

F29So group report

Take two

Retake

Stage 2 Report Group 11



Table of Contents

[Outline 2](#_Toc94793190)

[Purpose 2](#_Toc94793191)

[Scope 2](#_Toc94793192)

[Overview 3](#_Toc94793193)

[Company Website 3](#_Toc94793194)

[Project Site 5](#_Toc94793195)

[Scope 5](#_Toc94793196)

[Environment 5](#_Toc94793197)

[Database 8](#_Toc94793198)

[Hosting 8](#_Toc94793199)

[Sequence Diagrams 9](#_Toc94793200)

[Logging into an account 9](#_Toc94793201)

[Sending a chat message 10](#_Toc94793202)

[Drawing on a whiteboard 11](#_Toc94793203)

[Class Diagram 12](#_Toc94793204)

[Plan 13](#_Toc94793205)

[Altered Plan 16](#_Toc94793206)

[Updated Gantt Chart Table 19](#_Toc94793207)

# Outline

## Purpose

The purpose of this document is to outline the progress made on the online collaborative software that is being created for the client, Smart Information Flow Technologies (SIFT). This document will present the current state of the software. It will describe the features that have been implemented and our choice of technologies/software used. Through this we can provide a more detailed overview into how the software is built, why some technologies/software were used over others and what we plan to do in the future/what additional features we plan to use.

This document is intended for Dr. Rick Freedman and the Smart Information Flow Technologies (SIFT) team. Listed below are the individuals that may view this document:

1. Dr. Rick Freedman and any of their appointed associate: for the client to be able to discern what form the final product will take as well as what features it may potentially have.

2. Take Two Team: for the purpose of record keeping and comparing assorted design versions if the Take Two decide to make changes.

## Scope

The software being described in this document is, Retake. Retake is an online collaborative social media platform that aims to connect users and allow them to edit a media type simultaneously. The features will be discussed in detail as well as what functional and non-functional requirements they fulfil from the previous report.

We have also created a separate website to be linked to the main site. This will act as a landing page for the public and provide them with more information about the team, the project roadmap, our skills, and our goals.

## Overview

The sections to come will present the company website as well as a description of the design and choices made and how we hope it influences the audience.

The current system and its features will also be described in detail with diagrams to help visualise the system and help convey how we expect the different parts of the system to work with one another.

Finally, the document will contain a plan of what we plan to continue implementing, changes we will make and the features we plan to add.

# Company Website

<https://arfsi09.wixsite.com/lunar>

Retake is an online collaborative application to target content creators where they can connect, create, and share content. A promotional company website was created to help promote and provide basic information regarding the project. The information provided consists of facts and persuading techniques to help induce users to follow the production of this collaborative website.

Distinctive design styles were discussed but a moon theme was concluded to match the slogan “A new platform that’s out of this world”. The overall website was agreed to have a sleek and straightforward design with a majority black. This keeps the website clean from unnecessary clutter so any user visiting for information can be introduced with what they are with no hassle. This also portrays our product application as a simple application that allows users with the same interest to collaborate and share projects with ease.

Retake is for the people and all about bringing passionate people together, the mission is to assist employees, content creators etc., who are working from home and need a modern platform to assist the heavy workloads. This system will allow for instant messaging amongst users and a feature to create group sessions where participating users will be allowed to join and give the ability to modify the project alongside each other.

Some basic collaborative applications already exist but problems were observed, reviews were also carefully analysed which helped the design process. Some observers were also given the chance to read the roadmap, they were interested and left positive comments which were used on the promotional website. Displaying these comments on the website give potential users more confidence in following our project and reducing doubts. Also, it helps strengthen Retakes brand credibility and reliability.

To help gain interest in the project a roadmap was also used to show our vision and what we have achieved within a specific timeframe. Roadmaps help any website viewers to see our projected growth, the benefits include:

* Clarify business strategy
* Progress updates for specific points
* Prioritise new features
* Provides a framework for success and aim to keep working towards
* Provides insight into skills the team may have

Finally, a data encryption message is used as all data is fully encrypted by default, so any future user is always at rest, all content is stored and protected with top quality security. This security is constantly tested to the limit to ensure all our customers feel safe and the highest standards for security are met.

Retake will also be looking for partners to increase knowledge, expertise, and recourses to make the final product better and reach a greater audience. We will be looking to use the best talent and strengths going forward to ensure a top-quality final product.

# Project Site

Retake is a project that is aimed to allow all types of people to collaborate on projects. This will be achieved by allowing multiple users to work on projects at the same time, currently we are only focusing on adding a collaborative drawing app and chat functions.

## Scope

The scope for the project is to produce a web-based application allowing multiple users to communicate and cooperate on visual projects and share ideas with other users.

As per our previous report, we planned to implement audio and video editing features on the web-app, but we have decided to focus on implementing image/visuals uploading and editing, the chat and the whiteboard, which is the primary functionality of Retake.

## Environment

As is to be expected from any website developed to modern standards, Retake features a Front-end, Back-end, and a Database that stores data relating to the site and its audience

Front-end:

Most social medias and projects of this scale are now developed with JSX/NodeJS frameworks such as React.js, Next.js, and so we have elected to implement the front-end (and all visuals) of the web-app using the Angular framework. Angular is the most fitting development framework for this type of project as it facilitates Single Page Applications (SPA) that are, in simpler terms, a building blocks result of components. Gmail and most of Google’s digital services (Play Store, Arts and Scholars, Google.com) are exemplary of what is possible with implementations in Angular.

We believe SPAs provide a seamless user experience as no unrequested features are hardcoded (and hence loaded on every instance opened on a browser), as would be the case with ES6, and are instead developed as reusable components. This is useful for repeated features such as the chatrooms, as they are not only implemented standalone but also along with the whiteboard – this design choice is hence not repeatedly hardcoded, minimizing the risk of errors/bugs during implementation and release. Another benefit of a SPAs is that the singular page needs not listen/emit data to another hence persistent information such as persisted logins, settings tied to specific accounts, and so on only need to be implemented once and will remain consistent across all components regardless of modularity or how coupled every component-service pair is.

Diagram

Description automatically generated

Diagram

Description automatically generated

A picture containing text

Description automatically generatedA piece of paper with writing on it

Description automatically generated with medium confidence

The wireframes above are still our planned visual appearance. All function on the website is coded in Angular/TypeScript, but all visual elements will be HTML <div>’s styled by CSS.

Back-end:

Once again attempting to stick to modern standards, we decided on using Node JS with Typescript to serve the backend (this is the use of Express). The MEAN Stack is a popular concept and we opted to use Express as we believe it is the most compatible way of connecting an Angular front-end to a Postgres-SQL database. As some Angular libraries imported use TypeScript, we opted to code the everything except the Database in TypeScript.

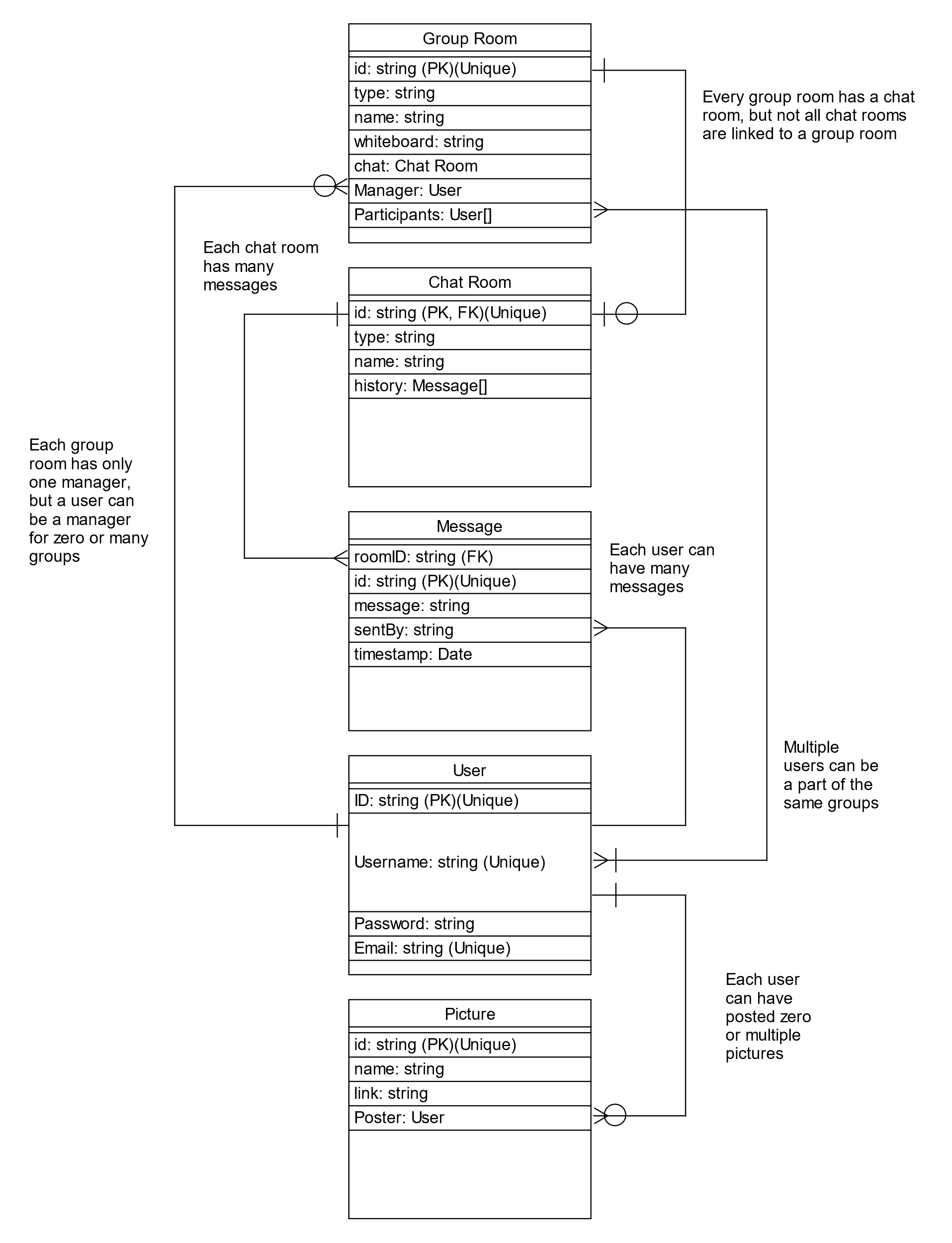
Overall, to list our coding environments/frameworks, one may refer to the following:

* Angular JS,
* TypeScript,
* Express (NodeJS)
* HTML and CSS,
* PostgreSQL

## Database

The database we decided on using was a Postgres database, which was because it was quite easy to setup with Heroku, and because it was free, secure, and easily scalable, which is very import for this project and if many users will be using this platform.

The entity relationship diagram shows how each of the tables are connected along with a sentence about why they are connected as such.



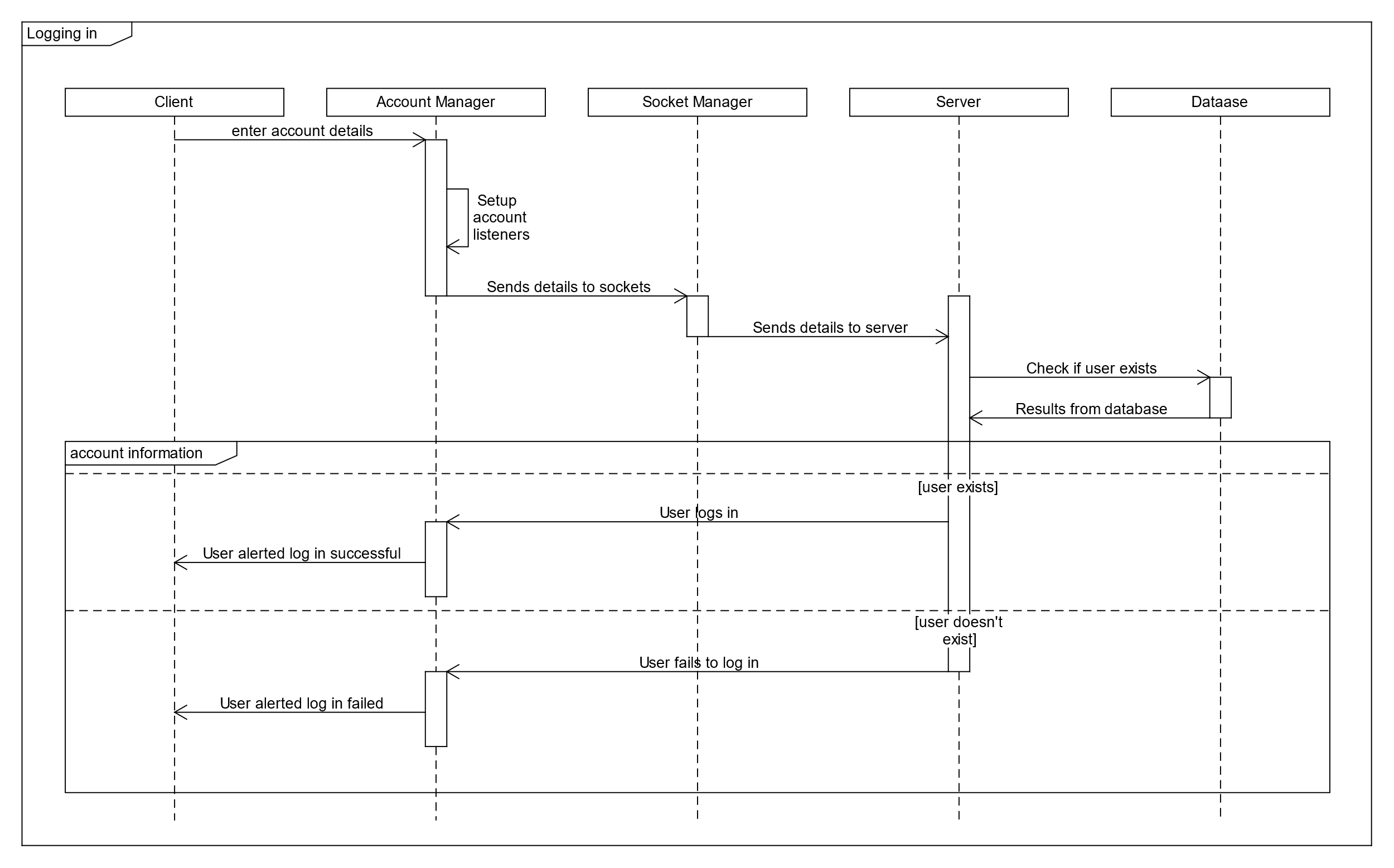
The database is currently setup with five tables, a table for storing the user information, a table to storing messages, a table for storing chat room information, a table for storing group room information and a table used to store posted image links.

## Hosting

We are currently hosting the website on Heroku, this is because we had used the site before, and they have built in databases that the apps can use. Heroku can be linked to GitHub to automatically run the website when the GitHub is updated. Once the website has been deployed for release to the public, it may be hosted by a more business-oriented or professional server but currently we have selected to use the free services of Heroku for testing purposes.

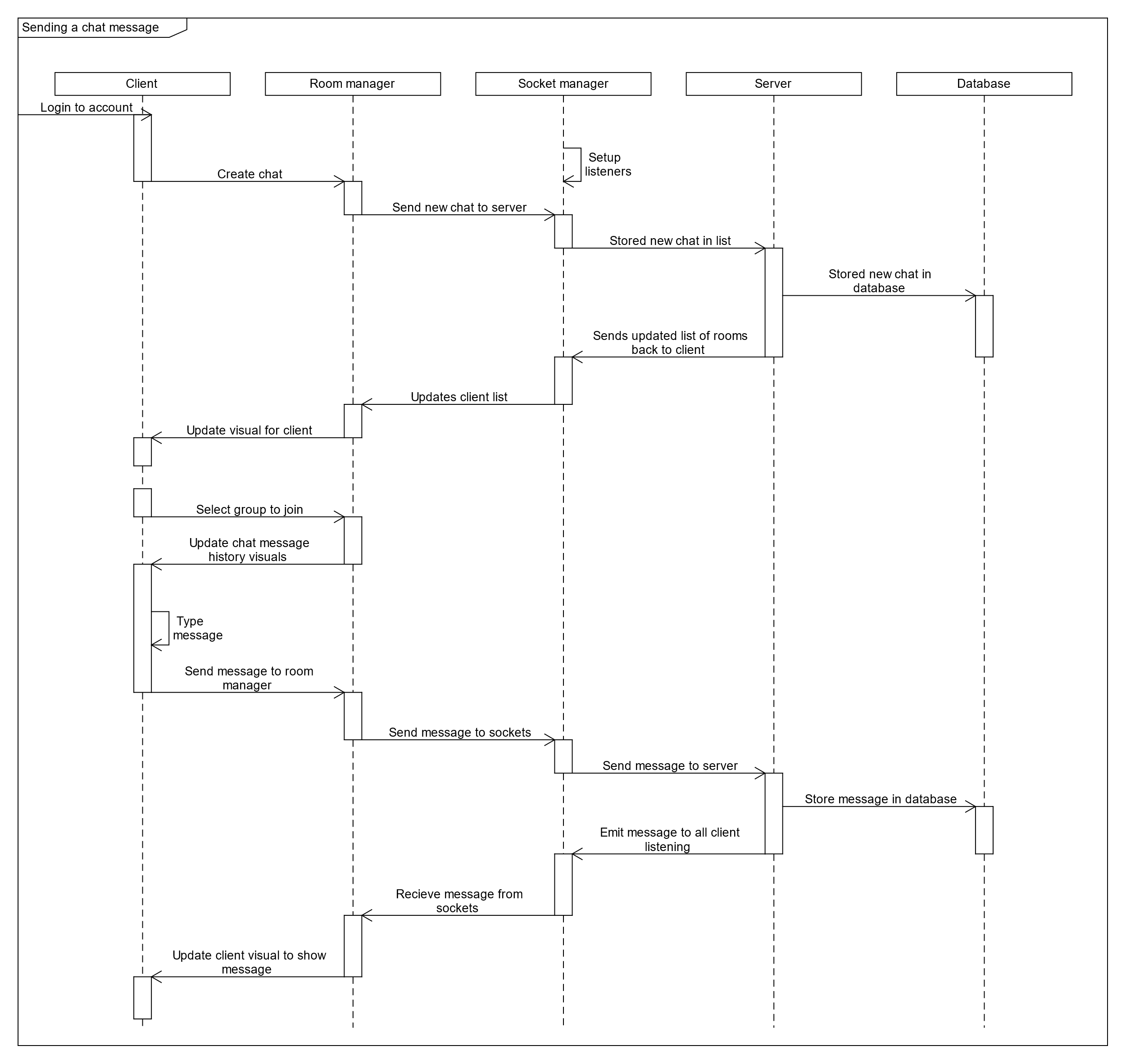
## Sequence Diagrams

### Logging into an account



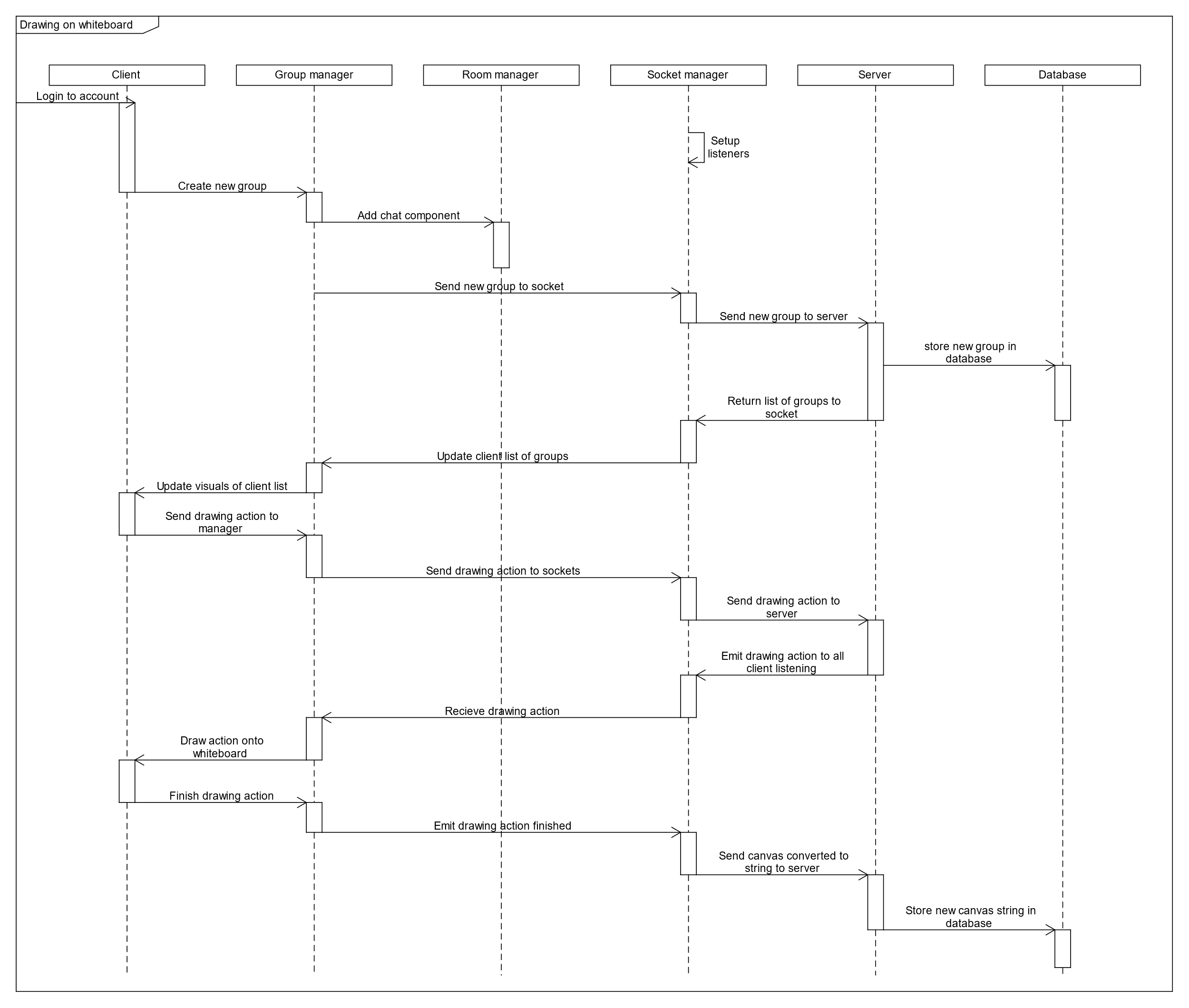
The above diagram shows how the classes interact and operate when a user attempts to log into an account from the client. It begins with the client enters their account details, this then gets pushed to the account manager which creates the listeners for the server messages for accounts the client details then get sent through the sockets to the server, which reads from the database and checks if the account details match a user in the database, if the account details match, the server will send the message to the client to login successfully, if the account details do not match the server will send the message to fail to log in.

### Sending a chat message



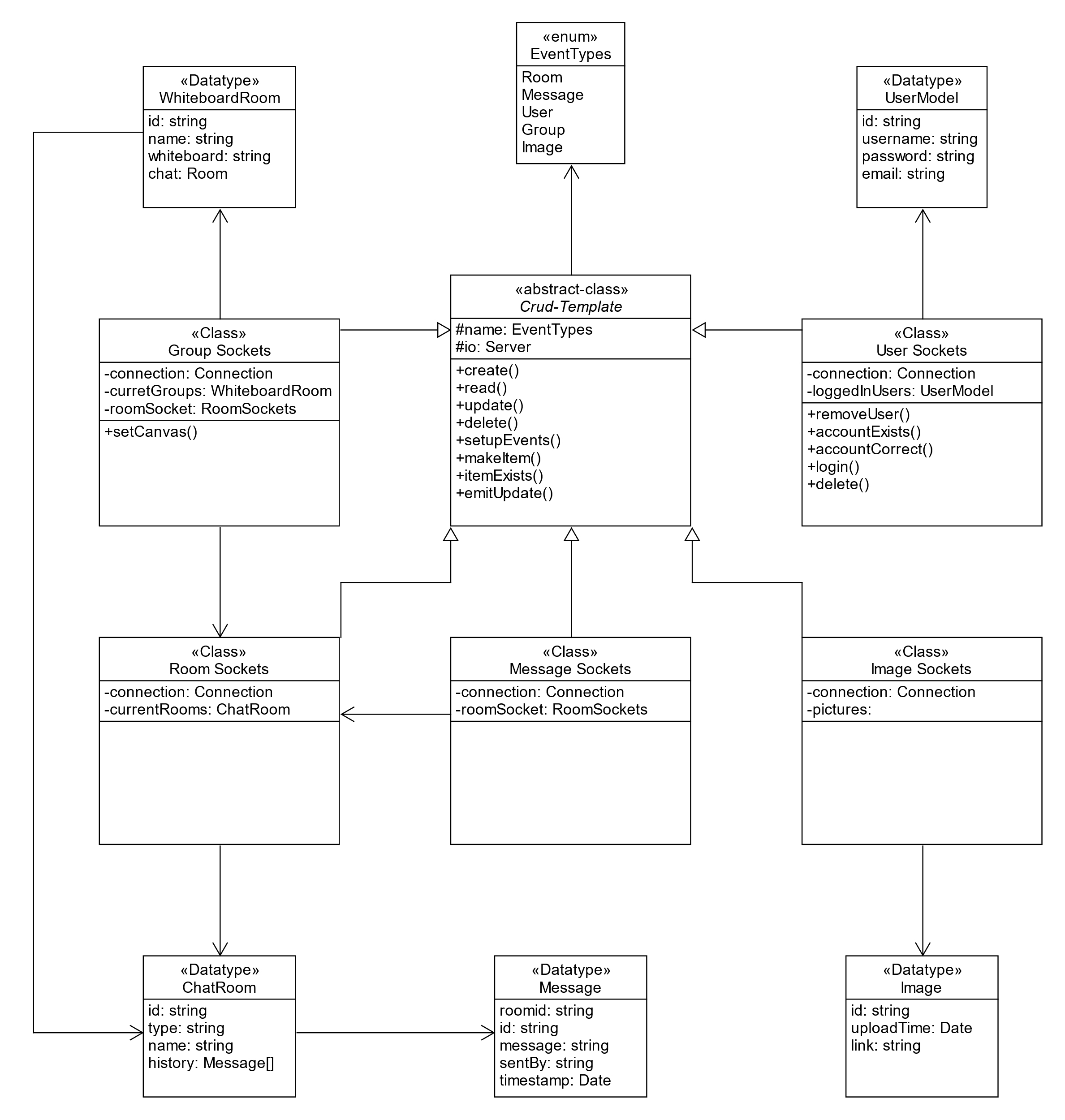
Sending a chat message is only supposed to work when the user has logged in (currently it does not matter if the client is logged in) so once the client has logged into an account, they can then create a chat, which will send a message to the server to create a new room in the current room list held on the server and in the database, this new list will be sent to the client. Once you select a group the message history will show, and once you type a message and press send, it will be sent to the server through the room manager then sockets, the new message will be stored in the database and forwarded onto the clients listening to the room and then all the clients’ visuals will update with the new message.

### Drawing on a whiteboard



Drawing onto the whiteboard is also only supposed to work once the user has logged into their account. Once the client has logged into their account and clicks on the create room button, a message is sent to the server through the group manager and socket manager to create a new room, the server then creates a new group, then sends the new list of rooms back to the client and updates the client list. When the client clicks on the canvas it sends the action to server via group manager and socket manager, the server then forwards the actions onto the other clients listening and the group manager then draws to the client canvas. When the client lifts the mouse button their canvas will be converted to a string and sent to the server and that is then stored in the database and current room list.

## Class Diagram



The diagram above shows the server-side file classes and how they are linked together. All the socket files, extend the abstract CRUD-Template this means that all the functions that are present in the template are also present in all the socket files. The CRUD-Template is an abstract class with the necessary function to setup and run, Create, Read, Update and Delete operations, due to it being an abstract class it can be used by different files.

The custom datatypes: UserModel, Image, Message, ChatRoom and GroupRoom, are the datatypes used by each of the socket classes, these types are helpful due to the server being written in typescript which allows types of functions to be defined.

# Plan

**FR-1**: Multiple people must be able to work on a single project at the same time

How important: Must have

When to implement: Iteration 1

The ability to have multiple logged-in users work on single or multiple projects at any moment in time.

This has been accomplished by allowing user to simultaneously draw on the created whiteboard at the same time and see the impact of the other users.

This was implemented by adding a mouse event to the client side to detect when the mouse has been pressed down and it creates an action that holds an x position, y position, width, height, and color which is sent to the server then distributed to all clients that are listening for actions of the room.  They are then drawn on each client separately, we could have updated the canvas on the server then sent the dataurl, but we feared that if there are a lot of users drawing at the same time the server might get overloaded having to draw on multiple canvas at the same time so having each client update the canvas helps stop this, the only problem with drawing on the client is if too many clients are drawing on the same canvas at the same time.

**FR-4**: Allow people to log into an account

How important: Could have

When to implement: Iteration 1

Different account types are needed:

* Guest accounts: These do not need to be logged in, they are assigned when entering the website if you are not logged in, you can look at images but cannot post or collaborate with other website users.
* User accounts: These need to be signed up and logged into, they allow you to post pictures, messages, and work on projects with other users.
* Manager accounts: These are not really a thing. They are just a subgroup of the user accounts and are only applicable when in a group, as many people as you want can be a manager in a group, they will all have the same privileges but normally it would only be a select few.
* Admin accounts: These are held by the company, they are used to moderate all groups and posts, they have the right to remove posted pictures and ban people’s accounts.

Logging into accounts has been implemented, currently only user accounts have been implemented they can create chats and groups.

Treemap chart

Description automatically generated with low confidence

Entering a username, password, and email, then allows you to click signup, which sends the account information to the server, which checks if the database already has an account with that username, if there is no account it uploads it the account to the database then when the user provides the account username and password and attempts to login the user will be alerted that they have been logged in successfully, otherwise they will be alerted that they failed to login.

**FR-6**: Have a whiteboard for doodling

How important: Could have

When to implement: Iteration 2

One of our collaborative systems is a whiteboard to allow multiple users to draw on a single page at the same time.

The canvas is implemented using the built-in html canvas element, which is extremely helpful due to being able to convert the canvas to a string to be able to update the canvas on the server, helps keep the load on the server lower, than having to update every action drawn onto the canvas.

# Altered Plan

**FR-2**: Be able to have group managers to moderate specific groups

How important: Must have

When to implement: Iteration 2

The group manager is either the user who creates the group or someone to who the original creator transfers the power to.

Having managers has not been implemented yet but is a priority in the next stages. Getting the main functionality of the groups working was the seen to be of a higher priority for the beginning stages.

**FR-3**: Be able to produce reports to managers about platform usage

How important: Should have

When to implement: Iteration 2

Group managers need different privileges from regular users, they need to be able to print out statistics for the group and access the settings of the group.

Having group statistics for the whiteboard, such as how many actions each user has drawn onto the whiteboard, or a complete history of each participants messages in the chat.

**FR-4**: Allow people to log into an account and create groups that they have access to

 How important: Must Have

 When to implement: Iteration 1

Logging into accounts has been implemented however currently group creation is public and is to be locked behind logged in accounts. We have used cookies to determine login status and MD5 hash to encrypt the password.

**FR-4.5**: Allow users to sign up

How important: Must Have

When to implement: Iteration 2

We intend to implement a new sign-up system for users, where they may do so with a valid email, username, and account. The username and emails will be entered directly into a database table for storage while the password will be encrypted with MD5 hash first.

**FR-5**: Personalized Post Feed

How important: Must Have

When to implement: Iteration 2

We intend for the homepage (when logged in) to be a feed of posts uploaded from the whiteboard.

**FR-7**: Have media-editing/ audio filters and editing tools

How important: Could have

When to implement: Iteration 4

Another collaborative system is to be able to edit audio with multiple people doing different parts at the same time.

Media-editing / audio filters would be nice to have, there might be a small amount of image editing when uploading an image but other than that, it is likely that adding media-editing and audio filters would take a longer time than we have for this project.

**FR-8**: Multiple environments to collaborate in

How important: Would like

When to implement: Iteration 4

It would be nice to have multiple environments for users to collaborate in, such as a whiteboard, video editing, and sound editing but this would not be compulsory and would only be added later in the project once the more important requirements are completed.

Linked to FR-7, this will be highly unlikely to happen apart from small image editing when uploading images.

## Updated Gantt Chart Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Duration (Days)** | **Start Date** | **End Date** | **Who** |
| FR-2 | 7 | 7/2/2022 | 13/2/2022 | David, Abdul, Harsh |
| Coding | 5 | 7/2/2022 | 11/2/2022 | David |
| Test | 2 | 12/2/2022 | 13/2/2022 | Harsh, Abdul |
| Document | 1 | 13/2/2022 | 13/2/2022 | David, Abdul, Harsh |
|  |  |  |  |  |
| FR-3 | 7 | 14/2/2022 | 20/2/2022 | David, Abdul, Harsh |
| Coding | 5 | 14/2/2022 | 18/2/2022 | David |
| Test | 2 | 19/2/2022 | 20/2/2022 | Harsh, Abdul |
| Document | 1 | 20/2/2022 | 20/2/2022 | David, Abdul, Harsh |
|  |  |  |  |  |
| FR-4 | 7 | 21/2/2022 | 27/2/2022 | David, Abdul, Harsh |
| Coding | 5 | 21/2/2022 | 25/2/2022 | David, Harsh |
| Test | 2 | 26/2/2022 | 27/2/2022 | Harsh, Abdul |
| Document | 1 | 27/2/2022 | 27/2/2022 | David, Abdul, Harsh |
|  |  |  |  |  |
| FR-5 | 7 | 28/2/2022 | 6/3/2022 | Harsh, David, JB, Abdul |
| Coding | 5 | 28/2/2022 | 4/3/2022 | Harsh, David |
| Test | 2 | 5/3/2022 | 6/3/2022 | Abdul, JB |
| Document | 1 | 6/3/2022 | 6/3/2022 | Harsh, David, Abdul |
|  |  |  |  |  |
| FR-7 | 7 | 7/3/2022 | 13/3/2022 | Harsh, Kearann, JB, David |
| Coding | 5 | 7/3/2022 | 11/3/2022 | Harsh |
| Test | 2 | 12/3/2022 | 13/3/2022 | David, Kearann, JB |
| Document | 1 | 13/3/2022 | 13/3/2022 | Harsh, Kearann, David |
|  |  |  |  |  |
| Finalization of code | 10 | 11/3/2022 | 20/3/2022 | David, harsh, Abdul, JB, Kearann |
| Expo | 1 | 23/3/2022 | 23/3/2022 | Everyone |