**Software Requirements Specification**

**Version 1.0**

**<<Resource Scheduler>>**

**March 21, 2025**

**Web Resource Scheduler system**

**Ocen John Bosco**

**Team Leader**

**Sep23/BCS/3511U/F**

**Submitted in partial fulfillment**

**Of the requirements of**

**CSC 2101**

**OPERATING SYSTEMS**

Table of Contents

[1. Introduction 3](#_Toc193367776)

[1.1 Purpose 3](#_Toc193367777)

[1.2 Scope of Project 3](#_Toc193367778)

[1.3 Glossary 3](#_Toc193367779)

[1.4 References 3](#_Toc193367780)

[2. Overall Description 4](#_Toc193367781)

[2.1 System Environment 4](#_Toc193367782)

[2.2 Functional Requirements Specification 4](#_Toc193367783)

[2.2.1 Customer Use Case 4](#_Toc193367784)

[2.2.2 Agent Use Case 4](#_Toc193367785)

[2.3 User Characteristics 5](#_Toc193367786)

[2.4 Non-Functional Requirements 5](#_Toc193367787)

[3. Requirements Specification 5](#_Toc193367788)

[3.1 External Interface Requirements 5](#_Toc193367789)

[3.2 Functional Requirements 5](#_Toc193367790)

[3.2.1 Generate Customer 5](#_Toc193367791)

[3.2.2 Assign Customer 6](#_Toc193367792)

[3.3 Detailed Non-Functional Requirements 6](#_Toc193367793)

[3.3.1 Logical Structure of the Data 6](#_Toc193367794)

[3.3.2 Security 6](#_Toc193367795)

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to provide a detailed description of the Resource Scheduler system. It outlines the system's functionality, interfaces, constraints, and how it reacts to external stimuli. This document is intended for stakeholders, developers, and testers to ensure a clear understanding of the system's requirements.

## 1.2 Scope of Project

The Resource Scheduler is a system designed to manage the assignment of customers to agents using various scheduling algorithms. The system aims to optimize customer waiting times and agent utilization rates. It supports the following scheduling algorithms:

* First-Come-First-Serve (FCFS)
* Round Robin (RR)
* Priority Scheduling
* Shortest Job Next (SJN)

The system is intended for use in environments where efficient task scheduling and resource allocation are critical.

## 1.3 Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Customer | A person or entity requesting service from the system. |
| Agent | A resource responsible for handling customer tasks. |
| Scheduling Algorithm | A method used to assign tasks to agents (e.g., FCFS, Round Robin). |
| Waiting Time | The time a customer spends waiting in the queue before being assigned. |
| Utilization Rate | The percentage of time agents spend working versus being idle. |

## 1.4 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

**1.5 Overview of Document**

This document is divided into two main sections:

1. **Overall Description**: Provides a high-level overview of the system, including its environment, use cases, and user characteristics.
2. **Requirements Specification**: Details the technical requirements, including functional and non-functional requirements.

# 2. Overall Description

## 2.1 System Environment

The Resource Scheduler operates in a simulated environment where:

* Customers are generated at random intervals and added to a queue.
* Agents are responsible for processing customer tasks.
* The system supports multiple scheduling algorithms to optimize task assignment.
* The system is designed to run on a local machine or server and can be containerized using Docker for deployment.

## 2.2 Functional Requirements Specification

### 2.2.1 Customer Use Case

**Use Case: Generate Customer**

**Brief Description:**

* The system generates customers at random intervals and adds them to the incoming queue.

**Initial Step-By-Step Description:**

1. The system initializes with a fixed number of customers to generate.
2. A new customer is created with a unique ID, service time, and priority.
3. The customer is added to the incoming queue.
4. The system logs the customer's arrival time.

### 2.2.2 Agent Use Case

**Use Case: Assign Customer**

**Brief Description:**

* The system assigns customers to agents based on the selected scheduling algorithm.

**Initial Step-By-Step Description:**

1. The system checks for available agents.
2. If an agent is available, the system assigns the next customer from the queue.
3. The agent processes the customer's task.
4. If no agents are available, the customer is added to the waiting queue.

## 2.3 User Characteristics

* **System Administrator**: Responsible for configuring the system, including the number of agents and scheduling algorithm.
* **End User**: Interacts with the system to view performance metrics (e.g., average waiting time, agent utilization rates).

## 2.4 Non-Functional Requirements

* **Performance**: The system should handle up to 1000 customers per simulation without significant performance degradation.
* **Scalability**: The system should be scalable to support additional agents and customers.
* **Usability**: The system should provide a user-friendly interface for configuring and monitoring simulations.
* **Reliability**: The system should operate without failure for the duration of the simulation.

# 3. Requirements Specification

## 3.1 External Interface Requirements

The system does not interact with external systems. All data is stored and processed locally.

## 3.2 Functional Requirements

## 3.2.1 Generate Customer

* **Trigger**: The simulation starts.
* **Precondition**: The system is initialized with a fixed number of customers.
* **Basic Path:**
  1. The system generates a new customer with a unique ID, service time, and priority.
  2. The customer is added to the incoming queue.
  3. The system logs the customer's arrival time.
* **Postcondition**: The customer is added to the queue.

### 3.2.2 Assign Customer

* **Trigger**: A customer is added to the queue.
* **Precondition**: The system has at least one agent available.
* **Basic Path:**
  1. The system checks for available agents.
  2. The system assigns the next customer from the queue to an agent.
  3. The agent processes the customer's task.
* **Postcondition**: The customer's task is completed.

## 3.3 Detailed Non-Functional Requirements

### 3.3.1 Logical Structure of the Data

The system uses the following data structures:

* **Customer**: Contains customer\_id, service\_time, priority, and arrival\_time.
* **Agent**: Contains agent\_id, is\_free, and working\_time.
* **Queue**: A FIFO (First-In-First-Out) data structure for managing customers.

### 3.3.2 Security

* The system does not handle sensitive data, so no special security measures are required.
* Access to the system is limited to authorized users.