

21BDS0340

Abhinav Dinesh Srivatsa

Software Engineering Lab

Bug Tracker SRS Document

Purpose

To create a bug tracker application for organisations and teams to keep track of bugs and manage collaboration

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.

User Classes

Class	Description
Organisation Owner	Has permissions to create and assign members to teams in their organisation. They also gain a view to manage their organisation. They also have every permission a team member has in any team.
Team Member	Has permissions to create bugs and maintain them.

Operating Environment

The expected operating environment for this application is in any modern browser versions. This application is not meant to be optimized for mobile view.

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.

Dependencies

This project has no dependencies, all development and requirements is independent and new

Business Rules and Functional Requirements

Profile

Attribute	Constraints	Changeability by user interface
ID	Unique, UUID v7, mandatory	No
Name	Mandatory, lower-case letters and numbers only, 20 characters max	Yes
Email	Unique, mandatory	Yes, but requires verification
Password	Mandatory, 1 lower, 1 upper, 1 number, 1 special, between 8-20 characters	Yes, but requires confirmation
Created At	Mandatory	No
Modified At	Mandatory	No

1. A profile cannot be deleted
2. A profile can be modified at any time
3. Email verification needs to be done before a profile can access anything in the bug tracker application
 - a. Email verification can be done by sending an email to the profiles email address and verifying a link
4. Each profile may have one organisation

Organisation

Attribute	Constraints	Changeability by user interface
ID	Unique, UUID v7, mandatory	No
Name	Mandatory, alphabets, numbers, spaces and any of the following special characters: !@#\$%^&*(),_.,?, 20 characters max	Yes, only by owner
Description	Unique, alphabets, numbers, spaces and any of the following special characters: !@#\$%^&*(),_.,?, mandatory	Yes, only by owner
Created At	Mandatory	No
Modified At	Mandatory	No

1. An organisation must be created by a profile
2. An organisation cannot be deleted, but can be transferred to another profile
3. An organisations name and description can be updated at any time by its owner
4. Each organisation can have a maximum of 10 teams

Team

Attribute	Constraints	Changeability by user interface
ID	Unique, UUID v7, mandatory	No
Name	Mandatory, alphabets, numbers, spaces and any of the following	Yes, only by owner

	special characters: !@#%^&*(),_.,?, 20 characters max	
Description	Unique, alphabets, numbers, spaces and any of the following special characters: !@#%^&*(),_.,?, mandatory	Yes, only by owner
Created At	Mandatory	No
Modified At	Mandatory	No

1. A team must be created by an organisations owner
2. A team cannot be deleted
3. An teams name and description can be updated at any time by its organisations owner
4. Multiple teams can have the same profile

Bug

Attribute	Constraints	Changeability by user interface
ID	Unique, UUID v7, mandatory	No
Name	Mandatory, alphabets, numbers, spaces and any of the following special characters: !@#%^&*(),_.,?, 20 characters max	Yes, only by owner team
Description	Unique, alphabets, numbers, spaces and any of the following special characters: !@#%^&*(),_.,?, mandatory	Yes, only by owner team
Status	Mandatory, one of: open, closed, cancelled	Yes, only by owner team and assigned to team
Priority	Mandatory, one of: 5, 4, 3, 2, 1	Yes, only by owner team
Assigned To	Mandatory team	Yes, only by owner team and assigned to team
Assigned By	Mandatory profile	No
Closed By	Mandatory profile	No
Created At	Mandatory	No
Modified At	Mandatory	No
Closed At		No

1. A bug must be created by a profile part of a team in an organisation
2. All members in the assigned to team must be emailed of the bug

Use Cases

Profile Creation

A new user creates a profile

Actor Actions	System Response
Actor signs up using email, name and password	An auto generated email is sent to the email address
Actor clicks on the link to verify themselves	The actor is now verified, and can now log in
Actor goes to the login page and enters their email and password	The actor is now signed in, and can access services using an access token handled by the application

Organisation and Team Creation

A profile creates a new organisation

Actor Actions	System Response
Actor clicks on a button to create a new organisation	The actor is redirected to the appropriate organisation creation page
Actor fills in the name and description	The actor now owns an organisation if none were owned before
Actor clicks on the button to create a new team	The actor is redirected to the appropriate teams creation page
Actor clicks on a button to create a new team	The organisation now owns a team

Team Profile Addition

A profile adds another profile to their organisations team

Actor Actions	System Response
Actor clicks on a button to add a team member	The actor is redirected to the appropriate team member addition page
Actor fills in the email	An email is sent to the profile to be added to the team
Actor 2 clicks on the link	The actor 2 is now part of actor 1's organisation and part of the particular team

Bug Creation

A bug is created by a team member

Actor Actions	System Response
Actor clicks on a button to create a bug	The actor is redirected to the appropriate bug creation page
Actor fills in the details and the assigned to field is another team	An email is sent to all the members in the assigned to team and a bug is created

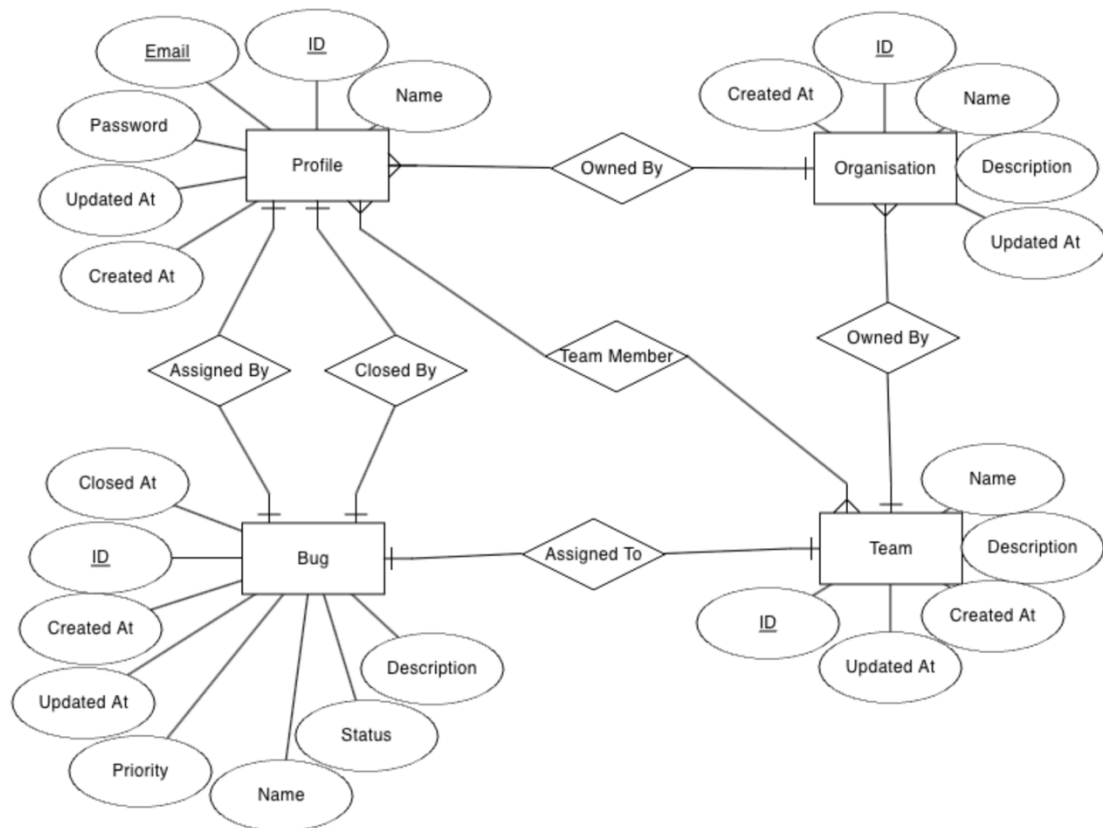
Bug Resolution

A bug is resolved by an assigned team member

Actor Actions	System Response
Actor clicks on a bug to resolve it	The bug is now marked as resolved

Data Requirements

ER Diagram



Data Dictionary

Attribute	Entity	Required	Type	Length	Comments
ID	Profile	Yes	UUID	36	
Name	Profile	Yes	Text	20	Lower case and numbers only
Email	Profile	Yes	Text		Email type
Password	Profile	Yes	Text		Hashed password
Created At	Profile	Yes	Time		Time when profile was created
Modified At	Profile	Yes	Time		Time when profile was last modified
ID	Organisation	Yes	UUID	36	
Name	Organisation	Yes	Text	20	Alphabets, numbers, spaces and any of the following special characters: !@#\$%^&*(),_ . ?

Description	Organisation	Yes	Text	200	Alphabets, numbers, spaces and any of the following special characters: !@#\$\$%^&*(),_.
Owner	Organisation	Yes	UUID	36	Reference: Profile
Created At	Organisation	Yes	Time		
Modified At	Organisation	Yes	Time		
ID	Team	Yes	UUID	36	
Name	Team	Yes	Text	20	Alphabets, numbers, spaces and any of the following special characters: !@#\$\$%^&*(),_.
Description	Team	Yes	Text	100	alphabets, numbers, spaces and any of the following special characters: !@#\$\$%^&*(),_.
Organisation	Team	Yes	UUID	36	Reference: Organisation
Created At	Team	Yes	Time		
Modified At	Team	Yes	Time		
Team	Team Member	Yes	UUID	36	Reference: Team
Profile	Team Member	Yes	UUID	36	Reference: Profile
Created At	Team Member	Yes	Time		
Modified At	Team Member	Yes	Time		
Name	Bug	Yes	Text	20	Alphabets, numbers, spaces and any of the following special characters: !@#\$\$%^&*(),_.
Description	Bug	Yes	Text	100	Alphabets, numbers, spaces and any of the following special characters: !@#\$\$%^&*(),_.
Status	Bug	Yes	Enum		One of: open, closed, cancelled
Priority	Bug	Yes	Enum		One of: 5, 4, 3, 2, 1
Assigned To	Bug	Yes	UUID		Reference: Team
Assigned By	Bug	Yes	UUID		Reference: Profile
Closed By	Bug	No	UUID		Reference: Profile
Created At	Bug	Yes	Time		
Modified At	Bug	Yes	Time		
Closed At	Bug	No	Time		

Data Maintenance

1. The application must retain bugs from all time
2. The application must not lose any data at all, no data deletions will be performed
3. Data warehousing can happen after a 5 year period of bugs

External Requirements

No external requirements are present for this project. This project has no dependencies to delay or wait for.

Non-Functional Requirements

1. The application and the backend must be online 99.99% of the time excluding maintenance breaks
2. The backend system must be containerised to run on any machine with containerd or Docker
3. The system must only have updates in off hours to maintain availability
4. The system must be deployed using the blue/green method to prevent crashes and to have safe rollbacks
5. The system must have automatic scaling based on user and network demand
6. The system must always show timestamps in the local time zone

Appendix

Task List

	🔍	Name	Duration	Start	Finish	Predecessors	Resource Names
1		Architecture Design	2 days	5/2/24 8:00 AM	6/2/24 5:00 PM		
2		Database Selection	1 day?	7/2/24 8:00 AM	7/2/24 5:00 PM	1	Abhinav Srivatsa
3		Backend Stack Selection	1 day?	8/2/24 8:00 AM	8/2/24 5:00 PM	2	Abhinav Srivatsa
4		Frontend Stack Selection	1 day?	9/2/24 8:00 AM	9/2/24 5:00 PM	3	Abhinav Srivatsa
5		Resource Gathering	1 day?	12/2/24 8:00 AM	12/2/24 5:00 PM	4	Abhinav Srivatsa
6		Database Setup	2 days	13/2/24 8:00 AM	14/2/24 5:00 PM	5	Abhinav Srivatsa
7		App Design Overview	2 days	15/2/24 8:00 AM	16/2/24 5:00 PM	6	Abhinav Srivatsa
8		Visuals Design	1 day?	19/2/24 8:00 AM	19/2/24 5:00 PM	7	Abhinav Srivatsa
9		Profile Views	1 day?	20/2/24 8:00 AM	20/2/24 5:00 PM	8	Abhinav Srivatsa
10		Organisation Views	1 day?	21/2/24 8:00 AM	21/2/24 5:00 PM	9	Abhinav Srivatsa
11		Teams Views	1 day?	22/2/24 8:00 AM	22/2/24 5:00 PM	10	Abhinav Srivatsa
12		Bug Views	1 day?	23/2/24 8:00 AM	23/2/24 5:00 PM	11	Abhinav Srivatsa
13		Backend Development	5 days	26/2/24 8:00 AM	1/3/24 5:00 PM	12	Abhinav Srivatsa
14		API Designs	5 days	4/3/24 8:00 AM	8/3/24 5:00 PM	13	Abhinav Srivatsa
15		Backend Testing via Postman	5 days	11/3/24 8:00 AM	15/3/24 5:00 PM	14	Abhinav Srivatsa
16		Frontend Development	5 days	18/3/24 8:00 AM	22/3/24 5:00 PM	15	Abhinav Srivatsa
17		View Screens	10 days	25/3/24 8:00 AM	5/4/24 5:00 PM	16	Abhinav Srivatsa
18		Frontend Tesing	5 days	8/4/24 8:00 AM	12/4/24 5:00 PM	17	Abhinav Srivatsa

Gantt Chart

