

Project Management SaaS Application Proposal

09.30.2023

Chris Richardson - 1001555307

Gabriela Aldrete - 1001712206

Marshal Marks - 1001901649

# Service/Application Overview

We will create a website with a dashboard that will serve as a cloud-based project management tool. The information entered into the dashboard will be sent to a backend Database for the user to access at any time.

The main features of this application will be:

* Users can create an organization or join an existing organization using a join link.
* A user’s access to information and/or actions are controlled by a role-based permissions system that can be modified by an administrator. The creator of the organization is set as an administrator by default.
* An administrator can create a new project and assign team members to that project from members within their organization.
* Users will be able to select between projects that they are currently working on from a “project hub”.
* Within any given project, a user will be presented with a meeting schedule and a list of active tasks.
* Users can update their unavailability and a backend API will automatically adjust any meetings scheduled to suggest a new best meeting time for the people on the attendance list.
* For each active task, users can view the deadline for that task, create subtasks, or check off tasks/subtasks they have completed.
* Each active task has a section to take notes, post comments, and upload relevant files.

# Components Used

For web hosting and computation, we will use Google Cloud Compute.

For data storage, we will use Google Cloud SQL.

For user identification, we will be using Google Firebase Authentication.

For publishing and pushing any notifications or updates, we will first use PubSub to accomplish this.

For file storage, we will interface with Google Drive as a means to lower the hosting costs.

# Architecture

The user will enter or request data via the front end web page. The data will flow through a specific API which will communicate to the database to store or retrieve the requested data. The following flow diagram describes how the components will be connected to each other.

FLOW DIAGRAM

SCH

TASK

DASH

ID

API

API

API

API

DATA

WEB

SITE

# Design

We will create the front-end website with React, JavaScript, HTML, and CSS.

We will create the back-end APIs with JavaScript running with NodeJS as well as some written in Python. Some members are more proficient in JS while other members are more proficient in Python, so we plan on interchangeably using these languages where feasible.

For user identification, we will be using Google Firebase Authentication. This service provides an API that can be used to register a new account with an email address and password or allow a user to log in with a previously registered email address and password. Once logged in, the API provides a unique user ID that can be used to personalize the information displayed on the screen to the current user. This API also contains a database that encrypts and stores the login details for the whole application, meaning that most of the back-end logistics of authentication and data storage are taken care of automatically.

# Implementation Plan

For the front-end design of the application, we will create the following components:

* Login and Profile page which require firebase authentication
* Dashboard page for the summary of all tasks
* Tasks information page for the detail description of the user tasks
* Schedule page for meetings and project deadlines.

For the back-end design of the application, we will create the following components:

* Database to store all the information entered by the users
* Several APIs for front-end/back-end communication
  + API for bulk-loading the information of the users.
  + API for loading personalized info for each user
  + API to add/remove comments
  + API to handle the meeting schedules
  + API to handle user unavailability

Visual Design and testing are part of the planning to implement this application

We are planning to tackle the development of the project in three phases. In addition, we will plan to meet on a weekly basis to discuss updates, blockers and work progress. This strategy will allow us to maintain a timeline and to help each other.

Phase ONE is planned to be completed by October 26th, 2023

Phase TWO is planned to be completed by November 30th, 2023

Phase THREE is planned to be completed by December 5th, 2023

The table below shows the breakdown of each task by responsibility and timeline

|  |  |  |
| --- | --- | --- |
| TASK | RESPONSIBLE | PHASE -DEADLINE |
| Login and Profile page which require firebase authentication | Marshal Marks | Phase I - Oct 26 |
| Dashboard page for the summary of all tasks | Marshal Marks | Phase I - Oct 26 |
| Tasks information page for the detail description of the user tasks | Gabriela Aldrete | Phase II - Nov 30 |
| Schedule page for meetings and project deadlines. | Marshal Marks | Phase II - Nov 30 |
| Database design/creation | Chirs Richardson | Phase I - Oct 26th |
| Data Creation for testing | Gabriela Aldrete | Phase I - Oct 26 |
| API for data loading | Marshal Marks | Phase I - Oct 26 |
| API for personalized data | Chris Richardson | Phase I - Oct 26 |
| API to add/remove comments | Gabriela Aldrete | Phase III - Dec 5 |
| API to handle meeting schedules | Chris Richardson | Phase II - Nov 30 |
| API to handle user availability | Gabriela Aldrete | Phase II - Nov 30 |
| Visual Design | Marshal/Gabriela | Phase III - Dec 5 |
| Testing Phase I | Three members | Phase I - Oct 26 |
| Final Product Testing | Three members | PhaseII/III |

# Test Plan

For basic unit testing and assessing the functionality of each individual component, we will implement the JEST testing framework and the Nodemon monitoring script. JEST will allow us to write unit tests for each JS script that are checked automatically when the code is run, and Nodemon is a script included within NodeJS that automatically restarts a script when the file is saved. This combination should provide a tight development loop that runs a custom set of unit tests every time the code is saved.

For more general application functionality, a lengthy testing phase will be reserved for the end of development to simulate this product being used in a real-world environment. We will create a series of dummy projects and test the functionality of the application in a live environment. If any major problems occur during this phase, we will fix bugs and errors where they occur.

# Team Work Contract

Team Name: Group 3 - Cloud Computing Date: 9/30/2023 .

|  |
| --- |
| GOALS: What are our team goals for this project?  What do we want to accomplish? What skills do we want to develop or refine? |
| Our goal is to develop and deploy a project management website. This website provides its users the ability to track projects, their associated notes, the workers currently assigned to it, assist on scheduling meetings, and other productivity related tasks. We want to accomplish this by developing our skills both in front-end and back-end website development |
| EXPECTATIONS: What do we expect of one another in regard to attendance at meetings, participation, frequency of communication, the quality of work, etc.? |
| We expect ourselves to maintain reliable means of communication, and to respond within an appropriate time span given the urgency of the received communications. Meetings, either in-person or online, will be held approximately, but not limited to, three times a week. These sessions will allow us to share progress, assist one another with any hurdles, address new/upcoming deadlines, or discuss impending changes. We expect ourselves to hold to our deadlines, or clearly communicate any limitation in advance so that the group can address it. As we intend to improve our skills throughout this project, our expectation for our own work is to grow throughout this project. At all times, each of us expects ourselves and each other to give a genuine attempt at all work, and to ask questions before any of us reach an impasse. |
| POLICIES & PROCEDURES: What rules can we agree on to help us meet our goals and expectations? |
| 1. Be responsive to inquiries or comments on the group’s What’sApp. 2. Upon agreeing to a meeting time, attend said meeting, or provide advance notice if a complication arises. 3. Follow through with all work one has volunteered to do. 4. Bring forward any, and all questions during meeting times. 5. Be honest with yourself and the team about your skill level. 6. Do not volunteer for more work than one can support at that time. 7. Attend class as often as possible. 8. Be respectful of each other and each other’s time. |
| CONSEQUENCES: How will we address non-performance regarding these goals, expectations, policies and procedures? |
| In the scenario that a member fails to meet the above sections, the group member has refused to coordinate with the remaining members, and does not apply themselves as expected by a student at the University of Texas at Arlington, the teacher and TA will be notified and invited to mediate a suitable solution. |

We share these goals and expectations, and agree to these policies, procedures, and consequences.



Team member name

Gabriela Aldrete

Team member name

Marshal Marks

Team member name

9/30/2023

Date

9/30/2023

Date

9/30/2023

Date

# Authorship and Workload Agreement

Marshal Marks, Chris Richardson, and Gabriela Aldrete, we, equally contributed to the completion of the above proposal.

I, Gabriela Aldrete, agree with the contents of this proposal, and the workload I have been assigned.

Gabriela Aldrete

I, Chris Richardson, agree with the contents of this proposal, and the workload I have been assigned.



I, Marshal Marks, agree with the contents of this proposal, and the workload I have been assigned.

Marshal Marks