**Курсов проект**

**Дисциплина: Проектиране и интегриране на софтуерни системи**

**Реализация на системата**

**Версия 1.1**

| **Фак. №** | **Име на студент** | **Секция от документа** |
| --- | --- | --- |
| **62309** | **Кирил Русев** | **Използвани технологии**  **Реализация на бизнес логиката** |
| **62287** | **Деница Йолова** | **Въведение**  **Реализация на базата от данни**  **Внедряване на системата**  **Разпределение на дейностите по реализацията** |
| **62312** | **Ралица Вунцова** | **Реализация на потребителския интерфейс** |
| **62297** | **Петя Ляскалиева** | **Използвани технологии**  **Реализация на бизнес логиката** |
| **62341** | **Мария Бошикьова** | **Използвани технологии**  **Реализация на базата от данни** |

**Януари, 2022**

**Съдържание**

[1](#_heading=h.gjdgxs) Въведение 3

[1.1 Цел 3](#_heading=h.30j0zll)

[1.2 Резюме 3](#_heading=h.1fob9te)

[1.3 Дефиниции и акроними 3](#_heading=h.3znysh7)

[2 Използвани технологии 5](#_heading=h.2et92p0)

[3 Реализация на базата от данни 6](#_heading=h.tyjcwt)

[4 Реализация на бизнес логиката 7](#_heading=h.3dy6vkm)

[5 Реализация на потребителския интерфейс 8](#_heading=h.1t3h5sf)

[6 Внедряване на системата 9](#_heading=h.4d34og8)

[7 Разпределение на дейностите по реализацията](#_heading=h.2s8eyo1) 10

# Въведение

## Цел

Настоящия документ описва имплементацията на системата MedRec, която представлява приложение за информация за различни лекарства. Целта на приложението е да улесни едновременно аптеките и пациентите при продажбата и закупуването на лекарствата. Ще се предостави възможност на аптеките да показват наличността на предлаганите от тях лекарства, а пациентите ще могат да преглеждат наличността в дадените аптеки и да търсят по желани от тях лекарства.

## Резюме

Основната функционалност на системата е предостоставяне на бърз достъп както на хората, така и на аптеките, до наличните в аптеките лекарства. Клиентите могат да търсят аптеки и да виждат наличните там лекарства, както и да виждат предписаните им рецепти чрез QR кодове. Аптеките могат да обновяват количеството на своите лекарства, да ги изтриват от базата данни или да добавят изцяло нови такива.

В документа са описани предназначението и основните функционалности на системата, както и използваните технологии за нейното създаване и работа.

Описана е базата данни с всички таблици в нея и връзките между тях, както и реализацията на самата бизнес логика на системата - кои са отделните части на системата и как те работят заедно. Потребителския интерфейс също е описан, като са представени екранни снимки, които да показват основните възможни операции в приложението.

## Дефиниции и акроними

* *Потребителски интерфейс* - интерфейс, при който елементите, предоставени на потребителя за управление, са изпълнени във вид на графични изображения (менюта, бутони, списъци и др.).
* *Приложение (app, application)* - софтуер, предназначен да помогне на потребителя да извърши определена задача
* *Уеб приложение* - приложение, до което потребителите имат достъп през мрежа като Интернет
* *Клиент* – част от компютърна или софтуерна система, която достъпва услуга, предоставена от сървър
* *Сървър* – стартирана инстанция на софтуерна система, която може да приема заявки от клиент и да връща подходящи отговори
* *База от данни* – колекция от информация, която е така организирана, че да може лесно да се достъпва, управлява и актуализира
* *Модул* – логически обособена софтуерна единица
* *Потребител* – човек, който използва компютърна или мрежова услуга

# Използвани технологии

Операционната система е Windows.

За реализацията на сървърната част на системата е използван Java Spring boot с Hibernate. Версия на Spring framework - 2.6.2. Версия на Java - 11. За deploy на сървъра се използва вградения уеб сървър Apache Tomcat 7.

Към сървъра е свързана външна функционалност за сканиране на QR кодове. Тя представлява Go сървър, който си комуникира чрез сокет със Spring сървъра. Версията на Go е 1.17.6.

Използва се библиотека gozxing на Go сървъра за декодиране на qr кодове, съдържащи информация за рецепти за лекарства. Резултатът е в XML формат, който чрез библиотеката goxml2json се конвертира в JSON формат и се връща на сървъра.

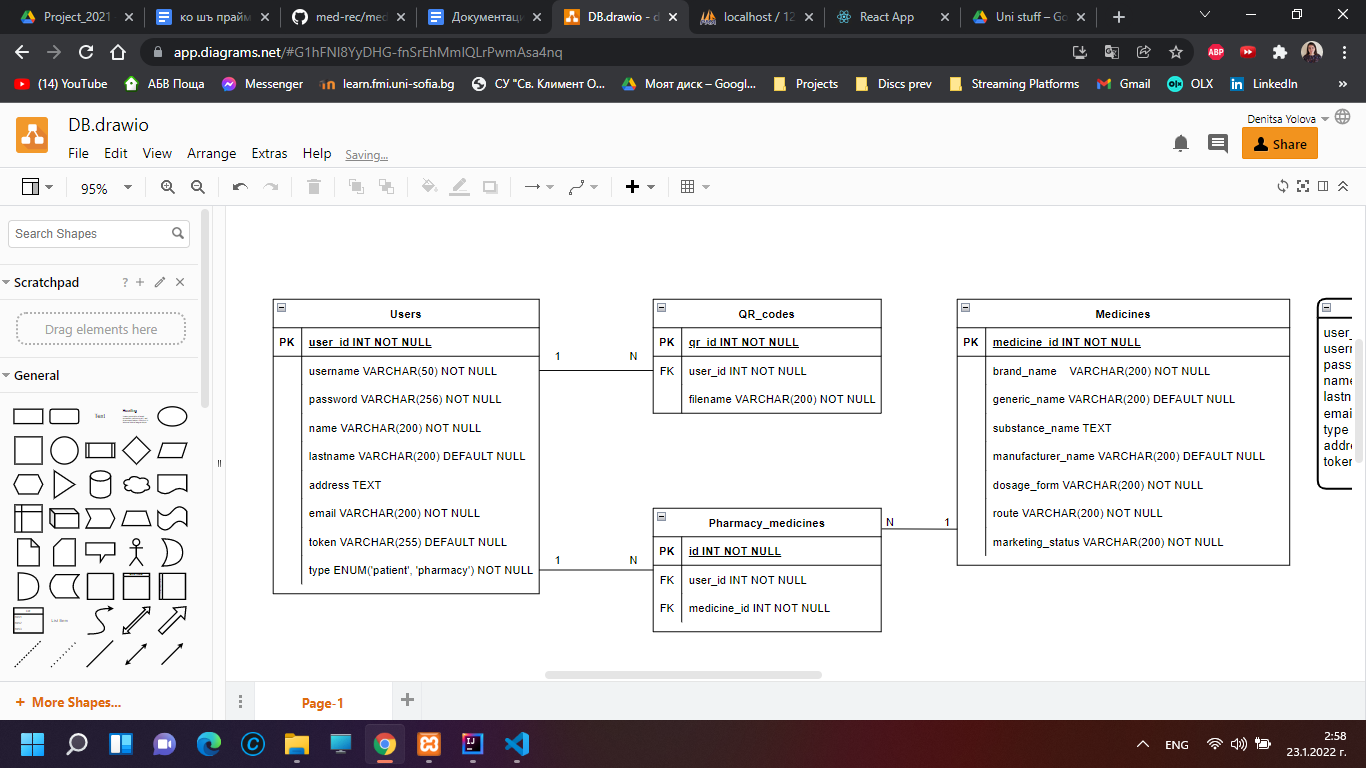
За клиентската част на системата са използвани Node.js, версия 16.13.2, и Reactjs (версия 17.0.2). За първоначалното изграждане на приложението, използващо Node.js и React Js е използван Create React App.

Архитектурата на клиентското и сървърното приложение е базирана на REST.

Използва се базата Drugs@FDA (<https://open.fda.gov/apis/drug/drugsfda/>), която е публично достъпна през API, за извличане на информация за лекарствата, които се добавят в базата данни.

За направата на базата данни е използвана MariaDB, версия 5.6.14. Тя бива хоствана на XAMPP 3.2.1. Платформата phpMyAdmin се използва да администриране и конфигуриране на базата данни, създадена с MariaDB.

# Реализация на базата от данни



Схемата на същности отношения (entity relationship) е представя базата данни на системата и връзките в нея.

Базата е съставена от няколко таблици, свързани по между си:

* **Users** - таблица, пазеща информацията на всички регистрирани потребители - аптеки и пациенти. Има Primary key user\_id и се свързва с таблиците QR\_codes и Pharmacy\_medicines с връзка един към много.
* **QR\_codes** - таблица с данните за QR кодовете, която се свързва с Users чрез своя Foreign key user\_id. Primary key-а на таблицата е qr\_id.
* **Medicines** - таблица, съхраняваща информация за всички видове лекарства в базата данни. Primary key-а е medicine\_id и таблицата е свързана с Pharmacy\_medicines с връзка един към много.
* **Pharmacy\_medicines** - таблица с Primary key id. Свързана е едновременно с Users и Medicines, като и с двете връзката е един към много.

# Реализация на бизнес логиката

Бизнес логиката на системата е реализирана чрез използването на Java Spring boot. Използваната архитектура за приложението е MVC (Model View Controller), като за целта проектът е разделен на следните части:

* Controller - приема HTTP заявки и след обработката им, връща резултат под формата на JSON обект и с определен статус код. Обработката на заявките се извършва в Service пакета. В DTO пакета се намират класовете, които се използват за предаване на информацията между клиента и сървъра.
* DTO (Data Transfer Object) - класове, които се използват за приемане на информация от заявки и за връщане на информация при процеса на комуникация между клиентската и сървърната част на системата. Чрез тях се определя каква част от информацията да бъде споделена с потребителския интерфейс.
* Exceptions - грешките (exceptions), които могат да бъдат хвърлени при обработката на данните, получени от клиента и взаимодействието с базата данни. Ако настъпи някаква грешка, се изпраща подходящ статус код като отговор на заявките на клиента.
* Model - всички основни модели на обектите, които се използват в приложението. Чрез Hibernate ORM на всяка таблица от базата данни се съпоставя един модел.
* Repository - отговарят за връзката между бизнес логиката и базата данни. Чрез Hibernate ORM директно могат да се изпълняват основни SQL заявки върху данните в базата - операции по създаване, промяна, търсене и триене (CRUD).
* Security - класове, осигуряващи сигурността при осъществяването на HTTP заявки към сървъра. Всички заявки трябва да съдържат Authentication header с валиден JWT token. Единствено заявките за автентикация - за логин и регистриране, са позволени без да има автентикиране на потребителя.
* Service - съдържа всички класове, които са отговорни за бизнес логиката на приложението. Методите в класовете приемат и връщат резултат DTO обекти, като част от комуникацията си с контролерите. За осъществяване на бизнес логиката си комуникират и с частите, отговарящи за базата данни - Model и Repository. Тук се намира логиката за автентикация, обработка на информацията за пациентите и аптеките, за сканиране на QR кодове и извличане на информация за лекарства от външна уеб услуга.

В системата се извършва комуникация с външна уеб услуга (Drugs@FDA API). Чрез HTTP GET заявки се взимат данните за лекарствата, които се добавят в базата данни.

Към системата е добавена подсистема за комуникация с отдалечена услуга за разчитане на QR кодове. Тя използва външни библиотеки, за да ги декодира в XML формат и да ги преобразува от XML в JSON формат.

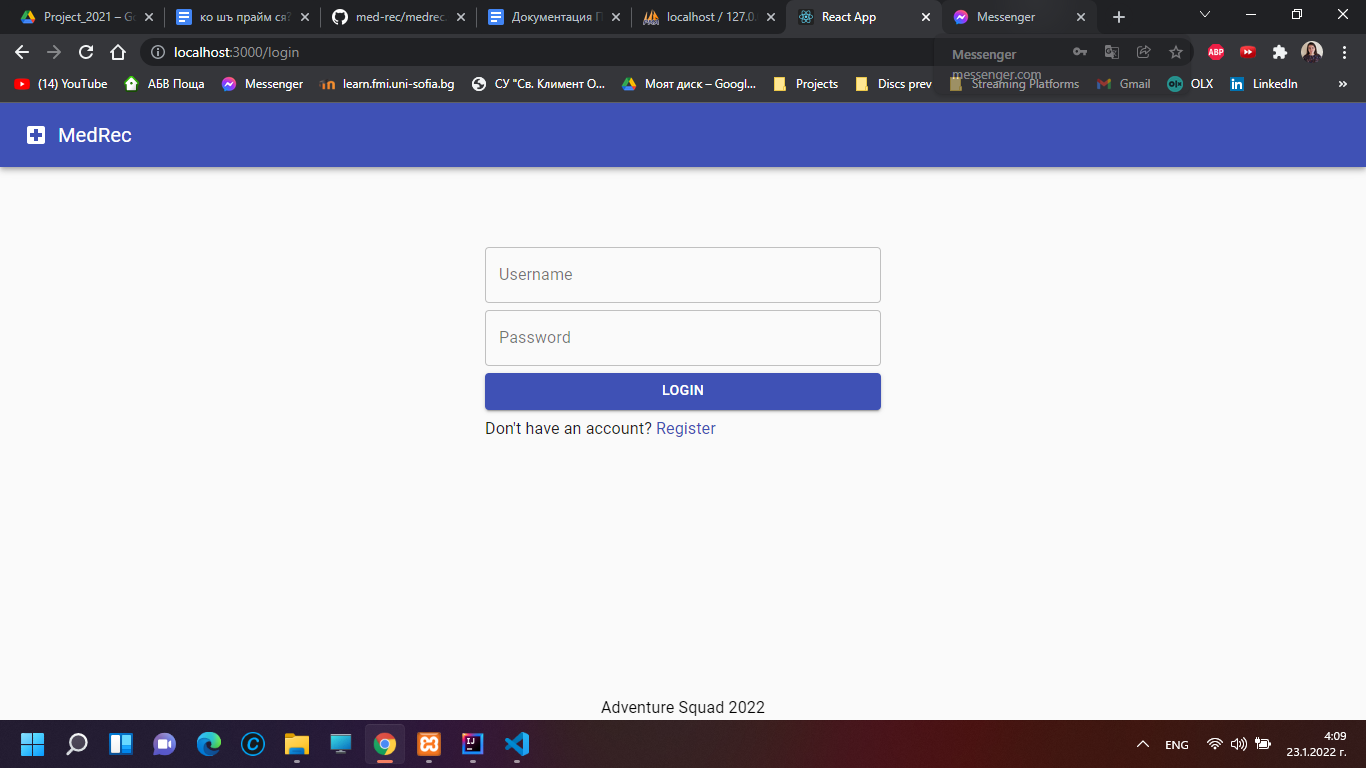
# Реализация на потребителския интерфейс

Първоначално на екрана се появява формата за вход или регистрация в системата на различните потребители. Тези потребители могат да са пациенти или аптеки. Пациентите могат както да се регистрират, така и да влизат в системата, а аптеките биват добавяни директно в базата от админ, след което могат да влизат в приложението.

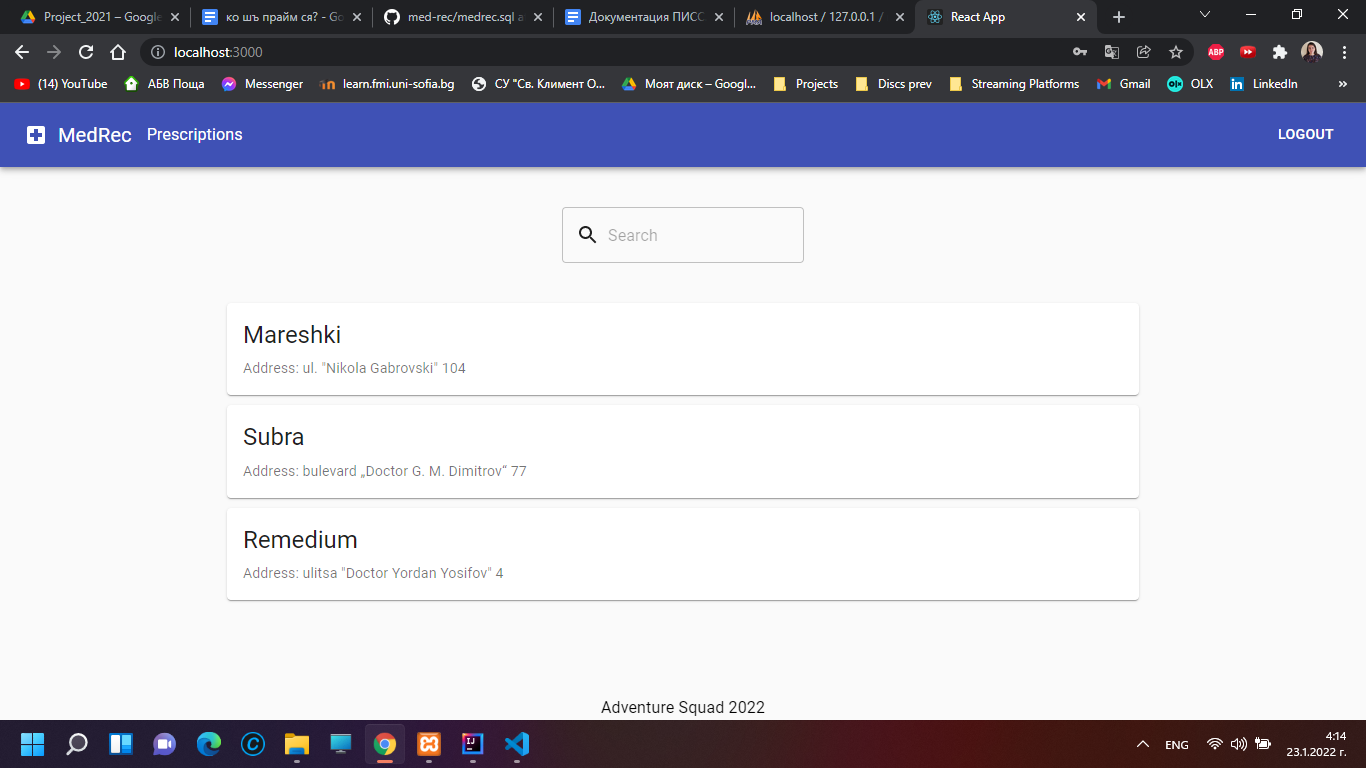
Ако потребителят е пациент, след влизане той бива пренасочен към страницата, показваща всички аптеки в системата. Той има възможност да търси аптеките по имена и да избира някоя от тях. Когато избере и натисне върху някоя аптека, пациентът бива пренасочен към страница, показваща наличните в тази аптека лекарства. Освен това, може да провери предписаните му рецепти и да види и отвори техните QR кодове. При натискане на QR кода се зарежда страница, показваща самата рецепта и начинът, по който да се приема даденото лекарство.

Ако потребителят е аптека, той се пренасочва към страница, показваща различните лекарства, налични там. Има възможност да се добави ново лекарство или да се изтрие вече съществуващо такова от системата.

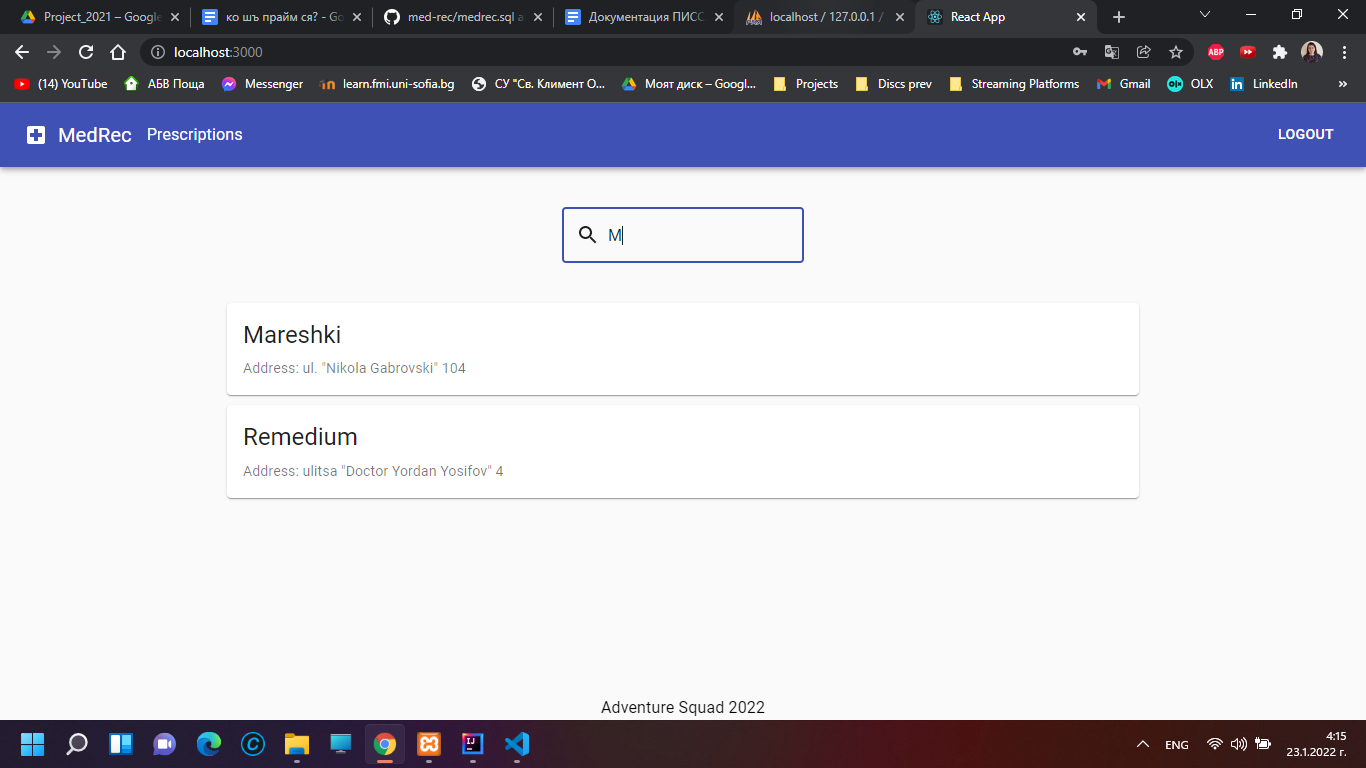
И в двата случая може да се излезе от системата, като потребителя бива върнат в първоначалната страница за вход.



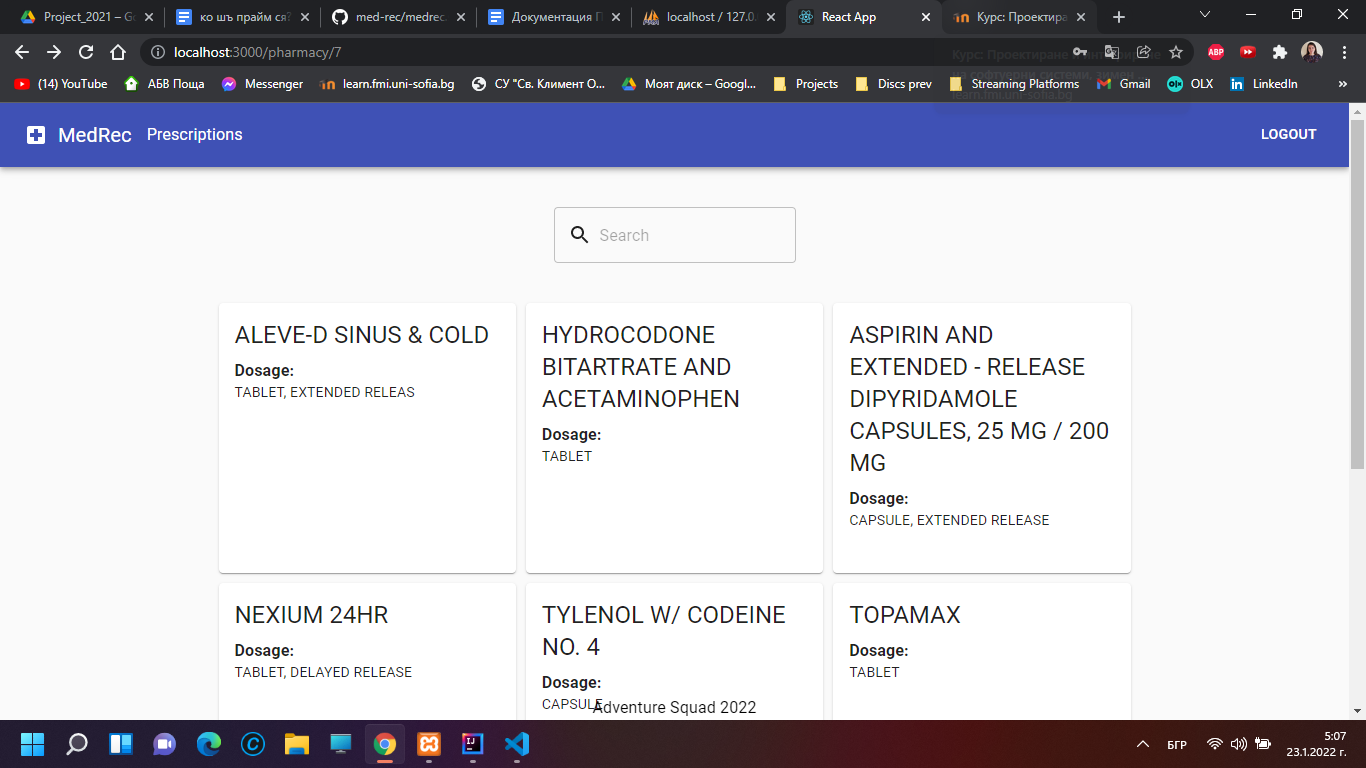
*Начален екран за вход в системата (1)*



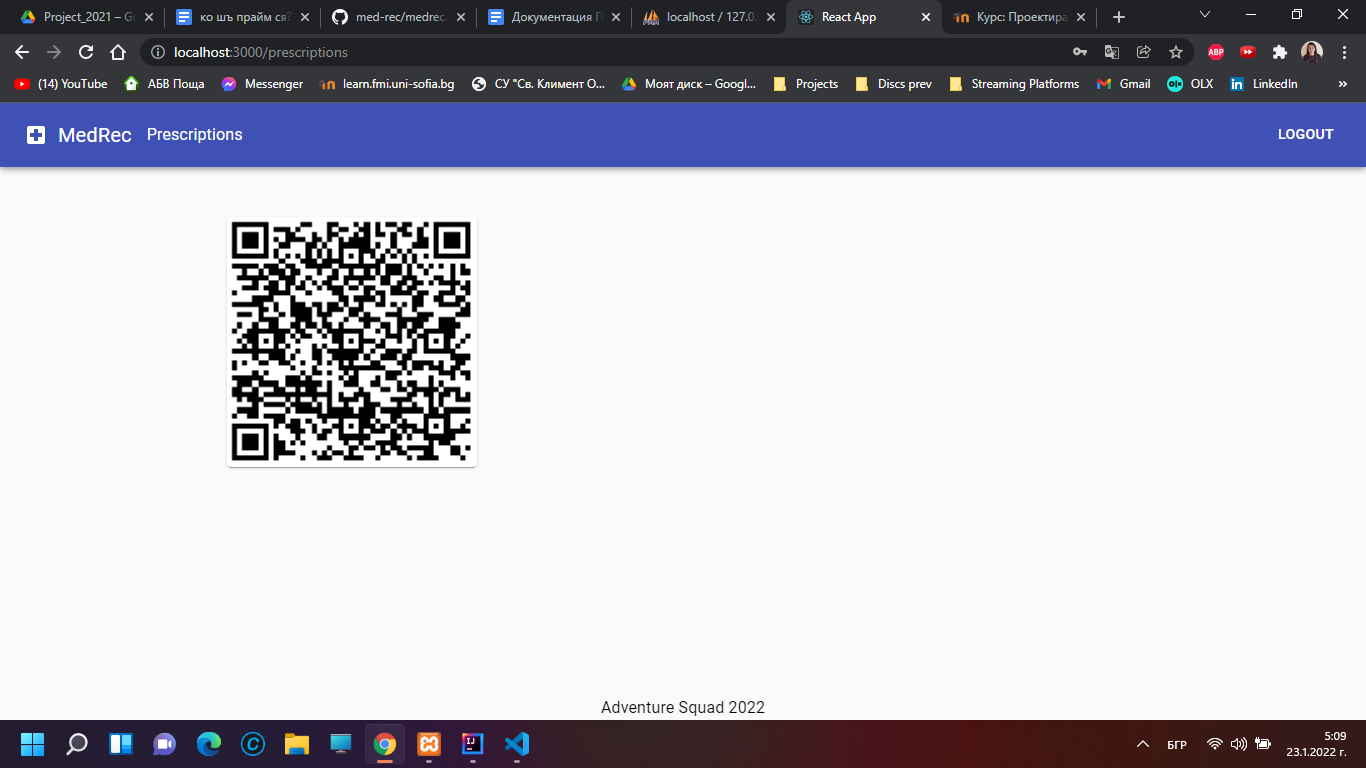
*Екран, показващ всички аптеки в системата(2)*

**

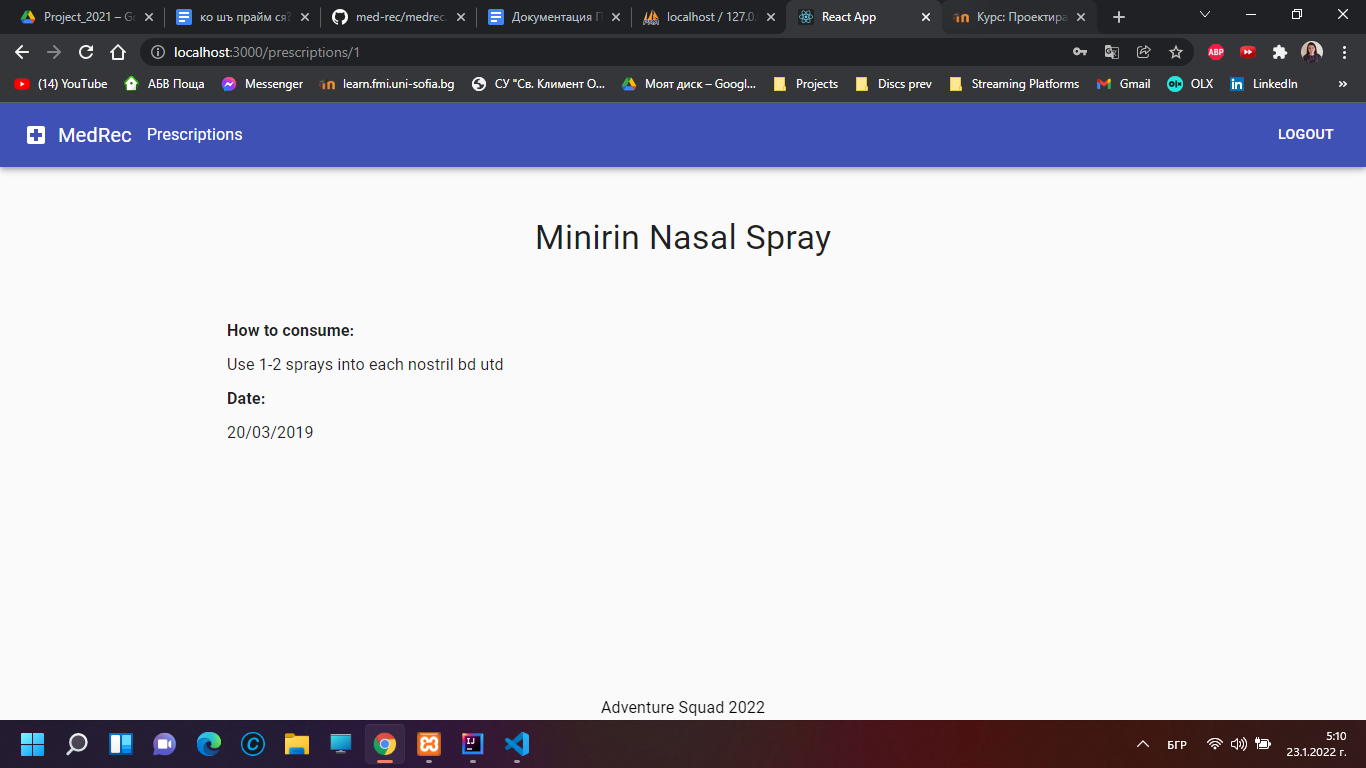
*Търсене на аптеките в системата по имена (3)*

**

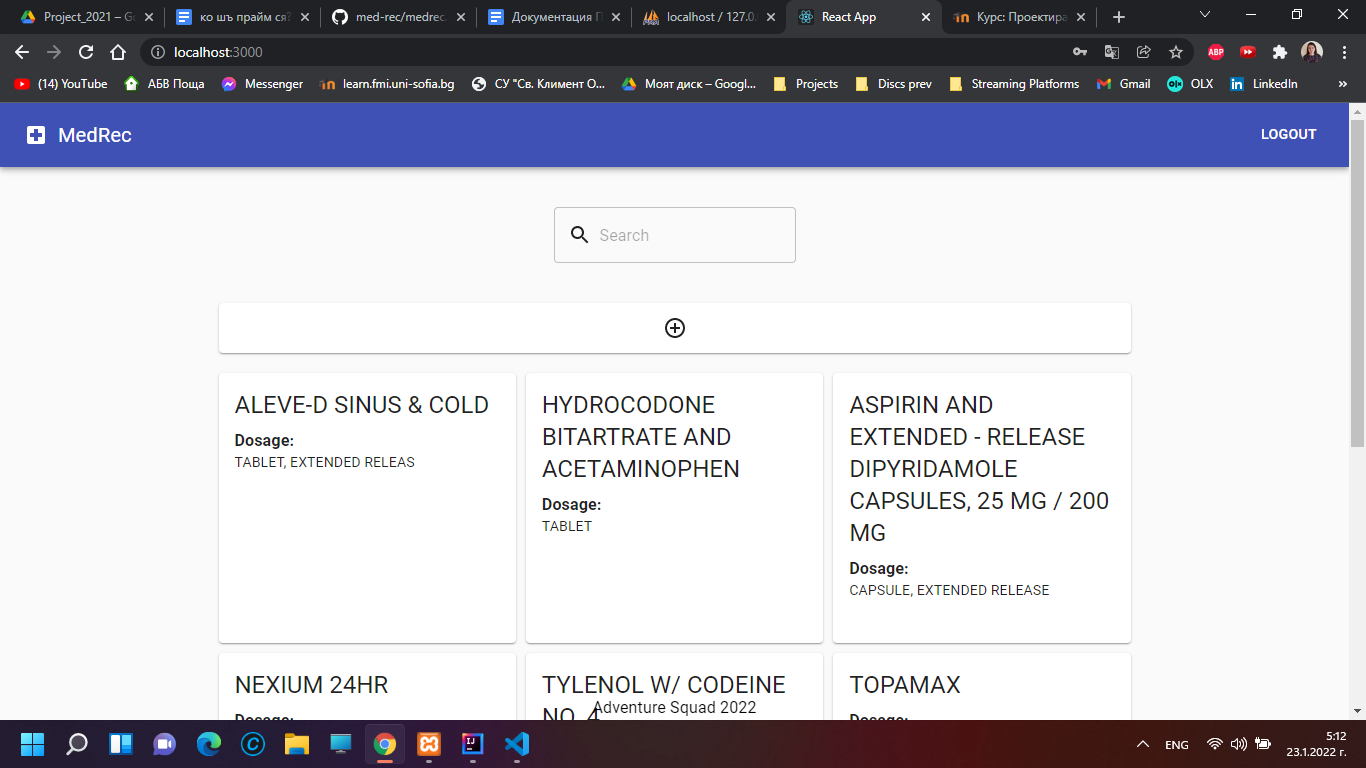
*Екран, показващ какви лекарства има в избраната аптека (4)*

**

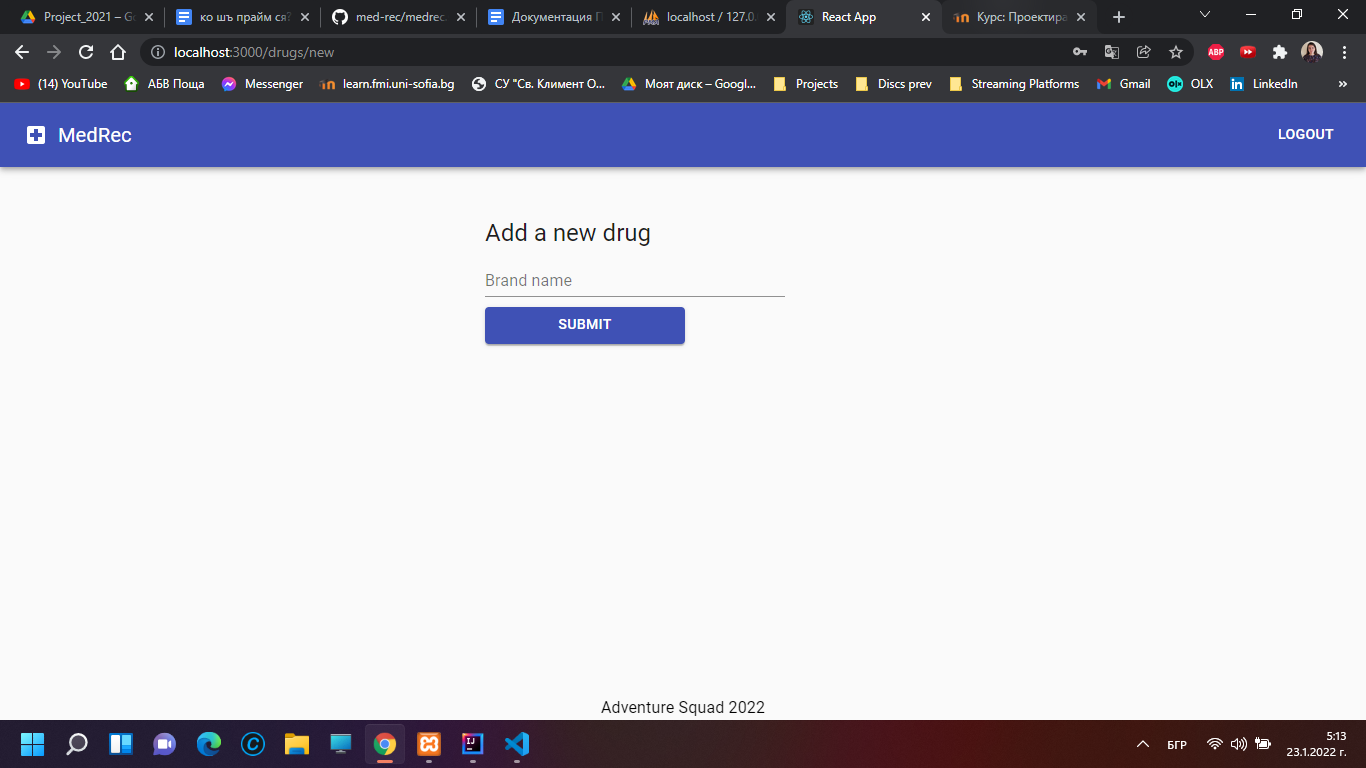
*При натискане на Prescriptions се показва екран с всички предписани на човека рецепти под формата на QR код (5)*



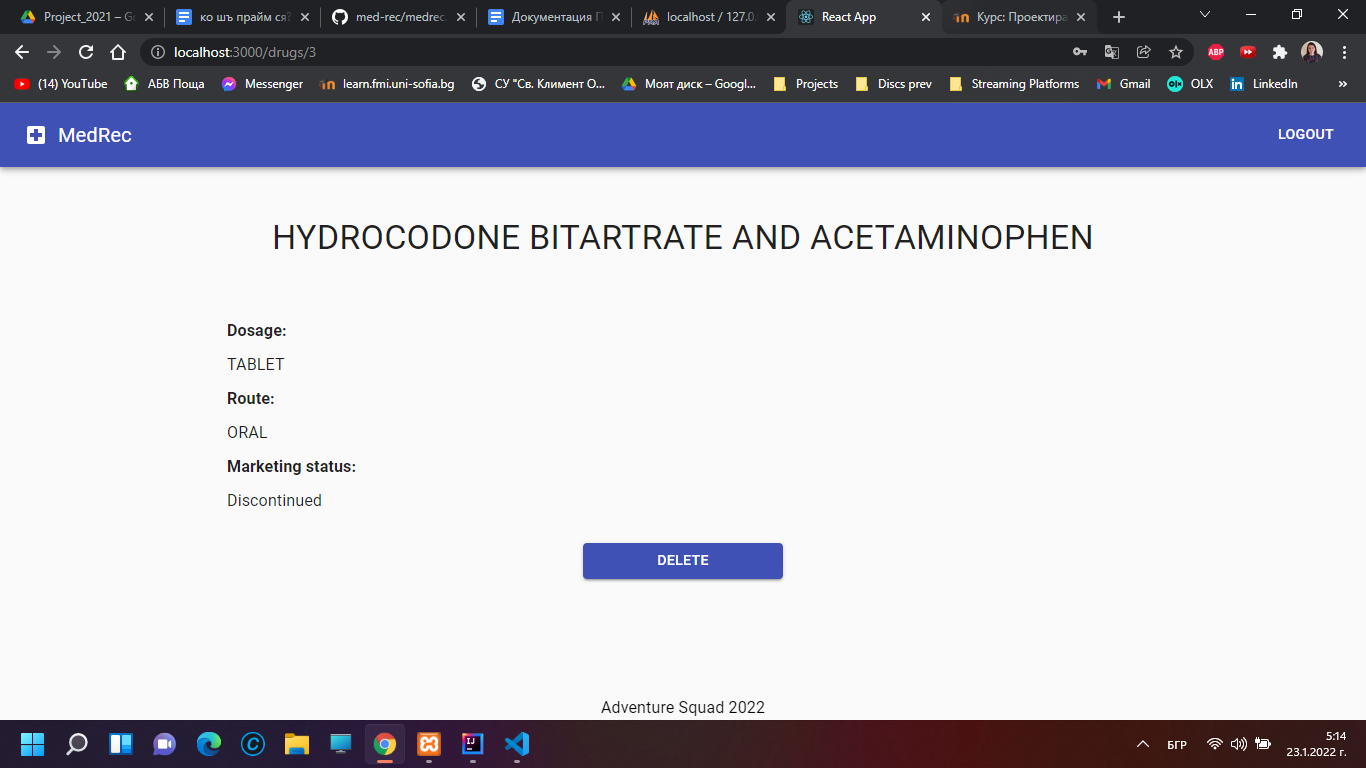
*След натискане на QR кода за рецептата, се показва информация за лекарството и как да се използва (6)*



*При влизане като потребител аптека се вижда екран с наличните лекарства в тази аптека и може да се добавя ново от знакът ‘+’ (7)*

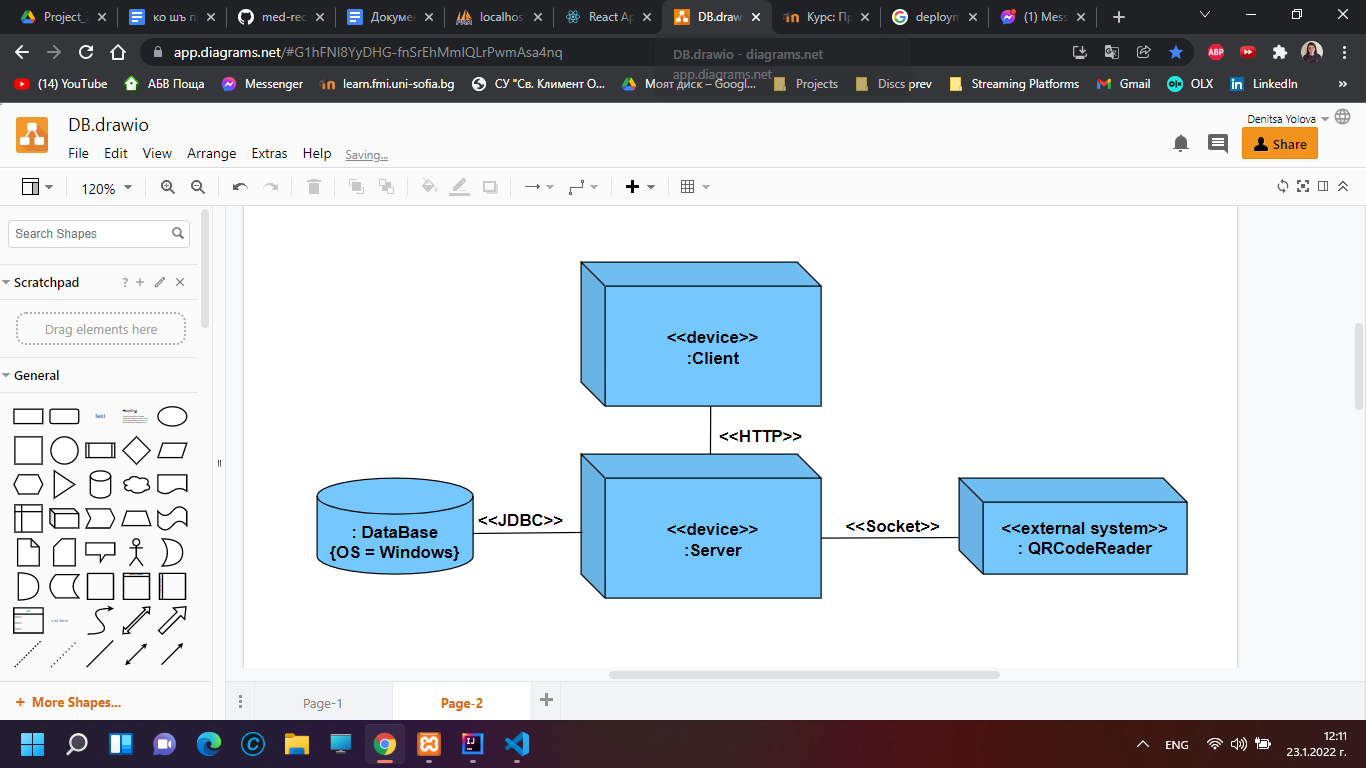


*Екран за добавяне на ново лекарство в аптеката (8)*



*При натискане върху дадено лекарство се вижда информация за него и има възможност да се изтрие от ‘Delete’ бутона (9)*

# Внедряване на системата



# Разпределение на дейностите по реализацията

* *Кирил Русев* - разработка на backend частта на системата (Spring сървъра и Go service-а)
* *Петя Ляскалиева* – разработка на backend частта на системата
* *Мария Бошикьова* – създаване на базата от данни
* *Ралица Вунцова* – разработка на frontend частта на системата
* *Деница Йолова* – документация + презентация