Contributions and Challenges

Igor wrote the proof of concept for our app that turned into the backbone for the rich text editor. Igor also set up our AWS service with EC2 servers. It required a lot of research and using an amazon product that he was unfamiliar with. Including research into how to optimize our design for the services that we chose to use. Igor also took time to explain the concepts to the group to make sure we all understood the architecture of the app. He also figured out how and if we should be making amazon accounts for work on the servers. Igor was also the primary writer of the requirements.

Thao's contribution was in setting up user authentication, Heimdall, for our app. She had to learn about JWT (JSON Web Tokens) in order to set it up. Through our current implementation, our back end is able to extract the current user's credentials from their JWT in order to determine if they are authorized to access certain resources. It was definitely an 'ah-ha' moment as we had thought that we would have to send the credentials with each API call that the user triggers. Thao also worked to figure out Kubernetes and how to use it to orchestrate the containers that our app's microservices will reside in. This included conversations with Igor with regards to the application's architecture.

Honor worked on the rich text editing part of the application. Her main challenge that she ran into was figuring out how the different text flags would work in our html-based text editor. She was able to figure out bolding quite easily but a conceptual challenge was met when she reached highlighting as there was no existing function to use to highlight text.

Riley worked on the text editor section of the application to provide the functionality for the cursor; the cursor originally reset to the beginning of the file each time someone typed, so she made it update whenever a change was made. Riley also worked on Karen which is the service that Heimdall uses to query the database of users to see whether or not a user can be authenticated. A challenge in this was maintaining the security of the user's passwords. The password is stored encrypted, and when an authentication request is made, the password in the request is compared to the encrypted password.

Requirements

This section of the progress report is our requirements for the finished project of our application. The requirements are what we will use to plan our ‘next steps’ for the incumbent semester.

Accounts

Users shall be *required* to register an account to use the application.

Users shall be able to register by *either*:

- providing an **email**, **username**, and **password**
- authenticating through their Google or Facebook account

Users shall have the *option* to set a *single* custom **avatar** for their account.

Users shall have the *option* to set up multi-factor authentication for their account.

Users shall have the *option* to set a custom **nickname** for themselves *for each* **conversation**.

Conversations (metadata)

Creating

Users shall be able to create *multiple* **conversations**.

Users shall be able to create a **conversation** by providing a **name** for the **conversation** and any number of **users** to be invited to the **conversation**. A **conversation** can be made up of *one or more* **users**.

Users shall be able to specify **users** to invite to a conversation by entering a **username** or **email**. This will send an **invitation** to the given user in the application and through their **email**.

Users shall be *required* to assign a *single* **role** *for each* **user** in a new conversation. The allowed roles are:

1. **OWNER**
   a) automatically assigned to creator
   b) one per **conversation**
2. **ADMIN**
3. **USER**

Users shall have the *option* to set a *single* custom **picture** for a new **conversation**.

Users shall have the *option* to set a *single* custom **description** for a new **conversation**.

Updating

Users shall be able to join a **conversation** by accepting an invitation for that specific **conversation**.

Users shall be able to decline an invitation to a **conversation**.

USER+ users in an existing conversation shall be able to invite other users to the **conversation**, with the following conditions:

- **OWNER** users can assign **ADMIN** or **USER** role to new **users**
- **USER** & **ADMIN** users can assign **USER** role to new **users**

USER+ users in an existing **conversation** shall be able to update the **name**, **picture**, and **description** of the **conversation**.

OWNER users in an existing **conversation** shall be able to update the role of any non-**OWNER** user.

Users in an existing **conversation** shall be to remove themself from the **conversation**.

ADMIN+ users in an existing **conversation** shall be able to remove any **USER** user from the **conversation**.

OWNER users in an existing **conversation** shall be able to remove any non-**OWNER** user from the **conversation**.

Deleting

OWNER users in an existing **conversation** shall be able to delete the **conversation**. This will require entering the **name**of the **conversation** and the **password** of the **user**.

Conversations (content)

Users shall be able to enter and delete text in a **conversation**.

Users shall be able to apply styling to text in a **conversation**. The styling options are:

- bold
- italics
- underline
- background colour (highlight)
- font family
- font colour
- font size

Users shall be able to format text in a **conversation** in the following ways:

- justification
- lists (ordered or undordered)

(REVIEW) Users shall be able to insert images and gifs in a **conversation** which will be anchored as characters.

Users shall be able to set text in a **conversation** as a hyperlink to a web page.

Users shall be able to click on hyperlink text in a **conversation** to open the linked web page in a new tab.

Users shall be able to type the **username** of another **user** in a **conversation** with a prefixed "@" symbol to send that user a notification.

Users shall be able to lock text in a **conversation** so that it cannot be modified.

Users shall be able to access a list of all the locked text in a **conversation**.

(REVIEW) Users shall be able to unlock previously-locked text in a **conversation** so that it can be modified, with the following conditions:

- locked text can be unlocked by the **user** who originally locked it
- locked text can be unlocked by any **user** with a greater **role** than the **user** who originally locked it

History

Users shall be able to access a list of changes made to a **conversation**.

Users shall be able to filter the list of changes made to a **conversation** based on:

- start time
- end time
- user
- keywords
- type of change (addition/deletion/styling/formatting)

Users shall be able to see a specific pre-filtered list of changes made to a **conversation** that shows only the changes made since they last had the **conversation** open.

Notifications

Users shall receive both a browser notification and an email when invited to a **conversation**.

Users shall receive both a browser notification and an email when tagged in a **conversation**.

Users shall receive a browser notification when another **user** modifies the text in a **conversation** while the browser tab is not open.

Users shall receive a browser notification and an email when their **role** has been changed in a **conversation**.

Users shall be able to *globally* mute notifications *for all* **conversations**. Global muting can be applied to just browser notifications or just emails. Global muting can selectively be applied to the following notification types:

- invitation
- text entered
- text re-styled/re-formatted
- user was tagged
- user's role changed

Users shall be able to mute notifications *for each* **conversation** for a set amount of time. **Conversation**-specific muting can be applied to just browser notifications or just emails. **Conversation**-specific muting can selectively be applied to the following notification types:

- text entered
- text re-styled/re-formatted
- user was tagged
- user's role changed

Users shall receive a visual indication when another **user** modifies the text in a **conversation** while the browser tab is open, but the conversation is not open.

Mini-map

Users shall be shown a **mini-map** when they have a **conversation** open. The **mini-map** will be a small section of the browser page that displays the entire **conversation** in a smaller format than the **conversation** area itself. The position of all active **users** will be displayed in the **mini-map**.

Users shall be able to click on a **user's** position indicator in the **mini-map**, which will adjust their main **conversation**view to the position of the **user** they clicked on.

Users shall be shown real-time inidicators of other **users** typing in the **conversation** next to the **users'** position indicators on the **mini-map**.

(REVIEW) Users shall be shown the position of the **tags** in a **conversation** that refer to them on the **mini-map**.