

21BDS0340

Abhinav Dinesh Srivatsa

Software Engineering Lab

Assignment – I

Scope

Objectives

The primary objective of this project is to create an effective bug and issue tracker for organizational level software development. This project focuses on ease of creation of issues, having detailed teams and organizations for collaboration, and a seamless process to manage bug and issue history.

Deliverables

The main deliverables included in the bug tracking software are the following:

Profile

- Profile creating and updating
- Login with email and password and session management with a single access token
- Email verification

Organisation

- Organisation creating and updating
- Managing organisations

Teams

- Creating of dedicated teams
- Adding members to teams

Bugs

- Bugs creating and updating
- Adding status and priority to bugs
- Assigning teams for bugs
- Maintaining a details bug and issue history

UI/UX

- Detailed bug and issue views
- A clean and concise interface to create organizational structures

Miscellaneous

- Detailed email views for team members when bugs and issues are created by the quality assurance team

Project Exclusions

This project will receive no updates to functionality and the user interface. The project will also not be hosted online. All testing will be done locally.

Constraints

The constraints for this project are listed below:

- Estimated time to complete project is 10 weeks
- Estimated budget is 0 (no budget constraint)
- Estimated workers is 1
- Customer satisfaction and revisions of user interface and functionality

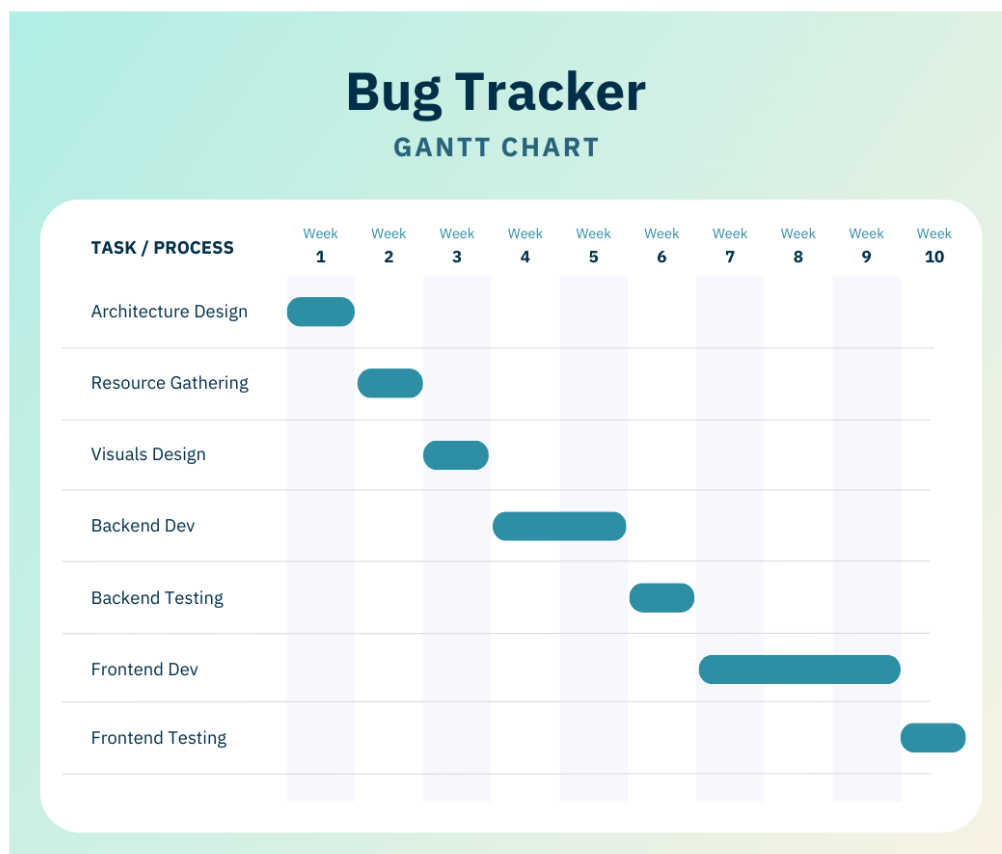
Assumptions

The organizational hierarchical structure and the team interface are assumed to be split properly between developers, product specifiers, and quality assurance. These team members may be named otherwise, but the assumption is that one profile may not be a part of two different teams.

Another assumption is that only a single profile may control an organisation, this sole person has the power to create teams, and dissolve the organisation. This role can be transferred to another profile at any point, transferring the ownership and authority to them.

Activities

The activities are listed below with their descriptions:



Architecture Design

This phase decides the architecture of the backend to fulfill the requirements of the project.

Resource Gathering

This phase gathers the resources specified in the architecture design, creating a preparation environment to start development.

Visuals Design

This phase is to create user interface wireframes to decide on views that need to be developed.

Backend Development

This phase is to develop the backend based on the architecture using the resources gathered. The backend should be ready for the testing phase to utilise.

Backend Testing

This phase is to test the backend functionality and see if all the features are met through APIs.

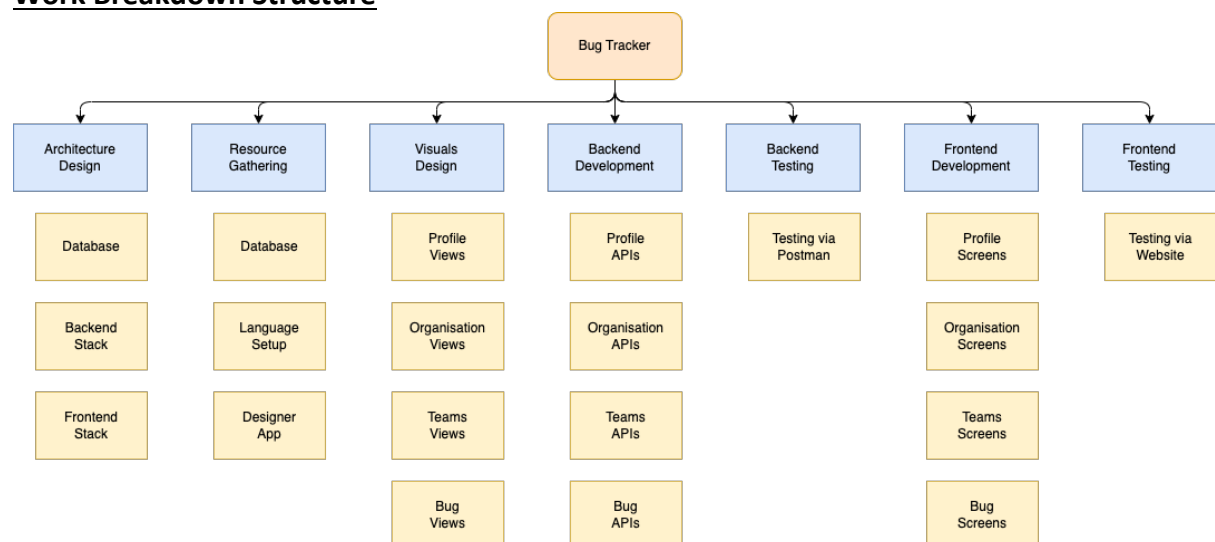
Frontend Development

This phase will start frontend development using the visuals designed earlier to create a website.

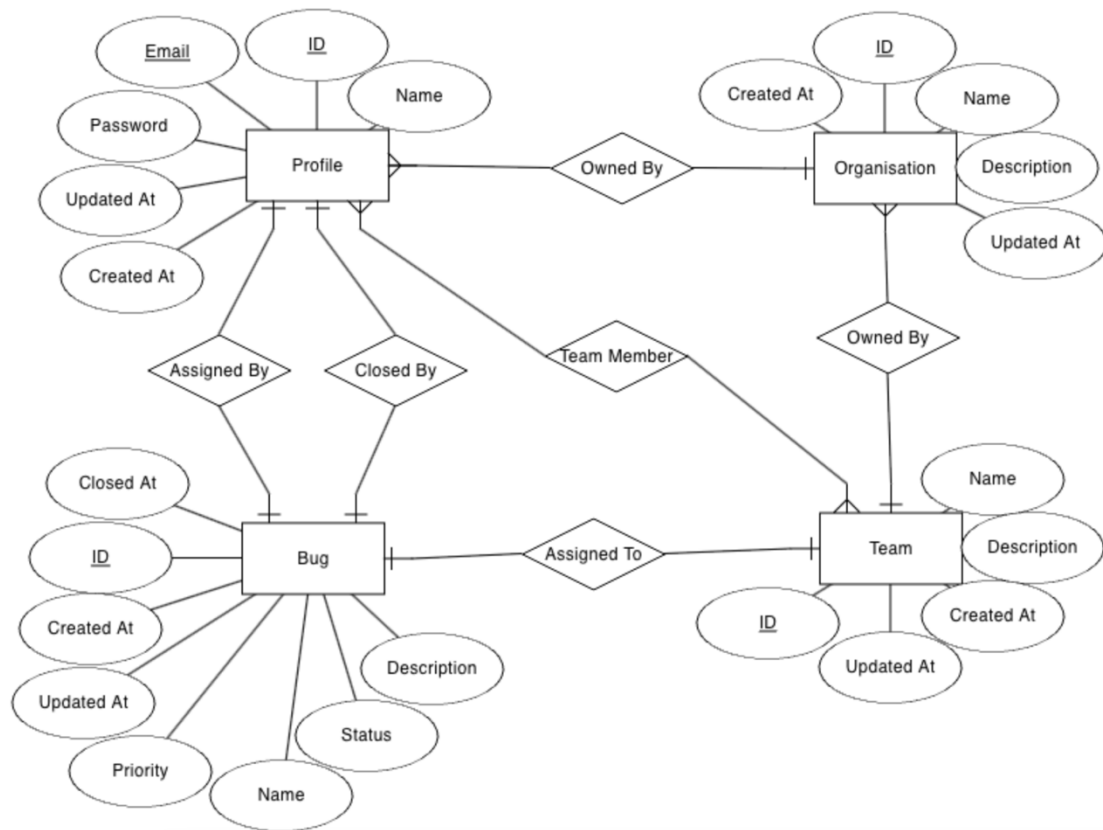
Frontend Testing

This phase tests the functionality and view of the frontend and concludes the development of the application.

Work Breakdown Structure



Entity Relation Diagram



Relational Schema

```
CREATE TABLE IF NOT EXISTS profile(  
  id UUID PRIMARY KEY,  
  name TEXT NOT NULL,  
  email TEXT NOT NULL,  
  password TEXT NOT NULL,  
  verified BOOLEAN NOT NULL DEFAULT false,  
  createdAt TIMESTAMPTZ NOT NULL,  
  updatedAt TIMESTAMPTZ NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS organisation(  
  id UUID PRIMARY KEY,  
  name TEXT NOT NULL,  
  description TEXT NOT NULL,  
  owner UUID NOT NULL,  
  createdAt TIMESTAMPTZ NOT NULL,  
  updatedAt TIMESTAMPTZ NOT NULL  
);  
  
ALTER TABLE organisation  
ADD FOREIGN KEY (owner) REFERENCES profile(id);  
  
CREATE TABLE IF NOT EXISTS team(  
  id UUID PRIMARY KEY,  
  name TEXT NOT NULL,
```

```

        description TEXT NOT NULL,
        organisation UUID NOT NULL,
        createdAt TIMESTAMPTZ NOT NULL,
        updatedAt TIMESTAMPTZ NOT NULL
    );
    ALTER TABLE team
    ADD FOREIGN KEY (organisation) REFERENCES organisation(id);

    CREATE TABLE IF NOT EXISTS teamMember(
        team UUID NOT NULL,
        profile UUID NOT NULL,
        createdAt TIMESTAMPTZ NOT NULL,
        updatedAt TIMESTAMPTZ NOT NULL,
        PRIMARY KEY (team, profile)
    );
    ALTER TABLE teamMember
    ADD FOREIGN KEY (team) REFERENCES team(id);
    ALTER TABLE teamMember
    ADD FOREIGN KEY (profile) REFERENCES profile(id);

    CREATE TABLE IF NOT EXISTS bug(
        id UUID PRIMARY KEY,
        name TEXT NOT NULL,
        description TEXT NOT NULL,
        status TEXT NOT NULL,
        priority TEXT NOT NULL,
        assignedTo UUID NOT NULL,
        assignedBy UUID NOT NULL,
        closedBy UUID,
        createdAt TIMESTAMP NOT NULL,
        updatedAt TIMESTAMP NOT NULL,
        closedAt TIMESTAMP
    );
    ALTER TABLE bug
    ADD FOREIGN KEY (assignedTo) REFERENCES team(id);
    ALTER TABLE bug
    ADD FOREIGN KEY (assignedBy) REFERENCES profile(id);
    ALTER TABLE bug
    ADD FOREIGN KEY (closedBy) REFERENCES profile(id);

```