

КУРСОВ ПРОЕКТ

ДИСЦИПЛИНА: ПРОЕКТИРАНЕ И ИНТЕГРИРАНЕ НА СОФТУЕРНИ СИСТЕМИ

РЕАЛИЗАЦИЯ НА СИСТЕМАТА

ВЕРСИЯ 1.0

ФАК. №	ИМЕ НА СТУДЕНТ	СЕКЦИЯ ОТ ДОКУМЕНТА
62494	Весела Петрова	<ul style="list-style-type: none">● РЕАЛИЗАЦИЯ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС
62391	НАДЕЖДА ФРАНЦЕВА	<ul style="list-style-type: none">● РЕАЛИЗАЦИЯ НА БИЗНЕС ЛОГИКАТА● ИЗПОЛЗВАНИ ТЕХНОЛОГИИ● РАЗПРЕДЕЛЕНИЕ НА ДЕЙНОСТИТЕ ПО РЕАЛИЗАЦИЯТА
62480	ИВАНА ДРАГНЕВА	<ul style="list-style-type: none">● ВЪВЕДЕНИЕ● ВНЕДРЯВАНЕ НА СИСТЕМАТА
62448	СИЛВИЯ ДЕЯНОВА	<ul style="list-style-type: none">● РЕАЛИЗАЦИЯ НА БАЗАТА ОТ ДАННИ

ЯНУАРИ, 2023

Съдържание

1	Въведение	3
1.1	Цел	3
1.2	Резюме	3
1.3	Дефиниции и акроними	3
2	Използвани технологии	4
3	Реализация на базата от данни	5
4	Реализация на бизнес логиката	6
5	Реализация на потребителския интерфейс	7
6	Внедряване на системата	8
7	Разпределение на дейностите по реализацията	9

1 ВЪВЕДЕНИЕ

1.1 Цел

Настоящият документ описва имплементацията на системата HallSystem, в която лекторите могат да запазват зали в различни факултети за определени часове. Целта на софтуерната система е да се даде на преподавателите правото да резервират стаи спрямо техните график и предпочитания.

1.2 Резюме

Приложението е предназначено за използване от преподавателите на Факултета по математика и информатика. Системата предоставя следните функционалности:

- Регистрация
- >Login - влизане в системата след регистрация
- Разглеждане на зали - показва се страница с детайли за зали в няколко факултета
- Резервиране на зала - резервира се определена зала в избрани от потребителя дата и часове
- Преглеждане на резервирани зали - показва се с страница с вече направени резервации от потребителя
- Logout - Излизане от системата

HallSystem използва принципите на разпределените софтуерни архитектури, позволяващи бързото обработване на заявки. Отделните композиционни модули са отделени един от друг, избягвайки конкуренцията на задачите.

За целта системата се състои от 4 отделни модула:

- Rest API - Тази част е основния компонент на системата. REST стандартът налага компонентът да е stateless - използва се единствено базата от данни, за съхранение на потребителските данни.
- Потребителски интерфейс - Визуализация на информацията, предоставена от Rest Api-то
- Бази данни - Тук се съхраняват данните за обектите, които присъстват в системата и техните взаимодействия.
- Web server - Необходим ни е за изпращане на заявки до базата с цел извличане на информация от нея.

1.3 Дефиниции и акроними

- Rest API - уеб интерфейс, използващ принципите на HTTP и REST

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

2 Използвани технологии

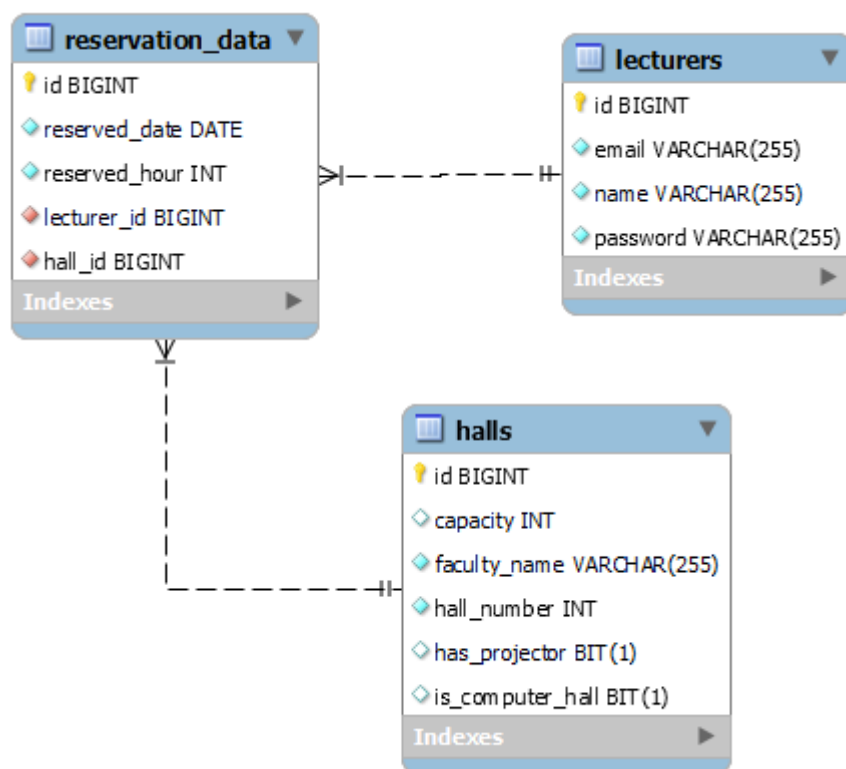
Използваме Java Spring Boot с основните му dependencies за backend бизнес логиката.

Реализацията на потребителския интерфейс е постигната чрез TypeScript, HTML и CSS с помощта на framework Angular. За допълнително стилизиране сме добавили Bootstrap. Служим си също така с Angular Material библиотека, която ни предоставя компоненти за потребителския интерфейс (падащо меню, бутони и т.н.)

За базата от данни използваме MySQL.

3 РЕАЛИЗАЦИЯ НА БАЗАТА ОТ ДАННИ

Използваната база данни е MySQL. Състои се от следните таблици: halls, lecturers, reservation_data. Тяхната структура е следната:



4 РЕАЛИЗАЦИЯ НА БИЗНЕС ЛОГИКАТА

Бизнес логиката е реализирана чрез използването на Java Spring Boot framework.

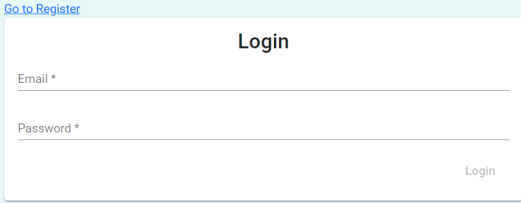
Имаме няколко логически единици в рамките на нашето Spring Boot приложение, всяка от които си има своите отделни функции:

- **Models** - това са всички основни обекти, които съществуват в системата като Hall, Lecturer и т.н. заедно с техните атрибути, за които трябва да се дефинира конструктор, setter и getter. В общия случай на всеки Model се съпоставя таблица в базата данни (чрез т.нар. ORM - Object Relational Mapping, като в случая е използван Hibernate ORM).
- **Repositories** - използват се за комуникацията между бизнес логиката и базата данни. Отново чрез ORM са предефинирани всички основни SQL заявки върху данните - т.нар. CRUD операции (Create, Read, Update, Delete) като се спестява тази задача на разработчика. По този начин приложението не е зависимо от използвания вид база данни и нейния синтаксис на SQL заявки. Единствената необходимост е да свържем Spring Boot приложението със съответната база, която смятаме да използваме (в конфигурацията на приложението) и след това имаме възможността да достъпваме и променяме данни с единния интерфейс на ORM.
- **Services** - тук се имплементира същинската бизнес логика на приложението - взаимодействието и обmena на информация между отделните обекти спрямо функционалните изисквания за системата.
- **Controllers** - служи, за да свърже HTTP заявките изпратени от клиентската част (браузъра) със съответния endpoint на сървърната страна, да предаде получените данни към бизнес логиката (съответния Service), те да бъдат правилно обработени и в последствие да върне подходящ отговор към клиента (било то html страница, данни в JSON формат или код за грешка).

5 РЕАЛИЗАЦИЯ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС

При отваряне на системата в браузъра, първият екран, който ни се показва, е екранът за вход на съществуващ потребител (лектор). Ако не е налична регистрация в системата, може да се посети екранът за регистрация, чрез който да се направи профил. След вход в системата виждаме основната страница, която ни показва всички зали, които са налични в базата данни, и при поставяне на мишката върху някоя зала, виждаме повече информация за нея (дали е компютърна, дали има наличен проектор и с колко места разполага). Системата има и екран за резервиране на зала, където избираме коя от наличните зали искаме да резервираме, за коя дата и от колко до колко часа. При натискане на бутона “Резервирай” ни се отваря последния екран на системата, който показва текущият потребител какви резервации на зали има. Разбира се, ако сме си свършили работата в системата, можем да излезем от нея чрез бутона за изход в менюто.

Екран за вход:



Go to Register

Login

Email *

Password *

Login

Екран за регистрация:

[Go to Login](#)

Register

Email *

Username *

Password *

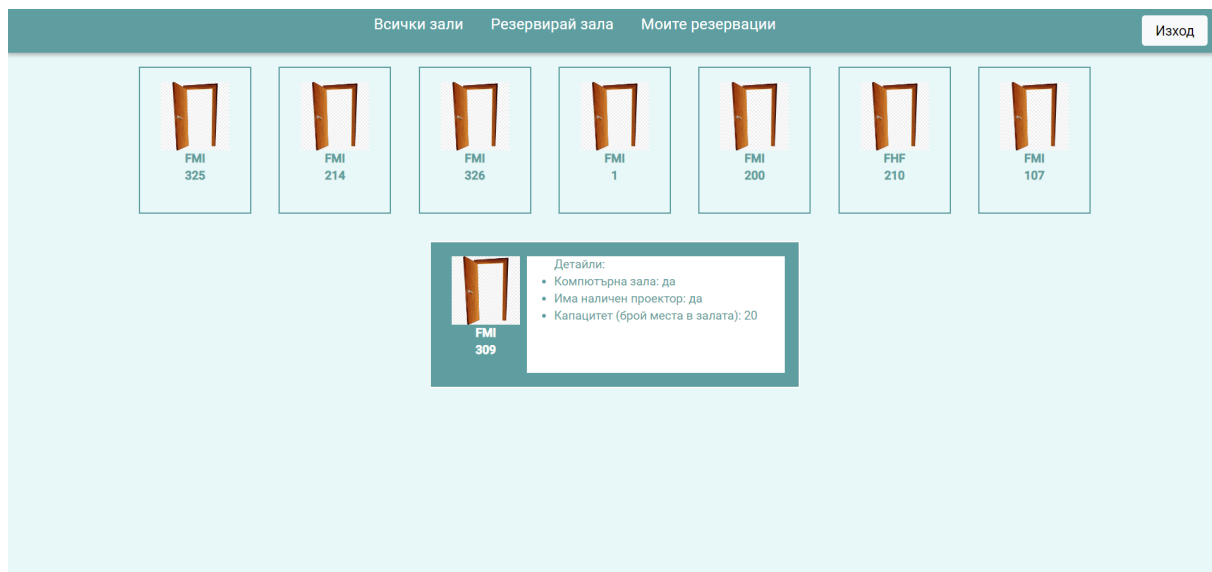
Password Confirmation *

Register

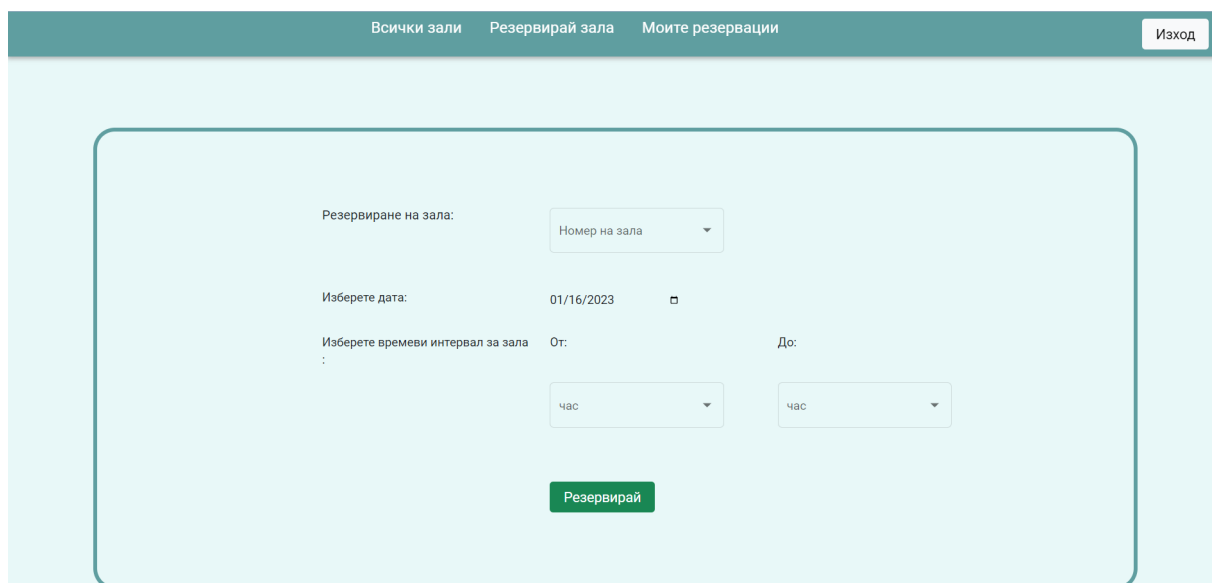
Основен екран, на който се виждат всички зали:



Основен екран, на който мишката е върху конкретна зала и виждаме повече информация за нея:



Екран за резервация на зала:



Екран, показващ всички регистрации на текущия потребител:

Номер на зала	Факултет	Дата на резервиране	Час на резервиране	Действия
325	FMI	16.01.2023	12:00	Изтрий резервация
325	FMI	16.01.2023	11:00	Изтрий резервация
325	FMI	16.01.2023	10:00	Изтрий резервация

6 ВНЕДРЯВАНЕ НА СИСТЕМАТА

- Създадохме нова директория в Github, до която всички имаме достъп. Там качвахме и взимаме промените по кода.
- Използвахме IntelliJ IDEA като среда за разработка на проекта.
- Инсталирахме Java 13.
- Използваме Maven, който се грижи за управлението на библиотеките, от които сме зависими (dependencies), като основната от тях е Spring Boot. Във файла pom.xml се намират тези dependencies, които Maven ни предоставя при билдване на кода.
- Добавихме Angular CLI, който служеше за разработката на фронтенда.
- Инсталирахме Node.js, за да използваме командите от пакета NPM (node package manager)
- Добавихме и MySQL база данни, която Spring използва.
- Системата може да бъде достъпена на локален сървър на адрес: <http://localhost:8080/>

7 РАЗПРЕДЕЛЕНИЕ НА ДЕЙНОСТИТЕ ПО РЕАЛИЗАЦИЯТА

Всички, участващи в разработката на системата сме се ангажирали с имплементиране на задачи, както за backend частта на проекта, така и за frontend частта. Разпределение на задачите:

- Весела Петрова - Съставяне на модели за базата данни, резервиране на зала (backend), изтриване на резервация (backend), основна страница (frontend)
- Надежда Францева - Функционалност за вход и регистрация (backend and frontend)
- Ивана Драгнева - Функционалност за резервиране на зала (frontend), функционалност за получаване на залите, които са резервирани от даден лектор (backend), функционалност за получаване на свободните часове за резервиране на зала (backend)
- Силвия Деянова - функционалност за основната страница (backend), меню (frontend), страница за резервациите, на текущо логнат потребител (frontend)