./

Report – Digital Memo

**NAME: - Anitha B**

**SF ID: -265441**

Course Code: <CODE>



Version Number:

Team Members :

Team No:

Module: **C Programming on Multiple platforms - PBL Approach**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SERIAL No.** | **CONTENT** | **PAGE No.** |
| 1 | Problem Statement | 3 |
| 2 | Description | 3 |
| 3 | Requirements | 4 |
| 4 | Architecture | 5-7 |
| 5 | Test Plan/Test cases | 8 |
| 6 | Expected results/Outcomes | 8 |
| 7 | Conclusion | 9 |

**Introduction**

**Problem Statement:**

To design and implement Digital Memo based on the details provided by the user to overcome traditional way of manually maintaining records

**Description:**

Digital Memo is based on concept to generate Personal Daily Records and to add records and update it. Here User can add their daily details safely and it’s not time consuming. Each Section contains high security so there is no chance of data loss. There’s maximum privacy to the recorded files. This system makes easy to store daily records. The whole project is designed in ‘C’ language and different variables and strings have been used for the development of this project. This mini project is easy to operate and easily understood by the users. This is a simple mini project in C, built as a console application without using any graphics features. It is easy to operate and understand by the users. There is no any error and warning contents in the project. The design is so simple that user won’t find it difficult to use and navigate.

In this console application, you can add, view, edit and delete records. Records can be added with many information such as duration of task, name, address, time and date. File handling has been effectively used to keep the records.

In this project, you need to enter a password to view, edit or delete record. Password is not required to view the main menu or to add record. We have used password in other mini projects in code with C; what’s little different in this project is that you can edit the password within the application i.e, after executing or upon running the project

# 

# Requirements:

**Hardware Requirements**

* Ram 512MB or higher.
* Minimum 10MB hard disk space.
* Intel/Pentium powered system.
* Processor speed 1.7 GHZ

**Software Requirements: -**

* Windows 8 or higher versions.
* Visual Studio/Code::Blocks software/Dev-C++.

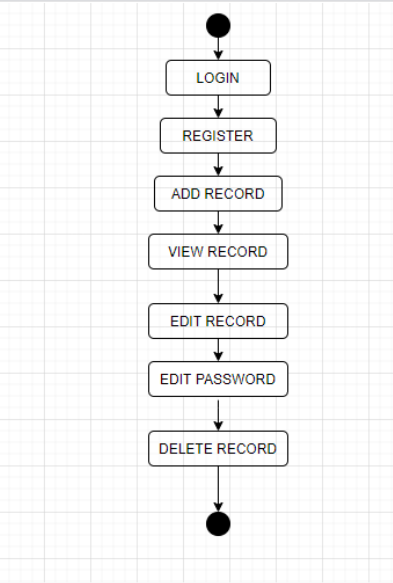
**Functional Requirements: -**

In this project we used the concept of file handling, data structures, pointer and functions.

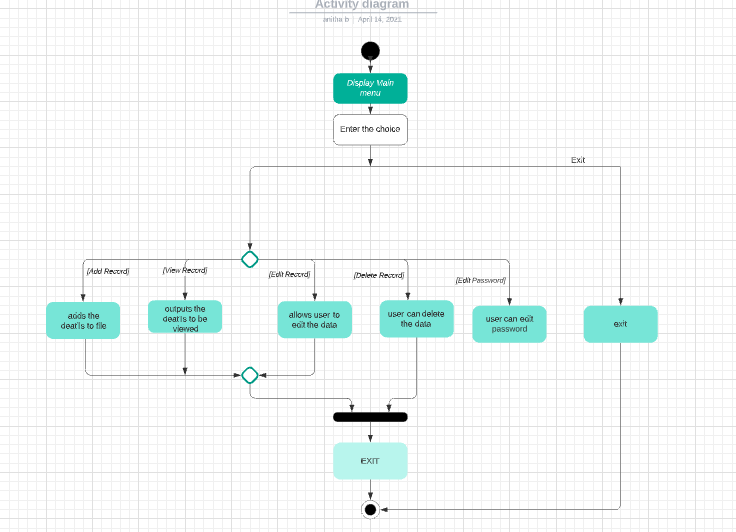
The basic user-defined functions used in this project are listed below:

* Password – contains/manages/handles password protection
* Add record – to add new diary record
* View record – to view added record in list
* Edit record – to modify and update an added record
* Edit password – to modify/change a password
* Delete record – to delete or remove a record permanently from system file.

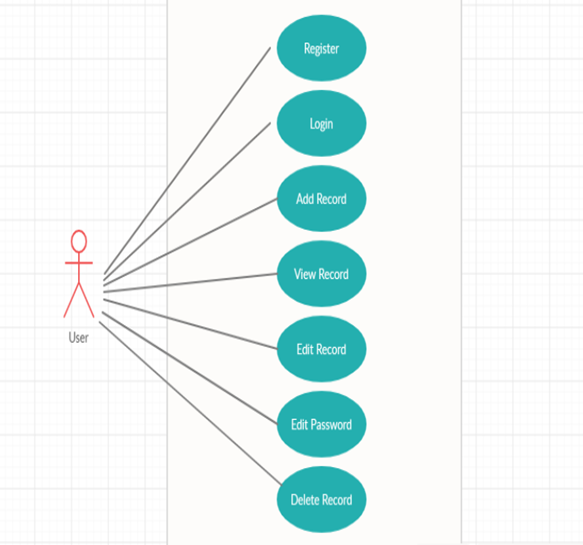
**Architecture:**



**Activity Diagram:**



## **Use Case Diagram:**



**Test Plan/Test cases**

|  |  |  |
| --- | --- | --- |
| **Test case Id** | **Test scenario** | **Test data** |
| 1 | Register the user | username- sam Pass - sam123 |
| 2 | Login the user | username – sam Pass – sam123 |
| 4 | Add Record | I am loving this new diary. |
| 5 | View Record | Sam can view the content of this diary. |
| 6 | Edit Record | Sam can edit the records if and when required. |
| 7 | Edit Password | Current password – sam123 New password – sam1234 |
| 8 | Delete Record | After entering password sam can delete the records. |

**EXPECTED OUTCOMES: -**

|  |  |
| --- | --- |
| **TEST CASE ID** | **EXPECTED RESULT** |
| 1 | Sam was registered successfully |
| 2 | Sam logged – in successfully |
| 3 | Entered data recorded successfully. |
| 4 | Sam can view – I am loving this new diary. |
| 5 | Sam can edit record |
| 6 | Sam successfully changes the password |
| 7 | Sam successfully deleted the record after entering correct password |

**Conclusion**

Based on the result of this research, it can be concluded that this system helps to maintain personal notebook easily. This project is only a humble to maintain the personal notebook. This project can be used to fulfill the requirements of the user who want to maintain the notes personally. This project receives the input details and store them for future use. With this Digital Memo the details can be accessed smoothly. The scope of this project is justifiable as it reduces manual work and is user friendly. This report gives an overview of all that one needs to get acquainted with in order to work with software project documentation specifically in the field of c language.

It was a diverse and enriching experience to learn about c language. This project will teach us the proper use of file handling, data structure, pointers and functions. I have achieved the user friendly, maintaining of personal notebook through this code can be managed efficiently. This will lead to an error free system of maintaining personal notebook. This helps in efficient management of time and user can concentrate on other activities. On the whole, this project helped to enhance technical skills, team work, documenting skills.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver. Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**