**Technical Specification Document (TSD)**

[Amendment/Version History. 3](#_jx55zebhgqxr)

[Introduction 4](#_ras8rame2f9c)

[Document Conventions 5](#_mgkpzxkr8xgi)

[References 6](#_ogld19j87yqz)

[Product Requirements 6](#_et3i4cnzcbjn)

[User Overview 7](#_5006onlizpfc)

[Operating Environment 8](#_1aq4y59hxg1i)

[Implementation / Design Constraints 8](#_iwbe7bw0ibvy)

[Documentation 9](#_8onnrbj0124)

[Assumptions / Dependencies 9](#_ct80zcmw8435)

[System Features 9](#_7r0qd1jr8jn5)

[Descriptions and Priority 9](#_x38nkk90ohm8)

[Stimulus/Response/Sequences 10](#_nnduvwq5unbz)

[Functional Requirements 10](#_ll3qqpq9xfxb)

[External Interface Requirements 11](#_kiz1t7u8r4o1)

[User Interfaces 11](#_hjnc544wh8px)

[Hardware Interfaces 12](#_dwyo12ppjogt)

[Software Interfaces 12](#_g7xzczj91df5)

[Communication Interfaces 12](#_1j4lrqsya8h6)

[Non-functional Requirements 13](#_jm20bjrg4yq6)

[Performance 13](#_jie47xge25th)

[Safety 13](#_bsbrv4y5xhrl)

[Security 13](#_n4p3xnf52qiw)

[Software Quality 14](#_jjnyasnjqsmn)

[Appendices 14](#_de7y1aoaws2j)

[Glossary of Terms 14](#_n740vuwbqf7a)

# Amendment/Version History.

| Version | Date | Author | Description |
| --- | --- | --- | --- |
| 0.1 | 2024-07-10 | Rubaet Mohammed | Initial outline and structure of the Technical Specification Document (TSD). Included primary sections: Overview, Product Requirements, System Features, External Interface Requirements, Non-functional Requirements, and Appendices. |
| 0.2 | 2024-07-12 | Rubaet Mohammed | Added preliminary sections: Overview, Product Requirements, and System Features. Defined key terms and initial product features. |
| 0.3 | 2024-07-14 | Rubaet Mohammed | Incorporated feedback from stakeholders on the initial outline and added detailed descriptions for product requirements and system features. Addressed initial comments on functional needs. |
| 0.4 | 2024-07-16 | Rubaet Mohammed | Added External Interface Requirements and Non-functional Requirements sections. Detailed user interfaces and software integration points. |
| 0.5 | 2024-07-18 | Rubaet Mohammed | Revised System Features section with priority descriptions and functional requirements. Detailed functional requirements for ticket management and SLA monitoring. |
| 0.6 | 2024-07-20 | Rubaet Mohammed | Detailed the User Interfaces and Software Interfaces under External Interface Requirements. Added mockups and interface design guidelines. |
| 0.7 | 2024-07-22 | Rubaet Mohammed | Enhanced Non-functional Requirements with detailed specifications for performance, safety, security, and software quality. Included benchmarks and performance metrics. |
| 0.8 | 2024-07-24 | Rubaet Mohammed | Added Appendices including Glossary of Terms, Analysis Documentation, and Issues sections. Linked related project documents for reference. |
| 0.9 | 2024-07-26 | Rubaet Mohammed | Integrated feedback from internal review and adjusted various sections for clarity and completeness. Revised based on review comments from the technical team. |
| 1 | 2024-07-28 | Rubaet Mohammed | Final draft ready for approval, including clickable Table of Contents and formatted document for ease of navigation. Conducted final review for completeness. |
| 1.1 | 2024-08-02 | Rubaet Mohammed | Updated with additional requirements based on the latest project scope changes. Included new sections on implementation constraints and detailed assumptions. |
| 1.2 | 2024-09-10 | Rubaet Mohammed | Final review, adjustments, and preparations for the implementation phase. Ensured alignment with project goals and updated to reflect the latest stakeholder feedback. |

# Introduction

The purpose of this document is to provide a comprehensive and detailed technical specification for the implementation of the FreshDesk ticketing system. This document serves as a foundational blueprint for the development team, project managers, stakeholders, and end-users, ensuring that every aspect of the system is meticulously planned and documented.

The primary goals of this Technical Specification Document (TSD) are as follows:

* **Define System Features**: Clearly outline the functionalities and capabilities of the FreshDesk ticketing system, detailing how it will handle ticket creation, management, tracking, and resolution. This includes defining user roles, access levels, and specific features such as SLA management, reporting, and analytics.
* **Establish Requirements**: Specify the functional and non-functional requirements of the system. Functional requirements will cover what the system should do, including detailed use cases and user stories. Non-functional requirements will address how the system performs, including aspects such as performance metrics, security standards, and compliance requirements.
* **Describe Interfaces**: Provide detailed descriptions of the external interfaces the FreshDesk system will interact with, including user interfaces, hardware interfaces, software interfaces, and communication interfaces. This section ensures that all integration points are well-documented, facilitating seamless interaction with other systems and tools.
* **Ensure Consistency and Clarity**: Use standardised document conventions to maintain consistency throughout the document. This includes bold headings for sections, italics for emphasis, and code formatting for technical configurations. Consistent documentation helps in avoiding misunderstandings and ensures that all stakeholders are on the same page.
* **Facilitate Development and Deployment**: By providing a detailed roadmap and specifications, this document aims to streamline the development and deployment process. Clear guidelines and requirements reduce ambiguities, minimise development time, and help in avoiding potential integration issues.
* **Support Stakeholder Communication**: Serve as a communication tool among various stakeholders, including developers, testers, project managers, and end-users. It ensures that everyone involved has a clear understanding of the system’s objectives, functionalities, and constraints, promoting better collaboration and coordination.
* **Guide Future Enhancements**: Lay the groundwork for future enhancements and iterations of the system. By documenting the initial implementation in detail, this TSD provides a reference point for ongoing improvements and scalability, ensuring the system can evolve to meet future needs.

In essence, this Technical Specification Document is crucial for the successful implementation of the FreshDesk ticketing system. It ensures that all aspects of the system are carefully considered, documented, and communicated, paving the way for a smooth development and deployment process that aligns with the project’s goals and stakeholder expectations.

# Document Conventions

This document uses the following conventions to ensure clarity, consistency, and ease of understanding:

* **Bold**: Bold text is used for sections, headings, and key terms that require emphasis. This helps in quickly identifying major parts of the document and important terminology.
* *Italics*: Italics are used for emphasis on specific words or phrases that need special attention or to highlight examples, references, or quotes.
* Code: Monospaced font is used for code snippets, technical configurations, and command-line instructions. This format helps in distinguishing technical content from regular text.
* **Numbered Lists**: Numbered lists are used to present steps or sequences of actions that need to be followed in a specific order.
* **Bullet Points**: Bullet points are used for listing items that do not require a specific order, making it easy to read and understand the content.
* **Tables**: Tables are used to organise and present data in a structured format, making it easier to compare and analyse information.

# References

This document should be read in conjunction with the following reference materials to provide comprehensive context and background information:

* **Project Scope Document**: Outlines the overall scope, objectives, and deliverables of the project.
* **Business Requirements Document**: Details the business needs and requirements that the FreshDesk ticketing system aims to address.
* **FreshDesk Official Documentation**: Provides official guidelines, technical details, and support documentation from FreshDesk.

# Product Requirements

**Features**

The FreshDesk ticketing system will include the following key features to ensure efficient and effective handling of support requests:

* **Ticket Management**:
  + Automated creation, assignment, and tracking of support tickets.
  + Ensures all customer issues and requests are logged systematically and managed efficiently.
* **SLA Management**:
  + Define and monitor Service Level Agreements (SLAs) to ensure timely resolution of tickets.
  + Automated alerts and escalations for tickets approaching or exceeding SLA limits.
* **Reporting and Analytics**:
  + Generate performance reports and analytics to provide insights into support operations.
  + Helps in identifying trends, measuring efficiency, and making data-driven decisions for continuous improvement.
* **Integration**:
  + Seamlessly integrate with existing communication channels such as email and chat.
  + Ensures that all customer interactions are captured and managed within the ticketing system, providing a unified view of support activities.

# User Overview

The FreshDesk ticketing system will be used by a variety of users, each with distinct roles and responsibilities:

* **Support Staff (Level 1, 2, and 3)**:
  + **Level 1 Support Staff**: These are the frontline support personnel who handle initial customer queries and perform basic troubleshooting. They log tickets, provide initial responses, and escalate more complex issues to higher levels of support. They require a user-friendly interface for quick access to ticket information and resolution workflows.
  + **Level 2 Support Staff**: These support agents handle escalated issues that require more in-depth technical knowledge. They perform detailed troubleshooting and resolution tasks, document solutions, and may sometimes provide guidance to Level 1 staff. They need robust tools for in-depth diagnostics and issue tracking.
  + **Level 3 Support Staff**: These are subject matter experts who manage the most complex and critical issues. They provide expert solutions, perform system configurations, and ensure that the issues are resolved permanently. They may also contribute to the knowledge base by documenting solutions to complex problems. They need access to advanced tools and system configurations.
* **System Administrators**: System administrators are responsible for the overall maintenance and configuration of the FreshDesk system. They manage user accounts, set permissions, ensure system security, perform regular updates, and handle system integrations. They require access to system settings, user management tools, and monitoring dashboards.
* **End-users (Clients and Internal Employees)**:
  + **Clients**: External customers who use the system to submit support tickets for issues or service requests. They require a simple and intuitive interface to log tickets, track their status, and receive updates. Clients expect timely responses and clear communication regarding their issues.
  + **Internal Employees**: Employees within the company who use the system to report technical issues, request IT services, and track the progress of their requests. They need an efficient way to communicate with the support team and get timely resolutions. The system should integrate seamlessly with their daily workflow tools.

# Operating Environment

The FreshDesk ticketing system will operate in the following environment:

* **Web-based Application**: The system will be accessible via modern web browsers such as Chrome, Firefox, and Edge. This ensures cross-platform compatibility and ease of access from various devices, including desktops, laptops, and tablets.
* **Cloud Infrastructure**: The system will be hosted on a secure and scalable cloud platform. This setup provides flexibility, scalability, and reliability, ensuring high availability and performance. It also allows for easy scaling of resources as the demand increases.

# Implementation / Design Constraints

The design and implementation of the FreshDesk ticketing system will be subject to the following constraints:

* **Compliance**: The system must comply with General Data Protection Regulation (GDPR) and other relevant data protection regulations to ensure user data is handled securely and legally. This includes implementing robust data encryption, access controls, and regular audits.
* **Integration**: The system must integrate seamlessly with existing email and chat systems to provide a unified communication platform for support activities. This ensures that all interactions are captured and managed within the ticketing system, providing a comprehensive view of support activities.
* **Budget and Timeline**: The project is subject to limited budget and strict timeline constraints, necessitating efficient use of resources and adherence to project schedules to ensure timely delivery within the allocated budget. Prioritisation of features and phased implementation may be required to meet these constraints.

# Documentation

Comprehensive documentation will be provided to ensure all users can effectively utilise the FreshDesk system:

* **User Manuals**: Detailed guides for end-users and support staff on how to use the system, including step-by-step instructions and troubleshooting tips. These manuals will cover all functionalities and provide scenarios for common tasks.
* **Training Materials**: Resources such as videos, tutorials, and FAQs to facilitate user training and onboarding. These materials will help new users get up to speed quickly and ensure consistent use of the system across the organisation.
* **Technical Support Documentation**: In-depth technical guides for system administrators and support staff, covering system configuration, maintenance, and troubleshooting procedures. This documentation will also include best practices for system management and integration.

# Assumptions / Dependencies

The successful implementation and operation of the FreshDesk system depend on the following assumptions and dependencies:

* **Basic Computer Literacy**: Users are assumed to have basic computer literacy, enabling them to navigate the web-based system and perform necessary tasks without extensive training.
* **Reliable Internet Connectivity**: Stable and reliable internet connectivity is essential for accessing the cloud-hosted system. Any interruptions in connectivity could affect system performance and user experience.
* **Timely Feedback from Stakeholders**: Regular and prompt feedback from stakeholders is necessary to address any issues and ensure the system meets user needs and expectations. This includes feedback during the development, testing, and post-implementation phases.

# System Features

# Descriptions and Priority

The system features are categorised based on their priority to ensure essential functionalities are implemented first:

* **High Priority**:
  + **Ticket Management**: Automated creation, assignment, and tracking of support tickets to streamline the support process and ensure no issues are overlooked. This feature includes categorising tickets, setting priorities, and tracking their status from creation to resolution.
  + **SLA Management**: Define and monitor Service Level Agreements (SLAs) to ensure timely resolution of tickets. Automated alerts and escalations for tickets approaching or exceeding SLA limits ensure compliance and customer satisfaction.
* **Medium Priority**:
  + **Reporting and Analytics**: Generate performance reports and analytics to provide insights into support operations, helping to identify trends, measure efficiency, and make data-driven decisions for continuous improvement. This includes dashboards, customisable reports, and key performance indicators (KPIs).
* **Low Priority**:
  + **Additional Integrations**: Seamless integration with other tools and platforms as needed, such as project management systems, CRM systems, or other communication tools. These integrations enhance the system's functionality and improve workflow efficiency.

# Stimulus/Response/Sequences

The system will handle various stimuli and responses to ensure efficient operation and user satisfaction:

1. **User submits a ticket** -> **System creates and assigns the ticket** -> **User receives confirmation**: Ensures users are informed that their issue is logged and being addressed. This process should be seamless and provide immediate feedback to the user.
2. **SLA breach imminent** -> **System sends alert** -> **Support staff takes action**: Ensures timely intervention to meet SLA requirements and resolve issues promptly. Automated alerts help prevent SLA breaches and maintain high service standards.

# Functional Requirements

The system must fulfil the following functional requirements to ensure comprehensive and efficient management of support tickets:

* **Automated Ticket Creation from Emails**: The system should automatically create tickets from incoming emails, ensuring all customer interactions are logged and tracked. This feature minimises manual entry and reduces the risk of missed tickets.
* **Real-time Tracking and Status Updates**: Provide real-time updates on the status of tickets, allowing users and support staff to track progress and ensure transparency. This includes status notifications, updates on actions taken, and expected resolution times.
* **SLA Monitoring and Alerts**: Monitor SLAs and send automated alerts for tickets that are approaching or exceeding SLA limits, ensuring timely resolution and compliance. This feature helps maintain high service quality and customer satisfaction.

# External Interface Requirements

# User Interfaces

The FreshDesk ticketing system will feature user interfaces designed to be intuitive and user-friendly, ensuring ease of use for all types of users:

* **Intuitive Web Interface with Dashboard for Ticket Management**:
  + The web interface will provide a comprehensive dashboard that displays all relevant information about ticket status, priority, and assignment. The dashboard will allow users to filter and sort tickets, view ticket details, and perform necessary actions such as updating status, adding comments, and reassigning tickets.
  + The design will be clean and clutter-free, making it easy for users to navigate through different sections and find the information they need quickly. Icons and colour coding will be used to indicate ticket status and priority.
  + Contextual help and tooltips will be available to guide users and provide quick assistance for common tasks and features.
* **Mobile-friendly Design**:
  + The system will be designed to be fully responsive, ensuring optimal functionality and appearance on mobile devices such as smartphones and tablets. This allows users to access the system and manage tickets on-the-go.
  + Key features such as ticket creation, updates, and notifications will be easily accessible from mobile devices, ensuring users can stay productive regardless of their location.

# Hardware Interfaces

* **Standard Computing Devices**:
  + There are no specific hardware requirements beyond standard computing devices such as desktops, laptops, tablets, and smartphones. The system is designed to be platform-agnostic, ensuring compatibility with various operating systems and devices.

# Software Interfaces

* **Integration with Email Servers (e.g., SMTP, IMAP)**:
  + The system will integrate with existing email servers to enable automated ticket creation from incoming emails. This ensures that all customer emails are captured as support tickets, reducing the risk of missed issues.
  + Both Simple Mail Transfer Protocol (SMTP) and Internet Message Access Protocol (IMAP) will be supported to accommodate different email server configurations.
* **APIs for Integration with Chat Systems and Other Tools**:
  + Application Programming Interfaces (APIs) will be provided to facilitate integration with chat systems and other tools. This allows for real-time communication between support staff and users, improving the speed and quality of support.
  + The APIs will be well-documented, with examples and use cases to assist developers in integrating third-party applications seamlessly.

# Communication Interfaces

* **HTTPS for Secure Communication**:
  + All communication between users and the system will be secured using Hypertext Transfer Protocol Secure (HTTPS). This ensures that data transmitted over the network is encrypted, protecting it from eavesdropping and tampering.
  + The system will use industry-standard encryption protocols to ensure data security and compliance with regulations such as GDPR.
* **WebSocket for Real-time Updates**:
  + The system will use WebSocket technology to provide real-time updates on ticket status and other notifications. This ensures that users receive immediate feedback and can respond quickly to changes.
  + Real-time updates will enhance the user experience by providing timely information without the need for manual refreshes or polling.

# Non-functional Requirements

# Performance

* **High Throughput**:
  + The system should be capable of handling up to 10,000 tickets per day without degradation in performance. This includes processing ticket creation, updates, and queries efficiently.
* **Low Latency**:
  + Response time for ticket operations should be under 2 seconds to ensure a smooth user experience. This applies to actions such as ticket creation, status updates, and retrieval of ticket information.

# Safety

* **Data Backup and Recovery Mechanisms**:
  + Regular data backups will be performed to ensure that all ticket information is securely stored and can be recovered in the event of data loss. Backup schedules and retention policies will be established to meet organisational needs.
* **Regular Security Audits**:
  + Periodic security audits will be conducted to identify and address potential vulnerabilities in the system. This includes both internal reviews and external penetration testing.

# Security

* **Role-based Access Control**:
  + The system will implement role-based access control (RBAC) to ensure that users have access only to the features and data necessary for their role. This helps to minimise the risk of unauthorised access and data breaches.
* **Data Encryption at Rest and in Transit**:
  + Data will be encrypted both at rest and in transit to protect sensitive information from being accessed by unauthorised parties. This includes using strong encryption algorithms and secure key management practices.

# Software Quality

* **Unit and Integration Testing**:
  + Comprehensive unit and integration tests will be conducted to ensure that all components of the system function correctly and integrate seamlessly. Automated testing tools will be used to run tests continuously during the development process.
* **User Acceptance Testing**:
  + User acceptance testing (UAT) will be performed to validate that the system meets the needs and expectations of end-users. This involves real users testing the system in a production-like environment and providing feedback on its functionality and usability.

# Appendices

# Glossary of Terms

* **CRM**: Customer Relationship Management
* **SLA**: Service Level Agreement
* **API**: Application Programming Interface

**Analysis Documentation**

* **Project Scope Document**: Comprehensive outline of the project’s objectives, deliverables, and constraints.
* **Project Charter**: Document defining the project’s vision, objectives, stakeholders, and the framework for execution.
* **Process Flow**: Visual diagrams and descriptions of the processes involved in ticket management and support workflows.
* **Data Mapping**: Detailed mapping of data fields and structures between the FreshDesk system and integrated tools.

**Issues**

* **Current Identified Issues**:
  + None identified at this stage.
* **Future Issue Tracking**:
  + This section will be updated as issues are identified and tracked during the implementation and operation phases.