

Library Book Loan System	Version: 1.0
Architecture Notebook	Date: 04/03/2017



Hacettepe University
Department of Computer Engineering
BBM478 - Software Engineering Laboratory

Architecture Notebook

Group One

Ahmed Şamil BÜLBÜL (21426749, Project Manager)
Halil İbrahim ŞENER (21328447, Software Architect)
Naciye GÜZEL (21580841, Software Developer)

Library Book Loan System	Version: 1.0
Architecture Notebook	Date: 04/03/2017

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.

With all of the topics given above, this document will show the architecture of the LBLS.

2. Architectural goals and philosophy

The main architectural goal is to provide a well-designed architecture that can be used easily by the librarian and the customer. It should give good user experience. While the architecture is being developed, all requirements of users that are described in the Vision and the System-Wide Requirements Specification document shall be taken into consideration.

The LBLS (Library Loan Book System) will be a desktop application. The system will not be driven by complex deployment concern. The LBLS will not be dependent any specific hardware. It will be an operating system independent application. So it will work on all operation systems that support Java. MySQL database will be used in the system.

3. Assumptions and dependencies

The system is designed to accommodate changes because it is assumed that new features can be added in future versions of the system. So the software can be easily developed and maintainable.

The architecture of the system is designed to be fulfilled by 3 team members.

It was designed with deciding that it will be a desktop application.

It will be dependent on the designed interface in Appendix B of the System-Wide Requirements Specification document.

It will be dependent the use case of the users.

4. Architecturally significant requirements

Review the Quality Requirements in the System-Wide Requirements Specification document for the more detailed description of the architecturally significant requirements.

5. Decisions, constraints, and justifications

- The system has to be develop using Java programming language.
- LBLS will be able to run in every operating system that runs Java.
- MySQL database will be used in the system.
- The LBLS will be a stand-alone desktop application that requires a database connection.
- The system will be distributed over GitHub, documentations will also be on GitHub.
- The system will provide functionality and easy-to-use graphical user interface.
- Manipulating users can only be done by librarians.
- Manipulating book can only be done by librarians.
- Guest users can only use one functionality which is "Search for Books"
- The performance of the system should not decrease under multiple users logged in.
- Security of user logins will be provided by SHA-256 (Secure Hash Algorithm 2)

Library Book Loan System	Version: 1.0
Architecture Notebook	Date: 04/03/2017

6. Architectural Mechanisms

Architecture mechanics are design patterns developed for finding solutions to common problems. The main architecture mechanic we'll be using is MVC.

Model-View-Controller

Main purpose of using MVC pattern is to provide a better framework to work. In the business domain, there are lots of data to deal with. Not every data will be shown to every user. Every user has different interaction with the system. To create an appropriate communication between data and user, there will be controllers. By doing this we get MVC.

MVC has got three main parts, model, view, and controller:

1. **Model:** Model is the main and most important component of the pattern. Its main purpose is explaining data structures and behaviors of the system. It is about the business domain of the system. It doesn't contact to view directly.
2. **View:** View is the graphical user interface to show information that is stored in models. Models' information is shown here.
3. **Controller:** Controller is the bridge between model and view. It takes input and converts it to commands for the model or view.

Attributes of the MVC are:

- **Security:** By using MVC, direct object accesses will not be operable. Objects will handle accesses to the database.
- **Reliability:** Any transaction which is not completed will not change database so objects should not be preserved.
- **Update frequency:** Objects are not changed frequently.
- **Latency:** User client and database should communicate fast enough to handle multiple users at a time.

As we can see controller has got the ability to create interactions between every component in the system.

7. Key abstractions

As our business abstractions, we have Customers, Librarians, and Books. These are the main three abstractions.

- Customer: Abstraction of the user that is capable of operating on the library system.
- Librarian: Abstraction of overseeing and managing the library system.
- Book: Abstraction of main material of the library system.

Other abstractions are:

- Views/Graphical User Interface: Depending on the user type, users will have different views.
- Database: Abstraction to store business abstractions.

Library Book Loan System	Version: 1.0
Architecture Notebook	Date: 04/03/2017

8. Layers or architectural framework

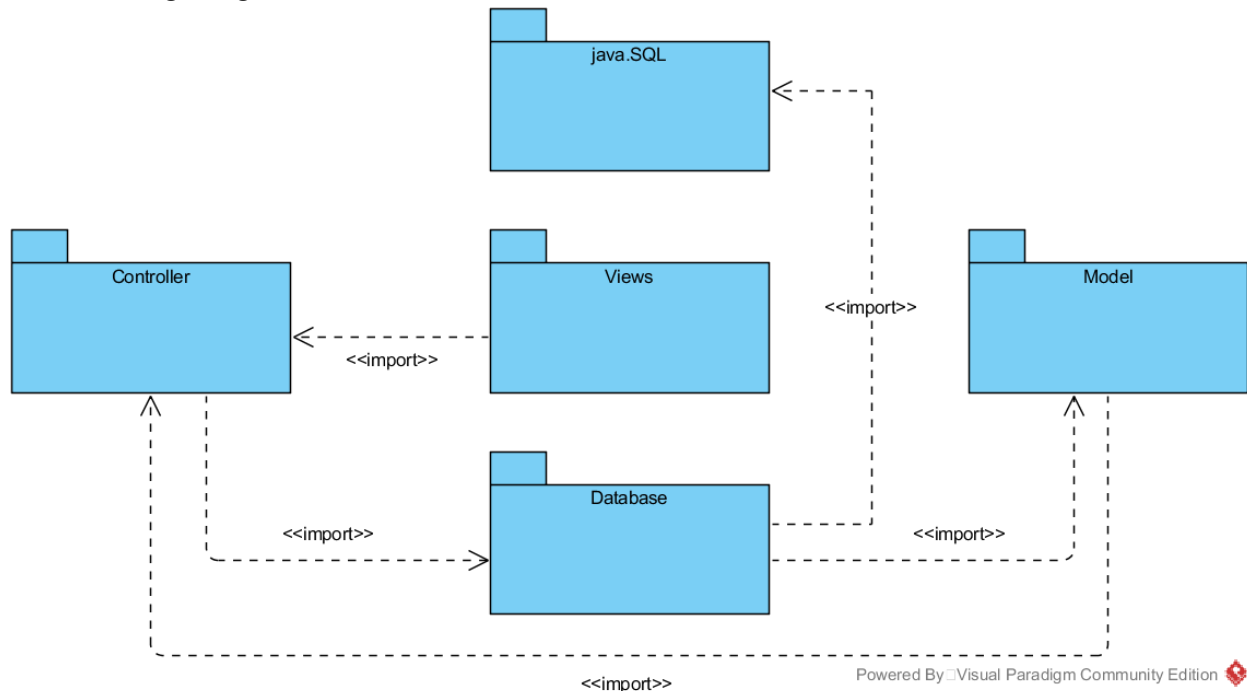
The architectural framework is defined by components. These components are defined as the physical aspect of an Object-Oriented software system. A component can be replaceable. If we keep interface and change component, the system should keep its working status without changes (or small changes) in the interface.

Our main architectural framework is 4+1 architectural view model. 4+1 is a view model designed by Philippe Kruchten. Its aim is to describe the architecture of software-intensive systems, based on the use of multiple, concurrent views. The 4 in 4+1 are logical, development, process and physical view. The 1 in the 4+1 is selected use cases or scenarios are used to illustrate the architecture.

9. Architectural views

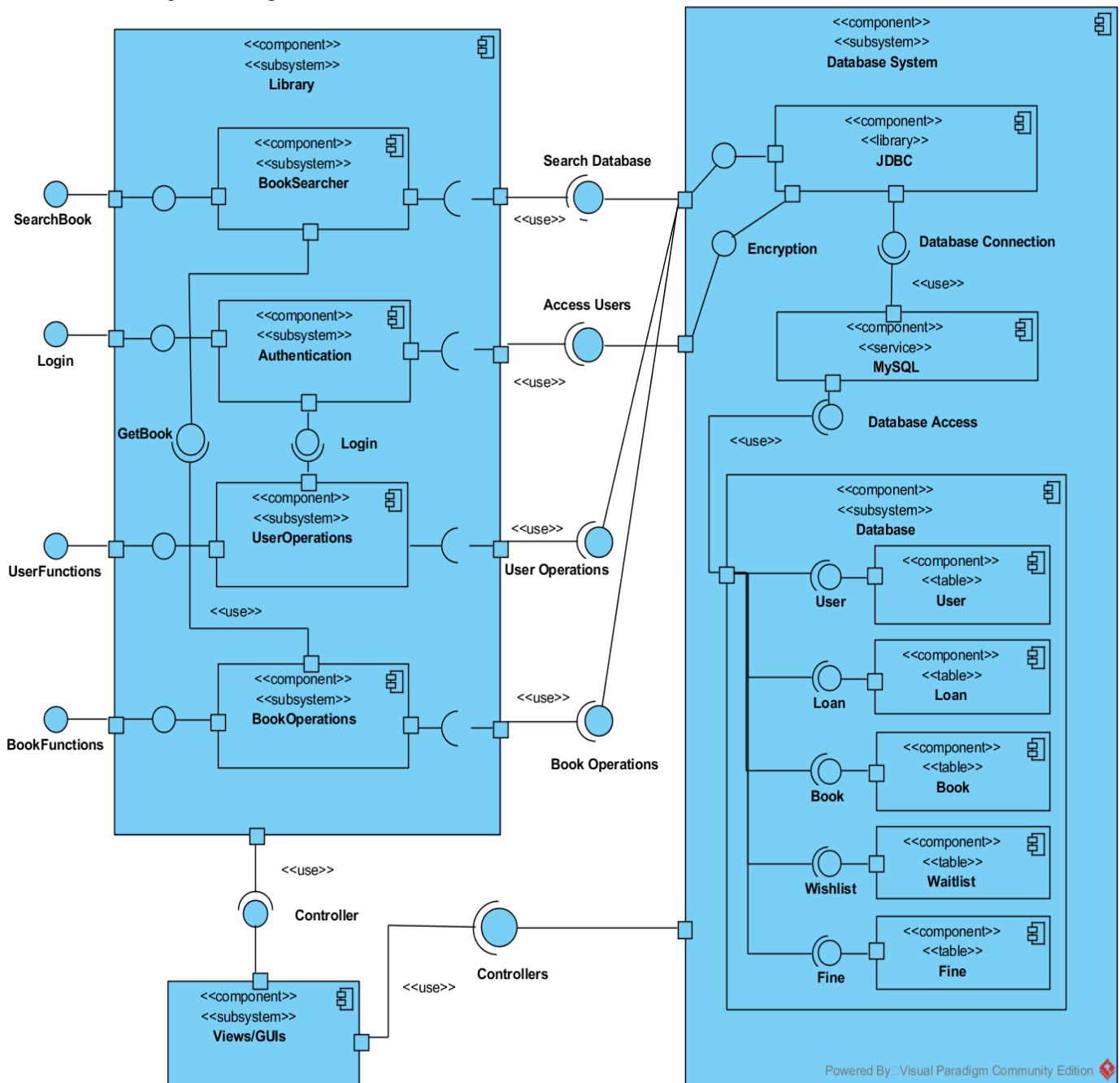
By using 4+1 Architectural View Model, we have got logical, development, process and physical views.

- **Deployment View:** It illustrates a system from a programmer's perspective and is concerned with software management. It has got two diagrams: Package diagram and Component diagram. Package diagram shows how packages communicate in the system. Component diagram shows how small *Package Diagram* is as below:



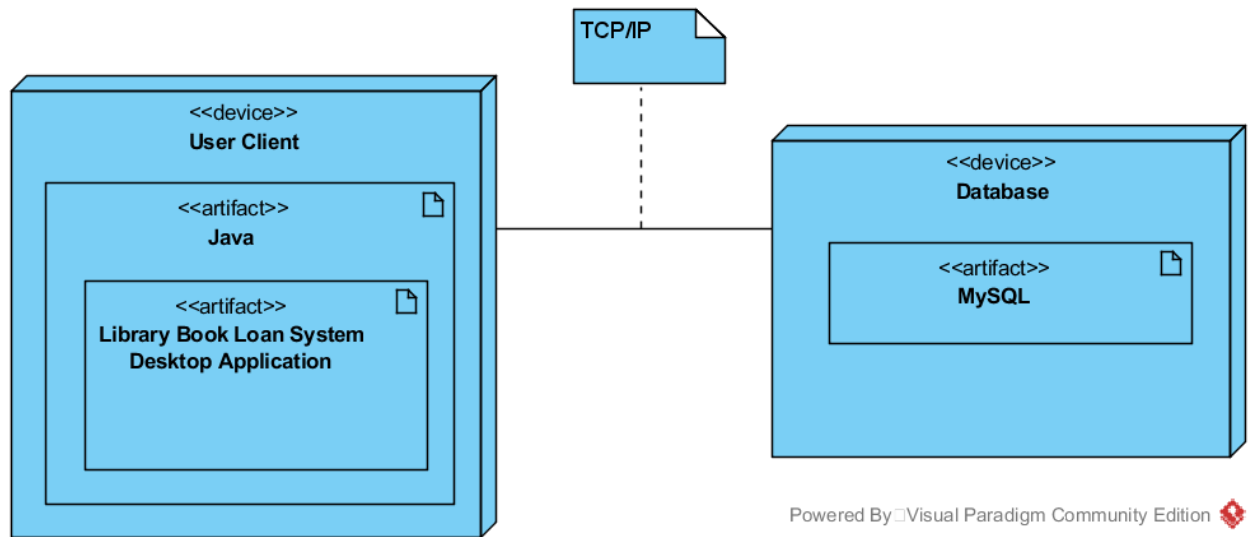
Library Book Loan System	Version: 1.0
Architecture Notebook	Date: 04/03/2017

Component Diagram is as below:



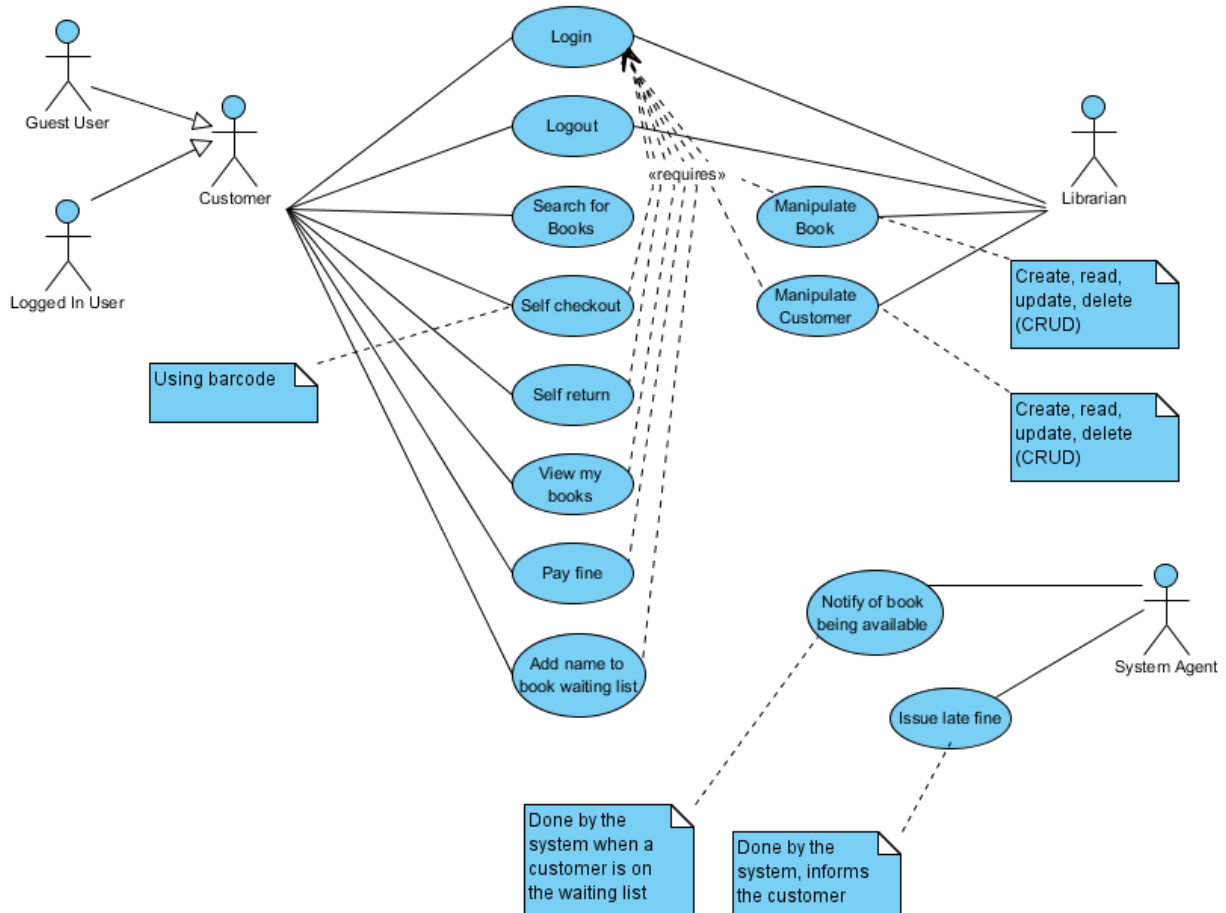
Library Book Loan System	Version: 1.0
Architecture Notebook	Date: 04/03/2017

- Physical View:** It has got Deployment diagram to show how the system will look in the physical layer. It shows software components and their hardware parts to show relations between them.
Deployment Diagram is as below:



Library Book Loan System	Version: 1.0
Architecture Notebook	Date: 04/03/2017

- Logical View:** The use case of the LBLS is as follows. Please refer to the System-Wide Requirements Specification document for the more detailed description of the use case.



- Process view:** This view contains Activity diagrams. To get a better understanding please refer to the System-Wide Requirements Specification document.