Your Books Everywhere!

Analysis and Design Document

Student: Bianca Elena Dondas

**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

You are tasked to build a book management service. A user should be able to create an account, choose a payment plan and login to search the book library. Payments can be done via a cash only policy and need to be validated by library staff. The library is managed by staff and can be filtered by release date, author, title, genre. If a book is available a user can add it to your library. If not the user can join a waiting list. Once a book has been read by a user it can be returned via the online library return function. This assigns the book to the next user in the waiting list after validation of the return by library staff. The service also provides users with dynamic recommendations based on latest trends (popular borrowed books) or user defined interests by genre or topic.

# Functional Requirements

* Create account
* Choose payment plan
* Filter by: author, title, genre
* Borrow & return service
* Validate payments and validate return book by staff

# Non-functional Requirement

* Password security: secured with md5 function

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>*

*]*

3. System Architectural Design

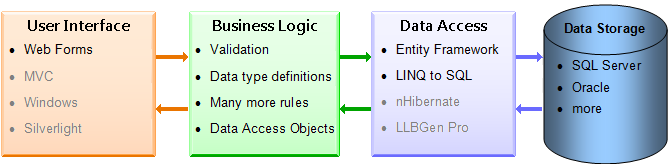
**3.1 Architectural Pattern Description**

**Business logic** is the part of the program that encodes the real-world business rules that determine how data can be created, stored, and changed. It is contrasted with the remainder of the software that might be concerned with lower-level details of managing a database or displaying the user interface, system infrastructure, or generally connecting various parts of the program.

Most commonly this is accomplished using 3 layers:

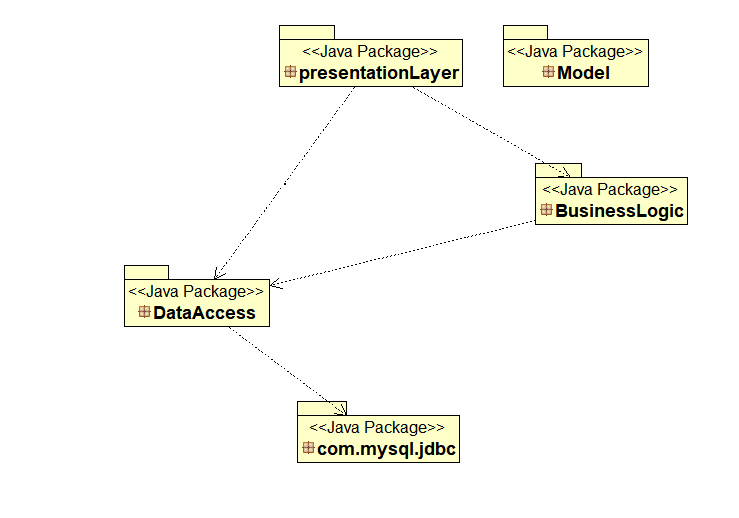
* **User Interface Layer (UI)**: Where all presentation and user interaction takes place. Displays and receives data to and from the user.
* **Business Logic Layer (BLL)**: Application processing. Coordinates data between the UI and DAL.
* **Data Access Layer (DAL)**: Where data management occurs. Typically using a database or web service*.*

Diagram of a layered architecture is presented below:

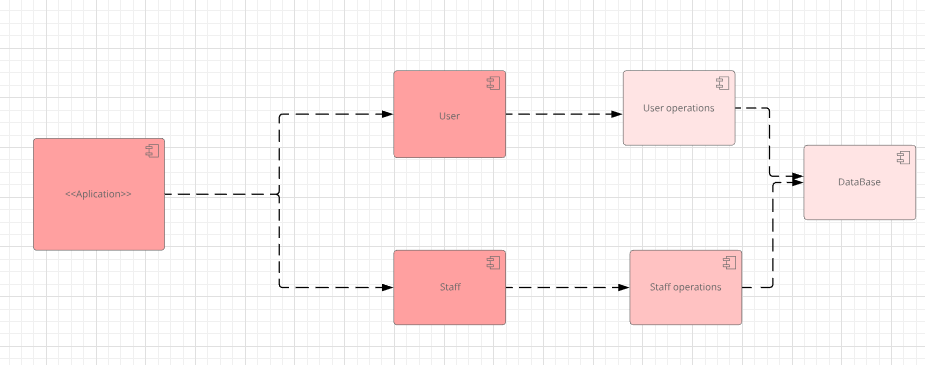


**3.2 Diagrams**

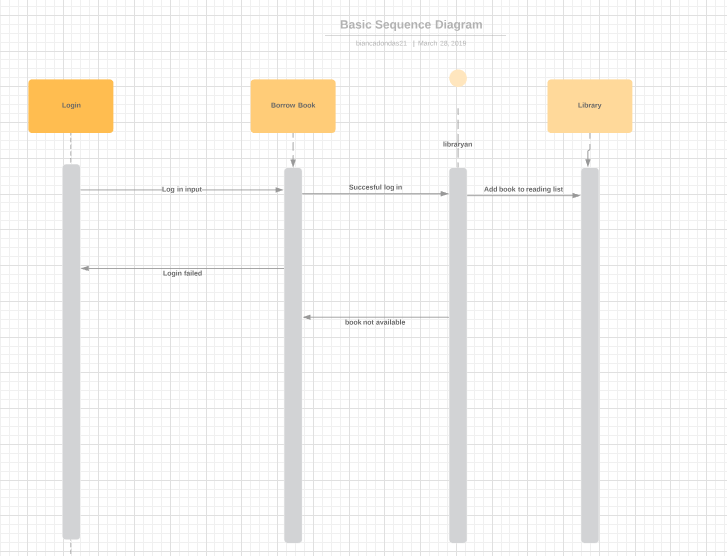
Package diagram:



Component diagram:



4. UML Sequence Diagrams

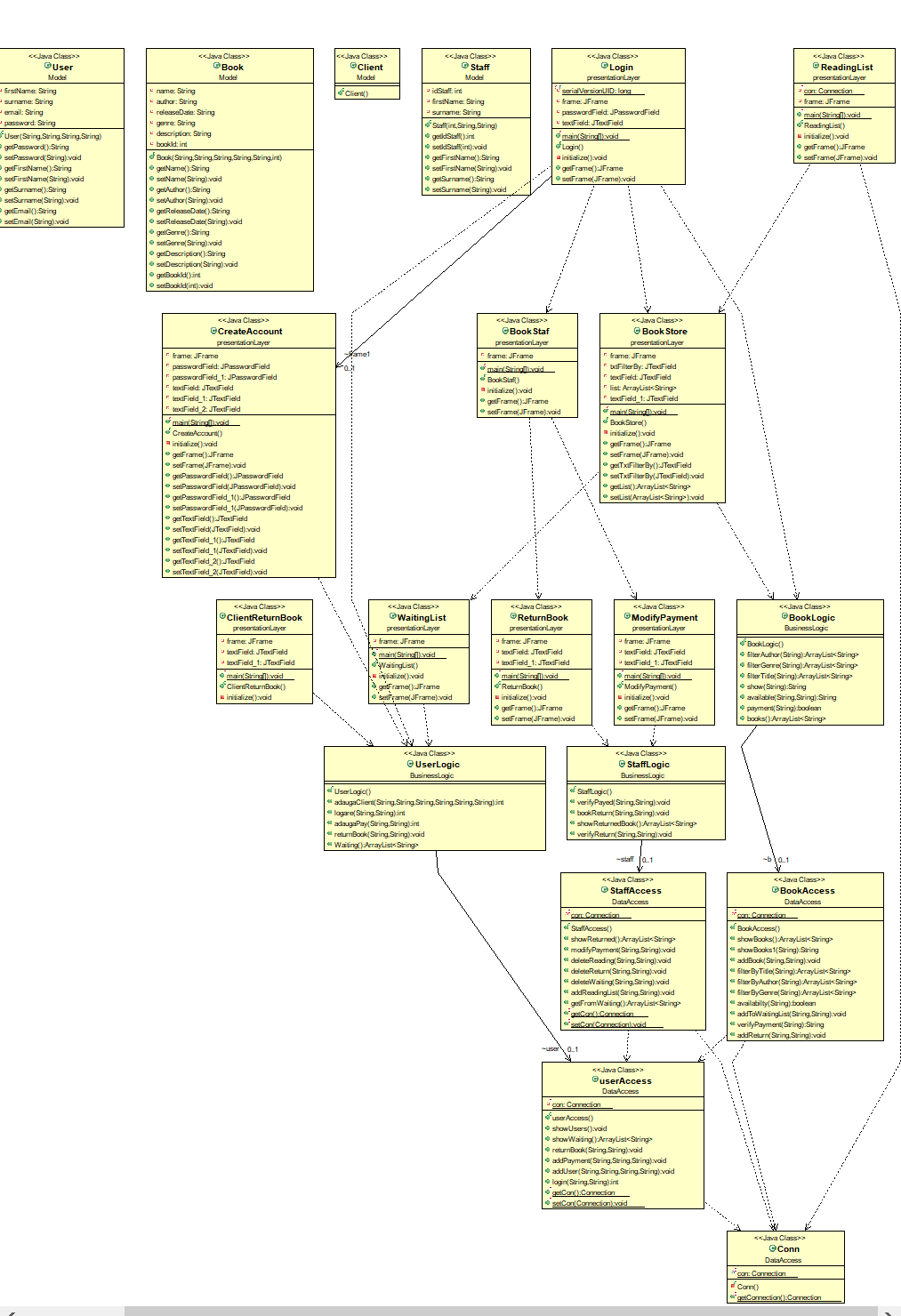


5. Class Design

**5.1 Design Patterns Description**

*[Describe briefly the used design patterns.]*

**5.2 UML Class Diagram**



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