***Software Requirement***

***Specification***

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Title** |  |
| **1** | **ABSTRACT** |  |
| **2** | **MODULES** |  |
|  | 2.1 User Management Module |  |
|  | 2.2 Ticket Management Module |  |
|  | 2.3 Ticket Listing and Filtering Module |  |
|  | 2.4 Versioning Module |  |
| **3** | **FUNCTIONAL REQUIREMENTS** |  |
|  | 3.1 Ticket Creation and Editing |  |
|  | 3.2 Ticket Management System |  |
|  | 3.3 Search and Filtering |  |
|  | 3.4 Ticket Commenting System |  |
| **4** | **NON-FUNCTIONAL REQUIREMENTS**  4.1 Performance 4.2 Reliability4.3 Usability 4.4 Scalability  4.5 Security |  |
| **5** | **USE CASE DIAGRAM** |  |
| **6** | **CLASS DIAGRAM** |  |
| **7** | **SEQUENCE DIAGRAM** |  |
| **8** | **FLOW DIAGRAM** |  |
| **9**  **10** | **ER DIAGRAM**  **CONCLUSION** |  |

**1 ABSTRACT**

This project aims to develop a comprehensive Ticketing Application with advanced features for efficient workflow management. The primary objective is to create different types of tickets, each with a specific set of attributes and a defined workflow state. The application will support versioning to track changes made to tickets and provide the ability to view previous versions. Additionally, users will be able to add comments to tickets, and files (including images, txt, docx, xlsx, pptx) can be attached and previewed.

**2 MODULES**

**2.1 User Management Module**

* Responsible for user authentication, authorization, and management.
* Allows users to register, login, and manage their profiles.
* Defines user roles and permissions to control access to various features and data within the system.

**2.2 Ticket Management Module**

* Users are provided with interfaces to create new tickets of different types, including User Story, Bug, and Customer Support Ticket.
* Manages ticket properties such as ID, description, acceptance criteria, story points, environment, priority, responsible person, etc.
* Implements workflow states for each ticket type, allowing tickets to transition between states like New, Active, In Progress, Resolved, and Closed.

**2.3 Ticket Listing and Filtering Module**

* Provides functionality to list all tickets in the system.
* Supports filters to allow users to search and filter tickets based on various criteria such as type, status, assignee, etc.
* Allows for sorting tickets based on different attributes such as ID, priority, creation date, etc.

**2.4 Versioning Module**

* Implements versioning support for tickets to track changes over time.
* Increases the version number of a ticket each time it is modified, ensuring a complete history of revisions.
* Allows users to view previous versions of tickets along with their details, providing transparency and accountability.

**3 FUNCTIONAL REQUIREMENTS**

### **3.1 Ticket Creation and Editing**

* Users should be able to create new tickets.
* Different types of tickets (User Story, Bug, Customer Support) should be supported.
* Tickets must capture relevant details such as ID, description, acceptance criteria, etc.

**3.2 Ticket Management System**

* Module for ticket creation, editing, and assignment.
* Workflow management for ticket status tracking (Open, In Progress, Pending, Closed).
* Priority management for categorizing the urgency of tickets.

### **3.3 Search and Filtering**

* Implement search functionality for tickets.
* Allow users to apply filters to find specific tickets.

**3.4 Ticket Commenting System**

* Users should be able to add comments to tickets.
* View the history of each ticket.

**4 NON-FUNCTIONAL REQUIREMENTS**

### **4.1 Performance**

* The system should handle a minimum of 1000 concurrent users.
* Response time for common actions should be within 2 seconds.

### **4.2 Reliability**

* The system should have 99.9% uptime.
* Regular backups of the database should be performed.

### **4.3 Usability**

* The user interface should be intuitive and user-friendly.
* Provide online help documentation.

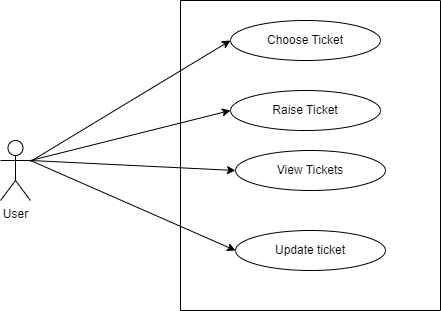
### **4.4 Scalability**

* The system should be scalable to accommodate future growth.
* Support additional features and modules.

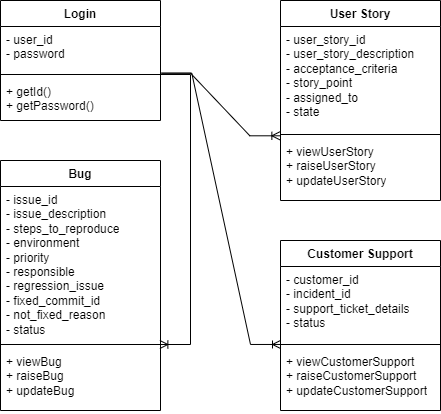
### **4.5 Security**

* All user data should be encrypted during transmission.
* Passwords should be stored securely using hashing algorithms.

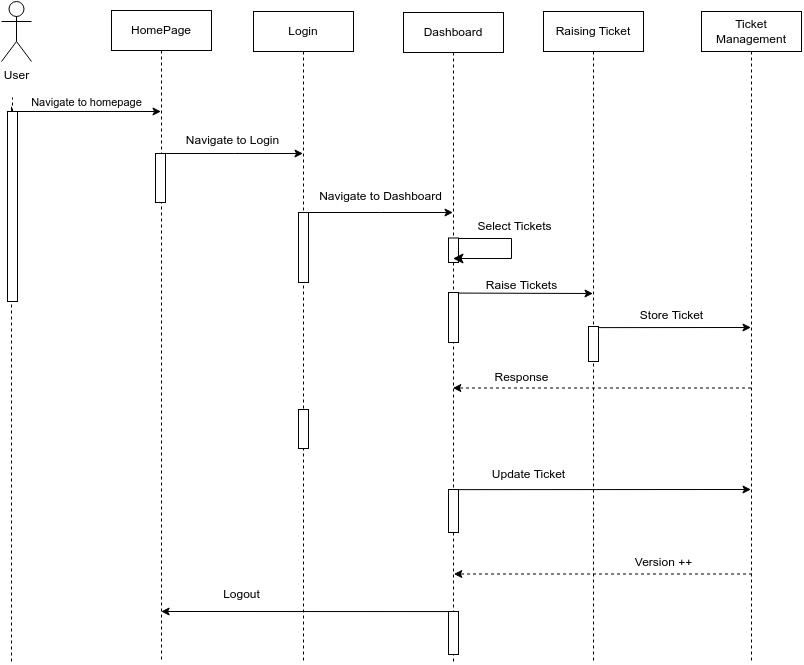
**5 USE CASE DIAGRAM**



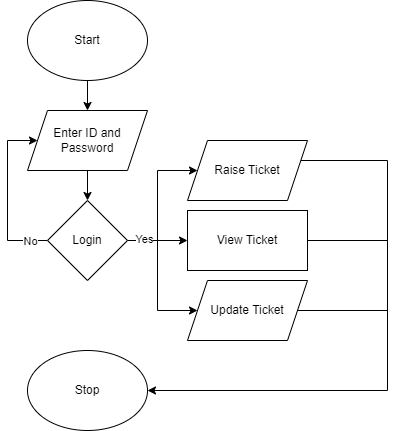
**6 CLASS DIAGRAM**



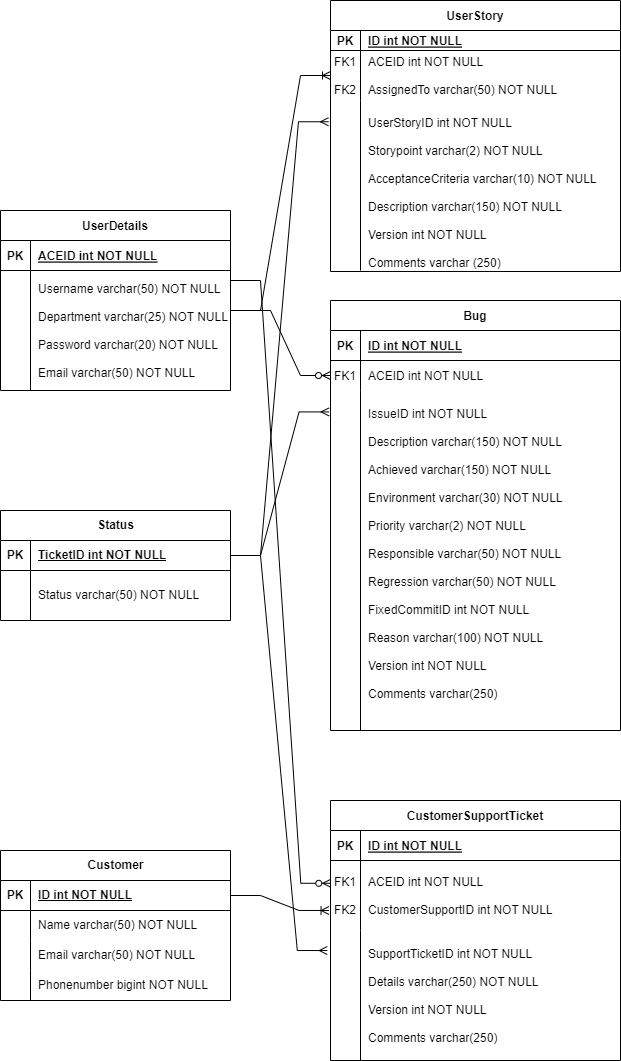
**7 SEQUENCE DIAGRAM**



**8 FLOW DIAGRAM**



**9 ER DIAGRAM**



**10 CONCLUSIONS**

In conclusion, the ticketing application needs to cater to the creation and management of different types of tickets, including User Story, Bug, and Customer Support Ticket, each with its specific attributes and workflow states. Key features such as listing and filtering tickets, allowing comments from users, supporting versioning, and enabling the viewing of previous ticket versions are essential for effective ticket management and collaboration. By implementing these requirements, the ticketing application can streamline workflow processes, enhance communication among stakeholders, and ensure the successful resolution of issues and tasks.