Project Name: I.T Support Ticketing System

Date: September 14, 2024

Document Version: 1.0

Prepared By: Mr. Brown, Thomas

# Introduction

The Non-Functional Requirements Document (NFRD) specifies the criteria by which the system’s performance, security, usability, and scalability are measured. Unlike functional requirements, these outline the quality attributes that ensure the IT Support Ticketing System performs efficiently and reliably for both customers and IT staff.

# Non-Functional Requirements

## Performance Requirements

* **Response Time**: All pages should load within 2 seconds under normal load conditions (up to 1000 concurrent users).
* **API Latency:** API responses should return within 200ms for standard requests (e.g., ticket retrieval, updates).
* **Real-Time Communication:** Real-time updates (e.g., ticket status changes) must be reflected on the user's dashboard within 2 seconds using WebSocket or Pusher.

## Scalability Requirements

* **Concurrent Users:** The system must support at least 10,000 concurrent users without significant degradation in performance.
* **Ticket Volume:** The system must be able to handle up to 1 million tickets in the database without significant impact on retrieval or search times.
* **Horizontal Scaling:** The cloud infrastructure should support horizontal scaling to accommodate growth in users, tickets, and real-time events.

## Availability Requirements

* **System Uptime:** The system should maintain at least 99.9% uptime, ensuring minimal downtime.
* **Backup and Recovery:** The system must implement automated daily backups of the database and ensure disaster recovery procedures allow restoration within 30 minutes in case of failure**.**

## Security Requirements

* **Authentication:** All users must authenticate securely using JWT-based token authentication. Passwords should be hashed using industry-standard algorithms (e.g., bcrypt).
* **Role-Based Access Control:** Data access must be limited based on user roles (e.g., Customer, IT Admin).
* **Data Encryption:** All sensitive data (passwords, tokens, etc.) must be encrypted both in transit (using HTTPS) and at rest.
* **Audit Trails:** All critical actions (e.g., ticket updates, status changes) must be logged and available for review to ensure traceability and accountability.

## Usability Requirements

* **User Interface (UI)**: The UI should be intuitive, following established design principles (e.g., Material Design or Ant Design standards). Key actions (e.g., ticket creation, updates) should be easily accessible with minimal clicks.
* **Accessibility**: The system should adhere to WCAG 2.1 guidelines for accessibility, ensuring it is usable by individuals with disabilities (e.g., keyboard navigation, screen reader compatibility).
* **Error Handling**: Clear, user-friendly error messages should be provided for common issues (e.g., network failures, invalid input).

## Maintainability Requirements

* **Code Modularity**: The system should be designed with modularity in mind (e.g., separating frontend and backend logic, using microservices where appropriate) to allow future updates and feature expansions.
* **Documentation**: Comprehensive developer and user documentation must be maintained, covering system architecture, APIs, and deployment processes.
* **Testability**: The system must be designed to facilitate automated testing, covering unit tests, integration tests, and end-to-end tests.

## Compliance Requirements

* **Data Protection**: The system must comply with relevant data protection regulations (e.g., GDPR) concerning user data collection, storage, and deletion.
* **Legal Compliance**: All financial transactions, if any, must comply with local financial regulations where the system is deployed.

# Monitoring and Reporting

## Monitoring

* **System Monitoring**: The cloud infrastructure must include monitoring for key metrics (e.g., CPU usage, memory, network bandwidth) to ensure that any performance issues are detected early.
* **Error Logging**: All critical errors must be logged and tracked in real time. System administrators should receive notifications for critical issues (e.g., system downtime, database failure).

## Reporting

* **Performance Reports**: Regular performance reports should be generated, highlighting key metrics like API response times, system uptime, and error rates.
* **Security Reports**: Regular security audits should be conducted, and reports should include details of vulnerabilities identified and actions taken to mitigate them.

# Assumptions

* The system will be deployed on a cloud platform (e.g., Cloudways) that supports scalability and redundancy.
* Users will access the system primarily via web browsers on both desktop and mobile devices.
* The system will experience high volumes of ticket traffic during peak hours, requiring efficient handling of real-time communication.

# Risks and Mitigation

|  |  |  |
| --- | --- | --- |
| RISK | IMPACT | MITIGATION STRATEGY |
| Performance Degradation | High: Impacts user experience | Optimize database queries and implement caching where applicable. |
| Security Breaches | High: Data loss or compromise | Regular security audits, encryption of sensitive data, and secure authentication. |
| System Downtime | High: Loss of productivity | Implement a redundant cloud infrastructure with automated failover. |

# Success Metrics

* System Uptime: Maintaining 99.9% uptime across all services.
* Response Time: All critical API calls should complete within 200ms.
* User Satisfaction: User feedback on system performance and usability should be positive, with minimal complaints regarding performance or downtime.

# Approvals

|  |  |  |
| --- | --- | --- |
| Name | Role | Signature |
| Mr. Thomas | Project Owner | [Signature] |
| IT Support Team | End-Users | [Signature] |
| Development Team | Implementation | [Signature] |

This document outlines the non-functional requirements that the IT Support Ticketing System must meet to ensure optimal performance, security, scalability, and usability.