eTimely

Development Plan

Team Members:

Matt Korte, Caleb Obi, Chris Gumieny, Samia Chowdhury

1.1 Project Overview

Our project is aimed at providing tools for restaurants as well as their staff to simplify the scheduling process. This means giving the staff a common place to communicate their work availability, requests off, and anything else work related to management and coworkers. As of right now, many restaurants rely heavily on word of mouth and memory, leading to frequent mistakes being made. We want to help mitigate these mistakes by creating a web application that allows management to create a schedule based off employee entered availability.

1.2 Project Purpose, Scope, Objective

The purpose in developing this application is to give businesses a way to manage and communicate employee schedules in an efficient and prompt manner. It is our hope that such a service will help reduce scheduling and communication problems between coworkers and management.

The scope of this project is to develop a web application that gives restaurants and other similarly sized businesses the tools to better create and communicate schedules. This will be done by having two separate account types, “business” and “staff.” Each staff member will have their own account in which they can post their weekly availability and request time off. This information will be sent to their workplace’s business account for approval. A manager will then use this information to construct a schedule for the staff. Once the manager creates a schedule, it will be available for the staff to view. In addition to this, we will be adding a functionality for two employees to send management a request to trade shifts. If approved, the schedule will automatically reflect that change.

As for the objective of this project, we want to create a web application that will help the business manage time effectively. In doing this, our web application will allow schedules to be made in less time, with less errors. This will cultivate a practice which will make better use of time and cut down on inefficiency.

1.3 Team Organization (Roles and Responsibilities)

Each member of the team will be involved in every element of this project. Team members will be held accountable for completing all tasks by the deadline. Special roles on the other hand have been assigned to each team member, giving them added obligations in managing elements of the project. The roles which have been assigned to each team member, as well as an explanation of what each responsibility includes are listed below:

- Matt Korte – Team Lead
    - The team lead will be responsible for overseeing every aspect of the project and will oversee assigning tasks between the team. They will also be the main point of contact for the GTA and instructor.

- Caleb Obi – Database/Backend Lead and Quality Assurance Lead
  - The backend lead will be responsible for managing and leading the development of server-side applications, as well as ensuring a seamless link between the backend and database.
  - The quality assurance lead will be responsible for managing the testing of all components of the project. They will also ensure that all code implementations meet the client's specifications and expectations.
- Chris Gumieny – UI/Frontend Lead
  - The frontend lead will be responsible for the development of our application's user interface. Additionally, they will ensure that all frontend implementations provide our users with a fantastic easy to use experience.
- Samia Chowdhury – Presentation, Documentation, and Communications Lead
  - The presentation lead will be responsible for organizing PowerPoint presentation materials, as well as making sure the team is ready for presentations.
  - Prior to submission, the documentation lead will create, maintain, and oversee the accuracy and promptness of each document. They will also aid the team by developing a template for each document.
  - The communications lead will oversee the planning and distributing of important information announced in class to the entire team.

1.4 Problem Resolution Policies

Our team will take on a preventative approach when it comes to problems that may arise. For example, to keep up with deadlines, we have a daily check-in on Discord. Everyone is expected to either update the team on what they will be working on or what they have worked on by the end of the day. We are also implementing a three-strike policy. A strike can be the result of missing a deadline unless it was clear to the rest of the group the person did their best but ran out of time. It can also be from missing a meeting with less than a week's notice and/or no valid excuse. Strike one results in a warning from the team. Strike two results in a discussion with our GTA. Strike three will result in a discussion with the professor. With any other problem when it comes to workload or differences in opinion, we will resolve conflicts by vote, with the section or team lead being the tiebreaker.

1.5 Project Plan (Iterations and Project Schedule)

Our team has decided to meet twice a week as a group as well as once a week with our assigned GTA. Meeting as a group two to three times a week will allow us to communicate effectively and stay updated on the work of the other team members. We will add more time if needed. Our tentative meeting schedule is as follows:

- Wednesdays 6:00pm – 9:00pm
- Fridays 6:00pm – 9:00pm
- Saturdays 2:00pm – 4:00pm (currently reserved for the GTA meeting)

During the gathering of requirements, our team will decide on a general idea for the project and create a list of features for the application. These features will be split up into three prototypes and will be worked on throughout the development phases. Each week, our team will decide on features to prioritize and will assign tasks accordingly. A tentative project schedule is shown below:

- Requirements Gathering (1/23/2022)
    - Setup GitHub Repository
    - Setup Jira for sprint planning
    - Setup database connection
    - Learn development technologies
    - Finalize functional and non-functional requirements
- Prototype 1 (2/6/2022)
    - Create a home landing page
    - User login and signup authentication
    - Allow employees to input availability
    - Navigation UI
- Design Specification (2/27/2022)
- Prototype 2 (3/6/2022)
    - Allow business accounts to view employee availability
    - Ensure that a schedule can be created by a business and shared among employees
    - Allow business accounts to assign employees roles
        - Ensure that schedules can be created by role
- Prototype 3 (3/27/2022)
    - Email notifications
    - Message Communication Features
        - Message board
        - Business to employee messaging
    - Allow employees to send a request to trade shifts with another employee
- Testing Plan (4/3/2022)
- Final Product (4/17/2022)
    - All features should be implemented and tested effectively

1.6 Configuration management plan

In terms of communication, our team plans to utilize both Discord and Zoom to communicate effectively. Our regularly scheduled team meetings, as well as our meetings with the GTA, will be held over Zoom. Discord will be used outside of meeting times to discuss any important announcements, project details, or concerns.

Our team will utilize Jira to assign and organize tasks among our team members. Tasks will be distributed to everyone by the team lead to ensure that everyone is actively working. Assigned tasks will be observed and verified by the corresponding lead for that aspect of the project. This system will make sure any changes made to the project are made in an efficient manner.

GitHub has been chosen by our team for our means of source control. It will allow us to merge changes together without affecting anyone else's progress. All team members will be required to push their changes into their own branch. Code merging with the main branch will be done by feature and will require approval that must be given by at least two other team members. In most cases, approval will be granted by the team leader and another member of the team.

1.7 Technologies

The following technologies were selected by our team and will be used for the goals of this project:

- Frontend
  - ReactJS v17.0.2 (Client User Interface)
- Backend
  - NodeJS v16.13.2 (Server Development)
  - ExpressJS v4.17.2 (API Development)
- Database
  - Cloud Firestore (NoSQL database)
- Testing
  - Jest v27.4 (Testing Framework)
- Hosting and Deployment
  - Firebase Hosting

Our team has decided to build an end-to-end product for our users using the technologies listed above. We have decided to use ReactJS to create all the user interfaces with which our users will interact. In addition, we have also decided to use NodeJS and ExpressJS to build all our backend servers which will handle all user authentication, routing, and data storage on our database. To ensure that our web application works as expected, our team will use the Jest framework to test all functional components. Finally, we intend to use firebase hosting to host and deploy our web application.