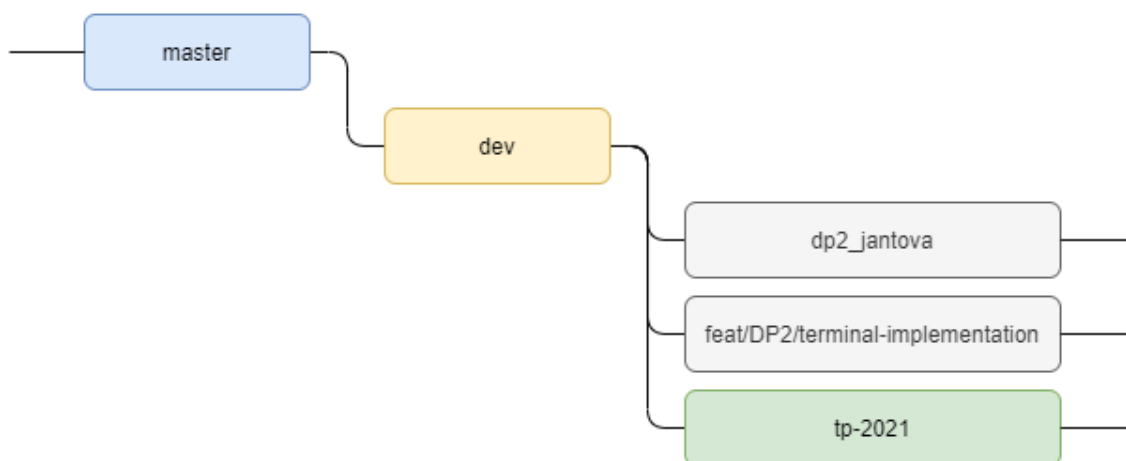
Cvičenie

Možeme robiť TP

Metodiky

GitHub

Vytvorili sme vlastnú vetvu z vetvy dev s názvom tp-2021.



Každá úloha z Jiry bola riešená v zvlášť vetve z vetvy tp-2021. Po dokončení úlohy následne vytvoril riešiteľ pull request a priradil do reviews ostatných členov tímu.

Opravit pridanie repozitarov do organizacie #79

matej-janecek wants to merge 1 commit into `tp-2021` from `feature/pridanie-repozarov-do-organizacie/tp-2021`

Conversation 0 | Commits 1 | Checks 0 | Files changed 5 | +90 -77

matej-janecek commented 7 hours ago

organization: true/false podla toho ci sa vytvara repozitar v organizacii alebo u seba

upraveny dev env 4aac713

matej-janecek requested a review from Laco1997 7 hours ago

matej-janecek assigned Laco1997 7 hours ago

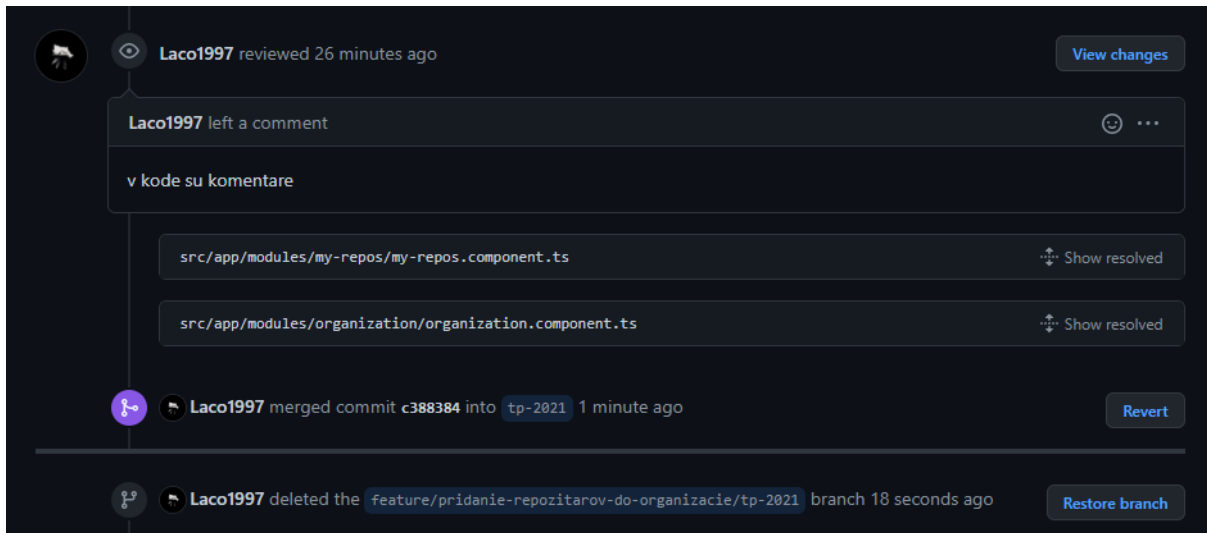
Reviewers: Laco1997

Assignees: Laco1997

Labels: None yet

Projects: None yet

Tí vykonali kontrolu úpravy zdrojového kódu a otestovali funkčnosť riešenej úlohy. V prípade že bolo všetko funkčné bol pull request mergnutý do vetvy tp-2021.



Týmto spôsobom sme dokázali efektívne sledovať stav dokončenia úloh.

Naming vetiev

<typ>/<nazov-tasku>/tp-2021

Možné typy sú:

- feature
- refactor
- bugfix

Príkladom je:

feature/pridanie-repozitarov-do-organizacie/tp-2021

Jira

Pre úspešné manažovanie úloh sme zvolili portál Jira od Atlassianu.

Vytvorenie úloh

Každá úloha je vytvorená na základe nejakého problému. Prvotne identifikujeme samotný bug (napr. nefunkčné prihlasovacie tlačidlo) alebo požiadavku (napr. potreba aktualizácie web stránky) a na základe týchto problémov a požiadaviek vytvoríme samostatnú úlohu, ktorú zatiaľ nezaevidujeme do Jiry.

Ohodnotenie a pridelenie úloh

Po úspešnom vytvorení danej úlohy, musíme každý task ohodnotiť (očakávaná náročnosť alebo predpokladaný čas na úspešné vykonanie úlohy) a niekomu priradiť. Pre ohodnotenie sme použili metódu *Planning poker*, na portáli <https://agilemana.com/>.

The screenshot shows the agilemana interface for a 'Sprint #3' planning poker session. At the top, there is a search bar for tickets/rooms, a 'Create room' button, and the user profile 'Laco Lelkes'. The main area is divided into three sections:

- Click to vote:** A grid of 10 buttons numbered 1 to 10 for voting.
- Stories:** A list of tasks with numbers indicating their complexity:
 - Umožnit uloženie prihlasovacích údajov do webového prehliadača (2)
 - Sortovať repozitáre podľa 'favorite' najprv a potom ostatné (4)
 - Opraviť pridanie repozitárov do organizácie (6)
 - Opraviť vymazanie chat roomky (6)
 - Opraviť editáciu chat roomky (7)
 - Spojzdnit samotný chat (9)
- People:** A list of team members with their avatars and names: Laco Lelkes (Moderator), Denis Plovárči, Marek Benovič, Jasmina Badová, Matej Janeček, Paula Kollarova, and edokrajcir.

At the bottom right, there is an 'Action' section with a blue 'Show results' button.

Pre každý task sme hodnotili predpokladanú náročnosť na vypracovanie na škále od 1 (nenáročný task) do 10 (náročný task). Po ohodnotení tasku každým členom, sme získali priemer, ktorý sme použili ako finálnu hodnotu. Následne Scrum Master pridelil jednotlivé úlohy jednému alebo viacerým členom na vypracovanie.

Správa úloh

Po vytvorení úloh, ich ohodnotení a pridelení jednotlivým (alebo viacerým) členom tímu, sme mohli začať samotné úlohy spracúvať a riadiť.

Atribúty úloh

Každá úloha obsahuje nasledujúce atribúty:

- *Name* - názov, krátky text ktorý vystihuje podstatu samotného tasku
- *Description* - opis, pre podrobnejšie priblíženie si úlohy
- *State* - stav, v akom stave sa daný task nachádza (viac v časti *Stavy úloh*)
- *Reporter* - tvorca samotného tasku
- *Assigned to* - člen alebo členovia, ktorí sú zodpovední za priebeh samotného tasku
- *Story point estimate* - náročnosť tasku pomocou *Planning poker* vyhodnotenia
- *Sprint* - do ktorého šprintu je daný task priradený

Issue: Vytvorenie web stránky pre tím (ASICDE2021-10)

Description: Web-stranka: ASICDE - TP1

- použitie uz existujúcej nepoužívanej stránky (vytvorena Eduardom Krajcirom) na ktoru boli nahodene informacie o projekte a zapisnice

Activity: Show: All Comments History (Newest first)

Details:

Assigned to	Eduard Krajčír
Original estimate	2d
Labels	None
Sprint	None +1
Story point estimate	2
Development	Branch
Reporter	Ladislav Leikes

Created October 13, 2021, 5:08 PM
Updated October 15, 2021, 5:57 PM
Resolved October 15, 2021, 5:57 PM

Stavy úloh

Stav úloh je riadený členmi tímu ktorí boli pre daný task pridelení. Každý task sa mohol dostať do nasledujúcich stavov. Stavy sú usporiadané v poradí podľa workflowu úloh.

- *Todo* - v prvotnej fáze jednotlivých úloh, sú úlohy zadelené do stavu *Todo*, čo naznačuje že na úlohe sa ešte nezačalo pracovať a je posiaľ len zadaná.
- *In Progress* - každá úloha je pridelená jednému alebo viacerým členom tímu, ktorí majú túto úlohu na zodpovednosti. Akonáhle sa na nejakej úlohe začne pracovať, tak sa premiestni zo stavu *Todo*, do stavu *In Progress*.
- *Test* - potom ako zodpovedná osoba dokončí prácu na úlohe, tak sa stav danej úlohy zmení z *In Progress* na *Test*. V danom stave je priestor na testovanie a preverenie danej úlohy, či sa vykonala správne. Do tohto stavu sa nemusí každá úloha podmienenčne dostať. Napríklad ak sa jedná o úlohu zameranú mimo SW (hodnotenie taskov, štúdium backend technológií, ...), tak je možné danú úlohu rovno premiestniť do stavu *Done*.
- *Done* - po ukončení a úspešnom overení úlohy sa môže daná úloha premiestniť do finálneho stavu s názvom *Done*.

Komunikácia

Cieľom metodiky komunikácie a správy dát je definovať všetky použité komunikačné platformy a nástroje, pomocou ktorých bola uskutočnená komunikácia v rámci tímu, okrem toho medzi všetkými osobami, ktoré sa nejakým spôsobom podieľali na vývoji projektu.

Používané nástroje

V nasledujúcej časti je popísaná funkcionálnosť komunikačných nástrojov, ktoré sme využívali:

- **Discord** – aplikácia Discord slúži na komunikáciu medzi členmi tímu, prostredníctvom ktorej sa členovia informujú o rôznych aspektoch projektu. Pre efektívnejšiu komunikáciu, sme si rozdelili náš server podľa jednotlivých kategórií na nasledujúce kanály:
 - **backend** - kanál pre backendovú časť projektu, pomoc a spolupráca členom tímu pri práci na backendovej časti
 - **consultations**
 - **credentials** – dôležité prihlasovacie údaje pre server
 - **deadlines** – prehľadný rozpis deadlinov
 - **frontend** – kanál pre frontendovú časť projektu, podobne ako pri kanály „backend“
 - **general** – všeobecné informácie, dohadovanie stretnutí, dôležité informácie, neočakávané problémy, informovanie o prograse, novinky
 - **info** – všetky potrebné informácie v rámci tímového projektu. Prihlasovacie údaje spoločného konta, konta v Jira na manažment, web a linky na spoločný disk, dôležité dokumenty, na ktorých spoločne pracujeme.
 - **Jenkins** – inštatné informovanie o buildovaní. Aplikácia Jenkins posiela automatizované správy s číslom buildu a názvom projektu, ktorý bol buildovaný.
 - **offtopic** – nezaradená konverzácia
 - **roles** – zadefinovanie primárnych rolí členov nášho tímu
 - okrem zmienených kanálov, mohli členovia tímu komunikovať aj medzi sebou v **súkromných správach**
- **Microsoft Teams** – aplikácia, cez ktorú prebiehali mítingy s vedúcim projektu a taktiež s ľuďmi, ktorí pracovali na projekte v minulosti. V tíme **ASICDE** sa okrem nášho **kanálu FIIT Timovy Projekt 2020-21** nachádzajú ďalšie 2 kanály:
 - **General** – verejný kanál pre vedúcich projektov, členov nášho tímu a všetkých študentov, ktorí pracujú na záverečných prácach
 - **Bakalárky a Diplomovky 2020-21** - slúži na komunikáciu medzi vedúcimi projektov a riešiteľmi záverečných prác. Do tohto kanála prispievame minimálne a skôr nám umožňuje informovať sa o konkrétnych funkcionalitách a zmenách v softvéri

Export evidencie úloh

Podrobný rozpis úloh nájdete na webovej stránke tímu.

Šprint 1

Issue key	Summary	Reporter	Created	Custom field (Story point estimate)
ASICDE2021-37	Vytvorit "forgot password" funkciu	Paula Kollárová	15/Oct/21 3:46 PM	6.0
ASICDE2021-36	Invite do organizácie - pridať status "pending" "accepted" alebo "declined"	Paula Kollárová	15/Oct/21 2:53 PM	9.0
ASICDE2021-35	Invite do organizácie - dať možnosť výberu áno/nie, nie ho tam hneď pridať	Paula Kollárová	15/Oct/21 2:50 PM	9.0
ASICDE2021-34	Ohodnotenie jednotlivých taskov podľa časovej nárčnosti	Ladislav Leikes	14/Oct/21 10:32 PM	1.0
ASICDE2021-33	Spojzdenie ASICDE aplikácie na OS Windows	Ladislav Leikes	14/Oct/21 8:40 PM	1.0
ASICDE2021-27	Umožnenie použitia klavesových skratiek na navigovanie sa vo webovom prostredí	Ladislav Leikes	14/Oct/21 2:33 PM	2.0
ASICDE2021-26	Umožniť otvorenie responzítara aj pomocou kliknutia na kartu	Ladislav Leikes	14/Oct/21 2:32 PM	2.0
ASICDE2021-25	Navrhnuť Pricing pod-stranku	Ladislav Leikes	14/Oct/21 2:28 PM	5.0
ASICDE2021-24	Navrhnuť Features pod-stranku	Ladislav Leikes	14/Oct/21 2:28 PM	5.0
ASICDE2021-23	Navrhnuť About us pod-stranku	Ladislav Leikes	14/Oct/21 2:28 PM	6.0
ASICDE2021-22	Umožnenie prihlásenia sa aj pomocou emailu	Ladislav Leikes	14/Oct/21 2:27 PM	4.0
ASICDE2021-21	Spraviť homepage stranku na responzivnú	Ladislav Leikes	14/Oct/21 2:27 PM	4.0
ASICDE2021-19	Prestudovanie existujúcich zdrojov a oboznámenie sa s prostredím	Ladislav Leikes	13/Oct/21 5:35 PM	0.0
ASICDE2021-18	Vytvorenie taskov a planu na Jira	Ladislav Leikes	13/Oct/21 5:23 PM	2.0
ASICDE2021-17	Zapisovanie poznámok z konzultácií #2	Ladislav Leikes	13/Oct/21 5:23 PM	1.0
ASICDE2021-16	Zapisovanie poznámok z konzultácií	Ladislav Leikes	13/Oct/21 5:21 PM	1.0
ASICDE2021-15	Spojzdenie timovej web stranky na VM	Ladislav Leikes	13/Oct/21 5:17 PM	2.0
ASICDE2021-14	Vytvorenie contentu na timovu web stranku	Ladislav Leikes	13/Oct/21 5:17 PM	1.0
ASICDE2021-13	Aktualizácia timovej web stranky	Ladislav Leikes	13/Oct/21 5:13 PM	1.0
ASICDE2021-12	Spojzdenie backendu na serveri	Ladislav Leikes	13/Oct/21 5:13 PM	4.0

ASICDE2021-11	Lokálne spojzdenie aplikacie	Ladislav Lelkes	13/Oct/21 5:11 PM	1.0
ASICDE2021-10	Vytvorenie web stránky pre tím	Ladislav Lelkes	13/Oct/21 5:08 PM	2.0
ASICDE2021-9	Spisanie bugov	Ladislav Lelkes	13/Oct/21 5:08 PM	2.0
ASICDE2021-8	Opraviť pole mena v My Account	Matej Janeček	12/Oct/21 2:20 PM	5.0
ASICDE2021-6	Pridať možnosť sa vrátiť z vizualizačného modulu do štruktúry repozitára	Ladislav Lelkes	07/Oct/21 9:35 PM	5.0

Šprint 2

Issue key	Summary	Reporter	Created	Custom field (Story point estimate)
ASICDE2021-57	Opraviť pretečenie containerov pri login/registrácii	Ladislav Lelkes	22/Oct/21 11:47 PM	7.0
ASICDE2021-56	Mergovanie branch-i	Ladislav Lelkes	22/Oct/21 5:30 PM	9.0
ASICDE2021-55	Spisanie ďalších bugov z back-endu	Ladislav Lelkes	21/Oct/21 2:31 PM	2.0
ASICDE2021-54	Spisanie ďalších bugov z front-endu	Ladislav Lelkes	21/Oct/21 2:25 PM	2.0
ASICDE2021-53	Pokračovanie v samostudiu ohľadom back-end technológii	Ladislav Lelkes	21/Oct/21 2:24 PM	4.0
ASICDE2021-52	Pokračovanie v samostudiu ohľadom front-end technológii	Ladislav Lelkes	21/Oct/21 2:24 PM	5.0
ASICDE2021-51	Exportovanie sprintu #1	Ladislav Lelkes	21/Oct/21 2:24 PM	1.0
ASICDE2021-50	Praca na dokumentácii	Ladislav Lelkes	21/Oct/21 2:23 PM	5.0
ASICDE2021-49	Implementácia prvej verzie pod-stranky Features	Ladislav Lelkes	21/Oct/21 2:23 PM	5.0
ASICDE2021-48	Implementácia prvej verzie pod-stranky Pricing	Ladislav Lelkes	21/Oct/21 2:23 PM	5.0
ASICDE2021-47	Implementácia prvej verzie pod-stranky About Us	Ladislav Lelkes	21/Oct/21 2:23 PM	6.0
ASICDE2021-46	Aktualizácia web stránky tímu	Ladislav Lelkes	21/Oct/21 2:22 PM	3.0
ASICDE2021-45	Spustenie back-endu na serveri	Ladislav Lelkes	21/Oct/21 2:22 PM	9.0
ASICDE2021-44	Umožnenie spustenia back-endu bez Dockera	Ladislav Lelkes	21/Oct/21 2:21 PM	4.0
ASICDE2021-43	Refaktoring back-end kódu	Ladislav Lelkes	21/Oct/21 2:20 PM	8.0
ASICDE2021-42	Refaktoring front-end kódu	Ladislav Lelkes	21/Oct/21 2:20 PM	8.0
ASICDE2021-41	Ohodnotenie úloh pomocou planning pokeru	Ladislav Lelkes	21/Oct/21 2:19 PM	2.0

ASICDE2021-40	Vytvorenie a pridelenie taskov na dalsi sprint	Ladislav Lelkes	21/Oct/21 2:19 PM	3.0
ASICDE2021-39	Aktualizovat plan timu	Ladislav Lelkes	21/Oct/21 2:18 PM	3.0
ASICDE2021-38	Pri prihlasovani sa, zobrazit adekvatnu chybovu hlasku ak portal neakceptuje heslo	Marek Benovič	21/Oct/21 12:11 AM	4.0

Šprint 3

Issue key	Summary	Reporter	Created	Custom field (Story point estimate)
ASICDE2021-86	Prepojenie BE s FE pri pozvanke do organizacie	Ladislav Lelkes	06/Nov/21 6:14 PM	10.0
ASICDE2021-84	Deploy dema na server	Ladislav Lelkes	06/Nov/21 2:54 PM	5.0
ASICDE2021-83	Optimalizacia front-endu	Ladislav Lelkes	03/Nov/21 2:46 PM	5.0
ASICDE2021-82	Optimalizacia back-endu	Ladislav Lelkes	03/Nov/21 2:46 PM	5.0
ASICDE2021-81	Spisanie dalsich bugov z front-endu	Ladislav Lelkes	03/Nov/21 2:46 PM	3.0
ASICDE2021-80	Refaktoring front-end kodu	Ladislav Lelkes	03/Nov/21 2:46 PM	8.0
ASICDE2021-79	Refaktoring back-end kodu	Ladislav Lelkes	03/Nov/21 2:46 PM	8.0
ASICDE2021-78	Pokracovanie v samostudiu ohladom front-end technologii	Ladislav Lelkes	03/Nov/21 2:45 PM	1.0
ASICDE2021-77	Pokracovanie v samostudiu ohladom back-end technologii	Ladislav Lelkes	03/Nov/21 2:45 PM	1.0
ASICDE2021-76	Mergovanie front-end branch-i	Ladislav Lelkes	03/Nov/21 2:44 PM	9.0
ASICDE2021-74	Aktualizacia web stranky timu	Ladislav Lelkes	03/Nov/21 2:44 PM	2.0
ASICDE2021-73	Ohodnotenie taskov pomocou planning pokeru	Ladislav Lelkes	03/Nov/21 2:43 PM	3.0
ASICDE2021-72	Prepojit BE s Pricing	Ladislav Lelkes	03/Nov/21 2:38 PM	10.0
ASICDE2021-71	Vyvorit members (Individual, Teacher, Admin, atd)	Ladislav Lelkes	03/Nov/21 2:38 PM	10.0
ASICDE2021-70	Pridat v 'Pricing' navigaciu pre zoznam cien pre Individual a Organization members	Ladislav Lelkes	03/Nov/21 2:33 PM	5.0
ASICDE2021-69	Umoznit zobrazenie repozitarov v Organizacii	Ladislav Lelkes	03/Nov/21 2:31 PM	7.0
ASICDE2021-68	Vytvorit forum pre organizaciu	Ladislav Lelkes	03/Nov/21 2:30 PM	10.0
ASICDE2021-67	Pridat v 'Repositories' sekcie 'Favorite' a 'Archived'	Ladislav Lelkes	03/Nov/21 2:29 PM	7.0
ASICDE2021-66	Umoznit funkcionalitu 'Transfer ownership'	Ladislav Lelkes	03/Nov/21 2:29 PM	7.0

ASICDE2021-65	Umožnit funkcionálnosť 'Share repository'	Ladislav Lelkes	03/Nov/21 2:27 PM	9.0
ASICDE2021-64	Vylepsit UI v Organizácii v časti Teams	Ladislav Lelkes	03/Nov/21 1:59 PM	6.0
ASICDE2021-63	Vylepsit UI pre chat room	Ladislav Lelkes	03/Nov/21 1:20 PM	9.0
ASICDE2021-62	Opravit editáciu chat roomky	Ladislav Lelkes	03/Nov/21 1:19 PM	7.0
ASICDE2021-61	Opravit vymazanie chat roomky	Ladislav Lelkes	03/Nov/21 1:19 PM	6.0
ASICDE2021-60	Opravit pridanie repozitárov do organizácie	Ladislav Lelkes	03/Nov/21 1:05 PM	6.0
ASICDE2021-59	Sortovať repozitáre podľa 'favorite' najprv a potom ostatné	Ladislav Lelkes	03/Nov/21 1:02 PM	4.0
ASICDE2021-58	Umožnit uloženie prihlasovacích údajov do webového prehliadača	Ladislav Lelkes	03/Nov/21 1:00 PM	2.0

Webové sídlo

Webové sídlo tímu: <https://team16-21.studenti.fiit.stuba.sk/>

