

HACETTEPE UNIVERSITY



COMPUTER ENGINEERING

SOFTWARE ENGINEERING LABORATORY

BBM487

Group 17

Members

ID	Name and Surname
21200894	Gismat Kazımlı
21228602	Hiledin Özer
20902958	Rinat Davidov
21228046	Burak Aykanat

Document Version: [Open Up Architecture Notebook](#)

Date: [04.04.2017](#)

Table of contents

1. Purpose.....	3
2. Architectural goals and philosophy	3
Technical Platform.....	3
Security	3
Reliability and Availability	3
Performance.....	4
3. Assumptions and dependencies.....	4
4. Architecturally significant requirements.....	4
5. Decisions, constraints, and justifications	4
6. Architectural Mechanisms.....	4
Architectural Mechanism	4
7. Layers or Architectural Framework	5
8. Architectural Views	6
1. Logical View	6
2. Use Case Diagram	6
3. Component View	7
4. General Entity Relationship Diagram	7

1. Purpose

Our goal in this document is to explain how we are shaping the system design, limiting the system's contents, quality standards and requirements. When we made these statements, we considered the system documentation that we had delivered before. And it should be noted that this document is a bridge between Analysis and coding.

2. Architectural goals and philosophy

As a project we will build a desktop software and the architecture of our system is based on the basic components. These key issues are in general a guarantee of the security of a desktop software and at the same time maintain the continuity of its service.

We will build the system with new features updates and adapt to the new rules. Thus, it will be able to maintain its validity as long-term software. Performance problems will vary depending on the access speed required to pull data into the database in general.

Basic architectural features of our system:

Technical Platform

Technical platform is JavaEE(Java Enterprise Edition) that we are using to create system.

Security

Our security rules provide features as Authentication, Authorization and Non-repudiation.

Authentication: Security will be provided for every user and library officer by entering their own username and unique id .

Authorization: A different interfaces with operations will be showed for each different kind of user.

Non-repudiation: Each user is responsible for their own account changes. Because the account can only be entered with the information that the user knows. So the user can not deny the changes.

Reliability and Availability

Consistent availability of the system increases reliability. Since we try to keep our system running 24/7 in our architecture, each user will be able to perform their own actions and increase their confidence in the system, since they have not experienced any problems. At the same time, while performing user transactions, another user will not be able to see his transactions. Thus, reliability will be ensured

In addition, these two features are very similar with usability. Reliability allows the user to make their requests at any time and when reliability doing that, usability provided convenience when securing.

Performance

The performance of your system will respond to any user in short time about 10 seconds and will perform any operation in a short time, except reaching data from server because it depends on the speed of the server.

3. Assumptions and dependencies

Our basic assumptions in our project;

- To ensure that our system can be updated in response to needs.
- When we try to achieve the updateable system, it needs to work with a specific version of a few programs like JDK etc.
- Since our project will be developed on Java platform, it will be compatible for any operating system which supports Java.

4. Architecturally significant requirements

Requirements are shown in following figure.

REFERENCE	REQUIREMENT's DEFINITON
R1	The User must have a desktop or a laptop computer
R2	The User must have input devices (keyboard and mouse)
R3	The user must have the current java version.
R4	The user must be logged in to loan or reserve a book
R5	The system must be recoverable to any accident
R6	The system must always be operational.
R7	The system must record and save each change that user did on his own account

Excepts this requirements our system must have another requirements such as extensibility and modularity.

5. Decisions, constraints, and justifications

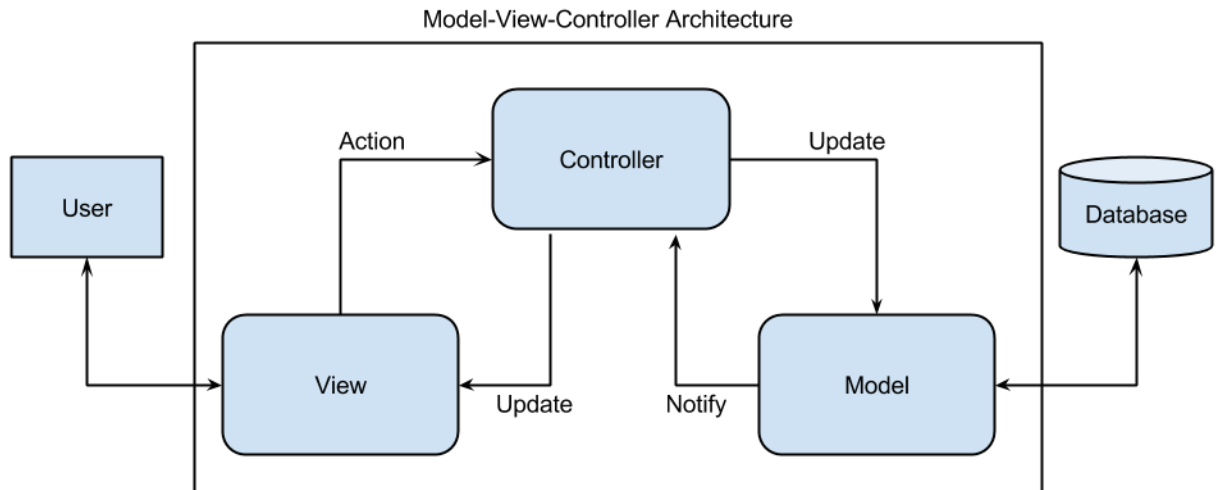
Our design, constraint and justifications for the system are that

- It will be developed using JavaEE Platform .
- Each user which can be ordinary or librarian admin will have a unique ID(or username) and password
- It will have valid response time for each users
- It will record all changes made by users in their accounts.
- We are using MySQL server for database management.
- We are using Swt(Standard Widget Toolkit) as an interface of one of java libraries.

6. Architectural Mechanisms

Architectural Mechanism

The architecture we build for the architecture of the library system is an MVC model. This model is a simple and appropriate model for our Library system.



In this pattern,

User which is going to use library system will only connect to the view part. View part will send the changes made by the user to the controller and controller will update information of this user.

7. Layers or Architectural Framework

Our system has three layers and these are;

User Layer: This is the fronted layer which is users will see and made it using GUI of Java.

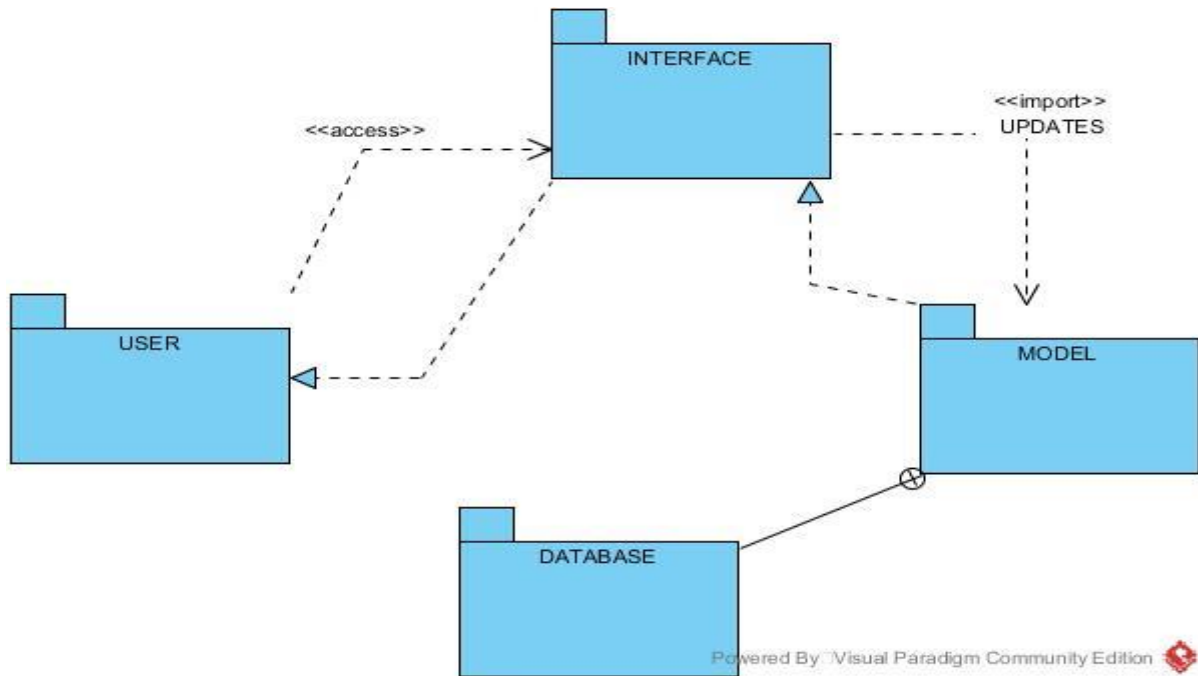
Working Layer: Layer of the controller that provides updates to the model. This layer is invisible to the ordinary user.

Database Layer : Layer of the database which is holding information of user. This layer is invisible to the ordinary user.

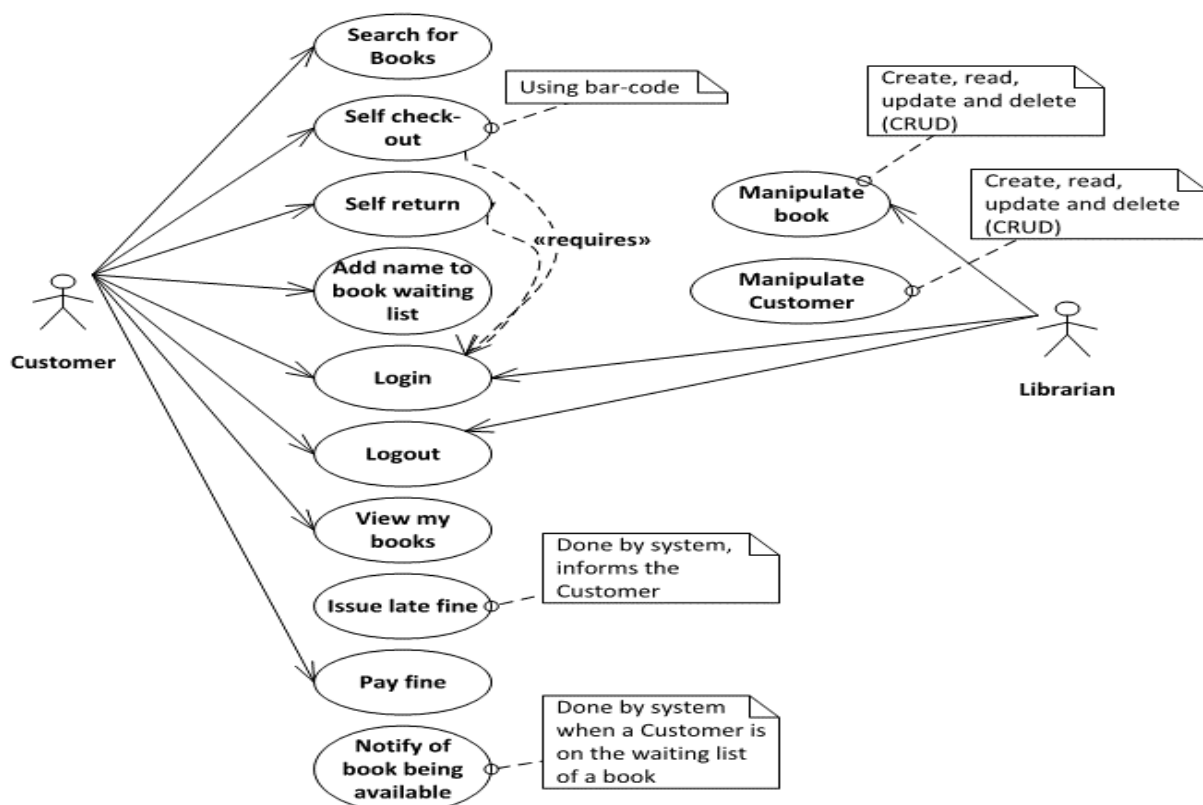
8. Architectural Views

1. Logical View

It determines how the overview of the system is, for this reason it shows the actual facts in general.

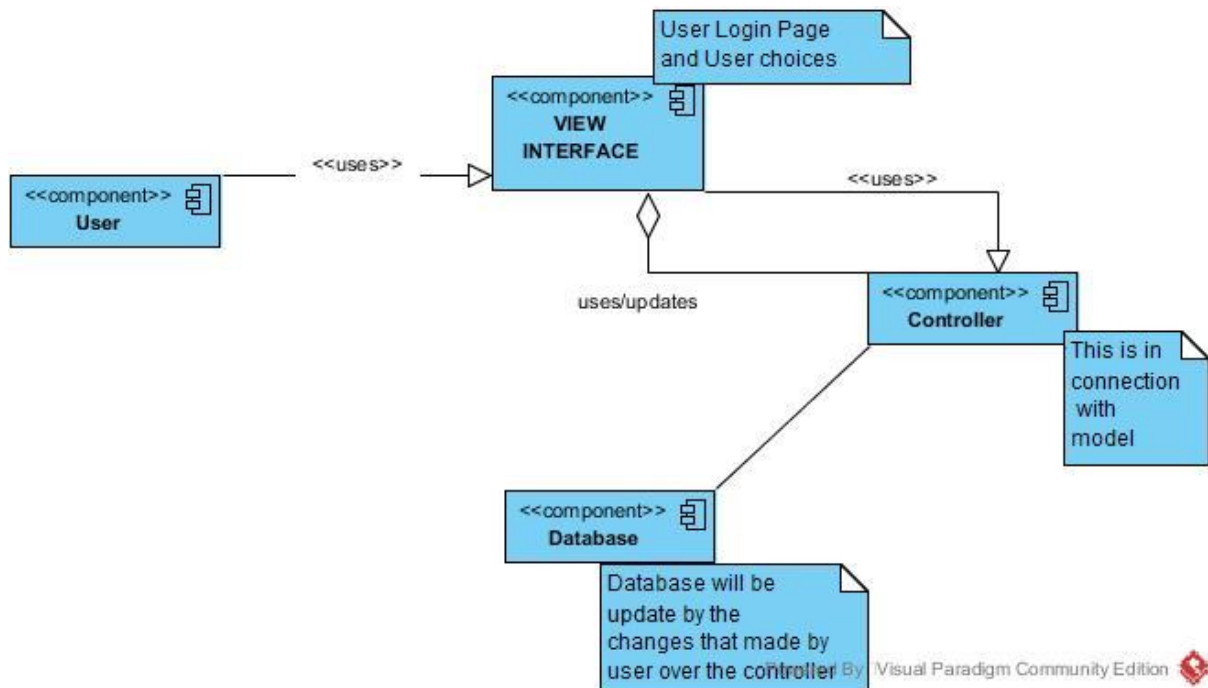


2. Use Case Diagram



3. Component View

Component diagram which show us basic operations is following.



4. General Entity Relationship Diagram

General entity diagram which shows us general relation between entity is following.

