Library Book Loan System
Architecture 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.
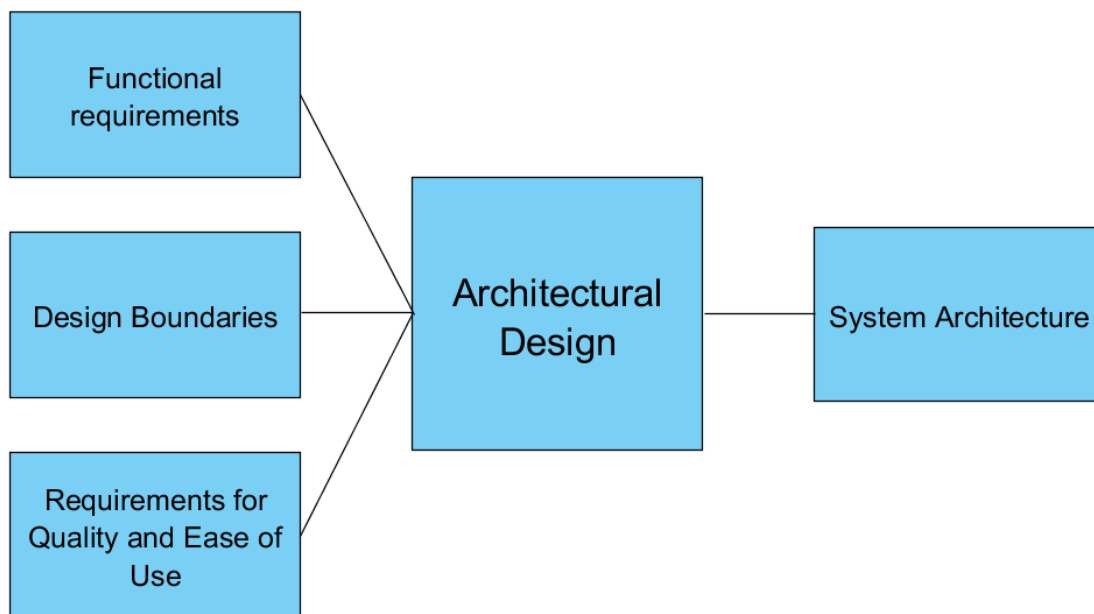
2. Architectural goals and philosophy

Our architecture is designed primarily considering the users. Easy to use, intuitive interface, responsive to needs, error-free and continuous operation, short response time and security are the most important goals of our system.

Our system is designed as a desktop application. Currently, it is only an application to be used in the library. However, we can improve our system and display remote users and borrowing information by providing remote access. But for the time being we are aiming to create a system that will be used only in the library by staying true to the plan.

We will use our shared database to run our system over a local network.

Our system will have 3 account types. A user with book borrowing and dropping functions, a librarian who can check the borrowing and dropping status of users books, and admin who can control the whole system.

We will take basic measures in the field of security. 3 types of accounts will only be able to enter their fields using their personal username and password. We will try to prevent the system from being an external intervention.

For now, our system will be offered only in Turkish.

3. Assumptions and dependencies

In order for our system to work, we need computers at certain points in the library and a local network between these computers. Our system is designed to be used in Windows operating systems.

Computers in the library can change over time and their operating systems can be updated. These changes can cause our system to need updating. Due to these problems over time, we will update our system.

4. Architecturally significant requirements

1. The library should be equipped with sufficient number of computers.
2. Windows operating system must be installed on the computers.
3. Our system should be installed on library computers.
4. The systems necessary for the use of the database should be established.
5. A local network must be established between computers.
6. Our program should be continuously open on these computers.
7. For book borrowing and dropping, users should login to the system.
8. Librarians must log in to the system to view and manage user and book situations.
9. Admin must log in to the system to view and manage the entire system.
10. If any problem is encountered, the technical support team should be informed.

5. Decisions, constraints, and justifications

- ➢ A local network is required to use the system. No internet access required.
- ➢ The ability to perform functions is more important than the interface of the program.
- ➢ For librarian and admin accounts, the maximum number will be determined before installation and these numbers will be set as limits.
- ➢ There will be no limit on the number of users.
- ➢ Adding and deleting books to the system is performed by the librarians.
- ➢ Librarians can perform user account creation.
- ➢ It is necessary to login to the system for the external operations of book listing.
- ➢ The system should respond as quickly as possible, depending on the hardware.
- ➢ Error and information messages should be understandable.
- ➢ The user manual for the system must be accessible to all users.

6. Architectural Mechanisms

Architectural Mechanism 1

In our project, we used MVC pattern as architectural mechanism.
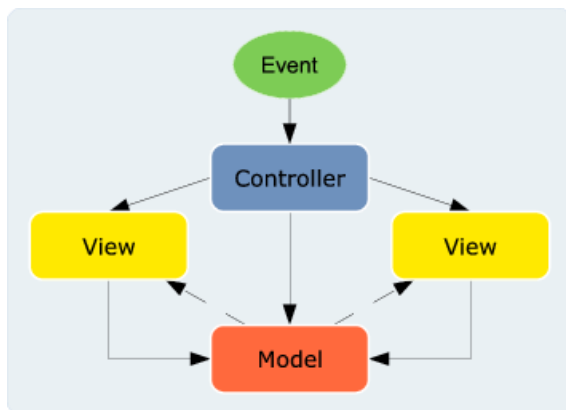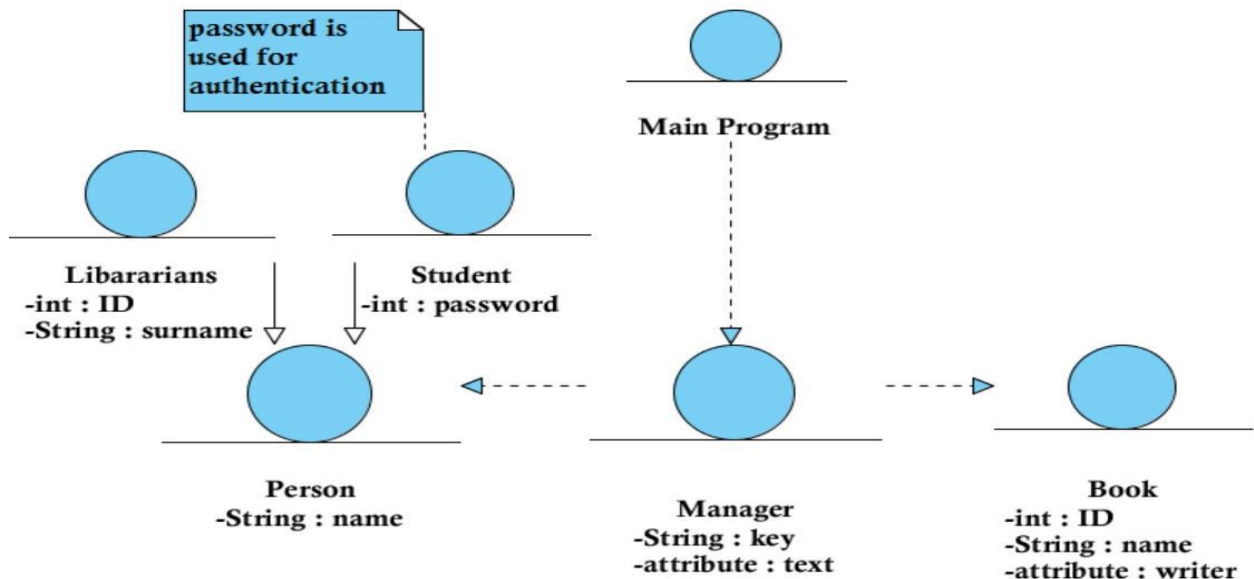

Architectural Mechanism 2

View: Includes interfaces that users can use. User would be member, visitor, librarian.

Model: It includes SQL server database for user, book operations.

Controller : That is connection between view and model and that can be user_controller and book_controller.

7. Key abstractions

Librarians: This is sub type of Person. Librarians can add, delete, and update books. They also can add a student to the system and remove him/her from system.
Student: This is sub type of Person. Students can ask for getting a book from system or return it. They have password for authentication.
Person: Describes the students and librarians information that is held by the system.
Manager: This abstraction describes the Controller Class in the MVC pattern. It includes all functions that run the application.
Book: Is the abstraction of book class that holds information about the book.

8. Layers or architectural framework

In Library Application we used MVC (Model View Controller) pattern. This pattern separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. In normal MVC Pattern there exists Model, View and Controller components

View: librarian, visitor, member interface
**Controller:** usercontroller and bookcontroller
Model: Librarian database

9. Architectural views

To choose the appropriate set of views, identify the stakeholders who depend on software architecture documentation and the information that they need. For an example of a set of views that have been used to represent 4+1 Views of Software Architecture.

4+1 is a view model designed by Philippe Kruchten for "describing the architecture of software-intensive systems, based on the use of multiple, concurrent views". The views are used to describe the system from the viewpoint of different stakeholders, such as end-users, developers and project managers. The four views of the model are logical, development, process and physical view. The 4+1 architectural view model can be graphically represented as follows:
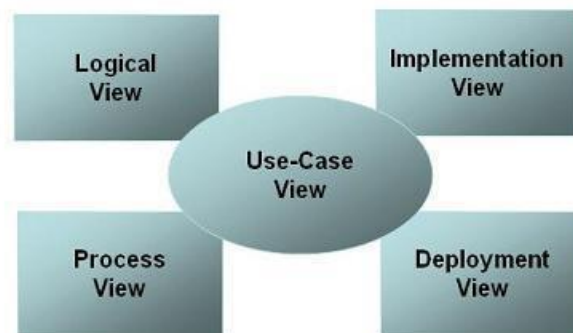
Deployment Diagram :
Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.
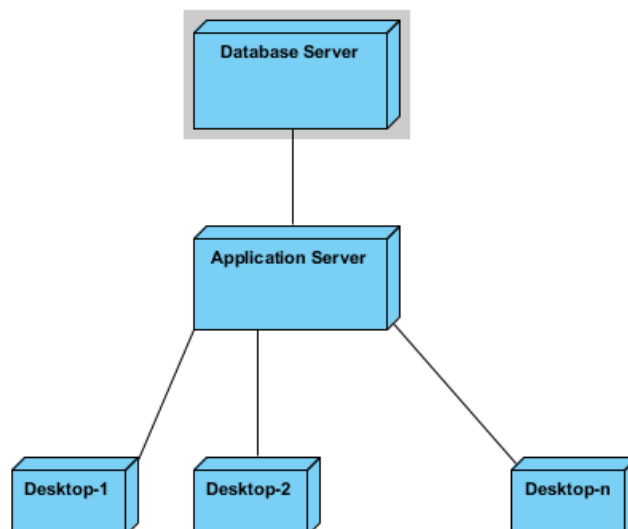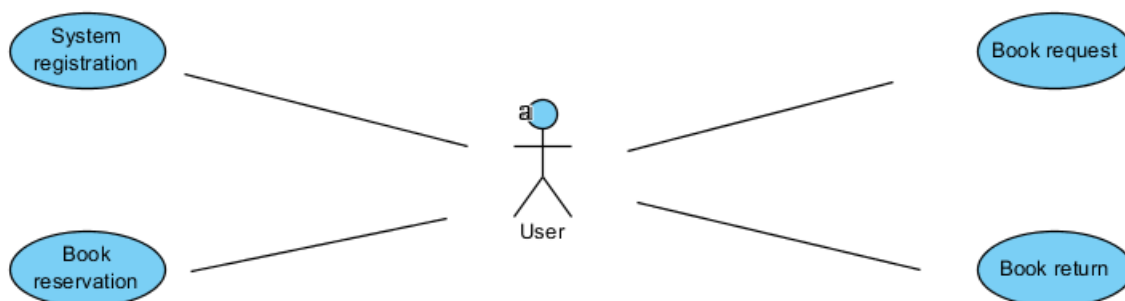
Component Diagram:

Component diagrams represent a set of components such as classes, interfaces and collaborations. They also show us the relationships between these components. In other words; component diagrams explain us the implementation view of a system. So, component diagrams are used to visualize the implementation.
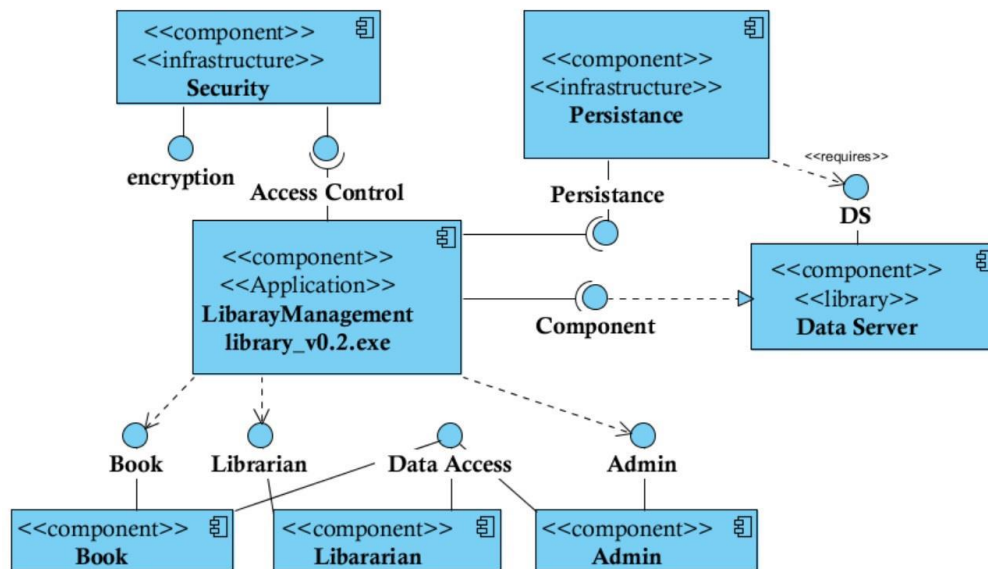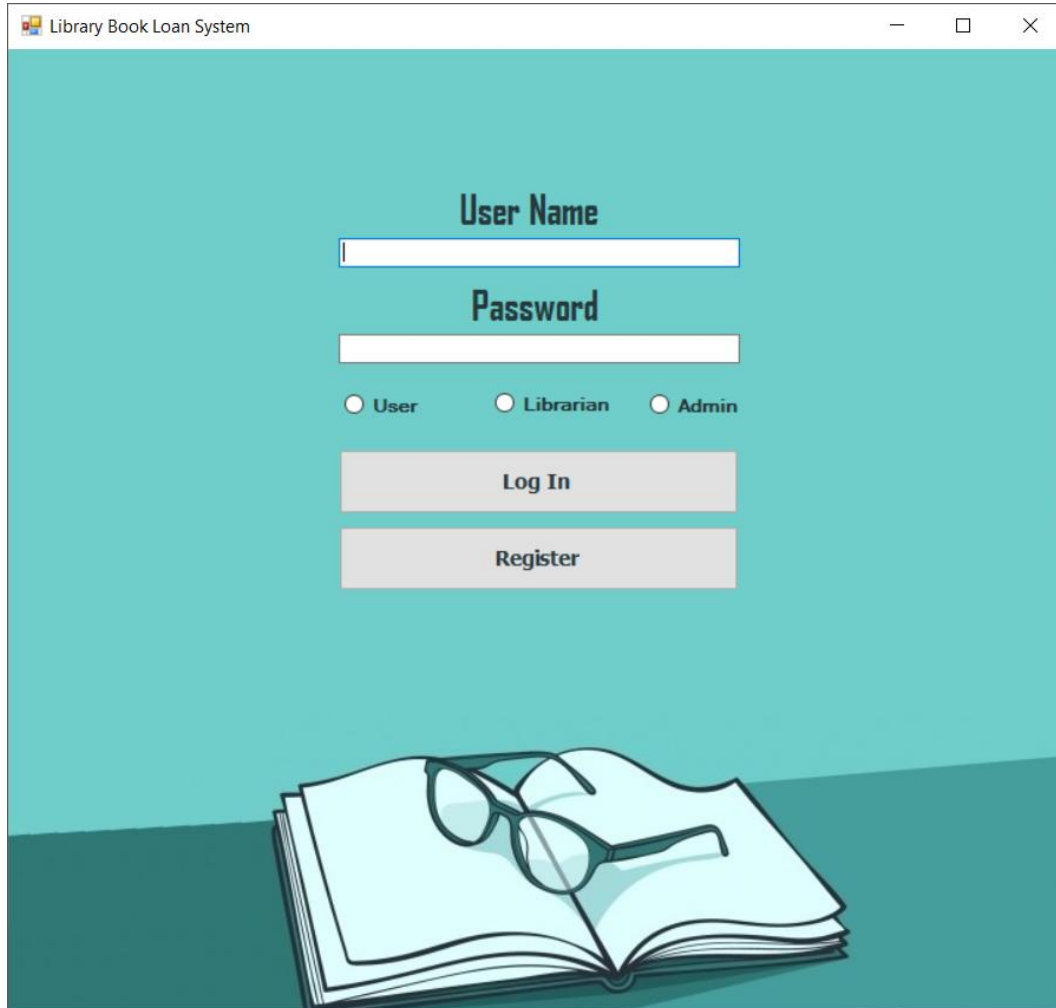

Use- Case View:
Describes functionality of the system, its external interfaces, and its principal users. This view is mandatory when using the 4+1 Views, because all elements of the architecture should be derived from requirements

Demo Program Interfaces

Here we choose our user type like User, Librarian or Admin. Then, we enter the system by entering username and password.
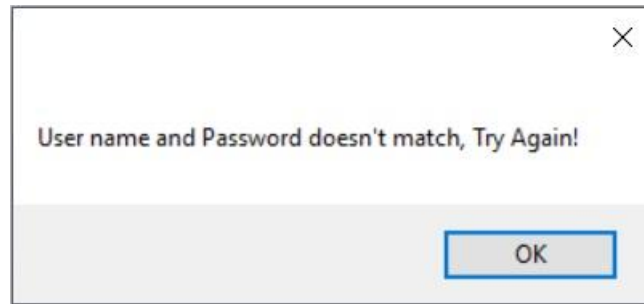
If we entered wrong Password, we get error message
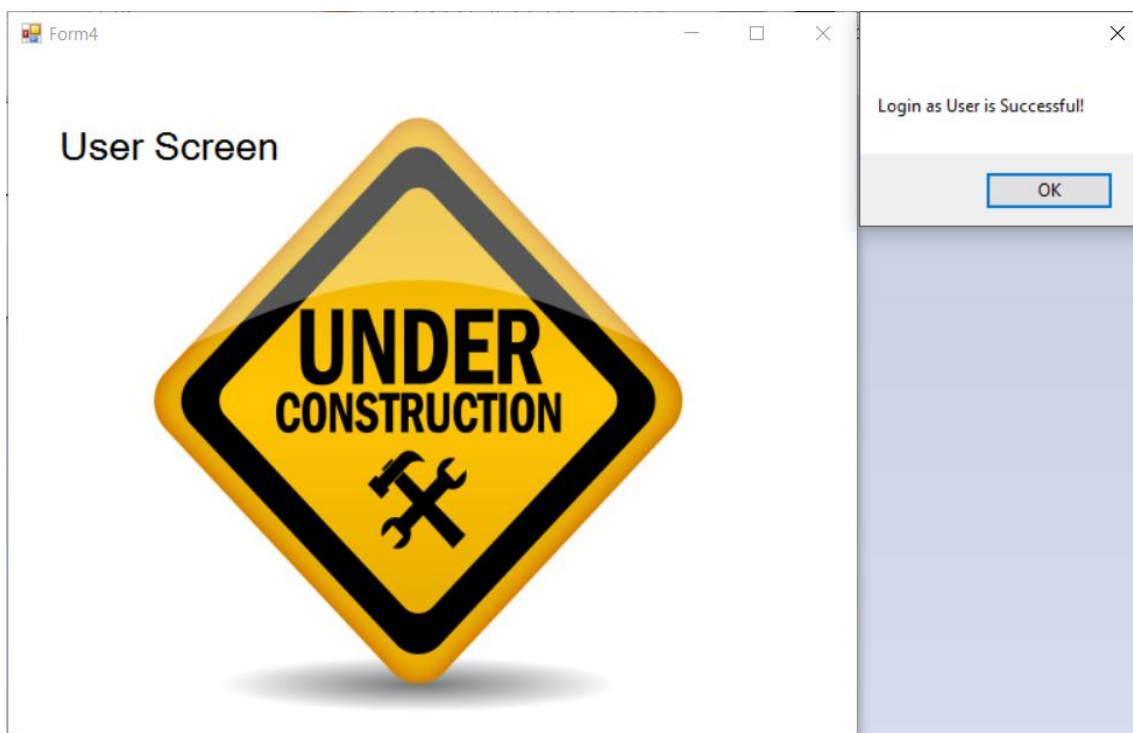
If we entered correctly, we get success message