



HACETTEPE UNIVERSITY

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

BBM487 SOFTWARE ENGINEERING LAB

Group 14

LBLS Desktop

Architecture Notebook

20926046
21327618
21228731

RIDVAN ALTUNEL
AYŞE AKCAN
MAHMUT TEKİNYER

Project Manager & Developer
Architect & Tester
Analyst & Technical Writer

Library Book Loan System	
Architecture Notebook	Date: 04/04/2017

Library Book Loan System Architecture Notebook

1. Purpose

The architecture is one of the most critical concepts in software development. It effects development process and effort directly, also it is associated with many of requirements. Therefore it must be well considered before the development process. Also generally customers are inclined to demand too much revision during project development, therefore the architectural parts of the system must be organized separately as much as possible to manage these revisions minimal effort.

In following sections of this document, these some concepts that related with the architecture of LBLS Desktop project will be detailed.

2. Architectural goals and philosophy

The main architectural goal is to develop this software in a short term development time and meeting all software requirements. To achieve this, proven technologies have been chosen while making architectural decisions.

Just before starting the development process, we changed our graphical user interface approach from JavaSwing to JavaFX because of JavaFX useful features such as styling .fxml (very similar to .xml) with .css files. Also JavaFX includes some build-in functions that gives developer to handle some user interactions that came up with mobile devices such as touch, swipe etc. This alteration will made this software more integrable to another devices such as touchscreen computers, mobile phones, tablets etc.

Technical Platform

LBLS Desktop application will be installed on desktop computers. Any operating systems must be supportable.

Transaction

LBLS Desktop application is transactional, leveraging the technical platform capabilities. Transaction management model of the Java and JDBC connected to MySQL server platform will be implemented.

Security

Software must be reliable because of taking critical information about user. Some of basic security behaviors of the application explained below:

Library Book Loan System	
Architecture Notebook	Date: 04/04/2017

- Authentication: Users can be login to system with their username and password. Also they must have a school number that defined in university database.
- Authorization: The features provided by the application are offered and displayed to user according to it's authority.
- Confidentiality: Critical data must be encrypted before sending to database server and getting data from there.
- Data integrity: Before a send / get action, all of operations must be done in complete.
- Auditing: Some critical actions must be explained in a text box to record it.

Availability

System must be available 7/24 just like libraries normally. Public holidays can be selected for maintenance activities. Critical situations can be effected this goal exceptionally.

Persistence

Data persistence will be handled by database part of the system.

3. Assumptions and dependencies

3.1 Assumpitons

Following assomptions affected the architectural decisions:

- As recommended, OpenUP has been adopted as project planning approach.
- Due to complexity of developing an application that includes graphical user interface, model view controller approach has been applied during object modeling and packaging the whole project.
- To manage data easily, MySQL used for database. Database stands online, managed in a free service with phpmyadmin panel.
- Because of developer member(s) experience Java has been selected as programming language. And JavaFX finally selected for user interface organizing after some research.

4. Architecturally significant requirements

Some of requirements are more significant than others while making architectural decisions. These effects system architecture directly, and should be analyzed in depth.

The following requirements are significant architeturally:

- Response time of the software must be maximal 3 seconds.
- The system should stay available 7/24 such as libraries.
- Data consistency must be assured to avoid disputes.
- The system may have concurrent users at the same time.
- These users can modify on same object.

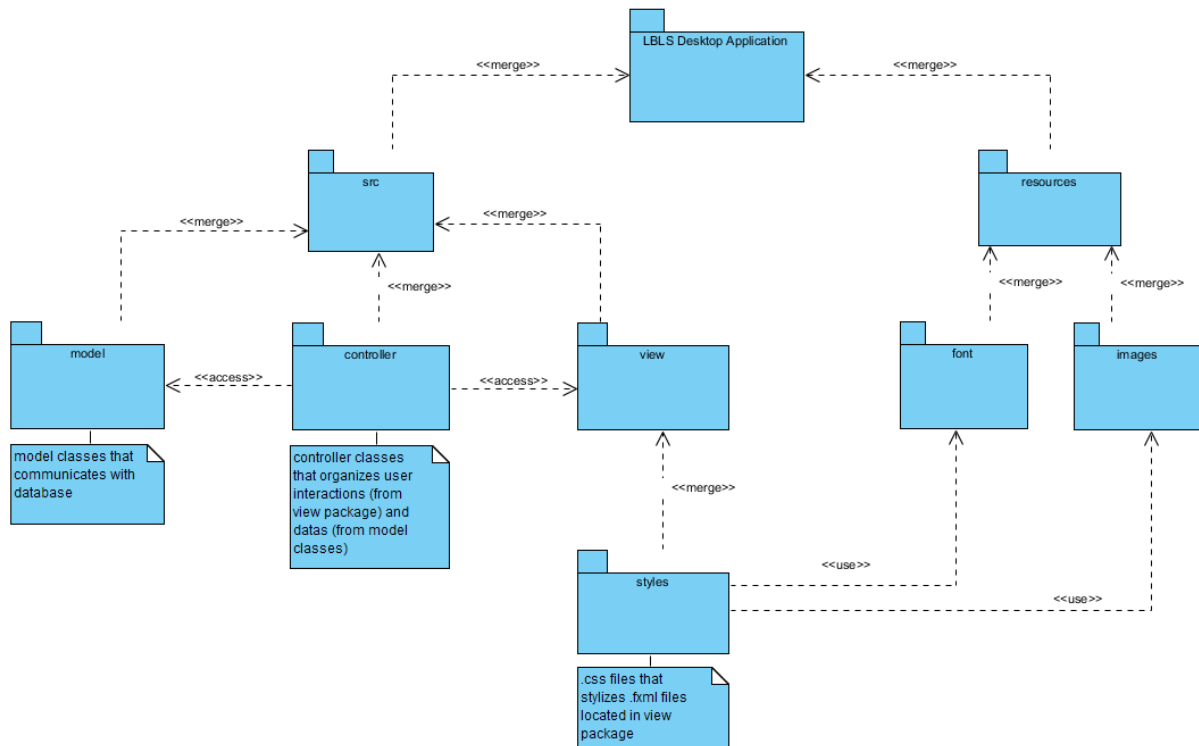
Library Book Loan System	
Architecture Notebook	Date: 04/04/2017

- Whole system includes some critical information about users, so these critical data must be encrypted while transferring between software and database.
- Software's front-end and back-end must be separated from each other in terms of easy management, arrangement and updates.
- Database connections should be handled in a separate class because of the possibility of moving another service.
- The software must be open to integration with other devices and platforms due to possibility of expansion.

5. Decisions, constraints, and justifications

- The system will be modeled as object-oriented, and coded in Java.
- Development will be implemented on Eclipse IDE due to ease of integration and development team's familiarity.
- System will be implemented using Model-View-Controller pattern to separate critical parts of application from each other.
- Model classes will be created according to database tables firstly. May an object relational mapping method will be chosen further due to complexity of database.
- Graphical user interface will be shaped with JavaFX .fxml files and .css styles. Java Scene Builder 2.0. will be used for organizing .fxml files. Adobe Photoshop CC will be used for designing icons, background images etc. Roboto font family will be used for typography of the software.
- MySQL will be used for data management of this project due to ease of use and portability.
- JDBC API will be used for connecting model classes with database.

6. Architectural Mechanisms



 **LBL Desktop** PackageDiagram

The architecture will be shaped with MVC pattern. JavaFX organizes graphical user interface with .fxml formatted files that very similar to xml files. Also there is facility that all of components of interface can be stylized with .css files. In our architecture, all of these .fxml files and .css files are located in View package. Every screen of this software handled in an independed .fxml file.

.css files located in a sub package named "syles". In .css files there are some references, paths to softwares resource folder that contains images, fonts etc.

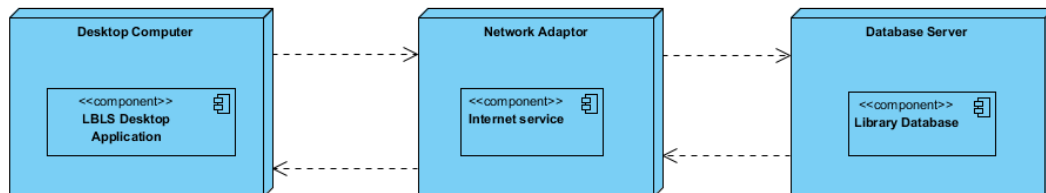
Each .fxml file has a reference to a controller class to handle user interactions clearly (this feature provided by JavaFX). So Controller package contains all of these screens controller classes. They communicate with model classes that are the reflection of database in our software. Every database tables and views rebuiled as Java classes in our model package.

Controller classes handles user interactions and reflect these actions to database with help of model classes and JDBC API.

Library Book Loan System	
Architecture Notebook	Date: 04/04/2017

7. Key abstractions

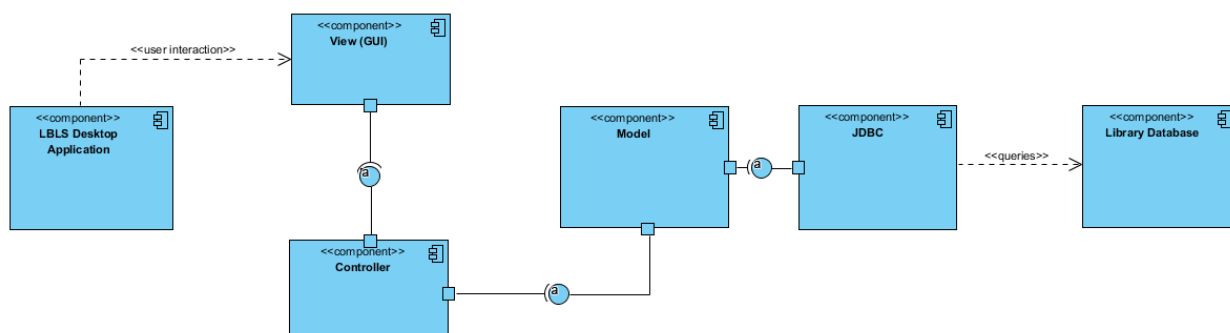
Following figure contains LBLS Desktop Deployment diagram:



LBLS Desktop Deployment Diagram

8. Layers or architectural framework

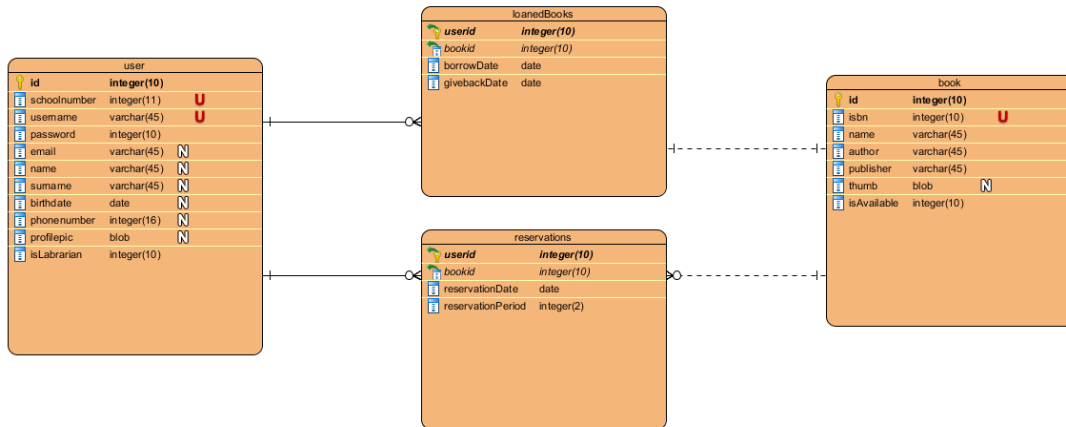
The LBLS Desktop Application component diagram is placed below:



LBLS Desktop Component Diagram

9. Architectural views

Following figure contains LBLS Desktop Application database Entity Relationship diagram:



— LBLS Desktop ER Diagram