

Department of Information and Communication Technology

Faculty of Technology

University of Ruhuna

**Employee Management System**

**Software Requirement Specification**

Group Project (ICT3183)

Project ID: 07

Submitted by:

1. Index no. – (Name)

Submitted to:

(Supervisor’s signature)

…………………………..

Name of the supervisor: Ms.Malsha Prabuddhi

Date of submission

Table of Contents

Table of Contents i

Revision History i

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 Project Documentation 2

2.7 User Documentation 3

2.8 Assumptions and Dependencies 3

3. External Interface Requirements 4

3.1 User Interfaces 4

3.2 Hardware Interfaces 4

3.3 Software Interfaces 4

3.4 Communications Interfaces 4

4. System Features 5

4.1 System Feature 1 5

4.2 System Feature 2 5

4.3 System Feature 3 (and so on) 5

5. Other Nonfunctional Requirements 6

5.1 Performance Requirements 6

5.2 Safety Requirements 6

5.3 Security Requirements 6

5.4 Software Quality Attributes 6

5.5 Business Rules 6

6. Other Requirements 7

Appendix A: Glossary 8

Appendix B: Analysis Models 9

Appendix C: To Be Determined List 10

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The Employee Management System (EMS) aims to streamline and automate various aspects of human resource management within an organization. This system is designed to enhance efficiency, accuracy, and transparency in handling employee-related information and processes. By providing a centralized platform, it enables HR professionals and organizational stakeholders to manage, monitor, and optimize workforce activities effectively.

Version 1.1

## Document Conventions

* Document Title
  + The title of the document should be prominently displayed at the beginning, clearly indicating that it is an SRS.
* Version Control
  + Include a version number and date to track the document's revisions.
  + This helps readers identify the most recent version and understand changes made over time.
* Table of Contents
  + Provide a table of contents to outline the structure of the document and help users navigate to specific sections.
* Headers and Footers
  + Consistently use headers and footers to display the document title, section names, and page numbers.
  + This maintains a professional appearance and aids in document navigation.
* Formatting Styles
  + Use consistent formatting styles for text, such as font type, size, and color. Bold, italic, and underline can be used consistently to highlight important points.
* Section Numbering
  + Number sections and subsections to create a logical hierarchy.
  + This makes it easier for readers to locate specific information and understand the relationships between different sections.
* Bullet Points and Numbered Lists
  + Use bullet points or numbered lists for clarity when presenting lists or sequences of information.
* References
  + If external documents or sources are referenced, provide a bibliography or list of references for readers to access additional information.
* Graphical Elements
  + Use consistent conventions for graphical elements, such as charts, diagrams, and tables. Include titles, labels, and legends to enhance comprehension.
* Naming Conventions
  + If the document refers to specific entities (e.g., software components, modules), establish and adhere to naming conventions to maintain consistency.

## Intended Audience and Reading Suggestions

### Intended Audience:

* Developers: Individuals responsible for designing, coding, and testing the software.
* Project Managers: Individuals overseeing the software development project, responsible for planning, scheduling, and resource allocation.
* Testers: QA professionals responsible for testing the software to ensure quality and functionality.
* Documentation Writers: Individuals responsible for creating user manuals, help guides, and other documentation related to the software.
* Users: End-users who will interact with the software, including employees, administrators, or customers.
* Marketing Staff: Individuals involved in promoting and marketing the software to potential users or clients.

### Description of Document Contents and Organization

The SRS document provides a detailed description of the software requirements, outlining its functionality, features, constraints, and user interactions. It is organized into sections that cover various aspects of the software, including,

* Introduction: Provides an overview of the software, its purpose, scope, and intended audience.
* General Description: Describes the context of the software, including its features, constraints, and assumptions.
* Specific Requirements: Details the functional and non-functional requirements of the software, including user interactions, system behavior, and performance criteria.
* External Interface Requirements: Describes the interfaces between the software and external systems, including hardware, software, and communication protocols.
* System Features: Lists and describes the specific features and functionalities of the software.
* Other Requirements: Covers additional requirements such as documentation, training, and support.
* Appendices: Includes supplementary information such as glossary, references, and supporting documentation.

### Reading Sequence Suggestions

* Begin with the Overview Sections
  + Start by reading the Introduction and General Description sections to understand the purpose, scope, and context of the software.
  + Proceed to the System Features section to get an overview of the specific functionalities and features of the software.
* Developer and Project Manager Focus:
  + Developers and project managers should focus on the Specific Requirements section to understand the detailed functional and non-functional requirements of the software.
  + They should also review the External Interface Requirements section to understand the integration points with external systems.
* Tester Focus:
  + Testers should pay close attention to the Specific Requirements section to understand the expected system behavior and performance criteria for testing purposes.
* User Focus:
  + Users should refer to the System Features section to understand the available functionalities and how they can interact with the software.
  + They may also find the General Description section useful for understanding the context and constraints of the software.
* Documentation Writer Focus:
  + Documentation writers should review the Appendices section for supplementary information such as the glossary and references to support their documentation efforts.

## Product Scope

### Software Description and Purpose

The Employee Management System (EMS) is a comprehensive software solution designed to streamline and automate various aspects of human resource management within an organization. Its purpose is to centralize employee-related processes, data, and interactions, thereby enhancing organizational efficiency and employee satisfaction. The EMS serves as a centralized platform for managing the entire employee lifecycle, from recruitment and onboarding to performance evaluation and offboarding.

### Benefits

* Efficiency
  + The EMS reduces manual administrative tasks, such as data entry and paperwork, allowing HR personnel to focus on strategic initiatives.
* Accuracy
  + By maintaining a centralized employee database, the EMS ensures that employee information is accurate, up-to-date, and easily accessible.
* Transparency
  + The system fosters transparent communication between employees, managers, and HR professionals by providing a unified platform for accessing and sharing information.
* Decision-making
  + With robust reporting and analytics capabilities, the EMS enables data-driven decision-making for talent management, resource allocation, and strategic planning.
* Employee Experience
  + By providing self-service tools and easy access to HR services, the EMS enhances the overall employee experience, leading to higher satisfaction and engagement.

### Objectives and Goals

* Efficient HR Processes
  + The primary objective of the EMS is to streamline HR processes, reducing administrative overhead and improving efficiency.
* Data Accuracy
  + Another objective is to maintain accurate and up-to-date employee data, ensuring compliance with regulations and minimizing errors.
* Enhanced Communication
  + The EMS aims to facilitate transparent communication between employees and HR personnel, fostering a culture of openness and collaboration.
* Strategic Decision-making
  + The system seeks to provide HR professionals with the necessary tools and insights to make informed decisions about workforce management and development.
* Employee Satisfaction
  + Ultimately, the goal of the EMS is to enhance employee satisfaction and engagement by providing a seamless and user-friendly experience throughout the employee lifecycle.

### Goals

* Operational Efficiency
  + By automating HR processes and reducing manual workload, the EMS contributes to overall operational efficiency, aligning with the corporate goal of maximizing productivity and resource utilization.
* Talent Management
  + The system supports strategic talent management initiatives by providing insights into workforce trends, enabling organizations to identify and nurture top talent, which aligns with the goal of attracting, retaining, and developing skilled employees.
* Employee Engagement
  + By prioritizing employee experience and communication, the EMS helps foster a positive work environment, aligning with the corporate goal of promoting employee engagement, satisfaction, and retention.
* Compliance
* Ensuring accurate and compliant handling of employee data, the EMS helps mitigate legal and regulatory risks, safeguarding the organization's reputation and integrity, thereby aligning with the corporate goal of adhering to ethical and legal standards.

In summary, the Employee Management System serves as a strategic tool that supports corporate goals and business strategies by enhancing HR processes, fostering employee engagement, and facilitating data-driven decision-making. Its objectives and goals are closely aligned with the broader objectives of the organization, contributing to its overall success and competitiveness.

## References

Dk, J. (2018) *Employee RMS SRS 1.0.docx*, *Academia.edu*. Available at: https://www.academia.edu/36634340/Employee\_RMS\_SRS\_1\_0\_docx (Accessed: 23 February 2024).

# Overall Description

## Product Perspective

For a cleaning service company's employee management system, the product perspective involves understanding how the system fits within the broader context of the company's operations and how it interacts with various stakeholders and systems. Here are some considerations for the product perspective:

1.System Context: The employee management system operates within the context of the cleaning service company's overall operations. It interacts with other systems.

such as

Managerial Oversight: Admin and supervisors use the system to oversee and manage the workforce. This includes tasks such as assigning cleaning tasks, monitoring employee attendance and performance and generating reports on employee productivity and service quality.

2.Interfaces: The system needs to interface with admin, supervisors and the accountant. Interfaces may include web portals.

3.Dependencies: The system may depend on external factors such as internet connectivity, server infrastructure, and access to employee data (e.g., contact information, schedules) stored in other databases.

4.Stakeholders: Stakeholders include cleaning staff who use the system to manage their schedules and tasks, managers who oversee employee performance and scheduling.

5.Integration: Integration with scheduling software is crucial for efficiently assigning tasks to employees based on availability.

6.Lifecycle Considerations: The employee management system should be designed to evolve with the company's needs over time. This includes accommodating changes in workforce size, organizational structure, and regulatory requirements. Regular updates and maintenance are necessary to ensure optimal performance and security.

7.Regulatory Compliance: The system must comply with labor regulations regarding employee scheduling, working hours, overtime, and employee data privacy. Compliance with industry-specific regulations, such as those related to cleaning products and safety protocols, may also be necessary.

By considering these aspects of product perspective, the cleaning service company can develop an employee management system that effectively meets the needs of its workforce, managers, and clients while ensuring compliance with relevant regulations and standards.

## Product Functions

**Admin:**

Employee Management:

* Add, Delete, Update and View Employee details
* Assign tasks to staff
* Monthly generate reports

**Supervisor:**

Inventory Management:

* Add, Delete, Update and View Stock
* Monthly generate report

Leave and Attendance:

* Manage staff leave
* View leave history
* Maintain Attendance
* Monthly report generates

**Accountant:**

Calculate Salary:

* Add salary
* Manage salary
* Calculate etf, epf
* Monthly report generates
* Message alert

## User Classes and Characteristics

**Admin**

Frequency of Use: Daily to Regularly

Subset of Product Functions Used: Scheduling, task assignment, performance monitoring, reporting

Characteristics: Have higher technical expertise compared to cleaning staff. They require access to a broader range of functions for managing the workforce effectively. They prioritize efficiency, data accuracy, and comprehensive reporting capabilities.

**Supervisors**

Frequency of Use: Daily to Regularly

Subset of Product Functions Used: manage leave and attendance, inventory management, reporting

Characteristics: Have higher technical expertise compared to cleaning staff. They require access to a broader range of functions for managing the workforce effectively. They prioritize efficiency, data accuracy, and comprehensive reporting capabilities.

**Accountant**

Frequency of Use: Daily to Regularly

Subset of Product Functions Used: calculate salary, performance monitoring, reporting

Characteristics: Have higher technical expertise compared to cleaning staff. They require access to a broader range of functions for managing the workforce effectively. They prioritize efficiency, data accuracy, and comprehensive reporting capabilities.

## Operating Environment

The operating environment for a cleaning service company employee management system involves various hardware, software, and network components. Here's a description of the environment in which the software will operate:

1.Hardware Platform:

The employee management system should be compatible with standard computing hardware commonly used in office environments. This includes desktop computers and laptops.

2.Operating System and Versions:

The software should be compatible with common operating systems used in business environments, such as:

- Windows: Windows 10 or later

3.Web Browser Compatibility:

The system should be accessible via popular web browsers to accommodate users accessing the system from different devices. Compatibility with the following browsers is recommended:

- Google Chrome

- Mozilla Firefox

- Microsoft Edge

4.Database:

The system may require a backend database to store employee information, schedules, and other data. Commonly used databases include:

- MySQL

- Microsoft SQL Server

By operating within this environment, the employee management system can effectively support the needs of the cleaning service company, providing efficient employee management and enhancing overall operational productivity.

## Design and Implementation Constraints

Design and implementation constraints for a cleaning service company employee management system may include:

1.Regulatory Compliance: The system must adhere to labor regulations governing employee scheduling, working hours, wages, and data privacy. Compliance with industry-specific regulations, such as those related to cleaning product safety, may also be necessary.

2.Hardware Limitations: The system should be designed to operate efficiently on standard computing hardware commonly used in office environments. Memory and processing constraints should be considered to ensure optimal performance, especially during peak usage times.

3.Interfaces with Other Applications: The system may not need to integrate with existing software applications used by the cleaning service company, such as scheduling software, payroll systems, CRM software, and accounting software.

4.Technology Stack: Specific technologies, tools, and databases may be mandated by the company's IT policies or chosen based on compatibility with existing infrastructure. For example, the use of a particular programming language, framework, or database management system may be required.

5.Security Considerations: The system must incorporate robust security measures to protect sensitive employee information and ensure data integrity.

6.Design Conventions and Standards: The system should adhere to design conventions and programming standards established by the company or industry best practices. Consistency in coding style, naming conventions, and architectural patterns can facilitate maintenance and collaboration among developers.

7.Maintenance Responsibility: Clarification on whether the customer's organization or the development team will be responsible for maintaining the delivered software should be established. This may impact design decisions, documentation requirements, and support agreements.

8.Scalability and Performance: The system should be designed to accommodate potential growth in the number of employees and clients served by the cleaning service company. Scalability considerations should be incorporated into the architecture to ensure that the system can handle increased user loads without degradation in performance.

Addressing these constraints during the design and implementation phases will help ensure that the employee management system meets the needs of the cleaning service company while complying with regulatory requirements, maintaining security, and aligning with technical and organizational standards.

## Project Documentation

Project documentation for a cleaning service company employee management system typically includes various components to ensure clarity, transparency, and ease of understanding for stakeholders involved in the project. Here are the key components that may be delivered along with the software:

1.Introduction:

- Provides an overview of the employee management system project.

- Describes the purpose, scope, and objectives of the system.

- Outlines the key stakeholders and their roles in the project.

- Introduces the structure and contents of the documentation.

2.Requirements Documentation:

- Details the functional and non-functional requirements of the system.

- Includes use cases, user stories, and acceptance criteria.

- Describes any constraints or dependencies that affect the system design and implementation.

3.Design Documentation:

- Describes the architecture and design of the employee management system.

- Includes system architecture diagrams, database schemas, and data flow diagrams.

- Documents the software components, modules, and their interactions.

4.Implementation Documentation:

- Provides instructions for installing, configuring, and deploying the software.

- Documents the coding standards, guidelines, and conventions followed during development.

- Includes details of third-party libraries, frameworks, and tools used in the implementation.

- Provides guidance for developers on how to contribute to the project and maintain the codebase.

5.User Manual:

- Guides users on how to use the employee management system effectively

6.Testing Documentation:

- Describes the testing approach, strategies, and methodologies used to validate the system.

- Provides information on unit testing, integration testing, system testing and user acceptance testing.

7.Deployment Documentation:

- Guides administrators on deploying the employee management system in production environments.

- Includes system requirements, hardware recommendations, and installation instructions.

- Documents procedures for configuring system settings, integrating with other applications, and setting up user accounts and permissions.

8.Maintenance and Support Documentation:

- Provides guidelines for maintaining and supporting the employee management system post-deployment.

- Includes information on monitoring system performance, applying updates and patches, and troubleshooting common issues.

- Documents procedures for handling user support requests, bug reports, and feature requests.

9.Change Log:

- Lists all changes, enhancements, and bug fixes made to the software across different versions.

- Provides transparency regarding the evolution of the system and the reasons behind specific changes.

The documentation may adhere to industry-standard formats such as IEEE Standard 830 for Software Requirements Specifications or be customized based on the preferences and requirements of the cleaning service company and its stakeholders.

## User Documentation

For a cleaning service company employee management system, the user documentation plays a crucial role in helping users understand and effectively utilize the software. Here are the user documentation components that may be delivered along with the software:

1.User Manual:

- Provides comprehensive guidance on how to use the employee management system.

- Includes step-by-step instructions for performing common tasks such as employee management, assigning tasks, leave and attendance, and reporting.

- Explains the user interface elements, navigation paths, and functionalities available in the system.

- May include screenshots or illustrations to aid understanding.

2.Onboarding Guide:

- Offers guidance specifically tailored for new users who are unfamiliar with the system.

- Walks users through the initial setup process, including account creation, profile setup, and system configuration.

- Provides an overview of key features and functionalities to help users get started quickly.

3.Training Materials:

- Includes tutorials, videos, or interactive demos to facilitate user training and skill development.

- Covers advanced features and best practices for maximizing productivity and efficiency.

- Can be delivered in various formats such as PDF documents, online courses, or recorded webinars.

The user documentation may be delivered in various formats, such as PDF documents, online help systems, or embedded within the software interface itself. It should be designed to be user-friendly, easily accessible, and comprehensive enough to address the needs of users with varying levels of expertise and experience.

## Assumptions and Dependencies

Assumptions and dependencies play a significant role in shaping the development and operation of a cleaning service company employee management system. Here are some key assumptions and dependencies that could affect the requirements stated in the Software Requirements Specification (SRS):

Assumptions:

1.Internet Connectivity: It is assumed that users of the employee management system will have reliable internet connectivity to access the system, update information in real-time, and communicate with other users.

2.User Training: It is assumed that users will receive adequate training on how to use the system effectively. This includes understanding the features, functionalities, and best practices for managing employee-related tasks.

3.Data Accuracy: The system assumes that the data entered by users, such as employee information, schedules, and task assignments, is accurate and up-to-date. Inaccurate data could lead to errors in scheduling and task assignment.

4.Hardware Compatibility: The system assumes compatibility with standard computing hardware. commonly used by employees. Compatibility issues could arise if users are using outdated or incompatible hardware.

5.Regulatory Compliance: It is assumed that the system will comply with relevant labor laws, regulations, and industry standards related to employee management, data privacy, and security. Non-compliance could result in legal or regulatory penalties.

6.Maintenance and Support: The system assumes that adequate resources will be allocated for ongoing maintenance, support, and updates. This includes addressing bugs, adding new features, and providing technical assistance to users.

7.Scalability: The system assumes that it will be able to scale to accommodate the growing needs of the cleaning service company, including an increasing number of employees, clients, and tasks.

Dependencies:

.

1.Data Migration: If migrating data from existing systems or databases, the project depends on the not successful extraction, transformation, and loading of data without loss or corruption.

2.Development Tools and Libraries: The project may depend on the availability and compatibility of specific development tools, libraries, and frameworks used in the software development process. Changes to these tools could impact the development environment and project workflow.

Addressing these assumptions and dependencies is crucial for mitigating risks and ensuring the successful development, deployment, and operation of the cleaning service company employee management system. Regular communication, collaboration, and monitoring of these factors throughout the project lifecycle are essential to minimize potential disruptions and ensure project success.

# External Interface Requirements

## User Interfaces

## A screenshot of a computer

Figure 1-Add Employee Interface

## A screenshot of a computer Description automatically generated

Figure 2- View Employee Details Interface

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# 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.>

## System Feature 1

## System Feature 2

## System Feature 3 (and so on)

**In your case you can write this whole section (Chapter 4 and its sub sections) based on a Use Case Diagrams, Use Case Scenarios, Initial Class Diagram, and Activity Diagram. (Initial Class Diagram and Activity Diagrams are optional) If you have a fairly large number of use cases you can make use of packages to group the use cases into multiple diagrams. Then for each package draw a separate use case diagram**

**Use Alistair Cockburn’s template from the unit Software Engineering II to document your use cases**.

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>