# Software Requirement Specification (SRS) Document

## 1. Introduction

IRT, a private company located in Kacyiru sector of Gasabo district in Kigali city, has established itself as a notable business entity in Rwanda's capital. However, the company's CEO currently faces substantial challenges in managing and delegating tasks to employees effectively. The traditional methods of task assignment and tracking have proven to be cumbersome, often involving multiple communication channels, manual follow-ups, and inconsistent documentation methods.

This inefficient task management process results in significant expenditure of time, effort, and resources, as the CEO must constantly track assignments, deadlines, and progress through various means such as emails, meetings, and phone calls. The lack of a centralized system leads to communication gaps, missed deadlines, unclear task priorities, and difficulties in evaluating employee performance. Additionally, the absence of real-time tracking capabilities makes it challenging to ensure accountability and maintain efficient workflow management across the organization. The Task Management Application will serve as a centralized platform that enables the CEO to effectively oversee and manage employee tasks through various functionalities. These include creating and assigning tasks, setting deadlines, monitoring progress, activating or deactivating tasks, and evaluating employee performance. The system aims to improve operational efficiency by providing clear task delegation, enhanced accountability, and effective performance monitoring capabilities within the organization.

1.1 Purpose  
The purpose of this document is to provide a comprehensive Software Requirement Specification (SRS) for the Task Management Application for IRT, a private company located in Kigali. The system aims to streamline task delegation by enabling the CEO to create, assign, view, edit, activate/deactivate, set deadlines, and evaluate tasks for employees efficiently.

1.2 Scope  
This SRS outlines the functional and non-functional requirements for the Task Management Application, covering all necessary elements to support the company's need for effective task management. The application will include user-friendly interfaces, a secure backend, and robust features designed to enhance productivityandaccountability.

## 1.3 Definitions, Acronyms, and Abbreviations

* IRT: Integrated Research and Technology.
* UI: User Interface
* API: Application Programming Interface.
* SRS: Software Requirement Specification.
* CEO: Chief Executive Offices

## 1.4 References

* Company documentation on existing task management processes.
* Industry best practices for task management applications.
* Industry standards for software requirements
* Internal company documentation

## 1.5 Overview

This document is structured to detail the complete set of requirements, including functional, non-functional, interface, and system architecture specifications and appendex.

# 2. General Description

2.1 Product Perspective  
The Task Management Application will be an independent system that facilitates task creation, assignment, and management. It will integrate seamlessly with IRT's existing IT infrastructure and will be accessible through web and mobile interfaces.

2.2 Product Functions

* Task Creation and Assignment: The CEO can create and assign tasks to specific employees.
* Task Viewing: Employees can view assigned tasks
* Editing: CEO can view assigned tasks, while the CEO can edit and update them.
* Task Activation: Tasks can be activated or deactivated based on priority.
* Task Deactivation: Tasks can be activated or deactivated based on priority.
* Deadline Management: Setting and modifying task deadlines.
* Task Evaluation: Reviewing task completion and providing feedback.

2.3 User Classes and Characteristics

* CEO: create account of employee, add task, assign, edit, and evaluate tasks.
* Employees: Limited access to view tasks, mark progress, and complete assignments.

2.4 Operating Environment

* Platform: Web-based and mobile application.
* Any Supported Browsers.
* Devices: Desktop, laptop, and smartphones

## 2.5 Design and Implementation Constraints

- Security: Compliance with data protection laws.  
- Scalability: Ability to support multiple users simultaneously.  
- Technology Stack: Built using html for the front end, php for the backend

2.6 Assumptions and Dependencies

* Reliable internet connectivity.
* Anyone have basic computer literacy and other electronic device

## 3. Functional Requirements

|  |  |  |
| --- | --- | --- |
| ID | Requirement Description | RESOURSE |
| FR1 | The system shall allow the CEO to create and assign tasks. | CEO OF IRT |
| FR2 | The system shall enable employees to view their assigned tasks. | CEO OF IRT |
| FR3 | The CEO shall have the ability to edit and update tasks. | CEO OF IRT |
| FR4 | The system shall provide a feature for activating Task. | CEO OF IRT |
| FR5 | The system shall provide a feature for deactivating tasks. | CEO OF IRT |
| FR6 | The system shall allow deadlines to be set and modified. | CEO OF IRT |
| FR7 | The system shall enable the CEO to evaluate and provide feedback on completed tasks. | CEO OF IRT |

## 4. Non-Functional Requirements

|  |  |
| --- | --- |
| Category | Requirement |
| Performance | The system should load task details within 2 seconds. |
| Security | Data encryption for task data storage and transfer. |
| Usability | The UI must be intuitive and user-friendly for non-technical users. |
| Reliability | The system should have 99.9% uptime. |
| Scalability | The system must support up to 1,000 concurrent users. |

# 5. Interface Requirements

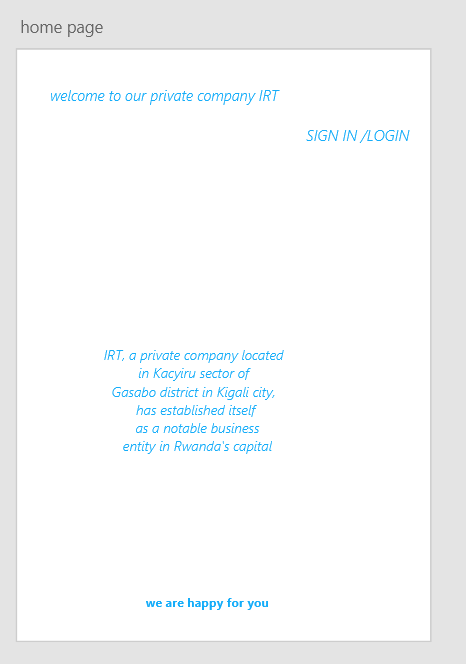
5.1 User Interface  
- Dashboard: A central dashboard displaying all tasks, categorized by status (active, completed, pending).  
- Forms: User-friendly forms for task creation and assignment, login interface.  
- report: Integrated calendar for tracking task deadlines.

5.2 Hardware Interface  
- Supports devices with a minimum of 1GB RAM and modern web browsers.  
  
5.3 Software Interface  
- Integration with IRT’s existing email system for notifications.  
- RESTful APIs for data communication.

# 6. System Architecture

6.1 Overview  
- The system will follow a client-server architecture.  
- Components:  
 - Frontend: html for dynamic UI.  
 - Backend: php for handling logic and APIs.  
 - Database:msql for scalable data storage.  
- Flow:  
 - User actions trigger requests from the frontend.  
 - Backend processes the request and interacts with the database.  
 - Processed data is sent back to the frontend.

## 7. Appendix

7.1. home page  


## 7.2 login page

