CAPSTONE 4ZP6A PROJECT

Project Plan

Version 1

FINDDIT

GROUP 19

Prakarsh Kamal

Japnit Singh

Kanwar Sandhu

Hriday Jham

Table of Contents

Team Meeting and Communication Plan

We will be using the following modes of communication:

- Google Drive and Google Docs for document sharing and collaboration
- Discord for team meetings
- Gitlab issue board for tracking our work through issues
- Gitlab for version source control
- Figma for creating design wireframes for the UI/UX

Team Member Roles

1. Prakarsh Kamal - Coordinator
    a. Responsible for creating design mockups
    b. Responsible for styling application screens to match the mockups
    c. Responsible for creating the card stack shown with restaurant data to the application users
    d. Unit testing using Jest

Functional Requirements responsibilities:

a. 4.1 Swipeable Stack UI

b. 4.2 Swipe to Vote

c. 5.1 Ranking Leaderboard

Non-functional Requirements responsibilities:

a. Appearance

b. Style

2. Japnit Singh

a. Responsible for setting up the Firestore database for the application

b. Responsible for creating backend functionality

c. Cost optimization algorithm

d. Manual verification testing

Functional Requirements responsibilities:

a. 2.1 Group Creation

b. 5.1 Ranking Leaderboard

c. 7.1 Cost function optimization algorithm

Non-functional Requirements responsibilities:

a. Speed and Latency

b. Reliability and Availability

3. Kanwar Sandhu

a. Responsible for creating frontend screens following the user flow

b. Integrating frontend with backend APIs

c. Unit testing using Jest

d. Cost optimization algorithm

Functional Requirements responsibilities:

a. 3.3 Group preferences
b. 3.2 Switch between groups
c. 7.1 Cost function optimization algorithm

Non-functional Requirements responsibilities:

a. Appearance
b. Style

4. Hriday Jham
   a. Firebase project setup for authentication
   b. Responsible for allowing application users to sign in with Google
   c. Integrating Google Maps widget with the app frontend
   d. Implement Notifications using Firebase Cloud Messaging

   Functional Requirements responsibilities:

   a. 1.1 User Registration
   b. 3.1 Join Group
   c. 6.1 Accessing History

   Non-functional Requirements responsibilities:

   a. Speed and Latency
   b. Reliability and Availability

Workflow Plan

A. We will be using Gitlab for version control. All code and team documents will be updated on the Finddit Gitlab repository. There are 2 important branches, "**main**" and "**develop**". When working on a task, each team member will create a branch from **develop** and work independently on that branch. Once the functionality is implemented, a merge(pull) request will be created and a team member will review the code. The team member will also pull and test the changes locally to ensure system functionality. After any suggested changes, the merge request will be approved and merged into **develop**. These steps are outlined below:

   a. Pull any changes from **develop**
   b. Create a new branch to work on
   c. Implement code changes for the assigned task with proper formatting
   d. Test changes on iOS and Android emulators
   e. Commit and push changes to the branch and create a merge request
   f. A team member will review and manually test the changes locally on their system
   g. Suggest any changes or errors to be fixed on the merge request
   h. Approve the merge request and merge it into **develop**

B. Yes, we will follow an agile approach. We have created an issue board on Gitlab to track issues and provide visibility on each member's tasks.
C. We are storing all data in our Firebase database.
D. It is not applicable since the project does not use ML.
E. We will be using Firebase Performance Monitoring for our app startup/load times as well as the response times for HTTP requests. We will do this on both iOS and Android platforms.

Proof of Concept Demonstration Plan

A. For the Proof of Concept Demonstration, we will have a mock, barebones, working app that showcases some of our functionality. There will be a frontend to see the user flow of authentication, group creation and preferences, restaurant card rendering, and card swiping. These features will only work for an admin user for the POC. All this will be hooked up to the backend server pulling data from the Google API and the database. To go along with the demonstration, we will have a slide deck to show our features and any decisions we made. It will also include some of our wireframes from Figma.

Technology

A.

- Programming Languages:
    - JavaScript
    - TypeScript
    - HTML
    - CSS
- Frameworks:
    - React Native
    - Expo
    - Nest.js
- Database
    - Google Firestore
- Cloud Service
    - Google Cloud
        - Firebase Firestore
        - Firebase cloud messaging
        - Google Maps APIs
- Developer Tools and Environments
    - Gitlab
    - Figma
    - VS Code
    - XCode
    - iOS Emulator
    - Expo GO App for emulating
    - Android Studio
    - Android SmartPhone Emulator

    - Ngrok for testing API locally

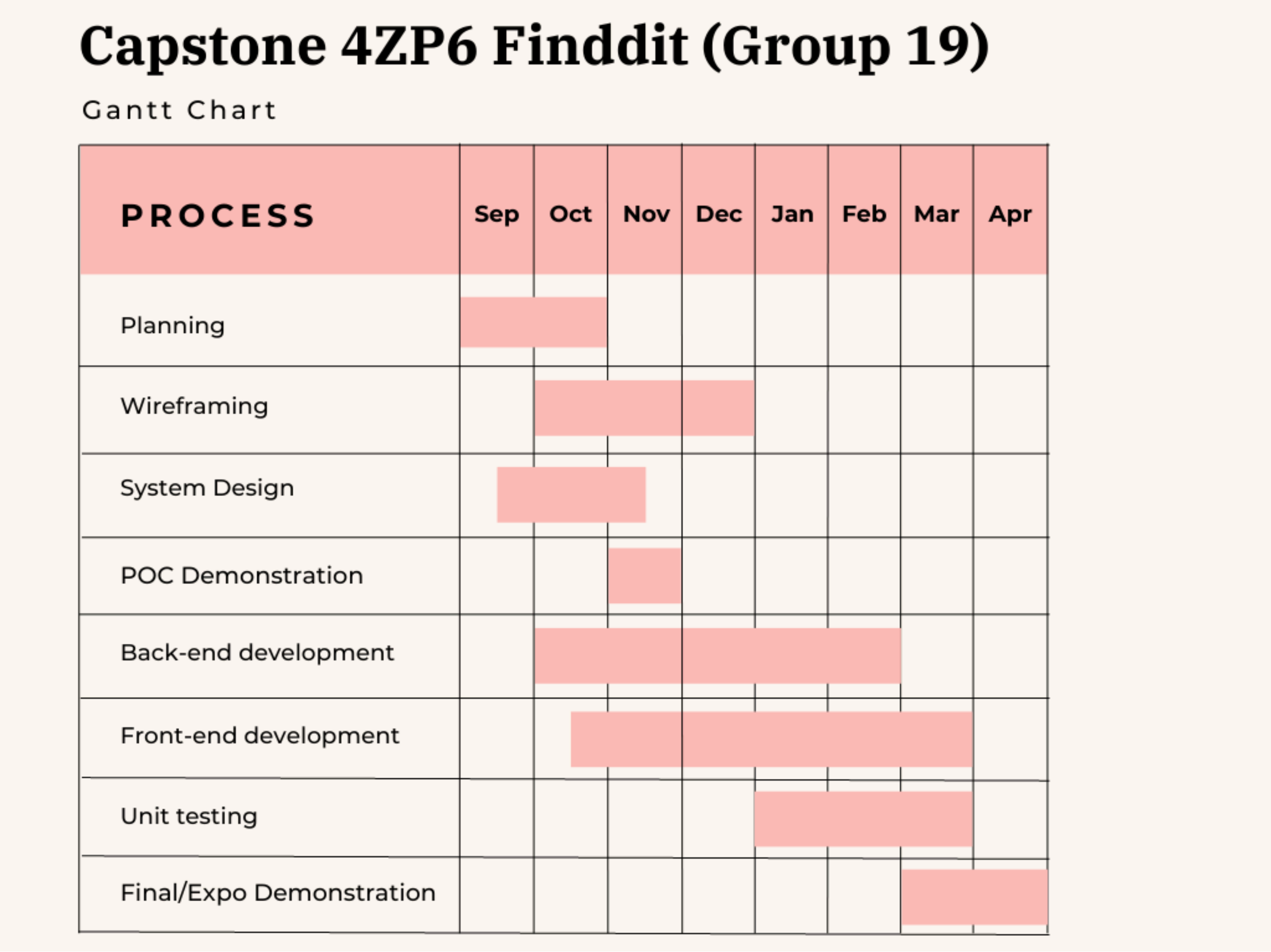
- Testing
    - Jest Framework for Unit Testing
    - System Testing
    - Manual Testing

B. Not Applicable

C. Not Applicable

Project Scheduling

Below is a Gantt Chart providing a high-level overview of our project schedule.



Capstone 4ZP6 Finddit (Group 19)

Gantt Chart

PROCESS	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Planning	■	■						
Wireframing		■	■	■				
System Design	■	■	■					
POC Demonstration			■					
Back-end development		■	■	■	■	■		
Front-end development		■	■	■	■	■	■	
Unit testing					■	■	■	
Final/Expo Demonstration							■	■