**Software Design - HW1**

**Group 2**

**1. Initial Thoughts**

**User interaction:**

* Volunteers need to register, update their profiles, view available events, and get notifications.
* Administrators should have access to features for creating and managing events, viewing volunteer profiles, and tracking volunteer participation history.

**Identify the key functionalities:**

* We will make a dashboard to show their current and completed assignments.
* We need an administration role that has access to tools to create and manage events, view volunteer profiles, and match them to tasks efficiently.
* We need an admin panel, with more features than regular users.
* The application must secure registration with email verification and login.
* Profiles of users should include location, skills, preferences, and availability, and can be edited at any time.
* Tools for administration to create and manage events.
* Real-time notifications for assignments, updates, and reminders.

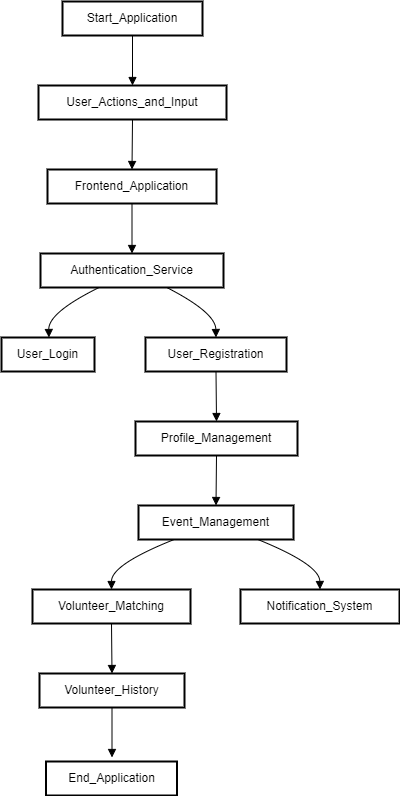
**Technology stack:**

* Front-end: React.js for a responsive user interface
* Back-end: Node.js with Express for handling API request
* Database: MongoDB to manage user profiles, event, and history
* Authentication: Auth0 to secure user authentication and email verification, easy to implement
* Notification: Firebase Cloud Messaging
* Deployment: Vercel
* Hosting: Amazon Web Services (AWS)

**2. Methodology: Agile**

Agile is more flexible than other methodologies. It allows changes to be easily made. It helps to improve communication between developers and clients. It reduces the risks associated with complex projects by breaking down them into smaller sprints. It is involved in the development process and provides feedback at each stage of the project, which often leads to increased customer satisfaction.   
We choose Agile because of its flexibility and adaptability. Agile allows for iterative development and continuous feedback, which is essential for adapting to a client's changing needs. The project is divided into manageable parts. Regular meetings and reviews enhance communication between developers, volunteers, and administrators, ensuring the application meets users' expectations.

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

****

**Main components:**

* Front-end: Web application built with react, provides user interfaces.
* Back-end: Node.js handles API requests, business logic, and communication with the database. Express framework provides features for the web application.
* Database: MongoDB stores user profiles, events, tasks, and volunteer history.

**Interaction:**

Users register and authenticate via the front-end, which communicates with Auth0 (the authentication service). Users then update their profiles, which are stored in MongoDB via the back-end. Admin can create and manage events from the admin panel, which updates the database.

**Third-party services and APIs:**

* Auth0: user authentication
* Google Map API: to determine and display volunteer and event locations.
* SendGrid: email verification and communication.
* Firebase Cloud Messaging: real-time notification.

| **Group Member Name** | **What is your contribution?** |  |
| --- | --- | --- |
| Matin Amoozadeh | Initial Thoughts and Key Functionalities. |  |
| Thinh Pham | Advocated for Agile methodology due to its flexibility and adaptability. |  |
|  |  |  |
| Shadmun Talukder Shahed | Outlined the main components: React-based front-end, back-end. |  |

It was a collaborative effort where we discussed each question thoroughly. In other words, all group members contributed to every aspect of the project.