

Daniel Ochoa Morales
Harrison Fisher
Jacob Le Nguyen
Nicholas Khine
Brian Le

Design Specification Document

1.1 Introduction

1.1.1. Goals and Objectives

- This document describes the important aspects of the implementation of clients sending messages to other clients via a server.

1.2. Statement of Scope

- Decisions in this document are made based on the following priorities (most important first): Usability, Maintainability, Usability, Portability, Efficiency.

1.3 Software Context

- Messages will be stored as a Message object then be sent to the server from a source client. The server will save the messages in a log and then send them to a destination client as another Message object.

1.4 Major Constraints

Issue 1: Where would the Message be stored in the server

Option 1.1: Store the information in the Message object inside a 2d array

Option 1.2: Store the information in the Message object inside another Message object

Decision: Option 1.1

2.0 Data Design

2.1 Client Side

- The client will send the text inside the textfield by pressing the send button and pass the object via **sendMessage**
- Messages sent to the client will be displayed in the gui.

2.2 Server Side

- All messages will be saved into a file.
- All users will be saved via **addUser** and removed via **removeUser**

3.2 Description of Client

- Client is either a regular user or an IT staff member

3.2.1 Client processing narrative

- Client Driver initializes the Login object for user validation.
- Once login authentication is validated, Client Driver initializes the Chat Window object display gui
- Client can choose to an existing conversation to view or create a new conversation
- Client would then be able to send a message to that conversation

3.2.2 Client Interface Description

- Login gui will have username textfield and password textfield.
- Gui interface will hold a conversation list on the far left, the conversation on the right, and a textfield for sending messages on the bottom.
- The conversation will show all messages from the conversation as output.

3.2.3 Client Processing Details

- **Login Window** will be initialized and authentication will be sent to the Server and return whether information is correct. Incorrect login will result in new Login authentication window
- Client would create a new conversation and when prompted with the recipient's id/name. Invalid recipients would return no response. Valid recipients would create a new conversation
- Clients could highlight an existing conversation and an actionlistener would call getChatHistory and display output on the conversation frame on the right.
- Client would type intended message on the message text field and press on the send button to send to conversation. The message inside the textfield would be converted into a **Message** object and passed to the server.

3.3 Description of Server

- The server will maintain a record of all messages sent.
- The server will maintain a record of all users.
- The server will direct messages from client to client(s).
- The server will verify user login information.

3.3 Software Interface Description

3.3.1 External Interfaces

- Login: Validates that username and password are correct format and sends login data to server for further processing.
- SendMessage: Send a message to other user(s)
- GetConversation: Retrieve a conversation between users
- CreateUser: Validate login information and send to server
- DeleteUser: Validate username and send to server
- GetChatLog: Request the chat log from server

3.3.2 Internal Interfaces

- AuthenticateUser: Logs a user in if they exist
- LoadData: When server runs, user and message data is loaded from file.
- SaveData: Save data to a file.

3.3.3 Human Interfaces

Chat Window

- Conversation History: Displays a conversation between multiple users.
- Conversation List: Shows a list of past conversations that can be displayed if clicked.
- Message Textfield: Stores input for message to be sent

- Send Button: Sends message from message textfield

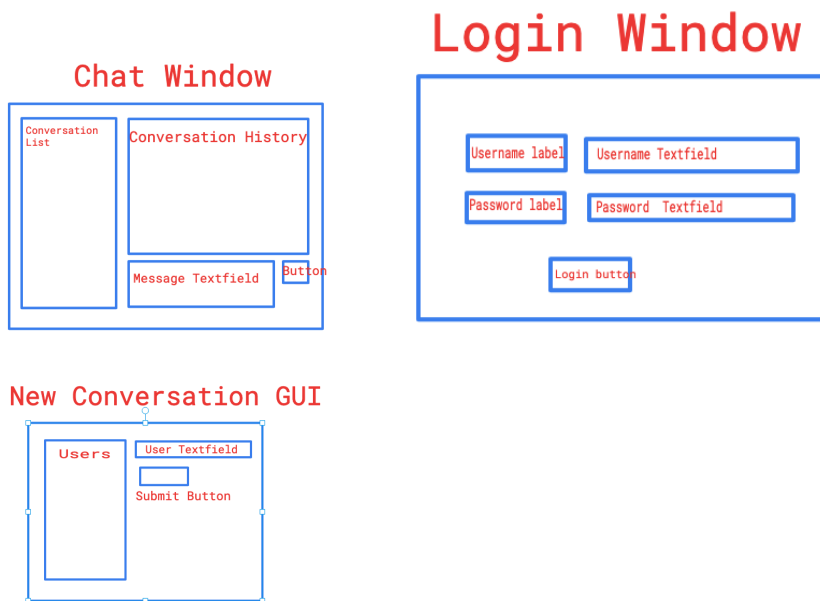
Login Window

- Username Text Field: Stores input for username
- Password Text Field: Stores input for password
- Login: Attempts to login given data from Username and Password text field.

New Conversation GUI

- Conversation name textfield: Stores input for conversation name
- User textfield : Stores input for user to add to conversation
- Submit Button : Loads Chat window with the selected user in conversation History.

4.0 User Interface Design



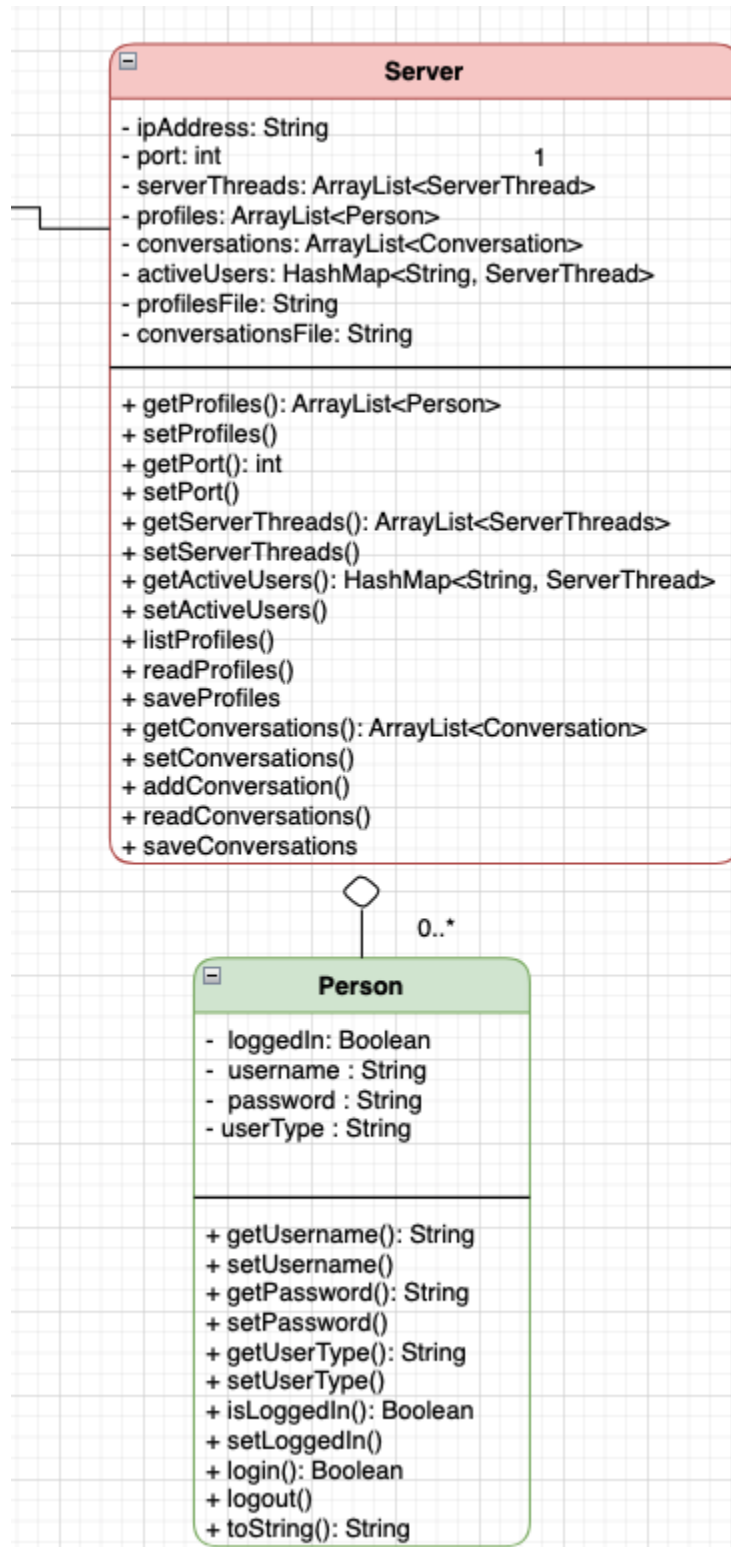
5.0 Restrictions, limitations and constraints

- Only IT can access user logs.
- Only IT can create and remove users.
- Only text messages will be sent.

6.0 Testing Issues

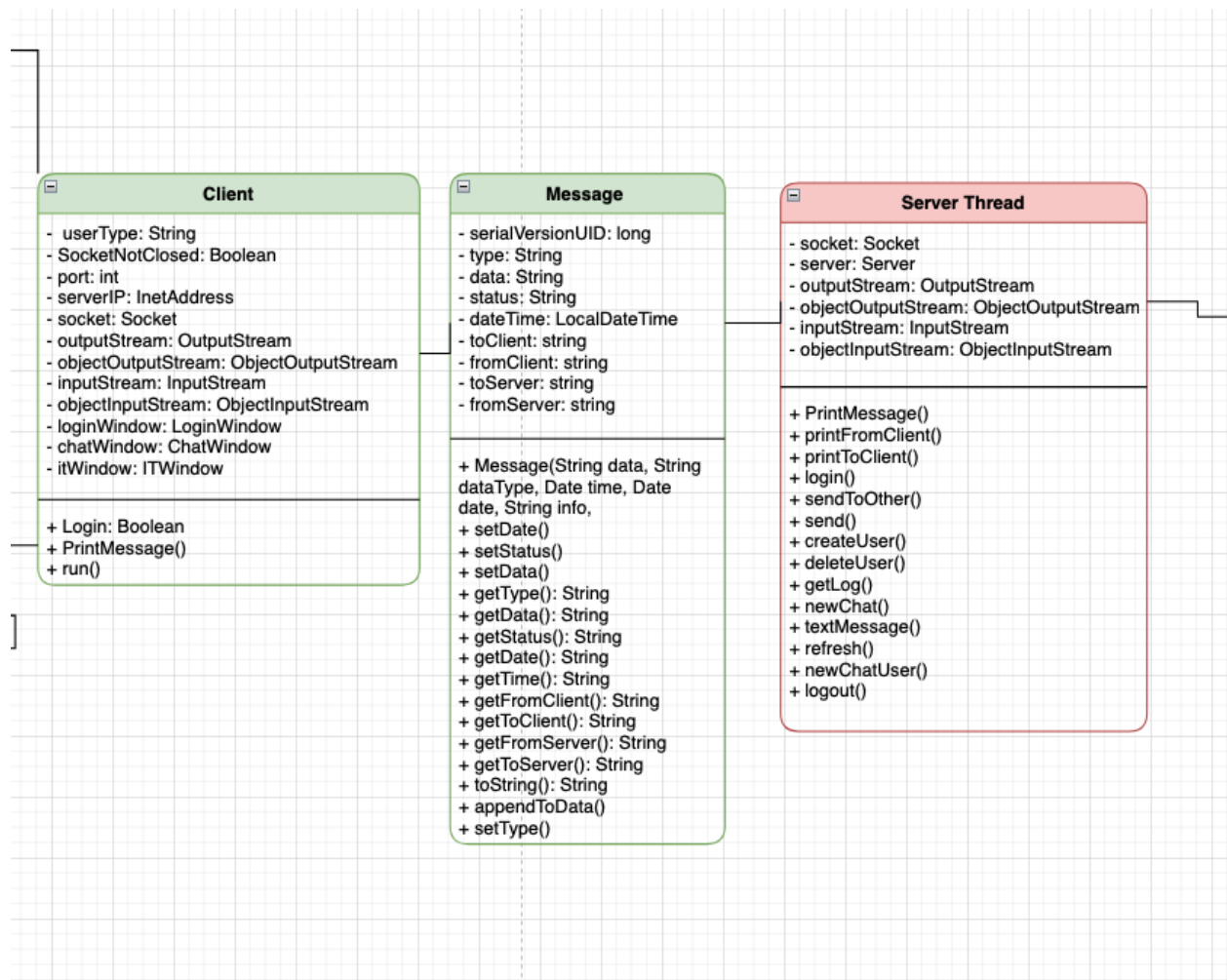
7.0 Appendices

Person/Server Classes:



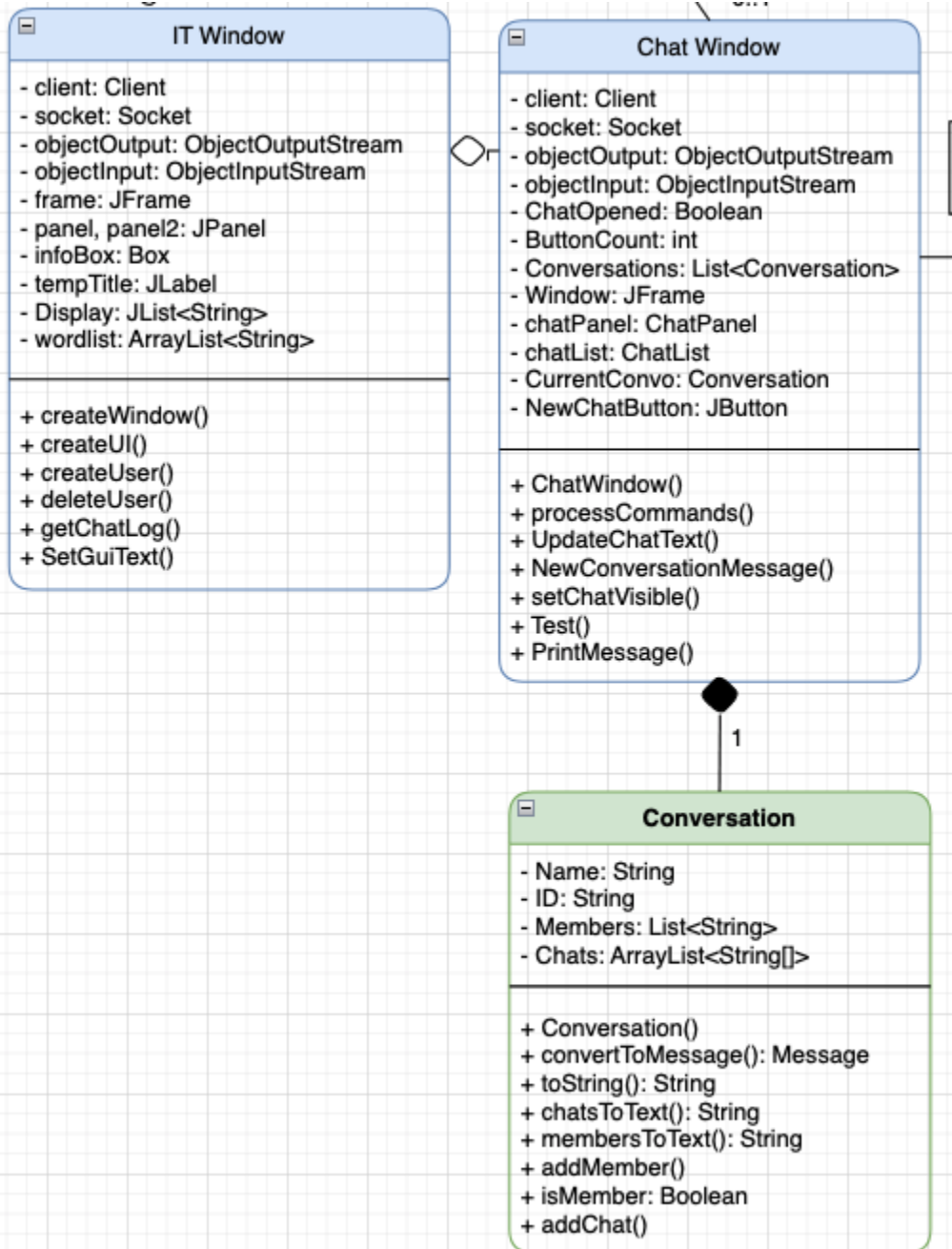
IT class removed. IT functionality will primarily be handled by the Server Thread class as it did too little to dedicate a class to. Person will now hold a String userType to store whether IT or Person.

Message Class with Server/User interaction:

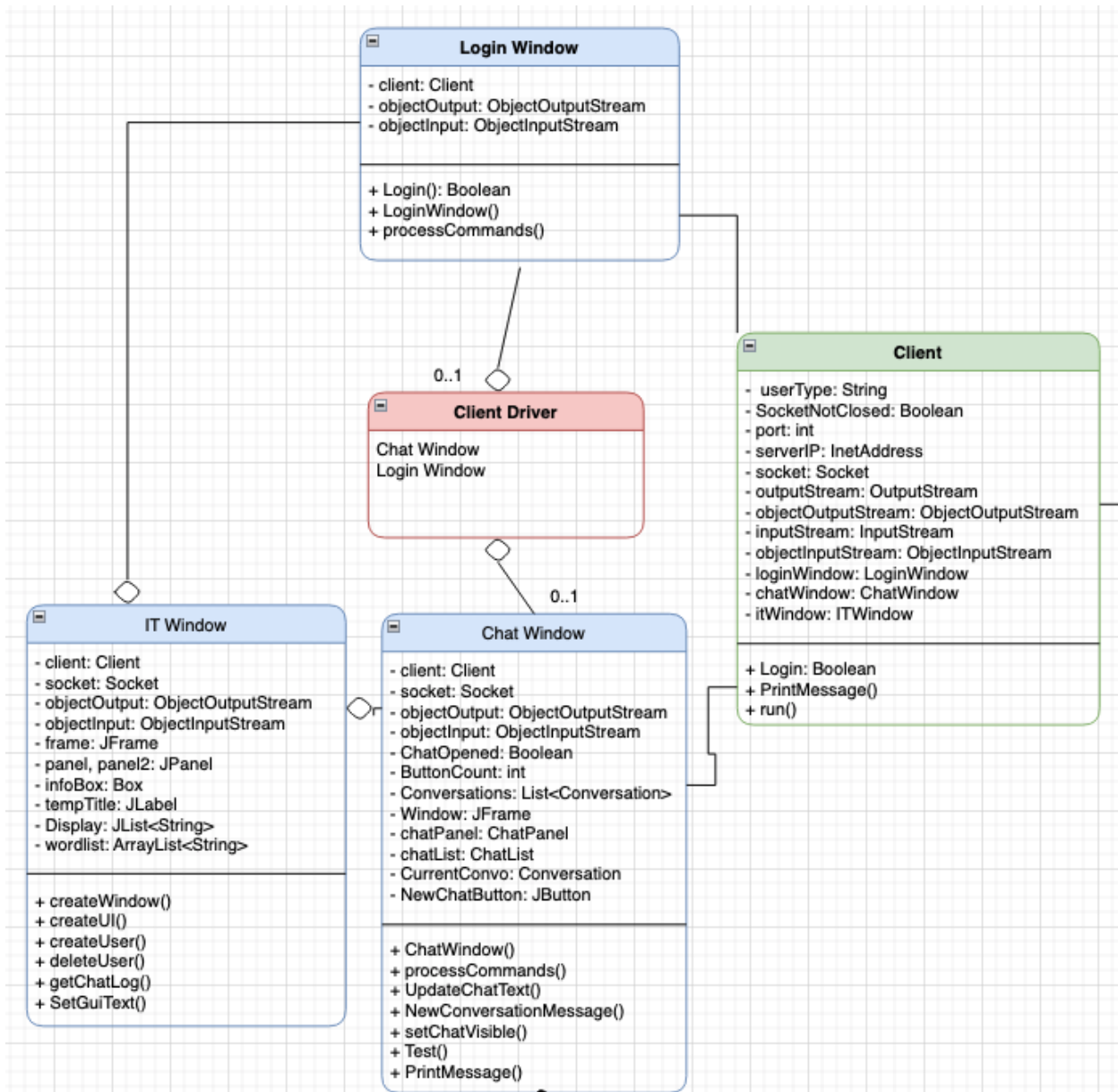


Relationship between Message will be between Client and Server Thread rather than with the Server class

Design for chat window:



Login window and Chat window design with client driver:



Added IT window for IT functionality to be displayed separately from the normal Chat Window

Overall Connection and Design:

