Node Messenger

Development Plan - Group 24

Ahmed, Tasin	Patel, Shardool	Elemary, Omar
400052747	400080843	40010169

September 28, 2018

The goal of our team is to recreate the Tinode Web Messenger to unite people through the power of the internet. This document will be useful in order to guide our team through the project, and ensure every member plays a part in the development of the messenger.

1 Meeting Plan

Each Tuesday at 5:30pm, the team will meet at Mills Memorial Library. The meeting will take place on the second floor booth area where the scribe will record interactions for the meeting plans. The project manager will delegate work and organize tasks in preparation for Wednesday night's lab.

2 Communication Plan

Communication will mainly take place in our Facebook group created on messenger where notes and ideas are exchanged. We will use Facebook to facilitate updates and inquire about any logistical issues. Any shared documents are uploaded to Google Drive and Overleaf so all members can participate in completing work together. Programming for the project will be done both separately through specifically assigned tasks and modules by the project manager or in group through Visual Studio Code's **Live Share** feature with communication facilitated by a Discord channel. The code and the documentation for the project will be available on GitLab repository giving all members access to project files while allowing project manager to keep track of revisions made.

3 Member Roles

The following table outlines some of the roles that each team member will take on over the course of this project.

Project Manager	Shardool Patel
Developers	Shardool Patel, Omar Elemary, Tasin Ahmed
Documentation	Shardool Patel, Omar Elemary, Tasin Ahmed
Git Manager	Shardool Patel, Omar Elemary
Team Scribe	Omar Elemary, Tasin Ahmed
Gantt Expert	~~Shardool Patel~~ Omar Elemary

4 Git Workflow Plan

The team will follow the feature-branch workflow for the development of Node Messenger. Feature branches will ensure the master branch does not contain any broken code and help make the application modular. In addition, the team can collaborate and receive feedback on features with the use of pull requests before they push it to master branch.

The team will use tags for approved components/features and milestone to track the completion of major components.

5 Proof of Concept Demonstration Plan

The team has to deal with the risks of implementing third party libraries that support client-server communication. The Proof of Concept will validate the use of back end web-socket/polling library such as socket.io by showcasing basic functionality of sending and receiving messages using a simple interface. By demonstrating a proof of concept that is functional, the team will mitigate the implementation risks. The product during proof of concept stages will be tested for the important functional requirements through unit testing for individual elements followed by integration testing between the modules and a system test for the whole structure. As this is a locally hosted web-app, the application modules and the app will already be pre-loaded on the demonstration device with all the necessary node.js modules installed and activated.

6 Technology

In order to create the software, we will be using a combination of HTML, CSS, Javascript, and React. HTML will be used to create the core of the software. CSS will be used to add visuals to the software. Javascript will be used to make the software dynamic and functional. React will be used in order to put together the user interface. We will be using Windows computers in order to test our software. The code will be written on Visual Studio Code text editor instead of an IDE **(Integrated Development Environment** for its many available extensions such as the **Live Share** feature used for group programming. Documents will be written and compiled through a LaTeX software **TexMaker**. If the document is worked on concurrently by all group members, the group will use the same LaTeX technology available through **Overleaf**.

7 Coding Style

We will be sticking to traditional HTML, CSS, and Javascript programming methods. The HTML file will only contain the core of the program. The CSS will contain all the color and design for our website. The Javascript file will control all the data, and functions in order to make the website work. Every individual tags in the HTML file will be indented to make it easier to read, and follow. We will use Camel Case for naming variables, and all uppercase for constant variable names. If we all follow the same coding style throughout the project, we will be able to code peacefully and without any conflict when merging code.

8 Project Schedule

The project schedule is outlined in ProjectSchedule/ProjectSchedule.gan. The schedule is not concrete and is subject to change and maintenance by our Gantt expert based on major team updates imposed by the project manager during team meetings. The Gantt project schedule will become a corner stone for our team progress throughout the development of the project.

The project schedule can be viewed through the Gantt Chart found at:
`https://gitlab.cas.mcmaster.ca/pates25/NodeMessenger/tree/master/ProjectSchedule`

9 Project Review

9.1 Revision 0

Date	Developer(s)	Change
2018-9-26	Tasin Ahmed	Document created
2018-9-28	Tasin Ahmed	Initial Revision
2018-9-28	Omar Elemary	Initial Revision
2018-9-28	Shardool Patel	Initial Revision

9.2 Revision 1

Date	Developer(s)	Change
2018-11-25	Omar Elemary	Final Copy