Text

Description automatically generated

**EMPLOYEE LEAVE MANAGEMENT**

**Software Requirement Specification (SRS) Document**

**Sprint 2 Implementation**

**Project Timeline:22-02-2023 to 6-03-2023**

**SOFTWARE REQUIREMENT SPECIFICATIONS**

**Project Name:** Employee Leave Management

**Document Title:** SRS

**Project Timeline: 22-02-2023 to 6-03-2023**

**INDEX**

**1. Introduction**

1.1 Purpose

1.2 Intended audience

1.3 Intended use

1.4 Scope

**2. Overall Description**

2.1 Assumptions and Dependency

**3. System feature and requirements**

**3.1 Functionality**

3.1.1 ELM\_01 -> **new\_employee\_registration()**

3.1.2 ELM\_02-> **employee\_login()**

3.1.3 ELM\_03-> **login\_interface()**

3.1.4 ELM\_04-> **load\_user\_data()**

3.1.5 ELM\_05-> **validate\_login()**

3.1.6 ELM\_06 -> **leave\_management\_interface()**

3.1.7 ELM\_07-> **manager\_login()**

3.1.8 ELM\_08-> **hr\_login()**

3.1.9 ELM\_09-> **load\_leave\_balance()**

3.1.10ELM\_10-> **employee\_leave\_data()**

3.1.11ELM\_11-> **validate\_leave()**

3.1.12ELM\_12-> **employee\_interface();**

3.1.13ELM\_13->**apply\_leave\_interface();**

3.1.14ELM\_14-> **check\_leave\_balance();**

3.1.15ELM\_15-> **employee\_leave\_history();**

3.1.16ELM\_16-> **show\_all\_leave();**

3.1.17ELM\_17-> **admin\_interface();**

3.1.18ELM\_18->  **modify\_leave\_balance();**

3.1.19ELM\_19-> **manager\_interface();**

3.1.20ELM\_20-> **give\_permission();**

3.1.21ELM\_21->  **show\_pending\_leave();**

**3.2 System requirement**

3.2.1 Software requirements

3.2.2 Hardware requirements

3.2.3 System features

**4. Data Flow Diagrams**

4.1 DFD level 0

4.2 DFD level 1

**5.Appendix**

**1.Introduction**

The introduction of the software requirement specification provides an overview of the entire software. The entire SRS covers overview description, purpose, scope, tools used and basic description. The aim of this document is to gather, analyse and give an in-depth insight into the **Employee Leave Management System** by defining the problem statement in detail.

The detailed requirements of theEmployee Leave Management Systemis provided in this document.

**1.1 Purpose:**

The purpose of this document is to show the requirements for the Employee Leave Management System which creates a direct contact between the

**1.2 Intended Audience:**

There is no such specific audience, it could be an Employee.

**1.3 Intended Use:**

* Management Team
* Employees

**1.4 Scope:**

This project aims to create the development of Employee Leave Management System This system consists of an application which will serve as a platform for employees to apply for leaves.

**2. Overall Description**

The Employee Leave Management System was designed to automate the employee administration and management and the strategic planning of leaves for employees. This program manages all the employee leave records.

Employee Leave Data Management System was developed primarily to handle information about the employee leaves that serves as an instrument for better management.

**2.1 Assumptions and Dependency:**

* System should have any flavour of Linux installed.
* System should have either 4GB or more RAM.
* The service is used preferably on a desktop or laptop.

**3. System Features and Requirements**

**3.1 Functionality:**

**3.1.1 ELM\_01 ->** **new\_employee\_registration():** This function will handle the registrations of new employees. Employees need to register themselves successfully to create login credentials.

**3.1.2 ELM\_02->** **employee\_login():** It will allow user/employee to login using credentials generated after registration.

**3.1.3 ELM\_03-> login\_interface():** It will create an interface for an employee to Register or login based on his role

**3.1.4 ELM\_04->** **load\_user\_data():** This function is used to load user data from a text file called “user\_data.txt” into a vector of Employee\_registration objects called “user\_data\_list”.

**3.1.5 ELM\_05->** **validate\_login():** It checks if the password and employee ID match the input and whether the role matches the user’s role.

**3.1.6 ELM\_06 ->** **leave\_management\_interface() :** Main interface of Employee Leave management system .

**3.1.7 ELM\_07-> manager\_login():** This function is responsible for displaying the manager's interface and handling the different options available to the manager.

**3.1.8 ELM\_08-> hr\_login():** This function provides a way for managers to log in and access the manager role functionalities in the system.

**3.1.9 ELM\_09->** **load\_leave\_balance():** This function reads a file "leave\_balance.txt" and stores its contents in a vector “leave\_balance\_data’’

**3.1.10ELM\_10->** **employee\_leave\_data() :** This function searches for the leave balance data of a specific employee based on their employee ID (eid)

**3.1.11ELM\_11->** **validate\_leave():** This function validates whether an employee can take a leave or not.

**3.1.12ELM\_12->** **employee\_interface():** This function displays the interface for an employee to interact with the leave management system. It provides three options for the employee: to apply for leave, check leave balance, or check their leave history.

**3.1.13ELM\_13->apply\_leave\_interface():** This function will create an interface for an employee to apply for leave by entering date, number of days and reason for taking leave.

**3.1.14ELM\_14-> check\_leave\_balance():** The function is used to check the leave balance of an employee based on their ID.

**3.1.15ELM\_15-> employee\_leave\_history():** This function is used to display the leave history of an employee based on their ID.

**3.1.16ELM\_16->** **show\_all\_leave():** It will display the leaves applied by all employees.

**3.1.17ELM\_17->** **admin\_interface():** It will show the interface of HR /Admin and will display the options available to him.

**3.1.18ELM\_18->** **modify\_leave\_balance():** This is a method of the class Admin that is used to modify the leave balance of an employee based on their ID and the type of leave to be modified**.**

**3.1.19ELM\_19-> manager\_interface():** This is a method of the Manager class that displays a menu of options for a manager user.

**3.1.20ELM\_20->** **give\_permission():** This is a method of the Manager class that is used to approve or reject a leave request for an employee based on their ID.

**3.1.21ELM\_21->**  **show\_pending\_leave():** This fuction will display the pending leaves of an employees

**3.2 System Requirements:**

**3.2.1: Software Requirements:**

* + - * Operating System -Windows 11
      * Server-side script - C++ Language
      * IDE - Putty
      * Libraries used - C++ libraries

**3.2.2: Hardware Requirements:**

* Processor - i5/Intel Processor
* RAM - 4GB (min)
* Hard Disk - 128 GB
* Keyboard - Standard Keyboard
* Mouse - Two or three button mouse
* Monitor - Any

**3.2.3: Tools and methods to be used:**

* Valgrind
* Gcov
* STL library
* C++ File Handling
* Make multiple files

**3.3 System Features:**

* **Supportability:** The system is easy to use.
* **Design Constraint:** The system is built using only C++ language.
* **Usability:** The Employee Leave Management system is used to apply ,accept and reject leaves.
* **Reliability & Availability:** The application is available 24/7 that is whenever an employee would like to use the application. An employee can simply Register and login.

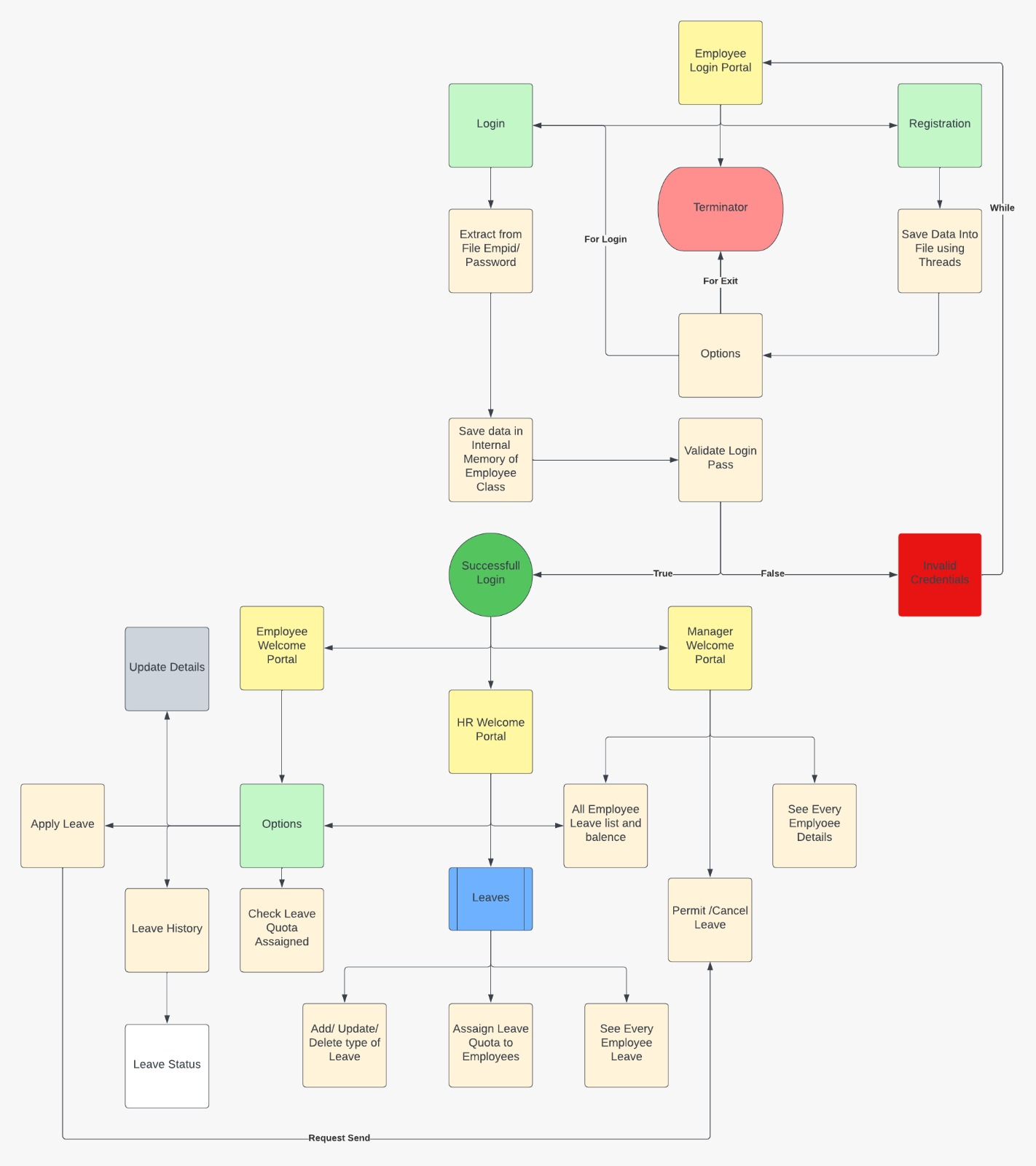
**4. Data Flow Diagrams**

**4.1 DFD Level 0:**

**Diagram

Description automatically generated**

**4.2: DFD Level 1:**



**6.Appendix**

1. C++ Language Reference: <https://en.cppreference.com/w/cpp>
2. Microsoft C++ Language Reference: <https://docs.microsoft.com/en-us/cpp/cpp/cpp-language-reference>
3. GeeksforGeeks C++ Reference: <https://www.geeksforgeeks.org/c-plus-plus/>
4. Programiz C++ Reference: <https://www.programiz.com/cpp-programming/library-function>
5. Tutorialspoint C++ Reference: <https://www.tutorialspoint.com/cplusplus/cpp_standard_library.htm>
6. W3Schools C++ Reference: <https://www.w3schools.com/cpp/default.asp>
7. Stack Overflow C++ Reference: <https://stackoverflow.com/questions/tagged/c%2B%2B>
8. CppShell C++ Reference: <https://www.cppshell.com/reference/>
9. CppReference C++ Reference: <https://en.cppreference.com/w/cpp/links/libs>