GINA CODY SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

SOEN 341 Software Process

Team Project, Winter 2025

Project Title:

ChatHaven, a seamless communication application

Objective.

This project will help you get a taste of software development skills firsthand. You will follow the Agile development approach; take advantage of GitHub distributed version control plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis to support your project management process. The project is divided into 4 incremental deliveries which we refer to as sprints based on Agile Scrum methodology, which will be used in this course.

Because of the short span of this project, you are not expected to deliver a marketable product, but the result should be at least a compelling middle-fidelity prototype that could serve as the basis for building a real product. Check these two links on prototype fidelity quite helpful: A Guide to Prototype Fidelity:

- https://www.webfx.com/blog/web-design/design-mockup-fidelity/
- https://www.webfx.com/blog/web-design/wireframes-vs-prototypes-difference/

Description

ChatHaven is a versatile communication platform designed for seamless interaction through text channels and direct messaging. Whether you're collaborating with a team, connecting with a community, or chatting privately with friends, ChatHaven offers an intuitive and organized space to stay connected. Users can create topic-specific channels, share media, and engage in one-on-one conversations, all while enjoying customizable features that enhance the experience. With a clean interface and robust privacy options, ChatHaven is the perfect hub for meaningful and efficient communication.

Baseline features:

1. **Text Channels for Group Communication**

- **Users can join a set of channels (e.g., "General," "Project Help," "Social").**
- **Messages sent in a channel are visible to all users in that channel.**

2. **Direct Messaging Between Users (DM)**
   - **Users can message other users. Conversations using DM can only be seen by people involved in the DM.**
3. **Role-Based User Permissions**
   - **Two roles: Admin and Member.**
   - **Admins can create/delete channels and moderate messages (delete inappropriate ones).**
   - **Members can only send and view messages**

This is not an exhaustive list of features/users for a system of this type.
By the middle of the semester, your team must propose a fourth feature of a similar effort that will be implemented and integrated with the baseline features.

Your project grade criteria consider originality and innovation. The highest marks will be given to teams that think out of the box and include other relevant users and functionalities.

As an example of applications in this domain, you can consider Discord, Ms-Teams, Slack, etc.

For the first sprint, you have to consider the following core features:

User Authentication & Management:

- Implement User's login (with roles).
- Functionality for Admin to create teams and assign users to specific channels. Ensure channels are visible to both admins and normal users.

Sprint 1 delivery instructions.

Create a GitHub repository named `<team_name-SOEN341_Project_W25>` and organize all project deliverables in subfolders for each sprint. For example, a folder for the minutes.

You also must submit one page to Moodle per team, including your team name, team members with IDs, and a link to your repository. The rest of the work is stored and evaluated directly in your GitHub repository.

The table below provides details on the activities and the corresponding deliverables that must be present in your repository. A detailed grading rubric can be accessed here. This will be used by the markers.

Activities	Sprint Details
1. GitHub setup and initialization 2. README file 3. Sprint Plan 4. User stories and Task breakdown 5. Code to demo	● README file with ○ Description of the Project ○ Team Members and Roles ● The Sprint planning (see Appendix A). ● User stories for Sprint 1 (Use GitHub issues). Identifying them with the prefix US.## ● Task Breakdown (derived from user stories, and assigned to at least one team member). Use GitHub issues, with the prefix Task.## ● Detailed log of each team member's contribution including time spent on each activity (document) ● Meetings Minutes file (minutes files must be named <teamName_Sprint#_meetingnumber_meeting_date> ● You need to demo at least one functionality from "User Authentication & Team Management" for example login functionality.

You must have regular meetings with your team and post their "minutes" in the corresponding subfolder in your repository.

At the end of each sprint, each team member will submit a detailed log of their activities, which will be considered to evaluate the individual contribution.

All members of the team should contribute equally to the project and all contributions must be traceable on GitHub. For informal communication among the team members, you may use the private forum in Moodle.

Total weight 3%.

Appendix A. Sprint Plan.

Create a table with the following columns:

- Issue # from your repository.
- User story/Task title.
- Story points associated with a User story (US)/task.
- Due date.
- Associated tasks/Task description.
- Priority (low, medium, high) based on your discussion with your TA.
- Risk (low, medium, high) and a brief explanation.
- Responsible. The team member who is accountable for the task.

As an example, consider a project from a previous term.

https://docs.google.com/spreadsheets/d/1XJVPkzoddaY0HJMaGUNn_-xZByVyE6RBzXC0nnBDX_g/edit?usp=sharing

https://docs.google.com/spreadsheets/d/1LNb9UJPpciZ9EqzjkkmBA0BT8gKLtKMuAKInAM_gUqQ/edit?usp=sharing

Remember that this table is the result of your team meetings, and the discussions with the TA, and/or instructor. All decisions must be documented in the minutes of your meetings, for example, how you estimate story points for each task.