Issue Tracking System

Senior Project

By

Bitsu Mamo - 115768

August 2023

Issue Tracking System

Vision

Introduction

The proposed Issue Tracking System(ITS) will function as a ticketing system, providing a structured approach for managing and resolving issues within a work environment. Upon integration into a system, it will offer four distinct roles: Creator, Resolver, Reviewer, and Administrator. They each have their own levels of access.

Roles

1. Creator

   Creators are users who can initiate an issue process byu creating new issues. Any user with an issue to report can assume the role of a creator. They will be able to submit details about the problem they encountered. The description can contain screenshots and information that would be useful to the issue.

2. Reviewer

   Reviers play a crucial role in the ITS as they are responsible for reviewing the reported issue. When an issue is submitted, the reviewer examines it to determine the validity and relevance. If the issue is valid, the reviewer proceeds to assign it a specific type and appropriate resolver to handle the task. However, if the issue is found to be invalid or its status is marked as closed.

3. Resolver

   The resolver is the user designated to work on the identified issues. Once a reviewer assigns an issue to a resolver, the resolver works toward finding a solution. Upo In resolving the issue the revolver updates its status to Resolved

4. Administrator

   The admin has the highest level of access. They have full control over the system and can perform all functions, including managing users, configuring the system, and accessing the dashboard.

General Problem Statement

Our client faces significant challenges in efficiently managing and tracking issues that arise within their workplace. The current manual system lacks essential features, leading to delays in issue resolution and hindering project progression. The absence of a collaborative system creates a communication gap among different teams.

One of the primary concerns is the inability to gain an overall view of the progress made on reported issues due to the lack of analytics. The client seeks real-time data analytics and reporting inorder to make informed decisions.

Data security is another critical point of concern for our client as they handle sensitive information. The current system's lack of access control raises security concerns, allowing unauthorized access to certain functionality and data.

To overcome these challenges, our client is in need of an Issue Tracking System(ITS) that addresses these problems of their current system. The ideal solution should offer an intuitive and user-friendly interface for issue reporting, reviewing, and resolution as well as security.

Furthermore, the proposed system must provide an extensive dashboard with real-time data analytics, empowering the client with valuable insight into issues. Proper access control mechanics should be implemented to assign specific permissions to users based on their roles, ensuing data security.

In conclusion, our client seeks an advanced Issue Tracking System that enhances issue management, collaboration, and data analytics within their workplace. By addressing these challenges the proposed ITS aims to facilitate efficient issue resolution, improve productivity, and support the client in achieving their business objectives.

Problems

1. ### Inefficient Workflow

   Our client's company lacks the ability of proper issue reporting, evaluation, assignment, and resolution. This inefficiency leads to delays in issue resolution, hindering the client ability to provide service.

2. ### Lack of Collaboration

   Once an issue has been found there is an issue of finding the correct reviewer and resolver. This leads to communication gaps and misunderstandings.

3. Data Gathering

   Our client needs a way to see aggregate data so that they can efficiently decide where most issues are caused so that they can find the cause of a certain issue.

4. Security and Access Control

   Our client needs proper access control over the functionality of each user. As of now anyone can create, resolve and review issues. The client needs specific permission for each user, so that they can limit the access to data.

To address these challenges, our proposed ITS aims to develop an efficient, user-friendly, and collaborative platform that caters to the diverse needs of our client. By providing a well detailed solution to empower our client with the necessary tools and insights.

Technical Problem System

Our task is to design and implement the ITS that addresses the specific need of our client. The ITS will provide a simple and intuitive user interface, enabling users to log in and access functionality based on their assigned roles.

The ITS will maintain a user database, allowing users to have multiple roles, each with their own privileges. The access levels will be carefully defined to restrict or enable various functionalities based on users roles.

The ITS will store and manage all issues created within the organization. Each issue will contain a detailed description, along with attachments related to the specific problem. The system will allow the assignment of a resolver, type, and status to each issue, which will be carried out by a designated reviewer. Collaborative efforts will be encouraged through comments, allowing users to add text and attachments to the issues for seamless teamwork.

The ITS will support efficient retrieval through filtering and search functionality, enabling users to find users to find issues quickly based on relevant criteria.

Administrators have complete access to all functionality and data within the ITS.

Stakeholder Description

Stakeholders

1. Creators

   Primary users of the ITS. They are the people that create issue tickets so that they can be resolved. They provide essential details, such as descriptions and attachments.

2. Reviewers

   They are responsible for assessing and validation issues reported by creators. They play a role in categorizing and assigning issues.

3. Resolvers

   They are designated users responsible for addressing and resolving the reported issues. They possess the ability of understanding the issue. They collaborate with creators and reviewers to solve issues.

4. Administrators

   They have superuser access. They can manage the system's configurations, users, and permissions. They play a crucial role in maintaining security and integrity.They monitor the functioning of the ITS.

5. Project Managers

   They are the people that oversee the development project. They come directly for the client and have vested interest in the project. They will actively collaborate with the development team to find a correct solution.

The successful implementation and adoption of the Issue Tracking System depends on the effective communication, collaboration, and engagement of these stakeholders. The system will be designed to cater to the needs and responsibilities of each stakeholder.

Needs and Features

Needs

1. Efficient Issue Reporting
2. Collaboration and Communication
3. Analytics
4. Data Security and Access Control

Features

1. ### User Roles and Access Levels

   Different roles for each user. Creator, Reviewer, Resolver, Admin. A set of roles can be assigned to users if they are granted access as multiple roles. This issues popper control over the system functionalities.

2. ### Issue Assignment

   The system should enable reviewers to assign issues to appropriate resolvers.

3. ### Collaboration

   The system should have a built in communication tool like issue thread comments.

4. ### Customizable Workflow

   Custom issue types and status which allows the client to align the system with their needs.

5. ### Analytics

   A dashboard with charts and grapes, to show metrics to the stakeholders. This give valuable insight into the organization

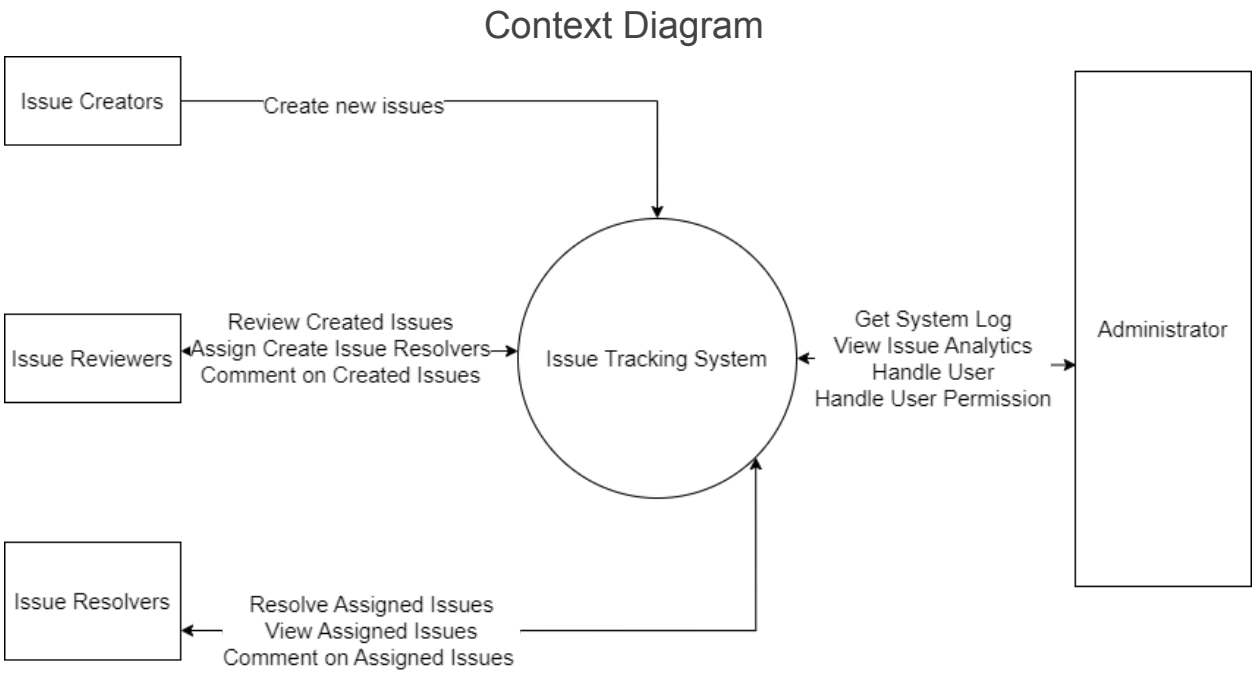
6. ### User Authentication

   The user of JWT token to login securely into the system.

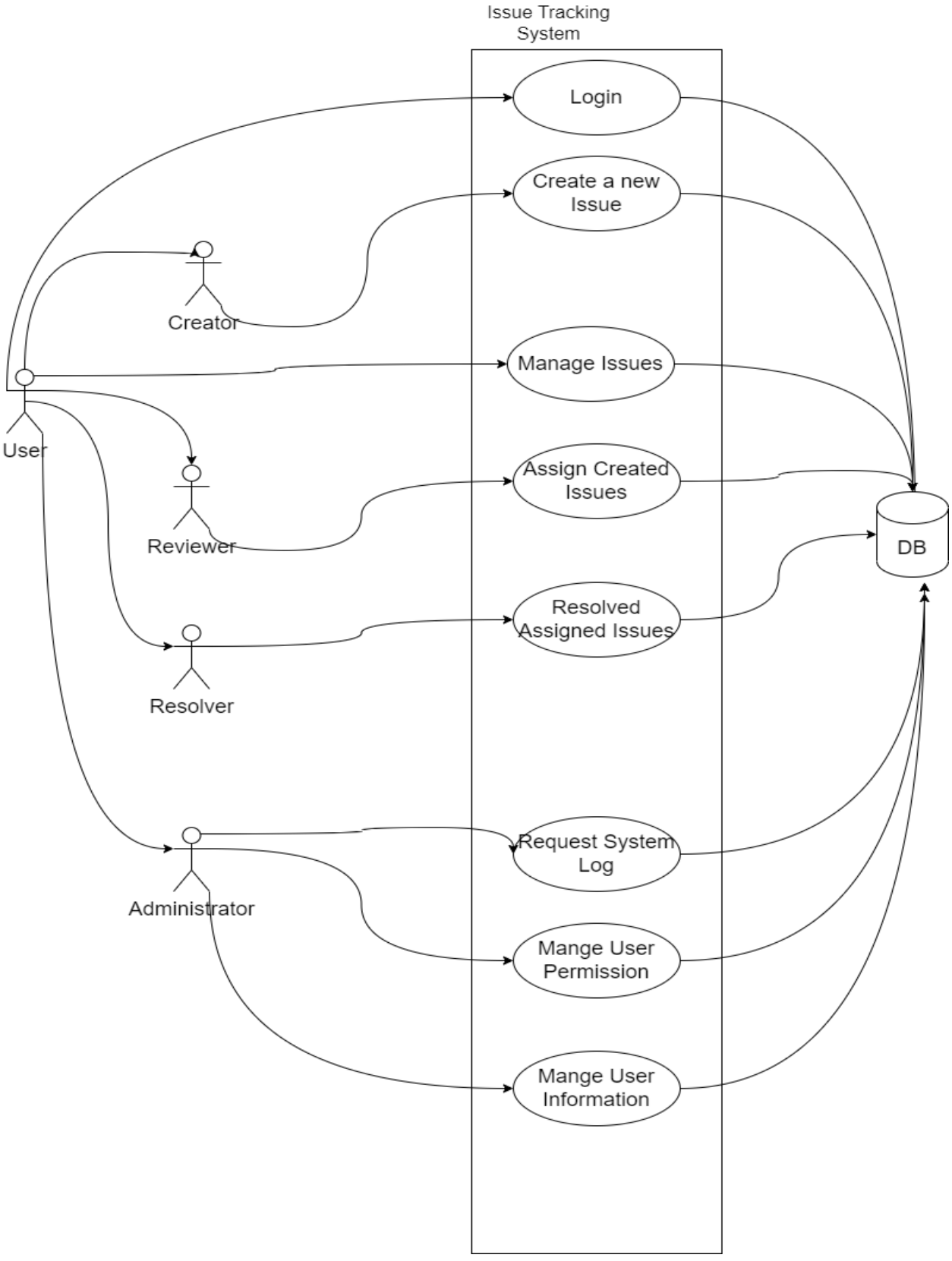
7. ### Search and Filtering

   Advanced search and filtering to help users find specific issues based on different criteria.

Requirements



Use Case Diagram



Main User Stories

As a user, I should be able to be authenticated and authorized, so that I can use the system properly.

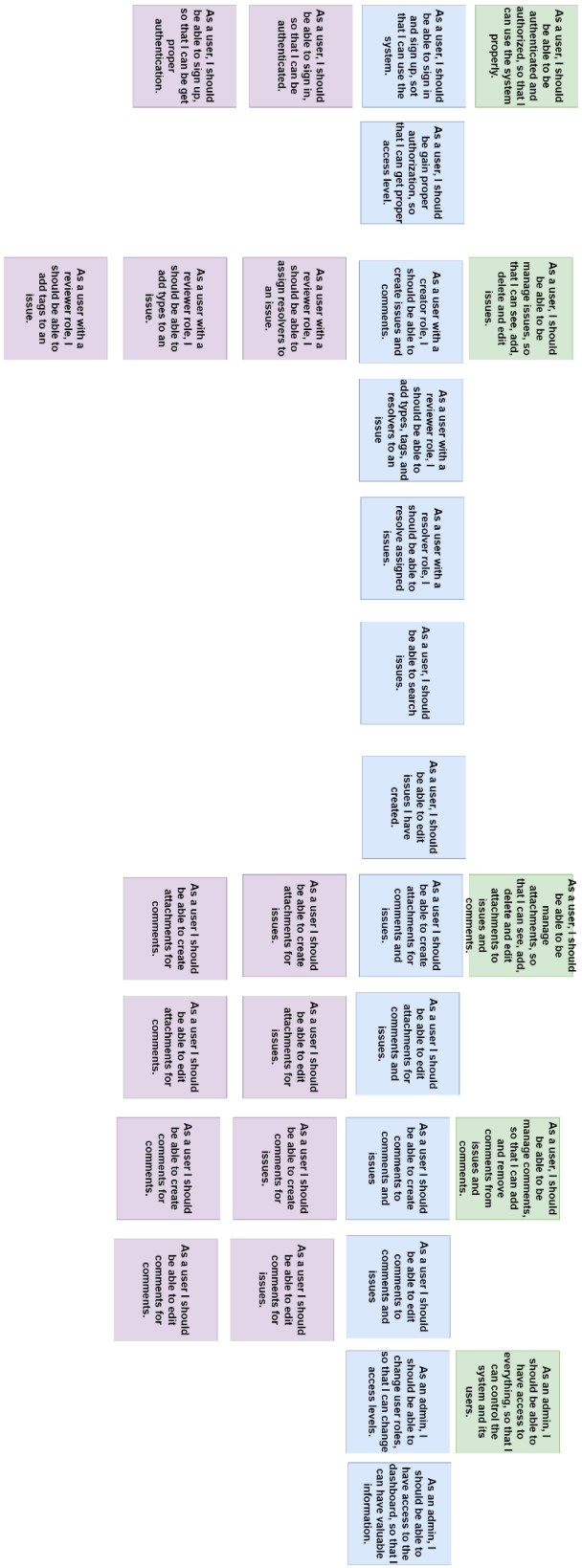
As a user, I should be able to be manage issues, so that I can see, add, delete and edit issues.

As a user, I should be able to be manage attachments, so that I can see, add, delete and edit attachments to issues and comments.

As a user, I should be able to be manage comments, so that I can add and remove comments from issues and comments.

As an admin, I should be able to have access to everything, so that I can control the system and its users.

User Story Map

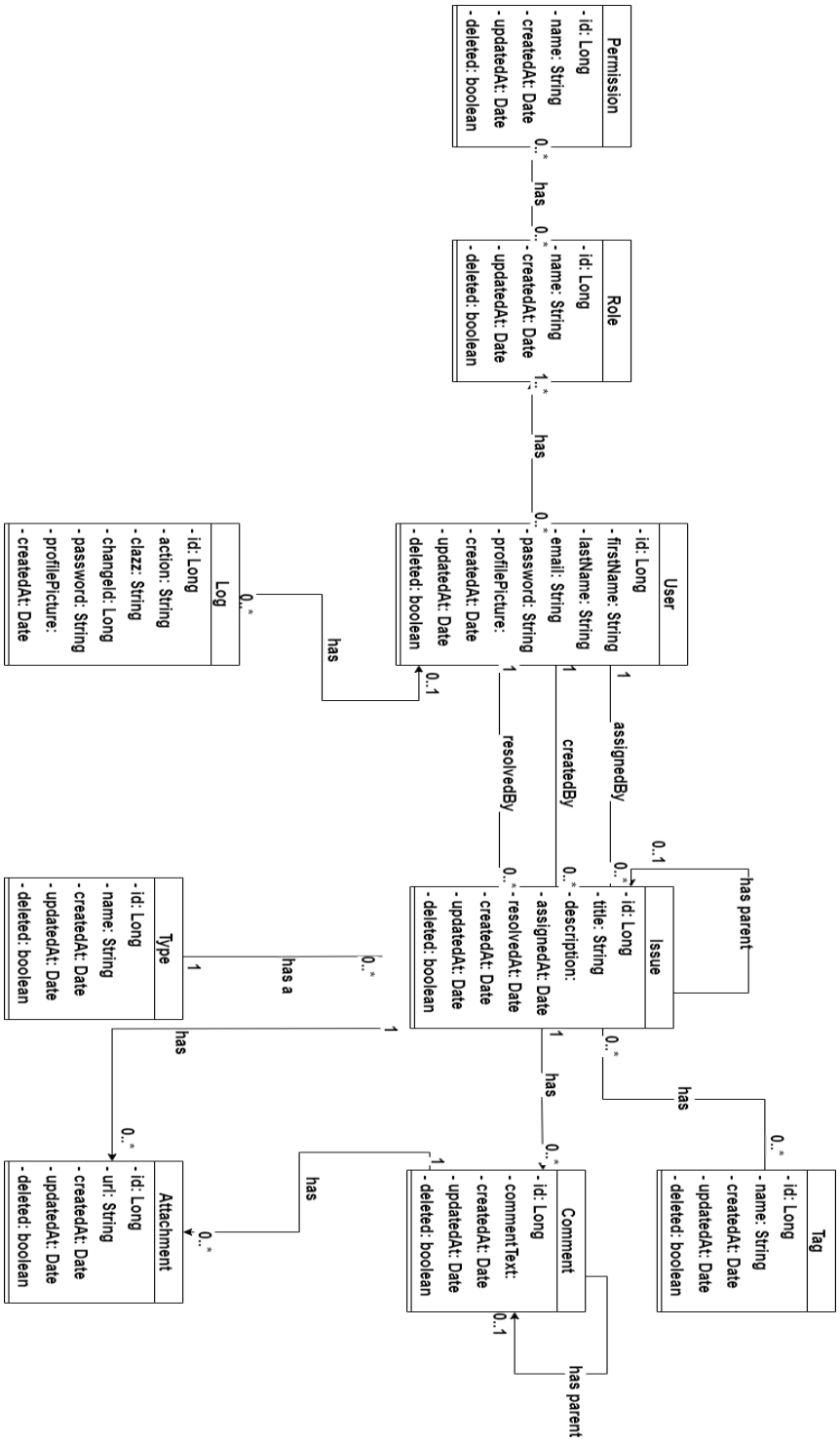


Acceptance Criteria

1. As a user, I should be able to be authenticated and authorized, so that I can use the system properly.
    a. I should be able to login using the correct credentials
    b. I should not be able to login without correct credentials.
    c. I should be able to sign up with a unique email address as a creator.
    d. I should not be able to sign up with a repeated email address.
2. As a user, I should be able to manage issues, so that I can see, add, edit, and delta issues.
    a. I should be able to create issues if I have a creator role only.
    b. I should be able to review an issue if I have a reviewer role only.
    c. I should be able to resolve an issue if I have a resolver role only.
3. As a user, I should be able to manage attachments.
    a. I can create attachments for issues.
    b. I can add multiple attachments for issues.
    c. I can create attachments for comments.
    d. Attachments are stored as urls.
4. As a user, I should be able to manage comments
    a. I can create comments for issues.
    b. I can create comments for comments
5. As an admin, I should be able to have access to everything, so that I can control the system and its users
    a. I should be able to get dashboard information so that I can get analytics
    b. I should be able to update user roles.

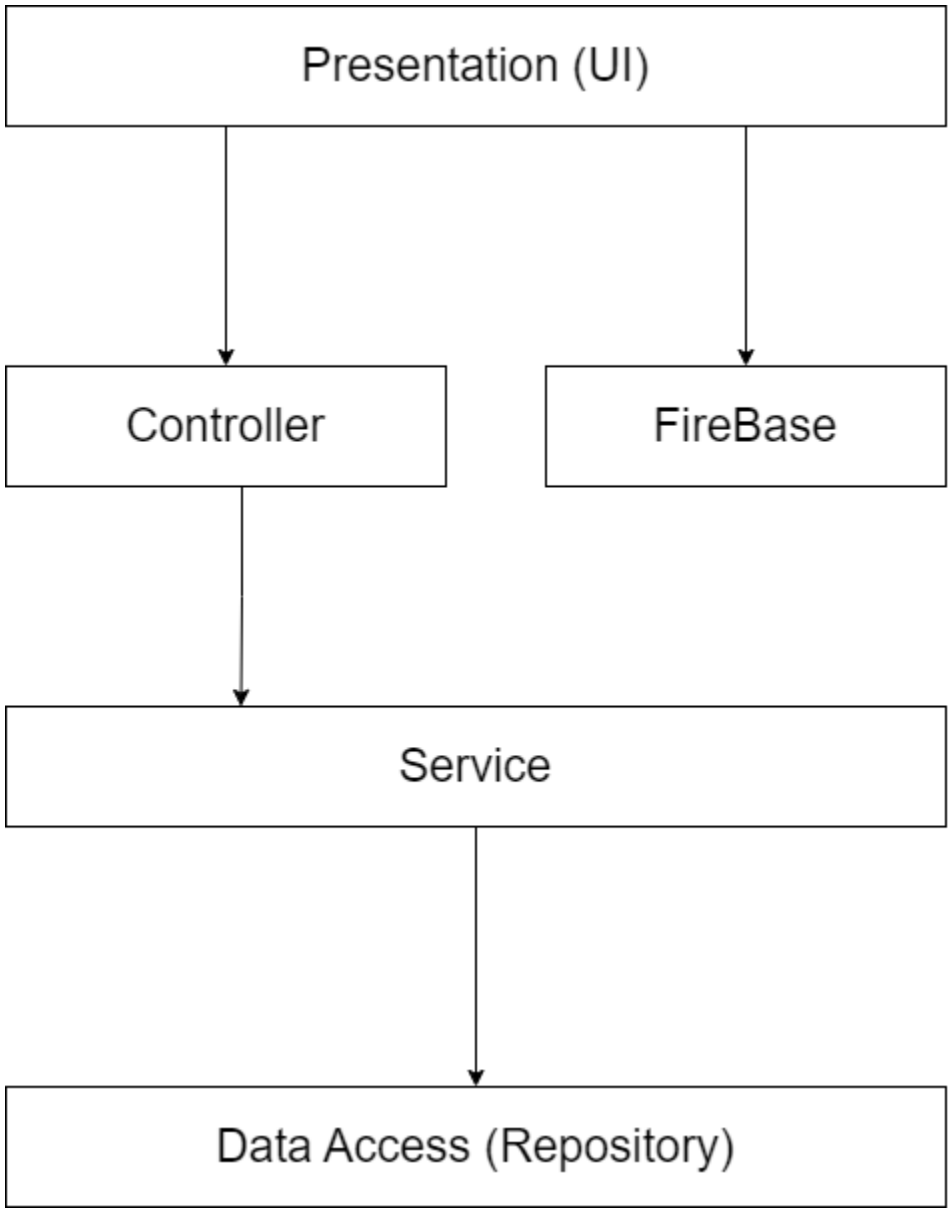
Analysis

Class Diagram

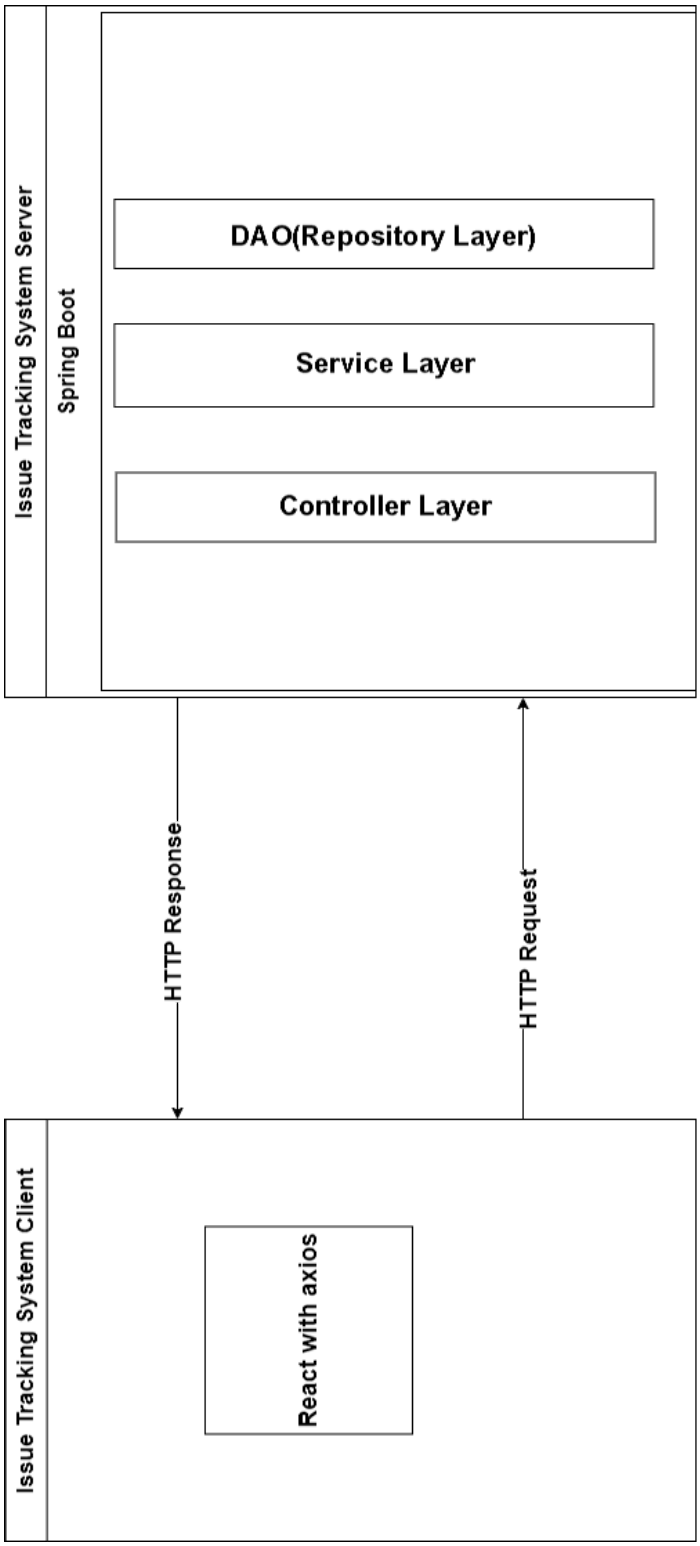


Architecture

Layer Diagram



Architecture Diagram



Overview of Design

Backend

The backend of the system will be developed using the Spring Boot framework. Spring Boot provides a robust and efficient environment for building RESTful APIs and managing the application's business logic. The Spring Boot backend will offer the necessary functionalities to support the frontend application built with React.

1. API Endpoints

   The backend will define various API endpoints using Spring's @RestController annotation. These endpoints will handle incoming HTTP requests from the frontend and interact with the relevant components of the application.

2. Service Layer

   The Spring Boot backend will incorporate a service layer with @Service annotated classes. This layer will encapsulate the core business logic of the application. It will interact with the data access layer and manage operations such as issue management, user authentication, and data processing.

3. Data Access

   Spring Data JPA will be utilized to interact with the database. Entities representing key components like users, issues, and comments will be defined. Spring repositories will provide the necessary methods to perform CRUD (Create, Read, Update, Delete) operations on these entities.

4. Security Configuration

   If required, the Spring Security framework will be integrated to manage authentication and authorization. The backend will handle user authentication, role-based access control, and secure API endpoints
   .

5. Documentation

   API documentation will be generated using tools like Swagger. This documentation will provide clear insights into the available endpoints, their functionalities, request parameters, and response formats. This will facilitate seamless communication between the frontend and backend development teams.

6. Database

   MySQL gives us robust functionality for our project. It will allow us to create different databases for testing and production.

FrontEnd

The frontend of the system will be developed using the React library along with Axios for making HTTP requests to the Spring Boot backend.

1. Component Structure

   The frontend application will be structured using modular components. Each component will focus on a specific part of the user interface, contributing to a more maintainable and organized codebase.

2. Routing

   React Router will be used to manage the different views of the application. Routes will be defined to handle various user interactions and display the appropriate components.

3. State Management

   React's state management capabilities will be leveraged for handling local component state. For more complex state management scenarios, a library like Redux may be considered.

4. API Integration

   Axios, a widely used JavaScript library, will be employed for making HTTP requests to the Spring Boot backend. Axios offers a convenient and streamlined way to interact with APIs, supporting GET, POST, PUT, and DELETE requests.

5. User Interface

   The user interface will be designed using React's component-based architecture. Libraries such as Material-UI or Ant Design may be used to create visually appealing and responsive UI components.

Connecting Knowledge With Transcendental Consciousness

In the design and development of the Integrated Issue Tracking System (ITS), the principles of Transcendental Consciousness will be integrated into both the backend, powered by Spring Boot, and the frontend, built with React and Axios. This innovative approach will create a more harmonious and mindful user experience throughout the entire system.

1. Mindful API Endpoint

   The API endpoints, designed with simplicity and clarity, will reflect the mindfulness inherent in Transcendental Consciousness. Each endpoint will carry intentionality, fostering an understanding of the interconnected nature of user interactions.

2. Conscious Data Processing

   The backend's service layer will extend beyond conventional business logic to embrace a conscious approach to data processing. The logic will align with the principles of mindfulness, responding to user actions with awareness and purpose.

3. Interconnected Data Flow

   Spring Data JPA's data access layer will be designed to embody the interconnectedness emphasized by Transcendental Consciousness. Entities and repositories will facilitate a harmonious flow of information, reflecting the interwoven nature of issues, users, and the application.

Science of Creative Intelligence

The application of Transcendental Consciousness as a foundational paradigm in the holistic process of developing full stack software reveals intriguing parallels between the unfolding layers of software architecture and the progressive layers of conscious awareness. This is a symbiotic relationship between the transcendental essence and the software development process.

At the core of the metaphorical software "universe" lies the database, analogous to the pure consciousness in Transcendental Consciousness. This database embodies the origin of all data and information, mirroring the fundamental essence of pure consciousness as the sap of all existence.

Much like the transcending steps in consciousness, the full stack software development traverses stages in both the frontend and backend components. These components are analogous to distinct layers of consciousness, each with its significance in the overall system.

Authentication serves as the threshold to access these deeper layers in both contexts. In software development, proper authentication is essential before venturing further into the system's functionalities. Similarly, in the realms of consciousness, a certain level of awareness is prerequisite for deeper transcendental experiences.

In conclusion, we can observe that Transcendental consciousness is the root of everything. Even in developing a full stack software. As there are steps of transcending there are steps and layers to the frontend and backend of every application. The root is the database such that the root is pure consciousness(the “database”) of everything. Before we can reach suck depth we first have to be authenticated and take the correct route passing through different levels.