

SOEN 341

Professor Rodrigo Morales

Winter 2025

Sprint 1- Members contribution and time log

Paria Jafarian – 40248494

User Story 1 Contributions : (2h30)

- I had to first come up with an overall simple and modern UI/UX design to make the app look professional
 - Added a nice-looking icon in front of ChatHaven
 - Made it responsive so that when the screen size changes, it still looks good
 - Worked on the login button to have a hovering effect (when the user goes on the button, it gets larger)
 - Worked on the username and password fields so that when the user types, the borders change color to blue
 - Added another simple hovering effect for the “don’t have an account, Sing up” so that it’s dynamic
-

User Story 2 Contributions: (30 min)

- I used the same styling as for the login page to stay consistent with the overall design
 - I used the same hovering effects for the username, password, and confirm password fields
 - I used the same hovering effect so that the sign-up button gets larger when the user goes on it
-

User Story 5 Contributions: (2h30)

- Creating different divs so that the different parts can be grouped together (this was a bit time-consuming to figure out)

- Creating a side bar for the different channels, a main section for the messages, and a typing field with an effect
- Added a nav bar (which should later be a component of its own) with a logout button with a hovering effect

Alexander Raymond - 40167836

User Story 1 Contributions: (5 hrs)

- Setup front end, back end and local MySQL server
- Developed the login page UI with fields for username and password.
- Implemented front end validation fields.
- Connected the login page to the backend using Node.js and Express.js.
- Created server.js file to handle requests with the MySQL database.
- Configured MySQL to store user credentials in a users table.
- Created account authentication that verifies credentials against database

User Story 2 Contributions: (1hrs)

- Built the account creation page with form fields for username and password.
- Implemented front end validation for matching passwords
- Created backend validation to check if the username already exists.
- Wrote SQL queries to insert new users into the users table.
- Set up database error messages.
- Deployed and tested the account creation functionality

User Story 3 Contributions: (3 hrs)

- Designed a basic database schema to store and manage user accounts
- Enforced unique usernames in MySQL.
- Deployed the database on Azure Cloud so that all team members could use a shared DB.

User Story 4 Contributions: (20 min)

- Created a landing page for messages where users are redirected upon successful login.

Additional Contributions: (1h30)

- Wrote setup notes to help the team set up their development environments.
- Helped the team with setting up MySQL, Node.js, and running the project locally.

Zachary Cohen - 40200376

User Story 03 Contribution: (5 hrs)

- Developed the frontend UI for “Forgot Password” page with fields for username and two password fields, and a submit button.
 - Adjusted the “Login” page to make it possible to navigate to the “Forgot Password” page.
 - Connected the “Forgot Password” page to the database using express.js and Node.js
 - Created paths to the page in App.js
 - Created a POST route in server.js to allow the user submission to edit the appropriate cell in the “users” database table
 - Removed extra text files in the backend
-

Additional Contributions: (4 hrs)

- Created the initial Github files (frontend, backend, Sprint 1-4, etc.)
- Developed User Story 04 making the database
- Started the burndown chart
 - Organized and refactored US.01-03
- Refactored Github to reflect user stories

Elliot Vinet – 40210063

User Story 5 Contributions: (1 hrs)

- Structured the database to ensure permissions are easily reachable
-

User Story 6 Contributions: (3 hrs)

- Added rudimentary buttons to test the backend functionality

- Wrote functions to handle button clicks and make server calls
 - Wrote the server-side handling of server call to update the database appropriately when creating channel
 - Added checks to prevent duplicate channels and errors
 - Wrote the server-side handling of server call to update the database appropriately when deleting channel
-

User Story 7 Contributions: (3 hrs)

- Added rudimentary buttons to test the backend functionality
 - Wrote functions to handle button clicks and make server calls
 - Wrote the server-side handling of server call to update the database appropriately when adding user to channel
 - Added checks to prevent duplicate users and errors
 - Wrote the server-side handling of server call to update the database appropriately when removing user from channel
-

Ulysse Allard - 40252823

User Story 5 Contributions: (3 hrs)

- Enable tracking from page to page - (3 hrs)
 - Set up local storage to keep track of signees for specific access (admin / normal users)
 - Ensure to delete from local storage when signees logout
 - Created a personalized message for when user sign-in
-

User Story 6 Contributions: (4 hrs)

- Styled the different buttons to match the general aesthetic of the app
 - Created the space and styled the private chat.
 - Worked on trying to enable admins to assign users
 - Worked on separating admin and user visibility
-

User Story 7 Contributions: (30 min)

- Styled the different buttons to match the general aesthetic of the app
-

Documentation: (2 hrs)

- Seated up labels, milestone, assignees
 - Modified the Environment for the project
 - Created Task and User Story
 - Set up the workflow and the .yml file
-

Yénita Aman – 40293370

User Story 5 Contributions: (4 hrs)

- Reviewed peer code and optimized it
 - Added functionality to the create channels button
 - Added functionality to the delete channels button
 - Allowed buttons to create modifications to the database
-

Additional Contributions: (3h15)

- Creation of document to help team members (20min)
 - Creation of the readme file, took longer due to lack of GitHub experience (1h15)
 - Edit of the detailed time log with the GitHub user stories and tasks (1h40)
-