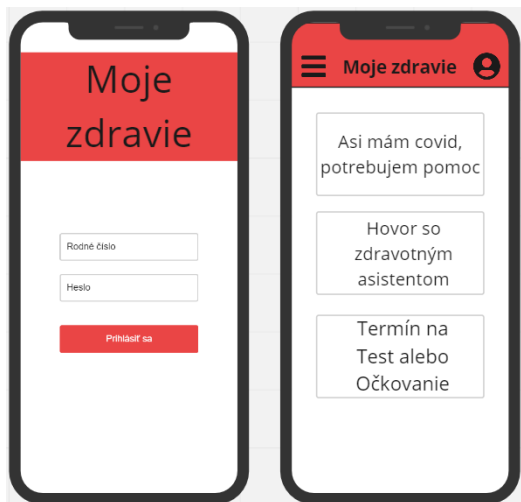


# Moje zdravie

Erika Váczlavová & Tamáš Zahradniček

GitHub: [https://github.com/erikavaczlavova/moje\\_zdravie](https://github.com/erikavaczlavova/moje_zdravie)

## Wireframe-y

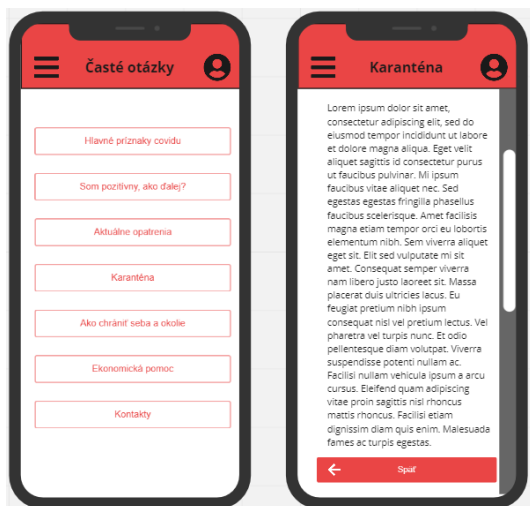


### Obrazovka – Prihlasovanie

Používateľ sa prihlási rodným číslom a heslom

### Obrazovka – Úvodná

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

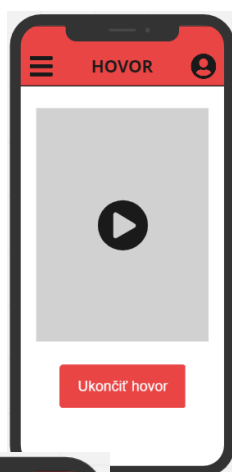


### Obrazovka – Časté otázky

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

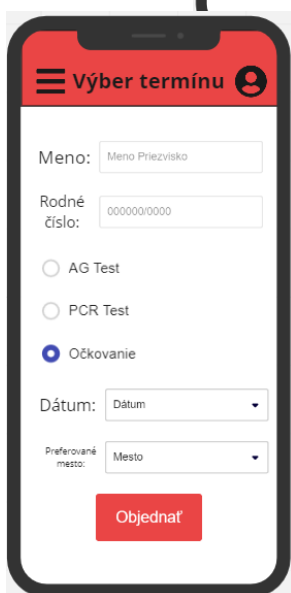
### Obrazovka – Otvorený link

Zobrazuje sa článok po rozkliknutí linku na často kladenú otázku



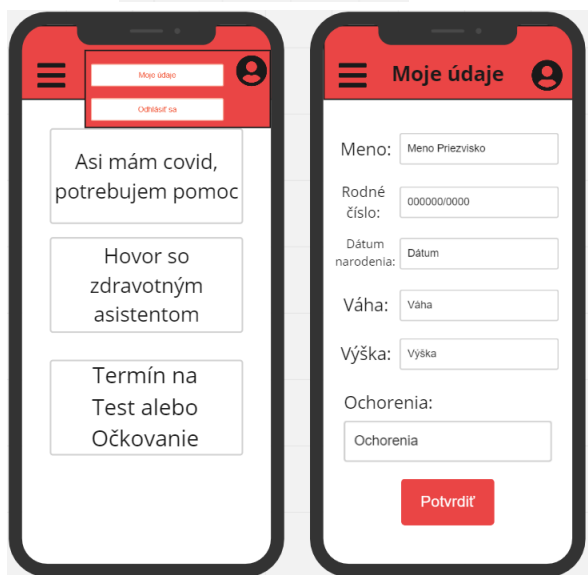
### Obrazovka – Hovor

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



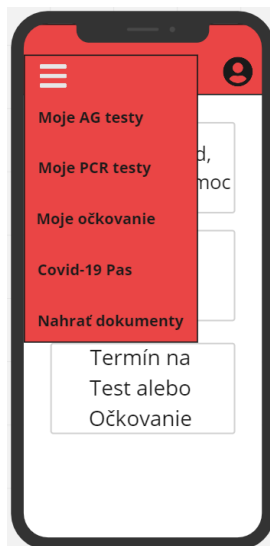
### Obrazovka – Výber termínu

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

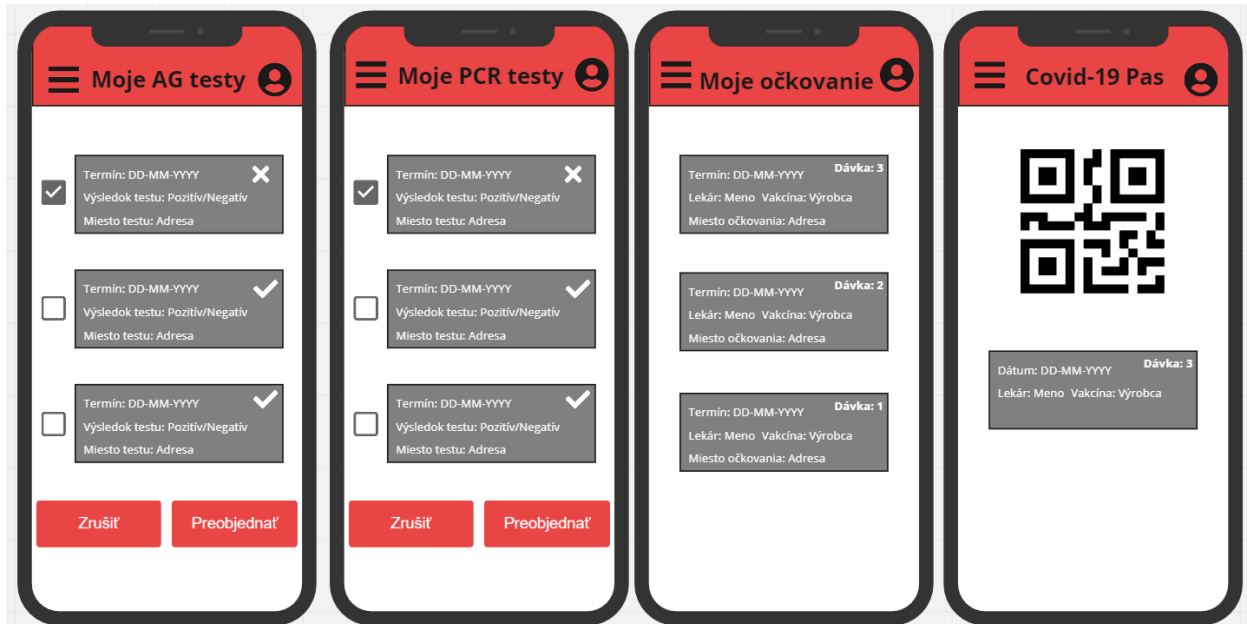
K tejto obrazovke sa dostaneme cez drop-down menu. Na tejto obrazovke si používateľ zadáva údaje a potvrdí ich tlačidlom.



### Drop-down hamburger menu

Zvolí sa jednotlivá možnosť, ktorá prepojí na ďalšiu obrazovku

### Obrazovky po zvolení jednotlivých možností z drop down menu



### 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 .pdf/.jpg a potom ich prezerať.

The image displays two mobile application screens side-by-side.

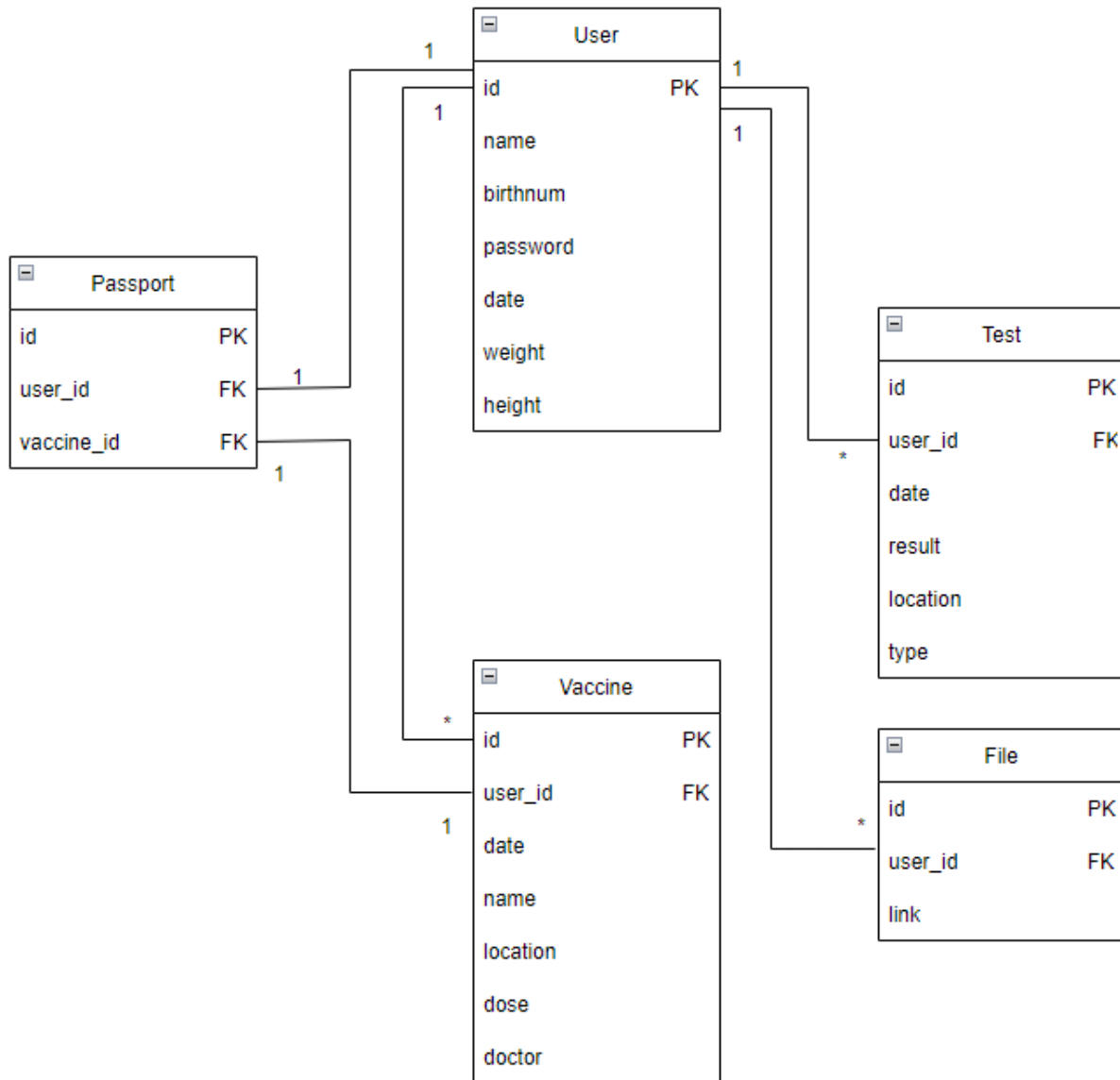
**Left Screen: Preobjednať termín**

- Header:** Red bar with a hamburger menu icon, the title "Preobjednať termín", and a user profile icon.
- Form Fields:**
  - Meno:
  - Rodné číslo:
  - ☒ AG Test
  - Pôvodný dátum:
  - Nový dátum:
  - Preferované mesto:
  - Nové Preferované mesto:
- Action:** A red button labeled "Preobjednať" at the bottom.

**Right Screen: Dokumenty**

- Header:** Red bar with a hamburger menu icon, the title "Dokumenty", and a user profile icon.
- Action:** A red button labeled "Vložiť dokument" at the top.
- Section:** A section titled "Dokumenty:" containing a grey box with the text:
  - Názov: Nález z pohotovosti
  - Formát: pdf/.jpg

## Návrh databázy



## API volania

<b>Test</b> Calls made to the <b>Test</b> endpoint		^
GET	/test gets a test	▼
PUT	/test adds a test	▼
POST	/test edits a test	▼
DELETE	/test deletes a test	▼
<b>Vaccine</b> Calls made to the <b>Vaccine</b> endpoint		^
GET	/vaccine gets a vaccine	▼
PUT	/vaccine adds a vaccine	▼
<b>User</b> Calls made to the <b>User</b> endpoint		^
GET	/user gets a user	▼
POST	/user edits an user	▼
<b>Passport</b> Calls made to the <b>Passport</b> endpoint		^
GET	/passport gets a passport	▼
<b>File</b> Calls made to the <b>File</b> 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
<b>user_id</b> * required integer (query)	pass an existing user id
<b>type</b> * 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

<b>Test 1: Úprava na screene „Moje údaje“</b>	
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"><li>1. Používateľ stlačí ikonu profilu</li><li>2. Používateľ klikne na tlačidlo moje údaje</li><li>3. Používateľ vyplní zobrazované polia správnymi údajmi</li><li>4. Používateľ klikne tlačidlo „Potvrdiť“</li><li>5. Zobrazí sa informácia o uložení údajov</li><li>6. Polia zostanú vyplnené zadanými údajmi</li></ol>
Výsledok: PASS	

<b>Test 2: Snaha získať informácie</b>	
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"><li>1. Používateľ stlačí tlačidlo „Potrebujem pomoc“</li><li>2. Používateľovi sa zobrazia často kladené otázky</li><li>3. Používateľ vyberie tú informáciu, ktorú hľadá</li><li>4. Informácia sa po zvolení zobrazí v druhom okne</li><li>5. Používateľ získal potrebné informácie</li></ol>
Výsledok: PASS	

<b>Test 3: Výber termínu</b>	
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"><li>1. Používateľ stlačí tlačidlo „Termín na test/očkovanie“</li><li>2. Používateľ skontroluje vopred vyplnené údaje databázou</li><li>3. Používateľ vyberá typ testu/očkovanie</li><li>4. Používateľ vyberie dátum</li><li>5. Používateľ vyberie preferované mesto</li><li>6. Používateľ klikne na tlačidlo objednať</li><li>7. Zobrazí sa informácia o rezervácii termínu</li></ol>
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"> <li>1. Používateľ stlačí ikonu profilu</li> <li>2. Používateľ klikne na tlačidlo moje údaje</li> <li>3. Používateľ zadáva rodné číslo/dátum v nesprávnom formáte</li> <li>4. Používateľ vyplnía zobrazené polia nesprávnymi údajmi</li> <li>5. Používateľ kline tlačidlo „Potvrdiť“</li> <li>6. Zobrazí sa informácia o nesprávnom vyplnení údajov</li> </ol>
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"> <li>1. Používateľ stlačí tlačidlo „Termín na test/očkovanie“</li> <li>2. Používateľ skontroluje vopred vyplnené údaje databázou</li> <li>3. Používateľ vyberá typ testu/očkovanie</li> <li>4. Používateľ nevyberie dátum</li> <li>5. Používateľ vyberie preferované mesto</li> <li>6. Používateľ klikne na tlačidlo objednať</li> <li>7. Zobrazí sa informácia o nesprávnom vyplnení údajov</li> </ol>
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"><li>1. Aplikácia zavolá GET (Swagger GET/test)</li><li>2. Vráti sa HTTP kód 200 OK spolu so zoznamom testov pre daného užívateľa vo formáte JSON</li><li>3. Aplikácia zavolá DELETE (Swagger DELETE/test)</li><li>4. Vráti sa HTTP kód 200 OK s hláškou Successfully deleted</li></ol>
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"><li>1. Aplikácia zavolá GET (Swagger GET /file)</li><li>2. Vráti sa HTTP kód 200 OK spolu so zoznamom dokumentov pre daného užívateľa</li><li>3. Aplikácia zavolá POST (Swagger POST/file)</li><li>4. Vráti sa HTTP kód 200 OK s hláškou Success</li></ol>
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"><li>1. Aplikácia zavolá GET (Swagger GET/user)</li><li>2. Vráti sa HTTP kód 200 OK spolu so všetkými údajmi, ktoré sú aktuálne nahraté v databáze</li><li>3. Aplikácia zavolá POST (Swagger POST /user)</li><li>4. Vráti sa HTTP kód 200 OK s hláškou Successfully saved</li></ol>
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"> <li>1. Aplikácia zavolá GET (Swagger GET/test)</li> <li>2. Vráti sa HTTP kód 200 OK spolu so zoznamom testov pre daného užívateľa</li> <li>3. Aplikácia zavolá DELETE (Swagger DELETE/test)</li> <li>4. Vráti sa HTTP kód 404 Not Found</li> </ol>
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"> <li>1. Aplikácia zavolá GET (Swagger GET/test)</li> <li>2. Vráti sa HTTP kód 200 OK spolu so zoznamom testov pre daného užívateľa</li> <li>3. Aplikácia zavolá POST (Swagger POST/test)</li> <li>4. Vráti sa HTTP kód 400 Bad Request</li> </ol>
Výsledok: FAIL	

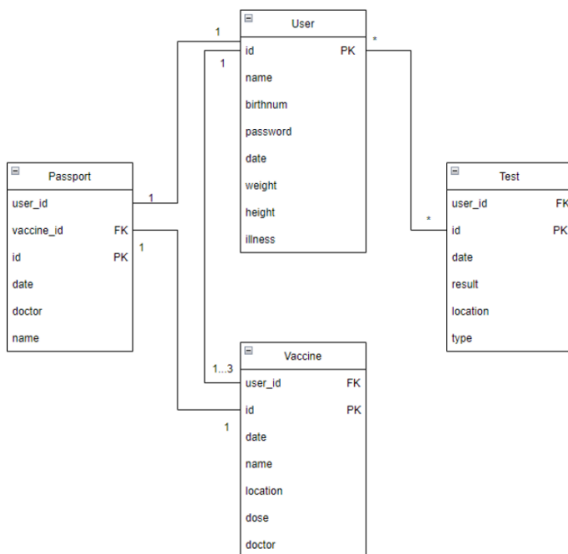
## Implementačný proces Back-end

- 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

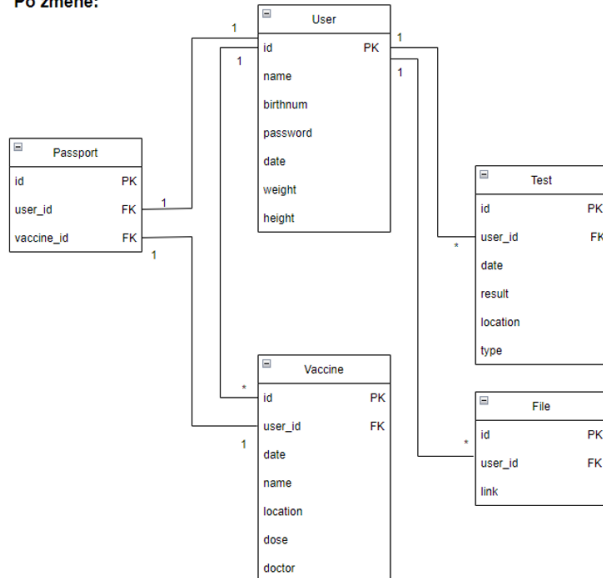
# Change log

## 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 <u>Test</u> 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 <u>Vaccine</u> endpoint	
GET	/vaccine gets a vaccine
PUT	/vaccine adds a vaccine
User Calls made to the <u>User</u> endpoint	
GET	/user gets a user
POST	/user edits an user
Passport Calls made to the <u>Passport</u> endpoint	
GET	/passport gets a passport
File Calls made to the <u>File</u> 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.