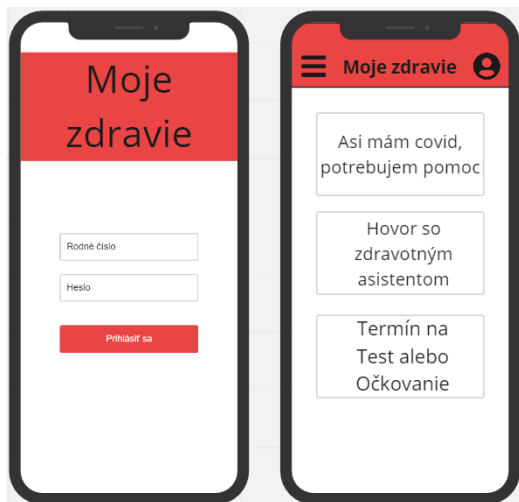


Moje zdravie

Erika Váczlavová & Tamáš Zahradniček

GitHub: https://github.com/erikavaczlavova/moje_zdravie

Wireframe-y

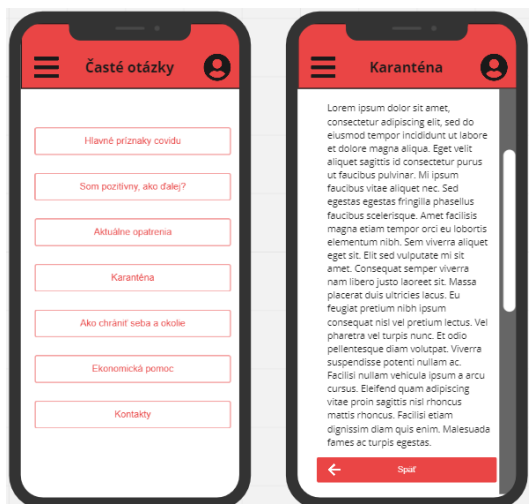


Obrazovka – Prihlasovanie

Používateľ sa prihlási rodným číslom a heslom

Obrazovka – Úvodná

Používateľ má možnosť zvoliť 3 možnosti:
„Asi mám covid..“, „Hovor...“, „Termín...“.

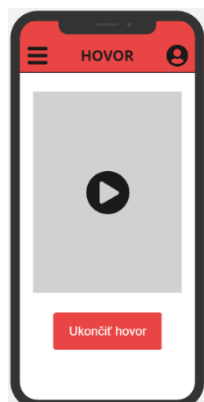


Obrazovka – Časté otázky

Obrazovka sa zobrazí po zvolení možnosti „Asi mám covid..“, zobrazia sa odkazy na najčastejšie otázky, ktoré je možné rozkliknúť.

Obrazovka – Otvorený link

Zobrazuje sa článok po rozkliknutí linku na často kladenú otázku



Obrazovka – Hovor

Obrazovka sa zobrazí po zvolení možnosti „Hovor so zdravotným asistentom“.

Obrazovka – Výber termínu

Obrazovka sa zobrazí po výbere možnosti „Termín na Test alebo Očkovanie“. Používateľ si vyberie možnosť na čo chce termín, kde a kedy.

Obrazovka – Moje údaje

K tejto obrazovke sa dostaneme cez drop-down menu. Na tejto obrazovke si používateľ zadáva údaje a potvrdí ich tlačidlom.

Drop-down hamburger menu

Zvolí sa jednotlivá možnosť, ktorá prepojí na ďalšiu obrazovku

Obrazovky po zvolení jednotlivých možností z drop down menu

The image displays four mobile app screens with a red header and a white body. Each screen has a hamburger menu icon and a user profile icon in the top right corner.

- Moje AG testy:** Three test entries. Each entry has a checkbox, a date field (Termin: DD-MM-YYYY), a result field (Výsledok testu: Pozitív/Negatív), and an address field (Miesto testu: Adresa). The first entry is selected (checkbox checked, result 'X'). The second and third entries are not selected (checkbox unchecked, result '✓'). At the bottom are two red buttons: 'Zrušiť' and 'Preobjednať'.
- Moje PCR testy:** Three test entries with the same layout as the AG tests. The first entry is selected (checkbox checked, result 'X'). The second and third entries are not selected (checkbox unchecked, result '✓'). At the bottom are two red buttons: 'Zrušiť' and 'Preobjednať'.
- Moje očkovanie:** Three vaccination entries. Each entry has a date field (Termin: DD-MM-YYYY), a dose field (Dávka: 1, 2, or 3), a doctor field (Lekár: Meno), a vaccine field (Vakcína: Výrobca), and an address field (Miesto očkovania: Adresa). The first entry is for Dávka: 3, the second for Dávka: 2, and the third for Dávka: 1.
- Covid-19 Pas:** A QR code at the top and a vaccination entry at the bottom with fields for date (Dátum: DD-MM-YYYY), dose (Dávka: 3), doctor (Lekár: Meno), vaccine (Vakcína: Výrobca), and address (Miesto očkovania: Adresa).

Obrazovka – Preobjednať

Na túto obrazovku sa používateľ dostane zvolením možnosti preobjednať pri zvolení jedného testu. Vyberie nové parametre a potvrdí.

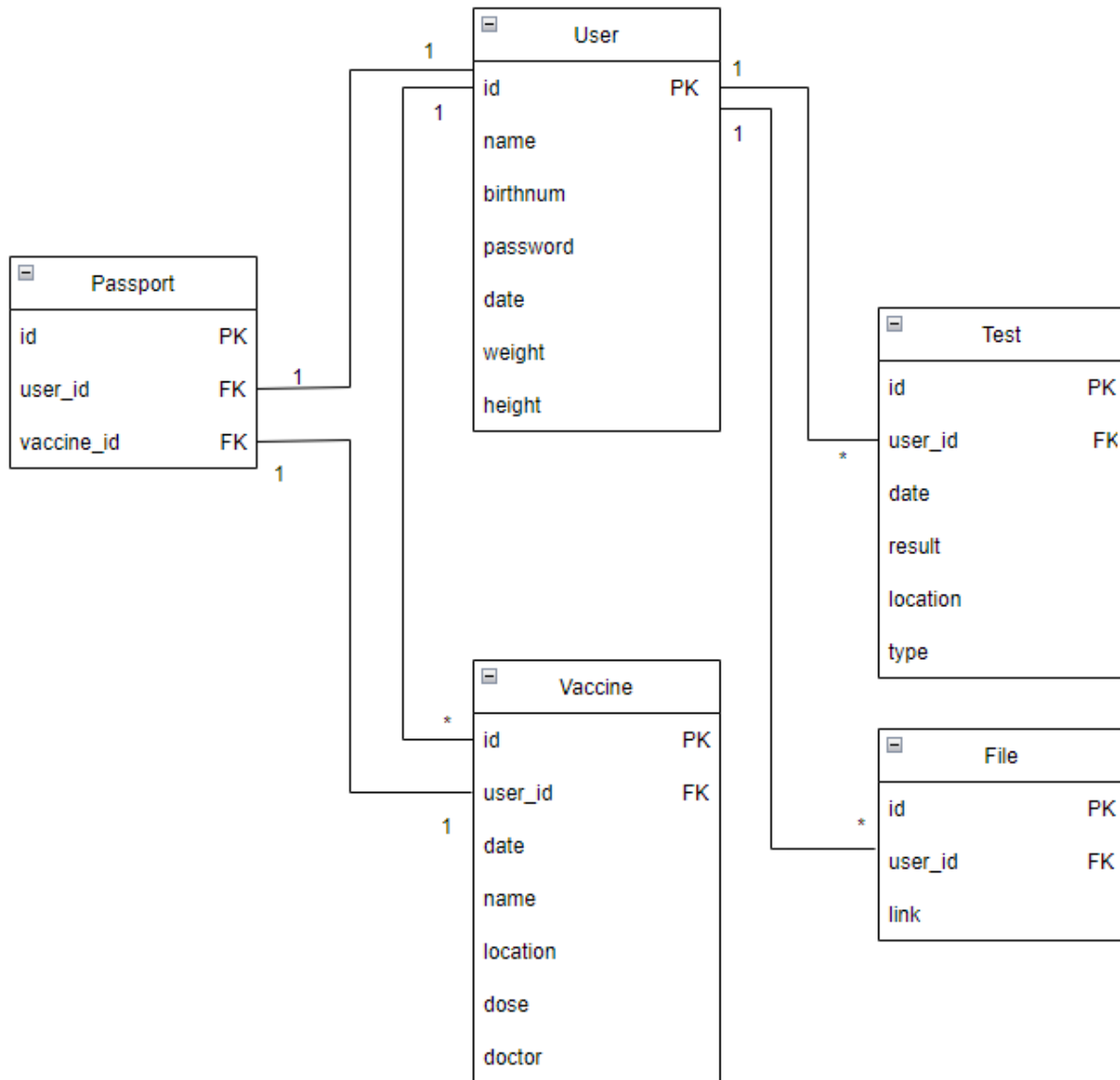
Obrazovka – Nahrať dokumenty

V tejto obrazovke je možné vkladať binárny obsah ako jpg a potom ich prezerať.

The image displays two mobile app screens with a red header and a white body. Each screen has a hamburger menu icon and a user profile icon in the top right corner.

- Preobjednať termín:** A form for booking a test. It includes fields for 'Meno' (Meno Priezvisko), 'Rodné číslo' (000000/0000), 'AG Test' (selected with a radio button), 'Pôvodný dátum' (DD-MM-YYYY), 'Nový dátum' (Dátum dropdown), 'Preferované mesto' (DD-MM-YYYY), and 'Nové Preferované mesto' (Mesto dropdown). At the bottom is a red button: 'Preobjednať'.
- Dokumenty:** A screen for uploading documents. It has a red button 'Vložiť dokument' and a section 'Dokumenty:' with a box showing 'Názov: Nález z pohotovosti' and 'Formát: pdf./jpg'.

Návrh databázy



API volania

Test Calls made to the Test endpoint			^
GET	/test	gets a test	▼
PUT	/test	adds a test	▼
POST	/test	edits a test	▼
DELETE	/test	deletes a test	▼
Vaccine Calls made to the Vaccine endpoint			^
GET	/vaccine	gets a vaccine	▼
PUT	/vaccine	adds a vaccine	▼
User Calls made to the User endpoint			^
GET	/user	gets a user	▼
POST	/user	edits an user	▼
Passport Calls made to the Passport endpoint			^
GET	/passport	gets a passport	▼
File Calls made to the File endpoint			^
POST	/file	adds a file	▼
GET	/file	gets a file	▼

Súbor s API volaniami je pridaný v GitHub repozitári projektu Moje Zdravie. Tento súbor je otvorablený napr. na linku: <https://editor.swagger.io/>, kde je každé API volanie otvorablené a je ho možné pozrieť do detailu. Každé volanie obsahuje vstupné parametre, ich typ a či sú povinnými poľami. Obsahujú vstupné aj výstupné vzorové JSON modely, tak ako aj HTTP response-y.

Ukážka jedného API volania otvoreného v Swagger:

GET /test gets a test

By passing in an existing user id and the test's type, it returns a list of tests that match the input parameters

Parameters

Try it out

Name	Description
user_id * required integer (query)	pass an existing user id
type * required string (query)	pass an existing test type Available values: AG, PCR

Responses

Response content type json/http response

Code	Description
200	list of tests Example Value Model <pre>{ "id": 1, "user_id": 1, "type": "AG", "date": "2020-08-20T00:12:33.001Z", "location": "Bratislava", "result": true }</pre>
404	Error: Not Found

Akceptačné testy – Front-end

Test 1: Úprava na screene „Profil“	
Vstupné podmienky:	Používateľ je prihlásený v aplikácii pod svojim rodným číslo a je pripojený na internet.
Výstupné podmienky:	V aplikácii bude zaznamenaná vykonaná zmena. Zmeny sa uložia/prepíšu do databázy.
Postup:	<ol style="list-style-type: none">1. Používateľ stlačí ikonu profilu2. Používateľ klikne na tlačidlo Zobrazíť profil3. Používateľ vyplní zobrazované polia správnymi údajmi4. Používateľ klikne tlačidlo „Uložiť zmeny“5. Zobrazí sa informácia o uložení údajov6. Polia zostanú vyplnené zadanými údajmi
Výsledok: PASS	

Test 2: Snaha získať informácie	
Vstupné podmienky:	Používateľ je prihlásený v aplikácii pod svojim rodným číslo a je pripojený na internet.
Výstupné podmienky:	Používateľ sa dostane k hľadaným informáciám.
Postup:	<ol style="list-style-type: none">1. Používateľ stlačí tlačidlo „Potrebujem pomoc“2. Používateľovi sa zobrazia často kladené otázky3. Používateľ vyberie tú informáciu, ktorú hľadá4. Informácia sa po zvolení zobrazí v druhom okne5. Používateľ získal potrebné informácie
Výsledok: PASS	

Test 3: Výber termínu	
Vstupné podmienky:	Používateľ je prihlásený v aplikácii pod svojim rodným číslo a je pripojený na internet.
Výstupné podmienky:	V aplikácii bude zaznamenaná vykonaná zmena. Zmeny sa uložia/prepíšu do databázy a používateľ získa žiadaný termín.
Postup:	<ol style="list-style-type: none">1. Používateľ stlačí tlačidlo „Nový test/očkovanie“2. Používateľ vyberie preferované mesto3. Používateľ vyberie dátum4. Používateľ vyberá typ testu/očkovanie5. Používateľ klikne na tlačidlo objednať6. Zobrazí sa informácia o rezervácii termínu
Výsledok: PASS	

Test 4: Úprava na screene „Moje údaje“ (negatívny)	
Vstupné podmienky:	Používateľ je prihlásený v aplikácii pod svojim rodným číslo a je pripojený na internet.
Výstupné podmienky:	V aplikácii nebude zaznamenaná vykonaná zmena. Zmeny sa neuložia/neprepíšu do databázy.
Postup:	<ol style="list-style-type: none"> 1. Používateľ stlačí ikonu profilu 2. Používateľ klikne na tlačidlo Zobraz profil 3. Používateľ zadáva rodné číslo v nesprávnom formáte 4. Používateľ vyplní zobrazované polia nesprávnymi údajmi 5. Používateľ klikne tlačidlo „Uložiť zmeny“ 6. Zobrazí sa informácia o nesprávnom vyplnení údajov
Výsledok: FAIL	

Test 5: Výber termínu (negatívny)	
Vstupné podmienky:	Používateľ je prihlásený v aplikácii pod svojim rodným číslo a je pripojený na internet.
Výstupné podmienky:	V aplikácii nebude zaznamenaná vykonaná zmena. Zmeny sa neuložia/neprepíšu do databázy.
Postup:	<ol style="list-style-type: none"> 1. Používateľ stlačí tlačidlo „Nový test/očkovanie“ 2. Používateľ vyberá typ testu/očkovanie 3. Používateľ nevyberie dátum 4. Používateľ vyberie preferované mesto 5. Používateľ klikne na tlačidlo objednať 6. Zobrazí sa informácia o nesprávnom vyplnení údajov
Výsledok: FAIL	

Akceptačné testy – Back-end

Test 1: Zrušenie termínu na test	
Vstupné podmienky:	Súčasťou volania musí byť validné id testu.
Výstupné podmienky:	V aplikácii sa zobrazí zoznam testov a zmaže jeden z nich.
Postup:	<ol style="list-style-type: none">1. Aplikácia zavolá GET (Swagger GET/test)2. Vráti sa HTTP kód 200 OK spolu so zoznamom testov pre daného užívateľa vo formáte JSON3. Aplikácia zavolá DELETE (Swagger DELETE/test)4. Vráti sa HTTP kód 200 OK s hláškou Successfully deleted
Výsledok: PASS	

Test 2: Vloženie dokumentu	
Vstupné podmienky:	Súčasťou volania musí byť validné id užívateľa
Výstupné podmienky:	V aplikácii sa zobrazí zoznam dokumentov a pridá sa ďalší.
Postup:	<ol style="list-style-type: none">1. Aplikácia zavolá GET (Swagger GET /file)2. Vráti sa HTTP kód 200 OK spolu so zoznamom dokumentov pre daného užívateľa3. Aplikácia zavolá POST (Swagger POST/file)4. Vráti sa HTTP kód 200 OK s hláškou Success
Výsledok: PASS	

Test 3: Zmena údajov	
Vstupné podmienky:	Volanie musí obsahovať všetky povinné polia s validnými informáciami a ID užívateľa.
Výstupné podmienky:	V aplikácii sú zmenené dáta viditeľné používateľovi a sú uložené v databáze.
Postup:	<ol style="list-style-type: none">1. Aplikácia zavolá GET (Swagger GET/user)2. Vráti sa HTTP kód 200 OK spolu so všetkými údajmi, ktoré sú aktuálne nahraté v databáze3. Aplikácia zavolá POST (Swagger POST /user)4. Vráti sa HTTP kód 200 OK s hláškou Successfully saved
Výsledok: PASS	

Test 4: Vymazanie termínu (negatívny)	
Vstupné podmienky:	Volanie obsahuje neexistujúce ID testu.
Výstupné podmienky:	Termín sa nezmaže.
Postup:	<ol style="list-style-type: none"> 1. Aplikácia zavolá GET (Swagger GET/test) 2. Vráti sa HTTP kód 200 OK spolu so zoznamom testov pre daného užívateľa 3. Aplikácia zavolá DELETE (Swagger DELETE/test) 4. Vráti sa HTTP kód 404 Not Found
Výsledok: FAIL	

Test 5: Neúspešné preobjednanie testu (negatívny)	
Vstupné podmienky:	Volanie obsahuje všetky povinné polia s validnými informáciami, ale neexistujúce ID testu.
Výstupné podmienky:	Termín testu sa neupraví.
Postup:	<ol style="list-style-type: none"> 1. Aplikácia zavolá GET (Swagger GET/test) 2. Vráti sa HTTP kód 200 OK spolu so zoznamom testov pre daného užívateľa 3. Aplikácia zavolá POST (Swagger POST/test) 4. Vráti sa HTTP kód 400 Bad Request
Výsledok: FAIL	

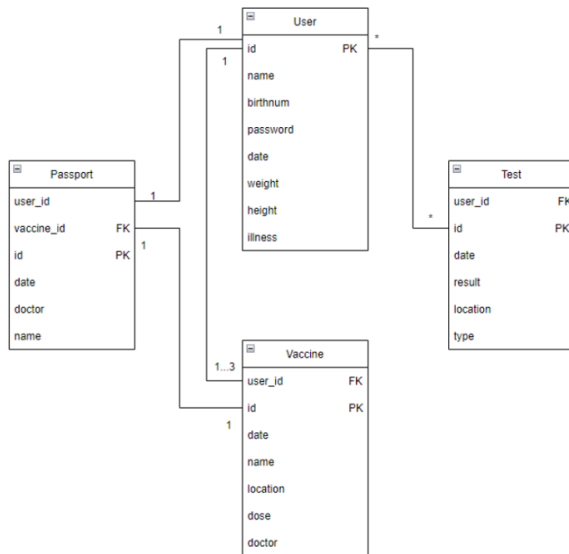
Implementačný proces Back-end

- Proces začal spoločným vytvorením Návrhu projektu
 - Implementačný proces začal nastavením a zjednotením implementačného prostredia. Prostredie na vývoj systému aplikácie je Python s použitím Django. Používaná databáza je Postgres SQL. Po iniciálnom nastavení projektu sme vytvorili spoločný Git Hub repozitár, do ktorého máme prístup obaja pracujúci na projekte.
 - K databáze pristupujeme a kontrolujeme jednotlivé záznamy po migráciách v nástroji pgAdmin4
 - Jednotlivé API volania sú testované pomocou nástroja Postman
-
- Pri implementovaní sme začali vytvorením jednotlivých tabuliek v databáze a ich prepojením pomocou PK a FK.
 - Ďalším krokom bolo implementovanie jednotlivých API volaní a pochopenie princípu ich fungovania
 - Jednotlivé API volania sa upravovali po ich testovaní, kde sa ošetrovali viaceré možné chyby
 - Kód bol upravovaný ešte na základe akceptačných testov
 - Popri implementácii API volaní bola vytváraná aj dokumentácia v Swaggeri
 - Pri každej zmene bol projekt push-nutý do Git hub repozitára, v ktorom spolupracujeme

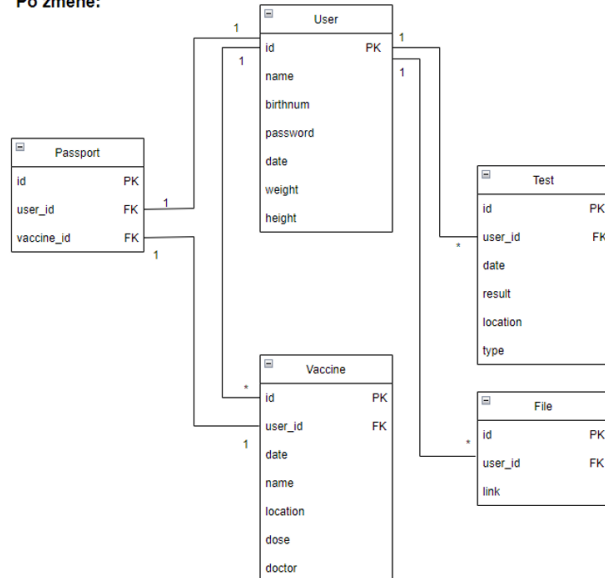
Change log (Milestone 2)

1. Zmena v databáze

Návrh databázy



Po zmene:



Odôvodnenie: Databáza bola upravená tak, aby obsahovala tabuľky na každú potrebnú informáciu. Bola pridaná tabuľka File, ktorá reprezentuje záznamy z multimediálnych záznamov, ktoré sa do aplikácie dajú vkladať, sú to súbory ako .pdf, .jpeg, .png a pod. V tabuľke Passport boli odstránené stĺpce date, doctor a name, pretože k nim prístupujeme cez záznamy z ostatných tabuliek pomocou FK.

2. Zmena formy API volaní

V Návrhu:

mojezdravie/user?name=JanVelky&birthnum=1234/5678&date=12-01-1990&weight=87&height=159&illness=skleroza
GET mojezdravie/test?user_id=2&type=AG
GET mojezdravie/test?user_id=4&type=PCR
DELETE mojezdravie/test?id=12
GET mojezdravie/vaccine?user_id=1
GET mojezdravie/passport?user_id=2
PUT mojezdravie/test?user_id=12&date=04-04-2022&location=Bratislava&type=AG
PUT mojezdravie/test?user_id=2&date=06-12-2022&location=Trnava&type=PCR
PUT mojezdravie/vaccine?user_id=6&date=05-29-2022&location=Nitra

Po zmene:

Test Calls made to the Test endpoint	
GET	/test gets a test
PUT	/test adds a test
POST	/test edits a test
DELETE	/test deletes a test
Vaccine Calls made to the Vaccine endpoint	
GET	/vaccine gets a vaccine
PUT	/vaccine adds a vaccine
User Calls made to the User endpoint	
GET	/user gets a user
POST	/user edits an user
Passport Calls made to the Passport endpoint	
GET	/passport gets a passport
File Calls made to the File endpoint	
POST	/file adds a file
GET	/file gets a file

Odôvodnenie: API volania sú v tejto fáze projektu prepracovanejšie a podrobnejšie v reálnej podobe podľa implementácie, zapracovali sme všetky potrebné zmeny a otestovali ich funkčnosť.

3. Zmena textov v akceptačných testoch pre back-end

Odôvodnenie: V textoch akceptačných testov boli pozmenené volania z návrhu, išlo iba o kozmetickú úpravu, gro a funkcionálnosť testu sa nezmenili. Texty boli pozmenené iba z dôvodu korektnosti a celistvosti projektu.

Milestone 3 – Final

Dôležité linky:

Backend : https://github.com/erikavaczlavova/moje_zdravie

Frontend: <https://github.com/erikavaczlavova/mta>

Swagger:

https://github.com/erikavaczlavova/moje_zdravie/blob/master/MTAA_API_Vaczlavova%2CZahradnecek.yaml

Implementácia Frontend

Frontend sme implementovali v React-Native npm. Ako implementačné prostredie sme využívali Visual Studio code.

Pri implementačnom procese sme si ako prioritu určili funkčnosť našej aplikácie a druhoradú vizuálnu stránku. Aplikáciu sme implementovali podľa návrhu, ktorý prešiel niekoľkými zmenami.

Implantačný proces sa začal implementáciou jednotlivých stránok/ obrazoviek aplikácie a vytvorením komponentov, ktoré sú používané a zobrazované podľa potreby a aktuálneho používania aplikácie. Pre jednoduchosť používania sme vytvorili navigačný bar na dolnej časti obrazovky, pomocou ktorého sa pohybujeme medzi jednotlivými obrazovkami.

Detaily:

- Pri každej akcii používateľa je jeho konanie potvrdené upozornením: kladným alebo záporným :

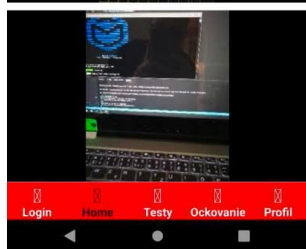


Obrázok 1 Ukážka upozornení z rôznych obrazoviek

Takéto upozornenia sa používajú v celej aplikácii s rôznymi inými správami, upravenými podľa vykonávaného úkonu.

- Využívame textové vstupy, 3 typy : vstup pre heslo, kde znaky nie sú viditeľné, vstup ktorý je upravovateľný a textový vstup ktorý sa nedá upravovať
- Využívame tlačidlá s animáciou stlačenia, tlačidlá vieme aj disable-ovať aby boli neklikateľné
- Využívame rôzne formátovania textov a rôznych elementov
- Vieme obsiahnuť všetky funkcionality opísané v dokumentácii a v testoch
- Využívame návrat na predchádzajúce obrazovky gestom /krokom späť

Implementácia WebRTC



WebRTC bolo implementované pomocou nasledujúcich linkov, návodov a git repozitárov:

<https://github.com/react-native-webrtc/react-native-webrtc>

<https://github.com/DipanshKhandelwal/react-native-webrtc-firebase>

<https://dipanshkhandelwal.medium.com/video-calling-using-firebase-and-webrtc-14cc2d4afceb>

WebRTC sme implementovali do našej aplikácie. Jeho využitie nájdeme pod tlačidlom „Hovor s asistentom“.

Akceptačné testy – Backend

Test 1: Zrušenie termínu na test – úspešný

Test 2: Vloženie dokumentu – úspešný

Test 3: Zmena údajov - úspešný

Test 4: Vymazanie termínu (negatívny) – úspešný

Test 5: Neúspešné preobjednanie testu (negatívny) - úspešný

Akceptačné testy – Frontend

Test 1: Úprava na screene „Profil“ - úspešný

Test 2: Snaha získať informácie - úspešný

Test 3: Výber termínu - úspešný

Test 4: Úprava na screene „Profil“ (negatívny) - úspešný

Test 5: Výber termínu (negatívny) – úspešný

Change-log

1. Zmena v dizajne frontend-u

Odôvodnenie: Zmeny nastali voči návrhu z dôvodu zjednodušenia používateľského zážitku, ľahšieho použitia a prehľadnosti. Farebná škála a základné obrazovky ostali bez väčších zmien.

2. Pridanie API volania Prihlásenie – Login

Odôvodnenie: Pridanie tohto volania bolo z dôvodu rozšírenia funkcionality aplikácie. Volanie využívame na obrazovke prihlásenia.

3. Prispôsobenie API volania POST Test

Odôvodnenie: Zmena je v možnosti zmeny dátumu. Miesto zmeny dátumu vieme meniť typ testu. Túto zmenu sme implementovali po vzájomnej dohode a praktickosti preobjednania.