

**SOFTWARE  
REQUIREMENTS  
SPECIFICATION**

**For**

**ISSUE TRACKING SYSTEM**

**Prepared by:-**

**Dhinakar.V**

**Sarvesh.P**

**Shane Alvin**

# **1.Introduction**

## **1.1 Purpose**

An issue tracking system is a crucial tool for businesses and organizations across various industries, designed to manage and oversee the resolution of problems, tasks, or requests within a project or system. Its purpose goes beyond merely recording issues; it encompasses a range of functionalities that are integral to efficient project management and ensuring smooth operations.

## **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 Document Project**

Document conventions for an issue tracking system involve standardizing the way information is documented, recorded, and organized within the system. These conventions ensure consistency, clarity, and ease of understanding for all stakeholders involved in issue tracking and resolution.

The scope of document conventions for an issue tracking system involves establishing a comprehensive framework that governs the documentation, organization, and management of information related to issues, tasks, bugs, or requests within the system. These conventions ensure consistency, clarity, and efficiency in handling and resolving issues.

Certainly! The scope of document conventions for an issue tracking system involves a comprehensive outline of guidelines, standards, and practices that govern the documentation, organization, and management of information within the system. These conventions aim to ensure consistency, clarity, and efficiency in recording, tracking, and resolving issues.

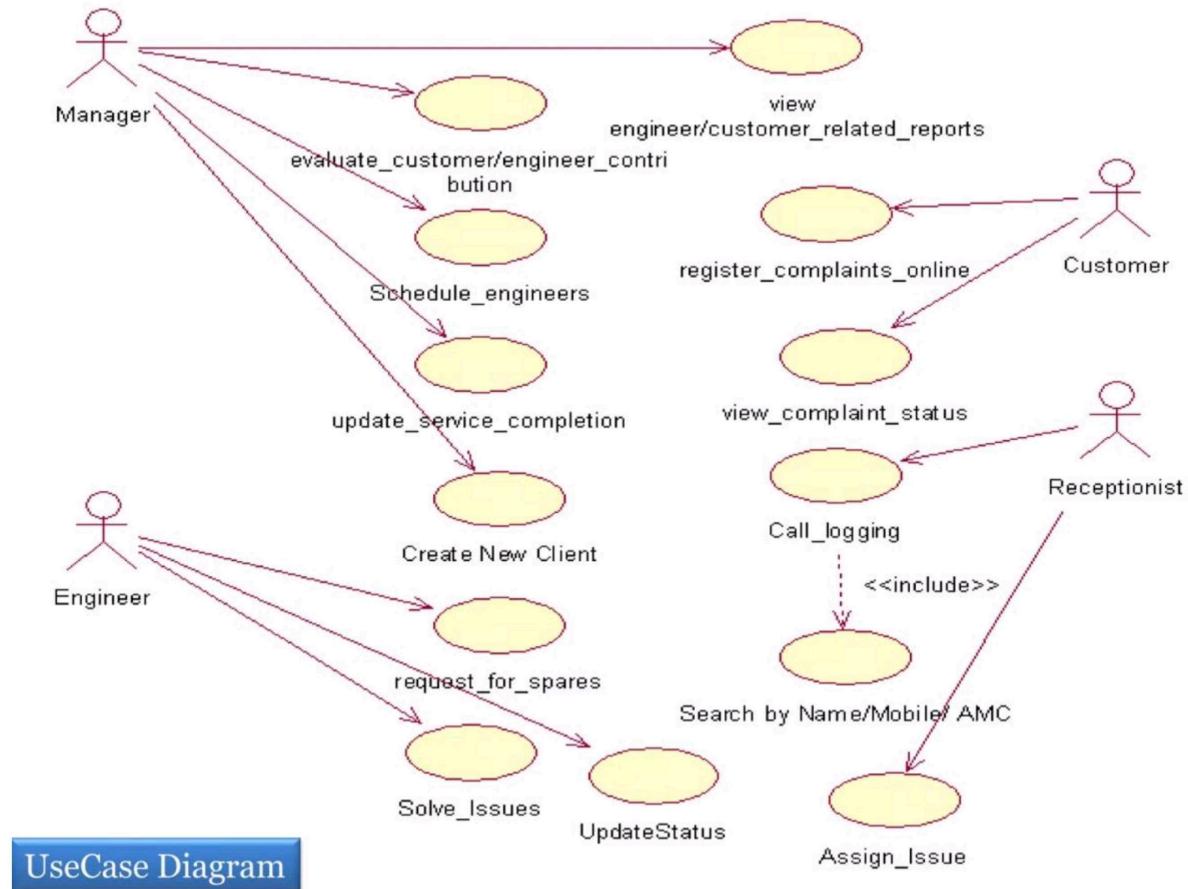
## **1.4 Definitions, Acronyms and Abbreviations**

- ✓ JAVA -> Platform Independence.
- ✓ SQL -> Structured Query Language.
- ✓ ER -> Entity Relationship.
- ✓ UML -> Unified Modelling Language.
- ✓ IDE -> Integrated Development Environment.
- ✓ SRS -> Software Requirement Specification.

## 2. Overall Descriptions

### 2.1 product Perspective

Use case diagram of Issue Tracking System.



An issue tracking system is designed to manage and track reported issues or problems within an organization or a software development project.

#### 1. Actors:

**User:** Interacts with the issue tracking system to report issues, view/update issues, and manage their own reported problems.

**Admin:** Manages the overall system, assigns issues, resolves conflicts, and oversees the system's functioning.

#### 2. Use Cases:

**Report Issue:** Users can report a new issue by providing details such as a description, priority, type, etc.

**View Issue Details:** Users and admins can view the details of a reported issue to understand its specifics.

**Update Issue:** Users and admins can update the status, priority, or other details of an issue.

**Assign Issue:** Admins can assign issues to specific individuals or teams responsible for resolving them.

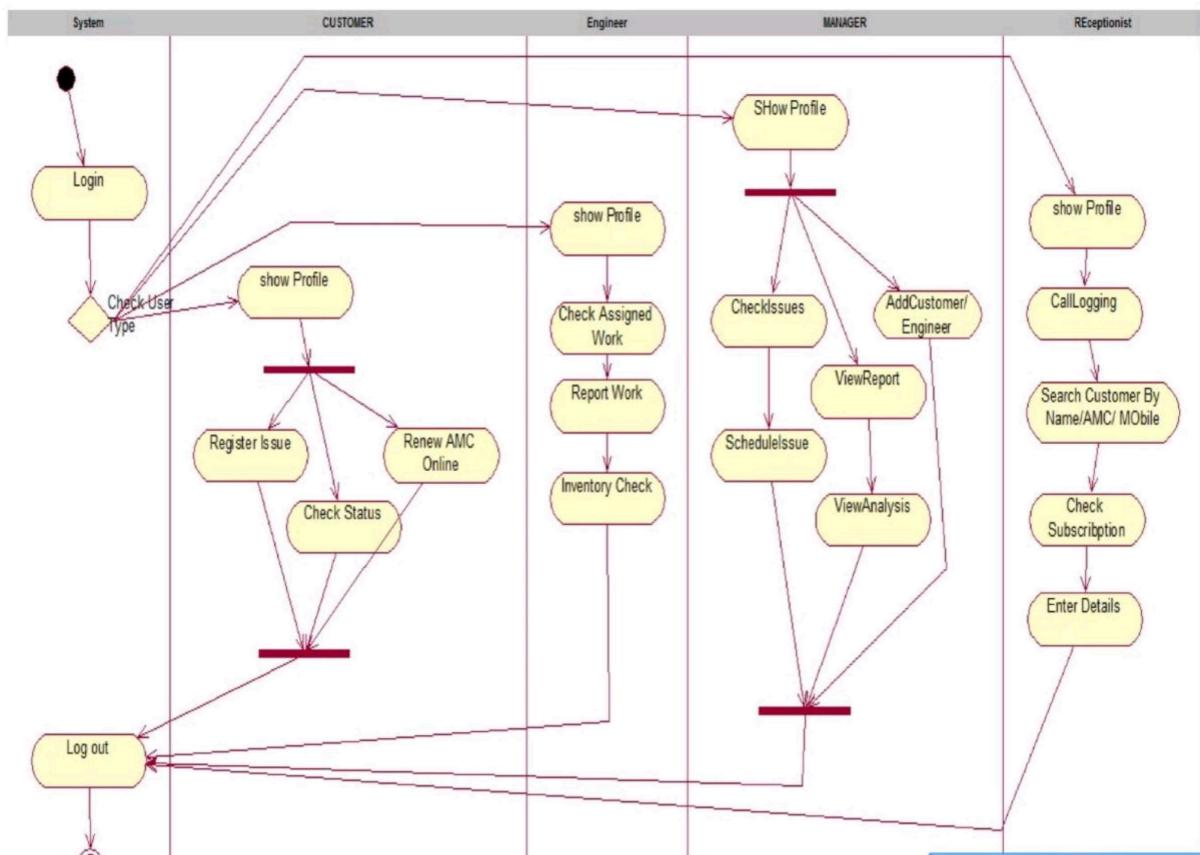
**Resolve Issue:** Admins mark an issue as resolved once the problem is fixed or the task is completed.

**Generate Reports:** Admins can generate reports based on resolved issues, current issues, time taken to resolve, etc.

This use case diagram represents the primary functionalities of an issue tracking system and the interactions between its users and the system itself.

## 2.2 product Function

Entity Relationship Diagram for Issue Tracking System



This is a basic representation and may be expanded or modified based on additional features or complexities within the issue tracking system. It covers the essential entities and their relationships, focusing on Users, Projects, and Issues as the core components . This expanded ERD includes entities like Comments and Attachments linked to Issues, enabling a more detailed tracking system where users can comment on issues and attached relevant files. Additionally, it includes more attributes for users, issues, and projects to capture essential information required for issue tracking and management.

## 2.3 User Classes and Characteristics

In an issue tracking system, various user classes or roles interact with the platform, each having distinct characteristics and permissions. Here are some common user classes along with their typical characteristics in such a system:

### **1. Administrator / Admin:**

#### **Characteristics:**

- Full system access and control.
- Ability to manage users, projects, and system configurations.
- Can create, modify, or delete projects and their settings.
- Access to all issues and extensive reporting capabilities.
- Authority to assign roles and permissions to other users.

### **2. Developer:**

#### **Characteristics:**

- Responsible for resolving and fixing issues reported in the system.
- Access to assigned projects and related issues.
- Can update issue status, add comments, and attach files for issue resolution.
- Limited access to administrative functions compared to the Admin role.

### **3. Reporter / User:**

#### **Characteristics:**

- Creates and reports issues within projects.
- Accesses own reported issues and tracks their status.
- Ability to comment on issues and add attachments for clarification or additional information.
- Limited access to other project issues or administrative settings.

### **4. Manager / Project Manager:**

#### **Characteristics:**

- Oversees specific projects or a set of projects.
- Can view issues and progress within assigned projects.
- May have the authority to assign issues to developers or reassign them.
- Limited administrative access but greater control within assigned projects.

### **5. Viewer / Stakeholder:**

#### **Characteristics:**

- Limited access to view issues and project progress.
- Typically for external stakeholders, clients, or users who need visibility into project statuses and issue resolution.
- Cannot modify or interact directly with issues but can track progress.

Each user class has its unique set of permissions, capabilities, and access levels within the issue tracking system. These roles ensure that the right individuals have appropriate access and control over the system's functionalities, contributing to efficient issue tracking, management, and resolution. These classes can be further tailored or expanded based on the specific needs and complexities of the system or organization using the issue tracking platform.

## 2.4 Operating Environment

The operating environment for an issue tracking system encompasses various aspects, including hardware, software, network infrastructure, and other environmental factors needed to support the system's functioning.

The operating environment for an issue tracking system is a multifaceted ecosystem encompassing various hardware, software, network infrastructure, and security components essential for its optimal functionality. At its core lies the hardware infrastructure, which could consist of servers, storage solutions, and user-end devices like desktops, laptops, or mobile devices. Supporting this hardware are critical software elements such as the issue tracking platform itself, a robust database management system (DBMS) for efficient data handling, and an operating system tailored to the system's requirements. This software setup is fortified by security measures including authentication protocols, encryption methods, and regular security audits to safeguard sensitive data and ensure compliance with industry standards.

## 2.5 Assumptions and Dependencies

Assumptions and dependencies within an issue tracking system represent the foundational conditions, expectations, and requirements that the system relies on for its proper functioning. Here are some typical assumptions and dependencies:

### Assumptions:

- ✓ **User Availability and Accessibility:** The assumption that users (reporters, developers, administrators) have consistent access to the internet and the issue tracking system, allowing them to report, update, and manage issues.
- ✓ **Data Integrity and Accuracy:** Assuming that the information entered into the system by users is accurate and valid, forming the basis for issue tracking and resolution.
- ✓ **System Stability:** Assuming that the hardware, software, and network infrastructure supporting the issue tracking system remain stable and operational, minimizing downtime or disruptions.
- ✓ **User Training and Familiarity:** Assuming that users have received adequate training or possess sufficient familiarity with the issue tracking platform to effectively use its features and functionalities.
- ✓ **Regular Backups and Recovery:** Assuming that regular backups of system data are conducted, enabling efficient recovery in case of data loss or system failure.

### Dependencies:

- ✓ **Internet Connectivity:** The system relies heavily on consistent internet connectivity for users to access the platform and perform various tasks related to issue tracking.

- ✓ **Hardware and Software Components:** Dependencies on servers, databases, operating systems, and issue tracking software functioning optimally to ensure seamless operations.
- ✓ **User Engagement and Input:** The system's effectiveness depends on users actively engaging with the platform, reporting issues accurately, providing updates, and collaborating for issue resolution.
- ✓ **Security Measures:** Dependencies on robust security protocols, firewalls, encryption methods, and regular security updates to safeguard sensitive data and prevent cyber threats.
- ✓ **Third-Party Integrations:** If the system integrates with other tools or platforms (e.g., version control systems, project management tools), dependencies exist on the proper functioning and compatibility of these integrations.

## 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

An issue tracking system's data requirements are fundamental elements necessary to comprehensively manage and resolve issues across projects or teams. At the core of this system lie crucial data components such as issue details, project information, user data, comments, attachments, and metadata, each playing a pivotal role in facilitating efficient issue tracking and resolution. These requirements encompass essential fields within the system, including unique identifiers for issues, projects, and users, along with descriptive information like issue titles, descriptions, statuses, and priorities. Key timestamps such as creation, update, and resolution dates offer a chronological view of issue lifecycles. User-related data, such as roles and assigned issues, aid in delegation and accountability.

Customization of these data elements can further tailor the system to meet specific organizational needs and workflows. Regular data management and utilization of these structured elements empower teams to address challenges promptly, ensuring a streamlined and effective issue resolution process.

### **3. External Interface Requirement**

Designing a Graphical User Interface (GUI) for an issue tracking system involves creating an intuitive and user-friendly interface that allows users to interact with the system efficiently. Here are key considerations and components typically included in a GUI for an issue tracking system:

#### **1.Dashboard Overview:**

- ✓ Issue Summary: A summary section displaying the total number of open issues, resolved issues, pending issues, etc.
- ✓ Project Overview: A section highlighting project-specific statistics, such as the number of issues per project or project statuses.

#### **2.Navigation Menu:**

- ✓ Projects: A section where users can view, select, and navigate between different projects.
- ✓ Issues: Tabs or sections to categorize and filter issues based on status, priority, assignee, etc.
- ✓ Reports: Access to generate and view reports, charts, or graphs illustrating issue metrics and trends.
- ✓ User Profile and Settings: Options for users to manage their profiles, preferences, and system settings.

#### **3.Issue Listing and Details:**

- ✓ Issue List: A table or list displaying issues with essential details like ID, title, status, assignee, priority, and creation/update dates.
- ✓ Issue Details View: Clickable items in the list to access detailed views of individual issues with descriptions, comments, attachments, and history logs.
- ✓ Search and Filter: Filters or search functionalities to easily find and narrow down specific issues based on criteria like status, priority, or keyword.

#### **4.Issue Creation and Editing:**

- ✓ New Issue Form: An interface allowing users to create new issues with fields for title, description, priority, and assignment.
- ✓ Edit Issue: Functionality to edit or update existing issues, change statuses, assignees, or other attributes.

#### **5.Collaboration and Communication:**

- ✓ Comments Section: A space for users to add comments, updates, or discussions related to specific issues.
- ✓ Attachment Upload: Ability to upload and attach files, screenshots, or documents to issues for additional context or documentation.

## **6.Reporting and Analytics:**

- ✓ Reports Dashboard: Graphs, charts, or visual representations depicting issue trends, resolution times, team performance, etc.
- ✓ Custom Report Generation: Options for users to generate custom reports based on specific criteria or metrics.

## **7.Notifications and Alerts:**

- ✓ Real-time Notifications: Alerts for new assignments, status changes, or mentions related to issues.
- ✓ Email Notifications: Optionally, email notifications to keep users informed about updates or changes to assigned issues.

## **8.Accessibility and Responsiveness:**

- ✓ Responsive Design: Ensuring the GUI is accessible and usable across various devices and screen sizes.
- ✓ Ease of Navigation: Intuitive navigation with clear labelling, consistent design elements, and logical workflows.

Designing the GUI for an issue tracking system should prioritize simplicity, clarity, and ease of use to empower users in efficiently managing, tracking, and resolving issues within projects or teams. Regular user testing and feedback integration can help refine and improve the interface to better meet user needs and preferences.

## **4.System Features**

Certainly, here are three key system features essential for an issue tracking system:

### **1.Customizable Workflow and Issue Fields:**

- Workflow Customization: The ability to define and customize issue workflows tailored to the organization's unique processes. This includes setting up statuses (e.g., Open, In Progress, Closed), transitions between statuses, and actions allowed at each stage.
- Custom Issue Fields: Providing flexibility to add custom fields to issues, allowing users to capture specific information relevant to their workflow or industry. These fields could include customer ID, product version, severity, or any other pertinent data points.

### **2.Collaboration and Communication Tools:**

- Commenting and Discussions: A feature allowing users to add comments, notes, or updates to issue threads, enabling real-time collaboration and information sharing among team members.
- Mentions and Notifications: Functionality that notifies relevant users when they are mentioned in comments or assigned to issues, ensuring timely responses and actions.

### **3.Reporting and Analytics Capabilities:**

- Reporting Dashboard: A centralized dashboard offering visual representations like charts, graphs, or summaries showcasing key metrics such as issue statuses, resolution times, backlog trends, and team performance.
- Custom Reports and Data Export: The capability to generate customized reports based on specific criteria or filters. Users should be able to export data for further analysis or reporting outside the system.

These features collectively enhance productivity, streamline communication, enable customization based on unique workflows, and provide valuable insights into issue resolution metrics for effective decision-making within the issue tracking system.

## **5. Other Non-Functional Requirements**

### **5.1 performance requirement**

Certainly, here are three essential performance requirements for an issue tracking system:

#### **1.Responsiveness and Speed:**

- Quick Response Times: The system should respond promptly to user actions, such as creating, updating, or retrieving issues, ensuring minimal latency.
- Fast Loading: Web interfaces and pages within the system should load swiftly, allowing users to access information and navigate seamlessly without noticeable delays.
- Efficient Query Processing: Database queries should be optimized to swiftly retrieve information, preventing performance bottlenecks and ensuring smooth user interactions.

#### **2.Scalability and Capacity:**

- Scalability: The system should be capable of handling increased user loads, data volumes, and concurrent transactions without degradation in performance.
- Capacity Planning: Adequate resources, including hardware and network infrastructure, should be provisioned to support current usage and accommodate future growth without impacting performance.
- Horizontal and Vertical Scaling: Implementing mechanisms for horizontal scaling (adding more servers) or vertical scaling (upgrading existing resources) to ensure continued performance as demands increase.

#### **3.Reliability and Uptime:**

- High Availability: The system should maintain a high level of availability, minimizing downtime or interruptions in service to ensure users can access it whenever necessary.
- Fault Tolerance and Redundancy: Implementing redundancy and failover mechanisms to handle system failures, ensuring continuity of service and minimal disruptions.
- Regular Monitoring and Maintenance: Continuous monitoring, proactive maintenance, and timely resolution of potential issues or performance bottlenecks to uphold system reliability and uptime.

Fulfilling these performance requirements ensures that the issue tracking system operates efficiently, handles increasing demands effectively, and delivers a responsive and reliable experience for users, ultimately contributing to improved productivity and streamlined issue management processes.

## **5.2 Safety Requirement**

Safety requirements for an issue tracking system primarily focus on safeguarding sensitive data, ensuring data integrity, and protecting the system against security threats.

Safety requirements for an issue tracking system pertain to the measures and protocols put in place to ensure the security, integrity, and confidentiality of data, as well as the system's availability and protection against potential threats. These requirements are essential to maintain a safe and secure environment for managing and tracking issues effectively.

## **5.3 Security Requirement**

### **1.Authentication and Access Control:**

- User Authentication: Implement strong authentication mechanisms like multi-factor authentication (MFA) to verify user identities before granting access to the system.
- Access Control: Enforce strict access controls based on roles and responsibilities (RBAC) to limit users' access to only the functionalities and data necessary for their tasks.

### **2.Data Encryption and Protection:**

- SSL/TLS) and data-at-rest within the system's databases to prevent unauthorized access to sensitive information. Data Encryption: Ensure end-to-end encryption for data transmission (using protocols like
- Sensitive Data Handling: Implement measures to protect sensitive data (such as personally identifiable information - PII) from unauthorized access or exposure.

### **3.Regular Security Updates and Patch Management:**

- Patch Management: Maintain an up-to-date system by applying security patches and updates to all software components regularly, including the issue tracking software, operating systems, and any third-party integrations, to address known vulnerabilities.

### **4.Monitoring and Incident Response:**

- Real-time Monitoring: Employ intrusion detection systems (IDS) and security monitoring tools to continuously monitor the system for any suspicious activities or unauthorized access attempts.
- Incident Response Plan: Have a well-defined incident response plan in place to promptly detect, respond to, and mitigate security incidents or breaches.

### **5.User Awareness and Training:**

- Security Training: Conduct regular security awareness programs and training sessions for users to educate them about security best practices, social engineering threats, and how to identify and report security incidents.

Implementing these security requirements helps fortify the issue tracking system against various threats, ensuring data protection, system integrity, and user confidentiality while maintaining compliance with relevant security standards and regulations. Regular security assessments, audits, and updates are essential to continually strengthen the system's security posture and mitigate emerging threats.

#### **5.4 Requirement Attributes**

Requirement attributes in an issue tracking system are key characteristics or properties assigned to each requirement or issue being managed. They include:

- 1. Unique Identifier (ID):** A distinct code or number assigned to each requirement for identification and tracking purposes.
- 2. Description:** Detailed information outlining the nature, purpose, and specifics of the requirement or issue.
- 3. Priority and Severity:** Priority levels indicating the importance and urgency of the requirement, while severity levels depict the impact of issues on the system.
- 4. Status:** The current stage or state of the requirement, tracking its progress from creation to completion or resolution.
- 4. Related Links/Dependencies:** Connections or dependencies with other items, showing relationships between different requirements or issues.
- 5. Assignee/Owner:** Designation of the person or team responsible for handling or resolving the requirement or issue.

#### **5.5 Business plan**

A business plan for an issue tracking system encompasses a strategic blueprint that delineates the vision, objectives, and operational framework for the creation, deployment, and sustenance of the system. It commences with an executive summary encapsulating the system's purpose, core features, and intended market. The plan navigates through a comprehensive business description elucidating how the issue tracking system meets distinct business needs and resolves challenges commonly encountered in project management and task tracking.

Ultimately, this comprehensive business plan serves as a foundational document, aligning strategies, resources, and objectives to propel the issue tracking system towards successful deployment, widespread adoption, and sustained growth within the market landscape.

#### **5.6 User Requirement**

User requirements for an issue tracking system represent the specific needs, expectations, and functionalities desired by the system's intended users. These requirements are pivotal in

shaping the system's design and usability. Users expect a seamless and intuitive interface that simplifies issue management.

## **6. Other Requirement**

### **6.1 Data and Category Requirement**

Data and category requirements for an issue tracking system encompass the foundational elements necessary to effectively categorize, organize, and manage diverse types of data related to issues within the system. This involves structuring and defining categories, fields, and metadata to capture, classify, and retrieve information efficiently. Firstly, defining the essential data categories entails establishing a taxonomy or classification system that categorizes issues based on attributes such as priority, status, severity, type, and project affiliation.

### **6.2 Appendix**

In the context of an issue tracking system, an appendix serves as a supplementary section containing additional information, supporting documents, or reference materials related to the system's documentation. This section offers extra details that complement the main content, aiding users, administrators, or stakeholders in better understanding the system's functionalities or operations. The appendix might include a glossary of terms, sample reports, technical diagrams, compliance information, user guides, or details about customizations and extensions.

### **6.3 Glossary**

#### **Alerts or messages Issue:**

- A task, problem, or item that requires attention or resolution within the project or system.

#### **Ticket:**

- An individual record or entry in the issue tracking system representing an issue, often containing details such as description, status, priority, and assigned personnel.

#### **Priority:**

- The level of urgency or importance assigned to an issue, indicating its relative significance compared to other issues.

#### **Status:**

- The current state or stage of an issue within its lifecycle, such as "Open," "In Progress," "Resolved," or "Closed."

#### **Severity:**

- The impact or seriousness of an issue on the system or project, ranging from low to critical.

#### **Workflow:**

- The defined sequence of steps or stages an issue passes through from creation to resolution.

**Assignment:**

- Allocation of an issue to a specific individual or team responsible for its resolution.
- Comments/Notes:
- Additional information, updates, or discussions attached to an issue, aiding in communication and collaboration.

**Attachment:**

- Files, documents, or images linked to an issue, providing supplementary information for its resolution.
- Custom Fields:
- Additional fields that can be tailored or added to capture specific information relevant to an issue or project.

**Audit Trail:**

- A record of all actions, changes, or modifications made to an issue, ensuring transparency and accountability.
- Notification:
- sent to users to inform them about changes, assignments, or updates related to issues they are involved in.

**6.4 class diagram**

A class diagram for an issue tracking system serves as a visual representation illustrating the structure and interactions among key entities within the system. It outlines the essential classes such as Issue, User, Comment, Attachment, Status, etc., along with their respective attributes and relationships. The Issue class typically encapsulates details like issue ID, title, description, status, priority, and associations with users, comments, and attachments. Users are represented with attributes like user ID, username, email, and roles, while comments and attachments link to specific issues. The Status class defines various states an issue can have, enabling effective tracking. Overall, the class diagram provides a blueprint for developers, offering a clear understanding of how different components relate and function within the issue tracking system's architecture.

