

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Tím 11

19. Automatizácia procesov KYC a AML

Dokumentácia k tímovému projektu



Vedúci práce: Tibor Vincze, MSc.

Kontakt: fiit_team_11@googlegroups.com

Akademický rok: 2021/2022

Obsah

I	Inžinierske dielo	1
1	Úvod	2
2	Globálne ciele pre ZS	3
3	Celkový pohľad na systém	4
4	Moduly systému	6
4.1	Analýza	6
4.1.1	Úvod do problematiky	6
4.1.1.1	Verifikácia KUV	7
4.1.2	Opis KYC	7
4.1.3	Pomocné technológie	7
4.2	Návrh	9
4.2.1	Návrh overovania osôb	9
4.2.2	Spracovanie dát	9
4.2.3	Popis logiky aplikácie	10
4.3	Implementácia	12
4.3.1	Funkcionality aplikácie	13
4.3.1.1	Fotenie	13
4.3.1.2	Integrácia OCR	14
4.3.1.3	Komunikácia s databázami	14
4.3.1.4	Ukladanie informácií	15
4.4	Testovanie	16
4.4.1	Jednotkové testy	16
4.4.2	Akceptačné testy	16

II Riadenie projektu	31
5 Úvod	32
6 Roly členov tímu a podiel práce	33
6.1 Roly členov tímu	33
6.2 Podiel práce na dokumentácii	33
7 Aplikácie manažmentov	34
7.1 Manažment splnenia úloh	34
7.2 Manažment komunikácie	34
7.3 Manažment práce s úlohami	34
7.4 Manažment testovania	35
7.5 Manažment verziovania	35
7.6 Manažment vývoja	35
8 Sumarizácie šprintov	36
8.1 Šprint č.1	36
8.2 Šprint č.2	39
8.3 Šprint č.3	43
9 Globálna retrospektíva	47
9.1 Retrospektíva šprintu č.1	48
9.2 Retrospektíva šprintu č.2	50
9.3 Retrospektíva šprintu č.3	52
Príloha A: Motivačný dokument	A-1
Tím	A-1
Motivácia	A-3
3. DataHub pre rôzne typy zariadení a ich spracovanie/ analýzu/ vizualizáciu (1)	A-3
19. Automatizácia procesov KYC a AML (2)	A-5
16. FIIT WIX (3)	A-6
Príloha A - Zoradenie všetkých tém podľa priority	A-7
Príloha B - Rozvrh tímu	A-8

Príloha B: Metodika Definition of Done	B-1
Príloha C: Metodika komunikácie	C-1
Príloha D: Metodika práce s úlohami	D-1
Príloha E: Metodika testovania	E-1
Príloha F: Metodika verziovania	F-1
Príloha G: Metodika vývoja aplikácie	G-1
Príloha H: Webové sídlo projektu	H-1

Časť I

Inžinierske dielo

1 Úvod

V tímovom projekte sa venujeme automatizácii procesov KYC (Know Your Client) a AML(Anti-money laundering), ktoré predstavujú v spoločnosti BDO veľkú administratívnu záťaž.

BDO predstavuje medzinárodnú sieť účtovných, daňových a poradenských spoločností, ktoré poskytujú vysoko kvalitné odborné služby pod spoločným menom BDO.

AML predstavuje pojem, ktorý zastrešuje regulačné procesy, ktoré musia mať firmy podľa zákona zavedené na boj proti praniu špinavých peňazí. KYC je súčasťou AML a predstavuje proces identifikácie a overenia totožnosti klienta za účelom zhodnotenia nebezpečia zneužitia obchodného vzťahu k legalizácii výnosu z trestnej činnosti, financovania terorizmu, korupcie a ďalšieho nezákonného jednanja.

BDO poskytuje služby, teda klient osloví BDO, že má záujem o ich služby. Predtým, ako sa ale podpíše zmluva, je potrebné prejsť procesom overenia, aby sa nešlo do vzťahu so spoločnosťou, o ktorej nič nevedia. Toto rozhodnutie je vždy manuálne a rozhoduje o tom manažment. Celý proces môže trvať aj niekoľko dní, preto je žiadúce tieto procesy zautomatizovať.

V tomto dokumente popisujeme vytvárané inžinierske dielo z rôznych pohľadov. V kapitole 2 máme zadané globálne ciele pre zimný semester. Celkový pohľad na systém nám popisuje kapitola 3. Moduly systému, vrátane analýzy, návrhu, implementácie a testovania, nájdeme v kapitole 4.

2 Globálne ciele pre ZS

Hlavným cieľom projektu je automatizácia procesu identifikácie a verifikácie klientov, ktoré zastrešuje pojem KYC(Know Your Client).

Začínali sme si niekoľko cieľov pre zimný semester:

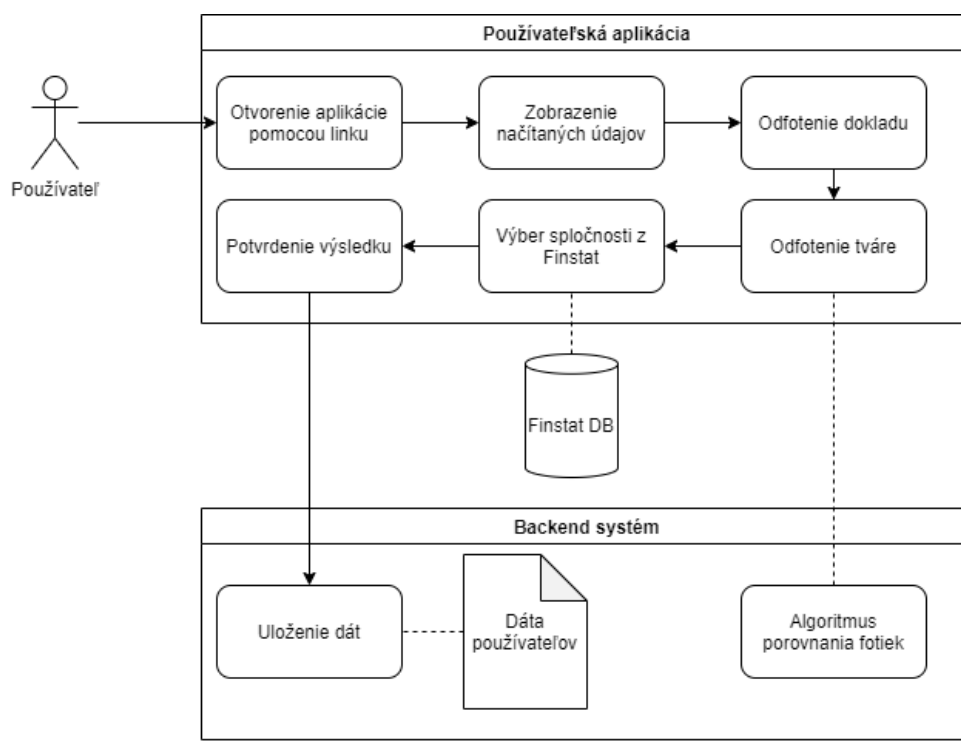
- Urobiť si prehľad dostupných technológií a existujúcich riešení
- Pripojenie sa na FinStat API
- Vytvorenie databázy, kde budú uložené získané údaje o klientovi
- Vytvorenie prototypu mobilnej aplikácie, ktorá zobrazí údaje vyčítané z dokladu totožnosti, overí totožnosť klienta vďaka porovnaniu selfie s fotografiou z dokladu totožnosti, získa aspoň základné údaje z FinStatu.

3 Celkový pohľad na systém

Náš systém sa skladá z 2 komponentov. Multiplatformovej aplikácie, ktorá je určená na rôzne typy zariadení ako je mobilný telefón ale aj počítač a druhá časť je náš backend. Aplikácia komunikuje s naším serverom pri overení tváre a potom pri uložení dát. V kroku výberu spoločnosti komunikuje naša aplikácia taktiež s databázou Finstat, ktorá je dostupná cez Finstat API.

Ako možno vidieť z obr.3.1 naša aplikácia má 6 základných funkcionalít. Tými sú:

1. Otvorenie aplikácie pomocou linku
2. Zobrazenie načítaných údajov
3. Odfotenie dokladu
4. Odfotenie tváre
5. Výber spoločnosti z Finstat
6. Potvrdenie výsledku

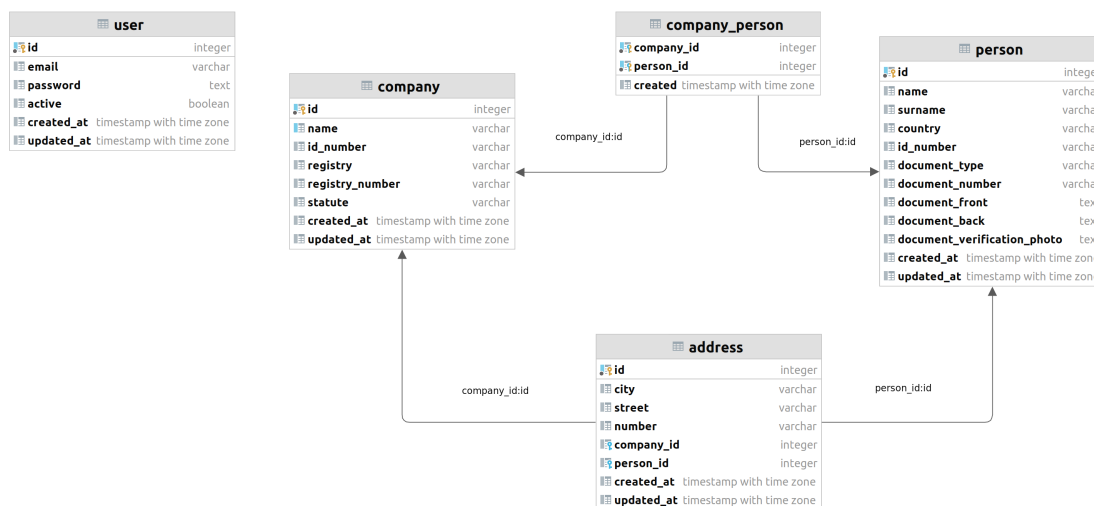


Obr. 3.1: Zjednodušený pohľad na systém

Funkcionality týchto častí sú popísané v kapitole Moduly systému. Náš backend poskytuje

zatiaľ iba 2 funkcionality a tými sú porovnanie fotiek z dokladu oproti fotke odfotenie používateľom. Služba vráti percento podobnosti ako veľmi sú si ľudia na fotkách podobní. Druhou funkcionalitou je uloženie dát na backende. Každý používateľ, ktorý prejde našou aplikáciou bude zaznamenaný na našom serveri, kvôli archivácii.

Ako možno vidieť z obr.3.2 backend má definovanú entitu user, ktorá slúži na uloženie všetkých používateľov, ktorí našu aplikáciu použijú. Potom v chode aplikácie sa získavajú všetky ostatné údaje. V entite person sa vyplnia údaje pomocou odfotenia dokladu, detto platí aj pre adresu používateľa ak použil doklad s týmito informáciami. Entita company sa dopĺňa na základe IČO z linku, cez ktorý sa používateľ do aplikácie dostal a zároveň sa overuje s portálom Finstat.



Obr. 3.2: Dátový model backendu

4 Moduly systému

4.1 Analýza

V tejto sekcii by sme sa chceli venovať analýze problému KYC a AML. Keďže sme sa nikto z tímu výrazným spôsobom nestretli s problematikou týchto procesov, bolo potrebné si prejsť, čo daná problematika zahŕňa. Už počas prvého stretnutia nám bolo product ownerom vysvetlené zhruba čo firma robí a ako aktuálne prebieha proces KYC, ktorému sa budeme v tejto časti najviac venovať.

4.1.1 Úvod do problematiky

Ešte predtým, ako ale prejdeme na vysvetlenie a analýzu, čo vlastne KYC je, je potrebné uviesť prečo tieto procesy potrebujeme. Aktuálny stav procesu KYC vo firme prebieha asi nasledovne. Ako prvé máme klienta, ktorý chce od našej spoločnosti nejakú službu, ktorú ponúkame. Keďže naša firma je finančná inštitúcia, je zo zákona potrebné vykonať niektoré úkony a tiež vyhodnotiť riziko spojené s uzatvorením takejto zmluvy. Ešte pred podpisom zmluvy musíme mať zoznam ľudí, ktorí budú z tejto služby profitovať, teda takýto ľudia sa nazývajú koneční užívatelia výhod(KUV). Zmluvy sa dajú podpisovať ako s fyzickými, tak aj s právnickými osobami a teda KUV môže byť aj viac. Všetkých KUV musíme zo zákona identifikovať a verifikovať pred uzatvorením zmluvy.

Aby bol tento proces prehľadnejší, dá sa zoradiť do nasledujúcich bodov:

1. Klient chce službu od našej spoločnosti
2. Identifikovanie KUV
3. Verifikácia KUV
4. Uloženie údajov
5. Vyhodnotenie rizika
6. Uzatvorenie zmluvy

4.1.1.1 Verifikácia KUV

Cieľom tohto projektu je uľahčiť proces identifikácie a verifikácie klientov a všetkých KUV. Aktuálne je tento krok manuálny a je potrebné s klientom prejsť tento krok fyzicky. V tomto kroku vyzveme klienta, aby nám ukázal OP alebo iný doklad totožnosti, ako napríklad pas. Z tohto dokladu je potrebné pre archivačné účely spraviť kópiu a uložiť do spisov.

Klient je tiež povinný nahlásiť zo zákona všetkých KUV a títo taktiež musia prejsť procesom identifikácie a verifikácie, ako bolo opísané vyššie.

4.1.2 Opis KYC

Z anglického Know Your Client (KYC) je proces, v ktorom je potrebné overiť identitu klienta a zároveň ho verifikovať. Je to vlastne štandard, ktorý platí v finančnom sektore pre verifikáciu klientov, ich rizika a finančného profilu. Pre identifikáciu sa používajú údaje ako Meno, Priezvisko, Rodné číslo, Dátum narodenia, Adresa trvalého bydliska, Číslo dokladu totožnosti atď.

Pri verifikácii potrebujeme zistiť, či klient je naozaj tou osobou, za ktorú sa pri identifikácii vydáva. Pri manuálnej verifikácii sa porovnávajú údaje ako fotka z občianskeho preukazu s tvárou klienta. Cieľom je tento proces plne automatizovať pomocou využitia pokročilých technológií. Ďalej potrebujeme zistiť, či klient alebo iná osoba z KUV nie je sankciovaná, teda hľadaná, alebo politicky exponovaná. Takéto údaje sa overujú v registri trestov alebo ak ide o zahraničnú osobu, tak pomocou celosvetovej databázy LexisNexis.

4.1.3 Pomocné technológie

V analýze sme si s product ownerom zadefinovali požiadavky, v ktorých sa hovorilo, kde všade by aplikácia mala fungovať. Požiadavka bola, aby aplikáciu bolo možné spustiť po rozkliknutí linku z email-u, teda vo webovom prehliadači alebo na mobilnom zariadení.

Po zvážení možností, ako napríklad Android aplikácia v Jave, Webová stránka v PHP, sa tým s product ownerom zhodol, že aplikácia bude vytvorená v open-source vývojárskom balíku Flutter, ktorý je implementovaný v programovacom jazyku Dart. Tento balík má

výhodu, že je možné jednu aplikáciu distribuovať na všetky požadované zariadenia. Teda jedna aplikácia beží ako na mobilnom zariadení, tak aj vo webovom prehliadači.

Ďalšou dôležitou technológiou je optical character recognition(OCR), teda technológia, ktorá z obrázka dokáže vyčítať text. Nad týmto textom vieme neskôr robiť potrebné operácie. Mali sme na výber viacero možností, ako napríklad vývoj vlastného riešenia, použitie open-source riešenia alebo použitie plateného riešenia. Po dôkladnej analýze všetkých možností sa tím s product ownerom rozhodol použiť platené riešenie OCR, pretože vývoj vlastného riešenia by trval prídlho a použité open-source riešenie nedosahovalo kvalitu, s ktorou by bolo možné ďalej pracovať.

4.2 Návrh

V tejto kapitole opíšeme aké riešenie sme navrhli na splnenie požiadaviek product ownera. Hlavným cieľom je automatizácia procesu identifikácie a verifikácie klienta. Toto plánujeme dosiahnuť vyvinutím aplikácie, ktorá bude fungovať ako vo webovom prehliadači počítača, tak aj na mobilných zariadeniach ako aplikácia. Táto aplikácia bude vyvinutá pomocou open-source vývojového balíka Flutter v jazyku Dart.

Hlavnými časťami logiky aplikácie je otvorenie aplikácie pri kliknutí na špeciálne generovaný link s informáciami o klientovi, ktorý tento link spustil, odfotenie dokladu totožnosti z oboch strán, odfotenie tváre klienta pre porovnanie s fotkou z dokladu totožnosti a následne uloženie dát pre archiváciu.

V ďalších podkapitolách sa pozrieme na to, ako sme navrhli niektoré časti uvedené vyššie.

4.2.1 Návrh overovania osôb

Overovanie osôb je navrhnuté pomocou linku, ktorý je generovaný špecificky pre klienta, ktorého potrebujeme identifikovať a verifikovať. Po kliknutí sa klientovi zobrazí aplikácia, ktorá od neho vyžiada odfotenie dokladu totožnosti, taktiež je ale možné vybrať fotografiu z albumu telefónu alebo počítača. Nad touto fotografiou vykonáme algoritmy, ktoré sú opísané v sekcii analýza alebo nasledujúcej podkapitole spracovanie dát. Potom klienta požiadame, aby sa odfotil, či už pomocou video kamery na počítači alebo fotoaparátu na mobilnom telefóne a pomocou tejto fotky overíme, či odfotená fotka klienta sa podobá na fotku z dokladu totožnosti. Z tohto máme informáciu, či sme klienta verifikovali a môžeme pokračovať do ďalšej časti aplikácie.

4.2.2 Spracovanie dát

Dáta z dokladu totožnosti sú spracované pomocou platenej verzie OCR algoritmu, ktorý nám spoľahlivo dá všetky potrebné údaje. Z fotografie pomocou algoritmu vyberieme všetky údaje, ktoré sú pre nás podstatné a požiadame klienta, aby odfotil doklad totožnosti z druhej strany. Tieto údaje zahŕňajú meno a priezvisko, dátum narodenia, číslo dokladu totožnosti a ďalšie.

Nad týmito dátami robíme základnú validáciu, napríklad pre dátumy, ktoré sa overujú na reálne intervaly alebo nad číslom občianskeho, ktorý má v sebe tiež dátum a predom definovaný počet znakov.

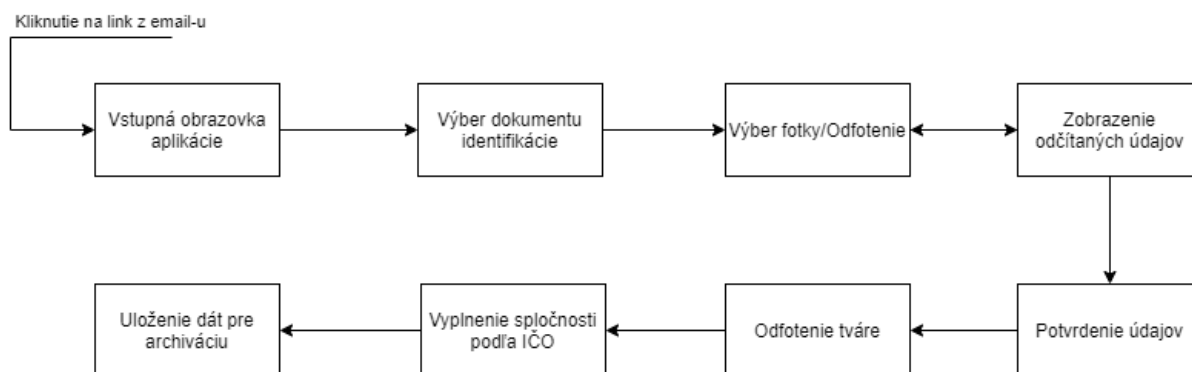
Tieto dáta dávame klientovi potvrdiť, pretože OCR algoritmus neposkytuje sto-percentnú úspešnosť a vtedy je klient povinný opraviť všetky údaje, ktoré mohli byť odčítané OCR algoritmom nesprávne.

Po potvrdení dát odčítaných z dokladu totožnosti je klient povinný pre úspešnú verifikáciu odfotiť svoju tvár. Fotka tváre je následne porovnaná s fotkou z dokladu totožnosti pomocou použitia umelej inteligencie na vyhodnotenie podobnosti fotiek.

Všetky tieto dáta nakoniec spojíme a odošleme na server, kde budú všetky verifikácie archivované. Tento krok je povinný zo zákona.

4.2.3 Popis logiky aplikácie

Aby sme sa pozreli na logiku celej aplikácie pre používateľa, máme tu nasledovný diagram, ktorý zobrazuje, aké kroky musí klient vykonať pre úspešnú automatizovanú verifikáciu pomocou našej navrhovanej aplikácie.



Obr. 4.1: Diagram chodu aplikácie

V prvom kroku musí používateľ kliknúť na link, ktorý mu môže prísť napríklad emailom. Ako sme už uviedli vyššie, tento link je špecificky generovaný pre každého klienta, pretože sú v ňom zahrnuté nejaké informácie pre našu aplikáciu, napríklad na priradenie verifikácie ku nášmu klientovi.

Po tomto kroku nasleduje otvorenie aplikácie a výber typu dokladu totožnosti, pomocou ktorého sa chce klient verifikovať. Po vybraní typu nasleduje odfotenie oboch strán

dokladu totožnosti a automatické odčítanie údajov, ktoré sa zobrazia na obrazovke používateľa. Ten je povinný tieto údaje potvrdiť.

Po dokončení predchádzajúceho kroku je klient vyzvaný odfoťiť si tvár tak, aby ju bolo možné overiť s tou, ktorá je na doklade totožnosti.

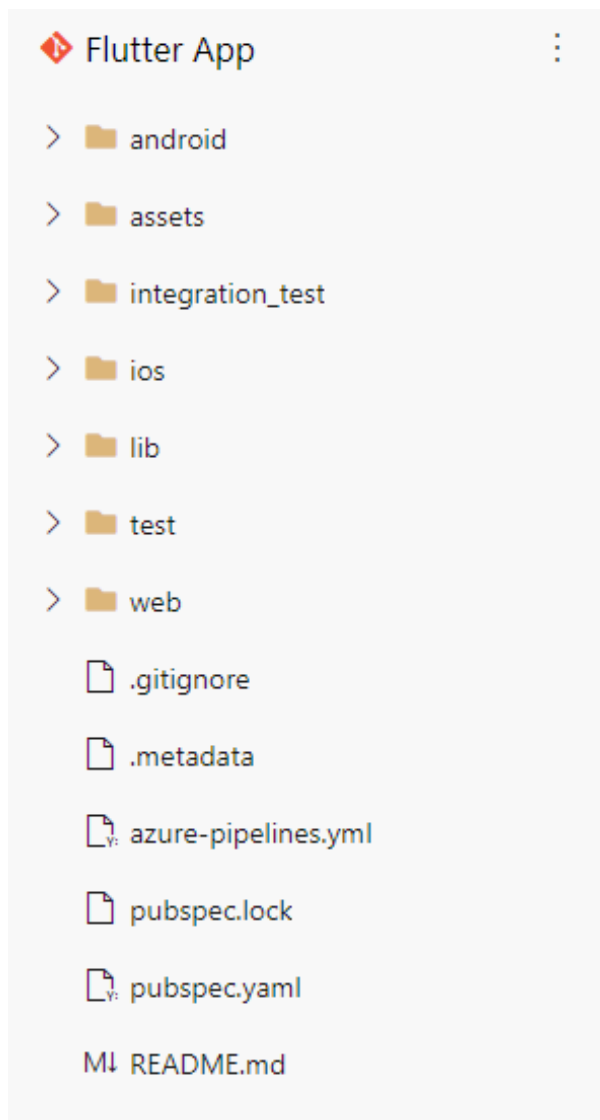
Na základe úspešného verifikovania všetkých potrebných údajov na konci odošleme záznam o úspešnej verifikácii na server kvôli spätnej dohľadateľnosti.

4.3 Implementácia

V tejto kapitole sa budeme venovať tomu, ako je naša aplikácia implementovaná. Potom niečo povieme o spôsobe, ako sú niektoré funkcionality navrhnuté, ďalej priblížime akým konkrétnym spôsobom je implementované OCR v našej aplikácii. Povieme tiež niečo o tom, ako je implementovaná komunikácia s databázami a ako posledné popíšeme, ktoré informácie je potrebné ukladať a ako.

Ako bolo spomenuté v časti návrhu, táto aplikácia je vyvinutá pomocou open-source vývojového balíka Flutter v jazyku Dart. Ako package manager sa používa PUB.

Z obrázka 4.2 vidíme štruktúru projektu v našom GIT repozitári.



Obr. 4.2: Ukážka repozitára

Obsah jednotlivých zložiek:

- *android* - súbory potrebné pre build android aplikácie. Sú tam konfiguračné súbory pre gradle a wrapper.
- *assets* - veci, ako napríklad obrázky, ktoré sú v aplikácií použité
- *integration_test* - všetky naše testy, ktoré sa spúšťajú.
- *ios* - podobná ako zložka android s tým, že sú tam konfiguračné súbory pre build iOS aplikácie.
- *lib* - naša hlavná zložka, v ktorej sa nachádzajú konkrétne .dart súbory, v ktorých je kód našej aplikácie. Pre každú obrazovku je definovaný separátny súbor a medzi týmito obrazovkami je implementovaný preklik, kedy sa načíta ďalší súbor. Toto riešenie zabezpečuje dobrú navigáciu v projekte a ľahšiu rozšíriteľnosť pre pridanie ďalšej obrazovky alebo nejakého kroku.
V tejto zložke sa tiež nachádza formulár ID karty a model s enumami, ktoré budeme neskôr potrebovať.
- *test* - unit testy, ktoré sa vykonávajú samé po pushnutí projektu do repozitára a tým sa kontrolujú čiastkové funkčnosti.
- *web* - keďže Flutter bol vyvinutý ako nezávislý balík na platforme, potrebujeme mať všetky potrebné typy platformy nakonfigurované. Naša aplikácia bola navrhnutá a je implementovaná tak, aby bola spustiteľná na Androide a iOS.

4.3.1 Funkcionality aplikácie

Tu by sme chceli opísať , akým spôsobom sú implementované niektoré funkcionality z našej aplikácie. Funkcionality, na ktoré sa pozrieme sú fotenie, integrovanie plateného OCR a komunikácia so servermi.

4.3.1.1 Fotenie

Pre spustenie funkcie fotenia v aplikácií bolo potrebné vyžiadať od používateľa potrebné povolenia napríklad v Androide. Potom sa fotoaparát spustí pomocou volania integrovanej knižnice MikroBlink, ktorá má volanie `scanWithCamera()`. Výsledok fotografie vráti táto

knižnica do Base64 stringu, s ktorým vieme ďalej niečo robiť. Využívame ho napríklad pre porovnanie tváre klienta a fotky občianskeho preukazu, kde táto knižnica vráti percento podobnosti, na základe ktorého vieme čiastočne vyhodnotiť, či je to tá istá osoba.

4.3.1.2 Integrácia OCR

Integrácia platenej OCR knižnice bola taktiež jednoduchá. Bolo potrebné importovať túto knižnicu a inicializovať `BlinkIdCombinedRecognizer()` a `BlinkIdOverlaySettings()`, kedy sa následne dalo použiť fotenie z tejto knižnice, ktoré je opísané v sekcii Fotenie.

4.3.1.3 Komunikácia s databázami

Naša aplikácia komunikuje s dvoma databázami. Prvou je externá databáza FinStat. FinStat je spoločnosť, ktorá agreguje údaje o slovenských firmách. V rámci našej aplikácie získavame z FinStatu údaje o firme na základe identifikačného čísla organizácie a následne zobrazujeme a ukladáme konkrétne údaje o firme.

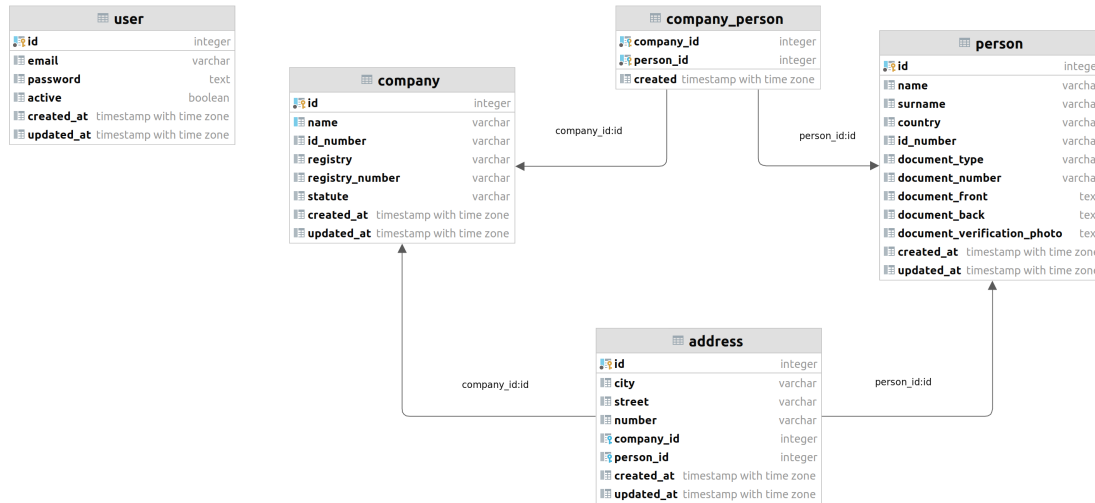
FinStat poskytuje REST aplikačné rozhranie, v rámci autentifikácie na API rozhraní je potrebné vytvoriť správny hash seed, ktorý sa skladá z saltu, verejného, privátneho API kľúča a hľadanej query, z tohto seedu je vypočítaný hash SHA256, ktorý je použitý v requeste. Výsledky sú vo formáte JSON, ktoré ukladáme v mobilnej aplikácii.

Druhou databázou je naša vlastná, ktorá slúži na ukladanie dát používateľov. Mobilná aplikácia komunikuje s naším backendom pomocou HTTP API POST a GET požiadavkami. Backend zabezpečuje logiku ukladania dát do databázy ako aj poskytovanie údajov pre mobilnú aplikáciu. Backend je riešený v jazyku Python pomocou rámca FastAPI. Databáza je PostgreSQL 14.1.

Prístup do databázy je umožnený tiež použitím nástrojov Adminer alebo pgAdmin. Dokumentácia backendu je generovaná automaticky pomocou Swagger. Celý backend stack je dostupný na našom servri v Docker kontajneroch a prístupný za použitia Nginx reverse proxy na užívateľsky prívetivých URL adresách.

4.3.1.4 Ukladanie informácií

Ako je uvedené vyššie, databáza je implementovaná v PostgreSQL 14.1, v ktorej sa ukládajú dáta z aplikácie do štruktúry podľa obrázka 4.3



Obr. 4.3: Ukážka dátového modelu

4.4 Testovanie

Testovanie tvorí jednu z najvýznamnejších častí vývoja softvéru. Slúži na overenie správnosti častí zdrojového kódu pomocou jednotkových testov a takisto slúži na uistenie sa, že projekt spĺňa očakávania klienta na základe vopred definovaných akceptačných testov.

4.4.1 Jednotkové testy

Jednotkové testy slúžia na odhalenie chýb na úrovni izolovaných jednotiek programu. Počas testovania porovnávame očakávaný a získaný výsledok a vďaka nej, vieme zabezpečiť správne fungovanie (jednotlivých častí) programu aj pre neočakávané vstupy.

4.4.2 Akceptačné testy

Nevyhnutným druhom testov sú akceptačné testy, pomocou ktorých overíme, či vyvíjaný softvér splní všetky akceptačné kritériá, čiže jeho prijateľnosť zo strany klienta.

1. Aplikácia má anglickú jazykovú verziu bez gramatických chýb.
2. Používateľ vie spustiť aplikáciu na telefóne s operačným systémom Android (od verzie 5.0).
3. Používateľ vie spustiť aplikáciu/proces overenia identity cez odkaz, ktorý dostane mailom.
4. Používateľ sa vie identifikovať pomocou svojho občianskeho preukazu.
5. Používateľ musí potvrdiť správnosť načítaných údajov z dokladu totožnosti na pokračovanie v procese overenia svojej identity.
6. Používateľ sa vie identifikovať pomocou svojho cestovného pasu.
7. Používateľ vie zadať údaje o svojej totožnosti, ktoré nie sú získavané z dokladov totožnosti.
8. Používateľ sa nevie identifikovať pomocou iných preukazov, ktoré neboli vydané štátom (ISIC karta, preukaz zamestnanca, preukaz na MHD,..).
9. Používateľ vie odstrániť z aplikácie už naskenovaný doklad totožnosti a nafotiť iný.

10. Používateľ vie opraviť nesprávne prečítané údaje z dokladu totožnosti.
11. Používateľ vie odfoťiť svoju tvár na identifikáciu svojej totožnosti bez opustenia aplikácie.
12. Aplikácia umožňuje získanie údajov o používateľovi z Finstatu.

Scenáre akceptačných testov

Na úspešné prejdenie akceptačného testu (Pass), test musí skončiť očakávaným výsledkom, inak test berieme ako neúspešný (Fail).

Aplikácia má anglickú jazykovú verziu bez gramatických chýb.

1. **Očakávaný výsledok:** Všetky obrazovky aplikácie sú v anglickom jazyku a neobsahujú gramatické chyby

Story	Criteria	True value
Prejdite aplikáciu.	Aplikácia má anglickú jazykovú verziu bez gramatických chýb	Všetky obrazovky aplikácie sú v anglickom jazyku a neobsahujú gramatické chyby

Používateľ vie spustiť aplikáciu na telefóne s operačným systémom Android (od verzie 5.0).

1. Spustite aplikáciu.
2. **Očakávaný výsledok:** Je zobrazená uvítacia obrazovka aplikácie.

Story	Criteria	True value
Spustite aplikáciu.	Spustenie aplikácie	Spustená aplikácia, zobrazená uvítacia obrazovka s tlačidlom 'Identify me!'

Používateľ vie spustiť aplikáciu/proces overenia identity cez odkaz, ktorý dostane mailom.

***Predpoklady:** Otvorený email na telefóne obsahujúci odkaz na aplikáciu*

1. Kliknite na odkaz v emaili.
2. **Očakávaný výsledok:** Aplikácia je spustená (je zobrazená uvítacia obrazovka aplikácie) s nastavenou premenou, ktorá obsahuje zoznam čísel IČO firiem, ktorých zastupuje používateľ.

Story	Criteria	True value
Kliknite na odkaz v emaili.	Spustenie aplikácie pomocou odkazu	Spustená aplikácia, zobrazená uvítacia obrazovka s tlačidlom "Identify me!" a nastavená premenná, ktorá obsahuje zoznam čísel IČO firiem, ktorých zastupuje používateľ

Používateľ sa vie identifikovať pomocou svojho občianskeho preukazu.

Používateľ musí potvrdiť správnosť načítaných údajov z dokladu totožnosti na pokračovanie v procese overenia svojej identity.

1. Spustite aplikáciu.
2. Kliknite tlačidlo 'Identify me'!
3. Naskenujte prednú stranu svojho občianskeho preukazu.
4. Naskenujte zadnú stranu svojho občianskeho preukazu.
5. Kliknite tlačidlo 'Confirm' na potvrdenie správnosti údajov.
6. **Očakávaný výsledok:** Do databázy sa uloží nová osoba s dokladom totožnosti používateľa, s načítanými údajmi z občianskeho preukazu a časovou pečiatkou uloženia do databázy, ktorá je vo formáte yyyy-MM-dd HH:mm:ss.SSSSSS+ZZ.

Medzi načítané údaje patria:

- First name – String
- Last name – String
- Address – Adress line1, City – String
- Document number – String vo formáte AA000000
- Sex – List (M/F)
- Issuing authority – String
- Nationality – String
- Date of Birth – vo formáte DD/MM/YYYY
- Age – Integer
- Date of Issue – vo formáte DD/MM/YYYY
- Date of Expiry – vo formáte DD/MM/YYYY
- Personal Id Number – String vo formáte YYMMDD/0000

Story	Criteria	True value
Spustite aplikáciu.	Spustenie aplikácie	Spustená aplikácia, zobrazená uvítacia obrazovka s tlačidlom "Identify me!"
Kliknite tlačidlo 'Identify me'!	Otvorenie fotoaparátu	Otvorený fotoaparát
Naskenujte prednú stranu svojho občianskeho preukazu.	Rozpoznanie priezviska Rozpoznanie mena Rozpoznanie štátneho občianstva Rozpoznanie pohlavia Rozpoznanie čísla OP Rozpoznanie dátumu narodenia Rozpoznanie rodného čísla Rozpoznanie dátumu platnosti OP Rozpoznanie dátumu vydania OP Rozpoznanie vydavateľa OP	Potvrdenie správnosti vo formáte String Potvrdenie správnosti vo formáte String Potvrdenie správnosti vo formáte String Potvrdenie správnosti: buď 'F' alebo 'M' Potvrdenie správnosti vo formáte AA000000 Potvrdenie správnosti vo formáte DD/MM/YYYY Potvrdenie správnosti vo formáte YYMMDD/0000 Potvrdenie správnosti vo formáte DD/MM/YYYY Potvrdenie správnosti vo formáte DD/MM/YYYY Potvrdenie správnosti vo formáte String
Naskenujte zadnú stranu svojho občianskeho preukazu.	Rozpoznanie adresy	Potvrdenie správnosti vo formáte dvoch Stringov: Address line 1 a City
Kliknite tlačidlo 'Confirm' na potvrdenie správnosti údajov.	Zobrazenie načítaných údajov Uloženie novej osoby s dokladom totožnosti používateľa do databázy	Zobrazenie polí s hodnotami z OP ('First name', 'Last name', 'Address', 'Document number', 'Sex', 'Issuing authority', 'Nationality', 'Date of Birth', 'Age', 'Date of Issue', 'Date of Expiry', 'Personal Id Number') a tlačidla na potvrdenie ich správnosti. V databáze je uložená nová osoba s dokladom totožnosti používateľa, s načítanými údajmi z OP a časovou pečiatkou uloženia do databázy. Medzi načítané údaje patria: 'First name', 'Last name', 'Address', 'Document number', 'Sex', 'Issuing authority', 'Nationality', 'Date of Birth', 'Age', 'Date of Issue', 'Date of Expiry', 'Personal Id Number'.

Používateľ sa vie identifikovať pomocou svojho cestovného pasu.

Používateľ vie zadať údaje o svojej totožnosti, ktoré nie sú získavané z dokladov totožnosti.

1. Spustite aplikáciu.
2. Kliknite tlačidlo 'Identify me'!
3. Naskenujte svoj cestovný pas.
4. Zadaajte 'Ilkovičova 2, 842 16 Bratislava' do pola 'Address'.
5. Kliknite tlačidlo 'Confirm' na potvrdenie správnosti údajov.
6. **Očakávaný výsledok:** Do databázy sa uloží nová osoba s dokladom totožnosti používateľa, s:

- načítanými údajmi z cestovného pasu,
- manuálne zadanou adresou vo formáte Stringov: Address line 1 a City
- a časovou pečiatkou uloženia do databázy

Medzi načítané údaje patria:

- First name – String
- Last name – String
- Document number – String vo formáte AA0000000
- Sex – List (M/F)
- Issuing authority – String
- Nationality – String
- Date of Birth – vo formáte DD/MM/YYYY
- Age – Integer
- Date of Issue – vo formáte DD/MM/YYYY
- Date of Expiry – vo formáte DD/MM/YYYY
- Personal Id Number – String vo formáte YYMMDD/0000

Story	Criteria	True value
Spustite aplikáciu.	Spustenie aplikácie	Spustená aplikácia, zobrazená uvítacia obrazovka s tlačidlom "Identify me!"
Kliknite tlačidlo 'Identify me'!	Otvorenie fotoaparátu	Otvorený fotoaparát
Naskenujte svoj cestovný pas.	Rozpoznanie priezviska Rozpoznanie mena Rozpoznanie štátneho občianstva Rozpoznanie pohlavia Rozpoznanie čísla pasu Rozpoznanie dátumu narodenia Rozpoznanie rodného čísla Rozpoznanie dátumu platnosti pasu, Rozpoznanie dátumu vydania pasu Rozpoznanie vydavateľa pasu	Potvrdenie správnosti vo formáte String Potvrdenie správnosti vo formáte String Potvrdenie správnosti vo formáte String Potvrdenie správnosti: buď 'F' alebo 'M' Potvrdenie správnosti vo formáte AA000000 Potvrdenie správnosti vo formáte DD/MM/YYYY Potvrdenie správnosti vo formáte YYMMDD/0000 Potvrdenie správnosti vo formáte DD/MM/YYYY Potvrdenie správnosti vo formáte DD/MM/YYYY Potvrdenie správnosti vo formáte String
Zadajte 'Ilkovičova 2, 842 16 Bratislava' do pola 'Address'.	Zobrazenie načítaných údajov Zadanie adresy používateľa	Zobrazenie polí s hodnotami z pasu ('First name', 'Last name', 'Document number', 'Sex', 'Issuing authority', 'Nationality', 'Date of Birth', 'Age', 'Date of Issue', 'Date of Expiry', 'Personal Id Number') a tlačidla na potvrdenie ich správnosti. Potvrdenie rovnosti hodnoty pola Address s hodnotou "Ilkovičova 2, 842 16 Bratislava"
Kliknite tlačidlo 'Confirm' na potvrdenie správnosti údajov.	Uloženie novej osoby s dokladom totožnosti používateľa do databázy	V databáze je uložená nová osoba s dokladom totožnosti používateľa, s načítanými údajmi z pasu, manuálne zadanou adresou a časovou pečiatkou uloženia do databázy. Medzi načítané údaje patria: 'First name', 'Last name', 'Document number', 'Sex', 'Issuing authority', 'Nationality', 'Date of Birth', 'Age', 'Date of Issue', 'Date of Expiry', 'Personal Id Number'.

Používateľ sa nevie identifikovať pomocou iných preukazov, ktoré neboli vydané štátom (ISIC karta, preukaz zamestnanca, preukaz na MHD,..).

1. Spustite aplikáciu.
2. Kliknite tlačidlo 'Identify me!'.
3. Naskenujte svoju ISIC kartu/preukaz zamestnanca/preukaz na MHD.
4. **Očakávaný výsledok:** Zobrazí sa informačná hláška, že dokument nebol rozpoznaný a ponúka možnosť skúsiť znova.

Story	Criteria	True value
Spustite aplikáciu.	Spustenie aplikácie	Spustená aplikácia, zobrazená uvítacia obrazovka s tlačidlom 'Identify me!'
Kliknite tlačidlo 'Identify me'!	Otvorenie fotoaparátu	Otvorený fotoaparát
Naskenujte svoju svoju ISIC kartu/preukaz zamestnanca/preukaz na MHD.	Neakceptovanie preukazu, ktorá nebola vydaná štátom.	Zobrazená informačná hláška, že dokument nebol rozpoznaný spolu s tlačidlom skúsiť znova.

Používateľ vie odstrániť z aplikácie už naskenovaný doklad totožnosti a nafotiť iný.

1. Spustite aplikáciu.
2. Kliknite tlačidlo 'Identify me'!
3. Naskenujte prednú stranu občianskeho preukazu.
4. Naskenujte zadnú stranu občianskeho preukazu.
5. Stlačte tlačidlo späť na odstránenie naskenovaného dokladu.
6. Naskenujte prednú stranu občianskeho preukazu.
7. Naskenujte zadnú stranu občianskeho preukazu.
8. Kliknite tlačidlo 'Confirm' na potvrdenie správnosti údajov.
9. **Očakávaný výsledok:** Všetky údaje súvisiace s občianskym preukazom, ktorý bol naskenovaný ako prvý boli odstránené z aplikácie. Do databázy sú uložené iba načítané údaje z občianskeho preukazu, ktorý bol naskenovaný druhýkrát. Medzi načítané údaje patria: 'First name', 'Last name', 'Address', 'Document number', 'Sex', 'Issuing authority', 'Nationality', 'Date of Birth', 'Age', 'Date of Issue', 'Date of Expiry', 'Personal Id Number'.

Story	Criteria	True value
Spustite aplikáciu.		
Kliknite tlačidlo 'Identify me'!		
Naskenujte prednú stranu svojho občianskeho preukazu.		
Naskenujte zadnú stranu svojho občianskeho preukazu.		
Stlačte tlačidlo späť na odstránenie naskenovaného dokladu.	Zobrazenie načítaných údajov Odstránenie už naskenovaného občianskeho preukazu	Všetky údaje súvisiace s občianskym preukazom, ktorý bol naskenovaný ako prvý boli odstránené z aplikácie a je znova otvorený fotoaparát
Naskenujte prednú stranu svojho občianskeho preukazu.		
Naskenujte zadnú stranu svojho občianskeho preukazu.		
Kliknite tlačidlo 'Confirm' na potvrdenie správnosti údajov.		

Používateľ vie opraviť nesprávne prečítané údaje z dokladu totožnosti.

1. Spustíte aplikáciu.
2. Kliknite tlačidlo 'Identify me'!
3. Naskenujte prednú stranu svojho občianskeho preukazu.
4. Naskenujte zadnú stranu svojho občianskeho preukazu.
5. Zmeňte hodnotu pola 'Address' na 'Ilkovičova 2, 842 16 Bratislava'.
6. Kliknite tlačidlo 'Confirm' na potvrdenie správnosti údajov.
7. **Očakávaný výsledok:** Do databázy sa uloží nová osoba s dokladom totožnosti používateľa, s časovou pečiatkou uloženia do databázy a načítanými údajmi z občianskeho preukazu, pričom hodnota pola 'Address' je zmenená na 'Ilkovičova 2, 842 16 Bratislava'. Medzi načítané údaje patria: 'First name', 'Last name', 'Address', 'Document number', 'Sex', 'Issuing authority', 'Nationality', 'Date of Birth', 'Age', 'Date of Issue', 'Date of Expiry', 'Personal Id Number'.

Story	Criteria	True value
Spustíte aplikáciu.		
Kliknite tlačidlo 'Identify me'!		
Naskenujte prednú stranu svojho občianskeho preukazu.		
Naskenujte zadnú stranu svojho občianskeho preukazu.		
Zmeňte hodnotu pola 'Address' na 'Ilkovičova 2, 842 16 Bratislava'.	Zobrazenie načítaných údajov Zmena načítanej hodnoty	Potvrdenie rovnosti hodnoty pola Address s hodnotou "Ilkovičova 2, 842 16 Bratislava"
Kliknite tlačidlo 'Confirm' na potvrdenie správnosti údajov.		

Používateľ vie odfoťiť svoju tvár na identifikáciu svojej totožnosti bez opustenia aplikácie.

1. Spustite aplikáciu.
2. Kliknite tlačidlo 'Identify me'!
3. Naskenujte prednú stranu svojho občianskeho preukazu.
4. Naskenujte zadnú stranu svojho občianskeho preukazu.
5. Kliknite tlačidlo 'Confirm' na potvrdenie správnosti údajov.
6. Kliknite tlačidlo 'Continue' na spustenie kamery.
7. Odfotíte si svoj tvár.
8. Potvrďte vhodnosť obrázka.
9. **Očakávaný výsledok:** Aplikácia zobrazuje text 'Success We have managed to identify you.'

Story	Criteria	True value
Spustite aplikáciu.		
Kliknite tlačidlo 'Identify me'!		
Naskenujte prednú stranu svojho občianskeho preukazu.		
Naskenujte zadnú stranu svojho občianskeho preukazu.		
Kliknite tlačidlo 'Confirm' na potvrdenie správnosti údajov.		
Kliknite tlačidlo 'Continue' na spustenie kamery.	Zobrazenie inštrukcií k tvárovej biometrii Spustenie predného fotoaparátu	Zobrazená obrazovka s inštrukciami k tvárovej biometrii Otvorený predný fotoaparát
Odfotožte si svoju tvár.	Odfotenie tváre	Zobrazený teraz nafotený obrázok tváre
Potvrďte vhodnosť obrázka.	Overenie identity na základe tvárovej biometrii Uloženie fotografie tváre do databázy	Zobrazená obrazovka s textom "Success We have managed to identify you" Uložená fotografia tváre používateľa spolu s úspešnosťou a časovou pečiatkou biometrii v databáze

Aplikácia umožňuje získanie údajov o používateľovi z Finstatu.

Predpoklady: *Otvorený email na telefóne obsahujúci odkaz na aplikáciu*

1. Kliknite na odkaz v emaili.
2. Kliknite tlačidlo 'Identify me'!
3. Naskenujte prednú stranu svojho občianskeho preukazu.
4. Naskenujte zadnú stranu svojho občianskeho preukazu.
5. Kliknite tlačidlo 'Confirm' na potvrdenie správnosti načítaných údajov.
6. Kliknite tlačidlo 'Continue' na spustenie kamery.
7. Odfotoťte si svoju tvár.
8. Potvrďte vhodnosť obrázka.
9. Kliknite tlačidlo 'Continue' na pokračovanie.
10. **Očakávaný výsledok:** Zobrazí sa obrazovka so zoznamom spoločnosti, je zobrazené ich meno a IČO, ktoré reprezentujú spoločnosti pre ktoré daný používateľ bol verifikovaný.

Story	Criteria	True value
Kliknite na odkaz v emaili.		
Kliknite tlačidlo 'Identify me'!		
Naskenujte prednú stranu svojho občianskeho preukazu.		
Naskenujte zadnú stranu svojho občianskeho preukazu.		
Kliknite tlačidlo 'Confirm' na potvrdenie správnosti údajov.		
Kliknite tlačidlo 'Continue' na spustenie kamery.		
Odfotíte si svoj tvár.		
Potvrďte vhodnosť obrázka.		
Kliknite tlačidlo 'Continue' na pokračovanie.	Získanie údajov o používateľovi z FinStatu	Zobrazená obrazovka so zoznamom spoločnosti, je zobrazené ich meno a IČO, ktoré reprezentujú spoločnosti pre ktoré daný používateľ bol verifikovaný.

Časť II

Riadenie projektu

5 Úvod

Tento dokument sa zaoberá riadením tímu na projekte KLYCK. Dokument obsahuje roly všetkých členov tímu ako aj podiel jednotlivých členov na dokumentácii. V nasledujúcej časti sú opísané jednotlivé manažmenty alebo činnosti, potrebné pre riadenie projektu. Progres na projekte môžeme vidieť v sumarizáciách šprintov. Súčasťou je aj globálna retrospektíva za zimný semester. Ako prílohou tohto dokumentu sú motivačný dokument, metodiky, exporty evidencií úloh a webové sídlo projektu.

6 Roly členov tímu a podiel práce

6.1 Roly členov tímu

Meno a priezvisko	Manažér	Rola v tíme
Filip Agh	Manažér vývoja	Back-End Developer
Adam Benovič	Manažér vývoja	Back-End Developer
Ľuboš Daniš	Manažér úloh	Team Leader, Scrum Master
Lenka Ivanová	Manažér dokumentácie	Back-End Developer
Aranka Klukaová	Manažér testovania	UX, Tester
Patrik Vdoviak	Manažér webového sídla	Front-End Developer

6.2 Podiel práce na dokumentácii

Meno a priezvisko	Percentuálny podiel
Filip Agh	16.6%
Adam Benovič	16.6%
Ľuboš Daniš	16.6%
Lenka Ivanová	16.6%
Aranka Klukaová	16.6%
Patrik Vdoviak	16.6%

7 Aplikácie manažmentov

Pre opis činností pre riadenie projektu využívame metodiky. V metodikách sú jednotlivé činnosti opísané podrobnejšie. Všetky metodiky sú priebežne aktualizované a nájdeme ich na webovom sídle projektu v časti „Metodiky”.

7.1 Manažment splnenia úloh

Pre rôzne typy úloh sme si zadefinovali kritériá, ktoré musia byť splnené, aby sme mohli povedať, že úloha je dokončená. Súbor takýchto kritérií sa nazýva Definition of Done. Definition of Done pre rôzne časti projektu nájdeme v prílohe B.

7.2 Manažment komunikácie

Na komunikáciu používame viacero kanálov. Hlavným kanálom je Microsoft Teams, kde mávame stretnutia s vedúcim tímu a ďalšími členmi organizácie BDO. Ďalej používame Discord na naše osobné stretnutia, hlavne vtedy, keď sa nemôžeme všetci stretnúť prezenčne. Okrem toho používame aj Messenger, kde riešime súkromné záležitosti ohľadom tímu.

Ďalšie komunikačné kanály sa nachádzajú v prílohe C.

7.3 Manažment práce s úlohami

Hlavným nástrojom na prácu a tvorbu úloh je Azure DevOps. V nástroji si vieme vytvárať šprinty, do ktorých priradíme úlohy. Každá úloha je niekomu priradená a je ocenená Effort-om and Remaining work. Každá úloha sa vždy nachádza v nejakom stave. Stav je možné meniť, podľa toho, aký progres bol na úlohe doposiaľ urobený.

V Metodike práce s úlohami v prílohe D sú všetky vyššie spomenuté položky opísané podrobnejšie.

7.4 Manažment testovania

Testovanie je nevyhnutné, pretože všetci sme len ľudia a robíme chyby. Niektoré chyby môžu byť dôležité, iné menej. Pre nás je dôležité, aby projekt spĺňal očakávania klienta. Testy delíme na jednotkové a akceptačné. Pri vytváraní testov a testovaní sa riadime zavedenou Metodikou testovania. Okrem navrhnutých jednotkových a akceptačných testov tiež testujeme aplikáciu manuálne.

Podrobnejšie informácie nájdeme v Metodike testovania. Metodiku testovania nájdeme v prílohe E.

7.5 Manažment verziovania

Pre verziovanie programov používame GIT ako version control system a Azure DevOps ako manažér pre version control system.

Repozitáre sú primárne rozdelené na dve časti. Prvou je repozitár web stránky tímu. Druhou sú repozitáre implementácie nášho riešenia projektu.

Ďalšie informácie ohľadom verziovania, repozitárov, vetvenia, rôl a pravidiel verziovania nájdeme v prílohe F.

7.6 Manažment vývoja

Hlavný vývoj multiplatformovej aplikácie je zabezpečený v open-source používateľskom rozhraní Flutter. Programovací jazyk Python používame len na časť identifikácie používateľa.

Stanovené pravidlá pre písanie kódu máme vypracované v Metodike vývoja v prílohe G.

8 Sumarizácie šprintov

8.1 Šprint č.1

Prvý šprint prebiehal v čase 12.10.2021 - 25.10.2021.

V prvom šprinte sme sa venovali predovšetkým zisťovaniu informácií k tímovému projektu, plánovaniu šprintov a prvotným jednoduchým implementáciám. Vzhľadom k tomu, že prvé stretnutie s vedúcim tímu sa konalo trochu neskôr, posunul sa nám aj štart prvého šprintu. Keďže už bolo nutné šprint začať kvôli časovému harmonogramu predmetu Tímový projekt, prvá polovica šprintu pozostávala ešte z oboznamovania sa s podmienkami predmetu. V druhej polovici sme vykonávali úlohy, ktoré sme si naplánovali v prvom týždni šprintu. Z tohto dôvodu boli úlohy vykonávané a zaznamenané v Azure DevOps prevažne v druhom týždni šprintu, čo je možné vidieť aj v Burndown charte (Obr. 8.1.)

V prvom šprinte sa nám podarilo vytvoriť návrh webovej stránky a základnú podobu webovej stránky podľa tohto návrhu. Vytvorili sa šablóny pre zápisnice a metodiky. Zápisnice sa prepísali do formálnejšej podoby. Napísalo sa niekoľko metodík, niekoľko ďalších metodík sa presunulo do druhého šprintu kvôli časovej tiesni. Kompletne sme preštudovali dokumenty k predmetu Tímový projekt a aj na základe toho sme spravili plánovanie. Vývojári sa venovali prvotným implementáciám, kde si vyskúšali OCR. Okrem toho si preštudovali best practices pre Flutter, ktorý sme si zvolili na vývoj.

Prvý šprint považujeme za úspešný s prihliadnutím na počiatočné podmienky. Napriek tomu si uvedomujeme, že máme v celom procese ešte medzery a budeme sa snažiť ich v ďalších šprintoch minimalizovať a stále sa zlepšovať.

Vybrané user stories

V prvom šprinte sme si vybrali user stories, ako je možné vidieť v Tabuľke 8.1. Podarilo sa vykonať všetky user stories alebo ich časť okrem jednej, týkajúcej sa metodík. Úlohy priradené danej user story, ktoré sa nestihli vykonať, sa presúvajú do druhého šprintu.

Názov	SP	Schválená
Ako používateľ chcem vedieť spustiť aplikáciu na overenie mojej identity	2	Áno
Ako vývojár potrebujem získať dáta fotky do aplikácie na ich spracovanie	3	Áno
Ako vývojár chcem použiť OCR na fotku dokladu totožnosti na extrahovanie dát	5	Áno
Ako tím potrebujeme návrh modelu na vytvorenie webovej reprezentácie projektu	2	Áno
Ako product owner chcem týždennú aktualizáciu webu na zabezpečenie aktuálnosti informácií	2	Áno
Ako tím potrebujeme vkladať dokumenty webovú stránku na sprístupnenie informácií verejnosti	2	Áno
Ako product owner chcem webovú stránku na vhodnú reprezentáciu nášho tímu	4	Áno
Ako tím potrebujeme vytvárať zápisy za účelom dokumentácie	11	Áno
Ako tím potrebujeme metodiky na zefektívnenie práce a ľahšie dodržiavanie dohodnutých pravidiel	10	Nie
Ako tím potrebujeme preštudovať požiadavky na úspešné splnenie predmetu	3	Áno
Ako vývojár potrebujem preštudovať Flutter Best Practices kvôli správneému vývoju	6	Áno
Ako tím potrebujeme návrh aplikácie na ďalší vývoj	2	Áno

Tabuľka 8.1: Vybrané user stories šprintu č.1

Práca na šprinte

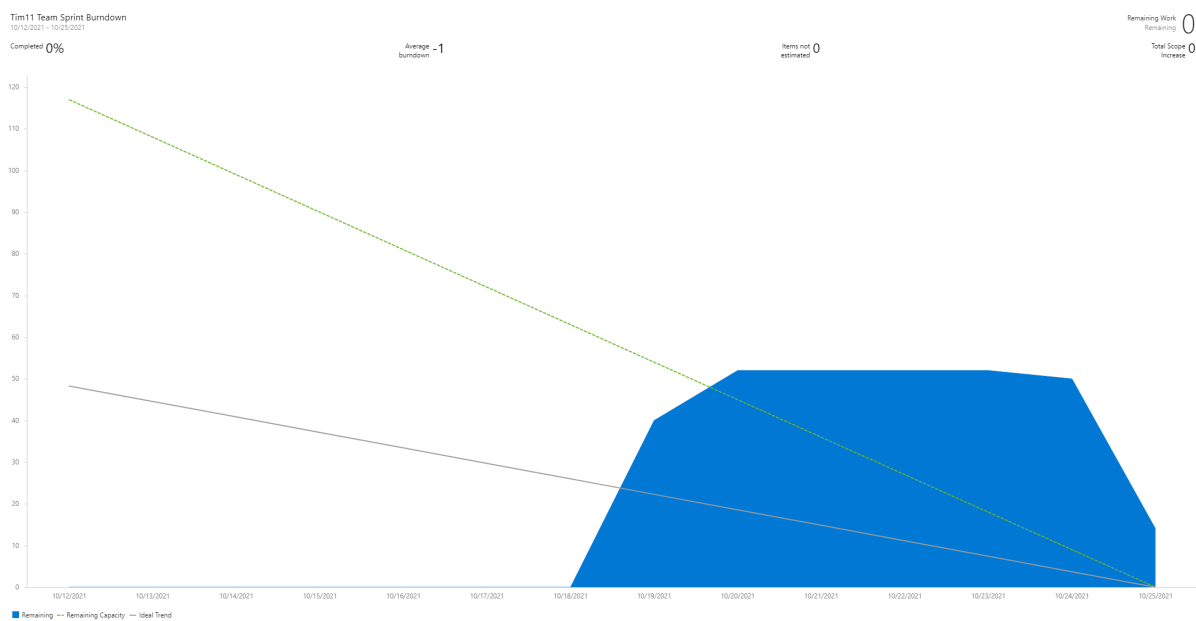
V Tabuľke 8.2 si môžeme všimnúť, ako pracovali jednotliví členovia tímu na šprinte. Každý člen dostal určitý počet úloh a tím prispel svojim podielom k celkovej práci. Členovia s nižším počtom úloh mali väčšinou na starosti úlohy, ktoré si vyžadovali viac času.

Meno a priezvisko	Počet úloh	Podiel práce (%)
Filip Agh	5	17%
Adam Benovič	2	17%
Ľuboš Daniš	4	17%
Lenka Ivanová	10	16%
Aranka Klukaová	9	16%
Patrik Vdoviak	6	17%

Tabuľka 8.2: Práca na šprinte č.1

Burndown chart

Na Obrázku 8.1 môžeme vidieť Burndown chart po prvom šprinte.



Obr. 8.1: Burndown chart pre šprint č.1

Export úloh

Export úloh po prvom šprinte je možné nájsť na webovom sídle tímu 11.

8.2 Šprint č.2

Trvanie šprintu: 26.10.2021 - 9.11.2021

V druhom šprinte sme sa venovali hlavne implementácii funkcionality v aplikácii a príprave dokumentov na odovzdanie v Míľniku 1. Z pohľadu projektu sa implementovalo získanie údajov z dokladu totožnosti a následné predvyplnenie týchto údajov do polí vo formulári na identifikáciu osoby. Podarilo sa urobiť porovnanie odfotenej fotografie s fotografiou z dokladu totožnosti. Implementovali sa obrazovky, na ktorých celý proces prebieha. Pre potreby pokračovania na projekte sa vykonalo preštudovanie dokumentácie API Finstatu. Posledná aktuálna verzia bola zároveň manuálne otestovaná.

Podstatnú časť šprintu tvorila tvorba dokumentov, resp. ich prvá podoba, ktoré budú súčasťou väčších dokumentov (inžinierske dielo a projektové riadenie) v Míľniku 1.

V rámci šprintu sa samozrejme vytvárali aj zápisnice a retrospektíva predošlého šprintu. Webové sídlo sa snažíme udržiavať aktuálne v rámci časových možností. Tiež sa vytvoril export úloh po predošlom šprinte.

Druhý šprint považujeme za úspešný a lepší ako ten predošlý. Z pohľadu naplánovaných úloh sme stihli väčšinu. Nepodarili sa spraviť dve úlohy, ktoré patria pod dokumentáciu. Z tohto dôvodu sa dve user stories presúvajú do ďalšieho šprintu.

Z poslednej retrospektívy sme sa snažili dodržať to, čo sme si povedali. Vďaka zavedeniu stand-upu mimo oficiálnych stretnutí sa nám podarilo robiť prácu častejšie, častejšie ju aj aktualizovať, čo sa pekne prejavilo aj na Burndown charte. Tiež sme zlepšili komunikáciu v situáciách, keď niekoho úloha závisí od ukončenia úlohy iného člena. Napriek citeľnému zlepšeniu procesov sme identifikovali nové nedostatky, ktoré by sme radi v budúcnosti odstránili.

Vybrané user stories

V Tabuľke 8.3 môžeme vidieť vybrané user stories pre 2.šprint. Nepodarilo sa vykonať dve úlohy z dvoch user stories, preto sa presúvajú do ďalšieho šprintu. Tieto úlohy sa týkajú dokumentácie.

Názov	SP	Schválená
Ako používateľ chcem, aby moje údaje boli získané z môjho dokladu totožnosti na identifikáciu mojej osoby	5	Áno
Ako product owner chcem vyhľadávanie údajov z FinStatu na identifikáciu používateľa	8	Áno
Ako používateľ chcem, aby mi boli zobrazené moje dáta získané z dokladu totožnosti na ich overenie	3	Áno
Ako product owner chcem týždennú aktualizáciu webu na zabezpečenie aktuálnosti informácií	8	Áno
Ako product owner chcem dokumentáciu aplikácie	3	Áno
Ako tím potrebujeme metodiky na zefektívnenie práce a ľahšie dodržiavanie dohodnutých pravidiel	13	Áno
Ako tím potrebujeme preštudovať požiadavky na úspešné splnenie predmetu	1	Áno
Ako používateľ chcem aplikáciu bez bugov na predchádzanie problémov, ktoré by mohli spôsobiť	3	Áno
Ako vývojár potrebujem preštudovať Flutter Best Practices kvôli správne mu vývoju	3	Áno
Ako product owner chcem navigačnú kostru aplikácie na získanie lepšieho prehľadu o vývoji	13	Áno
Ako product owner chcem overovanie tváre používateľa voči jeho občianskemu preukazu na overovanie jeho identity	20	Áno
Ako vývojár chcem implementáciu integrácie blinku	13	Áno
Ako tím chceme napísať prihlášku na zaregistrovanie sa do súťaže „TP Cup 2022“	5	Áno
Ako zákazník požadujem dokumentáciu šprintov	8	Áno
Ako tím potrebujeme vytvoriť dokument inžinierskeho diela	20	Nie
Ako tím potrebujeme vytvoriť dokument projektového riadenia	20	Nie

Tabuľka 8.3: Vybrané user stories šprintu č.2

Práca na šprinte

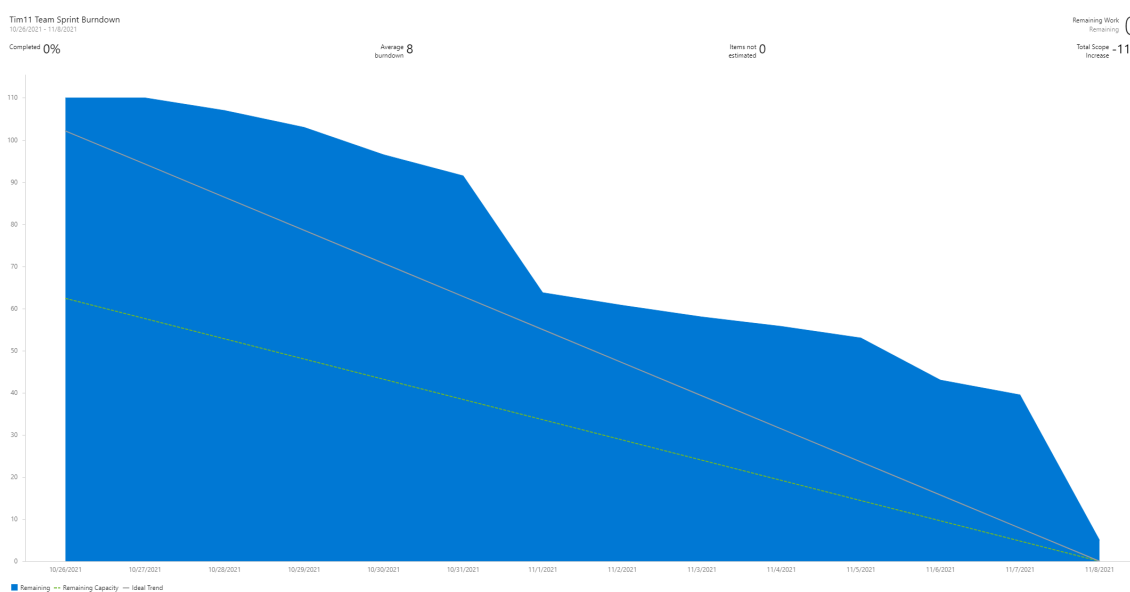
V Tabuľke 8.4 si môžeme všimnúť, ako pracovali jednotliví členovia tímu na šprinte. Každý člen dostal určitý počet úloh a tým prispel svojim podielom k celkovej práci. Členovia s nižším počtom úloh mali väčšinou na starosti úlohy, ktoré si vyžadovali viac času.

Meno a priezvisko	Počet úloh	Podiel práce (%)
Filip Agh	16	17%
Adam Benovič	10	17%
Ľuboš Daniš	7	17%
Lenka Ivanová	13	17%
Aranka Klukaová	6	16%
Patrik Vdoviak	8	16%

Tabuľka 8.4: Práca na šprinte č.2

Burndown chart

Na Obr.8.2 môžeme vidieť Burndown chart po 2.šprinte. Ako môžeme vidieť, prácu sa podarilo vykonávať pravidelnejšie. Počas šprintu sme objavili úlohy, ktoré boli nesprávne zaradené do 2.šprintu, preto počas behu 2.šprintu došlo k zníženiu počtu úloh. Dve úlohy sa nepodarilo stihnúť vypracovať.



Obr. 8.2: Burndown chart pre šprint č.2

Export úloh

Export úloh po druhom šprinte je možné nájsť na webovom sídle tímu 11.

8.3 Šprint č.3

Trvanie šprintu: 9.11.2021 - 22.11.2021

V treťom šprinte sme sa venovali hlavne príprave dokumentov na odovzdanie v Míľniku 1, ale aj implementácii novej funkcionality v aplikácii. Z pohľadu aplikácie sa podarilo naimplementovať spustenie aplikácie pomocou linku a vytvoriť unit testy pre aplikáciu. Ako ďalšie sa podarilo vytvoriť spojenie s FinStatom, získať a zobrazovať údaje z FinStatu. Spojenie bolo navyše otestované a zdokumentované. Ďalšiu časť práce na aplikácii tvorilo vytvorenie úložiska dát, kde sa ukladajú dáta získavané pri procese identifikácie a verifikácie klienta. Úložisko bolo taktiež riadne navrhnuté, otestované, zdokumentované a nasadené.

Najväčšiu prioritu mali v tomto šprinte dokumenty pre Míľnik 1. Spracovávali sme 2 veľké dokumenty a to Inžinierske dielo a Riadenie projektu. Jednotlivé časti dokumentov, ako sú uvedené v podmienkach predmetu, sme si rozdelili medzi ostatných členov. Dokumenty sa podarilo kompletne vypracovať podľa plánu.

Okrem spomenutých vecí sa samozrejme pracovalo aj na udržiavaní aktuálneho webového sídla tímového projektu, kde sa pridávali zápisnice, retrospektíva minulého šprintu a exporty úloh.

Tento šprint považujeme za úspešný, s prihliadnutím na fakt, že nedokončené 2 úlohy súviseli s práce neschopnosťou jedného z členov. Všetky ostatné úlohy sa podarili vypracovať, aj keď sme prekročili časovú kapacitu členov. V ďalších šprintoch chceme viac prihliadať na reálnu časovú kapacitu členov a prispôbovať tomu naplánovanú prácu.

Vďaka retrospektíve po 2.šprinte sa nám podarilo vylepšiť niektoré procesy. Pracovali sme pravidelnejšie, robili sme si niekoľkokrát do týždňa stand-upy. Dali sme si väčší pozor na zaznamenávanie práce v Azure DevOps, vďaka čomu sme získali väčší prehľad o práci v grafoch.

Vybrané user stories

V Tabuľke 8.5 môžeme vidieť vybrané user stories pre 3.šprint. Nepodarilo sa vykonať dve úlohy z dvoch user stories, preto sa presúvajú do ďalšieho šprintu. Tieto úlohy sa týkajú vývoja a konkrétne nastavenia CI a vytvorenia automatického deploymentu aplikácie.

Názov	SP	Schválená
Ako product owner chcem vyhľadávanie údajov z FinStatu na identifikáciu používateľa	8	Áno
Ako vývojár potrebujem poslať údaje na backend na ich ďalšie spracovanie	21	Áno
Ako používateľ chcem aplikáciu bez bugov na predchádzanie problémov, ktoré by mohli spôsobiť	8	Nie
Ako product owner požadujem vytvoriť dokument projektového riadenia	21	Áno
Ako product owner požadujem vytvoriť dokument inžinierskeho diela	34	Áno
Ako product owner chcem mať aktuálne dokumenty na webovej stránke	8	Áno
Ako product owner požadujem spustenie aplikácie pomocou linku	8	Áno
Ako product owner chcem mať vždy k dispozícii aktuálnu verziu aplikácie	3	Nie
Ako tím líder chcem, aby všetci členovia tímu boli oboznámení s metodikami	1	Áno
Ako product owner chcem mať plán na ďalší šprint	13	Áno
Ako product owner chcem vidieť stav projektu	5	Áno

Tabuľka 8.5: Vybrané user stories šprintu č.3

Práca na šprinte

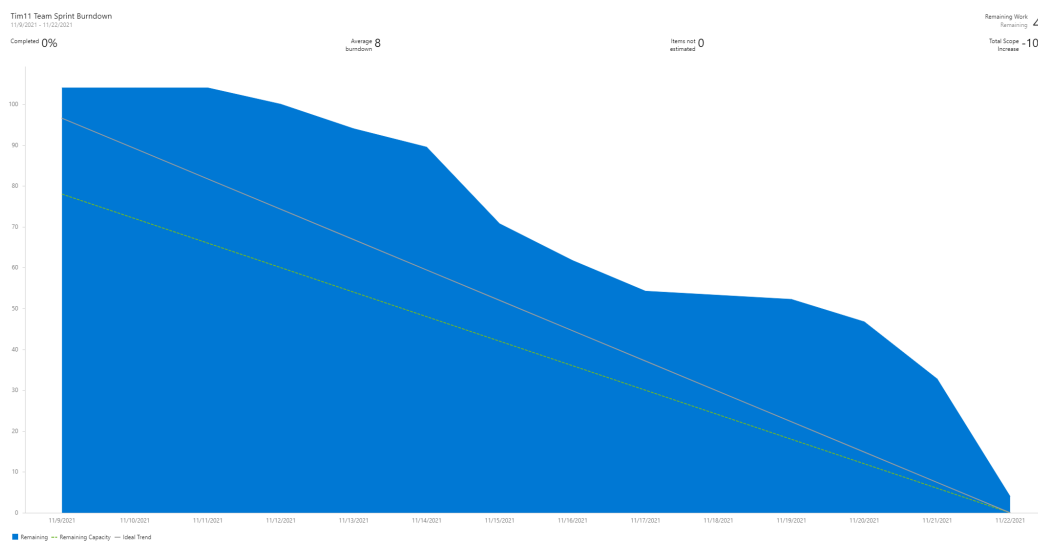
V Tabuľke 8.6 si môžeme všimnúť, ako pracovali jednotliví členovia tímu na šprinte. Každý člen dostal určitý počet úloh a tým prispel svojim podielom k celkovej práci. Členovia s nižším počtom úloh mali väčšinou na starosti úlohy, ktoré si vyžadovali viac času.

Meno a priezvisko	Počet úloh	Podiel práce (%)
Filip Agh	10	16.6%
Adam Benovič	6	16.6%
Ľuboš Daniš	11	16.6%
Lenka Ivanová	7	16.6%
Aranka Klukaová	5	16.6%
Patrik Vdoviak	5	16.6%

Tabuľka 8.6: Práca na šprinte č.3

Burndown chart

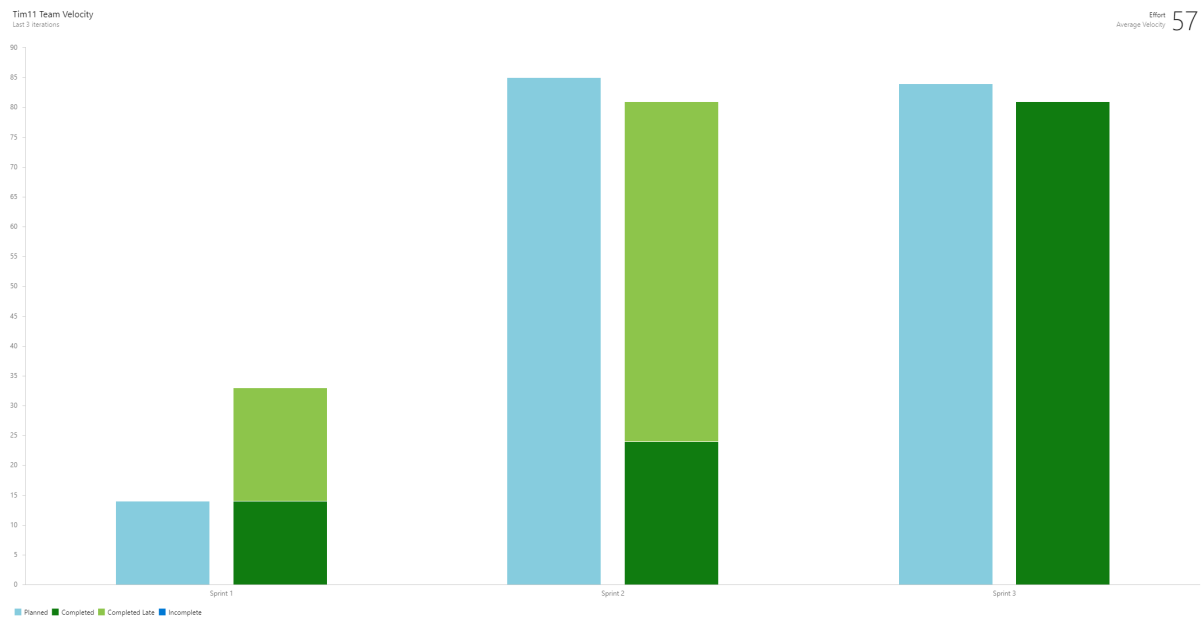
Na Obr.8.3 môžeme vidieť Burndown chart po 3.šprinte. Vidíme, že prácu sa podarilo vykonávať pravidelnejšie, aj vďaka zavedenému piatkovému stand-upu. V šprinte bolo naplánovaných veľa úloh, ktorých odhadovaný čas vykonania prevyšoval časovú kapacitu členov tímu. Nepodarilo sa nám vykonať 2 úlohy a to hlavne z dôvodu ochorenia jedného z členov.



Obr. 8.3: Burndown chart pre 3.šprint

Velocity report

Na Obr.8.4 môžeme vidieť velocity po 3.šprinte. V tomto šprinte sa podarilo vykonať takmer všetky naplánované úlohy. Napriek tomu sme zhodnotili, že v budúcich šprintoch budeme viac prihliadať na reálnu časovú kapacitu. Vďaka retrospektíve po 2.šprinte sme si dali tiež väčší pozor na uzavretie user stories po ich kompletnom vypracovaní, čo sa odráža na grafe.



Obr. 8.4: Velocity report po 3.šprinte

Export úloh

Export úloh po tretom šprinte je možné nájsť na webovom sídle tímu 11.

9 Globálna retrospektíva

Počas prvých troch šprintov sme zaznamenali zlepšovanie v rámci riadenia projektu, ktoré nám pomohlo pri ďalšej práci.

V prvom šprinte sme mali ešte problém s návalovou prácou a na konci s nepravidelným aktualizovaním práce na projekte. Problémy boli tiež spôsobené neskorším stretnutím s vedúcim tímu a začiatkom šprintu v čase, keď sme ešte nemali ujasnené požiadavky a princípy scrumu. Za pozitívne považujeme otvorenú komunikáciu v tíme, vďaka ktorej nebol problém si nedostatky vydiskutovať. Dohodli sme sa na dodatočnom piatkovom stand-upe, aby sme mali odsledovanú prácu aj počas týždňa mimo oficiálnych stretnutí. Okrem toho sme sa dohodli na pravidelnom aktualizovaní Azure DevOps a na tom, že si dáme vedieť, ak niekoho práca bude blokována prácou niekoho iného. Všetky tieto nedostatky sa nám podarilo odstrániť v druhom šprinte.

V druhom šprinte sme pokračovali vo veciach, ktoré sme identifikovali ako správne. Zaznamenali sme tiež problém v definícii user stories, ktoré boli príliš všeobecné a obsahovali úlohy, ktoré boli naplánované do viacerých šprintov. Tento problém sme sa rozhodli riešiť, ale kvôli už naplánovanej práci v treťom šprinte to bude možné aplikovať až od štvrtého šprintu. Okrem toho sme si zaumienili robiť exporty úloh každý týždeň a kontrolovať stav user stories priradeným zodpovedným, aby sa nestávalo, že user story bola dokončená, ale nebola tak označená. Tiež sme sa dohodli, aby nám vedúceho tímu dával väčšiu spätnú väzbu k users stories.

V treťom šprinte sme aplikovali návrhy pre zlepšenie po retrospektíve druhého šprintu. Výsledkom bola lepšia naplánovaná a odsledovaná práca. V grafoch sa už začal zobrazovať reálny stav práce na projekte. Po tomto šprinte sme si navyše zaumienili neplánovať viac, ako nám dovoľuje časová kapacita, aby sme predišli nedokončeniu úloh.

9.1 Retrospektíva šprintu č.1

Retrospektívy sa zúčastnili všetci členovia tímu(Tabuľka 9.1).

Meno a priezvisko	Zúčastnil/a sa
Filip Agh	áno
Adam Benovič	áno
Ľuboš Daniš	áno
Lenka Ivanová	áno
Aranka Klukaová	áno
Patrik Vdoviak	áno

Tabuľka 9.1: Účasť na retrospektíve šprintu č.1

S čím chceme pokračovať?

- otvorená komunikácia. Počas prvého šprintu bola komunikácia na veľmi dobrej úrovni. Každý člen prejavoval záujem, pýtal sa a diskutoval. Ku konfliktom neprichádzalo, odlišné názory sme si vedeli rozumne vysvetliť a došli sme nakoniec k spoločným záverom.
- prieskumná analýza. Veľmi sa nám osvedčila prieskumná analýza dostupných riešení, vďaka čomu sme sa vedeli vyhnúť plánovaniu vecí, ktoré reálne nepotrebujeme.

S čím potrebujeme prestať?

- nerobiť prácu návalovo a na konci. Radi by sme sa v budúcnosti vyhli robeniu práce návalovo a na konci. V prvom šprinte mal na tom veľký podiel aj posunutý začiatok.
- aktualizovanie práce až na stretnutí, aj keď práca bola hotová skôr. Všimli sme si, že aj keď bola práca hotová, nebola aktualizovaná v Azure DevOps. Tomuto by sme sa chceli v budúcich šprintoch vyhnúť.

S čím chceme začať?

- zaviesť stand-up každý piatok, aby sme mali prehľad, ako na tom sme
- pravidelnejšie aktualizovať prácu na šprinte (aj vďaka zavedeniu stand-upu)
- dať si vedieť, ak moja práca závisí od niekoho iného, neostať ticho
- lepšie popisovať user stories
- pridávať opisy k úlohám, aby bolo hneď jasné, čo sa očakáva

9.2 Retrospektíva šprintu č.2

Retrospektívy sa zúčastnili všetci členovia tímu (Tabuľka 9.2).

Meno a priezvisko	Zúčastnil/a sa
Filip Agh	áno
Adam Benovič	áno
Ľuboš Daniš	áno
Lenka Ivanová	áno
Aranka Klukaová	áno
Patrik Vdoviak	áno

Tabuľka 9.2: Účasť na retrospektíve šprintu č.2

S čím chceme pokračovať?

- stand-upy mimo oficiálnych stretnutí
- pravidelnejšie aktualizovanie práce
- pravidelná a otvorená komunikácia
- definovanie user stories v správnom formáte

S čím potrebujeme prestať?

- vytváranie veľmi všeobecných user stories. Dôvod je ten, že aktuálne majú user stories úlohy, ktoré sú rozplánované do viacerých šprintov.

S čím chceme začať?

- zapisovať nacenenie user stories story pointami do Azure DevOps. Dôvod je ten, aby sme to mali zaznamenané aj v grafe.
- robiť exporty úloh každý týždeň a nie iba po šprinte
- zapojiť viac vedúceho tímu do vytvárania a schvaľovania user stories. Potrebujeme ku každej osobitne spätnú väzbu. Doposiaľ to bolo vyjadrenie ako k celku.

- kontrola stavu user story priradeným zodpovedným. Stáva sa, že user stories nie sú označené ako done, aj keď sú hotové. Pôsobí to, že sa nespravili a neskoršia zmena stavu kazí grafy.
- skúsiť vytvárať user stories tak, aby sa dali vykonať v jednom šprinte. Tento bod nebude možné vykonať hneď v 3.šprinte, keďže už boli naplánované úlohy v rôznych šprintoch a pod jednou user story. Od 4. šprintu sa pokúsime vytvárať jednoznačné user stories vykonateľné v danom šprinte.

9.3 Retrospektíva šprintu č.3

Retrospektívy sa zúčastnili všetci členovia tímu (Tabuľka 9.3).

Meno a priezvisko	Zúčastnil/a sa
Filip Agh	áno
Adam Benovič	áno
Ľuboš Daniš	áno
Lenka Ivanová	áno
Aranka Klukaová	áno
Patrik Vdoviak	áno

Tabuľka 9.3: Účasť na retrospektíve šprintu č.3

S čím chceme pokračovať?

- pravidelná, otvorená a efektívna komunikácia.
- stand-upy mimo oficiálnych stretnutí
- kontrola stavu user story priradeným zodpovedným

S čím potrebujeme prestať?

- vytváranie veľmi všeobecných user stories, ktorých tasky pokrývajú viacero šprintov. Tento problém sme identifikovali už v predošlom šprinte, ale kvôli už naplánovanej práci nebolo možné to okamžite napraviť
- plánovanie viac ako máme časovú kapacitu

S čím chceme začať?

- pridávanie popisov user stories
- dávať si viac pozor na to, že v piatok je vhodné sa vyjadriť k progresu
- pravidelne zmenšovať zostávajúci počet hodín na úlohe, aby sa prejavoval progres. Tento problém sme identifikovali pri úlohách, ktoré trvajú počas celého šprintu. Chceme tým predísť dokončeniu úloh návalovo naraz na konci.
- po plánovaní skontrolovať, či sú správne priradení zodpovední pri jednotlivých user stories

Príloha A: Motivačný dokument

Tím

Naše pokročilé skúsenosti:

- Vývoj webových stránok(Front-end a Back-end) - PHP, Java, JS, TS - Angular, React, Vue
- Databázy - SQL(PostgreSQL, MySQL, MS SQL), Redis
- Vývoj na mikročipoch - C, C++, micropython - ESP32, Arduino
- Testovanie softvéru - manuálne a aj automatizované
- Verziovanie pomocou GIT
- Linux - bash a podobne
- Odata/ REST API
- Návrh UI a UX
- Biznis analýza
- Elasticsearch
- Apache
- Docker
- Github CI
- Dátová analýza v Python
- Strojové učenie

Základné skúsenosti:

- Vývoj v Unity pre AR/VR v C#
- OpenGL - C++

Chceme sa zlepšiť:

- Vývoj mobilných aplikácií (Android)
- Vývoj webových aplikácií - Full-stack
- Umelá inteligencia
- Strojové učenie

Naše silné stránky:

- Každý člen tímu má znalosti a skúsenosti s inými technológiami, vďaka čomu vieme zvládnuť problémy rôznych typov
- Komunikatívnosť
- Zodpovednosť

Naše predmety relevantné k téme:

- Aplikačné architektúry softvérových systémov
- Architektúra informačných systémov
- Architektúra softvéru
- Kvalita programových a informačných systémov
- Neurónové siete
- Pokročilé databázové technológie
- Objavovanie znalostí
- Vyhľadávanie informácií
- Testovanie softvéru
- Právo pre informatikov

Motivácia

3. DataHub pre rôzne typy zariadení a ich spracovanie/ analýzu/ vizualizáciu (1)

Túto tému sme si zvolili ako hlavnú tému, ktorú by sme ako tím chceli riešiť. Každý z nášho tímu si našiel časť v tomto zadaní, s ktorou má skúsenosti, vie pomôcť ďalším členom tímu, ako aj inú časť, v ktorej sa naopak chce zlepšiť.

Hlavný dôraz pri riešení tejto témy je z našej strany kladený na vytvorenie open source cloud friendly systému, ktorý by pomocou spomínaných systémov agregoval, distribuoval a vhodne reprezentoval získané údaje.

Taktiež treba pripomenúť, že tento projekt nemá skončiť zabudnutý a nepoužitý, čo nás ale teší, pretože máme možnosť spraviť systém, ktorý by bol užitočný pre komunitu, ktorá by ho mohla používať.

Náš tím by sa zamerlal na nasledovné body:

- Návrh robustnej architektúry na asynchrónne spracovanie za použitia serverless funkcií a vhodné distribuovanie eventov naprieč systémom
- Agregáčnne systémy, ktoré by agregovali dáta a sprístupnili požadované údaje pre verejné / privátne aplikačné rozhranie
- Front-end aplikácia na vizualizáciu získaných údajov podľa požadovaných kritérií

Vedľajšie ciele projektu:

- Vytvoriť vlastnú meteostanicu, integrovať ju so systémom
- Skúsiť si strojové učenie na agregovaných údajoch systému

Náš tím má nasledovné skúsenosti mimo školských projektov:

- nasadenie a udržiavanie systému v linux prostredí (docker)
- vývoj backend architektúry
- skúsenosti so všetkými typmi testovania softvéru
- nízkoúrovňové programovanie mikroprocesorov a senzorov
- analýza, spracovanie a vizualizácia údajov

Náš tím má len základné skúsenosti s vývojom frontend webových aplikácií, ale je naším cieľom tieto skúsenosti získať, ako aj prehĺbiť a rozšíriť iné skúsenosti s modernými technológiami naprieč naším tímom.

19. Automatizácia procesov KYC a AML (2)

Táto téma nás zaujala svojím použitím technológií a zameraním na aktuálne riešenie tému. Napriek tomu, že sme sa s touto témou ešte v praxi priamo nestretli, máme veľkú dávku motivácie, pretože sa chceme v daných technológiách naučiť pracovať.

Vidíme veľký potenciál v použití open-source riešení pre dosiahnutie komplexnej aplikácie. Väčšina z nás má skúsenosti s integráciou open-source knižníc, ako aj s integráciou a vývojom API.

Členovia nášho tímu majú skúsenosti relevantné k téme:

- vývoj Robotic Process Automation
- vývoj OCR pre mobilné aplikácie
- práca na nízkoúrovňových aplikáciách, ako napríklad logika generovania QR
- kódu pre bankové platby, podľa štandardu PayBySquare
- testovanie AML v platobnej inštitúcii Trustpay
- skúsenosť s automatizovanými testami
- manuálne testovanie Aplikácie pre eID
- UX
- skúsenosti s dátovou analýzou a strojovým učením

16. FIIT WIX (3)

Téma nás získala svojim zámerom umožnenia vytvorenia prehľadných a user-friendly stránok predmetov.

Okrem toho vývoj webových stránok je niečo, čomu sa venuje alebo o čo sa zaujíma a v čom sa chce zlepšiť každý člen nášho tímu. Ako bonus, keď to spojíme s našimi vedomosťami o UX, tak by sme vedeli vytvoriť aplikáciu, ktorú by používatelia používali s veľkou radosťou.

Na základe našich skúsenosti by sme zaručili, aby počas vývoja v centre pozornosti stáli používatelia.

Členovia nášho tímu majú skúsenosti relevantné k téme:

- SQL
- vývoj webových stránok v rôznych technológiách
- skúsenosti s UI/UX

Príloha A - Zoradenie všetkých tém podľa priority

Naša priorita	Číslo témy	Názov témy
1.	3.	DataHub pre rôzne typy zariadení, ich spracovanie / analýzu / vizualizáciu
2.	19.	Automatizácia procesov KYC (Know your client) a AML (Anti-money laundering)
3.	16.	FIIT WIX
4.	17.	Document Wizard
5.	13.	Navigácia v smartfóne pomocou rozšírenej reality
6.	8.	Educational Content Engineering Hub - Databáza otázok, odpovedí, úloh a riešení [ECEH-DU]
7.	20.	Aplikačné riešenie pre elektronický volebný systém
8.	15.	Ion Mobility Spectrometry for Rapid HEMP Potency Testing
9.	2.	Transformácia priestorov na bezpečné a inteligentné miesta na prácu [space2]
10.	12.	Spektrometrické rozpoznávanie túh do pera
11.	14.	IoT platforma na priemyselnú automatizáciu - malý pivovar
12.	5.	Vytvorenie inteligentného model-based agenta (umelá inteligencia na báze Knowledge grafov) pre tvorbu komplexných dátových štruktúr a vzťahov pre aplikovaný výskum v klinickej onkológii
13.	11.	(Q)SAR analýza fototoxických látok
14.	4.	Adverse Media Screening
15.	18.	Webové IDE pre ASIC [ASICDE]
16.	7.	Vizualizácia softvéru vo virtuálnej a rozšírenej realite [VizReal]
17.	9.	Monitorovanie a správa systému pre výrobný areál [LO-MON]
18.	10.	VANET siete
19.	6.	Platforma pre sledovanie dodávateľského reťazca s využitím technológie blockchain [S-Chain]
20.	1.	Nadstavby pre blockchain platobné brány [BlockBuild]

Príloha B - Rozvrh tímu

		8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00
Pondelok	Filip			VINF		Priestor pre TP						AS	
	Adam	AS		AS								AS	
	Aranka												
	Lenka			VINF						RREP		RREP	
	Lubos	AS		AS								AS	
	Patrik	VINF								AS			
Utorok	Filip			Priestor pre TP				AS	VISS	TP1			
	Adam												
	Aranka												
	Lenka												
	Lubos												
	Patrik												
Streda	Filip											MTS(1.28)	
	Adam	PDT										MTS(1.28)	
	Aranka	KPAIS				KPAIS					MTS (1.29)		
	Lenka			VINF								MTS(1.29)	
	Lubos												
	Patrik			VINF		ZKGRA							MTS(1.28)
Stvrtok	Filip					Priestor pre TP							
	Adam									AS (1.28)			
	Aranka									AS (1.28)			
	Lenka												
	Lubos	ZKGRA		ZKGRA									
	Patrik	ZKGRA											
Piatok	Filip		PDT				PDT						
	Adam		PDT										
	Aranka		PDT		PDT								
	Lenka												
	Lubos												
	Patrik												

(Každý vyznačený časový blok na riešenie TP ponúkame ako možný čas na stretnutia s vedúcim a na spoločnú prácu tímu bude slúžiť jeden zo zvyšných časových blokov.)

Príloha B: Metodika Definition of Done

Počas vývoja projektu je potrebné si zdefinovať rôzne pravidlá pre každý typ úlohy, kedy je dokončená. Takéto pravidlá sa nazývajú Definition of Done. V našom tímovom projekte rozlišujeme pravidlá Definition of Done zoradené od najmenej časti pre:

- Task
- User story
- Feature
- Epic
- Projekt

Definition of Done pre Task

Task je najmenšia časť Projektu a podčasť User story. Task môže obsahovať rôzny charakter úlohy a tie rozdeľujeme do nasledovných skupín:

1. Implementačná úloha
2. Review
3. Testovanie
4. Tvorba dokumentácie
5. Analytická úloha

Definition of Done pre Implementačnú úlohu

Pre implementačnú úlohu sme si definovali Definition of Done ako implementovanie požadovanej funkcionality popísanej v Tasku a sú zapracované pripomienky z review. Jej následné otestovanie a možné zdokumentovanie, ktoré je ale potrebné iba pri úlohách trvajúcich viac ako 2 hodiny.

Definition of Done pre Review

Tento typ Tasku je hotový vtedy, keď daný dokument v Tasku je overený a je k nemu popísaný feedback na úpravu alebo ukončený úspešne bez feedbacku.

Definition of Done pre Testovanie

Testovanie je úspešne ukončené práve vtedy, keď implementovaná úloha spĺňa všetky kritériá, ktoré sa od tejto implementácie očakávajú.

Definition of Done pre Tvorba dokumentácie

Pri vytváraní dokumentácie za jej úspešné ukončenie považujeme moment, kedy prejde Review iným členom tímu a tento člen k nej nemá žiadne pripomienky, výhrady alebo možnosti doplnenia, prípadne boli pripomienky zapracované. Pri dokumentácií ešte rozlišujeme ďalšie typy a tými sú:

- Aktualizácia webu - web je aktuálny pre daný týždeň
- Export úloh - úlohy sú exportované a dostupné na webe pre daný šprint
- Retrospektíva šprintu - retrospektíva je spísaná a dostupná na webe pre daný šprint

Definition of Done pre Analytickú úlohu

Pri analytickej úlohe definujeme čas ukončenia ako moment, kedy člen tímu, ktorý má túto úlohu priradenú, nadobudol dostatočné vedomosti o popísanom probléme úlohy natoľko, že dokáže bez ďalšej analýzy pokračovať vo výkone ďalšej nadväzujúcej práce.

Definition of Done pre User story

Úspešné ukončenie User story je naviazané na úspešné ukončenie všetkých podliehajúcich Taskov priradených tejto User story, otestovanie funkcionality User story a schválenie funkcionality celým tímom, ale to len v prípade, ak ide o funkcionality, ktorá sa skladá z minimálne 3 Taskov.

Definition of Done pre Feature

Definition of Done pre Feature definujeme ako úspešné ukončenie všetkých User stories podliehajúcich tejto Feature. Úspešnému ukončeniu Feature tiež podlieha schválenie tímom a zdokumentovanie Feature do výsledného dokumentu tímu.

Definition of Done pre Epic

Epic je hotový práve vtedy, keď všetky Feature podliehajúce tomuto Epiku sú úspešne ukončené. Úspešné ukončenie Epiku musí schváliť tím a taktiež vedúci tímu.

Definition of Done pre Projekt

Projekt je dokončený práve vtedy, keď všetky Epiky sú úspešne ukončené. Toto ukončenie podlieha schváleniu vedúceho tímového projektu.

Príloha C: Metodika komunikácie

Obsahom tejto metodiky je definovanie si komunikačných kanálov tohto projektu a uvedenie typov komunikácií, ktoré prebiehajú na ktorej platforme. Zoznam platforiem zoradený od najdôležitejšej, na ktorých prebieha komunikácia:

1. Azure DevOps
2. Discord
3. Teams
4. Messenger
5. Email

Azure DevOps

Cez tento kanál sa rieši komunikácia členov na úlohách. Členovia môžu ku každému prvku na tejto platforme pridať komentár a následne diskutovať s ostatnými členmi tímu.

Discord

Túto platformu používame prioritne na hlasovú a obrazovú komunikáciu medzi členmi tímu, počas stretnutí ak niektorý z členov tímu sa nemôže fyzicky zúčastniť stretnutia. Vždy jeden z členov tímu, väčšinou team leader, zdieľa obrazovku a má zapnutý zvuk. Taktiež sa na tejto platforme zdieľajú potrebné dokumenty, ktoré je v momente treba použiť.

Teams

Teams sa používa ako komunikačné médium pre spojenie sa s vedúcim tímu. Na tomto médiu sa rozoberajú dôležité veci ohľadom smerovania projektu, ktoré je potrebné prebrať medzi fyzickými stretnutiami s vedúcim tímu.

Messenger

Messenger sa používa ako súkromná komunikácia členov tímu. V tomto komunikačnom kanály sa riešia veci ohľadom stretnutia sa, upozornenia členov tímu a ostatné.

Email

Komunikácia pomocou Email-u sa využíva výlučne na preposielanie pozvánok. Na email taktiež chodia emaily z tímového aliasu, ktorý príde všetkým členom tímu.

Príloha D: Metodika práce s úlohami

Pre správu úloh sme si zvolili Azure DevOps. Pracovné položky sú v Azure DevOps zastrešené pod označením Work items. Systém je v angličtine, preto názvy položiek a stavy úloh budú v tomto dokumente označené taktiež anglickým jazykom.

Work items

V Azure DevOps rozlišujeme tieto položky:

- Epic
- Feature
- Product Backlog Item (User Story)
- Task

Predchádzajúce poradie prirodzene znázorňuje hierarchiu položiek. Kompletný projekt obsahuje niekoľko Epík, kde rozlišujeme najväčšie logické časti, na ktorých v projekte pracujeme. Pod Epic máme Features pre ďalšie rozdelenie logických častí v Epic. Pod Features vytvárame Product Backlog Items alebo User Stories, ktoré si vyberáme neskôr do šprintov. Každá User Story má pod sebou ďalšie Tasky. Task tvorí najmenšiu a najpodrobnejšiu položku v hierarchii.

Epic:

V projekte sme vytvorili tieto epiky:

- **Oboznámenie sa s technológiami** - združuje úlohy súvisiace so študovaním dokumentácií a best practices rôznych technológií.
- **Vytvorenie dokumentácie projektu** - združuje úlohy súvisiace s administratívou a dokumentáciou projektu, ako napríklad vytváranie zápisníc, retrospektív šprintov, metodík a ďalších dokumentov podľa požiadaviek predmetu.
- **Vytvorenie webovej stránky tímu** - združuje úlohy súvisiace s webovou stránkou tímu, ako napríklad vytvorenie návrhu stránky, vytvorenie webového sídla a pravidelné aktualizovanie obsahu.

- **Vývoj prototypu aplikácie na identifikáciu používateľa** - združuje úlohy súvisiace priamo s vývojom aplikácie.
- **TPcup** - združuje úlohy súvisiace s TP Cupom, ako napríklad vytvorenie prihlášky do súťaže.

Feature

Položky Features používame na logické rozdelenie častí projektu. Aktuálne Features sú napríklad Zimný semester a Letný semester v Epic Vytvorenie dokumentácie projektu.

User Story (Product Backlog Item)

User story popisuje funkcionality, ktorá tvorí hodnotu systému pre budúceho používateľa. User story musí spĺňať stanovený formát opísaný nižšie. Každá User story je ohodnotená Story pointami a má priradeného zodpovedného.

Formát:

Ako <typ používateľa> chcem <nejaký cieľ> <nejaký dôvod>.

Príklad: Ako používateľ chcem, aby boli moje údaje získané z môjho dokladu totožnosti na identifikáciu mojej osoby.

Task

Task alebo Úloha predstavuje v hierarchii najmenšiu položku. Úlohy sú organizované pod User stories. Každá úloha musí obsahovať výstižný názov, ktorý opisuje podstatu danej úlohy. Zároveň, vďaka hierarchii, do ktorej je úloha zaradená je cieľ úlohy väčšinou jasný. V prípade, že opis v hierarchii nie je postačujúci, je žiadúce doplniť krátky opis. Okrem názvu a opisu musí mať každá úloha priradeného riešiteľa a taktiež vyplnené polia Remaining work a Effort.

Práca s úlohami

Tvorba a zaradenie úloh

Úlohy vytvára tím líder a scrum master Ľuboš, po dohode a odsúhlasení členmi tímu. Úlohy sa vytvárajú na tímových stretnutiach. Ak nevieme novú úlohu zatiaľ zaradiť, necháme ju v Product backlogu ako Unparented, inak sa zaraďujú pod User Stories a do šprintov. Do prebiehajúceho šprintu sa neodporúča pridávať nové úlohy.

Stavy úloh

Úloha môže nadobúdať tieto stavy:

- **To Do** - úlohu je potrebné urobiť.
- **In Progress** - na úlohe sa pracuje.
- **Review** - úloha čaká na skontrolovanie alebo je kontrolovaná.
- **To Be Tested** - úloha čaká na pretestovanie alebo je testovaná.
- **Done** - úloha je dokončená, spĺňa Definiton of Done zadané v Metodike Definition of Done.

Počiatkový stav úlohy je To Do, následne prechádza do In Progress. Potom môže úloha prejsť do stavu Review alebo To Be Tested podľa potreby. Podľa Metodky Definiton of Done môže nakoniec prejsť do stavu Done.

Príloha E: Metodika testovania

Testovanie tvorí jednu z najvýznamnejších častí vývoja softvéru. Slúži na overenie správnosti očakávaných funkcionalít a takisto slúži na uistenie sa, že projekt spĺňa očakávania klienta, pričom ponúka priestor na opravu nájdených chýb a nedokonalostí riešenia.

Testovanie tried a metód

Berúc do úvahy povahu nášho projektu, testovanie každej triedy a metódy jednotkovými testami nebude nutné. Kvôli tomu bude správnosť fungovania zabezpečená hlavne pomocou manuálneho testovania a dôkladného ošetrovania všetkých možných spôsobov vykonania metód.

Jednotkové testy

Pre zložitejšie časti kódu budú jednotkové testy povinné, aby sme mali čo najväčšiu istotu, že aj zložité časti fungujú správne.

Na vytvorenie a vykonávanie týchto testov bude použitý test package Flutteru, kvôli ktorému budeme počas vytvárania testov dodržiavať nasledujúce pravidlá :

- v priečinku zdrojových kódov bude podpriečink, ktorý bude obsahovať všetky testy, a ktorý bude zároveň napodobňovať hierarchiu priečinkov s triedami,
- súbory s testami budú pomenované na základe testovanej triedy, ale na označenie, že ide o súbor s testami, budú mať „test” sufix a ich názvy sa budú písať snake_case-om,
- názvy metód na testovanie budú začínať slovom „test” a budú písané snake_case-om.

Akceptačné testy

Nevyhnutným druhom testov sú akceptačné testy, pomocou ktorých overíme, či vyvíjaný softvér splní všetky akceptačné kritériá, čiže jeho prijateľnosť zo strany klienta.

Akceptačné testy budú vytvorené na začiatku riešenia projektu na základe konzultácií s vedúcim tímu (product ownerom). Pre „akceptovanie” vyvíjaného softvéru musia prejsť všetky akceptačné testy.

Príloha F: Metodika verziovania

1 Úvod

Tento dokument definuje pravidlá a normy, ktoré určujú prácu tímu 11 z roku 2021 na predmete „Tímový projekt“ s repozitármi.

Zadefinujeme si dva hlavné typy repozitárov a to:

- repozitár web stránky projektu
- repozitáre implementácie nášho projektu

Zadefinujeme si roly účastníkov počas interakcie s repozitárom, ako aj formát textu pre commit a pull request.

V poslednej časti si predstavíme odporúčania a návody na konkrétne časti práce s repozitárom.

2 Repozitár

V tejto časti si opíšeme teoretické a praktické informácie o repozitároch.

2.1 Technické riešenie repozitárov

- Version control systém pre náš projekt: GIT
- Manažér pre version control system: Azure DevOps

2.2 Typ repozitáru

Repozitár pre web stránku projektu, s názvom: Tim11

Tento repozitár bude mať výnimku z nutnosti organizácie vetvenia repozitára, keďže do daného repozitára budú pribúdať dokumenty a linky a teda žiadne veľké zmeny. Za repozitár je zodpovedný (git master) Patrik Vdoviak.

Repozitáre, v ktorých sú implementácie nášho riešenia

Názov repozitára má opisovať projekt, ktorý zastrešuje. Pre repozitár platia všetky princípy a pravidlá, ktoré si opíšeme v ďalšej časti dokumentu.

3 Vetvenie

Git funguje na základe vetvenia, a teda si definujeme nasledujúce vetvy.

3.1 master

Vetva master reprezentuje produkčnú vetvu a obsahuje commity s tagom pre jednotlivé publikované verzie. Do mastera sa pushuje len hotfix tasky alebo pri vydaní novej major/minor verzie sa master posunie na aktuálny dev, patch verzie, čiže hotfixy sa aplikujú následne aj na dev vetvu pomocou rebase nad masterom.

3.2 dev

Vetva dev je aktuálna dev verzia repozitára, do ktorého sa pushujú nové featury, testy, alebo sa robí refaktor.

3.3 task

Keď začne developer riešiť task, musí si spraviť vlastnú vetvu. V prípade hotfixu je base master, inak dev vetva. Názov vetvy musí byť podľa formátu: číslo tasku[_názov tasku]()

4 Roly

V tejto časti si opíšeme, aké roly existujú v procese práce s repozitárom.

4.1 Git master

Git master je jedna osoba, ktorá je zodpovedná za master a dev vetvu repozitára, a teda merguje do týchto vetiev, je zodpovedná za riešenie konfliktov medzi vetvou master a dev, dohliada na dodržiavanie metodiky vo svojom repozitári.

4.2 Autor pull requestu

Autor pull requestu je člen tímu, ktorý dokončil svoj task a chce ho mergnúť do príslušnej hlavnej vetvy.

4.3 Reviewer

Je osoba, ktorá má skontrolovať zmyslupnosť a správnosť zmien v commitoch. Reviewer je iná osoba ako autor pull requestu.

5 Udalosti v procesoch verziovania

5.1 Vydanie verzie

Vydanie verzie je proces, kedy git master určí konkrétny commit v masteri a pridá mu tag s verziou {major}.{minor}.{patch}, do prvého oficiálneho vydania je major verzia "0". Major verzia sa mení, ak sa posúva master na dev a nastávajú zmeny, ktoré nie sú kompatibilné s predošlou minor verziou. Ak zmeny sú kompatibilné, tak sa mení minor verzia. Patch sa mení v prípade pridania nových hotfixov.

5.2 Commit

Developer, ktorý pracuje na tasku, ktorý má zmeny v repozitári, pravidelne a granularne commituje do svojej task vetvy.

Formát commitu:

```
Task #{id tasku} {feature, fix, test, refactor, format}:{description}
```

[krátky opis zmien]

(max 72 znakov na 1. riadok)

5.3 Pull request

Developer, ktorý dokončil a úspešne otestoval prácu na tasku, ktorý má zmeny v repozitári, si vytvorí pull request na svoju base vetvu. Ak sa vyskytnú konflikty, musí ich vyriešiť, prípadne sa poradí s git masterom daného repozitára. Po vytvorení pull requestu sa uistí, že je zvolený aspoň 1 reviewer. V prípade vyžiadaných zmien autor pull requestu musí urobiť úpravy a znova vyžiadať review.

Formát pull requestu:

```
#{id tasku}
```

[krátky opis zmien v pull requeste]

5.4 Merge pull requestu

Git master má za úlohu finálne zvalidovať každý pull request a keď je schválený osobou, ktorá robila review, tak ak nie sú konflikty, mergne pull request.

6 Pravidlá verziovania

- mergovanie je povolené len pre pull requesty do master/dev, inak sa použije rebase
- v prípade chyby alebo nutnosti zmeny commitu sa nerobí revert commit, ale posunie sa head na posledný validný commit a vytvoria sa lokálne nové commity, ktoré prepíšu chybnú pomocou force push.
- ak niekto urobí force push, upovedomí o tom všetkých developerov
- do repozitára sa neukladajú secrety, ak sa tak stane, developer musí secret force pushom odstrániť

7 Odporúčania

Na prácu s gitom odporúčam git gui rozhranie, hlavne pre git masterov, konkrétne pracovný nástroj Idea a plugin pre integráciu Azure DevOps

V prípade otázok, nejasností alebo problému kontaktujte git guru a autora metodiky verziovania:

Filip Agh

xaghf@stuba.sk

Príloha G: Metodika vývoja aplikácie

Na zjednotenie formy písania kódu počas vývoja požadovaných funkcionalít aplikácie potrebujeme stanoviť pravidlá pre písanie kódu v jazyku Python a v SDK Flutter.

Pre písanie kódu a vytváranie súborov a priečinkov bude využitý anglický jazyk a vývojové prostredie Android Studio.

Spolupráca časti aplikácie písanej v Pythone s časťou vyvinutou pomocou Flutter-u bude zabezpečená pomocou HTTP komunikácie(API).

Kvôli snahe o zjednodušenie orientovania sa v rámci zdrojového kódu bude kód rozdelený do viacerých repozitárov:

- Flutter App repozitár,
- Deep face repozitár.

Flutter

Flutter je open-source používateľské rozhranie SDK na vývoj (multiplatformového) softvéru.. V rámci aplikácie bude zabezpečovať chod mobilnej aplikácie.

Zásady písania kódu:

- súbory s obrazkami pomenúvame snake_case(face_capture_select),
- priečinky a ostatné súbory sa pomenúvajú v camelCase(form, idCardForm),
- zlučujeme súbory do priečinkov podľa potreby a významu, napr. formuláre v priečinku form,
- všetko ostatné sa drží konvencie popísanej v sprievodcovi štýlu.

Python

Časť aplikácie zabezpečujúca identifikáciu používateľa bude implementovaná pomocou jazyka Python a knižnice DeepFace. V rámci písania kódu v jazyku Python bude využitá konvencia “PEP 8 - Style Guide for Python Code”. IDE nastavenia sa nelíšia od základného nastavenia JetBrains PyCharm.

Príloha H: Webové sídlo projektu

Naše webové sídlo sa nachádza na odkaze <http://team11-21.studenti.fiit.stuba.sk/>.

Sídlo je pravidelne aktualizované. V momente kliknutia by sme mali vidieť najaktuálnejšiu stránku. Stránka pozostáva z hlavičky s tímovým logom, pätičky s e-mailom na náš tím a z ďalších piatich sekcií:

- **KLYCK**

Táto je len všeobecným názvom projektu spoločne s logom.

- **O projekte**

Obsahom tejto sekcie sú stručné informácie ohľadom toho, čo v projekte riešime alebo chceme riešiť.

- **Plán projektu**

Táto sekcia má čitateľom priblížiť to, čím sme sa zaoberali alebo chcem riešiť vo všetkých šprintoch.

- **O nás**

Ide o stručné predstavenie členov a vedúceho tímu spoločne s rolami všetkých účastníkov.

- **Dokumenty**

V tejto sekcii pridávame a aktualizujeme dokumenty, ktoré majú byť zavesené na webovom sídle. Dokumenty delíme do zápisníc, metodík, šprintov, evidencie úloh a ostatných nezarađených dokumentov.

Štruktúra webu

Web je implementovaný ako klasická statická stránka pozostávajúca z html, css a js súborov. Hierarchia jednotlivých súborov na virtual machine je nasledovná:

- *index.html* - Hlavná štruktúra webu vo formáte html.
- *./css/main.css* - V súbore je možné meniť štýly rôznych html tried a elementov.
- *./js/script.js* - Skriptovací súbor, kde meníme štýly niektorých html elementov na základe akcií vykonaných od používateľa.

- *./resources* - Hlavný adresár, kde sú uložené všetky dokumenty a obrázky, potrebné na vizualizáciu.
 - *./resources/evidencia_uloh* - Výpisy jednotlivých úloh všetkých členov tímu po každom šprinte.
 - *./resources/ine_dokumenty* - Dokumenty, ktoré nevieme zaradiť do iných kategórií, napríklad prihláška do TP Cupu alebo motivačný dokument.
 - *./resources/loga* - Obrázky tímové loga, ako aj ostatné logá použité na stránke.
 - *./resources/metodiky* - Metodiky, ktoré slúžia ako popis jednotlivých odvetví projektu.
 - *./resources/na_stiahnutie* - Obsahuje aktuálnu aplikáciu, ktorú je možné zo stránky priamo stiahnuť.
 - *./resources/obrazky* - Ostatné obrázky, primárne tie, ktoré sú použité ako fotky členov tímu.
 - *./resources/sprinty* - Retrospektívy všetkých šprintov.
 - *./resources/zapisnice* - Zápisnice z našich osobných stretnutí alebo stretnutí s vedúcim tímu.

Práca s dokumentami

Všetky dokumenty, ktoré sú zverejnené na našej webovej stránke, je možné zobrazit'. Navigáciou na našu stránku a po rozkliknutí vybranej kategórie sa zobrazia všetky dokumenty, ktoré tam boli pridané. Dokument sa zobrazí po kliknutí na ikonku v pravom stĺpci „Zobrazit’“.

Dokumenty ako aj obrázky, pred zobrazením, je nutné uložit' do správneho adresára. Následne, otvorením súboru index.html a navigovaním sa do správnej tabuľky, je nutné pridať nový riadok. Riadky sa pridávajú rovnako ako u všetkých doposiaľ vložených dokumentoch. Potom stačí upraviť názov, dátum a nastaviť správnu cestu k súboru, ktorý chceme pridať. Po znovu-zobrazení stránky by sme mali vidieť zmeny.