**MINI-PROJECT REPORT**

**Personal Diary Management System**



**NAME: - Abhishek Gawala**

**SF ID: -104245**

**TRACK: -S2**

**TABLE OF CONTENTS**

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

**INTRODUCTION:**

**PROBLEM STATEMENT**

In this project we are going to make a personal diary management system. This system would help the user to create, edit and view the diary created by the user on this system.

**DESCRIPTION:**

**Personal Diary Management System** is a console application used for creating, viewing and managing the personal diary of a user. In this project, user can keep their personal record like they do in a diary. They can update, add and view the records of the diary. User can keep records of the important things they do in their daily life, like meetings and the activities which they do in their everyday life.

Their diary will be password protected hence they can freely share their experiences without the fear of getting read by someone else.

**REQUIREMENTS**

**Software Used**: Code::Blocks IDE version 20.3 using GCC compiler.

**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.

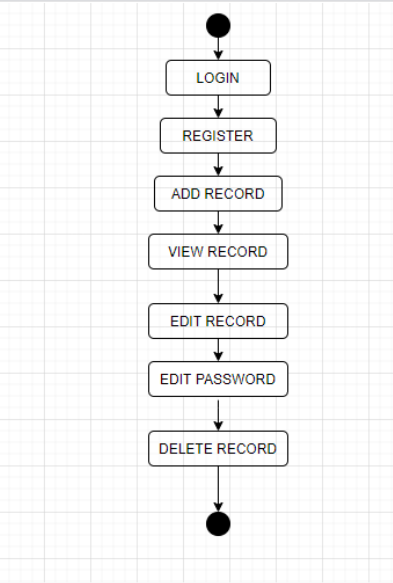
**Functional Requirements: -**

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

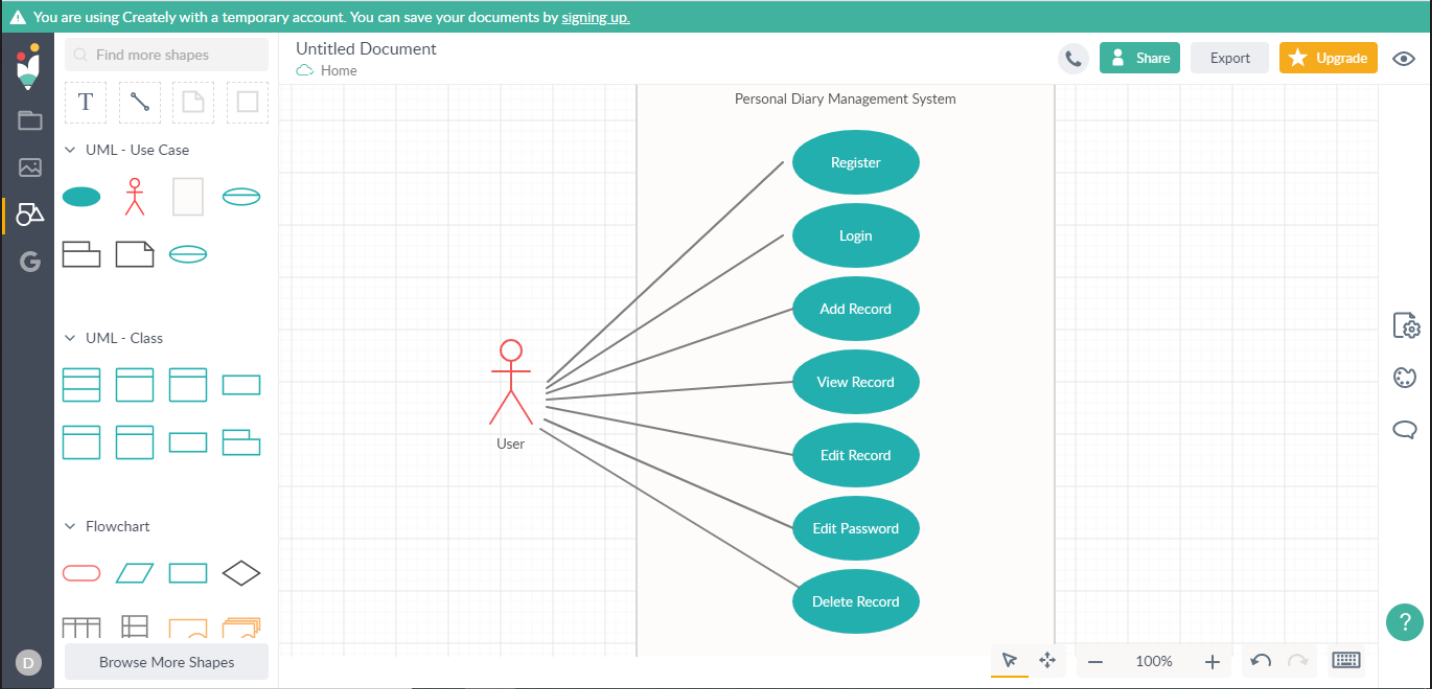
The basic user-defined functions used in this project are listed below:

* 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**



**USE-CASE DIAGRAM**



**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: -**

With the help of this personal diary management system, user can register, login, create, edit and view the created diary. User can also change the password. User gets the opportunity to save their personal information with the help of the password. This password protected system helps user to write their personal experiences and views which they otherwise don’t freely do in case of the traditional personal diary. This system therefore provides an edge with respect to the conventional manual hand-written diary.