***Online IT Service Desk System***

***Software Requirements Specification***

***(SRS) Document***

*Alfie B. Solis*  
*11/25/2024*

|  |
| --- |
| **Revisions** |

| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
| Initial Draft | Alfie B. Solis | All sections being Filled | 11/25/2024 |

|  |
| --- |
| **Review & Approval** |

**Requirements Document Approval History**

| **Approving Party** | **Version Approved** | **Signature** | **Date** |
| --- | --- | --- | --- |
| **Ria Mae H. Borromeo, PhD** |  |  |  |

**Requirements Document Review History**

| **Reviewer** | **Version Reviewed** | **Signature** | **Date** |
| --- | --- | --- | --- |
| **Ria Mae H. Borromeo, PhD** |  |  |  |

|  |
| --- |
| **Contents** |

[1. Introduction 4](#_Toc183413221)

[1.1 Introduction 4](#_Toc183413222)

[1.2 Scope of this Document 4](#_Toc183413223)

[1.3 Overview 4](#_Toc183413224)

[1.4 Business Context 4](#_Toc183413225)

[2. General Description 4](#_Toc183413226)

[2.1 Product Functions 4](#_Toc183413227)

[2.2 Similar System Information 4](#_Toc183413228)

[2.3 User Characteristics 4](#_Toc183413229)

[2.4 User Problem Statement 5](#_Toc183413230)

[2.5 User Objectives 5](#_Toc183413231)

[2.6 General Constraints 5](#_Toc183413232)

[3. Functional Requirements 5](#_Toc183413233)

[3.1 User Management 5](#_Toc183413234)

[3.2 Ticket Management 5](#_Toc183413235)

[3.3 Support Team Functionalities 5](#_Toc183413236)

[3.4 Knowledge Base Integration 6](#_Toc183413237)

[3.5 Reporting 6](#_Toc183413238)

[3.6 Administrator Function 6](#_Toc183413239)

[3.7 Search Function 6](#_Toc183413240)

[4. Interface Requirements 6](#_Toc183413241)

[4.1 User Interfaces 6](#_Toc183413242)

[4.2 Hardware Interfaces 7](#_Toc183413243)

[4.3 Communications Interfaces 7](#_Toc183413244)

[4.4 Software Interfaces 7](#_Toc183413245)

[5. Performance Requirements 7](#_Toc183413246)

[5.1 Response Time 7](#_Toc183413247)

[5.2 Throughput 7](#_Toc183413248)

[5.3 Scalability 7](#_Toc183413249)

[6. Other non-functional attributes 7](#_Toc183413250)

[6.1 Security 7](#_Toc183413251)

[6.2 Binary Compatibility 8](#_Toc183413252)

[6.3 Reliability 8](#_Toc183413253)

[6.4 Maintainability 8](#_Toc183413254)

[6.5 Portability 8](#_Toc183413255)

[6.6 Extensibility 8](#_Toc183413256)

[6.7 Reusability 8](#_Toc183413257)

[6.8 Application Affinity/Compatibility 8](#_Toc183413258)

[6.9 Resource Utilization 8](#_Toc183413259)

[6.10 Serviceability 8](#_Toc183413260)

[7. Operational Scenarios 9](#_Toc183413261)

[Scenario A: User Access 9](#_Toc183413262)

[Scenario B: System Settings and Maintenance Fields 9](#_Toc183413263)

[Scenario C: Ticket Creation 9](#_Toc183413264)

[Scenario D: Responding to Ticket 9](#_Toc183413265)

[Scenario E: Knowledge base 9](#_Toc183413266)

[Scenario F: Reports 9](#_Toc183413267)

[8. Preliminary Use Case Models and Sequence Diagrams 9](#_Toc183413268)

[8.1 Use Case Model 9](#_Toc183413269)

[8.2 Sequence Diagrams 10](#_Toc183413270)

[9. Updated Schedule 11](#_Toc183413271)

[10. Updated Budget 12](#_Toc183413272)

[11. Appendices 12](#_Toc183413273)

[11.1 Definitions, Acronyms, Abbreviations 12](#_Toc183413274)

## 1. Introduction

### 1.1 Introduction

The purpose of this document is to outline and detail the project’s requirements, specifying the system's functionality and any related limitations.

### 1.2 Scope of this Document

The software will be used by Western Caroline Trading Company (WCTC) employees, with the IT department staff serving as administrators. The functionalities, performance expectations, and non-functional attributes of the software will be explained in this paper.

### 1.3 Overview

A web information system that allows the users to submit IT support requests, such as technical issues, troubleshooting, software development, and enhancement tasks. Users can easily check the status of their support requests, while administrators can generate reports for insights and performance tracking. Additionally, the system provides a knowledge base and FAQs, allowing users to find answers to common questions independently.

### 1.4 Business Context

WCTC, founded in 1948, has evolved into a diverse enterprise with operations spanning multiple industries. Its business portfolio includes retail, office supply stores, hardware outlets, wholesale services, car sales and rentals, a brewery, a travel agency, and a chain of hotels. Given the breadth of its operations, the IT department is vital in managing the technological infrastructure and providing support across all these sectors, ensuring smooth daily operations, maintaining systems, and addressing any technical issues.

## 2. General Description

### 2.1 Product Functions

The platform enables users to create support requests and track their progress in real-time, minimizing the chances of miscommunication, delays, and unresolved issues. Additionally, users can access knowledge base resources for self-service support, empowering them to resolve common problems independently.

### 2.2 Similar System Information

There are several online service desk software options available such as HelpDesk, Zoho Desk, Zamad and HappyFox, however, they operate on a subscription model. Costs vary based on the number of users and included features, which can add up over time for larger teams or complex needs.

For organizations with a large workforce, such as WCTC with nearly a thousand employees, the recurring costs of these subscriptions can become prohibitively expensive over time. Each additional user increases the subscription fee, and scaling up to accommodate growing needs or implementing advanced features can drive costs even higher. As a result, adopting these commercial software solutions may not be a viable option for companies looking to manage their budgets efficiently while meeting the complex IT support requirements of a large team.

### 2.3 User Characteristics

The users will be current employees of WCTC, with no special requirements beyond basic computer operation skills.

### 2.4 User Problem Statement

Currently, users communicate their concerns through phone, email, or in person. While these methods provide immediate access to IT support, they often lead to inefficiencies and unresolved issues. Miscommunication can arise from incomplete or unclear information being conveyed during phone or in-person interactions. Similarly, emails can get lost in cluttered inboxes, or critical details may be overlooked during high volume of correspondence.

Additionally, without a centralized system to log and track requests, the IT support team may unintentionally overlook or deprioritize certain issues, especially during busy periods or when dealing with multiple simultaneous problems. This lack of visibility into the status of requests creates frustration for users, who may feel ignored or uncertain about the resolution timeline. The absence of a structured system also makes it challenging for IT teams to prioritize and manage tasks effectively, leading to delays and reduced overall efficiency. These shortcomings highlight the need for a more streamlined and transparent solution to improve communication, accountability, and resolution times.

### 2.5 User Objectives

The user needs a centralized platform to manage a wide range of IT tasks, including resolving technical issues, processing software enhancement requests, fixing bugs, and developing new applications.

### 2.6 General Constraints

The software is a web application that is compatible with any web browser; however, machines should be connected to the internet when accessing outside local area network.

## 3. Functional Requirements

### User Management

1. Users should be able to create accounts using their employee ID, last name and email address. The system must validate this information in the database.
2. Users should login using their credentials, and the system should authenticate with the database.
3. Can define user roles such as end-user, support team, and administrator. With permission to access specific functions.

### Ticket Management

1. Users should be able to submit tickets that detail their issues and requests. They can choose categories such as support or software requests.
2. The system should automatically assign tickets to an appropriate support team based on the predefined categories and priorities.
3. The user can assign ticket priority to low, medium and high.
4. The user should be able to track ticket status in real-time with open, in progress, on-hold, closed and canceled.

### 3.3 Support Team Functionalities

1. The support team should have access to dashboards of open tickets, sorted by priority and category.
2. Support staff should be able to view ticket details, provide updates, add comments, and mark tickets as resolved.
3. Support staff or administrators should be able to reassign tickets if necessary.

### 3.4 Knowledge Base Integration

1. Users and support staff should be able to access a knowledge base with articles and FAQs to help resolve common issues.
2. The system should suggest relevant articles based on keywords in tickets.
3. Authorized users should be able to create, edit, and delete articles.

### 3.5 Reporting

1. The system should allow administrators to generate reports on ticket volumes, resolution times, and support team performance.
2. Track usage statistics such as the number of tickets created, resolved, and average response times.

### 3.6 Administrator Function

1. Administrators should be able to add, modify, or deactivate users and update user roles and classifications.
2. Administrators can configure system settings, including ticket categories, user roles, and priority levels.

### 3.7 Search Function

1. Users should be able to search for tickets by keywords, categories, and status.
2. Filter tickets by priority, category, status, and assigned support staff.

## 4. Interface Requirements

### 4.1 User Interfaces

* **4.1.1 GUI**

This program will utilize a combination of front-end frameworks and libraries, namely Bootstrap, JavaScript, and Web Components, to create a responsive, user-friendly interface. Bootstrap will help establish a cohesive layout across different devices with its responsive grid and pre-designed components. JavaScript will add dynamic interactions to the interface, ensuring smooth, efficient user interactions. Additionally, Web Components will allow for the creation of custom, reusable elements, fostering modularity and simplifying future updates to the interface.

* **4.1.2 CLI**

Command line interface will be utilized by system administrator or software developer for tasks such as installing and updating software package as well as performing troubleshooting.

* **4.1.3 API**

RESTful API and Laravel Eloquent will facilitate communication between front-end and back-end systems using HTTP methods like GET, PUT, POST and DELETE.

* **4.1.4 Diagnostics or ROM**

Device troubleshooting, monitoring, and configuration management are typically provided by the machine vendor. However, this program will include a knowledge base that covers these areas, offering users quick access to troubleshooting guides and configuration tips directly within the service desk program.

### 4.2 Hardware Interfaces

1. Ethernet ports – networking between server and clients
2. Redundant Power – UPS system for power backup
3. PCIe – Network Interface Cards
4. SATA/ SAS/ NVMe – provide connection to hard drives
5. USB ports – for managing backup
6. KVM over IP – allow remote access to server

### 4.3 Communications Interfaces

1. HTTPS – secure communication with SSL/TSL encryption between clients and web application.
2. RESTful APIs – asynchronous communication between the client and the server.
3. SSL/ TSL encryption – ensure secure communication between the client and the server by encrypting HTTP traffic.

### 4.4 Software Interfaces

1. RESTful APIs – Laravel API routes and controllers, define and handle API routes and CRUD operations
2. SQL Database – Eloquent ORM interact with SQL databases.

## 5. Performance Requirements

### 5.1 Response Time

1. Ideal – The system should process user requests in under 500 milliseconds for most interactions such as ticket submission or data retrieval.
2. Acceptable – For more complex operations, such as generating reports or analytics, response times can extend up to 5 seconds.
3. Unacceptable – Any operation exceeding 5 seconds should be optimized, as it may negatively impact the user experience.

### 5.2 Throughput

1. The system must handle 50–100 transactions per second (TPS) during peak load periods.
2. The system should maintain consistent performance under sustained load, with the ability to handle burst traffic efficiently.

### 5.3 Scalability

1. Vertical Scaling – The system must support increases in hardware capacity, such as upgrading CPU, RAM, or storage on individual servers, to accommodate higher loads.
2. Horizontal Scaling - The system should allow adding additional servers or nodes to distribute the workload effectively and support for load balancing.

## 6. Other non-functional attributes

### 6.1 Security

The system will require login credentials with username and a securely encrypted password to prevent unauthorized access. This approach ensures that only authorized personnel can access system resources, enhancing security. No sensitive data is stored within the service desk system; however, if sensitive information is added in the future, it will be encrypted to maintain data security and confidentiality.

### 6.2 Binary Compatibility

This system will be compatible with any browser installed on their computer.

### 6.3 Reliability

To maintain continuous support and prevent service disruptions, the system will incorporate regular backups, real-time monitoring, and scalability features to handle increased user requests. These measures ensure data recovery in case of failures, allow prompt detection of performance issues, and enable the system to adapt seamlessly to growing demands, ultimately enhancing reliability and availability for users.

### 6.4 Maintainability

The system will follow a modular design, enabling specific components to be updated or modified independently, reducing the risk of impacting the entire system. Detailed documentation covering code, configurations, and processes will support IT staff in troubleshooting, maintaining, and improving the system over time. Regular software and security updates will ensure compatibility and system security, while regular testing will catch potential issues early, minimizing the risk of new bugs.

### 6.5 Portability

The system will be designed to function across various operating systems, browsers, and devices, including desktops, and laptops.

### 6.6 Extensibility

The system will be designed with a flexible architecture, allowing new features and functions to be added without disrupting core components. This will include modular code and APIs, making it easier to integrate with other software tools or accommodate future changes in requirements.

### 6.7 Reusability

The system will be designed with reusable components and code libraries that can be leveraged across different parts of the system or future applications. Modular components will be written with standardized functions, allowing sections of the code, such as authentication, ticket processing, and reporting, to be reused in other company systems, reducing development time and ensuring consistency.

### 6.8 Application Affinity/Compatibility

This system requires any browser and should be connected to LAN or internet.

### 6.9 Resource Utilization

The system should operate efficiently without overloading the CPU, maintaining responsiveness and minimizing resource contention. It should effectively utilize memory to manage large datasets and optimize disk I/O for seamless performance. Additionally, the software should minimize bandwidth usage to ensure efficient network communication.

### 6.10 Serviceability

The system allows the administrator to handle maintenance tasks efficiently through an intuitive, user-friendly interface, enabling them to perform updates and manage configurations.

## 7. Operational Scenarios

### Scenario A: User Access

The user shall register using employee number, last name, and email address. The system will check if the employee number and last name matched with the records from the payroll system.

### Scenario B: System Settings and Maintenance Fields

The system administrator will classify users as either support staff or regular users, determine their support status (such as pending, open, on-hold, closed, or cancelled), and create or assign support team groups accordingly.

### Scenario C: Ticket Creation

Users can submit support tickets and track their status within the system. The support group will be automatically assigned based on the classification of the support.

### Scenario D: Responding to Ticket

The support team can review pending tickets and reassign them to other staff or teams as necessary. They have the capability to open tickets and respond to user inquiries.

### Scenario E: Knowledge base

The user can access knowledge base for self-service support.

### Scenario F: Reports

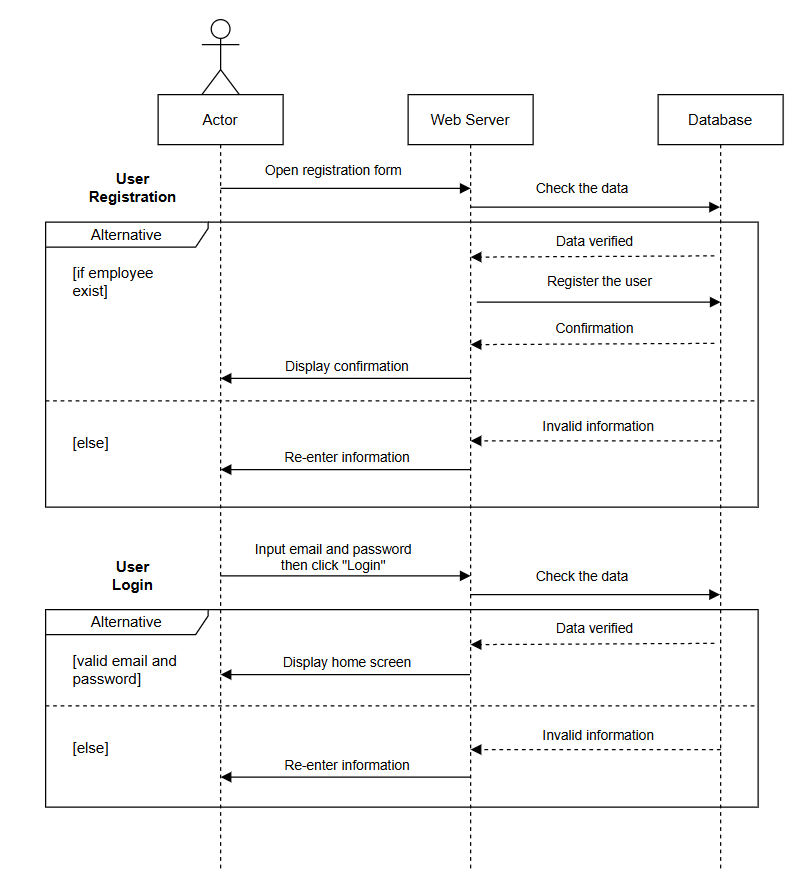
The user can generate reports based on the selected parameters.

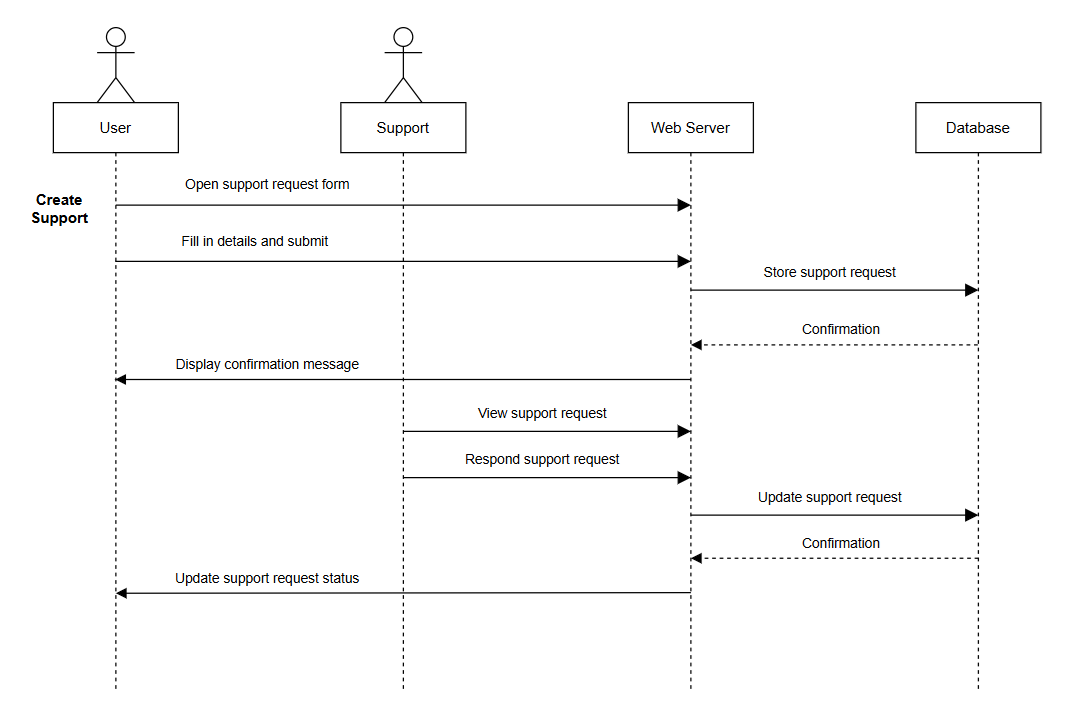
## 8. Preliminary Use Case Models and Sequence Diagrams

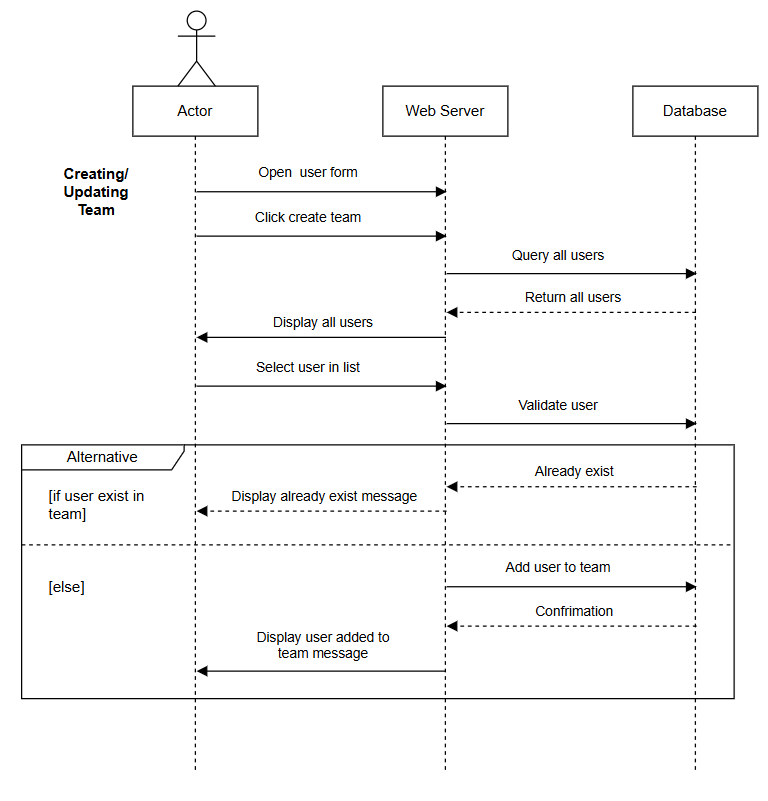
This section presents a list of the fundamental use case models and sequence diagrams that satisfy the system’s requirements. The purpose is to provide an alternative, "structural" view of the requirements stated above and how they might be satisfied in the system.

### 8.1 Use Case Model

### 8.2 Sequence Diagrams

****

****



## 9. Updated Schedule

A screenshot of a computer

Description automatically generated

## 10. Updated Budget

The system will be hosted internally utilizing the company’s existing resources, with no budget allocated for the project, unless a decision is made to migrate to cloud.

## 11. Appendices

### 11.1 Definitions, Acronyms, Abbreviations

WCTC – Western Caroline Trading Company

RESTful – is an interface that two computer systems use to exchange information securely over the internet.

Eloquent – a Laravel object-relational mapper (ORM) that allows to connect to database.

LAN – Local Area Network, collection of devices connected together in one physical location.

HTTP – Hyper Text Transfer Protocol, provide standard communication between a web browser and a web server.

HTTPS – a secured HTTP, with SSL/ TLS convention to supply encrypted communication and secured distinguishing proof of arranged data

HTML5 - a markup language used for structuring and presenting hypertext documents on the World Wide Web