CSSECDV: Case Study 1 Specifications

As a software security expert, you and your team have been hired by SECURITY Svcs, a startup software security firm which would be specializing in developing and selling security solutions to various clients. They have recently been in contact with your Instructor, who has endorsed your team to take on this task. With your knowledge and understanding on security and how to implement secure applications, **modify the program by implementing good security practices** that you have observed are needed by the application. For this milestone, only focus on the ones which have been discussed during the class, you may however, add in more security functionalities as you see fit.

The system of SECURITY Svcs is implemented using **Java Programming language** using the **NetBeans IDE**. As the team in-charge of implementing their security features, you have the option to use other IDE's, however you may have to import the SQLite library again for you chosen IDE. After implementing your solution, try to run the program again in NetBeans IDE since Security Svcs would be using that IDE to continue their application. To help you understand their application, a snippet of the file structure is as follows:

Controller	
Main.java	Driver class that initializes the application
SQLite.java	SQLite Driver to be used to access the database in **database.db**
Model	
User	Class reference to the users table in the database
View	
AdminHome.java	JPanel – Contains functionalities for the Admin Role
ClientHome.java	JPanel – Contains functionalities for the Client Role
Frame.java	JFrame – Contains the view panels of the entire application and has a direct reference to the Main.java class to be used to call the SQLite.java methods. Contains the HomePnl which is a JPanel where authorization would be implemented
Login.java	JPanel – Contains the first view of the application. This is where the user would login to the system to be authenticated
ManagerHome.java	JPanel – Contains functionalities for the Manager Role
Register.java	JPanel – Contains the view where new users may register to obtain a role code of "2" designating client privilege
StaffHome.java	JPanel – Contains functionalities for the Staff Role

Table 1. Application File Structure

For the currently iteration, the application is implemented to follow the Model-View-Controller framework and currently has 3 pages; the Login Page, Registration Page, and the Home Page. The functionality for moving from one page to the other has already been implemented.

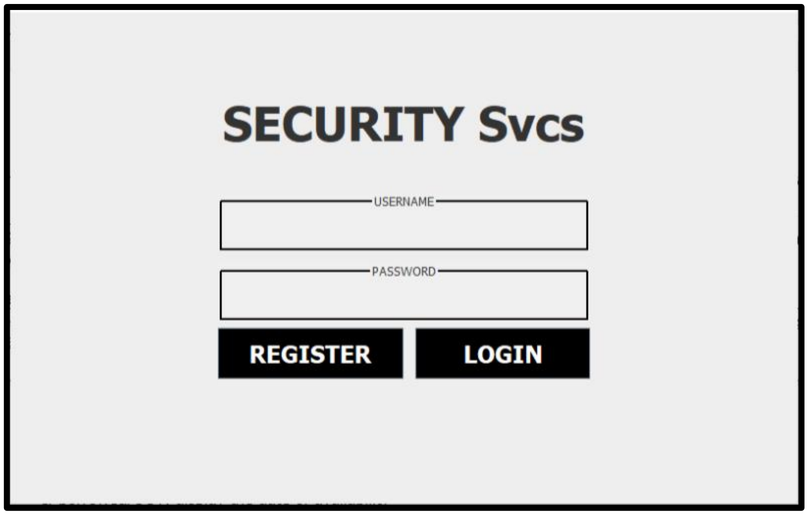

Figure 2. Login Page View

The Login Page contains fields for the USERNAME and PASSWORD input. It also contains buttons for REGISTER and LOGIN. The application currently implements a database service using SQLite Database, which would allow your team to use the application without the need to install MySQL Server or Workbench, at the **database.db** file in the home directory of the software.

As of the current iteration, the application has no implementations whatsoever with regards to verifying user authenticity. For the default users to be used for testing, these are defined on the Main.java page, with their corresponding credentials and role codes.

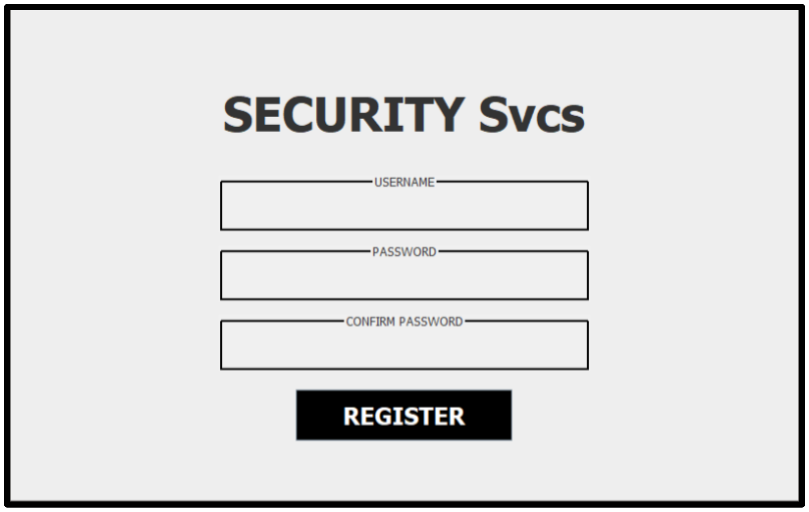

Figure 2. Register Page View

The Registration Page contains fields for the USERNAME, PASSWORD, and CONFIRM PASSWORD input. It also contains buttons for BACK, to redirect the user back to the Login Page, and REGISTER, which allows a user to insert credentials to create an account. Once a user completes the registration fields, he or she would be given a role code of “2” designating a registered user with client privilege.

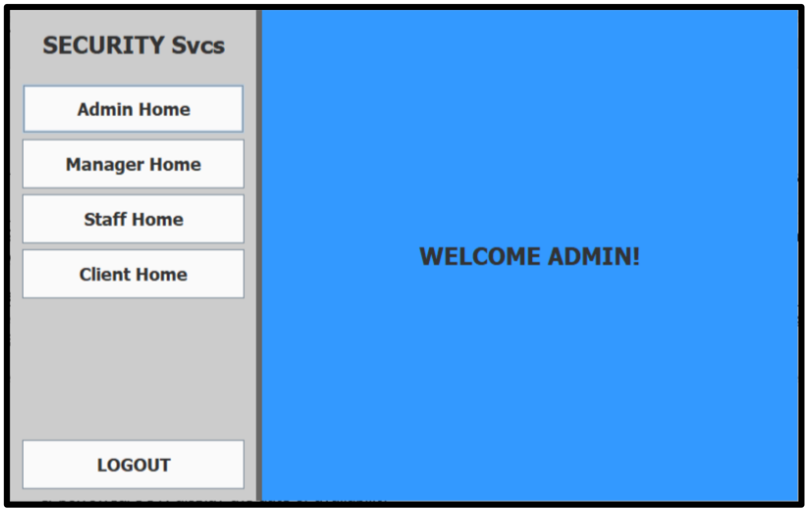

Figure 3. Home Page View

The Home Page contains the main functionality of the application. This is the view that a user would use in order to access functionalities which are available to him or her, depending on the role of that user. The application currently does not restrict user access based on the role, however it should be noted that a user should only be able to access his or her home page, depending on the role.

Users of the system are stored in the database with the following fields; as well as the different user types of the system are designated with the following role codes:

FIELD NAME	DATA TYPE
Id	Integer Auto Increment
Username	String
Password	String
Role	Integer

Table 2. Role Codes

ROLE	CODE
Administrator	5
Manager	4
Staff	3
Client	2
Disabled	1

Table 3. Role Codes

Based on your understanding of the application, implement good security practices for the application of SECURITY Svcs, particularly only on the **Login** and **Register** feature for now. The source code is provided to you in your machine. You are also to submit a document in landscape format; containing a table with 4 columns with the following contents: the security issue that you would address, the Java file name to be modified, the possible vulnerabilities if not addressed, and the solution to be implemented. You are to submit the application code and the documentation online.