Software Implementation and Testing Document

For

Group <7>

Version 2.0

Authors:
Alek Coupet
Bilal Taleb Sayed
Haidar Alhussaini

1. Programming Languages (5 points) We are using the following:

- **JavaScript** for the frontend (React) and backend (Node.js).

- **MongoDB** as the NoSQL database for task, user, and organization data.

2. Platforms, APIs, Databases, and other technologies used (5 points)

- **MongoDB Atlas**: Cloud-based MongoDB for database management.
- **Express.js**: Backend framework for handling APIs.
- **React.js**: Frontend framework for user interface.
- **Node.js**: Server-side JavaScript runtime.
- **FullCalendar**: Calendar integration for task deadlines.
- **JWT**: For user authentication and session management.

3. Execution-based Functional Testing (10 points)

We tested our new functionalities by applying them. We performed CRUD on Task, we used the invitation features to invite Users to teams and organizations, and used the assigning feature to assign tasks based on role (A sup can assign to members, a member can't assign to supervisor). We ran the app and could see our new implemented views. All new buttons were tested individually through multiple test rounds and case scenarios.

4. Execution-based Non-Functional Testing (10 points)

All Tasks created by Users appeared in Data base as expected. Updated Profiles were seen updated correctly in the database. The organizations and Teams along with invitations appeared in the data base in the expected spots.

5. Non-Execution-based Testing (10 points)

Team members reviewed their own code ensuring the use of generic programming and readability of their code.