

Dokumentácia Zadania 5 predmetu UPB - Správa používateľov

Patrik Karas, Peter Karas, Kristián Danišovič, Pavol Polednák, František Hančin

1 Popis Aplikácie

Aplikácia slúži lekárom na spravovanie pacientov. Umožňuje registráciu a prihlasovanie lekárov, pridávanie a zobrazenie údajov pacientov, možnosť vytvoriť žiadanku na vyšetrenie a zobrazenie zdravotnej karty pacienta v podobe pdf dokumentu. Aplikácia pozostáva z frameworku **React** na Frontend časti aplikácie a Backend časť beží na **Nest.js**. Komunikácia medzi Backend a Frontend časťami prebieha cez HTTP požiadavky. Ďalej aplikácia používa **postgres** databázu.

2 Rozbehanie aplikácie

Vykonaajte nasledujúce kroky, v tomto poradí, na úspešné spustenie aplikácie:

- **docker-compose down**
- odstráňte všetky staré images (napr. **docker rmi \$(docker images -a -q)**)
- **docker network create my-network**
- v priečinku *backend* zadajte: **docker-compose up -d**
- v priečinku *frontend/system-doctor-fe* zadajte: **docker-compose up -d**
- otvorte odkaz: **http://localhost:3001/**

3 Nové časti aplikácie

Od predošlej verzie aplikácie pribudli viaceré funkcionality, ktoré ucelili celkovú funkčnosť systému.

3.1 Rozšírenie databázového modelu

Jednotlivé tabuľky databázy sú vygenerované pomocou knižnice **typeorm**, ktorá funguje na princípe objektovo relačného mapovania a poskytuje nástroje ako parametrizované dopyty, ktoré pomáhajú predchádzať útokom ako je **SQL Injection**. Ďalej má nástroje na validáciu pred uložením dát do databázy. V databáze **postgres** ku pôvodnej tabuľke **user**, ktorá definovala údaje lekára taktiež pribudli nasledujúce:

- Tabuľka **request**: ukladá dáta žiadanky na vyšetrenie pre pacienta.

- Tabuľka **medicalResults**: ukladá dáta lekárskeho záznamu ku jednotlivému pacientovi.
- Tabuľka **patient**: ukladá dáta o pacientovi.

Aplikácia využíva taktiež objekty **DTO** ako prostriedok prenosu údajov medzi serverom a klientom. Objekty **DTO** pomáhajú oddeliť internú štruktúru údajov a zabezpečujú, že prenášané údaje sú jasne definované a bezpečné. Pre dodatočnú vrstvu bezpečnosti je implementovaná validácia pomocou anotácií z **Nest.js** a knižnice **class-validator**.

3.2 Pridanie nového pacienta

Po kliknutí na tlačidlo **Create Patient** v navigačnom menu sa zobrazí formulár na pridanie nového pacienta pozri obr.1. Je potrebné vyplniť povinné atribúty ako unikátny identifikátor pacienta **Birth ID** meno a adresu alebo je možné vyplniť dobrovoľné atribúty ako alergie a diagnózy.

The screenshot shows a web browser at the URL `localhost:3001/create_patient`. The navigation bar at the top includes buttons for **LOG OUT**, **DASHBOARD**, **LIST OF PATIENTS**, and **UPDATE PRIVATE KEY**. The main content area features a form titled **Create Patient**. The form contains the following fields and elements:

- First Name ***: A required text input field.
- Last Name ***: A required text input field.
- Birth ID ***: A required text input field.
- Residential City ***: A required text input field.
- Street ***: A required text input field.
- Allergy**: A text input field with a blue link **ADD ALLERGY** below it.
- Diagnosis**: A text input field with a blue link **ADD DIAGNOSIS** below it.
- CREATE PATIENT**: A blue button at the bottom right of the form.

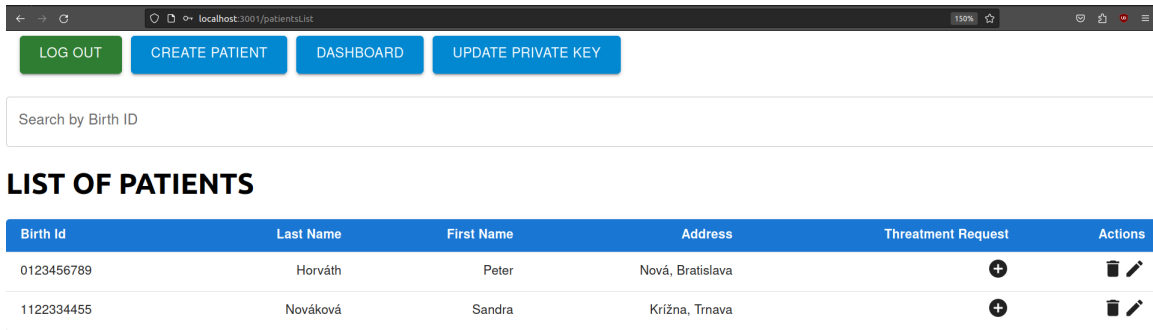
Obrázok 1: Podstránka na vytvorenie pacienta





3.3 Zoznam pacientov

Po kliknutí na tlačidlo **List of patients** v navigačnom menu sa zobrazí podstránka s tabuľkou obsahujúcou všetkých pacientov v systéme a ich základné údaje, pozri obr. 2. V hornej časti

sa nachádza *Search bar*, ktorý filtruje záznamy podľa identifikátora pacienta. Ďalej sa v tabuľke nachádzajú pri každom pacientovi nasledujúce funkcionality:

- ikonka *plus*, ktorá slúži na pridanie žiadanky na vyšetrenie pacienta.
- ikonka *koša*, ktorá odstráni existujúceho pacienta zo systému.
- ikonka *pero*, ktorá zobrazí podstránku detailu pacienta, pozri obr. 3

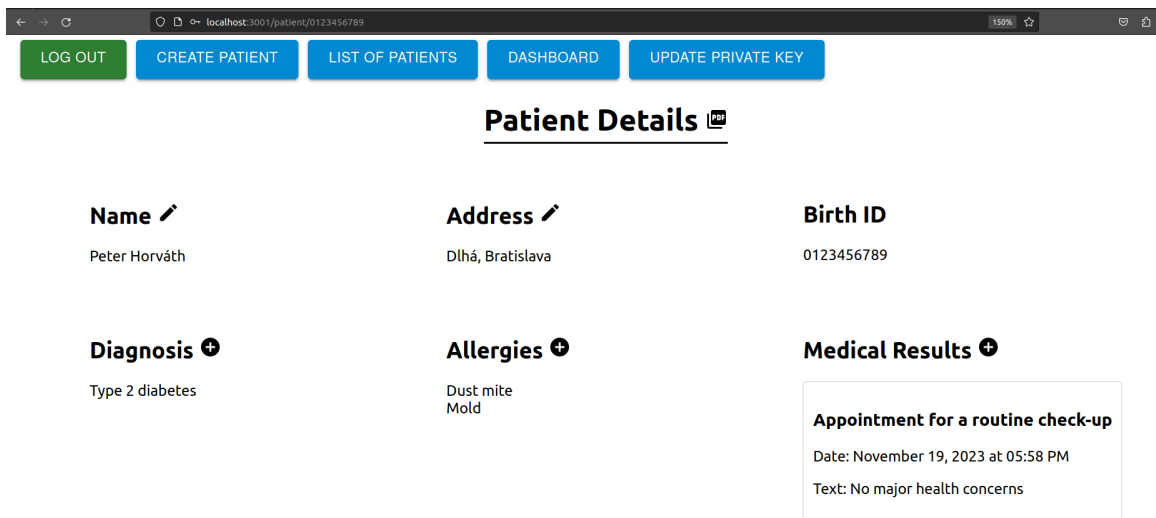


Birth Id	Last Name	First Name	Address	Threatment Request	Actions
0123456789	Horváth	Peter	Nová, Bratislava	+	 
1122334455	Nováková	Sandra	Krížna, Trnava	+	 

Obrázok 2: Podstránka zoznamu pacientov

3.4 Detail pacienta

Táto podstránka umožňuje editáciu atribútov, okrem identifikátora pacienta a pridávanie záznamu z vyšetrení. Ďalej obsahuje ikonku *pdf*, ktorá odošle požiadavku na backend s parametrom identifikátora pacienta, na backende sa vygeneruje dokument s údajmi pacienta, získaných z databázy a keďže ide o citlivé údaje, tak sa zašifrujú pomocou **public key** lekára. Následne pošle lekárovi dáta na frontend, kde sa odšifrujú pomocou **private key** lekára a zobrazí kartu pacienta v pdf formáte, ktorá sa stiahne.



Obrázok 3: Podstránka detailu pacienta

3.5 Modálne okno

Po prihlásení sa do aplikácie vyskočí modálne okno vyžadujúce **private key** doktora. Ak ho nemá k dispozícii, môže využiť funkciu priamo v aplikácii na generovanie kľúčového páru z predošlej verzie a použiť novo vygenerovaný **private key**. Ak ho nevyplní, tak modálne okno bude vyskakovať na každej podstránke a dáta posielané s backendu nebudú odšifrované a vyhodí chybové hlásenie. Ak **private key** zadá a prejde validácia formátu kľúča, tak sa hodnota uloží do *session storage*. Ak *session storage* registruje neprázdnu hodnotu, tak sa okno ďalej nezobrazuje. Pri dešifrovaní údajov z backendu sa následne použije ten **private key**. Ak je nesprávny key pair k public key tak sa dáta neodšifrujú a vyhodí chybové hlásenie. *Session storage* po vypnutí webovej aplikácie stráca údaje a teda neostávajú tam natrvalo.