**ABSTRACT**

Employee leave management system combine number of processes and systems to automate and easily manage employee data, leave request, track and grant leave. In many institution staff are entitled to different types of leave, these leave are granted according to institution policy. Administrative department is mostly responsible for managing and granting leave request. To this end, most institution used conventional method of requesting, granting and managing leave. In conventional method, leave is manually request by writing letter to head of department.

The head of department minutes and forward the request to higher staff for approval. This method is time consuming, prone to error, require more paper work and difficult to manage. Hence the need for an automated leave management system that is faster, error free, less paper work and easy to manage. The system was achieved by developing an automated employee leave management system using the requirement model. The System is implemented using standalone application which include C++ and FS and runs on Windows operating system.

**CHAPTER 1**

**INTRODUCTION**

An employee leave management system is a platform that enables staff and admin of an organization or institution to easily apply, correctly allocate, track and grant leave. In many institutions, staff are entitle to different kind of leave; study leave, sick leave, annual leave, leave without pay, research leave and maternity leave. These leave are been taken and recorded according to institution policy. The administrative department is mostly considered as one of the most important assets in every institution. It is a part of administrative department function to keep all the records of employee.

Every institution administrative department is information driven, and the admin staff drives and carry out day to day activities. In most institutions, conventional method of requesting and managing leave is been used. In conventional method, academic staff is required to manually write and submit leave application to administrative department through the Head of Department (HOD). The head of department minutes and forward the request to higher staff in charge, who approve or reject the request. This method is time consuming, prone to error, required more paper work and tedious to maintain. Hence the need for an automated method that is faster, error free, with no paper work and easy to manage.

Employee leave management is a standalone application that can be easily accessible by staff. and management of an institution. It makes it easy for an employee to request and track their own leave. Administrative department of an institution on the other hand can easily allocate, grant and manage all leave requests. Employee leave management system automatically reflect the request to the relevant superior officer for approval. If the superior officers reject the leave, a reason for rejecting the leave must be entered into the system and the employee who requested the leave will be notified and if the leave request requires a higher superior officer for approval before notifying the employee, the respective officer will be notified. The system will also notify other members of staff that are required to know. This will enable administrative department to administer leave or note to the next applicant, to track and manage the employee leave. For every leave requested by an employee, the system will automatically deduct the applicant leave from total leave and notify all parties involve the total leave taken, the remaining balance and when next the leave will be taken according to policy of an institution. The system can be used to request, approve and generates reports of leave but only on works on intranet.

**CHAPTER 2**

**AIM, OBJECTIVE AND SCOPE OF THE PROJECT**

**2.1 Aim of the project**

Employee Leave Management System is an Intranet-based application titled “Employment management System”, developed using C, C++ and FS. This project is aimed at developing a standalone leave management tool that can used in either an organization or a college and can be accessed throughout the organization and its departments.

**2.2 Objective of the project**

In the existing system, leaves are maintained using the attendance register for staff. The staff needs to submit their leaves manually to their respective authorities. This increases the paperwork & maintaining the records becomes tedious. Maintaining notices in the records also increases the paperwork. The main objective of the proposed system is to reduce the paperwork and help in easier record maintenance by having a particular centralized Database System, where Leaves and Notices are maintained. The proposed system modernizes and automates the existing system. It decreases the paperwork and enables easier record maintenance. It also reduces chances of Data loss. This module intelligently adapts to HR policy of the management & allows easy leave handling for better scheduling of workload. This system will provide a connection between the user (staff) and the database hence enabling quick retrieval of the information without any intervention from the human resource manager. In this system employees can apply for leave online he/she can see his/her balance leave.

**2.3 Scope of this project**

The scope of the project is limited to several processes: handling of employee leave application, managing leave balances, record management administration and leave management and It will generate the reports such as leave trends of the company, employee availability, employee leave balance, leave rejection and leave acceptance. The leave management system is designed in such a way that makes it possible to access it through any web browser program.

The leave management system was designed, developed and implemented taking the distinction of the leave of absence policies and types of leaves and the system is exclusively designed and developed for HR Department in general and the employees leave records section in particular. The employee leave section is solely responsible for keeping the leave and related records of employees and keeping track of their information.

**CHAPTER 3**

**REQIREMENT SPECIFICATION**

**3.1 System Requirements**

The basic requirements for the development of this mini project are as follows

**3.1.1 Hardware Configuration**

1. Processor: Intel core i3 or above
2. Ram:512 MB
3. Hard disk:20 GB

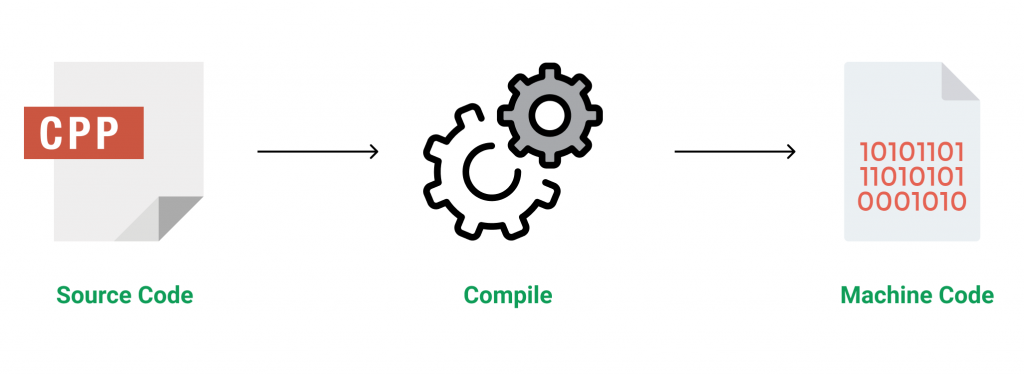
**3.1.2 Software Configuration**

1. Language: C & C++
2. IDE: DEV C++
3. Documentation tool: Microsoft office 2003 or above

**3.2 Development Environment**

**3.2.1 C++**

**C++** is a general-purpose programming language that was developed as an enhancement of the C language to include object-oriented paradigm. It is an imperative and a **compiled** language[7].



**Figure 3.1 Source code compilation**

C++ is a middle-level language rendering it the advantage of programming low-level (drivers, kernels) and even higher-level applications (games, GUI, desktop apps etc.). The basic syntax and code structure of both C and C++ are the same.

Some of the features & key-points to note about the programming language are as follows:

* Simple: It is a simple language in the sense that programs can be broken down into logical units and parts, has a rich library support and a variety of data-types.
* Machine Independent but Platform Dependent: A C++ executable is not platform-independent (compiled programs on Linux won’t run on Windows), however they are machine independent.
* Mid-level language: It is a mid-level language as we can do both systems-programming (drivers, kernels, networking etc.) and build large-scale user applications (Media Players, Photoshop, Game Engines etc.)
* Rich library support: Has a rich library support (Both standard ~ built-in data structures, algorithms etc.) as well 3rd party libraries (e.g. Boost libraries) for fast and rapid development.
* Speed of execution: C++ programs excel in execution speed. Since, it is a compiled language, and also hugely procedural. Newer languages have extra in-built default features such as garbage-collection, dynamic typing etc. which slow the execution of the program overall. Since there is no additional processing overhead like this in C++, it is blazing fast.
* Pointer and direct Memory-Access: C++ provides pointer support which aids users to directly manipulate storage address. This helps in doing low-level programming (where one might need to have explicit control on the storage of variables).
* Object-Oriented: One of the strongest points of the language which sets it apart from C. Object-Oriented support helps C++ to make maintainable and extensible programs. i.e. Large-scale applications can be built. Procedural code becomes difficult to maintain as code-size grows.
* Compiled Language: C++ is a compiled language, contributing to its speed.

**3.2.2 File Structure**

In computing, file system or filesystem (often abbreviated to fs) is a method and data structure that the operating system uses to control how data is stored and retrieved.[1] Without a file system, data placed in a storage medium would be one large body of data with no way to tell where one piece of data stopped and the next began, or where any piece of data was located when it was time to retrieve it [5]. By separating the data into pieces and giving each piece a name, the data is easily isolated and identified. Taking its name from the way a paper-based data management system is named, each group of data is called a "file”. The structure and logic rules used to manage the groups of data and their names is called a "file system”.

A file system consists of two or three layers. Sometimes the layers are explicitly separated, and sometimes the functions are combined. The logical file system is responsible for interaction with the user application. It provides the application program interface (API) for file operations — OPEN, CLOSE, READ, etc., and passes the requested operation to the layer below it for processing. The logical file system "manage[s] open file table entries and per-process file descriptors".[5] This layer provides "file access, directory operations, [and] security and protection".

The second optional layer is the virtual file system. "This interface allows support for multiple concurrent instances of physical file systems, each of which is called a file system implementation". The third layer is the physical file system. This layer is concerned with the physical operation of the storage device (e.g. disk). It processes physical blocks being read or written. It handles buffering and memory management and is responsible for the physical placement of blocks in specific locations on the storage medium. The physical file system interacts with the device drivers or with the channel to drive the storage device.

* + 1. **Files Operation**

A file is an abstract data type. To define a file properly, we need to consider the operations that can be performed on files. Six basic file operations. The OS can provide system calls to create, write, read, reposition, delete, and truncate files.

* Creating a file: Two steps are necessary to create a file.
  + Space in the file system must be found for the file.
  + An entry for the new file must be made in the directory
* Writing a file: To write a file, we make a system call specifying both the name of the file and the information to be written to the file. The system must keep a write pointer to the location in the file where the next write is to take place. The write pointer must be updated whenever a write occurs.
* Reading a file: To read from a file, we use a system call that specifies the name of the file and where (in memory) the next block of the file should be put. The system needs to keep a read pointer to the location in the file where the next read is to take place.
* Because a process is usually either reading from or writing to a file, the current operation location can be kept as a per-process current-file-position pointer.
* Both the read and write operations use this same pointer, saving space and reducing system complexity.
* Repositioning within a file: The directory is searched for the appropriate entry, and the current-file-position pointer is repositioned to a given value. Repositioning within a file need not involve any actual I/O. This file operation is also known as a file seek.
* Deleting a file: To delete a file, we search the directory for the named file. Having found the associated directory entry, we release all file space, so that it can be reused by other files, and erase the directory entry.
* Truncating a file: The user may want to erase the contents of a file but keep its attributes. Rather than forcing the user to delete the file and then recreate it, this function allows all attributes to remain unchanged (except for file length) but lets the file be reset to length zero and its file space released.

These six basic operations comprise the minimal set of required file operations. These primitive operations can then be combined to perform other file operations (i.e., copying). The OS keeps a small table, called the open-file table, containing information about all open files. When a file operation is requested, the file is specified via an index into this table, so no searching is required. When the file is no longer being actively used, it is closed by the process, and the OS removes its entry from the open-file table. Most systems require that the programmer open a file explicitly with the $open () system call before that file can be used. The $open () operation takes a file name and searches the directory, copying the directory entry into the open-file table.

This call can also accept access-mode information (create, read-only, read-write, append-only, and so on). This mode is checked against the file's permissions. If the request mode is allowed, the file is opened for the process. The $open () system call typically returns a pointer to the entry in the open-file table. This pointer, not the actual file name, is used in all I/O operations. The implementation of the $open () and $close () operations is more complicated in an environment where several processes may open the file at the same time. This may occur in a system where several different applications open the same file at the same time.

Typically, the OS uses two levels of internal tables:

* A per-process table. The per-process table tracks all files that a process has open. For instance, the current file pointer for each file is found here. Access rights to the file and accounting information can also be included.
* A system-wide table. Each entry in the per-process table in turn points to a system-wide open-file table. The system-wide table contains process-independent information, such as the location of the file on disk, access dates, and file size. Once a file has been opened by one process, the system-wide table includes an entry for the file.

Typically, the open-file table also has an open count associated with each file to indicate how many processes have e the file open. Each $close () decreases this open count, and when the open count reaches zero, the file is no longer in use, and the file's entry is removed from the open-file table. In summary, several pieces of information are associated with an open file.

**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

The following are the basic functionalities of the Leave management system

* MAIN SCREEN WINDOW

When you start the project from any compiler or by double-clicking the executable.exe file, you’ll see the screen shown below is main screen window of the menu.

* USER MENU

To add user you will need to press number 6 and it will direct you to the user management and it will show you the add user, modify user and show all users.

* ADD USER

If you press number 1 it means you will add new user, after you press number 1 you the program will ask you to enter your ID, name, age, salary, and your experience.

* APPLY LEAVE

If you press number 1 in the main menu it means you will apply a leave, after you press number 1 the program will ask you to enter the employee Id if the employee is exist it will show you the four options such as casual leave, sick leave, study leave, and parental leave.

* SEARCH LEAVE

If you press number 3 in the main menu it means you will SEARCH a leave, after you press number 3 the program will ask you to enter the employee Id if the employee is exist it will show you the information of the employee.

* MODIFY LEAVES

If you press number 4 in the main menu it means you will modify a leave of an employee, after you press number 4 the program will ask you to enter the employee Id if the employee is exist it will show you the four options if what kind of leave you want to modify.

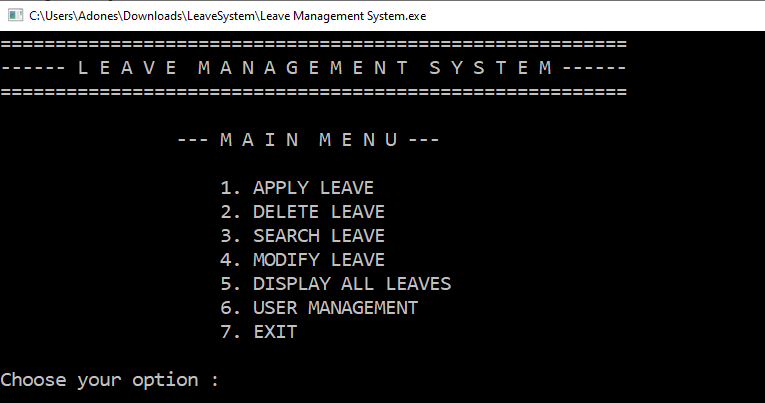
* DISPLAY ALL LEAVES

If you press number 5 in the main menu it means you will modify a leave of an employee, after you press number 5 the program will ask you to enter the employee Id if the employee is exist it will show you the date of the leaves of the employee.

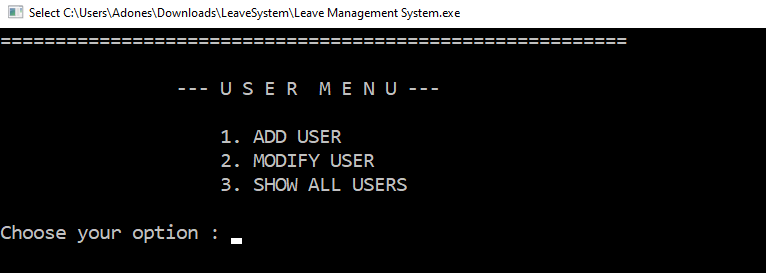
**CHAPTER 5**

**SAMPLE OUTPUT**

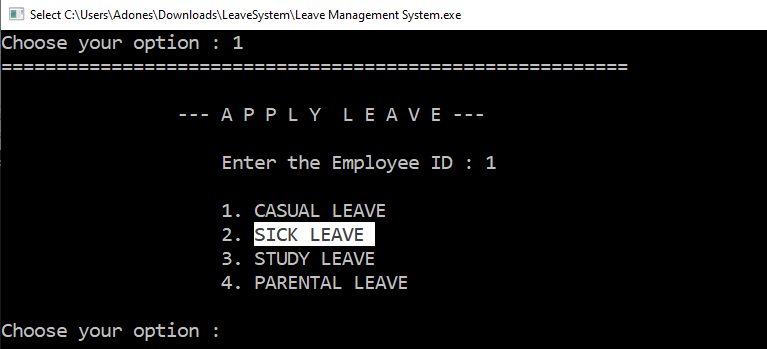
**5.1 Screenshots**



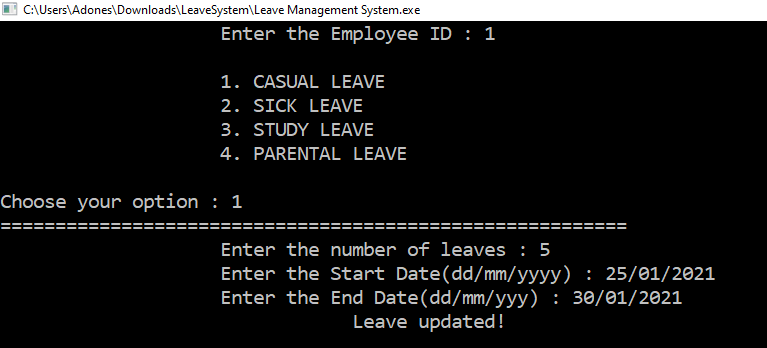
**Figure 5.1 Main Menu**



**Figure 5.2 User Menu**



**Figure 5.3 Apply Leave**



**Figure 5.4 Modify Leave**

**CHAPTER 6**

**CONCLUSION**

Employee leave management system for managing staff leave in higher institution has successfully been developed. The system was developed in line with standalone software model and implemented using web based technologies which include C++ and FS. The system serves as improvement in staff management, maintain accuracy, transparency and highlight the need to integrate advance technology in employee record and welfare management in higher institution. The developed system enables the employees in academic institutions to request and track their leave at their own convenient time in timely manner. Superior officers of the institution and administrative department can create leave policies, check transparency and plan activities ahead of time.

This C++ leave system is purely a school required project. You can download this source code and alter it to meet your client’s needs. This program is a simple employee database for an office or organization where the user can easily save employee records and leave records because it is not time-consuming. Additionally, the user can add employee information, examine a list of employees, and amend and delete employee information. Furthermore, the users will find this project to be simple to use and comprehend.

**BIBLIOGRAPHY**

[1] https://www.codewithc.com/mini-project-in-c-medical-store-management-system/

[2] https://www.codewithc.com

[3] [www.youtube.com](http://www.youtube.com)

[4] [www.quora.com](http://www.quora.com)

[5] [www.wikipedia.org](http://www.wikipedia.org)

[6] <https://www.w3schools.com/cpp/cpp_files.asp>

[7] https://www.geeksforgeeks.org/introduction-to-c-programming-language/