Курсов проект „Book List“

Дисциплина: Проектиране и интегриране на софтуерни системи

Фаза 3: Реализация на системата

Версия 1.x

|  |  |  |
| --- | --- | --- |
| **Фак. №** | **Име на студент** | **Секция от документа** |
| **855271** | Влад Тимофеев | Всички секции |
| **61880** | Даниел Павлов | Всички секции |

Януари, 2019

Съдържание

[1 Въведение 3](#_Toc374967428)

[1.1 Цел 3](#_Toc374967429)

[1.2 Резюме 3](#_Toc374967430)

[1.3 Дефиниции и акроними 3](#_Toc374967431)

[2 Използвани технологии 4](#_Toc374967432)

[3 Реализация на базата от данни 5](#_Toc374967433)

[4 Реализация на бизнес логиката 6](#_Toc374967434)

[5 Реализация на потребителския интерфейс 7](#_Toc374967435)

[6 Внедряване на системата 8](#_Toc374967436)

[7 Разпределение на дейностите по реализацията 9](#_Toc374967437)

[8 Приложения 10](#_Toc374967438)

# Въведение

## Цел

*Опишете на целта на документа.*

Целта на документа е да обясни начина на реализация на системата и да предостави информация за внедряване на нея.

## Резюме

*Създайте резюме на документа и опишете структурата му.*

*// TODO*

## Дефиниции и акроними

*Дефинирайте всички термини, понятия и акроними, използвани в документа.*

* БД – База данни
* API – application programming interface
* MVC – Model View Controller pattern

# Използвани технологии

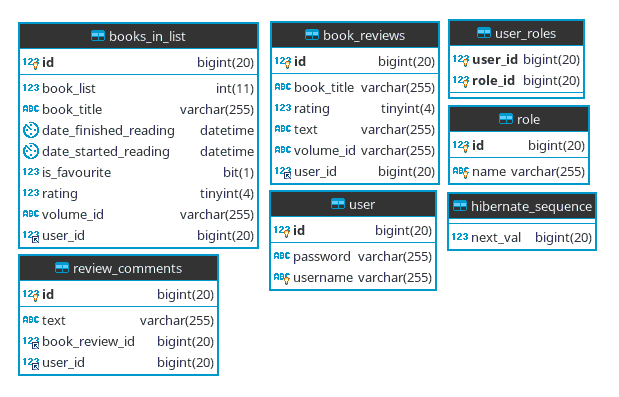
*Опишете технологиите и програмните средства, използвани за реализация на системата.*

* Средства за разработване:
  + git за version control & споделена работа
  + Insomnia за тестване на API
  + DBeaver за анализ на БД
  + VS Code / neovim за разработване на фронтенд
  + Intellij idea за разработване на сървърна част на проекта
* MVC Design Pattern – системата е базирана на този паттерн
* Уеб-клиент (фронтенд, view от mvc) :
  + Yarn, npm за package management
  + Angular + typescript за архитектура и база за разработване на потребителски интерфейс
  + RxJS за т.н. рекативно програмиране за асинхронни, callback-based заявки до сървърна част чрез http
  + Clarity за разработване на потребителски интерфейс
  + Node.js & webpack за хостване на фронтенд
  + SockJS за работа със сокети от страна на уеб-клиента
* Сървърна част (model & controller от mvc):
  + Maven за package management
  + Spring boot и JSON за създаване на JSON API
  + Google JAVA API за Достъп до Google books API
  + Websockets за работа със сокети от страна на сървъра
  + JWT Token за авторизация на потребители

# Реализация на базата от данни

*Опишете типа и структурата на базата от данни, ако има такава.*

(*/docs/db\_compact.png*)

Няколко коментари:

* Book\_in\_List – модел на книга която присътства в някой от списъците на потребителя.
  + Поле volume\_id е идентификатора на volume от Google Books API
  + is\_favourite – дали книгата е favourite за потребител
  + book\_list – число определяща списъка в който се намира книга
  + rating – рейтинг на книга от потребител който я е добавил в списъка
  + user\_id – собственник на книга в списък
* Book\_review - модел за ревю на книга
  + Поле volume\_id е идентификатора на volume от Google Books API
  + user\_id - автора на ревю
  + rating - оценка на автора на ревю за дадена книга
  + text – съдържание на ревю
* Review\_comment – модел за коментари на едно ревю
  + book\_review\_id – идентификатора на ревю към което се отнася даден коментар
  + user\_id – автора на коментар
  + text – съдържание на коментар
* User – модел за потребител на система
  + username – потребителско име
  + password – hashed password
* Role – модел за роли на ползватели
  + name – име на роля (напр. „ROLE\_USER“)
* User Role – модел за съотвествие между ползвател и роля.

# Реализация на бизнес логиката

*Опишете реализацията на бизнес логиката на системата с избраните технологии и програмни средства.*

*Тази секция може да се структурира от гледна точка на модулите на системата или въз основа на реализираните класове и интерфейси.*

# Реализация на потребителския интерфейс

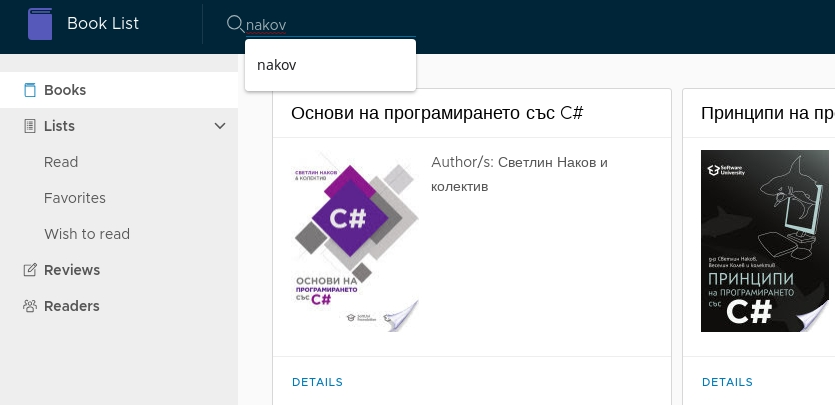
*Опишете реализираните елементи на потребителския интерфейс. Приложете примерни екранни форми.*

Потребителя използва системата чрез интерактивен уеб клиент под формата на SPA (single page application).

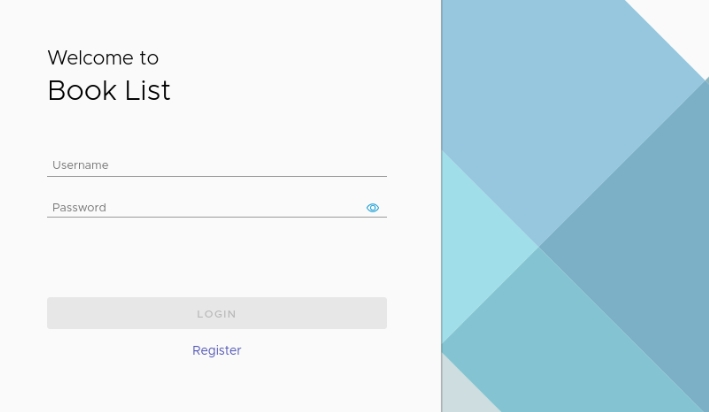
Ще опишем ключеви елементи на потребителски интерфейс:

*// TODO: insert screenshots*

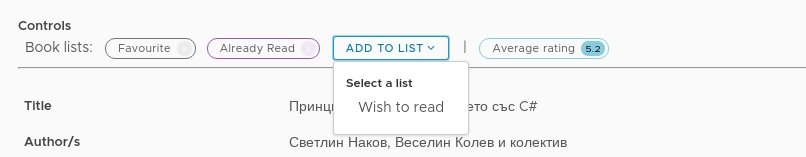
1. *Главна страница – страница от която всеки потребител почва работа с система, има основна навигация, както и връзки към страници за авторизация*
2. *Форма за търсене на книги и отобразяване на резултати от търсене – в главния header на потребителски интерфейс ползвател може да търси книги от база google books.*

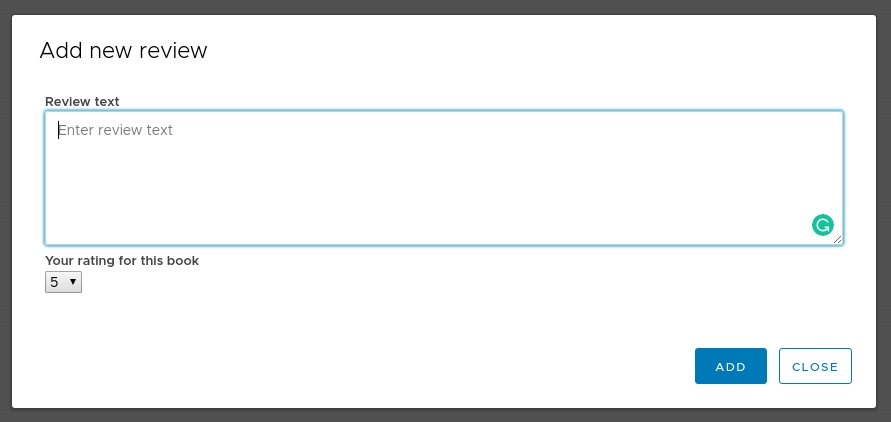
**

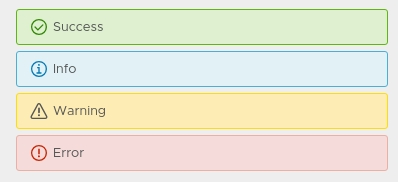
1. *Login Страница – страница на която неавторизиран ползвател може да се регестрира или да влезне със съществуващ аккаунт*



1. *Страница на детайлна информация относно книга – страница на която ползвател може да разгледа детайлна информация относно книгата, нейни ревюта и среден рейтинг. // TODO: add image :\*
2. *Интерфейс за добавяне книгата в списъците и отобразяване на среден рейтинг от ревюта на дадена книга.*

**

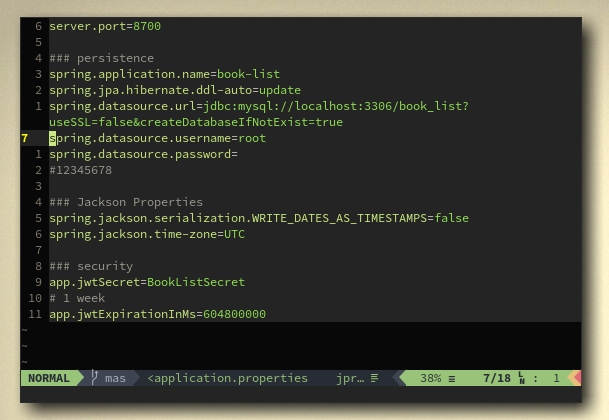
1. *Форма за добавяне на ревю*
2. *Информиране на ползвателя относно възникнали грешки*



# Внедряване на системата

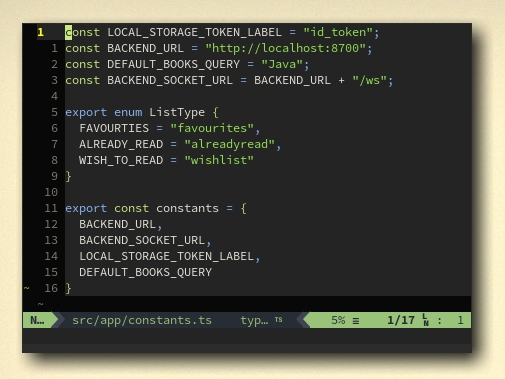
*Опишете изискванията и последователността от действия за внедряване на системата.*

1. Необходим софтуер за внедряване:
   1. git, mysql, java, maven, node.js, npm/yarn
2. Клониране на репозиторий от github: ***`git clone*** [***https://github.com/Pavlov7/book-list***](https://github.com/Pavlov7/book-list)***`***.
3. Внедряване на сървърна част (в **/book-service/** директория):
   1. Трябва да имате инсталирана MySQL
   2. Инсталирайте зависимости: **`mvn install`**.
   3. Редактирайте конфигурация: **`vim src/main/resources/application.properties`**:
      * Сменете данни БД, secret и expiration за JWT токен, порт на който ще слуша сървъра



* 1. Пуснете сървъра: **`mvn spring-boot:run`**.
  2. Отворете mysql конзола и в таблица **role** създайте нов ред с id=1 и name=“ROLE\_USER“
  3. Рестартирайте сървъра с команда **`mvn spring-boot:run`**.

1. Внедряване на клиентска част (в директория **/ui**)
   1. Инсталирайте зависимости: **`npm install`** или **`yarn`**.
   2. Редактирайте конфигурация: **`vim src/app/constants.ts`**.
      * Сменете BACKEND\_URL със URL на пусната сървърна част.



* 1. Стартирайте клиентски сървър с команда **`npm start`** или **`yarn start`**.

1. С помоща на софтуер за process management, който ползвате (например pm2) рестартирайте и двата сървъра
2. С помоща на nginx / apache направете ProxyPass към **4200** порт.

# Разпределение на дейностите по реализацията

*Опишете дейностите, които всеки член на екипа е извършил при реализацията на системата.*

* Първоначален скелетон за системата - Даниел
* Сокет връзка – Влад
* Всичко останало сме правили заедно

# Приложения

*Тази секция не е задължителна и се използва при необходимост.*

*Приложенията се включват директно или се реферират.*

*[Google Books API](*[*https://developers.google.com/books/docs/overview?hl=bg*](https://developers.google.com/books/docs/overview?hl=bg)*)*

*[Git repo](*[*https://github.com/Pavlov7/book-list*](https://github.com/Pavlov7/book-list)*)*

*[Spring Boot](*[*https://spring.io/projects/spring-boot*](https://spring.io/projects/spring-boot)*)*

*[Angular](*[*https://angular.io/*](https://angular.io/)*)*