**SOFTWARE**

**REQUIREMENTS SPECIFICATION**

**For**

# ISSUE TRACKING SYSTEM

**Prepared by:-**

**N.GOKUL**

**G.JAGAN**

**S.INDRA KUMAR**

## 1. Introduction

### 1.1 Purpose

The main objective of this document is to illustrate the requirements of issue tracking system.The document gives the detailed description of the both functional and non-functional requirements proposed by the client. The scope of this project is to provide a user-friendly environment for reporting, categorizing, assigning, and tracking issues in software development.The system aims to provide efficient tracking and management of issues or bugs in software development projects.This project describes the hardware and software interface requirements using ER diagrams and UML diagrams.

### 1.2 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

### 1.3 Scope of Development Project

The project involves transitioning the manual issue tracking system into a web-based application to enable users to access information related to reported issues, issue statuses, and collaborative solutions effortless.Primarily targeted at developers and project stakeholders, this application will serve as a comprehensive user interface for managing the issue tracking process, allowing users to report issues, assign them, and track their resolution status. The system is designed to facilitate collaboration among developers and streamline the overall issue management process.The application's utility extends to both existing and new projects, providing a centralized platform for efficient issue tracking, assignment, and resolution. It is particularly beneficial for software development teams and organizations where updates and modifications can be easily incorporated to meet specific project requirements.

### 1.4 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

### 1.5 References

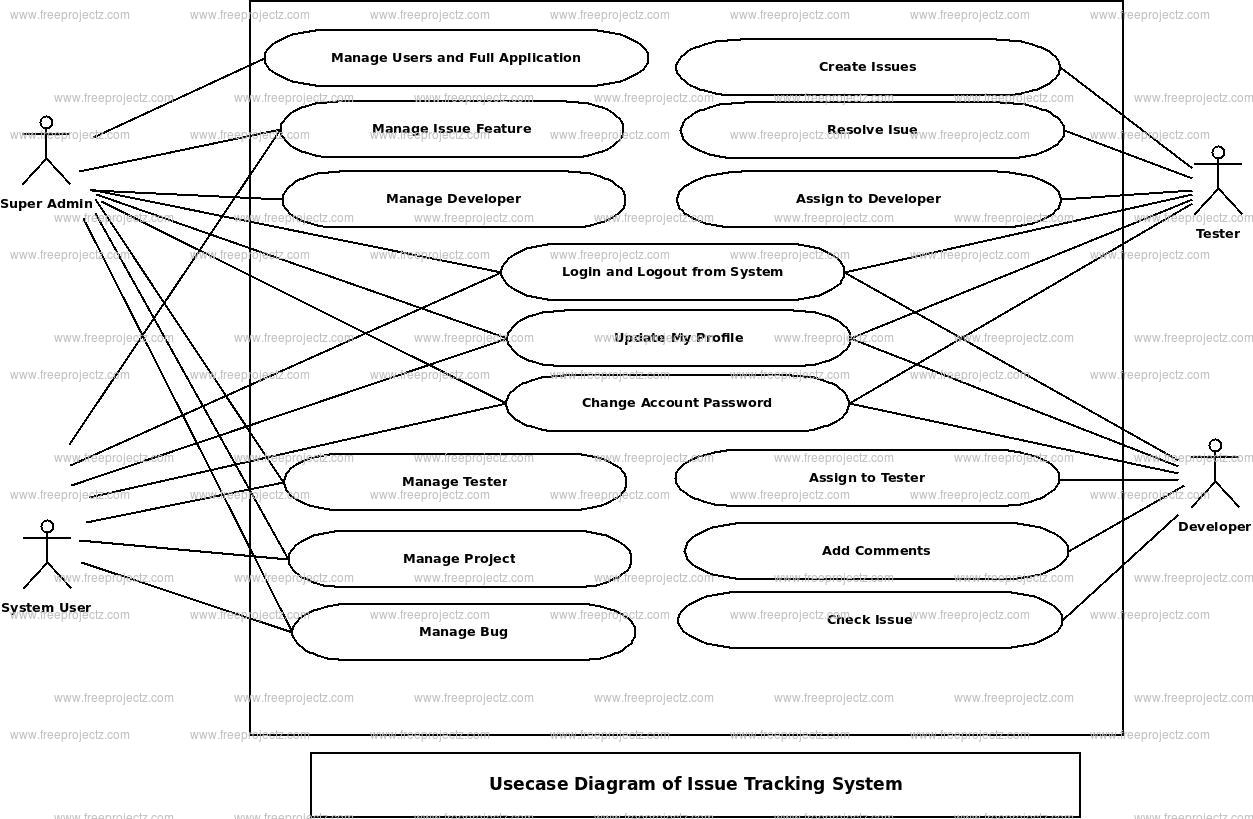
* + - Websites
    - jira
    - Software Requirements (Microsoft) Second EditionBy Karl E. Wiegers
* Software Engineering: A Practitioner’s Approach Fifth Edition By Roger S. Pressman
  + - Websites

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

## 2. Overall Descriptions

### 2.1 Product Perspective

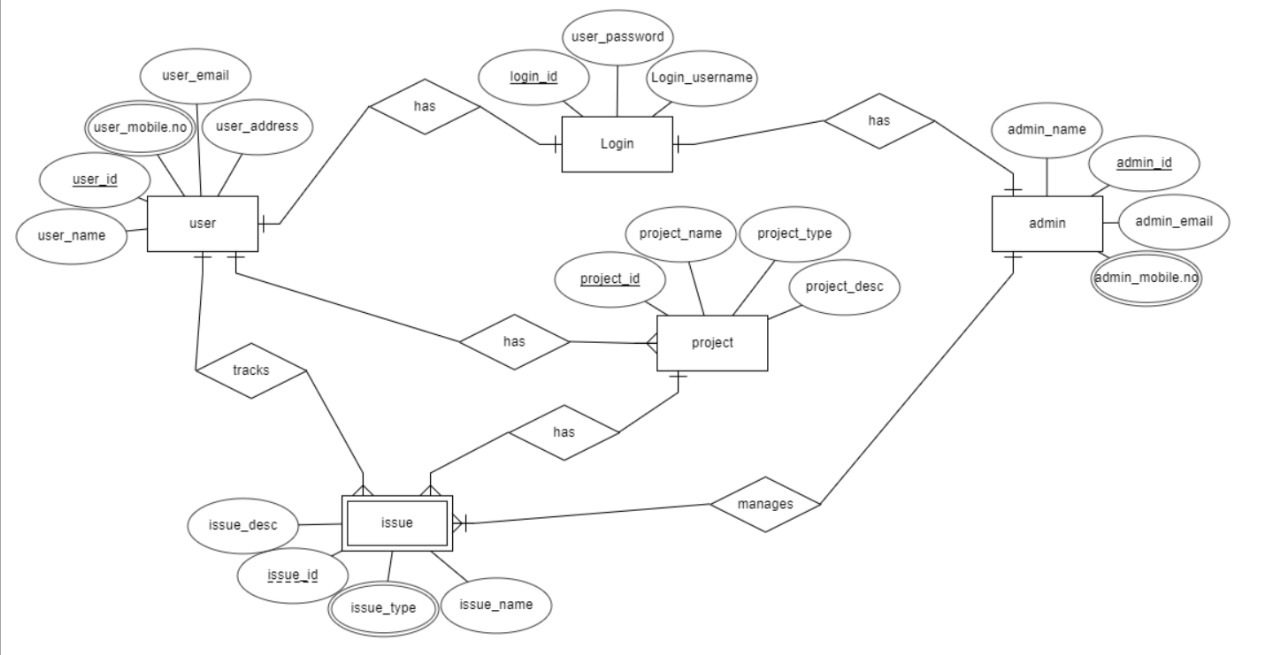
Use Case Diagram of Issue tracking system



This is a broad level diagram of the project showing a basic overview. This use case show all connections between the Admin,User,Tester and Developer.This case diagram fully explain the concept of this project about how the Developers report issues and how it is tracked , how it resolved after that the collaboration and contributions of the users.

### 2.2 Product Function

Entity Relationship Diagram of Library Management System



The Online Library System provides online real time information about the books available in the Library and the user information. The main purpose of this project is to reduce the manual work. This software is capable of managing Book Issues, Returns, Calculating/Managing Fine, Generating various Reports for Record-Keeping according to end user requirements. The Librarian will act as the administrator to control members and manage books. The member’s status of issue/return is maintained in the library database. The member’s details can be fetched by the librarian from the database as and when required. The valid members are also allowed to view their account information.

### 2.3 User Classes and Characteristics

The issue tracking system accommodates two main user classes: Administrators and Users

Administrator Features:

Administrators, acting as the controllers, have overarching privileges, enabling them to:

\* Report issues and assign them to developer

\* View different issue categories within the system.

\* Access the list of reported issues in each category.

\* Manage the resolution of reported issues.

\* Add new issues and their details to the tracking system.

\* Edit information for existing issues.

\* Generate reports on existing issues and their status.

\* Access all user accounts for administrative purposes.

User Features:

Users, which include developers and project stakeholders, have user-specific functionalities, such as:

\*Viewing available issue categories.

\*Accessing the list of reported issues within each category.

\*Contributing to issue resolution through collaboration tools.

\*Tracking the status of reported issues.

\*Requesting new features or reporting bugs.

\*Reviewing the history of issues they've reported.

\*Searching for specific issues based on their requirements.

This user-centric design ensures that Administrators have the necessary tools for efficient issue tracking and administration, while Users are provided with features tailored to their involvement in the development process.

### 2.3 operating environment

### Effective operating management for the issue tracking system is pivotal for seamless functionality and user satisfaction. This involves a strategic approach to system deployment, including meticulous installation and configuration procedures. Comprehensive user training materials should be developed and training sessions conducted to ensure that administrators and users are proficient in utilizing the system. To maintain optimal performance, regular monitoring, security measures, and a robust backup and recovery system are imperative. A responsive issue resolution and support mechanism, coupled with scalability considerations and performance optimization, contribute to sustained system efficiency. Adherence to compliance standards, thorough documentation, and a proactive feedback loop for continuous improvement round out the key components of successful operating management.

### 2.5 Assumptions and Dependencies

Assumptions:

* Database Availability
* User Authentication
* Internet Connectivity for Collaboration Tools
* Regular Backups
* Collaboration Tools Usage

Dependencies:

* Dependency: JavaFX Framework
* Dependency: Relational Database Management System (DBMS)
* Dependency: Java Development Kit (JDK) Compatibility
* Dependency: Access Control Mechanism

### 2.6 Requirement

Software Configuration:-

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database. Operating System: Windows NT, windows 98, Windows XP

Language: Java Runtime Environment, Net beans 7.0.1 (front end)

Database: MS SQL Server (back end)

Hardware Configuration:-

Processor: Pentium(R)Dual-core CPU

Hard Disk: 40GB

RAM: 256 MB or more

### 2.7 Data Requirement

The issue tracking system relies on various input sources to efficiently manage reported issues. This includes detailed issue reports containing descriptions, steps to reproduce, severity levels, and issue categories. Additionally, user information, authentication credentials, book details (if applicable), collaboration inputs such as comments and attachments, and system configurations contribute to the input data. The system generates crucial output data to provide users with valuable insights and facilitate informed decision-making. This encompasses the current status and detailed information of reported issues, user notifications on changes in issue status, and various reports categorizing issues based on type, severity, or status

## 3. External Interface Requirement

### 3.1 GUI

The software provides good graphical interface for the user and the administrator can operate on the system, performing the required task such as create, update, viewing the details of the book.

* Centralized dashboard provides visual project status snapshots.
* Easily create, view, edit, and update issues with customizable fields and workflows.
* Visual representation of issue workflows aids in tracking progress through different stages.
* Integrated communication tools for comments, file attachments, and real-time notifications.
* Robust search and filter options for quickly locating specific issues.
* Comprehensive tools for insights into project performance and data-driven decision-making.
* Mobile responsiveness ensures a seamless experience across various devices.
* Role-based access controls and security features safeguard sensitive information.

.Login Interface:-

* Secure Authentication: Use a secure mechanism for user authentication, such as usernames and passwords or multi-factor authentication.
* Forgot Password: Include a password recovery option through email or security questions.
* Account Lockout: Implement a policy to lock accounts after multiple failed login attempts to prevent unauthorized access.
* Session Management: Set session timeouts for automatic logouts after periods of inactivity.
* User Feedback: Provide clear error messages for login failures to guide users.
* Security Measures: Use HTTPS for secure data transmission and consider anti-bot measures like CAPTCHA.
* Remember Me Option: Include a "Remember Me" option for user convenience.
* Branding and Design: Maintain a consistent and visually pleasing design with branding elements.
* Multi-Factor Authentication: Consider implementing an additional layer of security with MFA.

## 4. System Features

An effective issue tracking system features essential capabilities such as streamlined issue creation, assignment, and tracking, along with customizable workflows aligning with organizational processes. Collaboration tools, including commenting and file attachments, facilitate communication among team members.

* Real-time notifications and alerts keep stakeholders informed, while robust search and filtering options ensure quick issue identification.
* The system's reporting and analytics tools enable the generation of insightful reports. Custom fields and attributes allow the capture of specific information, and integration with other tools ensures seamless collaboration across platforms.
* An audit trail and history feature enhances accountability, while automation and workflow customization streamline repetitive tasks. Role-based access control safeguards data integrity, and mobile accessibility provides on-the-go management.
* The system prioritizes security measures, including protocols and updates, and maintains a knowledge base for reference and guidelines, contributing to an efficient and collaborative issue management process.

## 5. Other Non-functional Requirements

### 5.1 Performance Requirement

The performance requirements for an issue tracking system encompass crucial aspects to ensure optimal functionality. This includes defining acceptable response times for user interactions, specifying throughput capabilities to handle concurrent requests, ensuring scalability to accommodate user growth and increased data volume, establishing reliability and availability targets with measures for uptime and redundancy, maintaining data integrity through validation checks, implementing caching and optimization techniques, considering network performance implications, addressing security measures without compromising system speed, conducting load testing to identify and alleviate bottlenecks, ensuring compatibility with different browsers and devices, and implementing effective logging and monitoring systems for proactive issue identification. Regular assessments and adjustments based on performance metrics contribute to a system that operates efficiently and provides a positive user experience.

### 5.2 Safety Requirement

* Safety requirements in an issue tracking system involve measures to protect user data, ensure system stability, and prevent unauthorized access.
* Encryption protocols should be employed to secure sensitive information, and access controls must be implemented to restrict system entry to authorized personnel only.
* Regular system backups and disaster recovery plans are essential to safeguard against data loss and ensure business continuity.
* Additionally, the system should adhere to industry standards and best practices for cybersecurity, undergo regular security audits, and promptly address any identified vulnerabilities.
* These safety measures collectively contribute to the reliability and security of the issue tracking system, fostering user trust and protecting against potential threats.

### 5.3 Security Requirement

* System will use secured database
* The Issue Tracking System prioritizes security through various measures.
* User authentication and authorization are rigorously implemented, ensuring that only authorized individuals access specific functionalities.
* Robust encryption protocols, such as HTTPS, secure data transmission, and sensitive data storage practices include encryption and protection against unauthorized access.
* Logging and auditing mechanisms track user activities for forensic analysis, while secure configurations and guidelines maintain a secure system setup..

### 5.4 Requirement attributes

* There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
* The project should be open source
* The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
* The user be able to easily download and install the system

### 5.5 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 admin nor member should cross the rules and regulations.

### 5.6 User Requirement

The users of the system are members and Librarian of the university who act as administrator to maintain the system. The members are assumed to have basic knowledge of the computers and internet browsing. The administrators of the system should have more knowledge of the internals of the system and is able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system. The proper user interface, user manual, online help and the guide to install and maintain the system must be sufficient to educate the users on how to use the system without any problems.

The admin provides certain facilities to the users in the form of:-  Backup and Recovery

* Forgot Password
* Data migration i.e. whenever user registers for the first time then the data is stored in the server
* Data replication i.e. if the data is lost in one branch, it is still stored with the server
* Auto Recovery i.e. frequently auto saving the information
* Maintaining files i.e. File Organization
* The server must be maintained regularly and it has to be updated from time to time

## 6. Other Requirements

### 6.1 Data and Category Requirement

In the Issue Tracking System project, various users play crucial roles in managing and resolving software-related issues. Administrators serve as system controllers, overseeing the entire process, managing user accounts, and ensuring system security. Developers actively address and resolve reported issues, collaborating with others to implement fixes. Project stakeholders contribute to project oversight, offering valuable insights and participating in discussions. General users, encompassing both issue reporters and observers, engage in the feedback and collaboration process. Together, these user roles create a collaborative environment, ensuring systematic issue tracking, resolution, and overall project success.

### 6.2 Appendix

A: Admin, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; L: Library, Librarian; M:

Member; N: Non-functional Requirement; O: Operating environment; P:

Performance,Perspective,Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

### 6.3 Glossary

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

ACRONYMS:

* API - Application Programming Interface:
* Used for communication between different software systems.
* SQL - Structured Query Language:
* Used for managing and querying databases.
* UI - User Interface:
* Refers to the graphical interface through which users interact with the system.
* VCS - Version Control System:
* Manages changes to source code over time (e.g., Git).
* QA - Quality Assurance:
* Ensures that the software meets specified requirements and quality standards.
* CI/CD - Continuous Integration/Continuous Deployment:
* Practices that automate the testing and deployment of code changes.
* LDAP - Lightweight Directory Access Protocol
* Used for accessing and maintaining distributed directory information services.
* HTTPS - Hypertext Transfer Protocol Secure:

CONVENTIONS:

* + Naming Conventions
  + Commit Message Conventions
  + Coding Style Conventions
* Workflow Conventions
* Documentation Conventions

### 6.4 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.