# Software Requirements Specification and Design document

**for**

# Gettr

**Version 1.2 approved**

**Prepared by Eric Paulin, Angel Samora, Christian Manibusan, Ethan E**

**The Elite Four**

**May 18, 2023**

**https://github.com/Peggys-jpg/GettrProject**

*Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.*

# Table of Contents

# Revision History

| Name | Date | Reason For Changes | Version |
|---|---|---|---|
| Eric Paulin, Christian Manibusan, Angel Samora, Ethan E | 4/2/23 | Initial SRS Draft | 1.0 draft |
| Eric Paulin | 4/30/23 | Frontend Framework Changes | 1.1 |
| Eric Paulin, Christian Manibusan, Angel Samora, Ethan E | 5/18/23 | Final Submission | 1.2 |

# 1. Introduction

## 1.1 Purpose

Gettr is a forum web application for CSUSM students and faculty to communicate with one another on their school and/or personal projects outside of school. The forum will allow both students and faculty to specify the type of project they wish to get done along with the tools, amount of people needed, and the timeframe for the project in their post. The forum will also allow students and faculty to create an account, which allows them to chat directly with other users as they wish. The main objective is to have a working and fully functional release build by the end of the semester but also create a platform where users can find others to help them turn their ideas into actual realized projects.

## 1.2 Document Conventions

The following SRS was mainly written in Times New Roman with a font size of 12. Words and phrases that are *italicized* are done so to stress their importance as a feature or show key details during the design process. Tables and flowcharts are also used throughout the document to easily list and portray information.

## 1.3 About our project team

The Elite Four is a group of four Computer Science students at CSUSM who wish to provide a platform where students and faculty can communicate with one another for their school and/or personal projects outside of school. Our team *strives* to make quality software that bridges the communication gap for our clients and serves as a place where they can easily discuss what projects they wish to create, but also find others who will aid them in turning those ideas into actual projects.

## 1.4 Intended Audience and Reading Suggestions

| **Intended Audience:** | **Pathway:** | **Section:** |
|---|---|---|
| Developers | 1. Introduction<br>2. Dependencies and constraints<br>3. Operating environment<br>4. Design and implementation constraints<br>5. User documentation | ● 1.1<br>● 2.2<br>● 4.4<br>● 4.5<br>● 4.6 |

- The developer pathway puts emphasis on understanding the technical aspects of the product so from the introduction heading into the section that covers dependencies and constraints,

to the operating environment, then to design and implementation constraints, and lastly finishing at the user documentation section.

| **Intended Audience:** | **Pathway:** | **Section:** |
|---|---|---|
| Project Managers | 6. Introduction<br>7. Communication<br>8. Design, Development, and Implementation Methods<br>9. Change Management<br>10. Product Perspective<br>11. Product functions | • 1.1, 1.3<br>• 2.1<br>• 2.3<br>• 2.4<br>• 4.1<br>• 4.2 |

- For project managers, it's *crucial* to understand the aspects of the project that impact development and the timeline of the project, so much of the pathway focuses on sections covering communication, development and implementation methods, change management, product perspective, and finishing off at product functions.

| **Intended Audience:** | **Pathway:** | **Section:** |
|---|---|---|
| Marketing staff | 1. Introduction<br>2. Product perspective<br>3. Product functions<br>4. Operating environment | • 1.1<br>• 4.1<br>• 4.2<br>• 4.4 |

- Anybody involved with marketing is going to want a greater sense of the product and what it can offer, so the pathway heavily emphasizes sections that cover the product perspective, the operating environment, and the target audience.

| **Intended Audience:** | **Pathway:** | **Section:** |
|---|---|---|
| Users | 1. Introduction<br>2. Product function<br>3. User Documentation | • 1.1<br>• 4.2<br>• 4.6, 5.1 |

- For users, being able to understand the product, its purpose, and how to use it is very important, so recommending users to read the sections on product function and user documentation will provide them with a general understanding on how to navigate our product.

| **Intended Audience:** | **Pathway:** | **Section:** |
|---|---|---|

| | | |
|---|---|---|
| Testers | 1. Introduction<br>2. Dependencies and Constraints<br>3. Design, Development, and Implementation Methods<br>4. Technical requirements<br>5. Design and Implementation Constraints<br>6. Assumptions and Dependencies | • 1.1<br>• 2.2<br>• 2.3<br>• 4.3<br>• 4.5<br>• 4.7 |

- Testers will generally focus on the user's experience and journey through a product, so focusing on sections that describe some of the systems in place along with the limitations of our project, technical requirements, assumptions, and dependencies will go a long way in their understanding of our project.

| **Intended Audience:** | **Pathway:** | **Section:** |
|---|---|---|
| Documentation writers | 1. Introduction<br>2. Dependencies and constraints<br>3. Technical requirement<br>4. Design and implementation constraints<br>5. User documentation | • 1.1 - 1.3<br>• 2.2<br>• 4.3<br>• 4.5<br>• 4.6, 5.1 |

- Documentation writers will want to read the sections of the SRS that go into detail about the dependencies and constraints within the project as well as the technical requirements and eventually end at the section covering user documentation where we provide a variety of components that users can use to help navigate our project.

## 1.5 Product Scope

| | |
|---|---|
| **Product description**: | A student forum (web application) specifically for CSUSM students to be a part of a platform where students and faculty can collaborate on projects |
| **Objective**: | Create a platform that'll make it easier for students and teachers to connect on projects |
| **Goals**: | Increase social and educational activity within students at CSUSM |
| **Benefits:** | Increased social activity, students wanting to collaborate on side projects with other students, students working on teacher conducted projects, etc. |

### 1.6 References

Material UI Library https://mui.com/
React.JS https://react.dev/
Java Spring Boot: https://spring.io/
Draw.io: https://app.diagrams.net/

# 2. Statement of Work

This section provides an outline of the general layout and scope of the project titled Getter and will help describe many of the expectations that we have as a team, the ongoing communication between the team (Elite Four) and faculty (CSUSM), as well as describe some of the limitations, dependencies, and the development of our project, along with a format of how our group will approach changes.

### 2.1 Communication

| **Type of meeting:** | **Form of communication:** | **Reason:** |
|---|---|---|
| Daily meeting | Daily discord communication between members | Discuss current goals and objectives for the day. |
| Weekly meeting | Weekly group calls, discord/zoom meetings | Discuss work, roadblocks, current objectives. |
| Weekend meeting | Weekend video calls, github contributions | Implementing new code, testing code, updating documentation. |
| Professor meetings (optional) | Zoom meetings/In-person (office hours) | If we run into problems that could potentially limit the scope of our product/project. Internal team problems if they do happen. |

### 2.2 Dependencies and Constraints

| **Dependencies:** | **Description:** |
|---|---|
| React.JS | React will be heavily utilized as the front-end framework of our choice. Our team will need to actively keep track of every update to React and its community libraries. |
| Material UI Dependency | Our project is going to be using Material UI as the main frontend library for our project, so being attentive to any new updates and changes will |

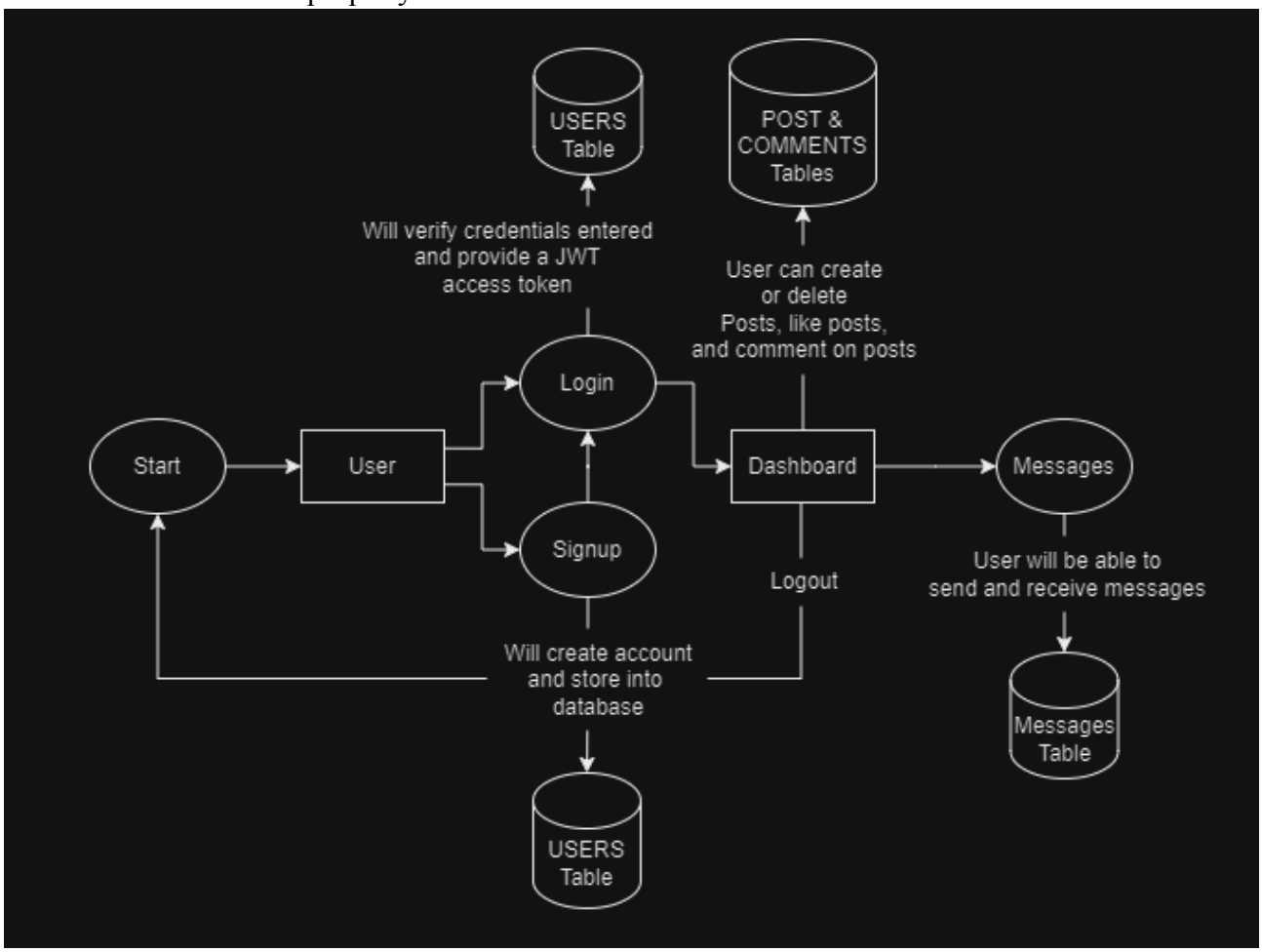
| | |
|---|---|
| | be pivotal during the development of Gettr. |
| Java Spring Framework | For back-end development our team will utilize the Spring framework. Any updates to the services offered by Spring are important to the development and maintaining development cycle.<br>**Spring Modules:**<br>- Spring Data (JPA, JDBC)<br>- Spring Web(Spring MVC, Apache Tomcat)<br>- Spring Security(JWT token)<br>- Spring Websocket(STOMP) |
| MySQL | During the development process we will be running a local instance of MySQL to store and retrieve data for the website. |
| Critical path dependency | We won't be able to test certain functionality until both the frontend and backend have been completely built (ex. testing if user credentials are being saved in the DB can't happen if the API, backend, and frontend aren't built first) |
| Finish-to-start dependency | Creating the API that connects the frontend to the backend cant be initially started until the backend database design is finished. |

| **Constraints:** | **Description:** |
|---|---|
| Time-constraints | Due date for the project |
| Resource constraints | Project members have academic and other outside responsibilities so availability could be a concern. |
| General Data Protection Regulation (GDPR ) | The GDPR specifies that if we decide to store user data from Europe, users will need to provide consent requiring us to add additional components and features to the web application. |
| California Consumer Privacy Act (CCPA) | Users have the right to know what information is being collected, the right to delete their personal information from our DB, the right to opt out of sharing their personal information, as well as other rights. Requiring us to build additional components and features. |

## 2.3 Design, Development, and Implementation Methods

We will be following the agile software development process, the main IDEs used across the group would be IntelliJ and Visual Studios Code, incorporate the React.js Framework along with JavaScript, CSS, HTML,and MUI on the Frontend, in terms of the backend, our team will utilize Java with the Spring Framework, specific modules being(Spring Security, JPA, Spring Websocket, and the Spring Web Development Server) these back-end operations will connect to a local instance

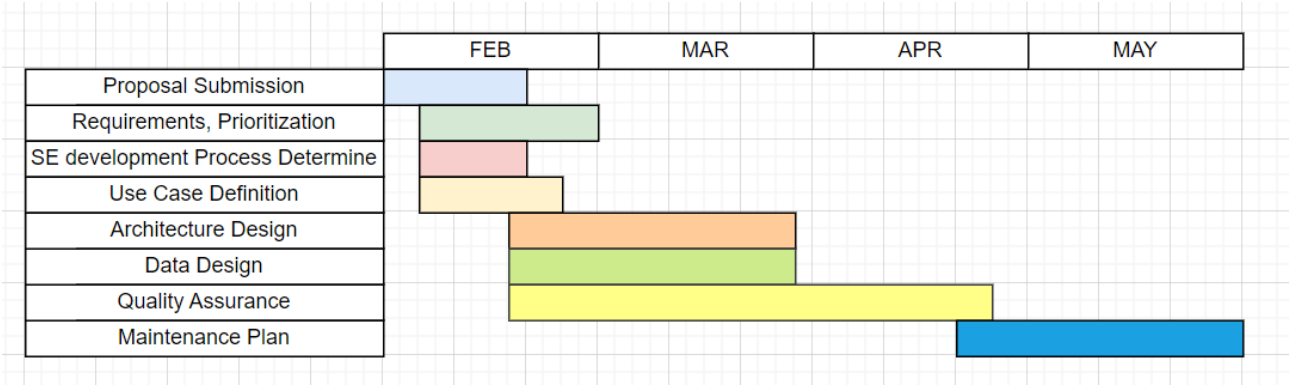
of the mySQL database when in development process. Our team will utilize GitHub for source control, every addition/assignment to the project will be carried out in separate branches and when pull requested, there will be a team review of the merge and will resolve any conflicts with the main branch, this process will also include a group testing to ensure there are no force pushes. As for backup and recovery procedures we will utilize Git's restore features to restore a previous version if needed. The intellectual property owner will be Christian Manibusan.




## 2.4 Change Management

If any changes to the scope, requirements, technology are required there will be a group meeting where the proposition takes place and the whole team will weigh their opinions on the proposal and decide whether or not the change is not only beneficial but within the scope and time constraints. The individuals held responsible for any unaccounted significant costs of the project's stability, quality, schedule, etc. will be the entirety of the development team.

# 3. Timeline




***Timeline is subject to change due to the nature of the agile process***

# 4. Overall Description

## 4.1 Product Perspective

The product to be developed derives ideas from common social media platforms today such as Reddit, Discord, Instagram, however it is unique in the fact that this platform is exclusive to CSUSM students to network with one another all for the pursuit of academic betterment and the creation of products/projects for personal gain. This product is not intended to replace any existing application, but to be its own contained product that all CSUSM students and faculty can use.

## 4.2 Product Functions

Major functions include *1.) Account creation and validation*, *2.) Dashboard Discussion (forums) 3.) Database 4.) Direct Messaging (User to User)*

## 4.3 Technical requirements

Technical requirements for the acquirer to access Gettr requires a device to have adequate internet connection in order to access the Gettr webpage. The supplier must be able to connect to a server and database to manage the web application information of the users as well as the ability to run an IDE to use the Material UI Framework to create web applications. Additionally, a supplier requires a system that is able to access the internet to use GitHub to collaborate on the project.

## 4.4 Operating Environment

For the environment, we will be using the Visual Studio Code for Frontend Development and the IntelliJ IDE for the backend. To develop a web-based application using JavaScript, we will also be implementing the Material UI Framework to achieve this. In order to collaborate on the project, we will also be using Github to share our progress and collaborate.

### 4.5 Design and Implementation Constraints

Being able to link the frontends JavaScript code along with the and backends Java code may prove hard as there are not many online resources that seamlessly do so. Learning how to utilize the relational database and the timing will be a small issue to the scale of our project as well.

### 4.6 User Documentation

User documentation on how to navigate through Gettr to utilize the web applications features will be provided in the form of the "About" web page. This "About" page will include information by The Elite Four on the team, their intentions with the site, how the web application can be used, along with notes on how to navigate the application.

### 4.7 Assumptions and Dependencies

For the project, we are planning to use the React.JS Framework along with Java's Spring Framework to develop a web application. Assuming that we will be sticking to these components of the project, change to any of these while working on the project will affect the development pace and the learning curve if we decide to use another framework or database.

## 5. External Interface Requirements

### 5.1 User Interfaces

Gettr will have a modern minimal aesthetic that is clean and simple. The web application will use a uniform Times New Roman font for the text as well as for bold text. The main page of Gettr will be the dashboard which will display the tabs of the different pages, along with the current forum posts. Besides that, a plus button in the bottom left corner of the dashboard page will allow users to create a new forum post for others to see.

### 5.2 Hardware Interfaces

In order to connect to the Gettr servers and use the web application, the only hardware needed is a device that is able to connect to the internet. Gettr uses the Material UI Library, which supports all device types whether it be a desktop, laptop, or mobile devices.

### 5.3 Software Interfaces

Database connectivity will be important in holding all the user information as well as the projects, so the MySql database and Spring Data will be used during the creation and maintenance of this project. Secured API Endpoints will be used to connect the frontend with the backend to offer GET, POST, PUT, and DELETE operations for user data.

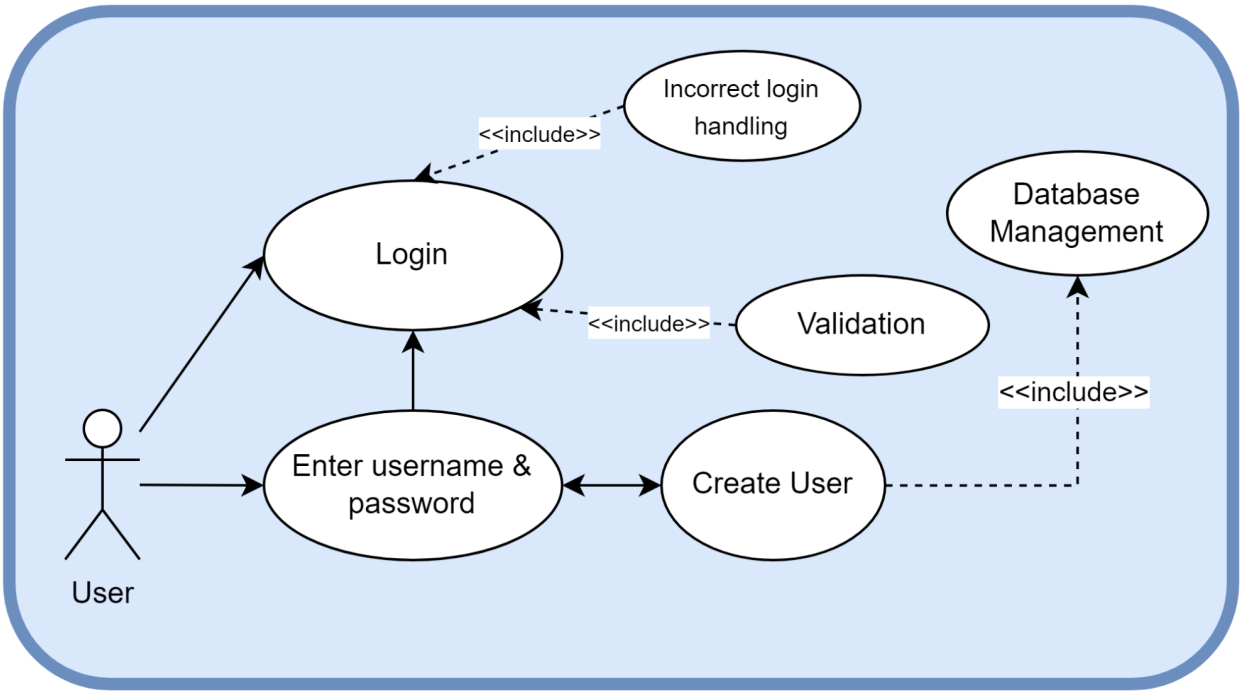
### 5.4 Communications Interfaces

The spring boot backend that we implemented allows the front-end developers to make specific calls to endpoints that don't require users to be authenticated within our system. Inside our security config file we set endpoints that don't require validation and also set requirements that users need to be validated for any of our protected endpoints, in particular endpoints that retrieve information from our database. We use the axios library to make the endpoint calls to our backend system that then processes the http requests inside of our controller classes that handle what gets returned depending on the endpoint that gets called.

## 6. System Features

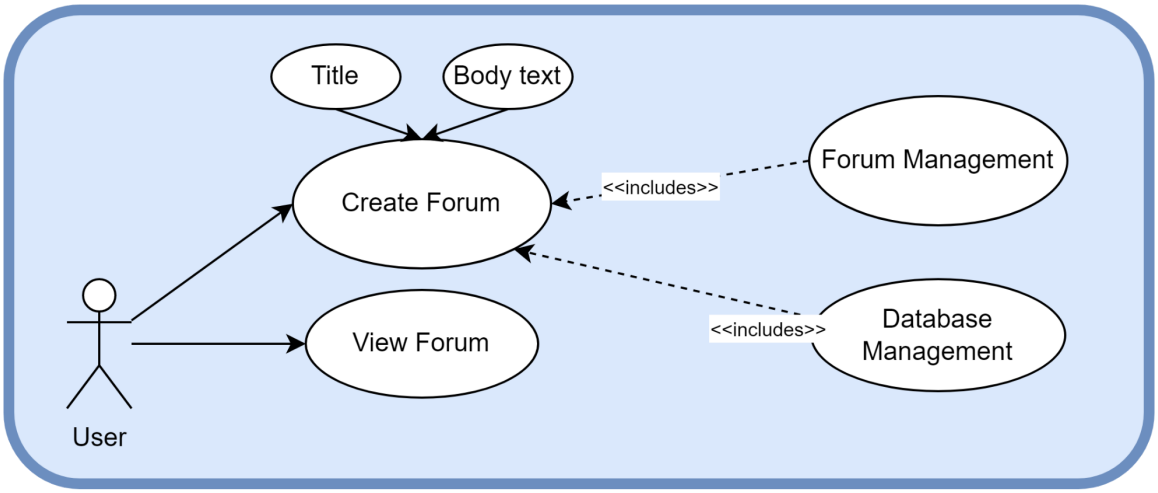
### 6.1 System Feature 1

<table>
<tr><td colspan="3">Overview: FR-01</td></tr>
<tr><td>Use Case:</td><td colspan="2">Login</td></tr>
<tr><td>Description</td><td>• Login will handle user account registration<br>• Validate user credentials<br>• Incorrect password handling<br>• User account information will be saved into database</td><td>Priority: Medium<br>Benefit: 7<br>Cost: 5<br>Risk: 2</td></tr>
<tr><td>Includes:</td><td colspan="2">• Includes incorrect password handling<br>• Includes validation<br>• Includes database</td></tr>
<tr><td>Actors:</td><td colspan="2">• Users(Students, Faculty)</td></tr>
<tr><td>Precondition:</td><td colspan="2">1. User has navigated to our website<br>2. User is greeted with a login page</td></tr>
<tr><td colspan="3">Sequence:<br>1. User is able to create account<br>2. User enters username<br>3. User enters password<br>4. User clicks login<br>5. System handles incorrect login information<br>6. Validation service verifies user credentials</td></tr>
<tr><td>Post Condition:</td><td colspan="2">• Users are redirected to the home page</td></tr>
</table>



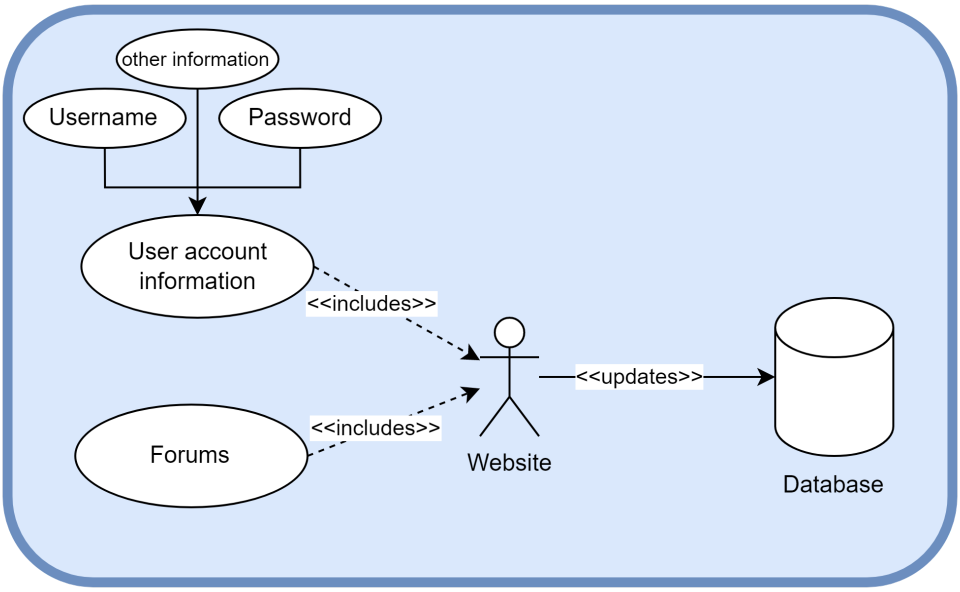

## 6.2 System Feature 2

| Overview: FR-02 | | |
|---|---|---|
| Use Case: | Users can create discussion boards | |
| Description | • Users will be able to create forum boards | Priority: High<br>Benefit: 10<br>Cost: 10<br>Risk: 10 |
| Includes: | • Includes database | |
| Actors: | • Users(Students, Faculty) | |
| Precondition: | 1. Users have logged in<br>2. Users are currently on the home page(dashboard) | |
| Sequence:<br>1. User from dashboard will be able to create a forum post<br>2. User from dashboard will be able to click and view existing forum posts | | |
| Post Condition: | • Creation of forum post | |



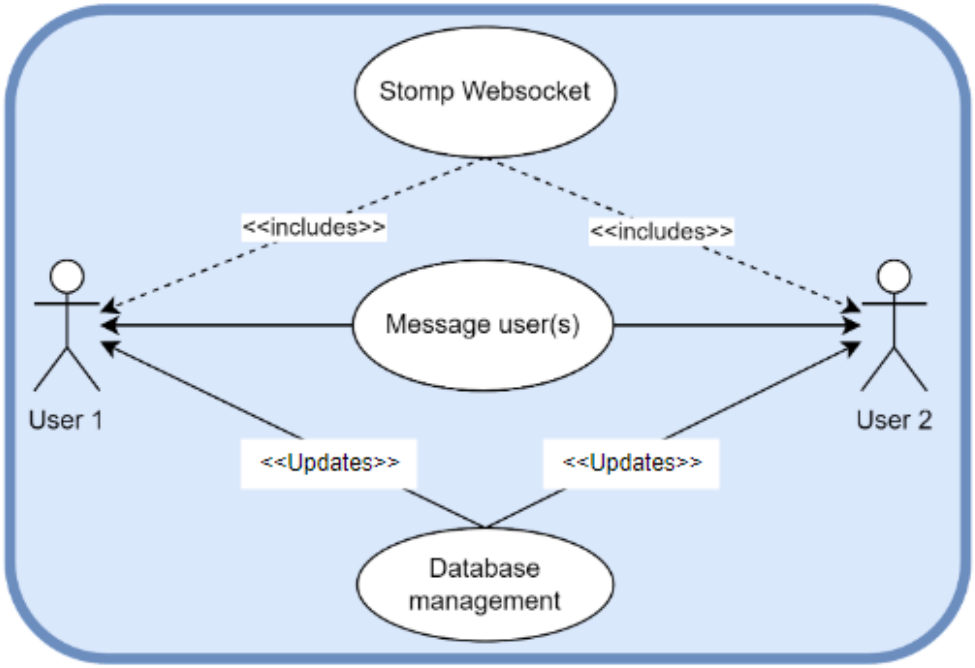

## 6.3 System Feature 3

| Overview: FR-03 | | |
|---|---|---|
| Use Case: | Website will store user information, forum posts, onto the database | |
| Description | • Website will take user account information when users register an account<br>  ○ Database stores user account information and any other sensitive information.<br>• Website will update database with forum posts | Priority: High<br>Benefit: 10<br>Cost:10<br>Risk: 2 |
| Includes: | • Includes user account detail<br>• Includes database | |
| Actors: | • Website | |
| Precondition: | 1. Users have created accounts<br>2. Existing forums | |
| Sequence:<br>1. User creates account<br>2. Website sends new account information(e.g. user, pass, email, etc) to database<br><br>1. User creates forum post<br>2. Website sends forum post and details to database | | |
| Post Condition: | • User account information successfully saved in the database<br>• Forum posts successfully saved in the database | |




## 6.4 System Feature 4

| Overview: FR-04 | | |
|---|---|---|
| Use Case: | Users can send messages to other user(s) | |
| Description | • Users are able to click on our messages tab and through it are able to send direct messages to any number of users | Priority: Medium<br>Benefit: 8<br>Cost:8<br>Risk: 7 |
| Includes: | • Websocket connection<br>• Connection to database | |
| Actors: | • User1, User2 | |
| Precondition: | 1. Users need to be logged in<br>2. Users need to click on messages tab | |
| Sequence:<br>1. Users navigate to messages tab<br>2. Search for user in the search bar<br>3. Type in a message to send<br>4. Send the message using our ->(arrow) | | |
| Post Condition: | • User that sends message will see the message that got sent in the chat box<br>• User on the receiving end will receive the message | |




# 7. Other Nonfunctional Requirements

## 7.1 Performance Requirements

- The product must be able to handle multiple users using the website to prevent crashing
- Website uptime needs to exceed downtime

## 7.2 Safety Requirements

- User account verification
- Website security to prevent attacks
- Server-side operations to prevent system vulnerabilities and exposing website internals
- Data validation

## 7.3 Security Requirements

| User type | Who | User Identity authentication method |
|---|---|---|
| Admin/Developers | • Gettr developing team (The Elite 4) | • Unique username and password |
| Real users | • Students<br>• Faculty | • Valid username and password<br>• Bot Captcha |

## 7.4 Software Quality Attributes

**Usability**: Gettr should be easy to use and navigate. Users should be able to create and manage forum posts, and search for content without encountering any major usability issues.

**Performance**: Users should be able to interact with the application quickly and efficiently, without experiencing significant lag or delays.

**Security**: Gettr should be highly secure, with appropriate measures in place to protect user data, prevent unauthorized access, and detect and mitigate potential security threats. User authentication and authorization should be robust and well-implemented.

**Availability:** Gettr should be highly available, with the ability to handle large volumes of traffic and user activity.

**Scalability:** The code base for Gettr should be made in a way to allow all future updates/features to be implemented in an easy and secure way.

**Compatibility**: Gettr should be highly compatible, with support for a wide range of devices, operating systems, and web browsers. Users should be able to access the application from any

device or platform without encountering any major compatibility issues.

## 7.5 Business Rules

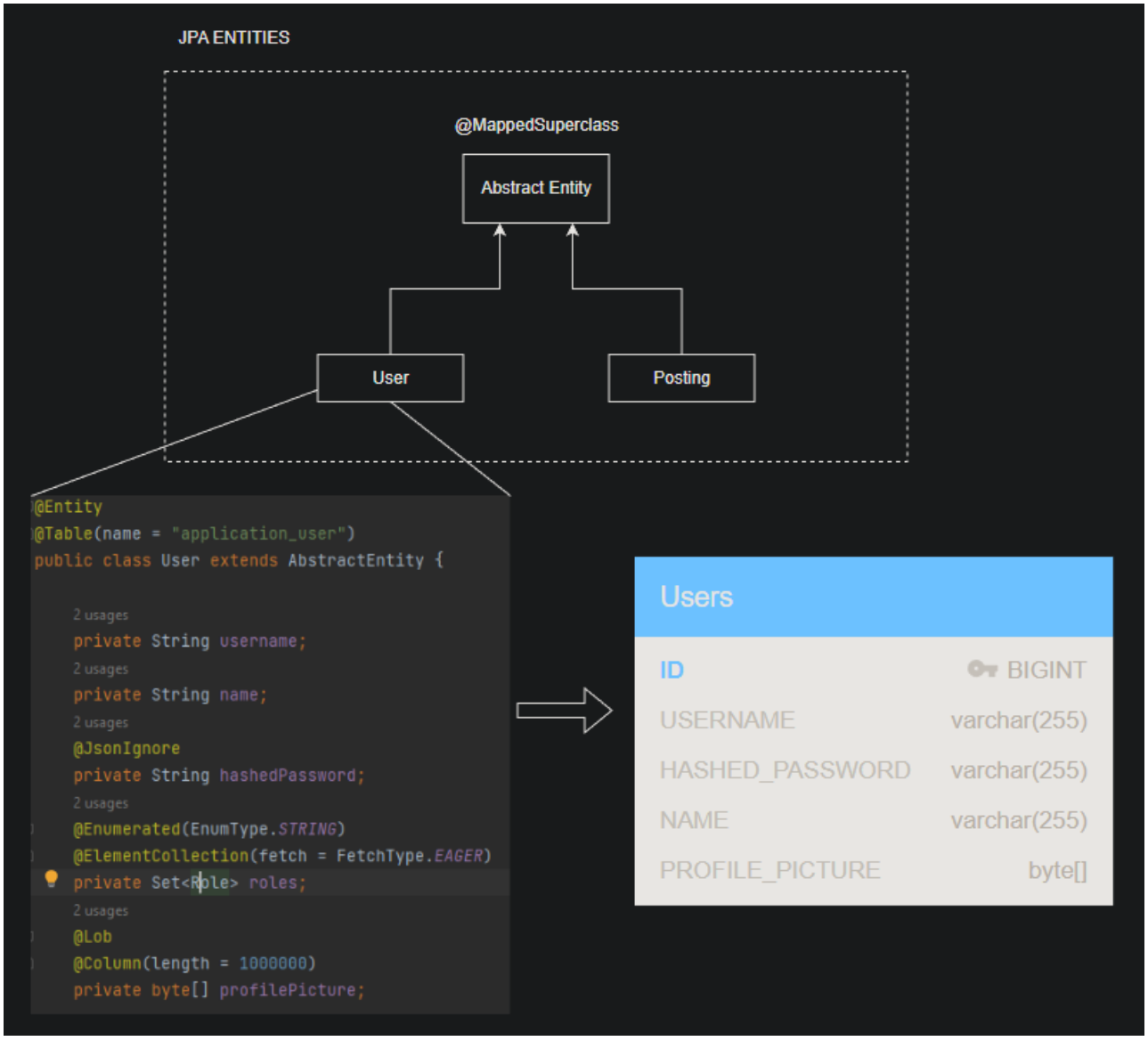
1. A user is eligible to use the website if they are students and/or faculty of a private or public university
2. Users must adhere to a code of conduct that prohibits offensive language, personal attacks, and discriminatory behavior. Any user who violates the code of conduct will be subject to disciplinary action, up to and including account suspension or termination.
3. Users' personal information, including their email addresses and other contact information, will be kept private and not shared with third parties. However, the forum web application may use aggregated data to improve the platform and enhance the user experience.

# 8. System Architecture

## 8.1 Architectural Design

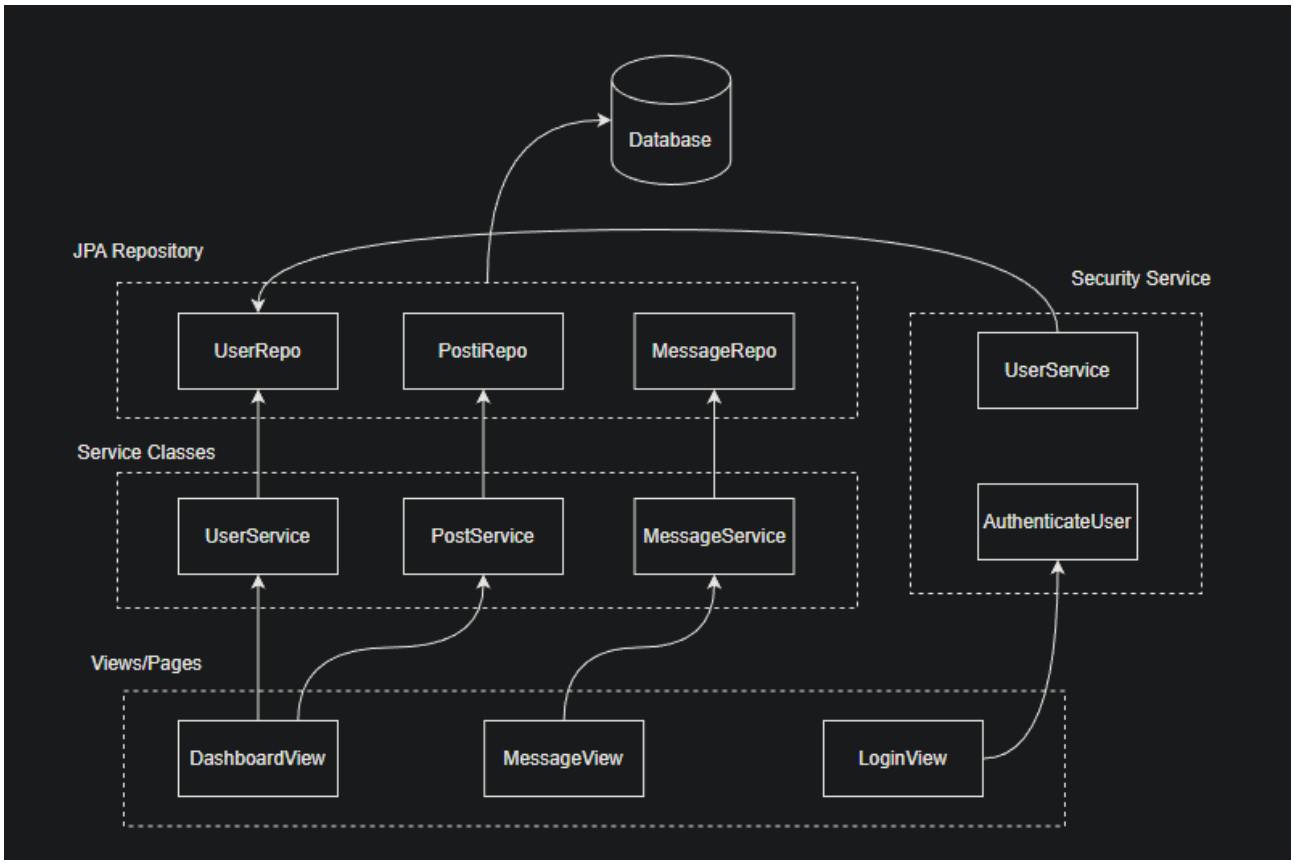
Current Design Version 1:

FIGURE 1




- Our team utilizes spring data which incorporates the JPA API and the Hibernate framework for “Object-Relational Mapping” (method of using an Object Oriented Programing language to access a database in an agnostic manner)
- Hibernate uses the JPA entities shown on the right to create a local instance of a mySQL database which we can then use during run-time.

FIGURE 2




- Our JPA repositories *Example.)* (*UserRepository*,*PostingRepository*,*MessageRepository*) include some basic CRUD methods and queries that have already been defined.
- We can also include our own custom Queries using the *JPQL* language.
- The service classes shown are the classes that have direct access to these repositories and we can define database functionalities for the views/pages to use here.

## 8.2 Decomposition Design

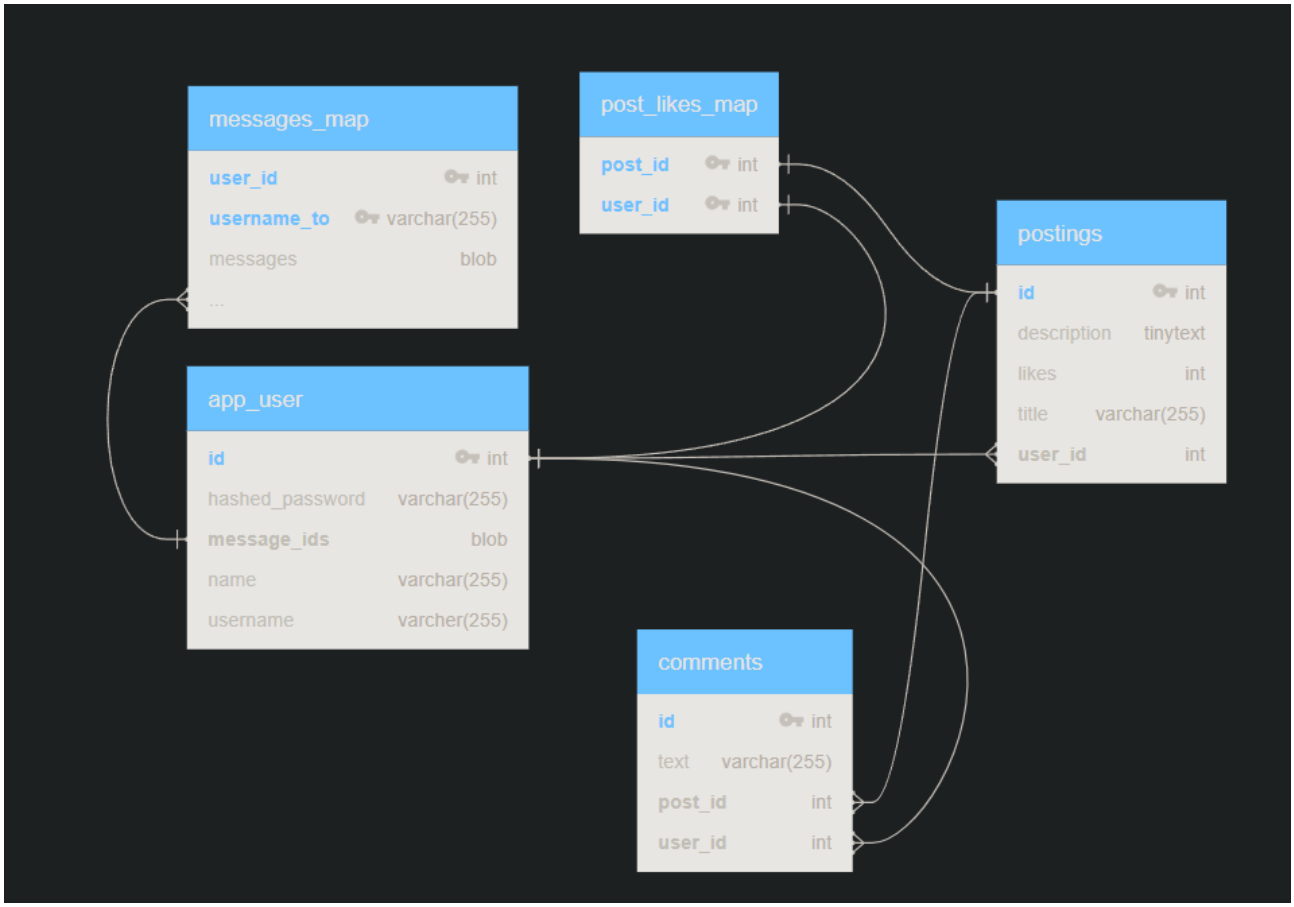
Shown in Figure 2 above

## 8.3 Design Rationale

The idea was to have the client accessible classes (ProfileView, ManageUsersView, DashboardView, and LoginView) to access our predefined service classes instead of directly accessing the repositories. The advantage of this design is that clients get to use predefined functionalities to access the database in a safe manner and to minimize any accidental ‘misuse’ of the database.

# 9. Data Design

## 9.1 Entity relationship design




## 9.2 Data dictionary

**POSTINGS -**

ID - uniquely generated id Primary Key
Title - Title description of the forum post
Description - The main content of the post
Likes - integer counter for how many likes a certain post has
Poster - Foreign key to User table

**APP_USER -**

ID - uniquely generated user id Primary Key
Username - the username of a given profile
Hashed_Password - the encrypted password of the user using password encoder
Name - the name of the user
Message_Ids - list containing concatenated Concatenated Foreign Keys *user_id* and *username_to* for to the *Messages_Map* table

**MESSAGES_MAP -**

User_Id - key 1 for user (sender-side)
Username_To - key 2 for user (receiver-side)
Messages - list holding all messages between the two above users

**POST_LIKES_MAP -**

Post_Id - key 1 for post liked
User_Id - key 2 for user who liked the above post

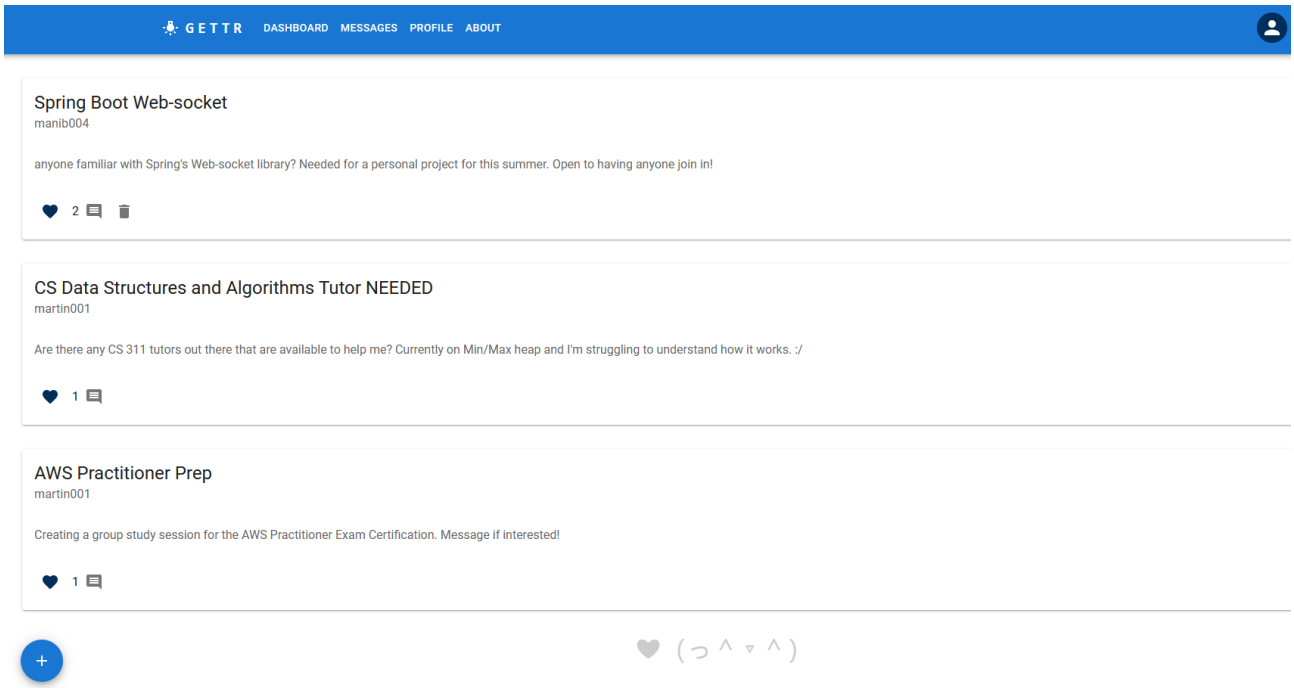
**COMMENTS -**

ID - unique id Primary Key
Text - the comment text
Post_Id - Foreign Key to the post that the comment belongs to
User_Id - Foreign Key to the user that made the comment

# 10. Other Requirements

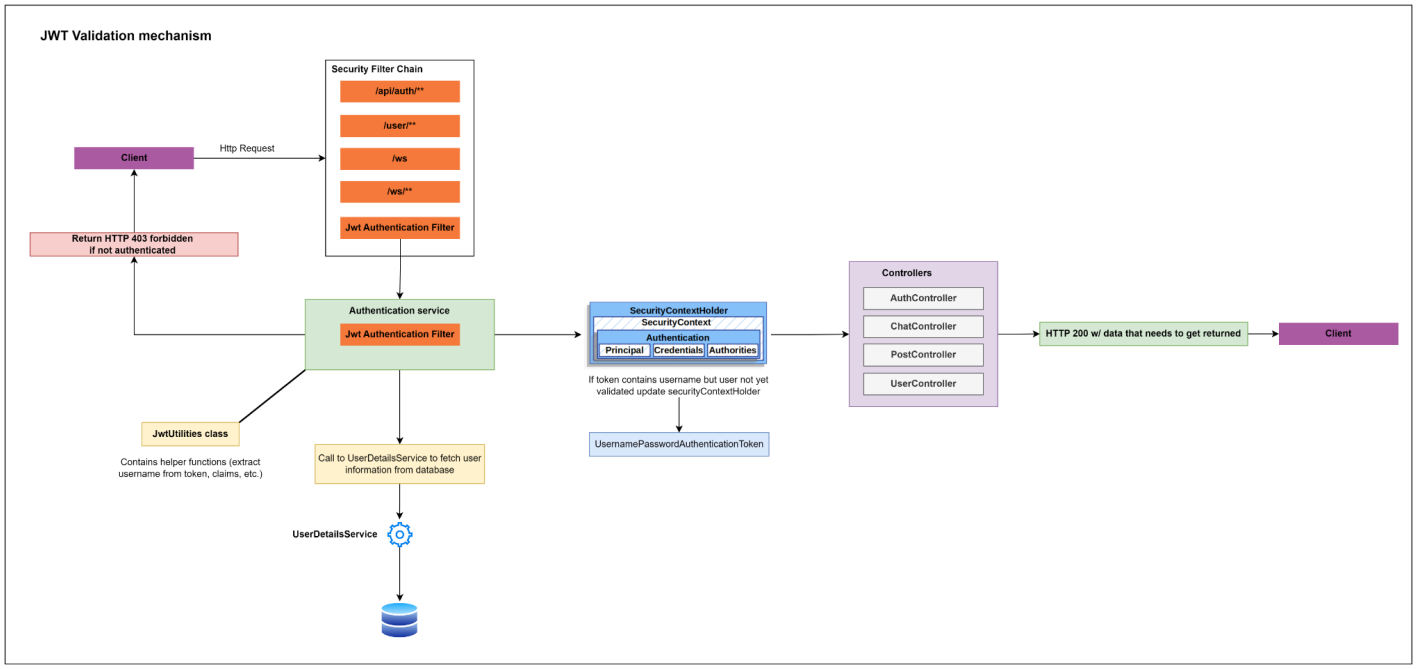
Since the forum's audience is CSUSM students and faculty, the color palette of the web application will be consistent with the official CSUSM colors that are provided by the University. The colors on the forum include University Blue (#002E5A), Cougar Blue (#0062a5), Spirit Blue (#3ab5e8), and 20 Gray (#d1d3d4). Below is a link from the University regarding the following colors:
https://www.csusm.edu/communications/brand-style-guide/fonts-and-color/index.html

# Appendix A: Screen Images



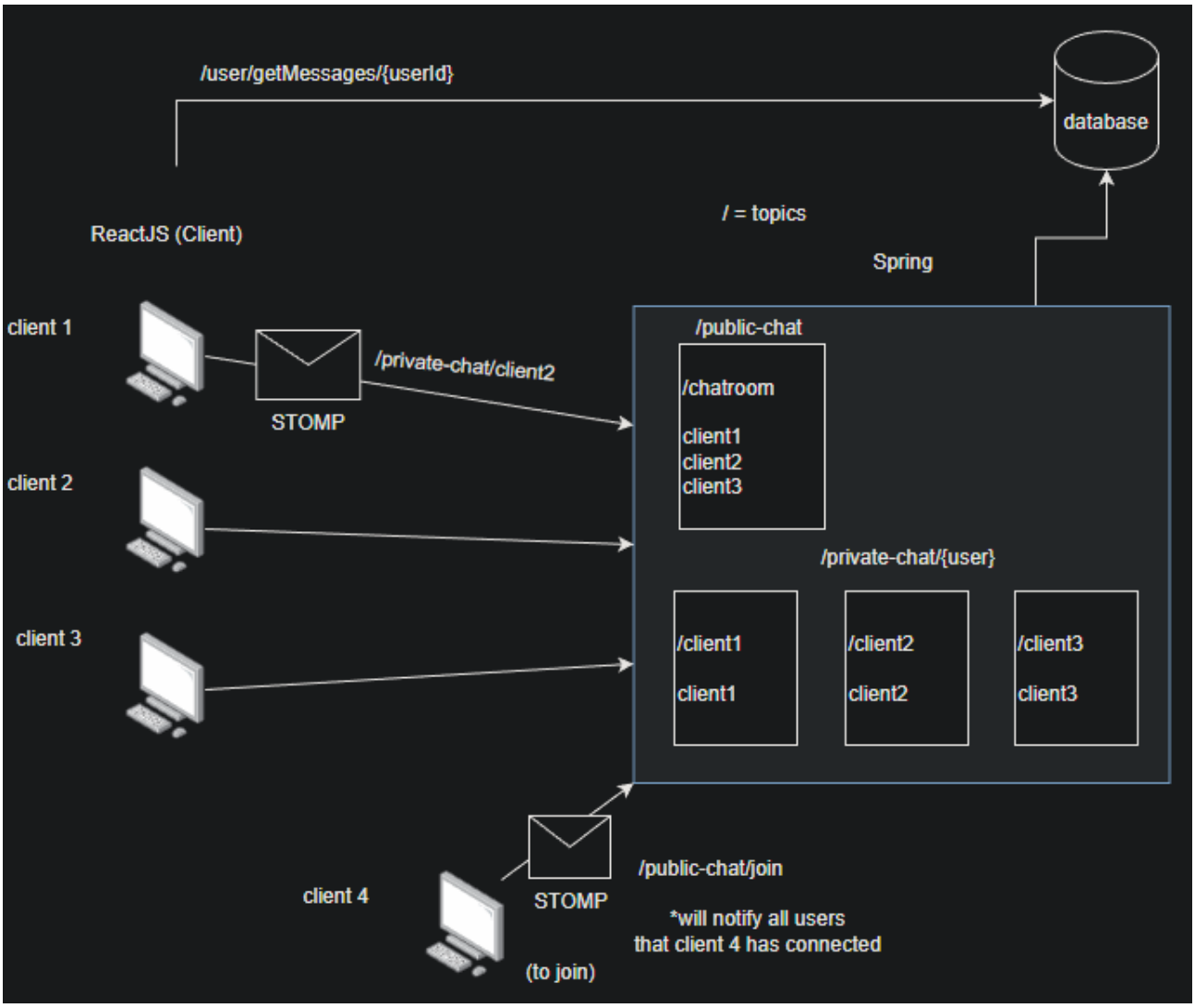

(Gettr Dashboard on Desktop)

# Appendix B: Analysis Models




* Security model flow chart

# Appendix C: WebSocket model




* STOMP websocket model