COSC 4353 ASSIGNMENT 1, GROUP 15:

Salma Abbady Dan Kalhori Geethika Komma

1. Initial Thoughts

- Consider user experience: How will users (volunteers and administrators) interact with the application?
    - Both volunteers and administrator will create logins, verify emails, and set up profile for first time
    - All details in profile except for email will be changeable later on (address, availability, etc)
    - Both user types will have access to calendar
    - Administrators will have edit access to create and manage events
    - Volunteers can look through all events available, filter by different filters, and sort by date/priority/distance
    - Dashboard that displays upcoming events, matched tasks, and notifications/messaging from administrators
    - Way to track volunteer performance and history for admin, as well as leave comments/feedback on individual tasks
- Identify the key functionalities: What are the essential features the application must have?
    - User registration and 2-factor authentication
    - Profiles with relevant information such as location, availability, etc.
    - History to track/show previous volunteer experience
    - Messaging/event system for volunteers and administrators to communicate with each other
    - Calendar system to track and manage events
- Technology stack: What technologies might you use for front-end, back-end, database, and other components?
    - Front-end: Flutter for web and mobile apps
    - Back-end: Flutter again to make the code the same across front and back for more simplicity/readability. Will handle API calls, etc.
    - Database: Firebase or MongoDB, most likely Firebase as it has a notification system as well

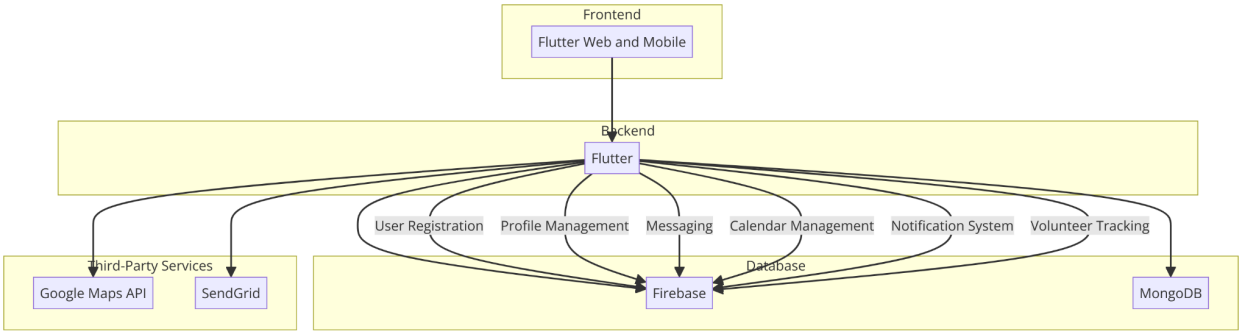
    - Using Flutter to allow for seamless coding between front-end, back-end, as well as database
    - Firebase can be used for both the database as well as the notification system, with simple integration into Flutter. We want to keep the different languages and separate pieces to a minimum

2. **Development Methodology**

- Explain why you would choose a particular development methodology (e.g., Agile, Waterfall, DevOps).
    - We will be using Waterfall because of the nature of these assignments and deadlines. Considering we already have most of our requirements laid out for the entire project, it is better for us to work on the planning, design, and implementation in their respective order
- Discuss how this methodology will help manage the project effectively.
    - Using a linear methodology will allow us to flush out all ideas and issues that we need to address before we start building the different components. The stipulation for us to finish each phase in order will help us manage our time on the project, making it deadline-friendly

3. **High-Level Design / Architecture**

- Create a diagram to illustrate the overall structure of your application.



- Identify the main components (e.g., front-end, back-end, database).
    - Front-end: Flutter
    - Back-end: Flutter
    - Database: Firebase and/or MongoDb
- Describe how these components will interact with each other.
    - The user interacts with the mobile app or webpage(frontend)

    - The frontend will communicate with the backend, usually by an API call to make a certain action, e.g. update profile, messaging, etc.
    - The back-end will communicate with the database for retrieving data, updating information, etc.
    - The back-end may also communicate with third–party services such as Google API, for Firebase and Google Maps, and SendGrid for email notifications
- Mention any third-party services or APIs you plan to integrate.
    - Google Maps API for event locations and mapping
    - SendGrid for email notification

GROUP MEMBER NAME: CONTRIBUTION

Salma Abbady: Initial thoughts, key functionality, technology stack

Dan Kalhori: Initial thoughts, development methodology, high-level design/architecture

Geethika Komma: Sanity/double check, proof-read, final check