Курсов проект

Дисциплина: Проектиране и интегриране на софтуерни системи

Фаза 2: Анализ на изискванията и проектиране на системата

Версия 1.0

|  |  |  |
| --- | --- | --- |
| **Фак. №** | **Име на студент** | **Секция от документа** |
| **62121** | Даниел Димитров | 3, 5 |
| **62151** | Виктор Христов | 2, 6 |
| **62167** | Иван Чучулски | 1, 4 |

Януари, 2021

Съдържание

[Съдържание 2](#_Toc61814653)

[1 ВЪВЕДЕНИЕ 3](#_Toc61814654)

[1.1 Цел 3](#_Toc61814655)

[1.2 Обхват 3](#_Toc61814656)

[1.3 Резюме 3](#_Toc61814657)

[2 АНАЛИЗ НА ИЗИСКВАНИЯТА 4](#_Toc61814658)

[2.1 Функционални изисквания 4](#_Toc61814659)

[2.2 Нефункционални изисквания 5](#_Toc61814660)

[2.2.1 Изисквания към потребителския интерфейс 5](#_Toc61814661)

[2.2.2 Производителност 5](#_Toc61814662)

[2.2.3 Наличност 6](#_Toc61814663)

[3 СИСТЕМНА АРХИТЕКТУРА 7](#_Toc61814664)

[3.1 Архитектурно проектиране 7](#_Toc61814665)

[3.2 Описание на декомпозицията 8](#_Toc61814666)

[3.3 Обосновка на проектирането 8](#_Toc61814667)

[4 ДАННОВО ПРОЕКТИРАНЕ 9](#_Toc61814668)

[4.1 Описание на данните 9](#_Toc61814669)

[4.2 Речник на данните 9](#_Toc61814670)

[5 КОМПОНЕНТНО ПРОЕКТИРАНЕ 10](#_Toc61814671)

[6 ПРОЕКТИРАНЕ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС 11](#_Toc61814672)

[6.1 Обобщение на потребителския интерфейс 11](#_Toc61814673)

[1) Екран за регистрация и логин 11](#_Toc61814674)

[2) Екран за търсене на книги чрез Google Books API и добавяне към профил 11](#_Toc61814675)

[3) Екран за разглеждане и редактиране на добавени книги 11](#_Toc61814676)

[6.2 Екранни изображения 11](#_Toc61814677)

[1) Екран за регистрация и логин 11](#_Toc61814678)

[2) Екран за търсене на книги чрез Google Books API и добавяне към профил 12](#_Toc61814679)

[3) Екран за разглеждане и редактиране на добавени книг 12](#_Toc61814680)

# ВЪВЕДЕНИЕ

## Цел

Настоящият документ описва изискванията към системата, както и различни аспекти от проектирането на системата, като нейната архитектура и основните потоци на данните.

## Обхват

Проектът има за цел създаване на система за търсене на книги по различни критерии, като всеки потребител има възможност да добавя книги в свои колекции, като например „любими“ и „за по-късно четене“.

Архитектурата на системата е клиент-сървър. Приложението на сървъра реализира функционалностите по управлението на потребителските профили, правенето на заявки към външното API, от което се взима информация за книгите и комуникацията с базата данни, където се запазват колекциите на всеки потребител. Приложението ще поддържа графичен интерфейс.

Клиентското приложение ще предлага възможност за създаване на профил и влизане в системата. След влизане в системата клиентът ще може да прави заявки за получаване на информация за книги по различни категории като заглавие, автор и издател на книгата. Клиентът ще има възможност да запазва книгите, към които има интерес, като информацията за това ще се съхранява в база данни на сървъра.

## Резюме

Документът се разделя на следните секции:

1. **Въведение** – обща информация за проекта и неговата документация
2. **Анализ на Изискванията** – Описание на функционалните и нефункционалните изисквания на системата
3. **Системна Архитектура** – Описание на декомпозицията на системата в подсистеми и взаимодействието между отделните модули
4. **Данново Проектиране** - Описание как данните се съхраняват, организират и обработват
5. **Компонентно Проектиране** – Описание какви са компонентите на системата
6. **Проектиране на Потребителския Интерфейс** – Описание на функционалността на потребителския интерфейс

# АНАЛИЗ НА ИЗИСКВАНИЯТА

## Функционални изисквания

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Изискване | Коментар | Приоритет |
| FR\_01 | Системата задължително трябва да позволява на потребителя да регистрира свой профил. | Потребителя ще регистрира свой профил чрез въвеждане на потребиртелски име и парола и натискане на бутон “Register”. | 1 |
| FR\_02 | Системата задължително трябва да позволява на потребителя да влиза в свой профил. | Потребителя ще влиза в свой профил чрез въвеждане на потребиртелски име и парола и натискане на бутон “Login to Account”. | 1 |
| FR\_03 | Системата задължително трябва да позволява на потребителя да търси книги по зададено име. | Потребителя ще търси книги чрез избиране на категория “Title”, въвеждане на името на книгата в текството поле и натискане на бутона “Search”. | 1 |
| FR\_04 | Препоръчително е системата да позволява на потребителя да търси книги по зададено издателство. | Потребителя ще търси книги чрез избиране на категория “Publisher”, въвеждане на името на издателството в текството поле и натискане на бутона “Search”. | 3 |
| FR\_05 | Препоръчително е системата да позволява на потребителя да търси книги по зададен автор. | Потребителя ще търси книги чрез избиране на категория “Author”, въвеждане на името на издателството в текството поле и натискане на бутона “Search”. | 3 |
| FR\_06 | Системата задължително трябва да позволява на потребителя да добавя книга в своята колекция със избран статус. | Потребителя ще избере книга, получена като резултат от търсене, ще избере статус и ще я добави чрез натискане на бутон “Add Book” | 1 |
| FR\_07 | Системата задължително трябва да позволява на потребителя да прекрати връзката със сървъра. | Потребителя ще затвори сесията със сървъра чрез натискане на бутон “Quit”. | 1 |
| FR\_08 | Необходимо е системата да позволява на потребителя да вижда информация за избрана книга. | Потребителя ще получи информация за описанието и изображение на книгата чрез селектиране на конкретна книга, получена като резултат от търсене. | 2 |
| FR\_09 | Системата задължително трябва да позволява на потребителя да филтрира книгите в своята колекция по зададено име и статус. | Потребителя ще филтрира книги чрез въвеждане на името на книгата в текството поле, избиране на статус и натискане на бутона “Search”. | 1 |
| FR\_10 | Необходимо е системата автоматично да зарежда и представя книгите в колекцията на потребителя | Потребителя трябва да вижда своите книги при отваряне на страницата със своята колекция. | 2 |
| FR\_11 | Системата задължително трябва да позволява на потребителя да премахне филтър за книгите в своята колекция. | Потребителя ще премагне филтър за книги чрез натискане на бутона “Clear Filter”. | 1 |
| FR\_12 | Необходимо е системата да позволява на потребителя да сменя статуса на избрана от него книга в неговата колекция. | Потребителя ще селектира книга от неговата колекция, селектира нов статус и натисне бутон “Change Status” за да приложи промяната. | 2 |
| FR\_13 | Необходимо е системата да позволява на потребителя да премахне избрана от него книга от неговата колекция. | Потребителя ще селектира книга от неговата колекция и натисне бутон “Remove Book” за да я премахне от своята колектция. | 2 |

## Нефункционални изисквания

### Изисквания към потребителския интерфейс

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Изискване | Коментар | Приоритет |
| NRU\_01 | Потребителите задължително трябва лесно да разбират потребителския интерфейс. |  | 1 |
| NRU\_02 | Потребителите задължително трябва бързо да се ориентират откъм функционалностите на потребителския интерфейс. |  | 1 |
| NRU\_03 | Системата задължително трябва да предоставя на потребителите графичен интерфейс. |  | 1 |

### Производителност

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Изискване | Коментар | Приоритет |
| NRP\_01 | Необходимо е системата да изпълнява заявките между клиента и сървъра за по-малко от 1 секунда. | Заявки между клиент и сървър са всички заявки, които не използват Google Books API. | 2 |
| NRP\_02 | Необходимо е системата да изпълнява заявките към Google Books API за по-малко от 2 секунди. |  | 2 |
| NRP\_03 | Системата задължително трябва да може да обработва заявки от 100 различни потребителя без забавяне. |  | 1 |

### Наличност

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Изискване | Коментар | Приоритет |
| NRA\_01 | Задължително е системата да е налична през 95% от времето. |  | 1 |

# СИСТЕМНА АРХИТЕКТУРА

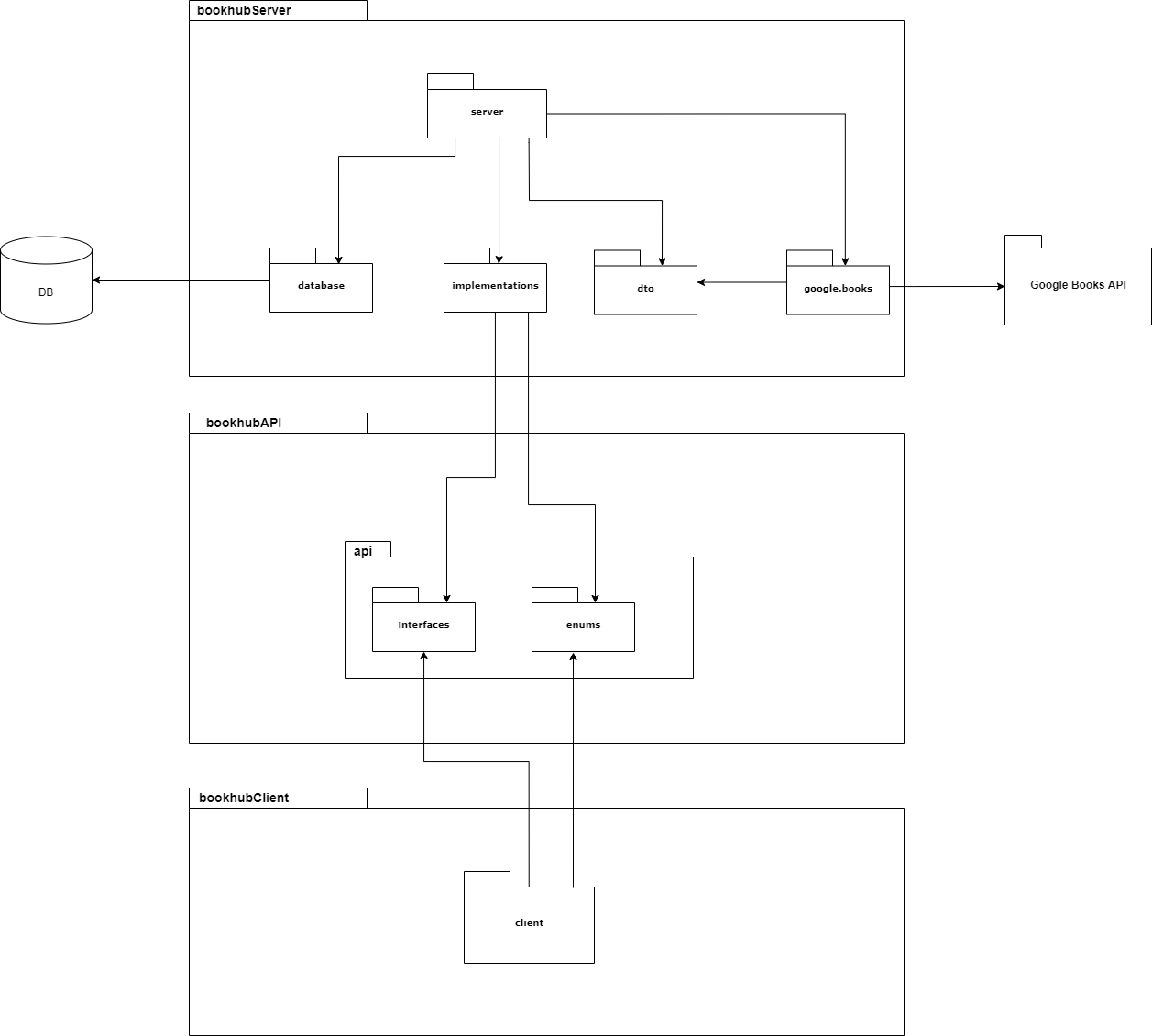
## Архитектурно проектиране

Системата се състои от 4 модула:

1. Модул **Client** – Неговата роля е да изпраща заявки към сървъра и да изобразя получената информация в удобен за потребителя вид. За комуникация със сървъра се използва модул BookhubAPI.
2. Модул **BookhubAPI** – Неговата роля е да съхранява данните и метода, чрез които сървърът и клиента ще комуникират.
3. Модул **Server** – Неговата роля е да получава заявки от клиента и да изпраща отговор обратно към него. За комуникация със сървъра се използва модул BookhubAPI. Сървърът също изпраща различни заявки към базата данни в зависимост от заявката на клиента и обработва получените кортежи.
4. Модул **Database** – Неговата роля е да съхранява информация за регистрираните потребител, за книги в системата и книгите добавени към колекциите на потребителите. Информацията, коята съдържа базата се използва от сървъра, за да се изпълнят клиентските заявки.

## Описание на декомпозицията

Следната диаграма представя декомпозицията на модулите.



## Обосновка на проектирането

Избрахме системата се реализира чрез софтуерния модел клиент-сървър. Този модел позволява висока сигурност, тъй като контролът за достъп до данните се осъществява на едно място, както и консистентност на данните, защото всички клиенти в даден момент достъпват едни и същи данни. Промените в данните се разпространяват много бързо в мрежата. Моделът клиент-сървър позволява на системата да отдели бизнес логиката в сървърния компонент и да остави презентационната логика на клиента. Също така имамe трети модул **BookhubAPI**, който играе ролята на контракт между клиента и сървъра, за да може да комуникират успешно по установени правила.

# ДАННОВО ПРОЕКТИРАНЕ

## Описание на данните

1. Информацията за потребителите ще се съхраняват в базата данни. Тя ще се извлича от сървъра при и трансформира в обекти, които ще се обработват от сървъра. Сървърът ще използва тези данни, за да контролира регистрацията и логина на потребителите, както и представяне на обща статистика за регистрираните потребители.
2. Системата ще извлича книгите чрез Google Books API. Информацията за книгите се разделя на множество от категории, но системата ще се интересува само от следните:

* **id** – уникален идентификатор на всяка книга
* **title** – има на книга
* **authors** – автори на книгата
* **publisher** – издателство на книгата
* **publishedDate** -дата на издаване
* **description** – описание на книгата
* **smallThumbnailLink** – хиперлинк към изображението на книгата

Данните за книгите се съхраняват в базата данни. Сървърът ще извлича тези данни при обработване на клиентските заявки и ще ги запазва в обекти, които след това ще може да върне на клиента по комуникационния канал. Съответно клиента ще може да използва получените обекти, за да визуализира резултатите на потребителския интерфейс и завърши заявките успешно.

1. Системата също ще използва базата данни, за да контролира запазените книги на всеки потребител. Тази информация ще се идентифицира уникално чрез потребителското име и идентификатора на книгата. Данните ще се използват от сървъра, за да се добавят и премахват книги на потребител, както и променяне на статуса на книгите спрямо получената от клиент заявка.

## Речник на данните

1. Данни за потребителите

* **username** – име на потребителя, използва се като текстов низ от символи
* **password** – парола на потребителя, използва се като текстов низ от символи и се хешира за да е неразпознаема

1. Данни за книгите

* **id** – уникален идентификатор на всяка книга, използва се като текстов низ от символи
* **title** – има на книга, използва се като текстов низ от символи
* **authors** – автори на книгата, използва като масив от текстови низове
* **publisher** – издателство на книгата, , използва се като текстов низ от символи
* **publishedDate** -дата на издаване, използва като тип дата
* **description** – описание на книгата, , използва се като текстов низ от символи
* **smallThumbnailLink** – хиперлинк към изображението на книгата, използва се като текстов низ от символи

1. Данни при добавяне на книга в колекция

* **username** – име на потребителя, използва се като текстов низ от символи
* **bookId** – уникален идентификатор на всяка книга, използва се като текстов низ от символи
* **status** – статусът избран от потребителя, използва се като избираем тип с 4 зададени стойности:

“Favourite”, “Want to Read”, “Have Read” и “Currently Reading”

# КОМПОНЕНТНО ПРОЕКТИРАНЕ

1. **Client** – този компонент съдържа графичният потребителски интерфейс за клиента на приложението и се грижи за представяне на данните, получени от комуникацията със сървъра.
2. **Server** – този компонент съдържа бизнес логиката на приложението, необходима за покриване на изискванията към него. Той комуникира с клиентите на приложението, оправя заявки към базата данни, в която се съхранява необходимата информация и изпраща заявки към RESTful услугата Google Books.
3. **BookhubAPI** – този компонент съдържа интерфейси и изброими типове, които са необходими за успешната комуникация между клиентите и сървърът.
4. **Database** – този компонент отговаря за съхраняването на данните, необходими за системата.
5. **Google Books API** – този компонент позволява на сървъра да консумира RESTful услугите, предоставени от Google Books API.

# ПРОЕКТИРАНЕ НА ПОТРЕБИТЕЛСКИЯ ИНТЕРФЕЙС

## Обобщение на потребителския интерфейс

Потребителския интерфейс се разделя на 3 екрана със следните функционалности:

### Екран за регистрация и логин

* Регистрация на потребител
* Влизане в потребителски профил

### Екран за търсене на книги чрез Google Books API и добавяне към профил

* Търсене на книги по зададено име, автор или издателство
* Селектиране на книги и изобразяване на тяхното описание и изображение
* Добавяне на селектирана книга в колекцията на потребителя със зададен статус

### Екран за разглеждане и редактиране на добавени книги

* Разглеждане на книгите в колекцията на потребителя
* Селектиране на книги и изобразяване на тяхното описание и изображение
* Филтриране на книги по име на книга или избран статус
* Сменяне на статус на избрана книга
* Премахване на книга от колекцията на потребителя

## Екранни изображения

### Екран за регистрация и логин

### Екран за търсене на книги чрез Google Books API и добавяне към профил

### Екран за разглеждане и редактиране на добавени книг