

Software Design Document

Class Chat

1. Introduction

Class chat is an online messaging software which is meant for educational purposes during lectures or outside classrooms. It will allow students to participate in online discussions in or outside the classroom and with the inclusion of the Professor inside the chat. Class chat will promote healthy and productive learning environment for students and an user interface which is easy to navigate. Class chat is a web based software and can soon replace other classroom discussion forums which are not as fast and student friendly as Class Chat.

2. Design Considerations

2.1 Assumptions

As the participation in a class discussions is not location based, the identity of the user will be based on the School email address. The functionality of the software will depend on internet and user participation.

2.2 Constraints

1) Retrieving data- Chat History will be saved on the server. Therefore, a student may join the chatroom late but still be able to view what occurred during the time they were not involved. Admins/Backend workers will not be able to view chatroom conversations or the participant's info, unless Law requires us to turn over such data.

2) Security verification - We verify users with a valid email address and password they created when they signed up on the app.

2.3 System Environment

Class Chat is currently implemented for IOS devices only.

3. Architectural Design

3.1 Overview

Responsibilities:

Class	Responsibility
User	Manages user account information.
Class	Manages information for each class.
Chat	Manages information and functions for each chat room.
Server	Access server for sending or retrieving data.
User Database	Store data for users.
Class Database	Store data for classes.
Chat Database	Store data for each chat room.

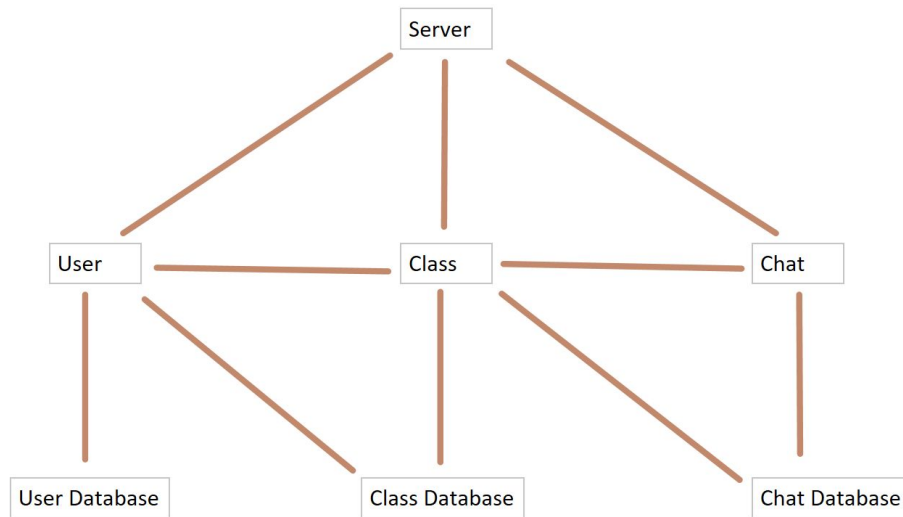
Actions:

Class	Action
User	Create user, delete user, set name, change name, security, verify.
Class	Create class, set class name.
Chat	Send message, receive message, show chat history, send/receive images
Server	Connects UI and database
User Database	Store, retrieve, search, delete.
Class Database	Store, retrieve, search, delete.
Chat Database	Store history, retrieve history.

3.2 Rationale

We chose this architectural design because it is simple and straightforward. Each class is created to be in charge of a specific aspect of Class Chat. The most important components of class chat are users, classes, and chat rooms, each can be managed using User, Class, and Chat classes.

3.3 Conceptual (or Logical) View



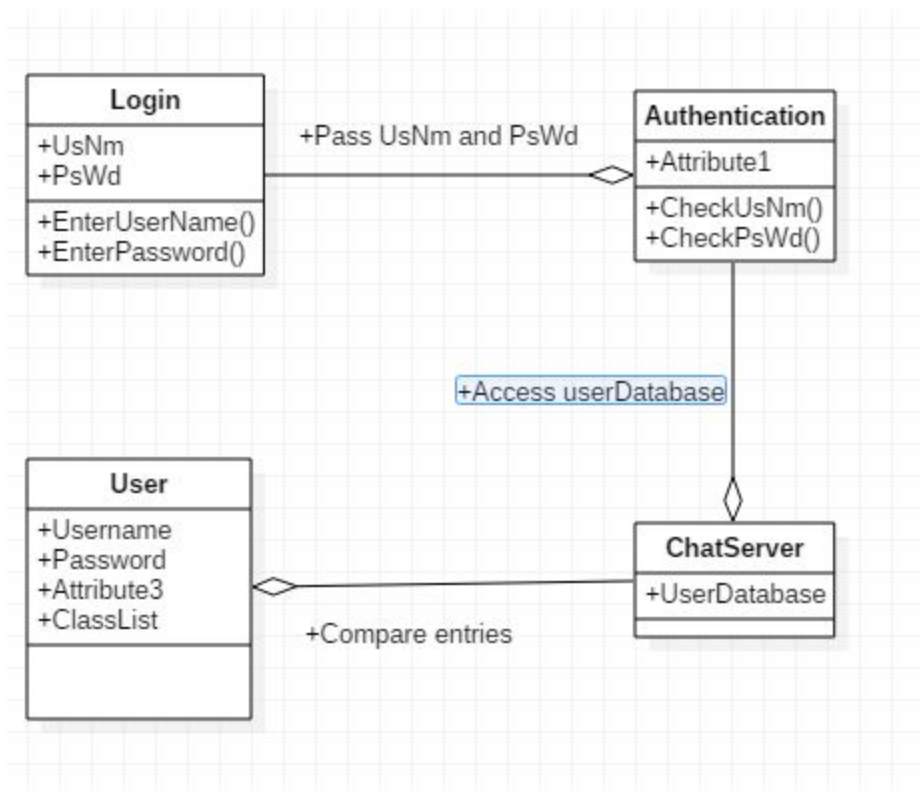
User, Class, and Chat are all connect to Server for upload and download data. User has access to Class Database to search for their class, and has access to a particular class he/she picks. Each class contains a Chat. A class can access its chat room and the chat room's database, which contains chat history. User, Class, and Chat are each stored in the corresponding database.

4. Low Level Design

Describe the low-level design for each of the system components identified in the previous section. For your component, you should provide details in the following UML diagrams to show its internal structure.

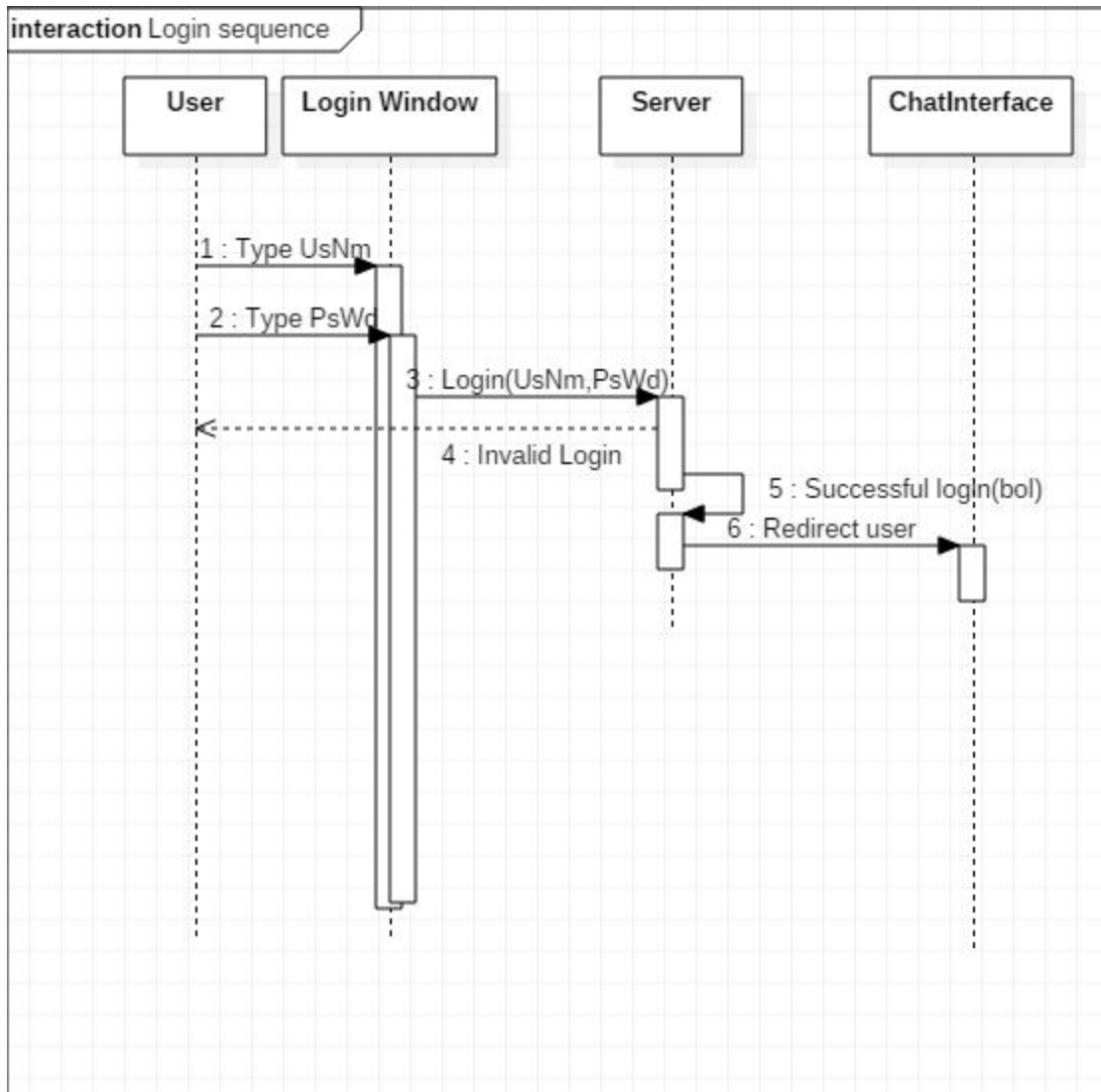
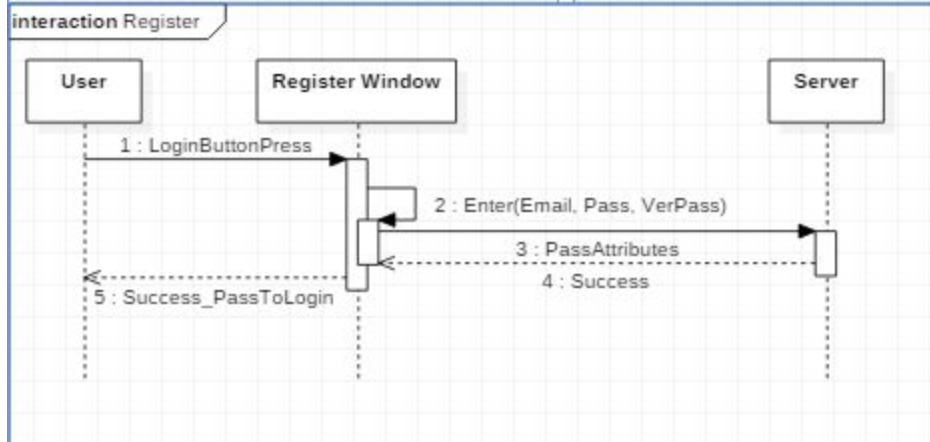
4.1 Class Diagram

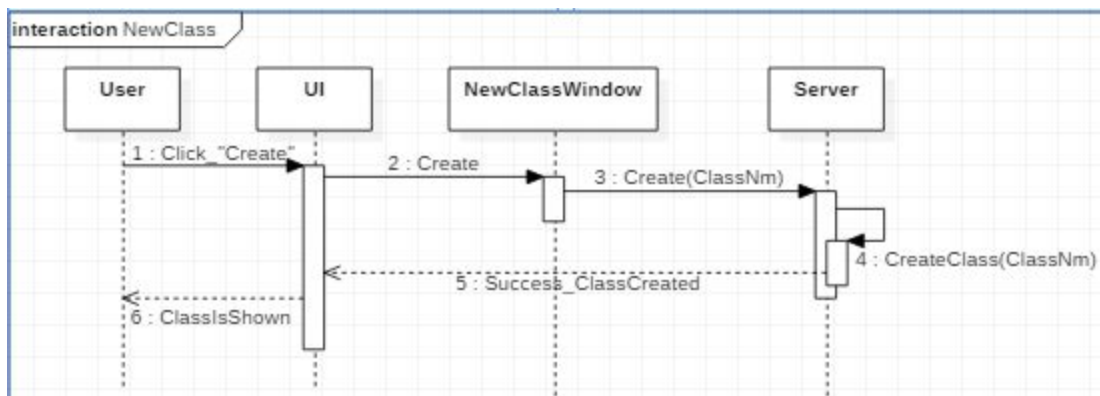
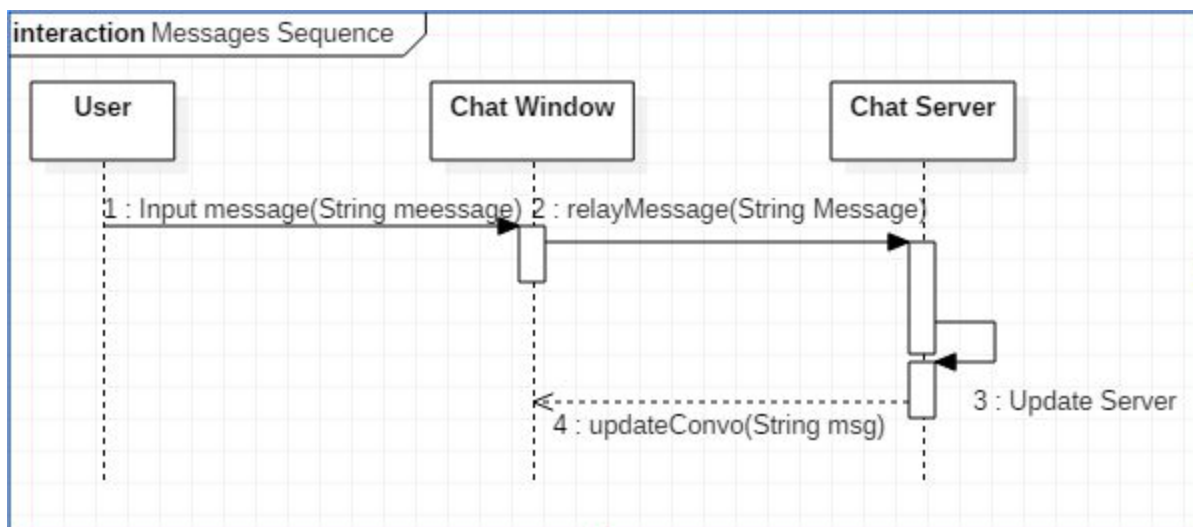
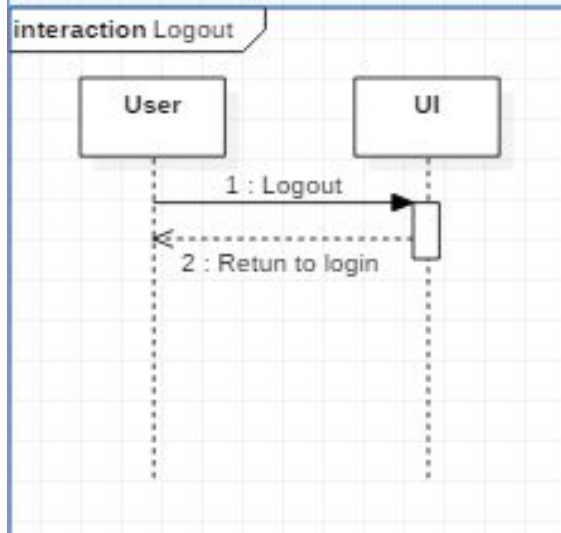
In the case of an OO design, the internal structure of a software component would typically be expressed as a UML class diagram that represents the static class structure for the component and their relationships (association, aggregation, generalization, etc.)

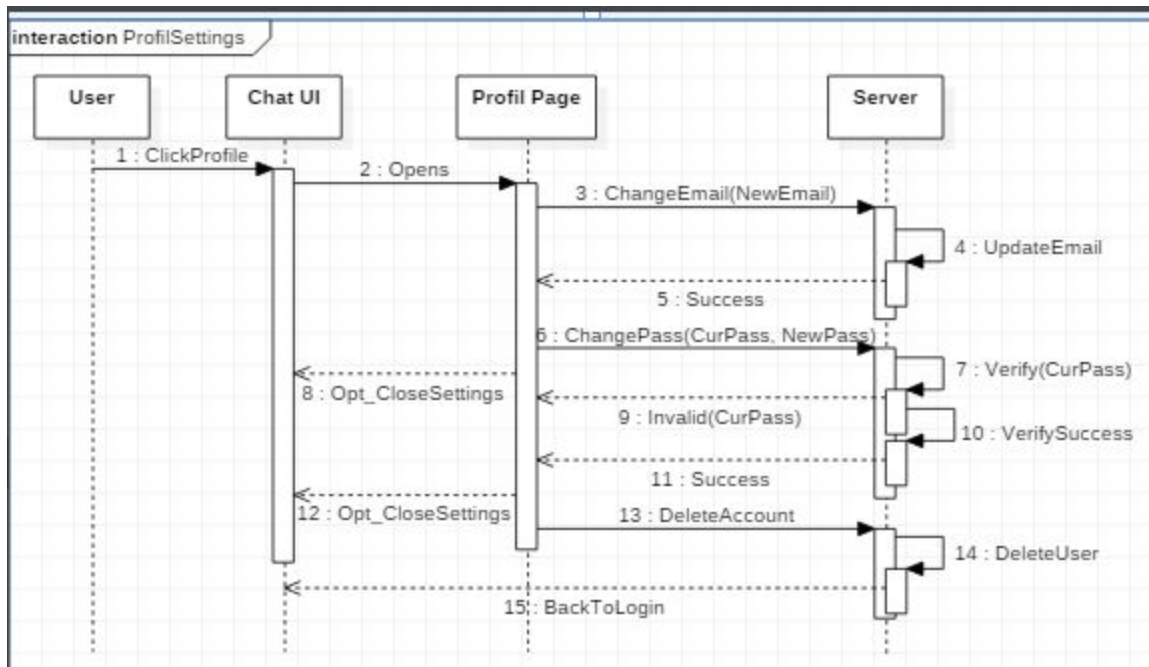


4.2 Sequence Diagram

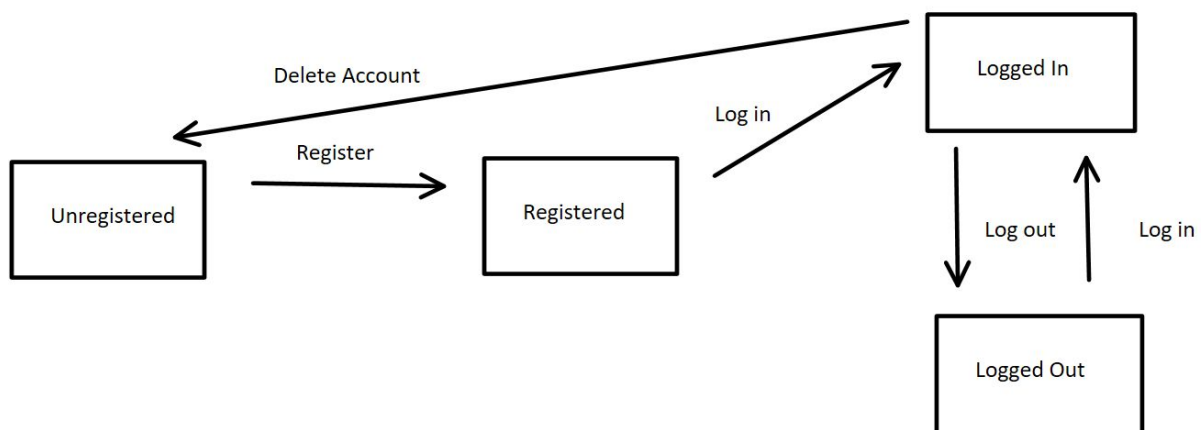
This diagram shows details of how and in which order objects within a component interact with one another.







4.3 State Diagram



5. User Interface Design

