
# Disaster Relief Management System

## Developed By

**Huzaifa Tanzeel (20K-0305)**

**Agha Maarij Amir (20K-0160)**

**Rayan Ali (20K-1083)**

**Imtiaz Mushfiq (20K-0313)**

# Contents

# 1 Introduction

The motivation for this project were the severe floods that affected Pakistan earlier this year, the government and the relief workers need a platform which can make communication between the two bodies easier, hence leading to maximum efficiency in relief work. Unfortunately, the absence of any such platform made collaboration very difficult and motivated us to work on this idea. The purpose of the disaster relief management system is to ease relief work and to create a convenient and easy-to-use collaboration platform for the government and organizations, working for relief operations. The system is based on a relational database with its relief work management functionalities. We will have a database server supporting hundreds welfare/relief work organizations all under control of an administrator. Above all, we hope to provide a comfortable user experience

# 2 Project Requirements

## 2.1 Functional Requirements

1. The project infrastructure must be set up.
2. A frontend using ReactJS, HTML, CSS, and Bootstrap must be created.
3. A backend using NodeJS and JavaScript must be created.
4. Documentation of integration components and processesA database schema must be created using MySQL Workbench.
5. A login page, home screen, and navigating pages for disasters, relief programs, organizations, and products must be created for PDMA.
6. A login page, home screen, and navigating pages for Relief programs, Requirements, Commitments, Fulfillments must be created for Organizations.
7. PDMA user must be able to Login in the system.
8. PDMA user must be able to add and remove products that they need for raising requirements in a relief program.
9. PDMA user must be able to add new organization categories.
10. PDMA user must be able to approve organizations after they sign up.
11. PDMA user must be able to manage disasters by adding, removing, and viewing them from the system.
12. PDMA user must be able to manage disaster categories by adding new categories in the system.
13. PDMA user must be able to manage disaster locations by adding, removing, and viewing locations for disasters.

14. PDMA user must be able to create relief programs for the disasters.
15. PDMA user must be able to raise product requirements for the relief programs for the respected disaster location.
16. Organizations must be able to signup.
17. Organizations must be able to Login after approval from PDMA.
18. Organizations must be able to register in relief programs.
19. Organizations must be able to view product requirement summary for their designated relief programs.
20. Organizations must be able to commit against a particular requirement raised by the PDMA for a particular disaster location in a relief program.
21. Organizations must be able to commit against their Commitments.
22. Organizations must be able to view their Commitments and Fulfillments.
23. Organizations must be able to Logout from the system.
24. The system shall maintain sessions for every user to enable the user to resume their activity upon switching different screens of the system.

### 2.2 Non-Functional Requirements

1. The frontend must be responsive.
2. The system must be scalable to handle a large number of organizations, relief programs, and product requirements.
3. The system must be secure, and user authentication and authorization must be implemented.
4. The system must have good performance and response times.
5. The system must be easy to use and intuitive for PDMA users and organizations.

## 3 IDEs/Tools/Languages Used

- Reactjs, HTML, CSS, and Bootstrap are used for the frontend development of the project.
- Nodejs and JavaScript are used for the backend development of the project.
- MYSQL Workbench 8.0 is used as the database management system.
- GitHub is utilized as a tool for contributing and communication between team members.
- Jira is used for assigning tasks to the group members.
- Lucid chart is used for creating the work breakdown structure of the project.

# 4 Tasks Distribution

The project team consists of four members: Huzaifa Tanzeel, Rayan Ali, Maarij Aamir, and Imtiaz Mushfiq. The task distribution for the project is as follows:

- Huzaifa is responsible for requirements engineering, database design and schema creation, some backend development, and task management among team members.
- Rayan Ali is responsible for creating the UI of the project, designing UI for various website pages, and testing the project.
- Maarij is responsible for managing backend development, writing client and server-side code for backend logic.
- Imtiaz is responsible for Testing, creating Gantt chart, WBS, and different diagrams for the project.

# 5 Work Breakdown Structure (WBS)

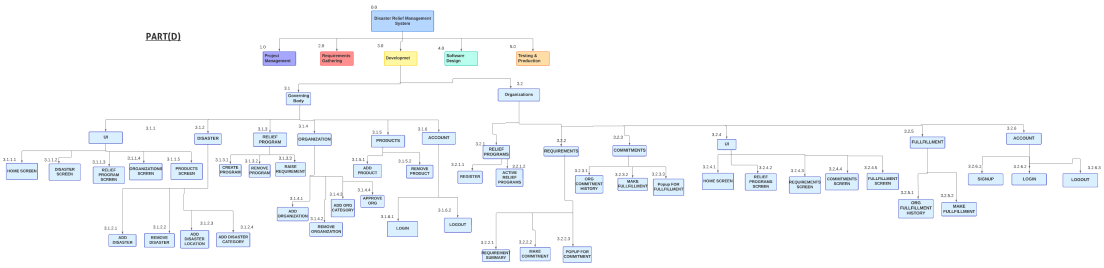


Figure 1: Project WBS

# 6 Project Gantt Chart

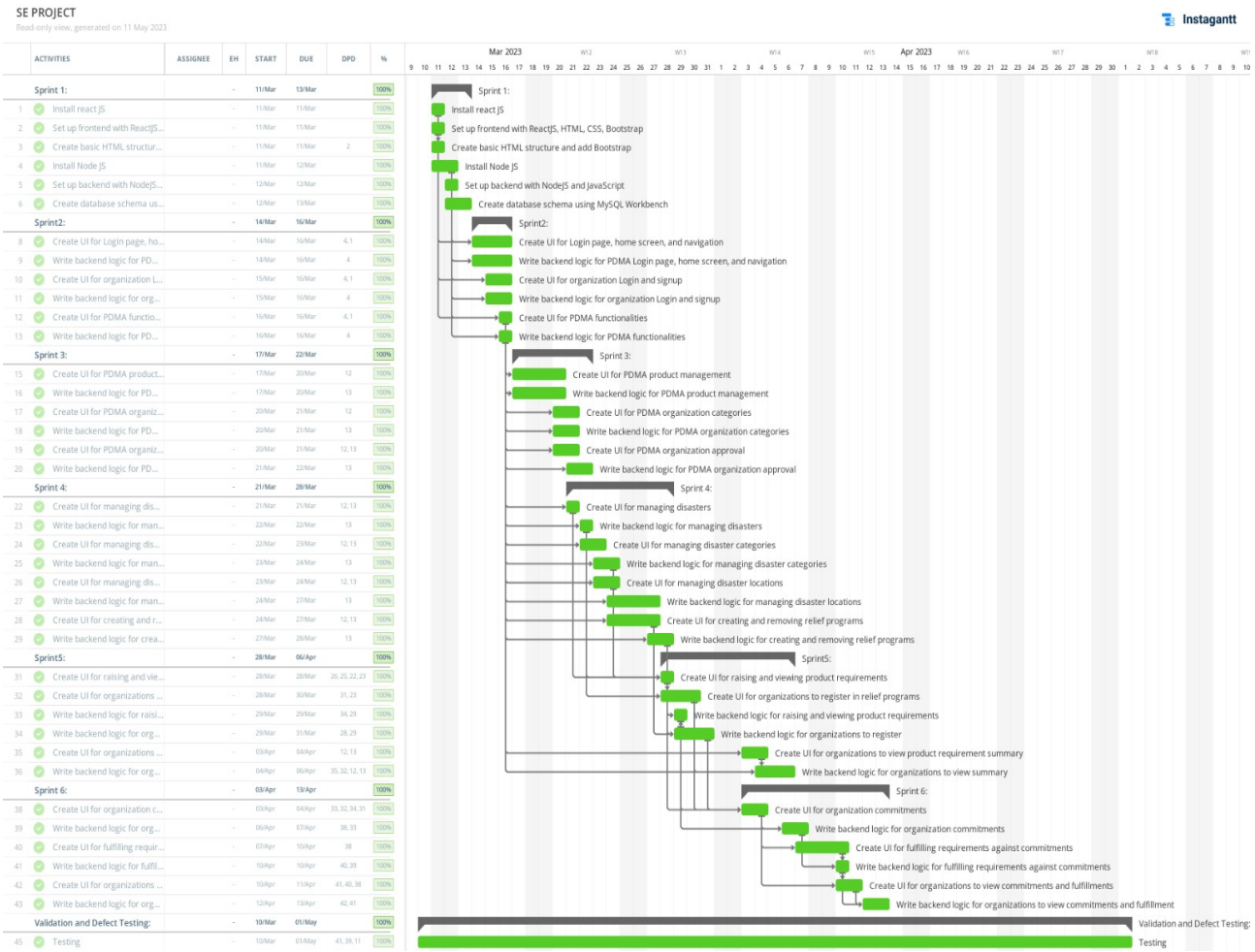


Figure 2: Project Gantt Chart