SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**Issue Tracking System**

**Prepared by:-**

*AJAY PRANAO M*

*BIBIN SAMUEL NAYAGAM S*

*HARI KRISHNAN T*

# Introduction

## Purpose

## Document Conventions

The main objective of this document is to illustrate the requirements of the project Issue Tracking system. The document gives a detailed description of the both functional and non-functional requirements proposed by the client. An issue-tracking software is a tool that makes it possible to track and resolve issues end-to-end. It provides a centralized platform from which all issues are recorded. This project describes the hardware and software interface requirements using ER diagrams and UML diagrams. The entire document should be justified.

* + - Convention for the Main title

Font face: Times New Roman Font Style: Bold

Font Size: 14

* + - Convention for Subtitle

Font face: Times New Roman Font Style: Bold

Font Size: 12

* + - Convention for body

Font face: Times New Roman Font Size: 12

## Scope of Development Project

The bug tracking system is the perfect or unique solution to track the bugs of a solution, product, or application. The bug tracking system allows a single or set of developers to continue track of not finished bugs in their product successfully. Bug tracking systems can also be called defect tracking systems (DTS). For good performance the bug tracking system can increase a lot, the accountability and productivity of single employees by giving positive feedback and backing up the workflow. The bug tracking software allows or group of testers or individual testers to keep the path of unfixed bugs in their software successfully. The Bug tracking software can track bugs, handle code changes, share information with teammates, submit and review connects, and control standard assurance.

## Definitions, Acronyms and Abbreviations

JAVA -> platform independence SQL-> Structured query Language ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment SRS-> Software Requirement Specification

## References

* + - Books

 Software Requirements and Specifications: by Jack Wade

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements

<https://github.com/amira921?tab=repositories> -Amira Taha

# Overall Descriptions

## Product Perspective

Use Case Diagram of Issue Tracking System

## Product Function

Entity Relationship Diagram of Issue Tracking System

There are three types of users of the system.

1. Administrator The following are the submodule in the administration module

● login

● Add Staff

● Add Project

● View customer’s bugs

● Assign bug to staff

● View bug case flow

● Send solution messages

2. Customer/end user

The following are the submodules in the customer/end-user module

● Register

● Login

● Send bugs

● View bug case flow

● View solution messages

3. Staff The following are the submodule in the staff user module

● Login

● View assigned bugs

● Send solution messages

● Assign bug to staff

● Share bugs

● View bug case flow

## Operating Environment

## Assumptions and Dependencies

## Following are the assumptions and dependencies that are related to this e Bug Tracker System:

## ● This project is a stand-alone project so it will not affect the system where it will be embedded.

## ● This project is a web-based project.

## ● This system will not depend on any other module. It will be web-based so everyone will independently contact it.

## ● It will not affect the environment at all.

## ● Roles and tasks are predefined.

## Requirement

## Data Requirement

## Each part of the user interface intends to be as user-friendly as possible. The fonts and buttons used will be intended to be very fast and easy to load on web pages. The pages will be kept light in space so that it won’t take a long time for the page to load. The starting page will be the home page of the system. All the operations in the user(customer/staff/admin) main page can be used if the user (customer/staff/admin) is logged in to his/her account. Some pages display the details accordingly.

# External Interface Requirement

## GUI

## Intuitive Navigation:Design a clear and intuitive navigation system that allows users to easily move between different sections of the issue-tracking system.Use a logical hierarchy for menu items and ensure that common actions are easily accessible.Dashboard:Include a customizable dashboard that provides a quick overview of key metrics and important information.Allow users to personalize their dashboard to display relevant data and widgets.Issue Creation and Editing:Create a user-friendly form for creating and editing issues.Include mandatory and optional fields, and provide helpful tooltips or inline guidance to assist users in filling out the form accurately.Search and Filters:Implement robust search functionality to allow users to quickly find specific issues.Provide advanced filtering options to help users narrow down their search based on various criteria (e.g., status, priority, assignee). Collaboration and Comments:Enable users to add comments to issues for collaboration.Include the ability to mention or tag specific users to notify them of relevant comments. Reports and Analytics:Include reporting tools to generate custom reports and analytics.Provide visual representations of data, such as charts and graphs, for better insights.Accessibility:Ensure that the GUI complies with accessibility standards to accommodate users with disabilities.Provide alternative text for images, support keyboard navigation, and maintain good contrast ratios.Feedback Mechanism:Implement a feedback mechanism, such as a suggestion box or user surveys, to gather input on the usability and effectiveness of the GUI.

# System Features

# An essential tool for software development teams is a bug-tracking system, which facilitates the discovery, reporting, prioritization, and resolution of problems with software applications. Features that improve teamwork and expedite the bug-tracking procedure should be included in the system. These are essential components of an issue-tracking system:

# Creation of Issues:

# Incorporate mandatory data like summary, description, priority, severity, and environment specifics in the form fields.

# Attachments: To offer further background information, let users attach logs, screenshots, or other pertinent items.

# Other Non-functional Requirements

## Performance Requirement

Logging: Implement logging mechanisms to track system events and errors.Monitoring Tools: Specify tools and metrics for monitoring system performance

## Security Requirement

Access Control: Define user roles, permissions, and access levels to ensure proper data protection.Data Encryption: Specify encryption requirements for data transmission and storage.Authentication and Authorization: Specify the mechanisms for user authentication and authorization.

## Requirement attributes

## Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data. This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither the admin nor the member should cross the rules and regulations.

## User Requirement

## User requirements for an issue-tracking system capture the needs and expectations of the system's end-users. Understanding these requirements is crucial for designing a system that aligns with user workflows and provides a positive user experience.

# Other Requirements

## Data and Category Requirement

There are different categories of users namely teaching staff, Librarian, Admin, students, etc. Depending upon the category of user the access rights are decided. It means if the user is an administrator then he can be able to modify the data, delete, append, etc. All other users except the Librarian only have the right to retrieve the database information. Similarly, there will be different categories of books available. According to the categories of books their relevant data should be displayed. The categories and the data related to each category should be coded in a particular format.

**User Authentication and Access Control**

**Intuitive User Interface**

**Issue Creation and Editing**

**Reports and Analytics**

## Appendix

An appendix for an issue tracking system documentation typically includes supplementary materials and information that can provide additional context, details, or resources for users, administrators, or developers.

## Glossary

The following is the list of conventions and acronyms used in this document and the project as well:

Issue:A task, bug, enhancement, or any work item that needs tracking and management within the system.Bug: An error, flaw, or unexpected behavior in the software that needs to be addressed and fixed.Task: A unit of work or activity that needs to be completed as part of a project or development cycle.Enhancement: A proposed improvement or addition to existing features or functionalities.Workflow:The sequence of steps or stages through which an issue progresses, from creation to resolution.Status: The current state or condition of an issue within the workflow (e.g., open, in progress, resolved).Priority: The level of importance or urgency assigned to an issue, determining its order in the queue for resolution.Severity: The impact of an issue on the software's functionality, ranging from minor to critical.Assignee: The person or team responsible for working on and resolving a specific issue.Reporter: The person who reported or identified the issue within the system.Watcher: A user who is monitoring or "watching" a particular issue to receive notifications about The status indicates that an issue has been resolved and is considered closed. Version: A specific release or iteration of the software, often associated with a set of features and bug fixes.Tag: A label or keyword associated with an issue for grouping or categorization purposes.API (Application Programming Interface): A set of rules and protocols that allows different software applications to communicate with each other.Plugin: An additional piece of software that extends or adds functionality to the core issue-tracking system.