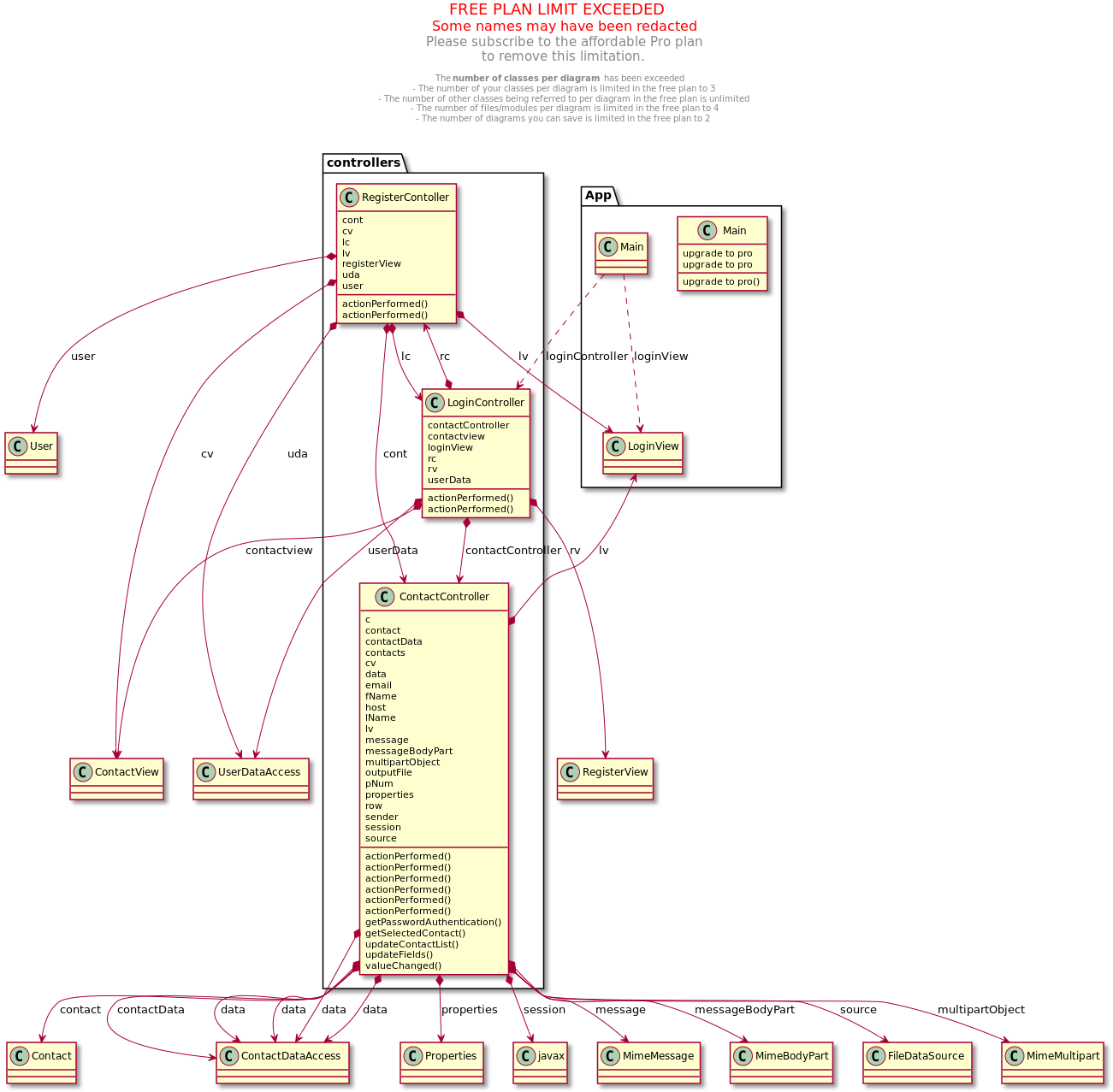
Database Tables:

A screenshot of a computer screen

Description automatically generated

UML Diagram:



Features:

|  |  |  |
| --- | --- | --- |
| Feature | Implementation | Comments |
| Login | Full |  |
| Register | Full |  |
| Input Check on Register and Login | Full |  |
| Add Contact | Full |  |
| View Contact List | Full |  |
| Editing/Updating Contacts | Full |  |
| Searching for Contacts | Full |  |
| Password Hashing | Full |  |
| Contact View input field checks | Full |  |
| Logout option | Full |  |
| Create CSV file of all contacts | Full |  |
| Email CSV to email of user choice | full | There is an error that pops up don’t worry the program still runs fine and will email the recipient. |
| Delete contact | Full |  |

Explanation of Main Code:

1. Main.java

* Purpose: This file typically contains the main method which is the entry point of the Java application. It likely initializes the application, sets up the main window, and ties together different parts of the program.
* Key Components: Look for initialization of the main application window, setting the visibility of the UI, and possibly loading initial data or configuration settings.

2. ContactController.java

* Purpose: Manages interactions between the view (ContactView) and the model (Contact and ContactDataAccess). Handles user actions like adding, updating, and deleting contacts.
* Key Components: Action listeners for UI elements (buttons, lists), methods for updating the UI based on model changes, and invoking data access methods.

3. LoginController.java

* Purpose: Handles the login functionality. This controller manages user authentication, directing the flow based on user input at the login screen.
* Key Components: Authentication logic, UI updates for login status, and transition to other views (like the main application view) upon successful login.

4. RegisterContoller.java

* Purpose: Manages the user registration process. Similar to the LoginController but focused on creating new user accounts.
* Key Components: Methods for collecting user input, validating it, and interacting with UserDataAccess to create new accounts.

5. User.java

* Purpose: Represents the user entity. This class likely contains user-related data such as username, password, and any other user attributes.
* Key Components: Attributes, getters and setters, and possibly some business logic methods related to users.

6. Contact.java

* Purpose: Represents a contact entity. This class contains data related to a contact such as name, phone number, etc.
* Key Components: Contact attributes, getters and setters.

7. UserDataAccess.java

* Purpose: Provides data access methods related to users. This includes saving new users, retrieving user data, and possibly authentication methods.
* Key Components: Database connection setup, SQL queries for user operations.

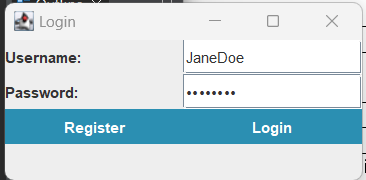
8. ContactDataAccess.java

* Purpose: Handles all data access operations for contacts. Manages database interactions necessary to add, update, retrieve, and delete contacts.
* Key Components: SQL queries, database connection management, methods for transforming database rows into Contact objects.

9. util.java

* Purpose: This class only holds Hashing passwords
* Key Components: password hashing

Feature Screenshots:



A screenshot of a login box

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

References:

PhoneBookExample.zip from class, Professor Adu Baffor

JavaMail API, this is for allowing email to be sent

Activation.jar, this is also for email to be sent

SMTP-1.6.2.jar, this is the temp server made to establish a connection with the recipient of the email.