Software Requirements Specification

Smart Attendance and Access Management System (SAAMS)

|  |  |  |
| --- | --- | --- |
| **Author** | **Validation** | **Approval** |
|  |  |  |
|  |  |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Prepared / Edited By** | **Changes** |
| 0.1 | 04/11/2024 | Adithya Pokala | Initial Draft |

Glossary

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
|  |  |
|  |  |
|  |  |

Table of Contents

[1 Introduction 4](#_Toc181648288)

[1.1 Purpose 4](#_Toc181648289)

[1.2 Scope 4](#_Toc181648290)

[2 System Context 4](#_Toc181648291)

[2.1 Overview 4](#_Toc181648292)

[2.2 Product Functions 4](#_Toc181648293)

[2.3 User Characteristics 4](#_Toc181648294)

[2.4 Constraints 5](#_Toc181648295)

[2.5 Assumptions and Dependencies 5](#_Toc181648296)

[3 Specific Requirements 6](#_Toc181648297)

[3.1 External Interface Requirements 6](#_Toc181648298)

[3.2 Functional Requirements / System Features 6](#_Toc181648299)

# Introduction

This project is a solution for ...

## Purpose

This document aims to capture the requirements for the software to cater the smart attendance and access management.

## Scope

The product contains a desktop application developed using Electron and a REST API developed using .NET Core framework. The REST API contains endpoints for User management and User Operations. The desktop application contains interface for managing users, handling different user roles, performing user operations and report generations. TODO: explain in detail.

# System Context

## Overview

The …

## Product Functions

This project will help users with

* …

## User Characteristics

The following application will have …

## Constraints

…

## Assumptions and Dependencies

…

# Specific Requirements

## External Interface Requirements

…

## Functional Requirements / System Features

### REST API

#### Login Endpoint

The request endpoint for login will be as follows:

|  |  |  |
| --- | --- | --- |
| ID | Endpoint | HTTP Method |
| RA\_EP\_LE\_1 | /api/login | POST |

The request data for the endpoint will be as follows:

|  |  |
| --- | --- |
| ID | Data |
| RA\_RQ\_LE\_1 | {  "userName": "string",  "password": "string"  } |

The response from the endpoint will be as follows:

|  |  |  |
| --- | --- | --- |
| ID | Response | HTTP Status |
| RA\_RS\_LE\_1 | Under nominal conditions then the response is -  {  "message": "Login successful",  "status": true,  "user": {  "id": uint,  "createdAt": "YYYY-MM-DDThh:MM:ss.xxxZ",  "updatedAt": "YYYY-MM-DDThh:MM:ss.xxxZ",  "userName": "string",  "password": "string",  "firstName": "string",  "lastName": "string",  "email": "string",  "phone": "string",  "sex": "string",  "dateOfBirth": “YYYY-MM-DD”,  "dateOfJoining": “YYYY-MM-DD”,  "roleId": 0,  "companyId": 0,  "designationId": 0,  "departmentId": 0,  "shiftId": 0  },  "users": null  } | 200 |
| RA\_RS\_LE\_2 | If the request contains no data  {  "message": "Login failed",  "status": false,  “user”: null,  “users”: null  } | 400 |
| RA\_RS\_LE\_3 | If the username or password is wrong  {  "message": "Login failed",  "status": false,  “user”: null,  “users”: null  } | 404 |