SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**Issue Tracking System**

**Prepared by:-**

*Kalaivanan R*

*Aadavan A S*

*Aakash S*

# Introduction

## Purpose

The main objective of this document is to illustrate the requirements of the project Issue Tracking System. An issue tracking system is a crucial component of project management and software development that helps organizations efficiently manage and resolve problems, bugs, and tasks throughout the development lifecycle. It serves as a centralized hub for recording, monitoring, and addressing issues that may arise during the cour of a project. Whether it's software defects, feature requests, or general tasks, an issue tracking system provides a systematic approach to handling these issues, ensuring that they are documented, assigned, and resolved in a timely manner. This project software interface requirements using ER diagrams and UML diagrams.

## Document Conventions

* + - Entire document should be justified.
    - Convention for Main title

Font face: Times New Roman Font style: Bold

Font Size: 14

* + - Convention for Sub title

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

This project is dedicated to crafting an Issue Tracking System tailored for software development projects. Developers will be equipped with the ability to seamlessly report issues, monitor their progression, and collaborate on effective solutions. The scope involves defining access controls to ensure secure usage and establishing communication channels such as real-time chat and forums for efficient collaboration. Detailed requirements gathering will focus on issue reporting, categorization, and status tracking, providing a user-friendly and comprehensive system. Acknowledging the pivotal role of collaboration features, the system will integrate tools for real-time communication and knowledge sharing among developers. Key features encompass collaboration tools, customizable issue reporting forms, and a streamlined system for issue assignment and status tracking. The overarching objective is to enhance communication, streamline issue resolution, and provide a centralized platform for efficient collaboration within software development projects.

## 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

* + - Websites

 Jira,Github issues,Gitlab issues

Software Requirements (Github)

* + - Websites

[**http://www.slideshare.net/**](http://www.slideshare.net/)

[**http://ebookily.net/doc/srs-library-management-system**](http://ebookily.net/doc/srs-library-management-system)

# Overall Descriptions

## Product Perspective

Use Case Diagram of Issue Tracking System

*searches*

1

1 *requests*

1

1

1..\*

\*

search\_book



1..\*

check\_limit

check\_availability

User 1

issue\_book

*request\_renew*

<<include>>

*monitors\_request*

1

*monitors\_renew* 1

1

*performs*

*give\_book*

<<include>>

0..\*

1..\*

renew\_book

verify\_member

<<include>>

*take\_book*

1

1 Librarian

Student

0..\*

1..\*

*adds\_new\_book*

*perform\_transaction\_updation*

Staff

\*

\*

return\_book

View\_logs

<<extend>>

add\_book

\*

calculate\_fine

update\_record

This diagram is a graphic representation of the interactions among the elements of an issue tracking system. It shows the methodology used in system analysis to identify, clarify, and organize system requirements of an issue tracking system. The main actors of the system in this use case diagram are Super Admin, System User, Tester, and Developer, who perform different types of use cases such as Manage Issue, Manage Issue Feature, Manage Tracking, Manage Developer, Manage Tester, Manage Project, Manage Bug, Manage Users, and Full Issue Tracking\_System\_Operations.

## Product Function

Entity Relationship Diagram of Issue Tracking System



This diagram depicts the interactions between the pieces of an issue tracking system. It demonstrates the system analysis process used to define, clarify, and arrange the system requirements of an issue tracking system. In this use case diagram, the main system actors are Super Admin, System User, Tester, and Developer, who perform various types of use cases such as Manage Issue, Manage Issue Feature, Manage Tracking, Manage Developer, Manage Tester, Manage Project, Manage Bug, Manage Users, and Full Issue Tracking System Operations.

## User Classes and Characteristics

In the context of an issue tracking system, there will be two types of users: customers and admins. Customers will be able to use all the app’s features, while admins will have access to additional features such as managing product listings and discounts.

The features that are available to the Developers are:-

* + - **Issue reporting and categorization.**
    - **Issue assignment and status tracking.**
    - **Collaboration tools for developers**
    - **Access control and communication channels**
    - **Customizable workflows**
    - **Reporting and analytics**
    - **Integration with other tools**

The features that are available to the Maintainers are:-

* + - **Customizable workflows**
    - **Access control and communication channels**
    - **Reporting and analytics**
    - **Integration with other tools**
    - **Backup and recovery**
    - **Security**
    - **Maintenance and support**

## Operating Environment

Issue tracking systems are versatile tools with varying operating environments. Typically web-based, users access them through browsers such as Google Chrome or Mozilla Firefox. For self-hosted solutions, server requirements including the operating system (e.g., Linux or Windows) and web server (e.g., Apache, Nginx) must be considered. Database compatibility, often supporting systems like MySQL or PostgreSQL, is crucial. Integration with version control systems may necessitate compatibility with tools like Git or Mercurial. Client-side requirements, such as necessary plugins, should be addressed. Reliable network connectivity is essential for web-based systems. Security measures, including firewall configurations and HTTPS, are critical for protecting sensitive data. Scalability considerations involve planning for growing team or project needs. Browser compatibility ensures seamless user experience. Always consult the official documentation for precise information tailored to the specific issue tracking system in use.

**2.5** **Assumptions and Dependencies**

The assumptions are:-

* + - The users have access to a reliable internet connection for web-based systems.
    - The issue tracking system will be accessed through commonly used web browsers (e.g., Chrome, Firefox, Edge, Safari).
    - For self-hosted solutions, the hosting environment (server) meets the specified requirements outlined in the system's documentation.
    - The database management system, such as MySQL or PostgreSQL, is properly configured and available.
    - Integration with version control systems assumes the presence and proper configuration of the required version control software (e.g., Git, Mercurial).
    - Users have the necessary permissions and access rights to perform actions within the issue tracking system.
    - Regular backups of the issue tracking system data are implemented to prevent data loss.
    - Security measures, including firewalls and HTTPS, are in place to protect sensitive information.
    - The team is trained or has access to resources for effective utilization of the issue tracking system.
    - The issue tracking system can be scaled to accommodate growing project or team requirements

**The dependencies are:-**

Internet Connectivity:

The system relies on a stable internet connection, especially for web-based implementations.

Web Browsers:

Users need access to compatible web browsers (e.g., Chrome, Firefox, Edge, Safari) to interact with the system.

Server Environment:

Self-hosted systems depend on a properly configured server environment, including the operating system and web server.

Database Management System:

The issue tracking system is dependent on a compatible and properly configured database management system (e.g., MySQL, PostgreSQL).

Version Control Software:

Integration with version control systems requires the presence and correct configuration of specific version control software (e.g., Git, Mercurial).

User Permissions:

The proper assignment of user permissions is essential for controlling access and actions within the issue tracking system.

Backup System:

Regular backup procedures are a dependency to ensure data integrity and prevent loss in case of unforeseen issues.

Security Measures:

Implementation of security measures, such as firewalls and HTTPS, is crucial to protect sensitive data.

Training and Resources:

Users depend on adequate training and access to resources for effective utilization of the issue tracking system.

Scalability:

The ability to scale the system to accommodate growing project or team requirements is a critical dependency.Internet Connectivity:

The system relies on a stable internet connection, especially for web-based implementations.

Web Browsers:

Users need access to compatible web browsers (e.g., Chrome, Firefox, Edge, Safari) to interact with the system.

Server Environment:

Self-hosted systems depend on a properly configured server environment, including the operating system and web server.

Database Management System:

The issue tracking system is dependent on a compatible and properly configured database management system (e.g., MySQL, PostgreSQL).

Version Control Software:

Integration with version control systems requires the presence and correct configuration of specific version control software (e.g., Git, Mercurial).

User Permissions:

The proper assignment of user permissions is essential for controlling access and actions within the issue tracking system.

Backup System:

Regular backup procedures are a dependency to ensure data integrity and prevent loss in case of unforeseen issues.

Security Measures:

Implementation of security measures, such as firewalls and HTTPS, is crucial to protect sensitive data.

Training and Resources:

Users depend on adequate training and access to resources for effective utilization of the issue tracking system.

Scalability:

The ability to scale the system to accommodate growing project or team requirements

is a critical dependency.

## Requirement

Software Configuration:-

For self-hosted solutions, a dedicated server or servers are needed.

Multi-core processors (the exact number depends on the expected load).

Hardware Configuration:-

Sufficient RAM (Random Access Memory) to

handle concurrent user connections and data processing.

Adequate storage space for the database and system files.

Fast and reliable storage with enough capacity to

handle the expected volume of data.

SSDs (Solid State Drives) are often preferred for

improved performance.Database: MS SQL Server (back end)

## Data Requirement

An issue tracking system requires comprehensive data management to effectively monitor and resolve project issues. Essential data includes information about each issue, such as its title, description, status, priority, assignee, reporter, creation date, and due date. User data, encompassing user profiles, roles, and permissions, is crucial for secure access control. Collaboration and comments data facilitate communication, while version control integration captures code changes associated with specific issues. Notification data ensures timely updates, and reporting and analytics data provide insights into project progress. An audit trail, including change history and user activity, enhances transparency and accountability. Configuration data, encompassing custom fields and project settings, allows tailoring the system to specific project needs. Integration data handles connections with third-party tools, and backup and recovery data ensures data integrity and system continuity

# External Interface Requirement

## GUI

Designing the graphical user interface (GUI) for an issue tracking system involves creating an intuitive and efficient layout to support users in managing and tracking project issues. Here are key elements and considerations for the GUI:

Dashboard Overview:

Provide a dashboard with a summarized overview of open issues, recent updates, and key project metrics. This gives users a quick glance at the project's status.

Issue Listing:

Display a list of issues with relevant details such as issue ID, title, status, priority, and assignee. Implement filtering and sorting options to help users quickly find specific issues.

Issue Details View:

Create a detailed view for each issue, presenting information like the description, comments, attachments, and history of changes. Include options for editing and updating issue details.

Workflow Visualization:

Visualize the workflow stages of each issue, making it easy for users to understand the current state and progression of issues.

User Profiles and Permissions:

Design a user management section where administrators can configure user profiles, roles, and permissions. This ensures controlled access to the system.

Collaboration Features:

Integrate features for collaboration, including comments, file attachments, and mentions. A threaded comment system can enhance communication around specific issues.

Notification Center:

Implement a notification center to alert users about updates, assigned tasks, and relevant activities. Users should have the ability to customize notification preferences.

Search and Filters:

Include a robust search functionality and filters to enable users to locate specific issues efficiently based on various criteria such as keywords, status, and assignee.

Reporting and Analytics:

Incorporate a reporting section with customizable reports and analytics dashboards. Users should be able to track key metrics and generate insights into project performance.

Configuration Settings:

Offer a section for configuring system settings, including custom fields, project settings, and integration options. This allows customization based on the specific needs of the project.

Mobile Responsiveness:

Ensure that the GUI is responsive and optimized for mobile devices, allowing users to access and manage issues on the go.

Intuitive Navigation:

Design an intuitive navigation system with clear menus and sections, making it easy for users to move between different areas of the application.

Feedback Mechanism:

Include mechanisms for users to provide feedback or report issues with the system. This promotes continuous improvement.

Dark Mode:

Consider adding a dark mode option for users who prefer a darker interface, promoting user comfort during extended usage.

Login Interface:-

The login interface for the issue tracking system should feature standard fields for username and password, implementing secure password practices. It may include options like "Remember Me"

for user convenience and a "Forgot Password" link to facilitate secure password recovery.

Search:-

The issue tracking system's search functionality is designed with a prominent quick search bar, facilitating users in swiftly accessing specific issues using keywords or IDs. Advanced search options enable precise filtering based on criteria such as status, assignee, and priority.

Categories View:-

The categories view in the issue tracking system provides users with a structured and organized overview of issues based on predefined categories or labels. This feature allows for easy grouping and classification of issues, aiding users in quickly identifying patterns or common themes.

# System Features

The issue tracking system incorporates several key features to streamline project management. Its user-friendly interface includes functionalities such as robust issue creation and management, offering detailed tracking options like status, priority, and assignee. Collaborative tools, including comments and file attachments, enhance communication among team members. The system's reporting and analytics features provide valuable insights, while customization options, such as configurable workflows and project-specific settings, ensure adaptability to diverse project requirements. These features collectively contribute to an efficient and flexible issue tracking environment, supporting effective collaboration and decision-making within the project team.

# Other Non-functional Requirements

## Performance Requirement

The issue tracking system is designed to deliver responsive performance with quick loading times, ensuring users can efficiently navigate and interact with the interface. It should handle concurrent user activities seamlessly, maintaining optimal performance even during periods of increased usage or high traffic. The system aims for low latency in retrieving and updating issue data, providing a smooth and responsive user experience.

## Safety Requirement

Safety requirements for the issue tracking system include robust data encryption during transmission, strict access controls, and comprehensive audit trails to ensure data security and user accountability. Regular security audits, compliance with regulations, and user training on security practices further contribute to maintaining a secure and resilient system.Safety requirements for the issue tracking system include robust data encryption during transmission, strict access controls, and comprehensive audit trails to ensure data security and user accountability. Regular security audits, compliance with regulations, and user training on security practices further contribute to maintaining a secure and resilient system.

## Security Requirement

## 

Security requirements for the issue tracking system encompass the implementation of strong authentication measures and access controls to safeguard sensitive data. Additionally, regular security audits and adherence to privacy regulations are crucial for maintaining a resilient and compliant security posture.

## Requirement attributes

Requirement attributes for the issue tracking system encompass clarity and specificity, ensuring that each requirement is clearly defined and addresses a specific aspect of functionality or performance. Additionally, attributes like traceability and prioritization facilitate effective project management by enabling the tracking of requirements through the development process and prioritizing them based on importance..

## Business Rules

Business rules for the issue tracking system dictate how issues are created, assigned, and resolved, establishing clear guidelines for user interactions. Additionally, these rules may define escalation procedures, ensuring timely issue resolution and effective project management.

## User Requirement

User requirements for the issue tracking system encompass a user-friendly interface that facilitates seamless issue creation and management. Users expect efficient collaboration tools, including comments and file attachments, along with an intuitive search functionality for quick issue retrieval. The system should offer customization options, allowing users to configure workflows and project settings to align with their specific needs. Access controls and security features are crucial for protecting sensitive data, while robust reporting and analytics capabilities provide valuable insights into project progress. Mobile accessibility is essential to ensure users can manage issues on the go, enhancing overall user satisfaction and productivity. Regular user training and responsive support mechanisms contribute to a positive user experience and effective utilization of the system.

# Other Requirements

## Data and Category Requirement

The issue tracking system must efficiently manage a diverse set of data elements, encompassing issue titles, descriptions, status, priority, assignees, reporters, creation dates, and due dates. Collaborative data, including comments, attachments, and integration data from version control systems, should be securely handled to facilitate effective communication among team members. User data, consisting of profiles, roles, and permissions, requires robust security measures to safeguard sensitive information. Customization options for configuration data, such as custom fields and project settings, must be flexible to adapt to varying project needs. To ensure accountability and transparency, the system should maintain an audit trail, logging change history, user activity, and regular backups for reliable system recovery. Additionally, the system should support categorization of issues based on customizable labels, providing an organized structure for users to classify and navigate issues efficiently. Features like saved searches enhance user productivity, allowing for quick retrieval and management of categorized issues within the system.

## Appendix

The project's appendix includes essential supplementary materials. This encompasses a glossary, user manual, system architecture diagram, security documentation, technical specifications, testing details, change log, user training materials, project timeline, feedback summary, and disaster recovery plan, providing comprehensive insights into the issue tracking system's development, functionality, and maintenance.

## Glossary

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

Issue:

A task, bug, or feature request that is being tracked and managed within the system.

Status:

The current state or condition of an issue, indicating whether it is open, in progress, resolved, or closed.

Priority:

The level of importance assigned to an issue, determining its urgency in terms of resolution.

Assignee:

The individual or team responsible for addressing and resolving a specific issue.

Reporter:

The person who initially identifies or reports an issue within the system.

Workflow:

The predefined sequence of states that an issue progresses through, from creation to resolution.

Comments:

Additional information, updates, or discussions related to an issue, facilitating collaboration among team members.

Attachments:

Files, documents, or images linked to an issue for reference or additional context.

Version Control:

The system or process used to manage changes to source code, often integrated with the issue tracking system.

Audit Trail:

A record of changes made to issues, providing a historical timeline of modifications and user activities.

Custom Fields:

Additional data fields that can be configured and customized based on project requirements.

Dashboard:

A visual overview of key project metrics, issues, and progress accessible from the main interface.

Access Control:

The mechanism for regulating user permissions, ensuring appropriate levels of access to system features.

Integration:

The connection or interaction between the issue tracking system and other tools, such as version control or project management tools.

Backup:

A copy of the system's data created for the purpose of recovery in the event of data loss or

system failure.

## Class Diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes’ structure and their relationships to each other frozen in time represent the static model. In this project there are certain main classes

which are related to other classes required for their working. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization. The relationships are depicted using a role name and multiplicities. Here ‘Librarian’, ‘Member’ and ‘Books’ are the most important classes which are related to other classes.

