**CMPE 131 Sec 04**

**Team 4**

**Andrew Schmidl, Ricky Singh, Prajjwal Singhal**

**Software Requirements Specification**

**For**

***Let’s Chat***

[**https://young-shelf-23845.herokuapp.com/**](https://young-shelf-23845.herokuapp.com/)

[**https://github.com/RickyRajinder/ChatApp**](https://github.com/RickyRajinder/ChatApp)

**Version: 1** **Date: (05/13/2018)**

**1. Introduction & Scope**

The assignment defines a chat application, which should be updated in real time and should allow the user to send and receive text messages to other users. The app should also be able to store the details of the user and the text messages sent by the users in a database, and users should have their own unique username and a password of their choice that would allow them to login through a web browser.

**2. Program Description**

**2.1 User Requirements**

* Program will send text messages between users
* Users will be able to signup and login
* User Interface of the application should be user friendly
* Program will be fast and should be able to send messages instantly

**2.1 System Requirements**

* Program will be internet based
* A database is required to store text messages and user details

**2.1 Functional Requirements**

* Be able to send messages back and forth
* Signup using email, username, and password
* Authenticate users using username and password
* Information of one user should not be visible to other user
* Password should not be visible while user is typing it in
* There should be a database in which the user details and messages are stored
* Multiple users in one chat
* Update chat in real time
* App should run on any major browser (Chrome, IE, Firefox, etc)

**2.2 Non-Functional Requirements**

* Front end of the project should be in HTML
* Backend of the project should be in PHP
* Use XAMPP for server
  + Apache server
  + SQL database

**2.3 Constraints**

* A Users account detail should not be visible to other users
* User should not be able to login without a valid username and password

**2.4 Assumptions**

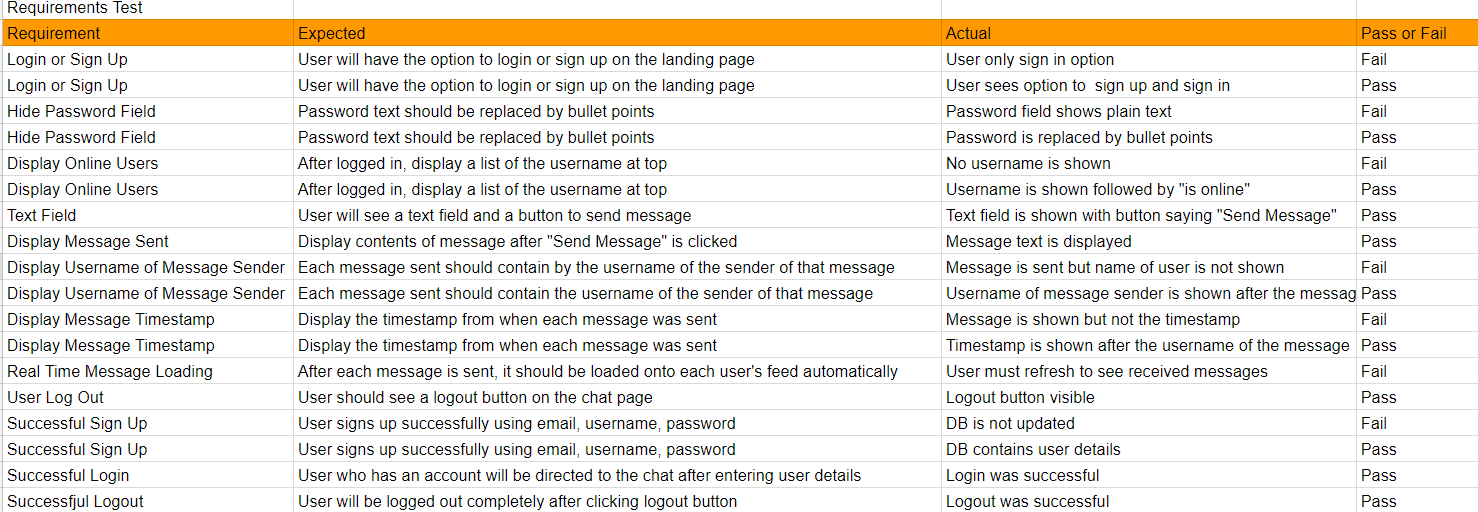
* User has a valid user id and password
* User should have internet connection

**3. Performance Measures**

* Ease of website access
* Bug Free code
* Time taken for the a text message to send
* Chat updating
* Security

**4. Testing**

* Program should be tested with valid and invalid username and password combinations
* Should be tested on whether the password is hidden or not
* Should be tested to see if all kinds of text messages are sent
* Program would be tested to check if the user is safely able to log out safely or not



**5. Logic Description**

1. Code starts at the login page
2. User can input username and password for login
   1. If validated - would be sent to a chat page
   2. Else - Show “404: Combination does not exist”
3. User can input registration information - email, username, password
4. User can access the chat page after login
   1. User can open chats with other users
   2. User can type out messages and send messages to other users
5. Log out option
   1. User logs out

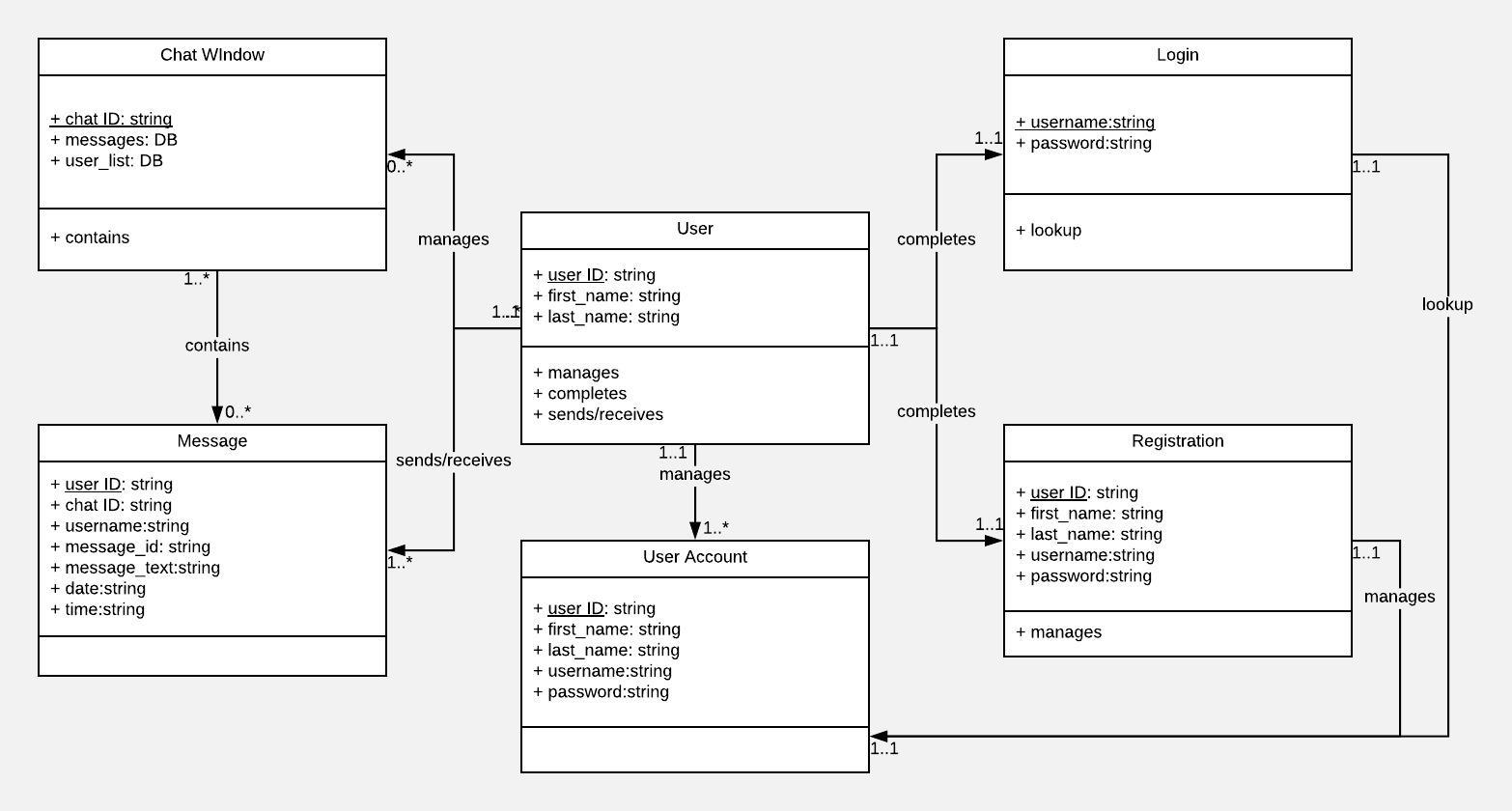
**6. System Architecture**

**Low Level Architecture: Model-View-Controller (MVC)**

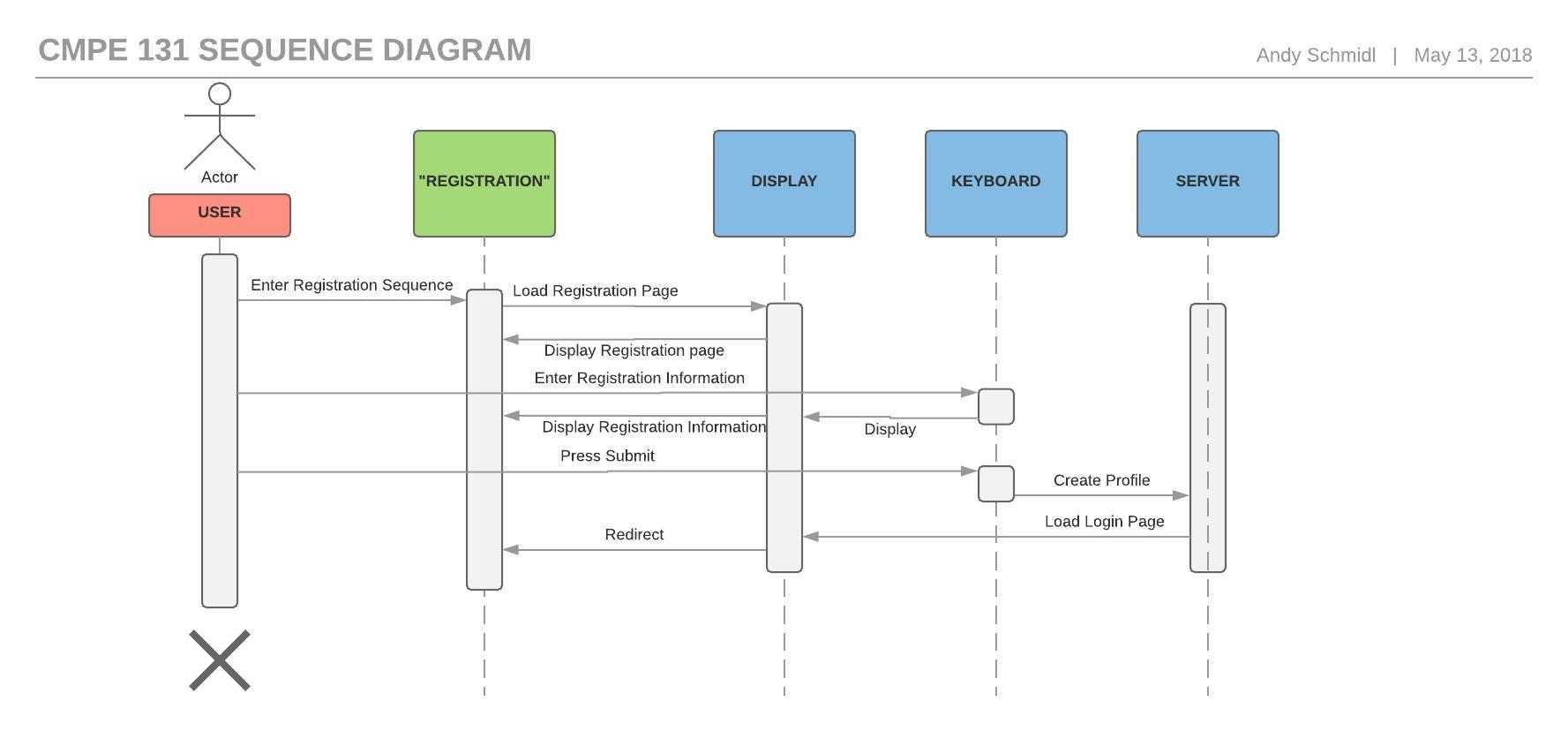
* Model:
  + Apache server
  + SQL database
* View:
  + HTML code
* Controller
  + PHP code

**Interface Design**

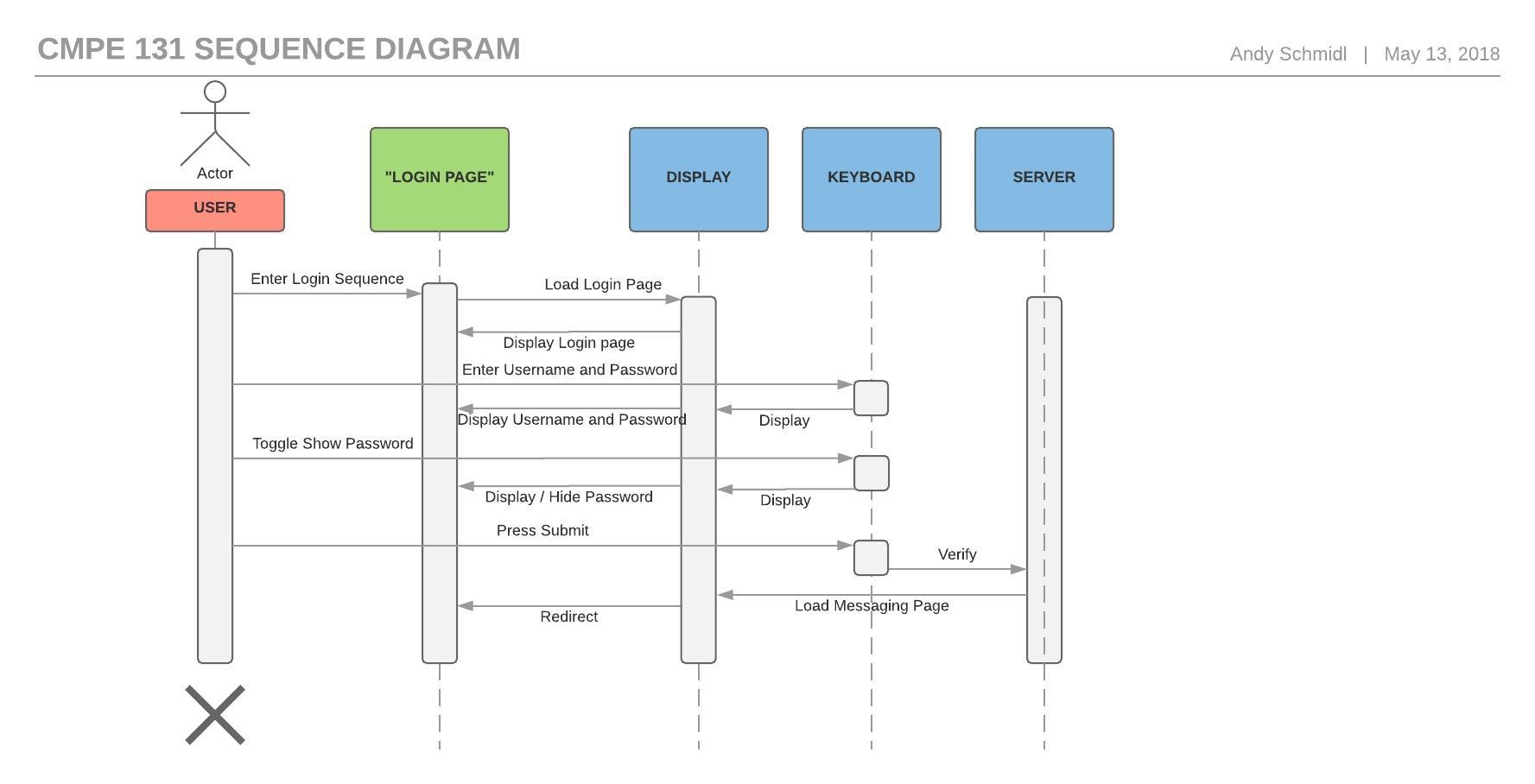
* Login Page
  + Login
    - Username Field
    - Password Field
    - Show/Hide Password Option
    - Login Button
  + Sign Up
    - Email Field
    - Username Field
    - Password Field
    - Signup Button
* Messenger Page
  + Chat Window
    - Message Field
    - Submit Button
  + Logout Button
  + Username Display

**System Architecture: Class Diagram**

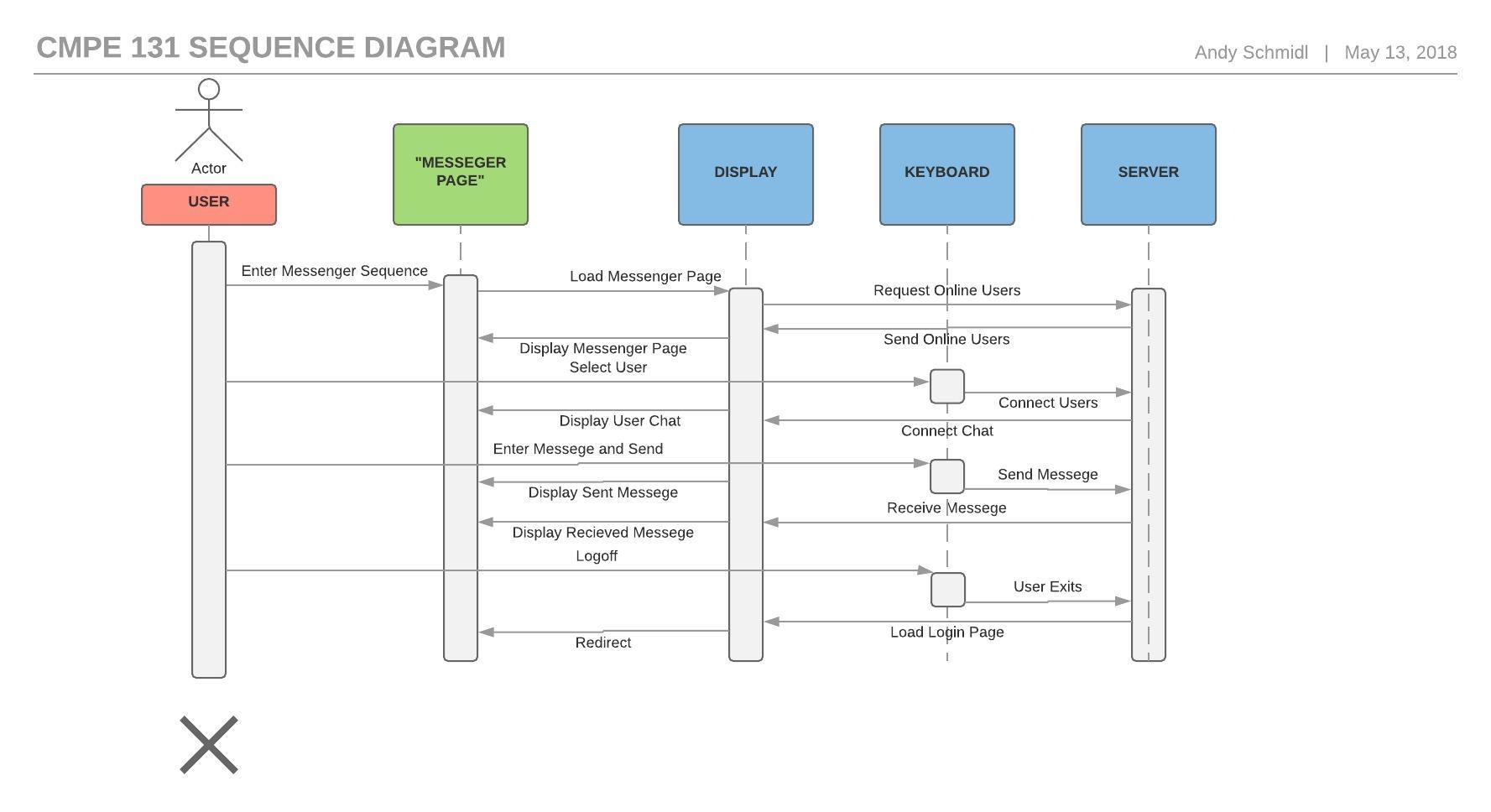
Class diagram displays the relationships between the user and the different objects in the system.

**System Architecture: Sequence Diagrams**

The first sequence diagram details the registration sequence from the User perspective. The scenario name is “Registration” and the objects are the display, keyboard and server.

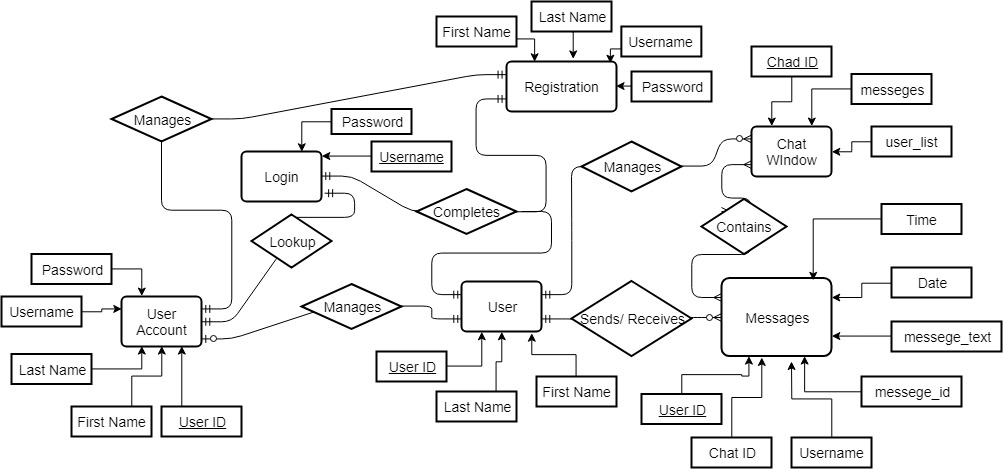


The second sequence diagram details the login sequence from the User perspective. The scenario name is “Login Page” and the objects are the display, keyboard and server.

****

The third sequence diagram details the registration sequence from the User perspective. The scenario name is “Messenger Page” and the objects are the display, keyboard and server.

**System Architecture: ERD Diagram**

The ERD diagram details the system and the relationships between different objects and actors in the system.

**Conclusion**

By following the software engineering incremental build process, we were able to design, develop, and implement a working in-browser chat application. We reached our goal of building a functioning online messaging application. We met all of our functional and non-functional requirements except for one; this was real time chat updating. In order to further improve our application, we would implement websockets in PHP in order to update the chat window instantly without the need to refresh the page to see a new message. Additionally, we can improve the application by implementing CSS styling for a more visually appealing login/signup page and chat page. We can also enable users to edit their user information and invite new users to chat on the app via email.