Software Requirements Specification

for

Employee Management System

Version 1.0 .0

Prepared by IJSE Software Company

IJSE

11th of October 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1   Purpose

The purpose of this document is to define the business requirements for the development of a Employee Attendance System (EAS) for efficiently tracking and maintaining employee attendance records, and providing essential information to the HR department and employees.

1.2   Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3   Intended Audience and Reading Suggestions

This document is intended for project stakeholders, including the HR department, developers and other relevant personnel involved in the development and implementation of the Employee Attendance System.

1.4   Project Scope

The Employee Attendance System will track and manage employee attendance based on company policies. It will provide HR with attendance summaries. Employees will be able to access their own attendance.

1.5   References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Features

<Summarize the major features the product contains or the significant functions that it performs or lets the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or a class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.>

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7   Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

3.   System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

3.1   Employee Registration

3.2   Attendance Tracking

3.3   Reporting

3.4   Security

4.   External Interface Requirements

4.1   User Interfaces

You can reach the UI using the below link
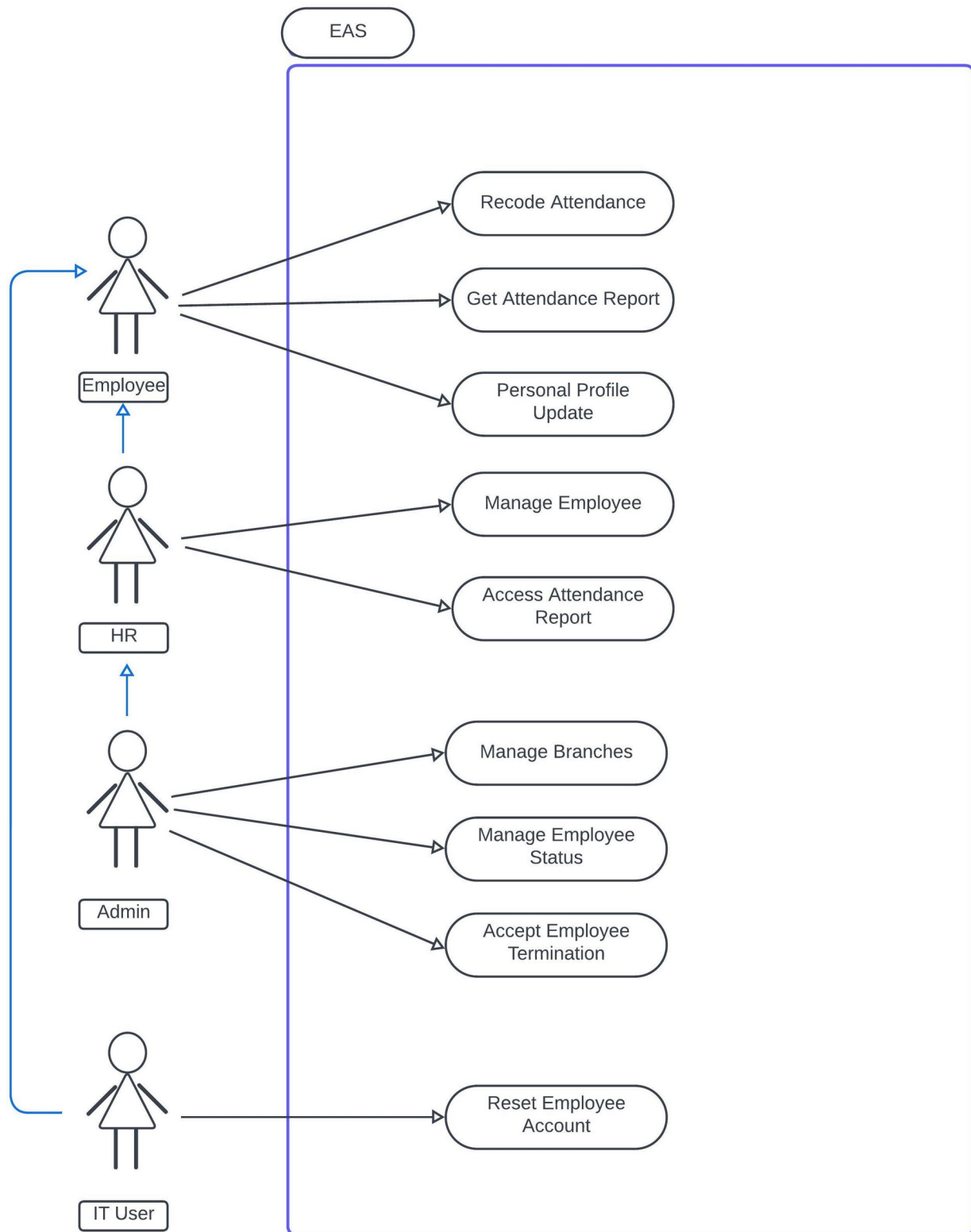
https://www.figma.com/file/sWWzQX4JRENkDvqDV7nKjd/Use-case?type=design&node-id=0-1&mode=design&t=nbAfdXoob1xUrRy7-0

4.2   Hardware Interfaces

4.3   Software Interfaces

4.4   Communications Interfaces

5. Use Case Diagram

6. Entity Relationship Diagram