Курсов проект

Дисциплина: Проектиране и интегриране на софтуерни системи

Фаза 2: Анализ на изискванията и проектиране на системата

Версия 1.x

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

Ноември, 2018

Съдържание

[Съдържание 2](#_Toc531471934)

[1 ВЪВЕДЕНИЕ 2](#_Toc531471935)

[1.1 Цел 2](#_Toc531471936)

[1.2 Обхват 3](#_Toc531471937)

[1.3 Резюме 3](#_Toc531471938)

[2 АНАЛИЗ НА ИЗИСКВАНИЯТА 4](#_Toc531471939)

[2.1 Функционални изисквания 4](#_Toc531471940)

[2.2 Нефункционални изисквания 7](#_Toc531471941)

[2.2.1 Изисквания към потребителския интерфейс 7](#_Toc531471942)

[2.2.2 Производителност 8](#_Toc531471943)

[2.2.3 Наличност 8](#_Toc531471944)

[2.2.4 Сигурност 8](#_Toc531471945)

[2.2.5 Съответствие със стандарти 9](#_Toc531471946)

[3 СИСТЕМНА АРХИТЕКТУРА 10](#_Toc531471947)

[3.1 Архитектурно проектиране 10](#_Toc531471948)

[3.2 Описание на декомпозицията 11](#_Toc531471949)

[4 ДАННОВО ПРОЕКТИРАНЕ 12](#_Toc531471950)

[4.1 Описание на данните 12](#_Toc531471951)

[4.2 Речник на данните 12](#_Toc531471952)

[5 КОМПОНЕНТНО ПРОЕКТИРАНЕ 13](#_Toc531471953)

[6 ПРОЕКТИРАНЕ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС 13](#_Toc531471954)

[6.1 Обобщение на потребителския интерфейс 13](#_Toc531471955)

[6.2 Екранни изображения 14](#_Toc531471956)

[7 МАТРИЦА НА ИЗИСКВАНИЯТА 14](#_Toc531471957)

[8 ПРИЛОЖЕНИЯ 14](#_Toc531471958)

# ВЪВЕДЕНИЕ

## Цел

Настоящият документ описва дизайна и архитектурата на система за четене и менажиране на книги, състояща се от няколко интегрирани подсистеми.

## Обхват

Сървис за онлайн четене, правене на ревюта и оценяване на книги от библиотека с книги. Системата ще обеденява читатели от целия свят и трябва да дава възможност за колективно анализиране на съдържание на книги, съответно подобрявайки резултат на четене на книга чрез комуникация с други читаещи. Освен това, използвайки база на оценки и ревюта на книги по категории, сървиса ще подпомага на хора със избиране на книги за четене, спестявайки време за търсене на най-добри варианти.

## Резюме

Документа описва системата от високо ниво. Представен е анализ на изисванията към сървисите и към потребителския интерфейс. Предоставени са диаграми на архитекурата и данните в системата. Изобразен е и част от потребителския интефейс в много начален стадий.

# АНАЛИЗ НА ИЗИСКВАНИЯТА

## Функционални изисквания

Примерен формат на таблица за описание на функционални изисквания.

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Изискване | Коментар | Приоритет |
| FR\_01 | Всеки потребител задължително трябва да има възможност за търснене и разглеждане на книги от библиотека | Независимо дали е регестриран потребителя трябва да има тази възможност | 1 |
| FR\_02 | Всеки потребител задължително трябва да има възможност за разглеждане на коментари и оценки на всяка книга | Независимо дали е регестриран потребителя трябва да има тази възможност | 1 |
| FR\_03 | Всеки нерегистриран потребител задължително трябва да има възможност за регистрация в системата зада стане пълномощен ползвател на нея | Независимо дали е регестриран потребителя трябва да има тази възможност | 1 |
| FR\_04 | Всеки потребител задължително трябва да има възможност за разлеждане описание на разликите във функционалността на системата за регистрирани и нерегистрирани потребители | Например едно от преимуществата на регестриран потребител е да има възможност за коментиране на книга и участване в обсъжданията. | 1 |
| FR\_05 | Регистриран потребител задължително трябва да има възможност за съставяне на списък от Избрани/Харесани книги | Списъка трябва да е съставен от уникални елементи и да има възможност за редактиране на съдържанието му (добавяне, махане на елементи) | 1 |
| FR\_06 | Регистриран потребител задължително трябва да има възможност за съставяне на списък от Прочетени книги | Списъка трябва да е съставен от уникални елементи и да има възможност за редактиране на съдържанието му (добавяне, махане на елементи) | 2 |
| FR\_07 | Регистриран потребител задължително трябва да има възможност за съставяне на списък от книги които иска да прочете | Списъка трябва да е съставен от уникални елементи и не може да съдърджа книги от списъка на „Прочетени книги“ | 2 |
| FR\_08 | Регистриран потребител задължително трябва да има възможност за публикуване на ревю за дадена книга | Всяко ревю се състои от оценка на книга и съдържание на ревю | 2 |
| FR\_09 | Регистриран потребител препоръчително трябва да има възможност за редактиране на ревю публикувано от него за дадена книга |  | 3 |
| FR\_10 | Ревю задължително трябва да съдържа оценка на дадена книга от ревюиращия | Оценка е число между 1 и 10 което описва колко е харесал дадена книга | 4 |
| FR\_11 | Един потребител не може да има повече от едно ревю за една книга | Ако потребител иска да промени мнението си – той може да редактира предишно публикувано ревю | 3 |
| FR\_12 | Регистриран потребител задължително трябва да има възможност за коментиране на публикувано ревю за дадена книга | Система да позволява ползватели да коментират ревюта | 4 |
| FR\_13 | Регистриран потребител задължително трябва да има възможност за коментиране за книга | Система да позволява ползватели да коментират и обсъждат дадена книга | 3 |
| FR\_14 | Регистриран потребител препоръчително трябва да има възможност за редактиране на негов коментар |  | 5 |
| FR\_15 | Коментари препоръчително трябва да се подновяват в реално време | Ако потребителя публикува един коментар, този коментар трябва да се появи на екрани на други потребители които в момента разгледат същата книга без презареждане на страница | 3 |
| FR\_16 | Препоръчително е системата да предоставя за други разработчици възможност за написане на нов клиент за системата чрез ползване на API. | Например за писане на android клиент | 6 |
| FR\_17 | Системата задължително трябва да ползва google books api за достъп до голяма библиотека на книги |  | 1 |
| FR\_18 | При регистрация на нов потребител, системата задължително трябва да изисква потребителско име, парола и потвърждение за паролата. | Ако пароли не са еднакви да дава грешка при регистрация | 1 |
| FR\_19 | Страница за книга задължително трябва да съдържа: име, автор, кратко описание, ревюта и коментари на ползватели на система |  | 1 |
| FR\_20 | Всяко ревю на книга задължително трябва да съдържа автора, съдържание на ревю, оценка и коментари | Коментари са коментари на други ползватели относно дадено ревю.  Оценка е оценка на книга от ревюиращия | 1 |
| FR\_21 | Всеки коментар необходимо да съдържа автора, съдържание на коментар и отговори. | Отговори са коментари на други ползватели за даден коментар | 2 |
| FR\_22 | При създаване на ново ревю на книга система задължително трябва приема оценка. | Оценка е оценка за дадена книга от ревюиращия. | 1 |
| FR\_23 | Оценка на дадена книга задължително трябва да приема стойност от 1 до 10 | Ако стойността е друга система препоръчително да дава предупреждение и грешка ако ползвател отказва се да поправя стойност | 1 |
| FR\_24 | Едно ревю на книга препоръчително да е ограничено с 2000 символа | Ако стойността е друга система препоръчително да дава предупреждение и грешка ако ползвател отказва се да поправя стойност | 3 |
| FR\_25 | Един коментар на книга препоръчително да е ограничен с 200 символа | Ако стойността е друга система препоръчително да дава предупреждение и грешка ако ползвател отказва се да поправя стойност | 3 |
| FR\_26 | Търсене на книги задължително трябва да примема като параметър текст и полета по които да се търси книга. | Възможни полета са „автор“, „описание“, „оценки“ | 1 |
| FR\_27 | Връщане на списък на книги препоръчително да има възможност да бъде конфигуриран от заявката | Сортиране, лимит, номер на страница на резултати. | 3 |
| FR\_28 | Списъци „Харесано“ и „Прочетени“ необходимо да съдържат уникални за този списък книги | Една книга може да участва в даден списък само вендъж. | 2 |
| FR\_29 | Списъци „Харесано“ и „Прочетени“ препоръчително да са сортирани | Сотрирани по дата на добавяне на книга в даден списък | 3 |
| FR\_30 |  |  |  |

## Нефункционални изисквания

*Изброените нефункционални изисквания са примерни и могат да бъдат модифицирани и/или разширени.*

### Изисквания към потребителския интерфейс

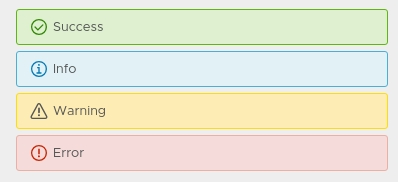
*Опишете характеристиките на потребителския интерфейс – формати, шаблони, структура на менюто, съобщения за грешка и др. съобщения, функционални клавиши и др.*

*Графичен интерфейс е single-page приложение което взаимодейства със сървъра с помощта на rest api.*

*Шаблон на една страница на интерфейса:*

1. Header секция – хоризонтална лента, най-отгоре на страницата, която трябва да включва:
   1. Лого на сайта
   2. Search bar (търсачка) за книги
   3. Бутон „регистрация“ ако потребителя не е автентикиран или потребителското име на потребителя ако е автентикиран
   4. За автентикиран потребител: при кликане на името се опваря dropdown меню с линкове. Единия води до страницата с опциите на потребителя, другия е за log out.
2. Content секция
   1. В ляво да има side menu със следните линкове:
      1. Books – да води към страница с всички книги; има търсене, кратка информация за показаните книги, pagination; да може да покаже най-четените книги, най-високо оценените книги
      2. Lists – група от линкове, които да водят към различни списъци; редовете на списъците да могат да бъдат редактирани от потребителя на същата страница
         1. Read – списък с прочетените книги от потребителя
         2. Favorites – списък с любимите книги на потребителя
         3. Wish to read – списък с книги, които потребителя иска да прочете
      3. Reviews – да води към страница с всички книги; да има търсене, кратка информация за показаните ревюта, pagination; да може да покаже най-високо оценените ревюта
      4. Readers – да води към страница с всички потребители; от тук текущия потребител да може да открие други потребители със сходен вкус като него
3. Footer секция трябва да включва:
   1. Секция за контакти
   2. Връзка към страница с правата за ползване на система
   3. Секция за съобщаване на проблем в система

Съобщения за грешки, информация, успешно действия, предупреждения ще имат съответни цветове и икони и ще се показват отгоре на страницата, под header-a:



### Производителност

* *90% от всички страници за ползватели необходимо да дават отговор за 5с или по-малко.*
* *Препоръчителна поддръжка и спазване на предишни изисквания за поне 10000 едновременни ползватели*
* *При разглеждане на коментари на една книга от N+1 ползватели (страница вече е заредена) , N ползватели необходимо да бъдат уведомени от сървъра относно коментар на друг ползвател не по-късно от 2с от време на публикуване на коментар*
* *Необходима поддръжка и спазване на предишни изисквания за поне 100 едновременни ползватели разглеждащи коментари на една книга*

### Наличност

* Система препоръчително да бъде достъпна за ползване 24/7, допускайки прекъсвания за макс 1 час за deploying на нова, тествана версия на система.

### Сигурност

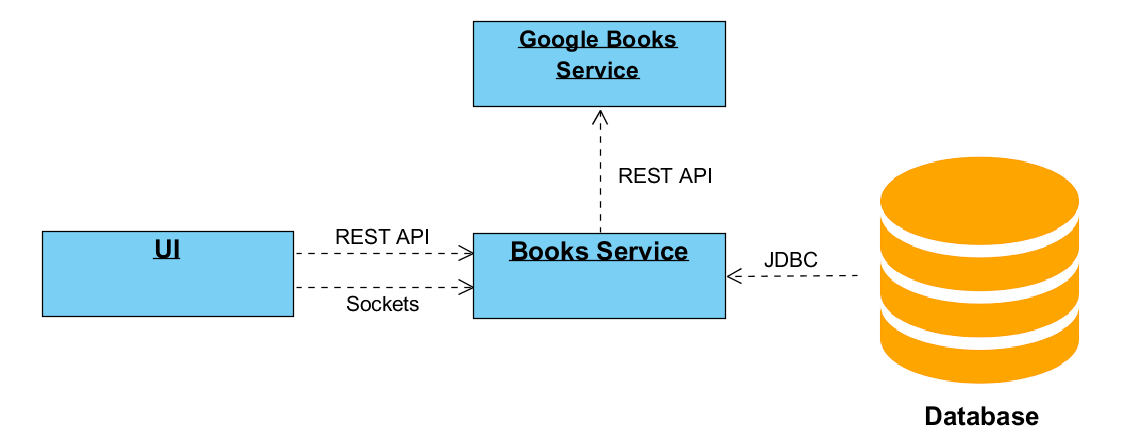
* Системата задължително трябва да има механизъм за контрол на достъпа (идентификация), базиран на потребителско име и парола.
* Задължително ограничаване на достъпа до приватни данни на потребители.
* Системата препоръчително трябва да изисква потвърждение от потребителя при извършването на необратими действия като изтриване на ревю.

### Съответствие със стандарти

* *Задължително и*зползване на система за контрол на версии – Git
* Препоръчително използване на система за tracking на задачи

# СИСТЕМНА АРХИТЕКТУРА

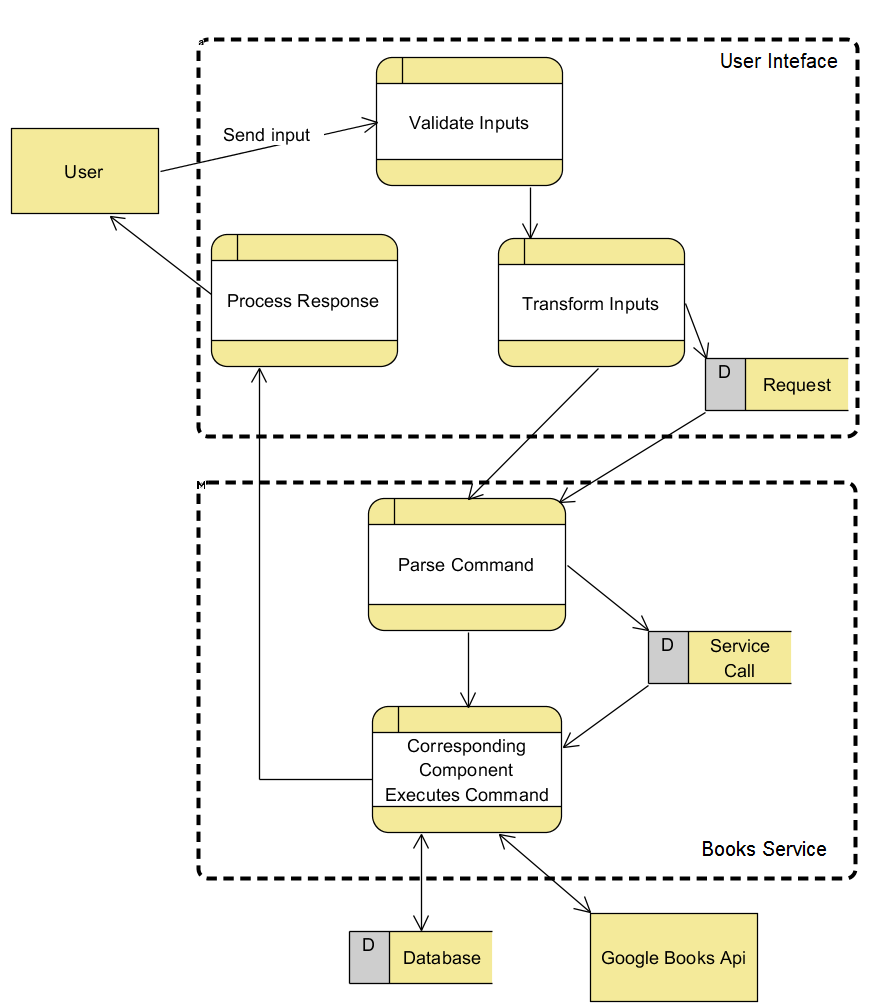
## Архитектурно проектиране



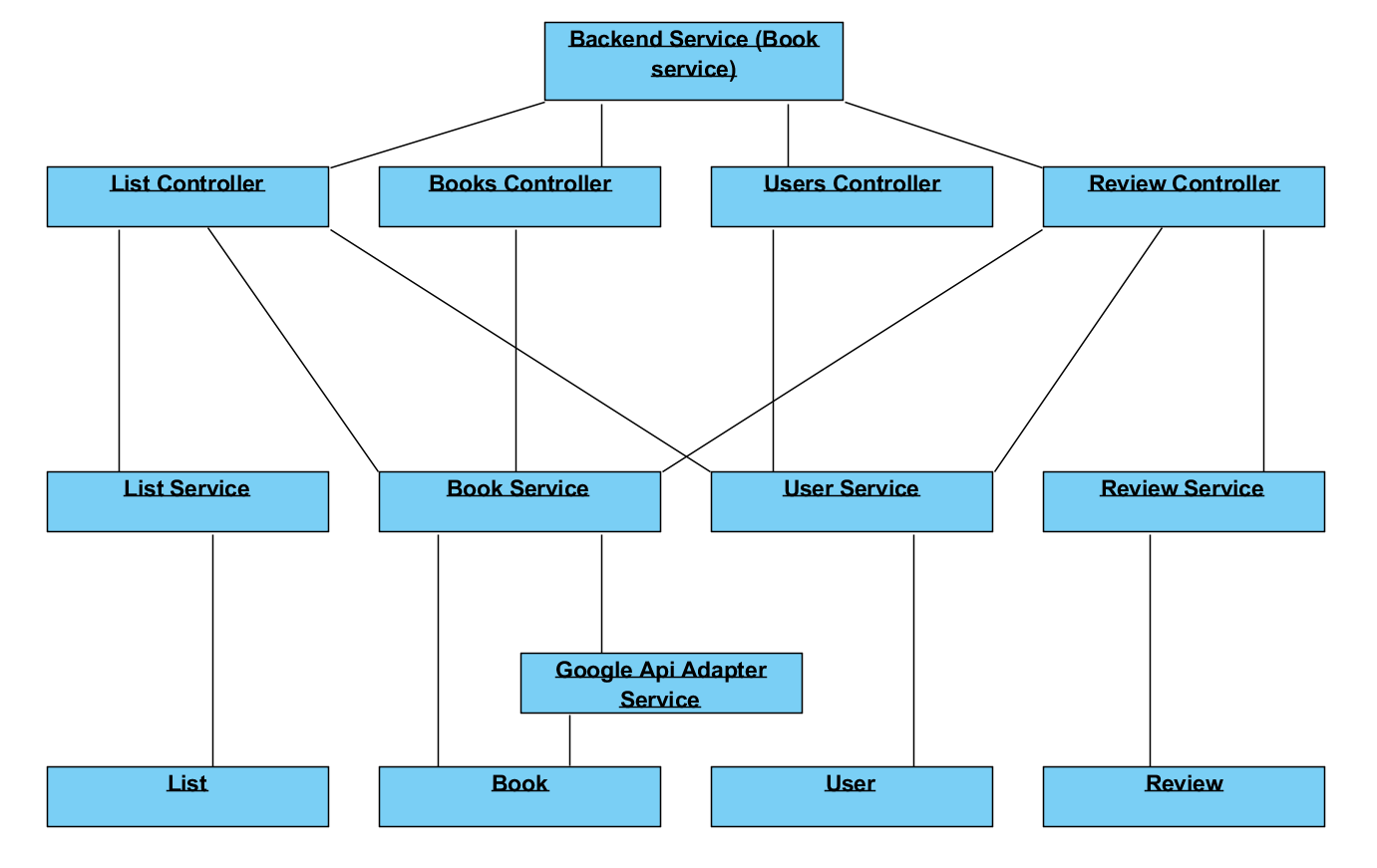
Системата се състои от 3 сървиса – UI, Books service, Google Books service.

* UI – потребителския интерфейс, написан на Angular 2. Той представлява отделен процес, работещ с node.js, който си комуникира с backend-a чрез REST api и socket-и.
* Books Service – Централен сървис/компонент от системата. Занимава се с ауторизация, обработва заявките на потребителя, запазва данните в база данни. Комуникира си с Google books service a чрез google books REST api.
* Google books service – Предоставя голямо количество книги и информация за тях.

## Описание на декомпозицията



Dataflow диаграма на данните. Когато потребител използва системата, неговите команди биват валидирани и обработени от потребителския интерфейс, който праща заявка към сървисите. Те, от своя страна правят допълнителна обработка на данните и ги трансформират до команди. Нужните данни биват събрани от базата или google books api след което се формира отговор за потребителя. Отговора се праща на интерфейса, който го показва на потребителя.



# ДАННОВО ПРОЕКТИРАНЕ

## Описание на данните

Основните данни и съдържание на всяка книга се пазят в базата на google и биват предоставени от google books api. Допълнителната информация се пази в базата на системата. Данните от google api-то и допълнителните данни биват обединени от модела „книга“ в текущата система. Всички останали обекти – потребители, ревюта и списъци – също се пазят в базата на системата. Зарличните видове обекти са тясно свързани, затова базата от данни е от релационен тип.

## Речник на данните

* Потребител
  + Име
  + Парола
* Книга
  + Загравие
  + Автор
  + Дата на издаване
  + Страници
  + Жанр
  + Рейтинг
  + Брой читатели
  + Коментари
  + Съдържание
* Ревю
  + Загравие
  + Оценка
  + Коментари
  + съдържание
* Списък
  + Книги
  + Оценки
  + Тип

# КОМПОНЕНТНО ПРОЕКТИРАНЕ

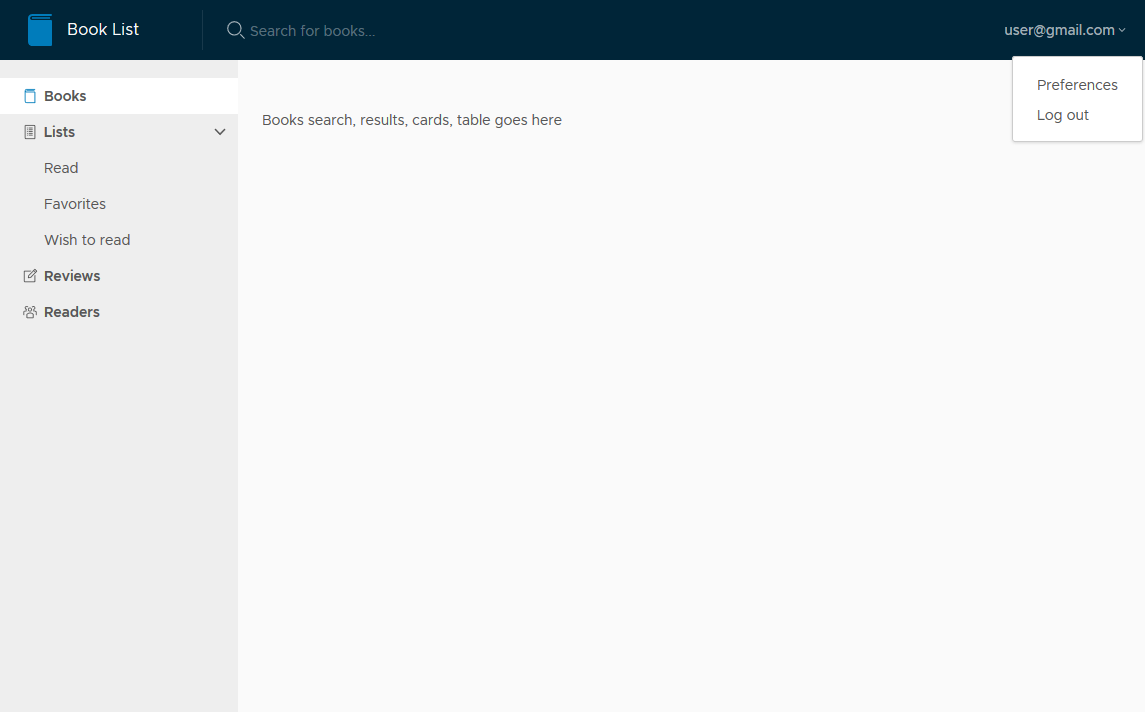
* User controller – компонент обработващ всички видове http заявки свързани с акаунта на потребителя
* Book controller - компонент обработващ всички видове http заявки свързани с книги
* Review Controller - компонент обработващ всички видове http заявки свързани с ревюта
* List Controller - компонент обработващ всички видове http заявки свързани със списъците на потребителите
* User service – сървис за четене/писане на данни за потребителския акаунт от/към базата
* Book service – сървис за четене/писане на данни за книги от/към базата
* Google api adapter service – сървис който извършва комуникацията между компонента Books servie и google books api
* Review Service – сървис за четене/писане на данни за ревюта от/към базата
* List Service – сървис за четене/писане на данни за списъците на потребителя от/към базата

# ПРОЕКТИРАНЕ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС

## Обобщение на потребителския интерфейс

Потребителя ще използва системата чрез интерактивно уеб приложение под формата на SPA (single page application). То ще бъде медиатор между потребителя и сървисите в системата. Данните ще бъдат изобразени във таблици и форми от където потребителя ще може да ги чете, модифицира и изтрива.

## Екранни изображения



# МАТРИЦА НА ИЗИСКВАНИЯТА

|  |  |
| --- | --- |
| *List Controller* | FR\_05, FR\_06, FR\_07, FR\_16, FR\_28, FR\_29 |
| *Books Controller* | FR\_02,FR\_1, FR\_19, FR\_26, FR\_27 |
| Users *Controller* | FR\_01, FR\_03, FR\_04, FR\_07, FR\_16, FR\_18, |
| Review *Controller* | FR\_02, FR\_08, FR\_09, FR\_10, FR\_11, FR\_12, FR\_13, FR\_14, FR\_15, FR\_16, FR\_20, FR\_21, FR\_22, FR\_23, FR\_24, FR\_25, |

# ПРИЛОЖЕНИЯ

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