BBM 487 – SOFTWARE ENGINEERING LABORATORY

LIBRARY BOOK LOAN SYSTEM

Architectural Notebook

GROUP II

Özlem DEMİRTAŞ	**21327901**
Umut ÖZTÜRK	**21328394**
Rahmi Berk ŞEFKATLİ	**21427402**

Library Book Loan System
Architectural Notebook

1. Purpose

This document describes the philosophy, decisions, constraints, justifications, significant elements, and any other overarching aspects of the system that shape the design and implementation.

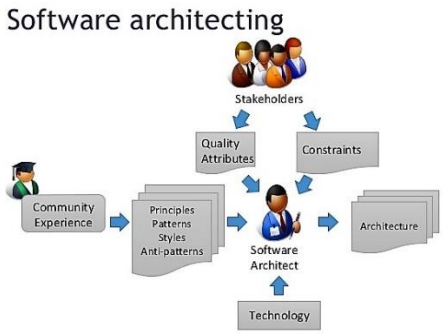

Figure 1

Software Architecture dependencies

2. Architectural goals and philosophy

The main architectural goal is to provide a well-defined architecture that facilitate maintenance and future development.

1. The system is designed as desktop application. For this reason, it will be executed only at library computers.
2. Database will be installed on computer before use.
3. Application's performance will be dependent on the response time of the database.
4. The system will not have a complex architecture. The application will be simple and easy to use.
5. The amount of afford to change the system will not be high.
6. **Transaction**: System's functionalities will be structured using Java and MySQL.
7. **Reliability:** System has to be very reliable due to the importance of data and the damages incorrect or incomplete data can do. Advertisers and premium users should not be disappointed. For this reason, the system will accurately perform member registration, member validation, book transaction and search.

8. **Security:** Proper user authentication will be provided. Users (members and librarians) will be logged with username and password to perform operations.
   Members will be allowed to loan books, view his/her books, pay fine, return the loaned book or check the book's last delivery date.
   Librarians will be allowed to manipulate users and books.
9. **Performance:** The system will respond to the member in less than two seconds from the time of the request submittal.
10. **User Interface:** All interface going to be represented in English. But it will be easily modifiable to extend language support.

3. Assumptions and dependencies

3.1.1 Related software, hardware and operating system

The Library Book Loan System will not be dependent on any specific hardware but the system will be executed under Windows operating system.

3.1.2 End-user characteristics

There are no special requirements for users because of Library Book Loan System will be quite easy in apply. Only knowledge of English (all interface is going to be represented in this language) is required.

3.1.3 Possible and/or probable changes in functionality

New features can be added in future versions of the system considering deadline and the performance of the application.

4. Architecturally significant requirements

For a more in depth description of the architecturally significant requirements, review the system wide requirements document.

1. The Library Book Loan System should be installed on computer before use.
2. Also, database should be installed on computer.
3. The system is designed as a desktop application. For this reason, the program needs to be open.
4. It will be executed under Windows operating system.
5. Member must be logged in if he/she wants to loan book, view his/her books, pay fine, return the loaned book or check the book's last delivery date.
6. Librarian must be logged in if he/she wants to manipulate (add, delete and update) books and users, notify of book being available or issue late fine.
7. If any connection lost or any problem occurs about the system, it will be recovered. System will record them every one hours.

5. Decisions, constraints, and justifications

1. Library Book Loan System will be implemented as desktop application. For this reason, Internet connection is not needed.
2. The user interface will be simple and easy to understand and use. In this way, users can perform the various tasks easily and in an effective way.
3. The error messages will be clear.
4. A small handbook will be prepared to support end-users.
5. Number of librarians and members will be limited considering the performance of the application.
6. Members and books of the library will be manipulated (add, delete and update) by only librarians.
7. Unregistered users will only be able to search books.
8. Other operations will be done after login.
9. The system will respond to the member in less than two seconds from the time of the request submittal. But the system will be allowed to take more time when doing large processing jobs (the response time can be change according to the hardware features).

6. Architectural Mechanisms

Architectural Mechanisms are small parts of an architecture that address a specific important concern, provide a common way of doing something or are a good example of how the architecture behaves and is structured. Architectural Mechanisms are described in terms of structure and behavior.

The Library Book Loan System is designed using MVC (model – view- controller) design pattern.

Short description of MVC

1. Model represents knowledge. A model could be a single object (rather uninteresting), or it could be some structure of objects.
2. View is a (visual) representation of its model. It would ordinarily highlight certain attributes of the model and suppress others. It is thus acting as a presentation filter.
3. Controller is the link between a user and the system. It provides the user with input by arranging for relevant views to present themselves in appropriate places on the screen.

Architectural Mechanism 1

Model: MySQL Server

1. Purpose: To store application's data objects (books and users).
2. Attributes: Attributes of the books (name, author name, barcode, printing date and number of pages) and users (username, name, surname and password)
3. Function: When a new book or user is created, it is stored in an instance of the model.

Architectural Mechanism 2

View: User Interfaces

1. Purpose: To interact users with the application.
2. Attributes: Users can be visitor, member or librarian.
3. Function: Informs users about transactions.

Architectural Mechanism 2

Controller: User Controller and Book Controller

1. Purpose: To update the view when transaction occurs.
2. Attributes: Users can be visitor, member or librarian.
3. Function: Sends commands to model (books and users) to update the models' state.

7. Key abstractions

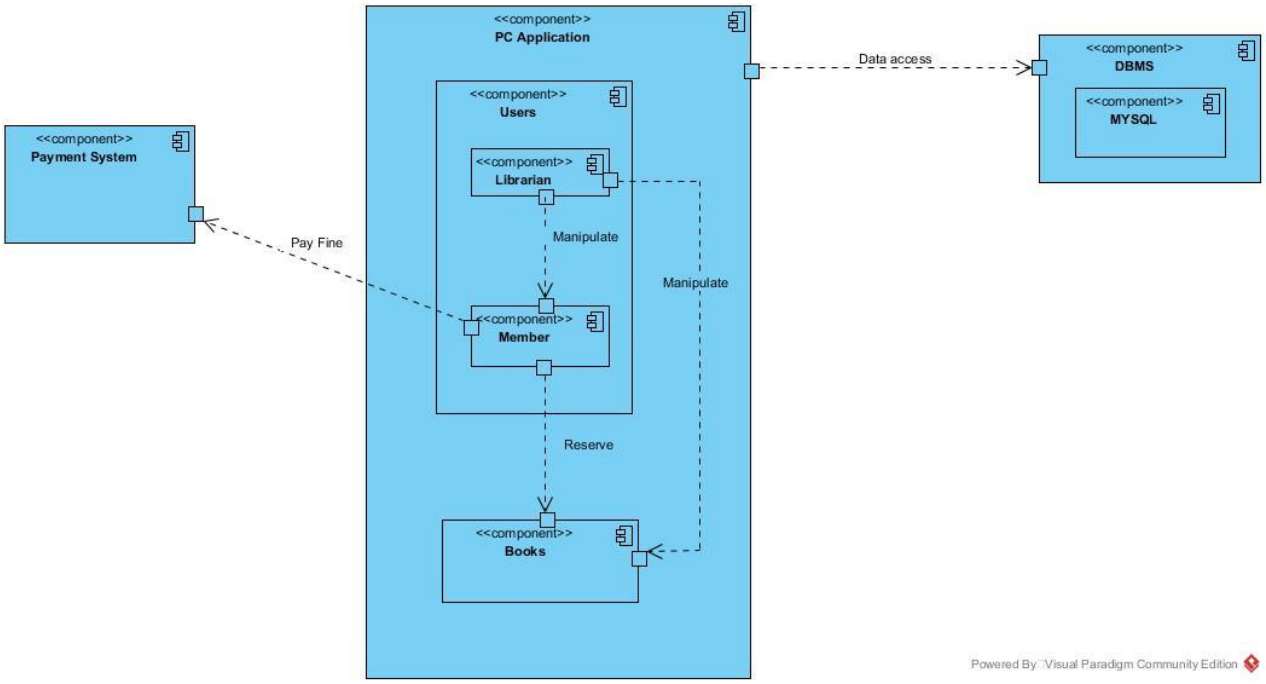

Figure 2 Key components

Key abstractions of the system are shown in Figure 2.

- Application are used by librarians and members.
- Database organizes the book and user information.
- Payment System allows members to pay their fine properly.
- Librarian and member interfaces informs users about their operations.

8. Layers or architectural framework

The Library Book Loan System is designed using Kruchten's 4 + 1 architectural view model. We think that this framework is suitable for our project because it allows to address separately the concerns of the various "stakeholders" of the architecture: end-users, developers, system engineers, project managers etc. and to handle separately the functional and nonfunctional requirements.

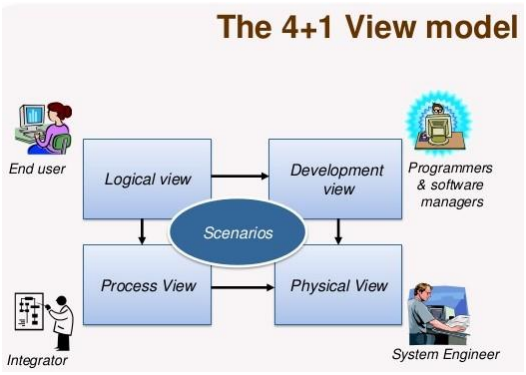

Figure 3

Kruchten's 4 + 1 architectural view model

The system's architectural design pattern is The Model-View-Controller (MVC). Model is a database which stores the information of books and users, view is the user interface and controller is a link between the model and view (user controller and book controller).

9. Architectural views

Logical:

Package Structure: As shown in Figure 4, there exist some fundamental portions in our system. Librarian manipulates (add, delete, update) books and users. Member pays fine, searches and reserves books. Control, sends commands to librarian and member, and updates the books and users' states. GUI, informs users about transactions and DBMS stores book and user information.

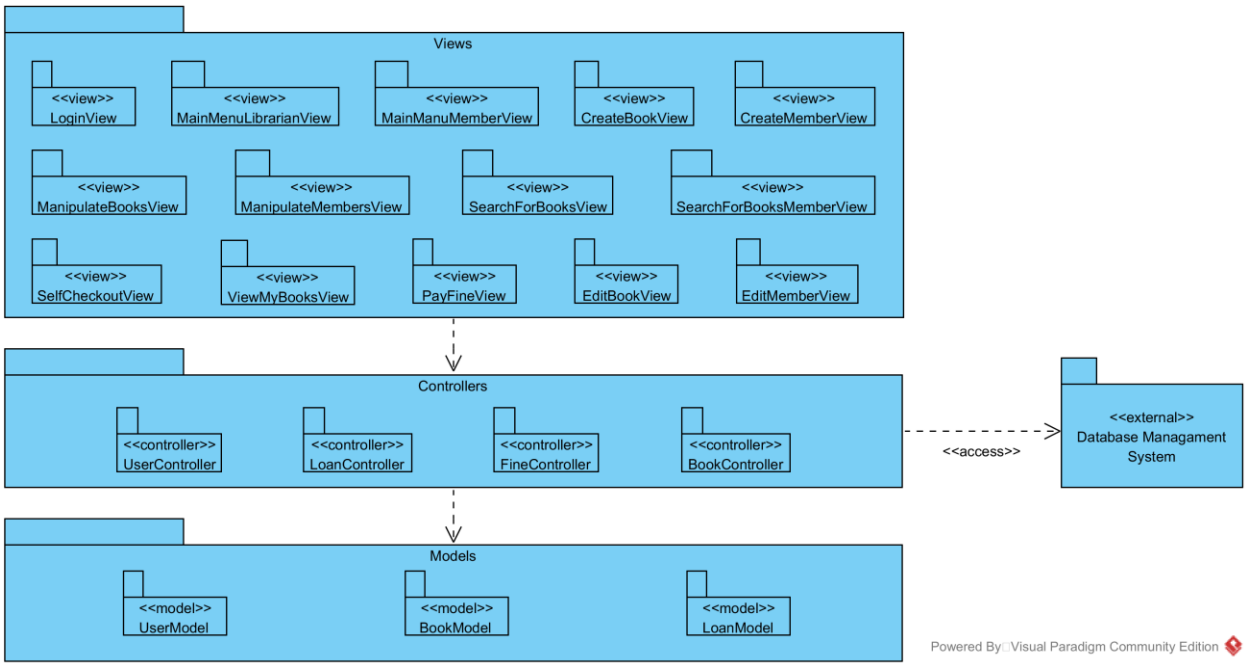

Figure 4 Package diagram

Critical Interfaces: Demo program interfaces are shown in Figure 5, 6 and 7. These interfaces are important because Login interface directs the members and librarians to different pages to for different transactions. For example, only a librarian can add member to the system, or only a member can view his/her books.

PS: These user interfaces are explained because they are implemented in demo application.

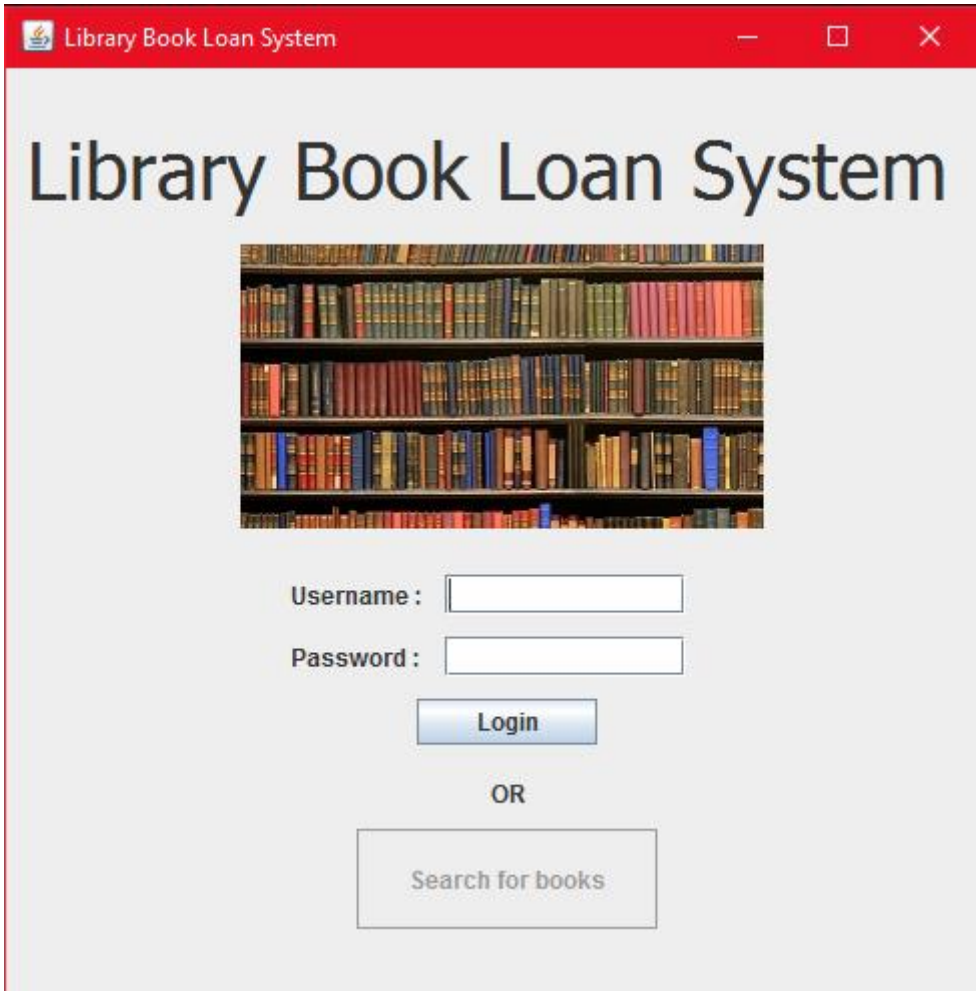

Figure 5 Login interface

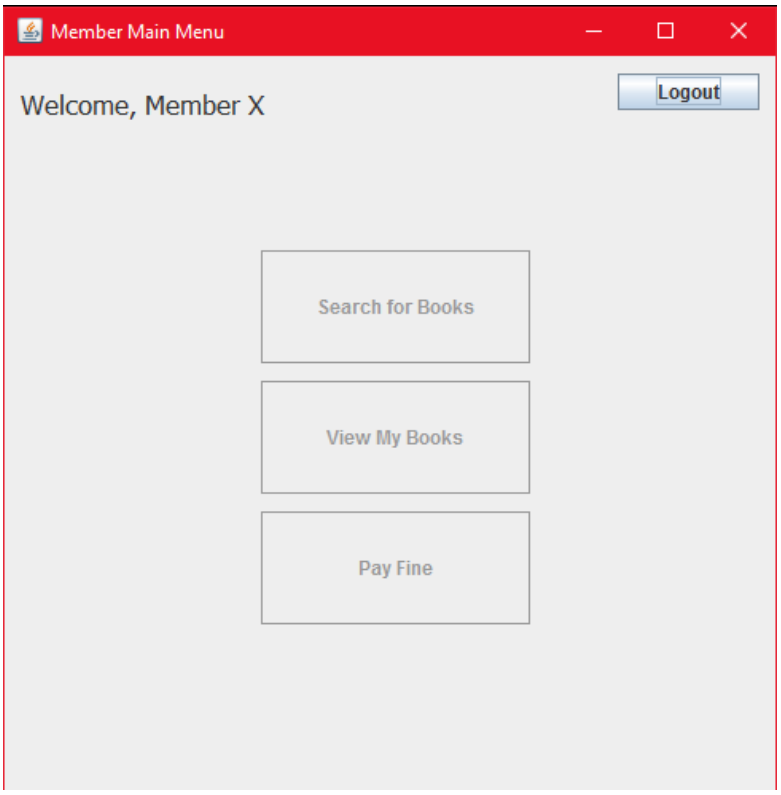

Figure 6 Member page

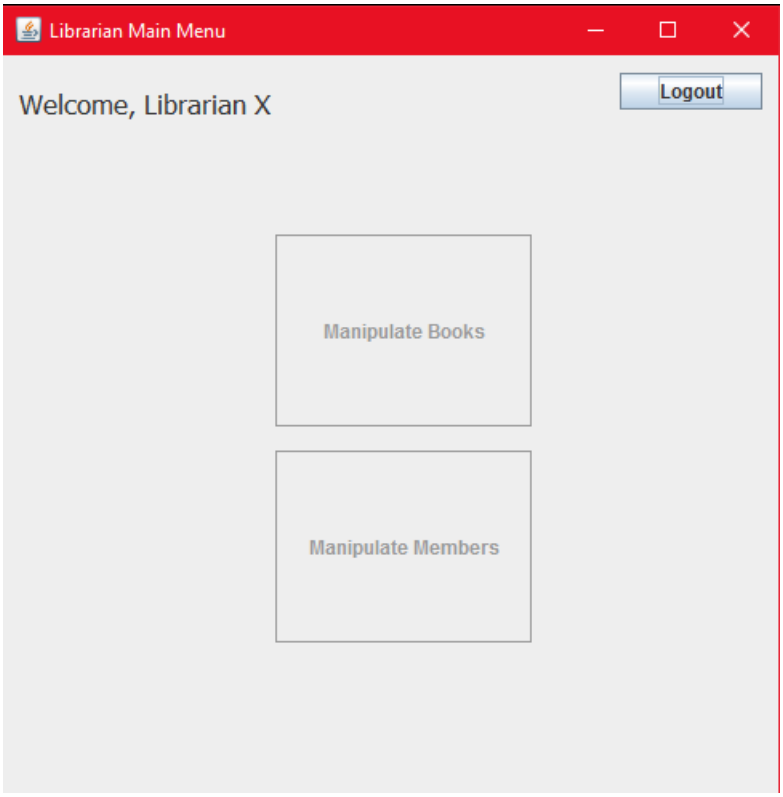

Figure 7 Librarian page

At below, there are deployment and use case diagrams to explain the logical structure of the system in detail.

Deployment Diagram: Deployment diagrams are used to describe the static deployment view of a system. They consist of nodes and their relationships. There is a deployment diagram like below in our system.

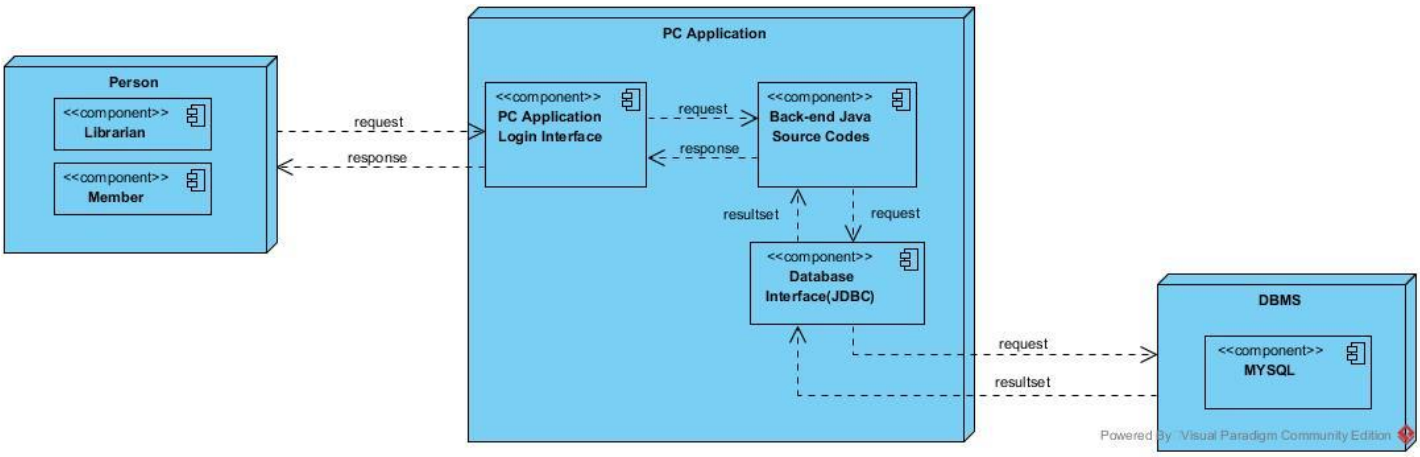

Figure 8 Deployment Diagram

Use case for Demo:

Use cases are defined like below in use case diagram.

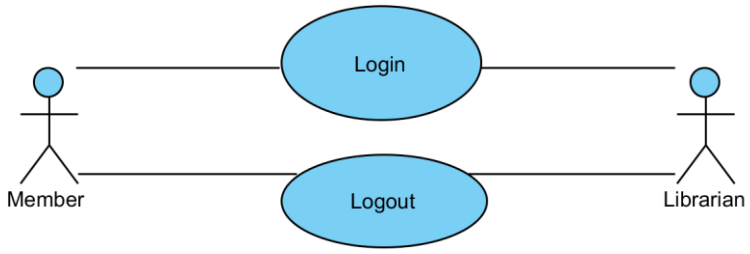

Figure 9 Use Case Diagram