Design Document

CSCE 361 - Spring 2018

Team 16

1. Introduction

The purpose of this document is to describe the functions and capabilities of the Online Note Sharing System. This document will use relation diagrams to demonstrate the relationships between different parts of the system, as well as the architecture that the system will be using. This document is intended for the software engineers that will be building and maintaining this system.

2. Architecture

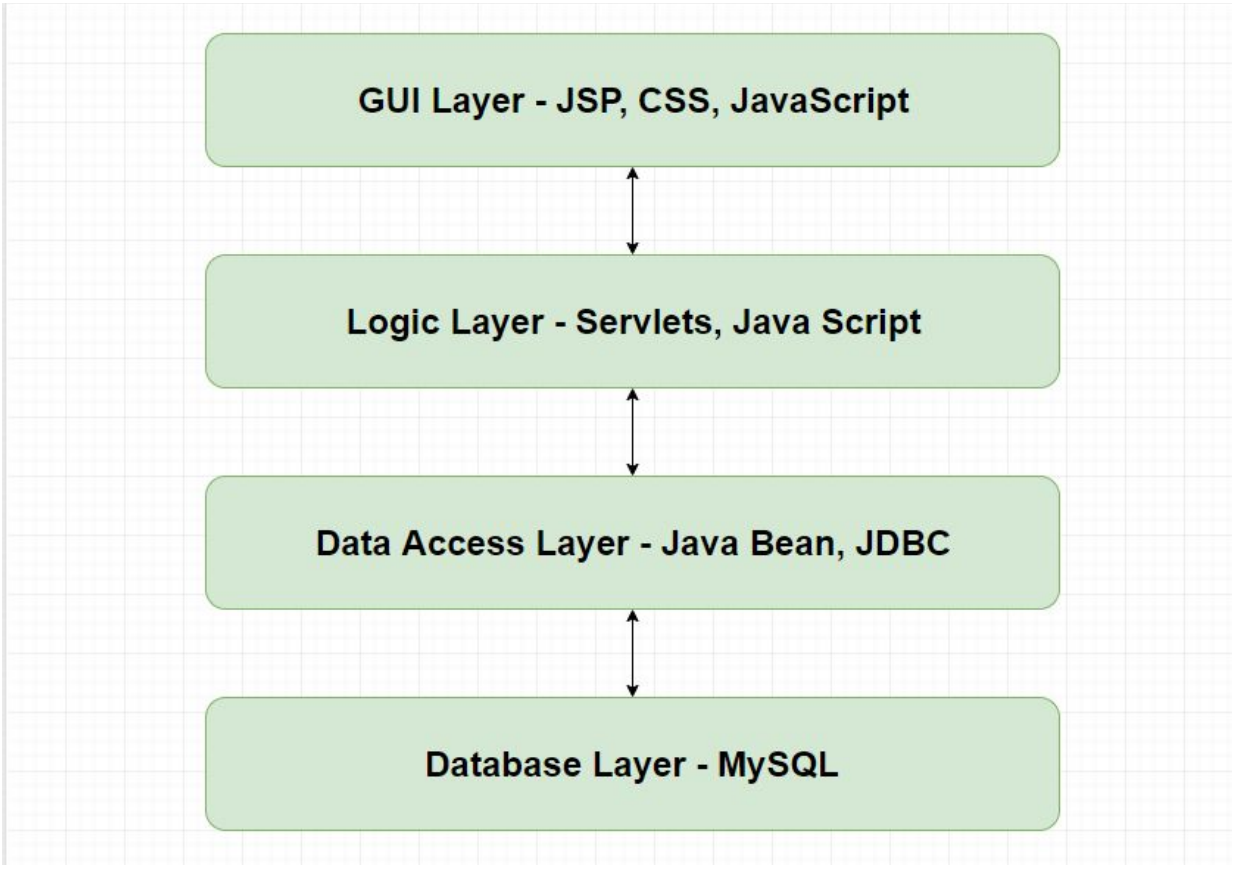

Figure 1: Architecture Diagram

2.1 Introduction

The architectural design of the system will be made using the layered model. On the lowest level we have the database layer; using MySQL we will store data such as notes and user informations. Data Access Layer can interact with the Database Layer via an API called JDBC (Java Database Connectivity). Java Bean classes will house the information retrieved from the JDBC. This will allow Servlets to call objects from the

Java Bean classes, thus insuring a connection between these layers. Lastly, our GUI (Graphical User Interface) layer will showcase the visuals of the program and we can use JSP expression language to retrieve information from the logic layer.

2.2 Modules

2.2.1 Database Layer

The database layer will be responsible for housing all our persistent data from the server in a secure manner. We will be defining the relationships between the stored data. The system will be using MySQL for it's scalability. For more information regarding the data relationships, will be available on Section 4.

2.2.2 Data Access Layer

The data access layer will be managing the data flow between the logic module and the database module. Using JDBC we can retrieve and input data from the database. The data that is retrieved is turned into objects which then is stored as javabean objects. Data Access Layer will also work with session variables from the server and convert them into java objects.

2.2.3 Logic Layer

The logic layer is responsible for doing all the necessary business logic using the java objects. This module will work heavily to supply the GUI layer with the desired information while also updating the data access layer with new information. The logic layer will also use the request and response objects from the server to make the necessary redirection/forward. We will be implementing the logic layer using Servlets.

2.2.4 GUI Layer

The main responsibility of the GUI layer is to create user interfaces so that the users can interact with them. This layer allows the users to login, write their notes, and view/sell notes. The layer will compose of JSP pages that will incorporate HTML, Java, and JavaScript to create a skeleton for UI. The style class will house the CSS code to give our JSP pages visual upgrades.

For the first phase, we will have the following JSP pages: login, homepage, registration page, notes, and account settings. Each of the pages will be interacting with the Apache TomCat to display dynamically generated user web pages. In the second phase, Marketplace.jsp will be implemented. In the future, we will have the system support android and since android works well with java. This will allow most of our code to be reused. For more information regarding the GUI Layer, will be on Section 3.3.

3. Class Diagrams

3.1 Data Table Classes

The system will be using a MySQL database to hold all of the persistent data necessary. This includes information of students, teacher, notes and the relationship between the three. It also includes a market entity that holds two separate lists; one containing all the customers and the other containing all the verified users.

3.1.1 Database Diagram

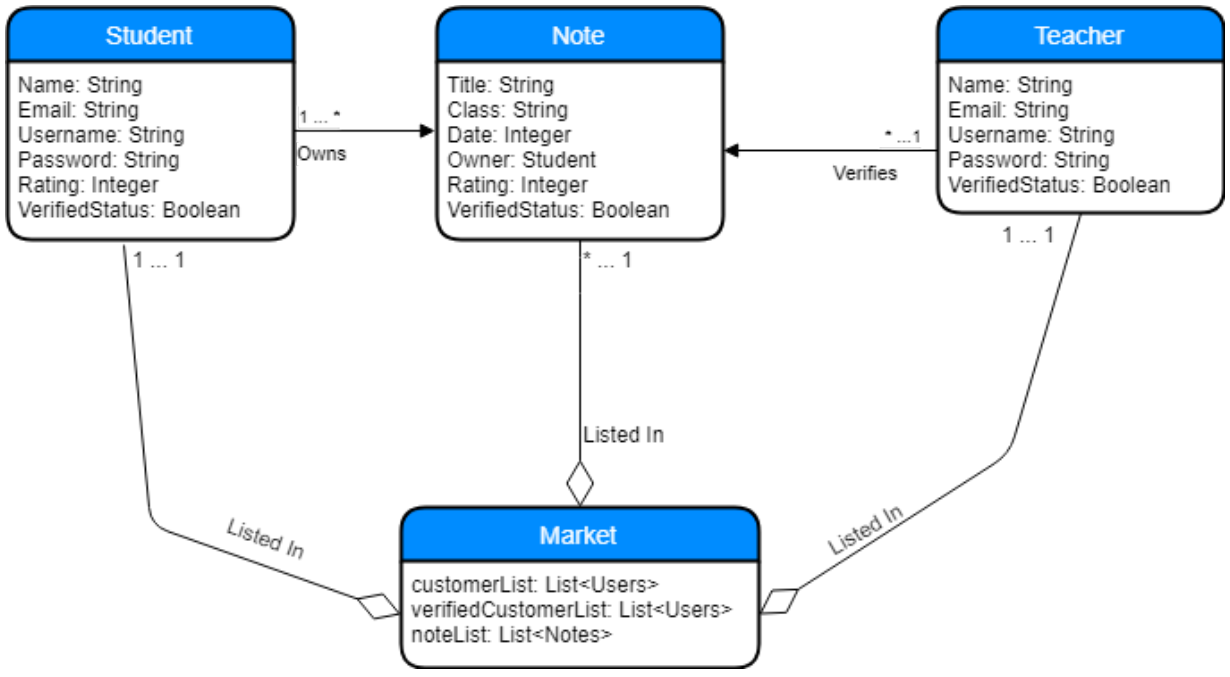

Figure 2: Database Diagram

3.1.2 Diagram Information

The data in the database shall be organized in the following tables and columns

Student: Holds the data for a single student user of the system. Including information such as the students name, email, username, password and whether they are a verified user or not.

Teacher: Holds the data for a single teacher user of the system. Including information such as the teachers name, email, username, and password.

Note: This table will hold all the data pertaining to the notes uploaded to the system. This includes information such as the owner of the notes, the class the notes are for, and a rating on how good the notes are.

Market: This table will hold three separate lists. The first list contains all the customers in the system. The second list contains all the verified users. The third list will contain all the notes for sale.

3.2 Class Information

Classes will be implemented in Java for the data access layer and used throughout the system. The classes will replicate the diagram pictured above by having each table correspond to a different Java class. For example, the student class will have a variable for Name, Email, Username, Password, etc.. The logic layer will then utilise these classes by passing their information up to the GUI layer to be displayed to the user.

3.3 GUI Layer

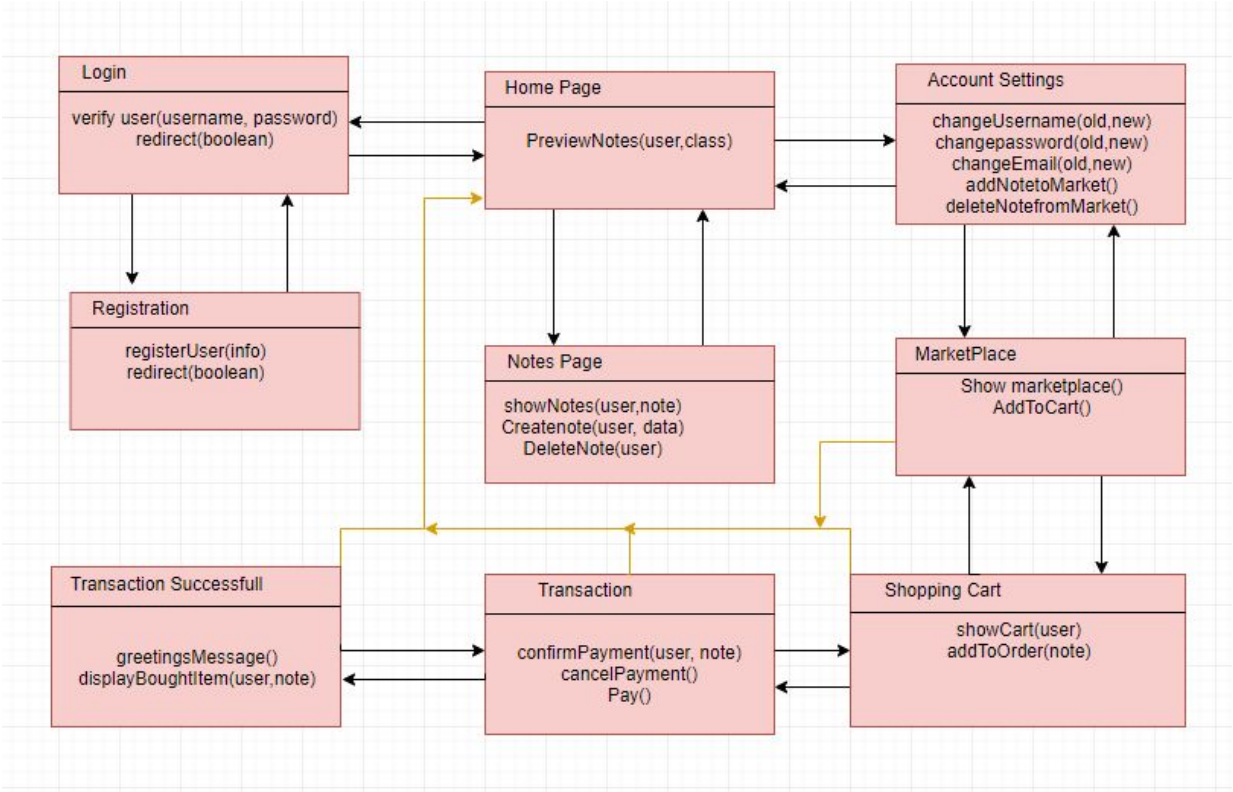

Figure 3: GUI Diagram

For the first phase, the GUI Layer will consists of 5 JSP pages which the user will be able to interact with.The login.jsp: consists of two text fields for username and password and a link to the registration page. If the verifyUser function return false, then error message will be popped.The Registration page has three text fields for: username, password, and email. If the register function returns false, then an error message is popped. The Homepage will preview all the notes which the user has written and also links to all pages. Account Settings.jsp will have 3 text field to change username, password and email. A pop up window will appear if a change has occurred.

For the second phase, the GUI Layer will add 4 JSP pages: MarketPlace, Shopping Cart, Transaction, and Transaction Successful. In addition, the account settings is upgraded to include functions to allow users to upload their notes and add them to market. MarketPlace will show all the notes up for sale with a button to add a note to the shopping cart. The Shopping Cart will allow users to add or delete the items in their cart. Transaction page have text fields for entering credit card information. The transaction page will show what the user what they just bought with a thank you greeting.