CAS 703
Low-Level Design Specification

1 Introduction

This document provides a low level design specification for QuickMessenger app by providing details in a systematic manner to support project management and system development processes. This document first captures Controller Classes, sequence diagrams, and finally Detailed Case diagram. The document is intended for software engineers and developers.

1.1 Purpose

QuickMessenger subsystem allow a user to register him/herself into the application, secure own preferences, invite others as a peer to peer or a group chat either in the form of text, allowing for exchange of multiple media/documents whilst allowing for audio/video conferencing as well. With an appealing and minimal effort UI and back end which is secure, it prioritises a user values. Ease of use, re-use as much as possible and creating a design which is easy to maintain from the onset. Model View controller are used to create overall functionality in this application.

1.2 System Description

QuickMessenger mobile application designed specifically for iOS and Android platforms allows registered users to communicate efficiently using text, as well as audio visually and share various forms of objects such as multi-media and documents. Users can communicate in a secure manner either peer to peer or in a group. Group management is a core component of this system and provides a social platform facilitation as an alternative to Social Media in a secure manner.

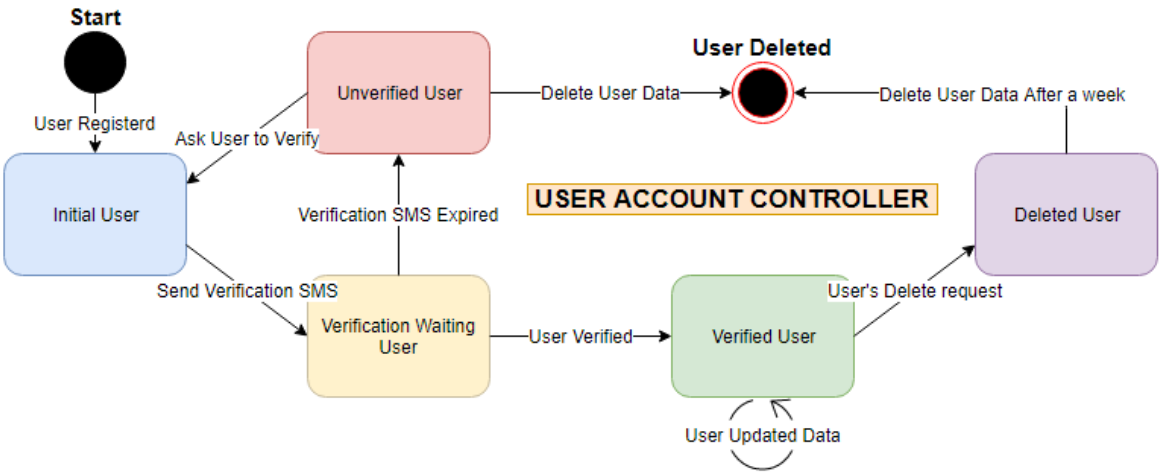
As opposed to other kinds of messaging, instant messaging is interactive with near real time conversation; this requires very low latency in the delivery of messaging.

1.3 Overview

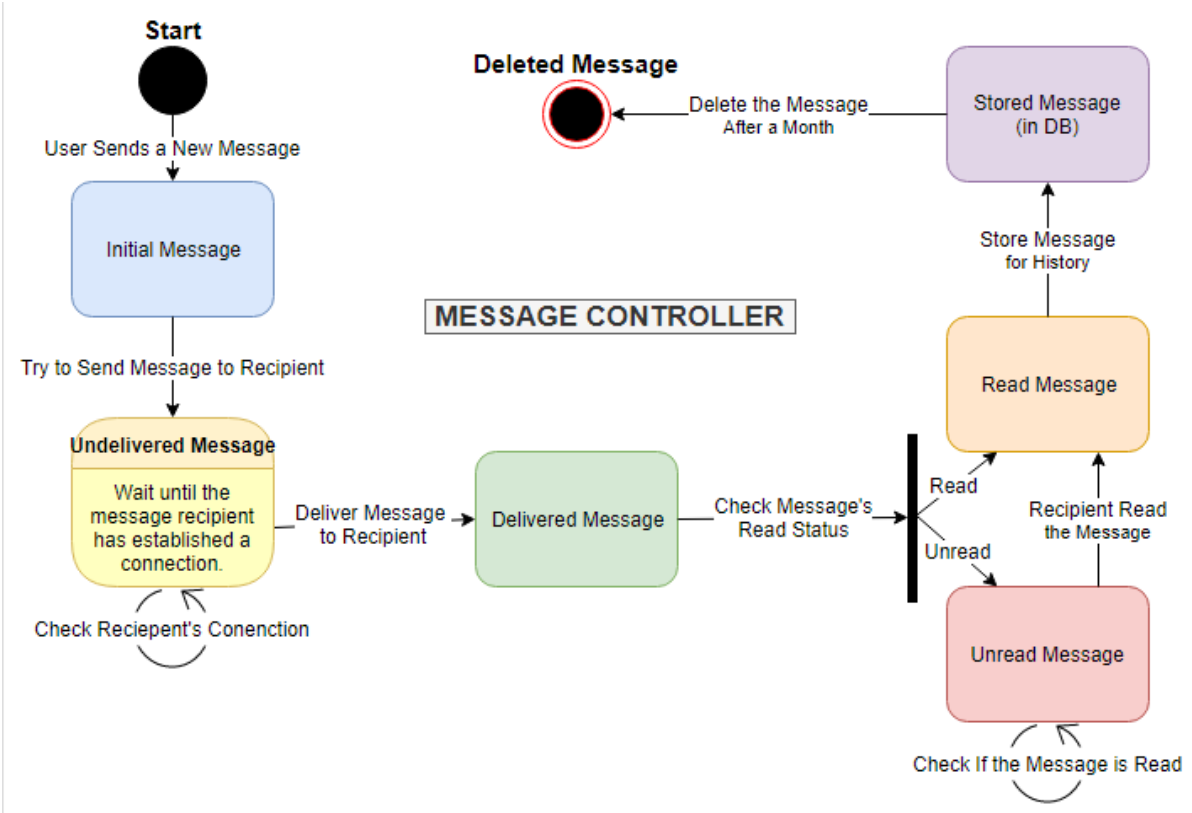
The document follows a layering process by first capturing State Charts for Controller Classes for QuickMessenger are described. For each sub-system there are high-level descriptions as well as the design-decisions made during the design process. UML Class-diagrams and sequence-diagrams have been added to show the interactions between the class-diagrams. Team initially designed UML class diagrams before we began our requirements that we gathered at the start of the project.

2 State Charts for Controller Classes

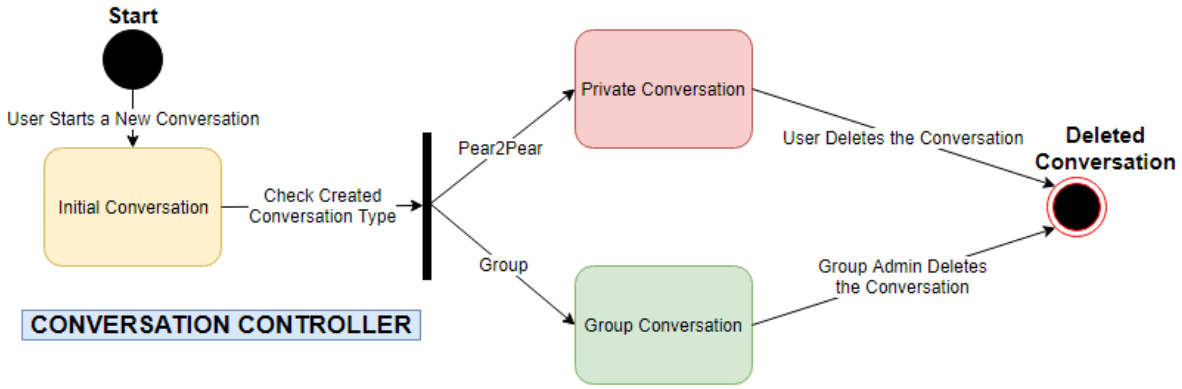
2.1 User Account Controller



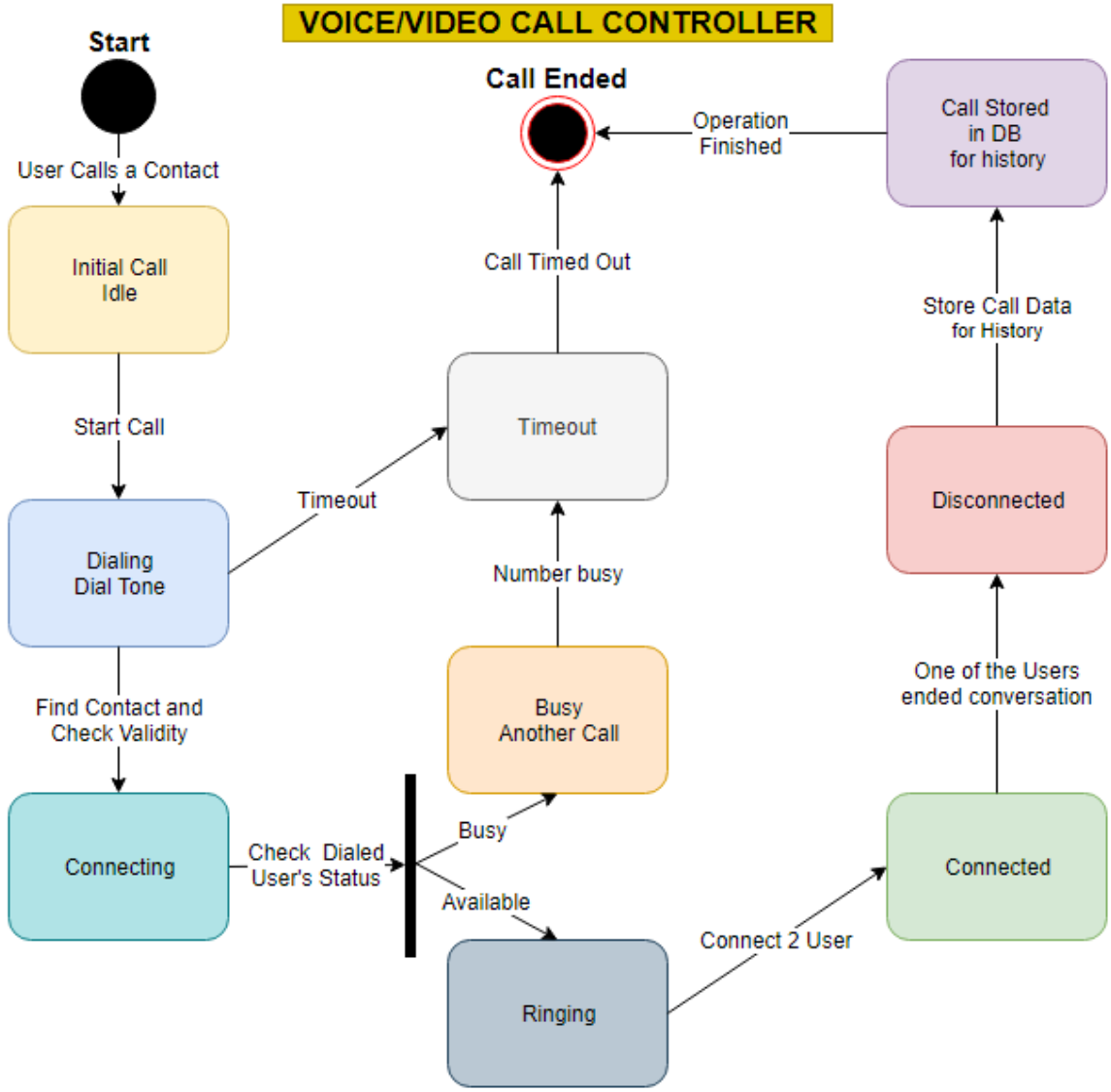
2.2 Message Controller



2.3 Conversation Controller

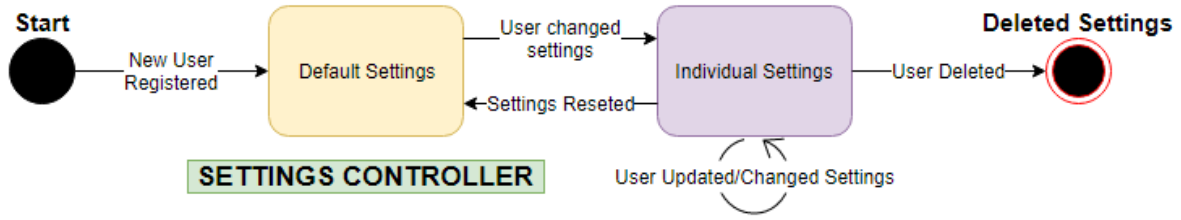


2.4 Voice/Video Call Controller



2.5 Settings Controller

This class applies to both Account Settings and Chat Settings.



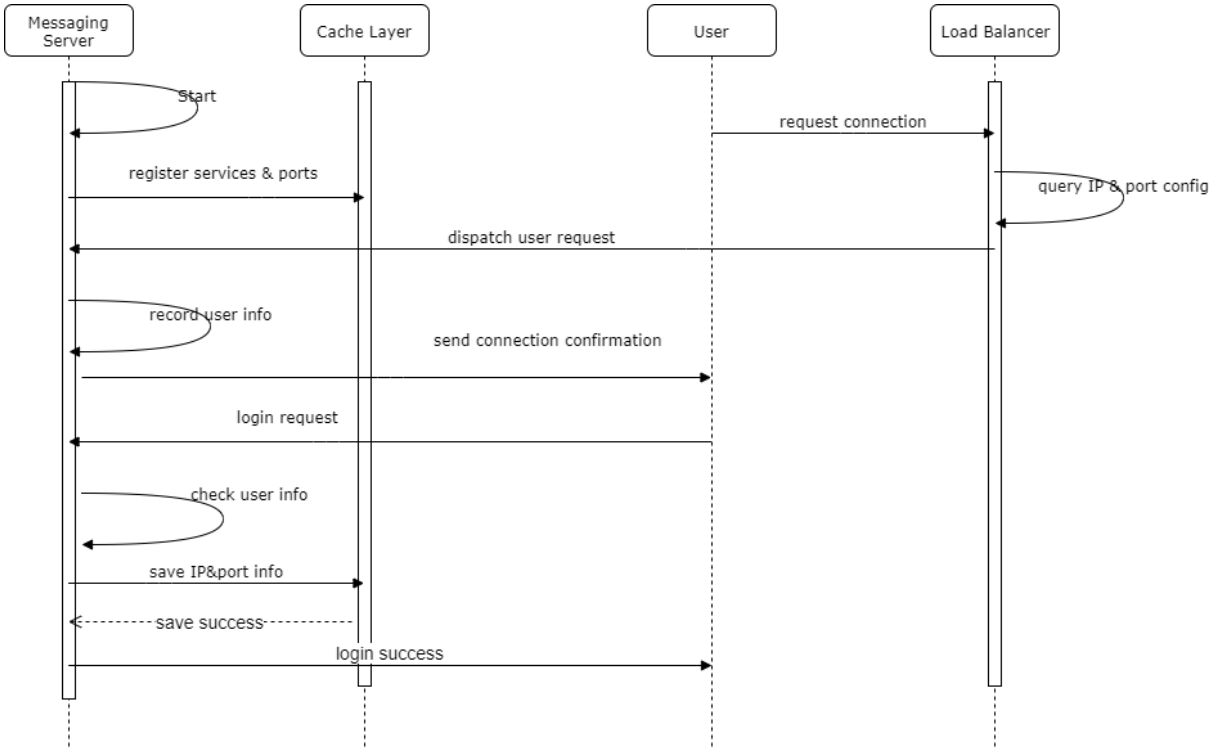
3 Sequence Diagrams

3.1 System initialization & User login

The following sequence diagram shows system initialization process and user login process. When the IM servers start, they register some important services, for example user information lookup, to the cache layer at server side. When a user tries to login into the system, it takes two steps: 1) Connection setup between client and IM servers; 2) User information check at the server side.

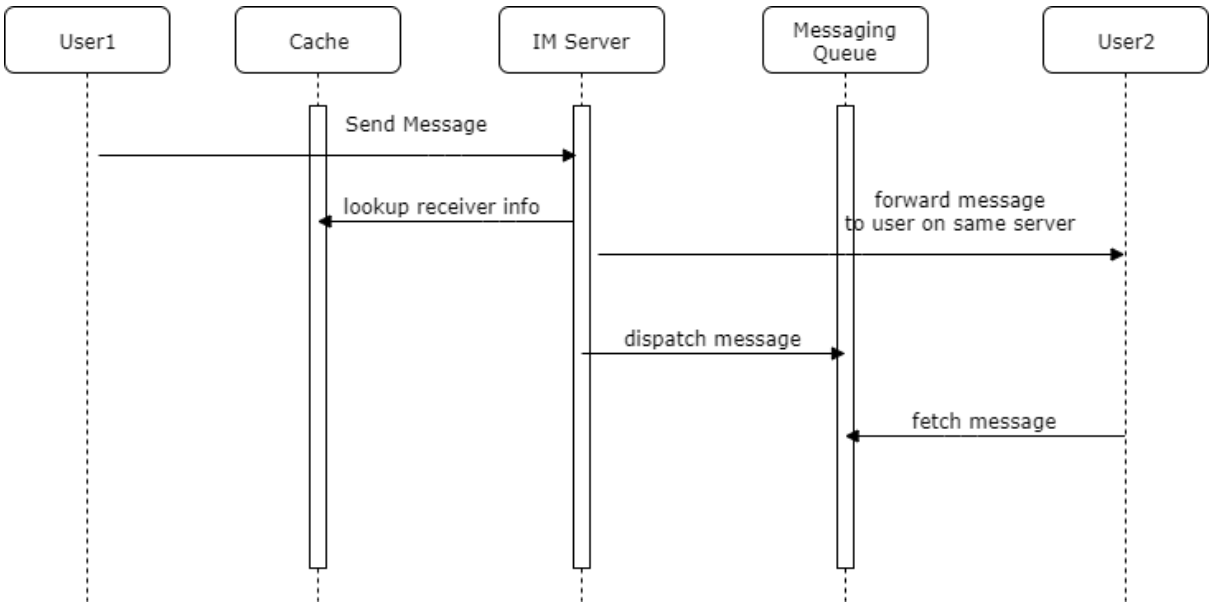
The connection setup is initiated by a user, the request is handled firstly by the load balancer, which assigns the request to a certain IM server. After recording the user's information, the IM server will send back a connection confirmation to the user's client.

After receiving the connection confirmation, the user can request login to assigned IM server. The IM server is responsible for validating the user's information through back-end database. If the validation was successful, the user-server binding information would be saved to the cache layer. Finally, a login confirmation would be sent to the user.



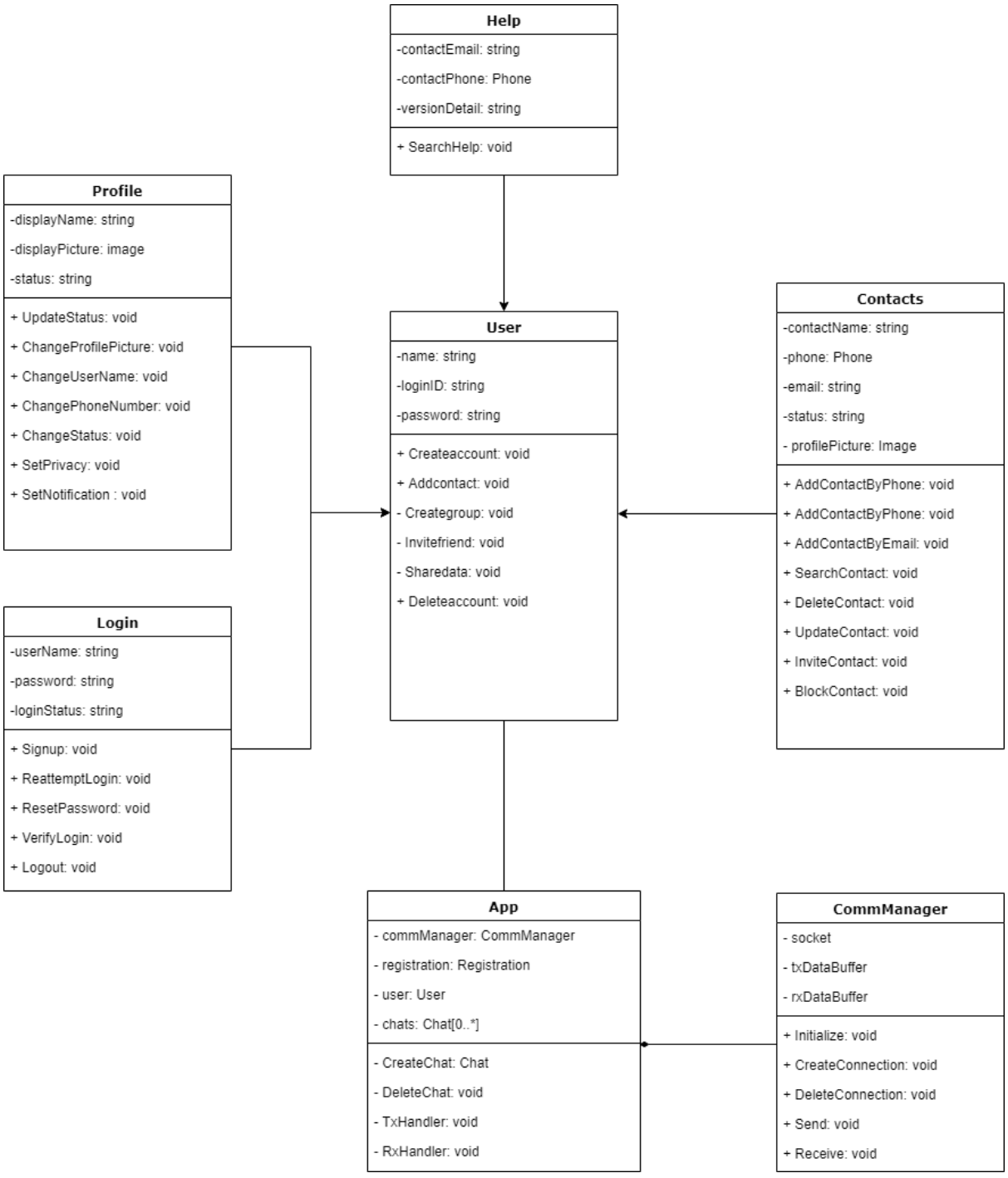
3.2 Messaging

When two users communicate through the messaging subsystem, there are two scenarios: 1) The two users are assigned to the same IM server by load balancer; 2) The two users are located on different IM servers. In the first scenario, the IM server forwards the messages directly to receivers. In the second scenario, different IM servers transfer messages through a messaging queue.

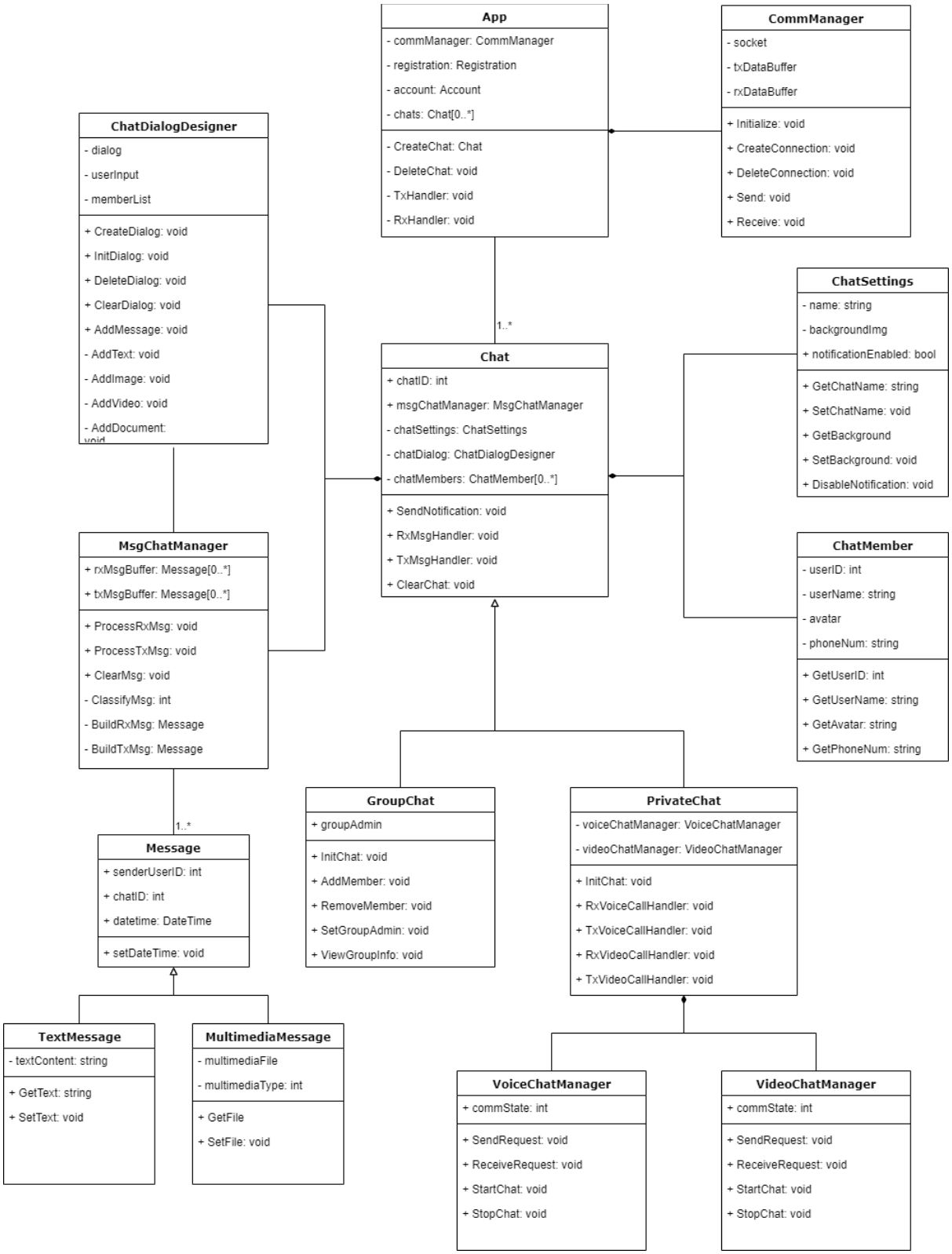


4 Detailed Class Diagram

4.1 User Account



4.2 Messaging



A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.

Group Members	**Work**
Sajid Rahim	Section 1 and 4.1 (Account Management)
Baran Kaya	Section 2
Hong Sun	Section 3
Xiaodong Xu	Section 4.2 (Messaging)