SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**Issue Tracking System**

**Prepared by:-**

*Srimathi B*

*Praveena JP*

*Simi A*

# Introduction

## Purpose

The main objective of this document is to illustrate the requirements of the project Issue Tracking system. The document gives the detailed description of the both functional and non-functional requirements proposed by the client. Issue Tracking for Improving Software Reliability is an automated system that can be useful to admin and developers in any functional organization. Issue Tracking System gives the facility to define the tasks in the organization and also allows the admin to track the issues spent by the developer for that particular task. It can help admin for Issue estimation per project or application. This project describes the hardware and 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

## The "Issue Tracking for Improving Software Reliability" project aims to automate and enhance task management within functional organizations, providing valuable tools for both administrators and developers. The system encompasses user roles and permissions, enabling secure access control for admins and developers. Users can create and manage tasks, with essential details such as descriptions, priorities, and statuses. Task assignment and progress tracking functionalities empower admins to allocate work efficiently and monitor development progress. Developers can log time spent on tasks, facilitating accurate project timelines and resource allocation. Project and application management features allow categorization and analysis on a per-project or per-application basis.

## The system supports issue estimation and planning, aiding administrators in resource allocation based on task complexities. Reporting and analytics provide insights into project status and issue resolution. Integration with version control systems ensures alignment of code changes with specific tasks. Customization options, security measures, user-friendly interfaces, and scalability considerations contribute to a comprehensive solution. Documentation and training resources ensure effective system utilization, while ongoing maintenance and support address evolving needs, making the project a robust and adaptable asset for organizational efficiency and software reliability improvement.

## 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: A Lexicon of Practice, Principles and Prejudices (ACM Press) by Michael Jackson

Software Requirements (Microsoft) Second EditionBy Karl E. Wiegers

Software Engineering: A Practitioner’s Approach Fifth Edition By Roger S. Pressman

* + - Websites

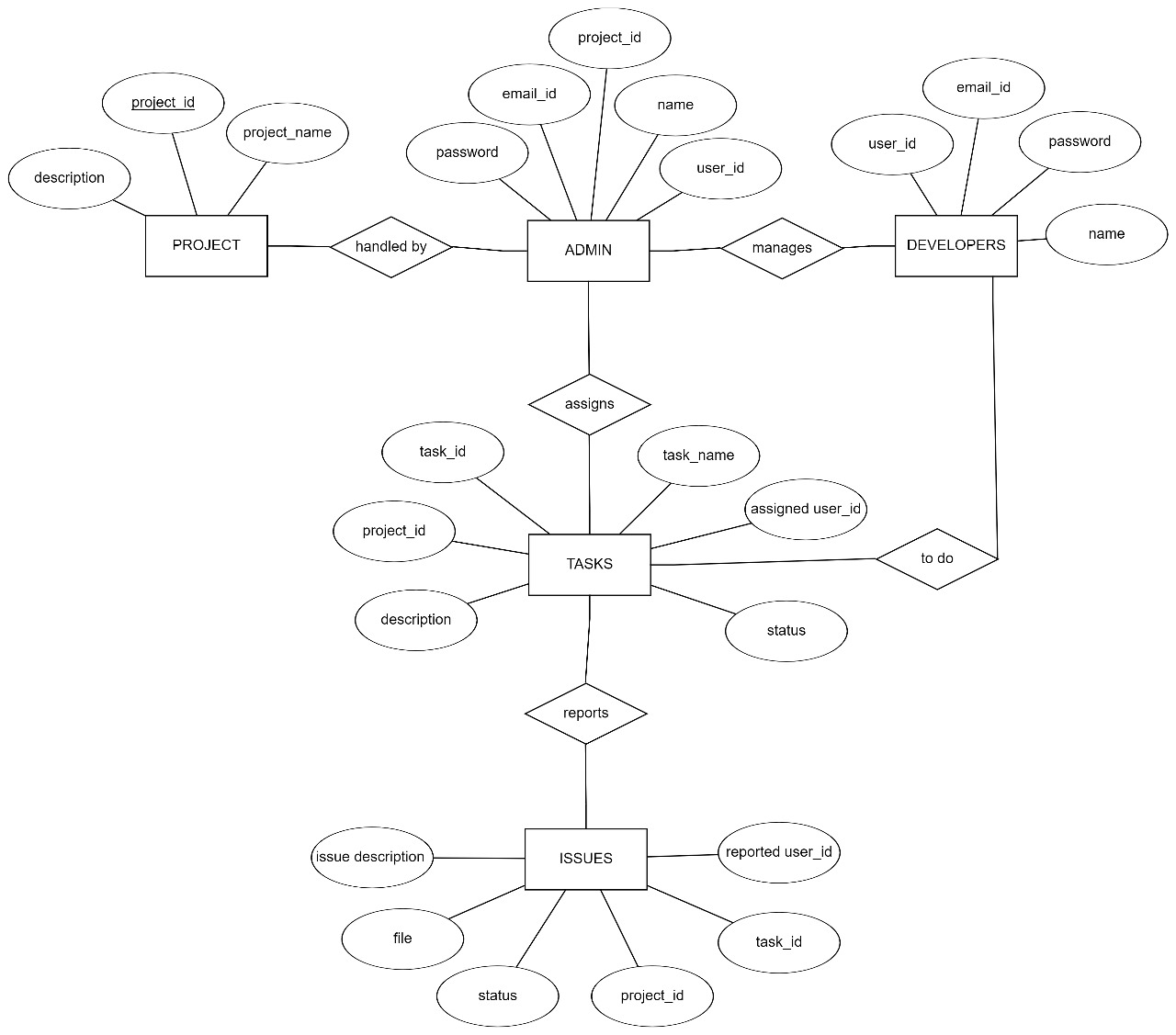
[**http://www.slideshare.net/**](http://www.slideshare.net/)

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

# Overall Descriptions

## Product Perspective

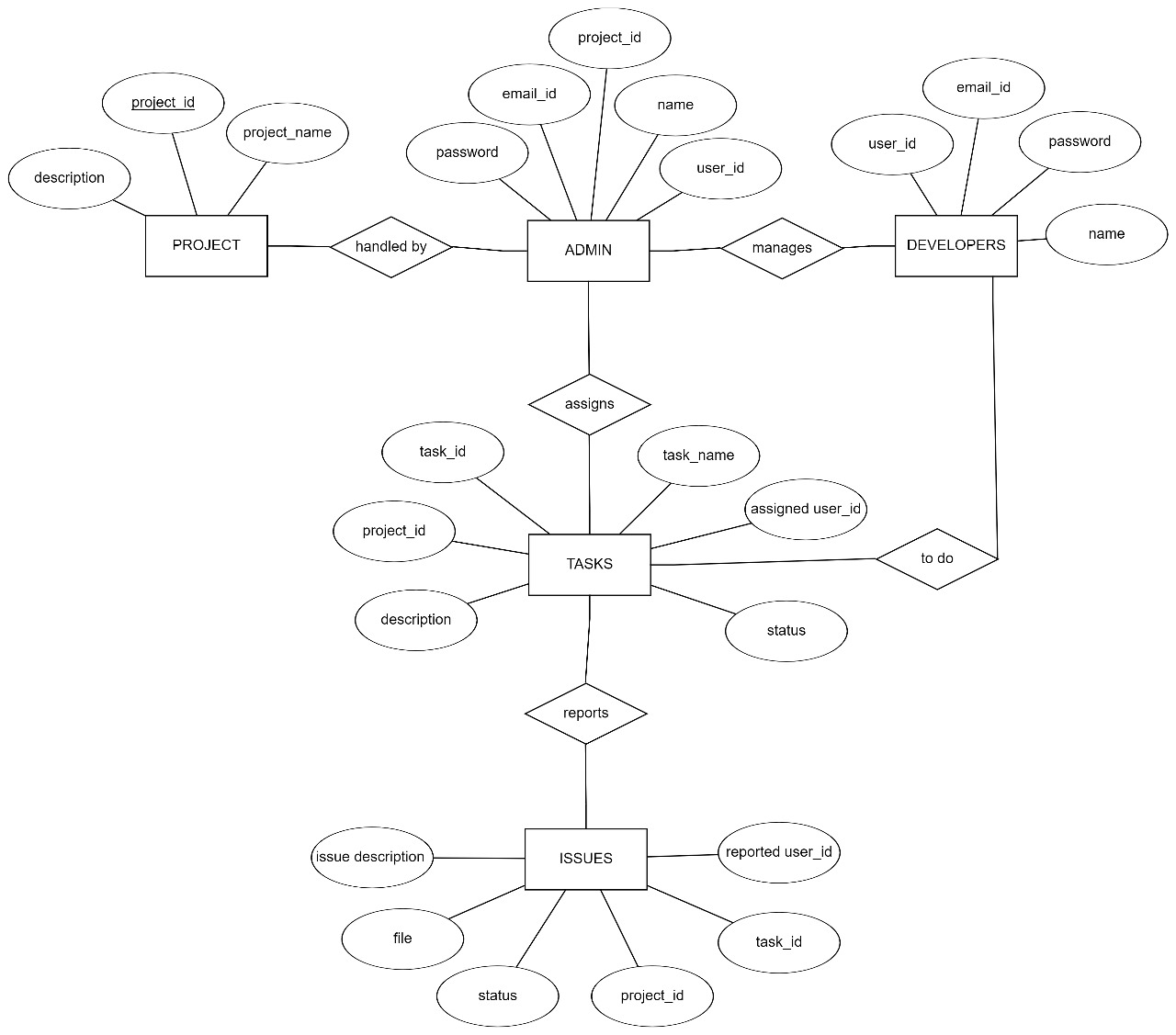
Use Case Diagram of Issue Tracking System



These are just basic classes, and the actual implementation might require more refinement based on specific requirements and the chosen architecture (e.g., MVC, RESTful API). Additionally, you might want to consider using a framework like Spring or Java EE for building the web application, and a database like MySQL or PostgreSQL for data storage. Remember to follow best practices for coding standards, security, and maintainability when developing an issue tracking system.

## Product Function

Entity Relationship Diagram of Issue Tracking System



**2.3 User Classes and Characteristics:**

These are just basic classes, and the actual implementation might require more refinement based on specific requirements and the chosen architecture (e.g., MVC, RESTful API). Additionally, you might want to consider using a framework like Spring or Java EE for building the web application, and a database like MySQL or PostgreSQL for data storage. Remember to follow best practices for coding standards, security, and maintainability when developing an issue tracking system.

The features that are available are:-

User Authentication and Authorization:

* Secure login with role-based access.

Dashboard:

* Overview of project status and recent activities.

Project and Issue Management:

* Create and manage projects.
* Track and update issues with details like priority and status.
* Assign issues to team members.

Attachments and Comments:

* Attach files and add comments to issues.

Custom Fields:

* Add custom fields for project-specific information.

Issue Workflow:

* Customize workflows for issue life cycle.

Reporting and Analytics:

* Generate reports and view data through charts.

Filtering and Search:

* Advanced search functionality for specific issues.

Integration and Extensibility:

* Link issues to code repositories.
* APIs for tool integration.

Administration and Configuration:

* Admin tools for user management and permissions.
* Customization options for fields and workflows.

.

## 2.4Operating Environment

## The operating environment of an issue tracking system encompasses the hardware infrastructure, operating system, and database system it relies on, ensuring compatibility and performance. It operates within a network environment, supporting various client devices and integrating with other systems. Security measures, user authentication, and authorization mechanisms are crucial to safeguard data. Scalability is essential for accommodating growth, and a robust backup and recovery strategy prevents data loss. Compliance with industry regulations may be necessary, and user training and support ensure effective utilization of the system within the organization.

## 2.5Assumptions and Dependencies

The assumptions are:-

* Assumes users have a basic understanding of issue tracking concepts and system usage.
* Assumes consistent user engagement for effective issue reporting and resolution.
* Assumes users input accurate and relevant information when reporting issues.
* Assumes established and understood workflows for issue lifecycle management.
* Assumes timely updates and communication from users and team members regarding issue status and resolution.
* Assumes proper access permissions are granted to users based on their roles and responsibilities.
* Assumes the availability of responsive support for user inquiries and issue resolution.
* Assumes seamless cooperation with integrated tools and systems for comprehensive issue tracking.

The dependencies are:-

## Depends on the compatibility and proper functioning of a specific DBMS for data storage and retrieval.

## Depends on the availability and proper configuration of a web server if the issue tracking system is web-based.

## Depends on the support and compatibility with a specific operating system on which the system is deployed.

## Depends on a reliable user authentication and authorization mechanism to control access and protect against unauthorized use.

## Depends on consistent and reliable network access to ensure users can connect to the system and collaborate on issue tracking activities.

## 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

## Data Requirement

For software development or project-related issues to be managed and tracked efficiently, an issue tracking system built with Java needs a strong collection of data components. One of the basic data requirements is an exhaustive representation of issues, which is usually contained in a class called "Issue". A unique identifier, a summary for a succinct description, a thorough explanation, the issue's current status (open, in progress, closed), its priority level, assignee information, creation date, and a way to record comments or updates should all be included in this class.

# External Interface Requirement

**3.1 User Interface (UI):**

Intuitive and device-agnostic interface for easy user interaction.

Integration:

APIs for seamless integration with other tools, including email, version control, and project management systems.

Reporting and Analytics:

Compatibility with external reporting and analytics tools for data analysis.

Authentication:

Integration with LDAP/Active Directory for user authentication.

Mobile Access:

Support for mobile access through responsive design or dedicated applications.

Notification Services:

Integration with various notification services for timely alerts.

# System Features

Issue tracking system features include workflow management, collaboration tools, real-time status tracking, customizable fields, and reporting for efficient issue resolution and comprehensive project oversight.

* Efficient tracking of issues through defined workflows for systematic resolution.
* Collaboration tools and reporting capabilities for streamlined team communication and informed decision-making.

# Other Non-functional Requirements

## 5.1 Performance Requirement

## The performance requirements of an issue tracking system implemented in Java are crucial for ensuring the system's responsiveness, scalability, and reliability. Firstly, the system should exhibit low latency in processing user requests, providing quick response times when creating, updating, or querying issues. Efficient algorithms and optimized database queries are essential to achieve this responsiveness, minimizing the time it takes to perform common operations. An effective issue tracking system in Java must prioritize low latency, scalability, reliability, efficient resource utilization, and incorporate strategies like caching to meet performance requirements in diverse and demanding development environments. Continuous monitoring and optimization are essential to ensure that the system can adapt and perform well under varying workloads.

## 

## Safety Requirement

## Safety requirements for an issue tracking system include robust data encryption, secure authentication measures, and strict access controls to protect user data and ensure system integrity. Regular security audits are crucial for identifying and addressing potential vulnerabilities, maintaining a secure operational environment.

## 5.3 Security Requirement

Security requirements for an issue tracking system entail implementing robust user authentication, access controls, and data encryption to safeguard against unauthorized access and protect sensitive information. Regular security assessments and updates are essential to address potential vulnerabilities and maintain a secure environment.

## 5.4 Requirement attributes

## The system should have an intuitive user interface for ease of use.

## User interactions, such as creating and updating issues, should be straightforward and well-guided.

## The system must be available for use during regular working hours.

## It should handle high loads and concurrent user activity without compromising performance or stability.

## Response times for common operations, such as creating and updating issues, should be minimal.

## The system should scale efficiently to accommodate a growing number of users and issues.

## 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.

## User Requirement

User requirements for an issue tracking system include an intuitive and user-friendly interface for seamless issue creation, tracking, and collaboration. Additionally, users expect efficient communication tools, customizable workflows, and reporting features to streamline issue resolution and management processes

* Easy navigation and a user-friendly design for efficient issue tracking.
* Customizable workflows to align with diverse project requirements.
* Effective communication tools for seamless team collaboration.
* Straightforward process for creating and submitting issues with relevant details.

# 6.Other Requirements

## 6.1Data and Category Requirement

The data requirements for an issue tracking system include capturing essential details such as issue description, status, priority, assignment, and timestamps to ensure a comprehensive record of each issue. Additionally, category requirements involve organizing issues into distinct categories or types, facilitating efficient sorting, filtering, and analysis for improved issue management. A well-defined categorization system enhances the system's ability to prioritize and address various types of issues effective.

## 6.2 Appendix

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

## Glossary

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

Administrator: A login id representing a user with user administration privileges to the software

User: A general login id assigned to most users

Client: Intended users for the software

SQL: Structured Query Language; used to retrieve information from a database

SQL Server: A server used to store data in an organized format

Layer: Represents a section of the project

User Interface Layer: The section of the assignment referring to what the user interacts with directly

Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed

Data Storage Layer: The section of the assignment referring to where all data is recorded

Use Case: A broad level diagram of the project showing a basic overview

Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system’s cases, their attributes, and the relationships between the classes

Interface: Something used to communicate across different mediums

Unique Key: Used to differentiate entries in a database

**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.

