Your Books Everywhere!

Analysis and Design Document

Student:Lazăr Denisa Ștefania

**Group:30238**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 3

3. System Architectural Design 3

4. UML Sequence Diagrams 3

5. Class Design 3

6. Data Model 3

7. System Testing 3

8. Bibliography 3

1. Requirements Analysis

# Assignment Specification

It is commonly known that reading has a positive impact on one’s intellectual

development, by enriching both knowledge in certain domains and vocabulary.

The task for this assignment is to build a book management service. It will serve as remote library, so the user can borrow/buy different books from the comfort of his own home, without needing to go to a library that maybe is far apart.

# Functional Requirements

In this section, the functional requirements of the application will be discussed. They are as follows:

* **Account creation 🡪** The app must offer a user the possibility to create an account by completing a form with his/her personal details.
* **Payment plan choosing 🡪** A user can choose between the following payment plans: *weekly*(he/she can pay at the beginning of every week for being able to search and eventually borrow/buy different books), *monthly* or *yearly*.
* **Staff management 🡪** The library will be managed by staff, whose main tasks are to validate payments, borrowing and returns made by the library’s users.
* **Library filtering 🡪** The books belonging to the library can be filtered by release date, author, title and genre.
* **Borrow service 🡪** If a certain book is available, the user can borrow it. Otherwise he/she will have to wait. It is important to mention that for each book there will be a waiting queue. So, if the desired book is not available the user will join the waiting queue.
* **Return service 🡪** Once a user has finished reading the book, he/she will return it. Following this, the book will be assigned to the next user in the waiting queue.
* **Recommendation service 🡪** The app will recommend different books to a user based on previous borrowed books or on specified genre and topic.

# Non-functional Requirements

This library management app has to be:

* **secure** (a potential attacker must not have access to the database or other sensitive information such as other users’ passwords)
* **portable** (it should be available to be used on different types of devices)
* **easily maintainable** (the effort that will be made for different upgrades/bug fixes must remain at a reasonable low level)

2. Use-Case Model

*[Create the use-case diagrams and provide one use-case description (according to the format below).*

*Use-Case description format:*

*Use case: <use case goal>*

*Level: <one of: summary level, user-goal level, sub-function>*

*Primary actor: <a role name for the actor who initiates the use case>*

*Main success scenario: <the steps of the main success scenario from trigger to goal delivery>*

*Extensions: <alternate scenarios of success or failure>*

*]*

This section will show the use case diagrams corresponding to the application and describe one use case in a more detailed manner.

3. System Architectural Design

**3.1 Architectural Pattern Description**

*[Describe briefly the used architectural patterns.]*

**3.2 Diagrams**

*[Create the system’s conceptual architecture; use architectural patterns and describe how they are applied. Create package, component and deployment diagrams]*

4. UML Sequence Diagrams

*[Create a sequence diagram for a relevant scenario.]*

5. Class Design

**5.1 Design Patterns Description**

*[Describe briefly the used design patterns.]*

**5.2 UML Class Diagram**

*[Create the UML Class Diagram and highlight and motivate how the design patterns are used.]*

6. Data Model

*[Present the data models used in the system’s implementation.]*

7. System Testing

*[Present the used testing strategies (unit testing, integration testing, validation testing) and testing methods (data-flow, partitioning, boundary analysis, etc.).]*

8. Bibliography