



Софийски университет „Св. Климент Охридски“

ФМИ

Курсов проект

Fullstack application development

Node JS, Express JS, React, MongoDB

Bookster – веб приложение

Изготвил:

Стефани Богданова Николова

СИ, 3-ти курс, 2-ра група

Фак. № 61945

1. Основни цели и задачи

Системата Bookster предоставя възможност за разглеждане на колекции от книги, списания и статии. Bookster има за цел да запознае потребителите с основните идеи на различните произведения и да събуди у тях интерес да ги прочетат. В системата, потребителите могат да пазят информация за вече прочетените произведения, произведенията, които четат в момента, както и такива, които искат да прочетат в бъдеще.

Bookster дава информация на потребителите за новите и популярните произведения. Потребителите имат възможност да оценяват и коментират различните произведения, както и да търсят произведения по ключови думи. Системата поддържа две различни роли потребители – гост и регистриран потребител. Регистрираните потребители могат да следят своите колекции, да оценяват и коментират. Гост потребителите могат единствено да разглеждат информацията публикувана в системата.

2. Целева група

Основни видове потребители:

- **Анонимен потребител(гост)**
 - Разглежда началната страница на системата
 - Разглежда списъците от книги и списания и статии
 - Разглежда книги, списания и статии групирани по категории
 - Разглежда детайлите за книга, списание или статия
 - Търси произведение по ключови думи
 - Регистрира се
- **Регистриран потребител**

- Редактира информацията в профила си
- Добавя и премахва произведения от колекциите си
- Добавя коментари
- Оценява книги, списания или статии
- Вписва се в системата
- Отписва се от системата

3. Анализ на нуждите

Основни нужди на системата:

- Да предоставя информация за книги, списания и статии
- Да улесни потребителите, които обичат да четат
- Да подпомогне избора на следващо произведение, което потребителят иска да прочете
- Да предоставя на потребителя възможност да пази индивидуални списъци от вече прочетени произведения или такива, които искат да прочетат
- Да дава достъп на потребителите до оцененките, които произведението е получило
- Да запознае потребителите с новите и популярните произведения

4. Изисквания към системата

4.1 Функционални

1. Системата трябва да предоставя достъп на всички потребители до началната страница на сайта.
2. Системата трябва да визуализира списъци от най-новите и най-популярните книги, статии и списания
3. Системата трябва да предоставя възможност за разглеждане на пълните списъци от книги, списания и статии

4. Системата трябва да предоставя възможност за разглеждане на списъци от произведения, групирани по категория
5. Системата трябва да предоставя възможност за детайлно разглеждане на произведение
6. Системата трябва да позволява търсене на произведение по ключова дума
7. Системата трябва да поддържа два вида потребители – гост и регистриран потребител.
8. Системата трябва да позволява на анонимните потребители да разглеждат информацията на сайта, без да я редактират или добавят нова
9. Системата трябва да позволява на гост потребителите да се регистрират чрез попълване на регистрационна форма, задавайки уникално потребителско име
10. Системата трябва да позволява на регистрираните потребители да се вписват в системата, задавайки потребителското си име и парола
11. Системата трябва да позволява на регистрираните потребители да се отписват от системата.
12. Системата трябва да предоставя възможност на регистрираните потребители да оценяват книга, списание или статия
13. Системата трябва да предоставя възможност на регистрираните потребители да коментират книга списание или статия
14. Системата трябва да предоставя възможност на регистрираните потребители да добавят и премахват произведения от следните списъци:
 - Прочетени произведения
 - Произведения, които потребителят чете в момента
 - Произведения за бъдещо четене
15. Системата трябва да дава възможност на регистрираните потребители да редактират информацията в личния си потребителски профил

4.2 Нефункционални

16. Системата трябва да бъде лесна за използване и интуитивна
17. Потребителите на системата трябва да могат да се запознаят с пълната функционалност за не повече от 5 минути
18. Всеки потребител трябва да открие функционалността, която търси за не повече от 1 минута.

19. Системата трябва да забрани достъпа на нерегистрираните потребители до функционалностите характерни само за регистрирани потребители
20. Паролите на потребителите в базата трябва да бъдат в неявен вид (хеширани)
21. Всеки потребител трябва да получава Access Token (JWT), след вписване в системата
22. Системата трябва да отговаря на всички заявки за по-малко от 1 секунда
23. Системата трябва да бъде реализирана с помощта на следните технологии: React JS, Redux JS, Node JS, Express JS
24. Системата трябва да бъде реализирана с помощта на две отделни и независими проложения : клиент и сървър

5. Случаи на употреба

1. Преглед на началната страница

Всеки потребител може да разглежда началната страница на системата. В началната страница, системата визуализира навигационното меню, списък на най-популярните книги и списания, списък на най-новите книги и списания и 3 от популярните статии.

2. Преглед на пълен списък с произведения

Всеки потребител може да разглежда пълните списъци от книги, списания и статии, публикувани в системата.

3. Преглед на списък с произведения, групирани по категория

Всеки потребител може да разглежда списъци от книги, списания и статии, които са групирани по категория.

4. Преглед на детайлите на произведение

Всеки потребител може да разгледа детайлната страница на книга, списание или статия, където открива информация за заглавието, автора, датата на публикуване, рейтинга, описанието и коментарите на произведението.

5. Търсене на произведение по ключова дума

Всеки потребител може да търси произведение по ключова дума, като за ключови думи се считат думи, които се срещат в заглавието или в категорията на произведението.

6. Регистрация

Всеки потребител може да се регистрира в системата, като попълни регистрационна форма, която включва : уникално потребителско име, имейл и парола.

7. Вписване в системата

Всеки регистриран потребител може да се впише в системата чрез попълване на форма за вписване, която включва потребителско име и парола.

8. Добавяне на коментар към произведение

Всеки вписан в системата потребител може да добавя коментар към книга, списание или статия.

9. Оценяване на произведение

Всеки вписан в системата потребител може да дава оценка от 1 до 5 на книга, списание или статия.

10. Редактиране на информация в профила

Всеки вписан в системата потребител може да редактира информацията в личния си профил. Потребителите могат да променят : потребителско име, парола и имейл.

11. Добавяне на произведение към списък

Всеки вписан в системата потребител може да добавя книга, списание или статия към личните си колекции.

12. Премахване на произведение от списък

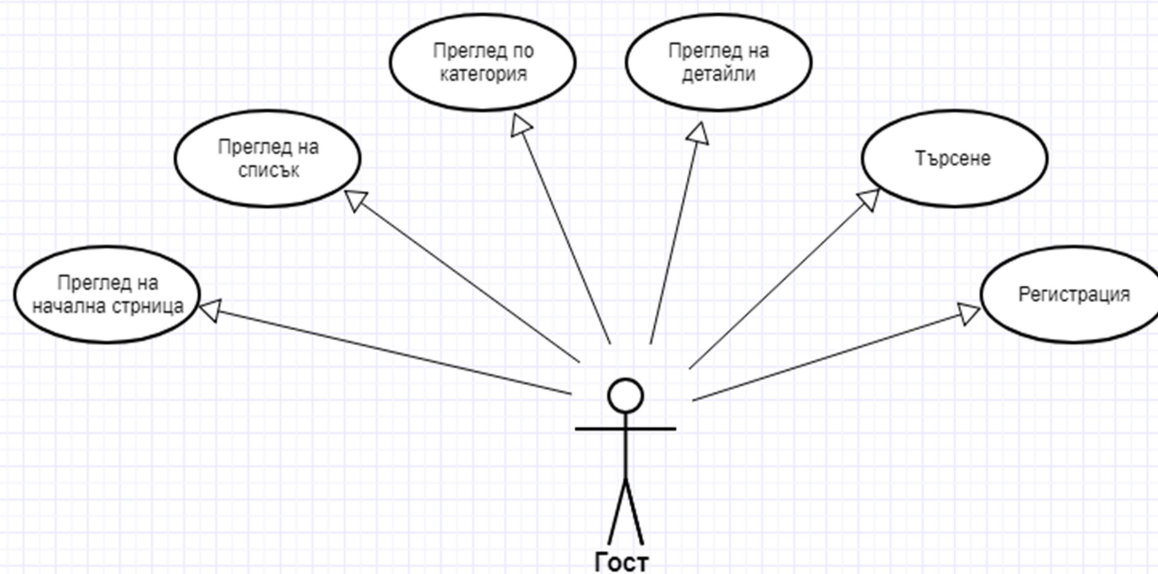
Всеки вписан в системата потребител може да премахва книга, списание или статия от личните си колекции.

13. Отписване от системата

Всеки вписан в системата потребител може да бъде отписан.

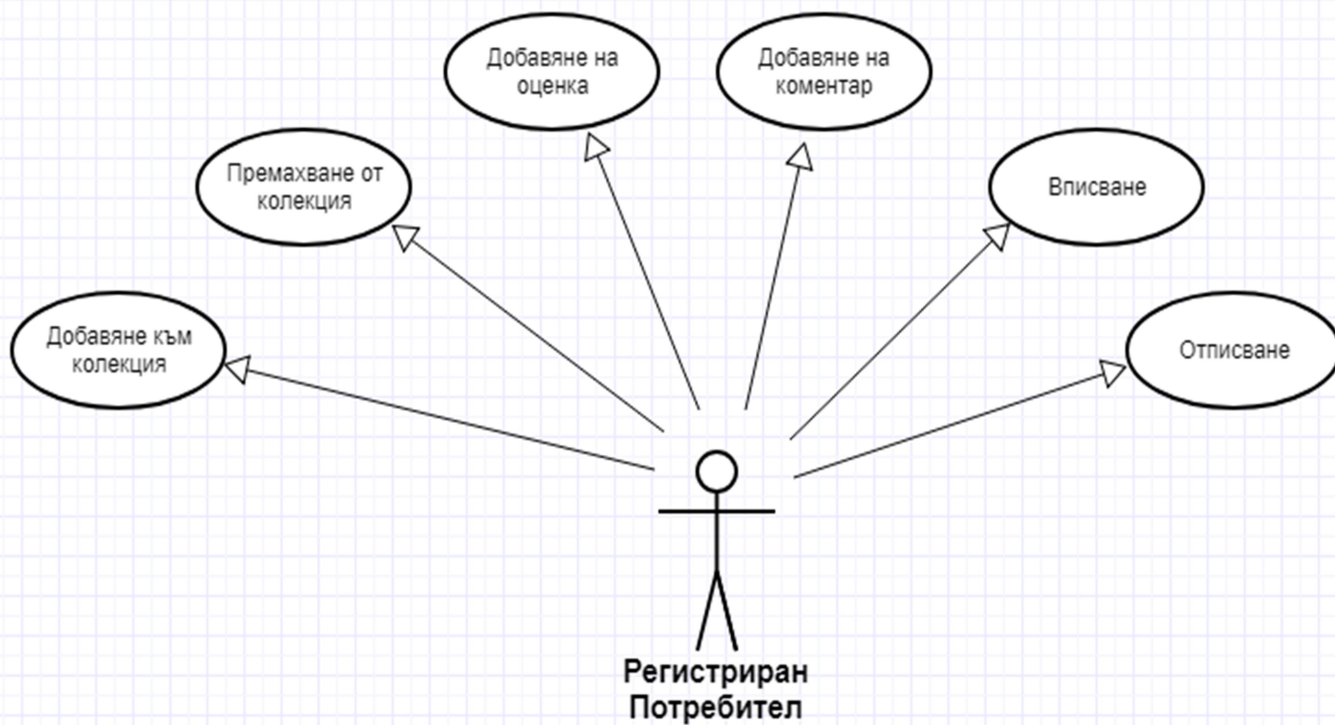
6. Диаграми на потребителските случаи

Диаграма на потребителските случаи на анонимен потребител



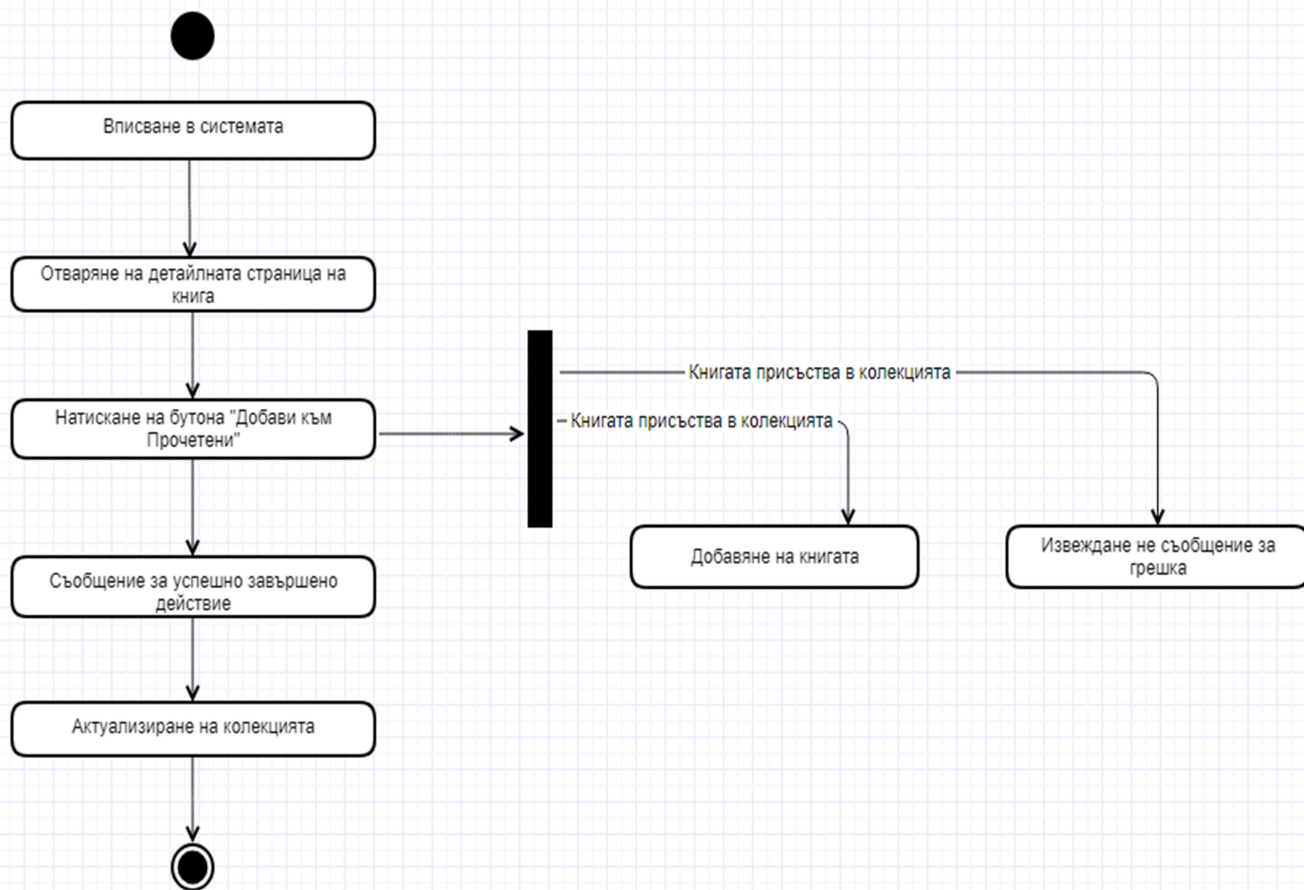
Диаграма на потребителските случаи на регистриран потребител

*Регистрираният потребител има достъп до всички функционалности на гост потребителите

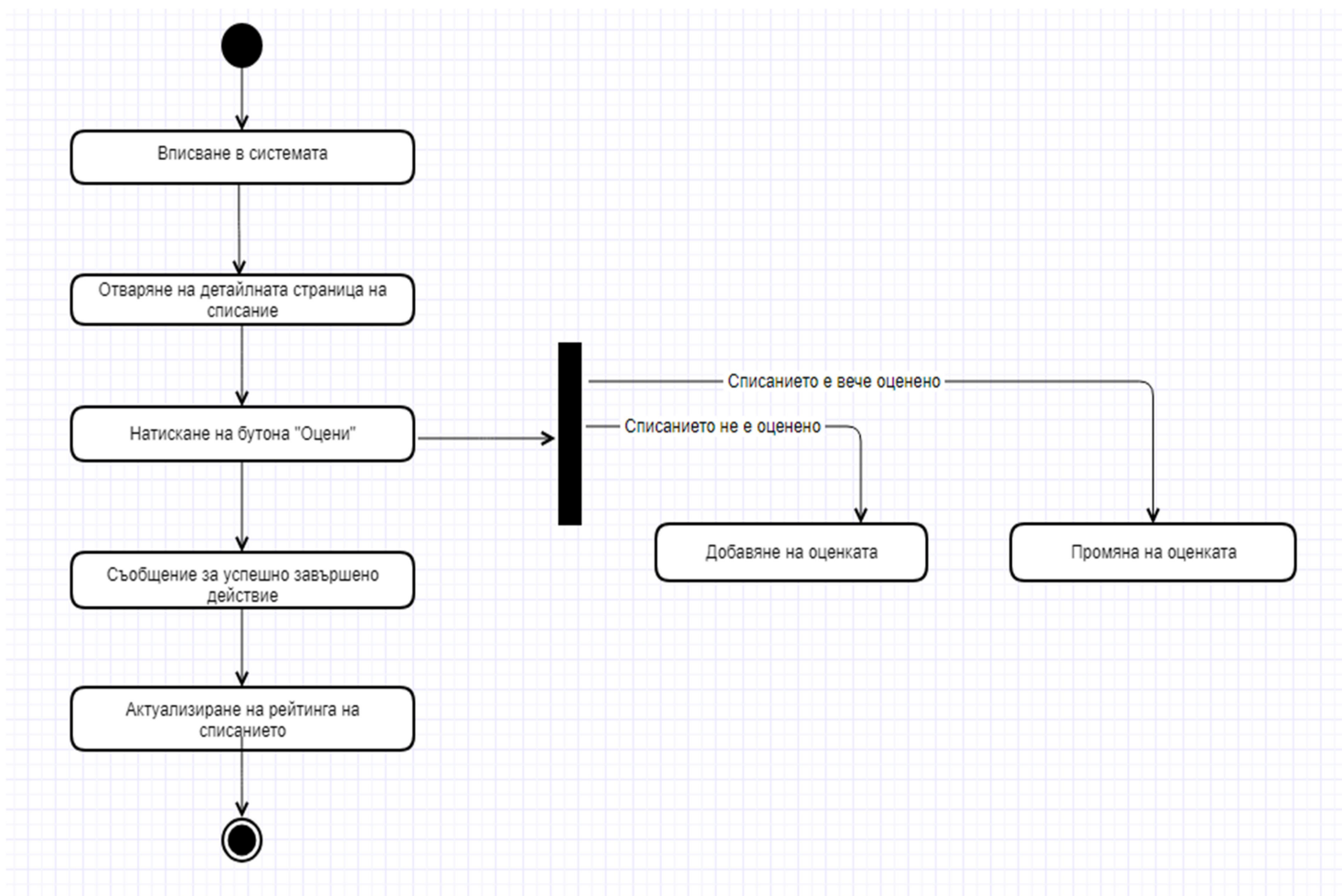


7. Диаграми на активностите

Диаграма на активностите за добавяне на книга към списък с „Прочетени книги“



Диаграма на активностите за оценяване на списание



8. Използвани технологии

- Клиент
 - React JS
 - ReactRouter
 - Redux
 - Semantic UI for React

- ES Lint
- Сървър
 - Node JS
 - Express JS
 - Passport JS
 - MongoDB
 - Mongoose

9. Основни функции и взаимодействия

9.1 Home

Home компонентът отговаря за визуализиране на началната страница. Той използва компонентите FeaturedPosts, NewPosts и PopularPosts.

9.2 Books

Books компонентът отговаря за всички функционалности свързани с книгите. Той взаимодейства с компонентите AllBooksList, BooksByCategory и BookDetails.

9.3 Magazines

Magazines компонентът отговаря за всички функционалности свързани със списанията. Той взаимодейства с компонентите AllMagazinesList, MagazinesByCategory и MagazineDetails.

9.4 Articles

Articles компонентът отговаря за всички функционалности свързани със статиите. Той взаимодейства с компонентите AllArticlesList, ArticlesByCategory и ArticleDetails.

9.5 Profile

UserProfile компонентът отговаря за визуализирането и последващото редактиране на потребителския профил.

9.6 YourCollections

YourCollections компонентът отговаря за визуализиране и редактиране на колекциите на потребителя. Той взаимодейства с компонентите ReadPosts, ReadingPosts, ToReadPosts.

9.7 Search

Search компонентът отговаря за търсенето на публикация по ключова дума.

10. Стратегии за навигация

URL	Изглед
/	Home
/books	AllBooks
/books/:id	BookDetails
/category/books/:id	BooksByCategory
/articles	AllArticles
/articles/:id	ArticleDetails
/category/articles/:id	ArticlesByCategory
/magazines	AllMagazines
/magazines/:id	MagazineDetails
/category/magazines/:id	MagazinesByCategory
/register	Register
/login	Login
/profile/:username	UserProfile
/collections/:username	YourCollections
/search	Search

11. Инсталация

След като приложението е свалено локално и имаме инсталирани:

MongoDb, Node JS и React JS

1. Пускаме MongoDB сървърът.
2. В папката client, отваряме конзола и стартираме приложението с `npm start`
3. В папката server, отваряме конзола и стартираме приложението с `npm start`

12. Оценяване и бъдещо развитие

Потребителският интерфейс на приложението беше тестван от няколко различни по възраст и занимание хора. Те дадоха своите оценки, забележките им бяха взети предвид и интерфейсът беше променен.

Насоки за бъдещо развитие:

- ✓ Различни изгледи за списание, статия или книги
- ✓ Търсене по повече параметри
- ✓ Интеграция с GoodReads уеб приложение
- ✓ Използване на клъстър за запазване на изображенията

13. Използвани материали

[1] <https://reactjs.org/>

[2] <https://react.semantic-ui.com>

[3] <https://expressjs.com/>

[4] <http://mongoosejs.com/>

[5] <http://www.passportjs.org/>