Table of contents

Introduction

In the digital age, managing personal information efficiently is crucial. A personal diary management application helps individuals to keep track of their daily activities, thoughts, and important events. This report outlines the development of a console-based personal diary management application using the C programming language. The application aims to provide a simple, user-friendly,lightweight interface for users to create, read, update, and delete diary entries.

Project Overview

Personal Diary Management system is a lightweight, console based application which is password protected to ensure safety of private data. Its small size of 70 Kilo Bytes is ideal for it to be used with any machine , even those with low powered CPUs. Besides supporting CRUD functions (Create, read, update, delete) , it keeps track of records and maintains privacy by requiring a password in order to access and do operations. In order to create this simple application, only around 600 lines of code were needed.

Project Features and Implementation

This lightweight console based journaling application supports basic CRUD functions like-

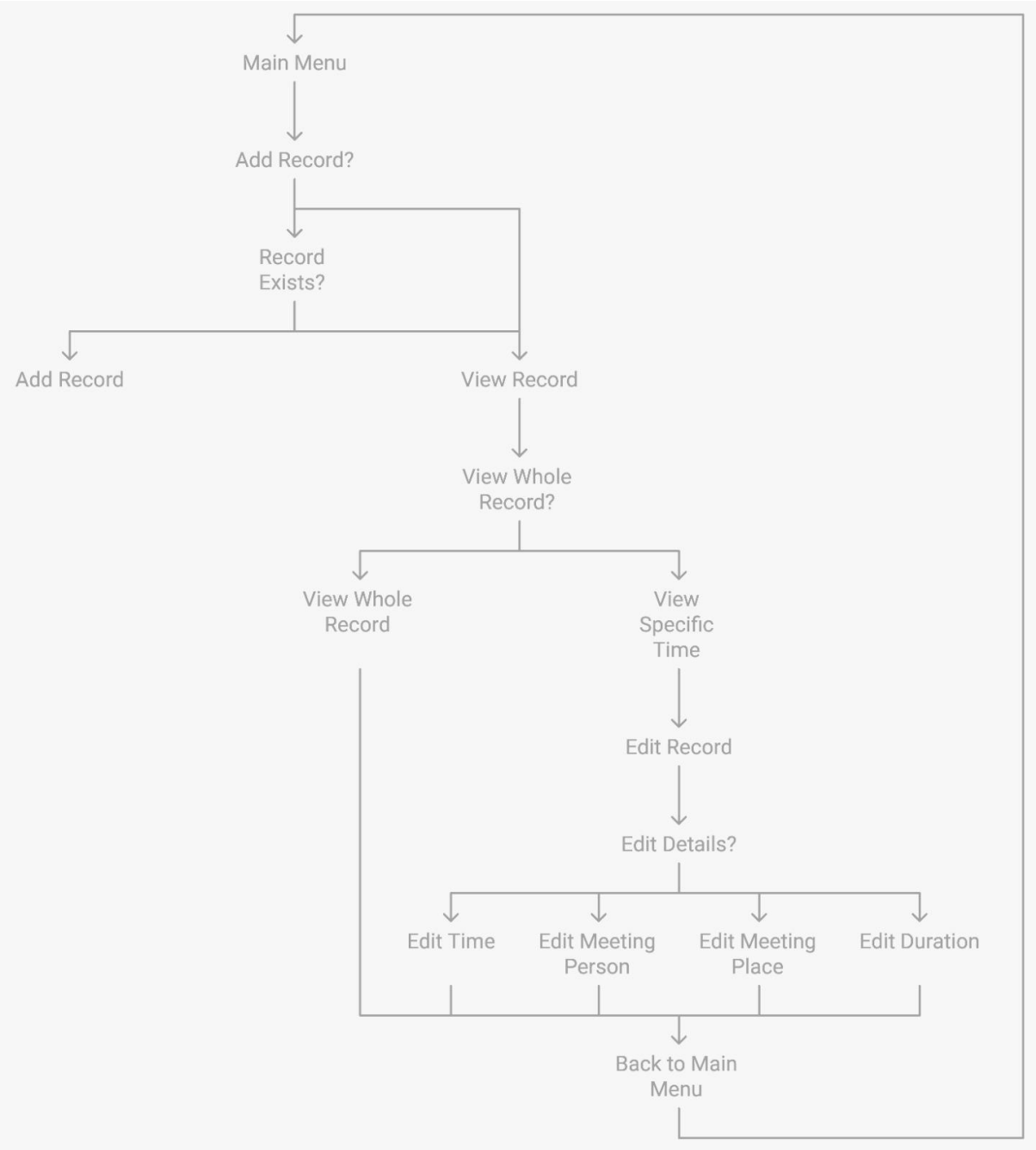
1. **Addition of record**
2. **Viewing records**
3. **Editing records**
4. **Deleting records**
5. **Setting Password**
6. **Editing password**

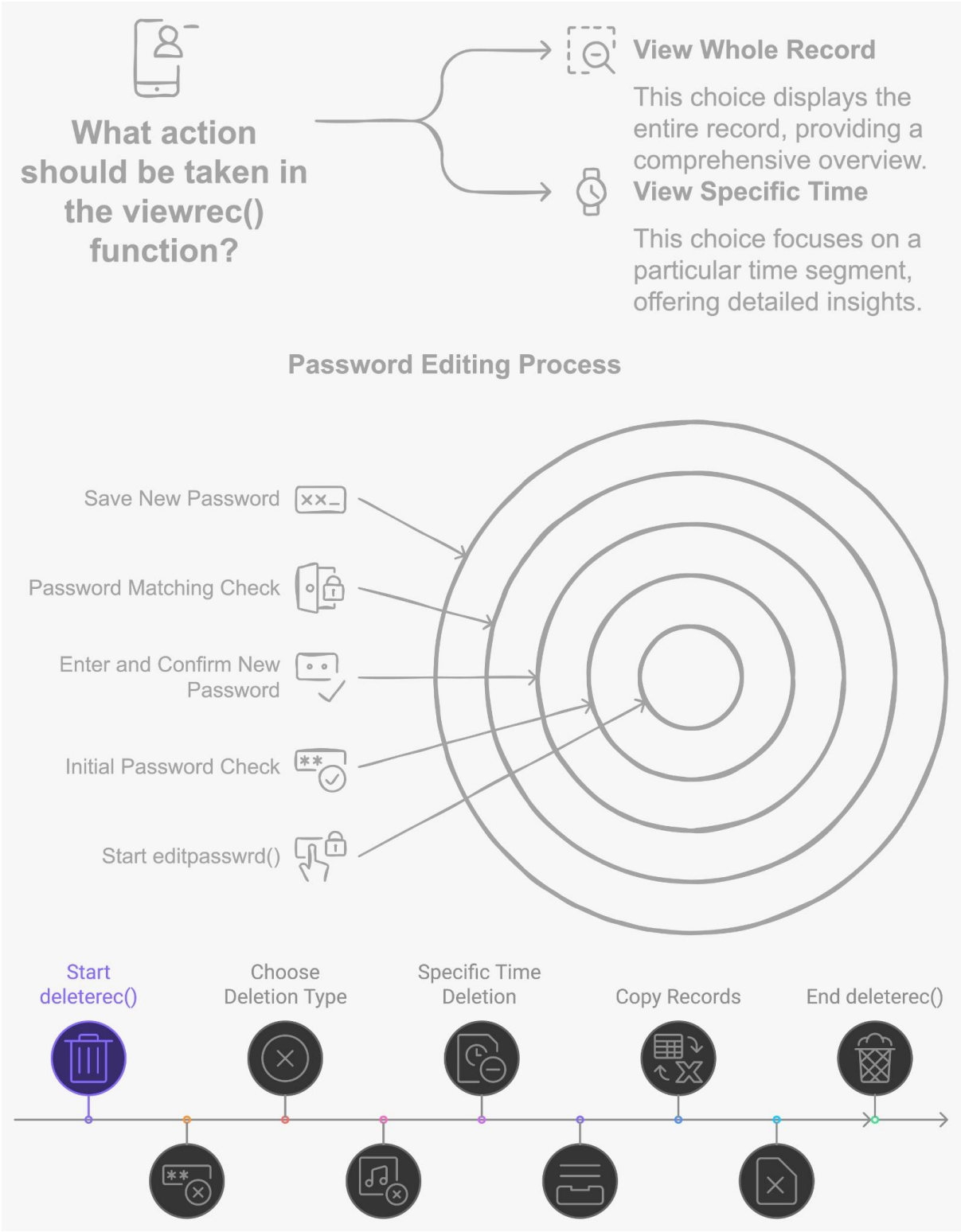
Used Libraries and their intended function

Function name	Personnel who wrote the function	Intended Purpose
addrec	Riyadh Ahammed	Addition of records
viewrec	Yasir Obayed	Viewing saved records
editrec	Riyadh Ahammed	Editing existing records
Deleterec	Rafi Saad	Deleting saved records
passwrd	Rafi Saad Rashad Rafid	Protecting records from intrusion
editpasswrd	Yasir Obayed Rashad Rafid	Setting a new password to access records

Code

Thought process:




What action should be taken in the viewrec() function?
View Whole Record
This choice displays the entire record, providing a comprehensive overview.
View Specific Time
This choice focuses on a particular time segment, offering detailed insights.
Password Editing Process
Save New Password
Password Matching Check
Enter and Confirm New Password
Initial Password Check
Start editpasswrd()
Start deleterec()
Choose Deletion Type
Specific Time Deletion
Copy Records
End deleterec()

Main Code:

```
#include<windows.h>
#include<stdio.h>
void addrec();
void viewrec();
void editrec();
void deleterec();
int passwrd();
void editpasswrd();
struct record
{
   char time[6];
   char name[30];
   char place[25];
   char duration[10];
   char note[500];
} ;
void addrec( )
{
   system("cls");
```

```
FILE *fp ;
char another = 'Y' ,time[10];
struct record e ;
char filename[15];
int choice;
printf("\n\n\t\t*****************************\n");
printf("\t\t* WELCOME TO THE ADD MENU *");
printf("\n\t\t*****************************\n\n");
printf("\n\n\tENTER DATE OF YOUR RECORD:[yyyy-mm-dd]:");
fflush(stdin);
gets(filename);
fp = fopen (filename, "ab+" ) ;
if ( fp == NULL )
{
   fp=fopen(filename,"wb+");
   if(fp==NULL)
   {
      printf("\nSYSTEM ERROR...");
      printf("\nPRESS ANY KEY TO EXIT");
      getch();
      return ;
```

```
    }
}
while ( another == 'Y'|| another=='y' )
{
    choice=0;
    fflush(stdin);
    printf ( "\n\tENTER TIME:[hh:mm]:");
    scanf("%s",time);
    rewind(fp);
    while(fread(&e,sizeof(e),1,fp)==1)
    {
        if(strcmp(e.time,time)==0)
        {
            printf("\n\tTHE RECORD ALREADY EXISTS.\n");
            choice=1;
        }
    }
    if(choice==0)
    {
        strcpy(e.time,time);
        printf("\tENTER NAME:");
```

```
        fflush(stdin);
        gets(e.name);
        fflush(stdin);
        printf("\tENTER PLACE:");
        gets(e.place);
        fflush(stdin);
        printf("\tENTER DURATION:");
        gets(e.duration);
        fflush(stdin);
        printf("\tNOTE:");
        gets(e.note);
        fwrite ( &e, sizeof ( e ), 1, fp ) ;
        printf("\nYOUR RECORD IS ADDED...\n");
    }
    printf ( "\n\tADD ANOTHER RECORD...(Y/N) " ) ;
    fflush ( stdin ) ;
    another = getchar( ) ;
}
fclose ( fp ) ;
printf("\n\n\tPRESS ANY KEY TO EXIT...");
getch();
```

```
}
void viewrec( )
{
   FILE *fpte ;
   system("cls");
   struct record customer ;
   char time[6],choice,filename[14];
   int ch;
   printf("\n\n\t\t*********************************\n");
   printf("\t\t* HERE IS THE VIEWING MENU *");
   printf("\n\t\t*********************************\n\n");
   choice=passwrd();
   if(choice!=0)
   {
      return ;
   }
   do
   {
      printf("\n\tENTER THE DATE OF RECORD TO BE VIEWED:[yyyy-
mm-dd]:");
      fflush(stdin);
      gets(filename);
```

```
fpte = fopen ( filename, "rb" ) ;
if ( fpte == NULL )
{
   puts ( "\nTHE RECORD DOES NOT EXIST...\n" ) ;
   printf("PRESS ANY KEY TO EXIT...");
   getch();
   return ;
}
system("cls");
printf("\n\tHOW WOULD YOU LIKE TO VIEW:\n");
printf("\n\t1.WHOLE RECORD OF THE DAY.");
printf("\n\t2.RECORD OF FIX TIME.");
printf("\n\t\tENTER YOUR CHOICE:");
scanf("%d",&ch);
switch(ch)
{
case 1:
   printf("\nTHE WHOLE RECORD FOR %s IS:",filename);
   while ( fread ( &customer, sizeof ( customer ), 1, fpte ) == 1 )
   {
      printf("\n");
```

```
        printf("\nTIME: %s",customer.time);
        printf("\nMEETING WITH: %s",customer.name);
        printf("\nMEETING AT: %s",customer.place);
        printf("\nDURATION: %s",customer.duration);
        printf("\nNOTE: %s",customer.note);
        printf("\n");
    }
    break;
case 2:
    fflush(stdin);
    printf("\nENTER TIME:[hh:mm]:");
    gets(time);
    while ( fread ( &customer, sizeof ( customer ), 1, fpte ) == 1 )
    {
        if(strcmp(customer.time,time)==0)
        {
            printf("\nYOUR RECORD IS:");
            printf("\nTIME: %s",customer.time);
            printf("\nMEETING WITH: %s",customer.name);
            printf("\nMEETING AT: %s",customer.place);
            printf("\nDUARATION: %s",customer.duration);
```

```
                printf("\nNOTE: %s",customer.note);
             }
          }
          break;
       default:
          printf("\nYOU TYPED SOMETHING ELSE...\n");
          break;
       }
       printf("\n\nWOULD YOU LIKE TO CONTINUE
VIEWING...(Y/N):");
       fflush(stdin);
       scanf("%c",&choice);
    }
    while(choice=='Y'||choice=='y');
    fclose ( fpte ) ;
    return ;
}
void editrec()
{
    system("cls");
    FILE *fpte ;
    struct record customer ;
```

```
   char time[6],choice,filename[14];
   int num,count=0;
   printf("\n\n\t\t*********************************\n");
   printf("\t\t* WELCOME TO THE EDITING MENU *");
   printf("\n\t\t*********************************\n\n");
   choice=passwrd();
   if(choice!=0)
   {
      return ;
   }
   do
   {
      printf("\n\tENTER THE DATE OF RECORD TO BE EDITED:[yyyy-
mm-dd]:");
      fflush(stdin);
      gets(filename);
      printf("\n\tENTER TIME:[hh:mm]:");
      gets(time);
      fpte = fopen ( filename, "rb+" ) ;
      if ( fpte == NULL )
      {
         printf( "\nRECORD DOES NOT EXISTS:" ) ;
```

```
    printf("\nPRESS ANY KEY TO GO BACK");
    getch();
    return;
}
while ( fread ( &customer, sizeof ( customer ), 1, fpte ) == 1 )
{
    if(strcmp(customer.time,time)==0)
    {
        printf("\nYOUR OLD RECORD WAS AS:");
        printf("\nTIME: %s",customer.time);
        printf("\nMEETING WITH: %s",customer.name);
        printf("\nMEETING AT: %s",customer.place);
        printf("\nDURATION: %s",customer.duration);
        printf("\nNOTE: %s",customer.note);
        printf("\n\n\t\tWHAT WOULD YOU LIKE TO EDIT..");
        printf("\n1.TIME.");
        printf("\n2.MEETING PERSON.");
        printf("\n3.MEETING PLACE.");
        printf("\n4.DURATION.");
        printf("\n5.NOTE.");
        printf("\n6.WHOLE RECORD.");
```

```
printf("\n7.GO BACK TO MAIN MENU.");
do
{
   printf("\n\tENTER YOUR CHOICE:");
   fflush(stdin);
   scanf("%d",&num);
   fflush(stdin);
   switch(num)
   {
   case 1:
      printf("\nENTER THE NEW DATA:");
      printf("\nNEW TIME:[hh:mm]:");
      gets(customer.time);
      break;
   case 2:
      printf("\nENTER THE NEW DATA:");
      printf("\nNEW MEETING PERSON:");
      gets(customer.name);
      break;
   case 3:
      printf("\nENTER THE NEW DATA:");
```

```
    printf("\nNEW MEETING PLACE:");
    gets(customer.place);
    break;
  case 4:
    printf("\nENTER THE NEW DATA:");
    printf("\nDURATION:");
    gets(customer.duration);
    break;
  case 5:
    printf("ENTER THE NEW DATA:");
    printf("\nNOTE:");
    gets(customer.note);
    break;
  case 6:
    printf("\nENTER THE NEW DATA:");
    printf("\nNEW TIME:[hh:mm]:");
    gets(customer.time);
    printf("\nNEW MEETING PERSON:");
    gets(customer.name);
    printf("\nNEW MEETING PLACE:");
    gets(customer.place);
```

```
            printf("\nDURATION:");
            gets(customer.duration);
            printf("\nNOTE:");
            gets(customer.note);
            break;
         case 7:
            printf("\nPRESS ANY KEY TO GO BACK...\n");
            getch();
            return ;
            break;
         default:
            printf("\nYOU TYPED SOMETHING ELSE...TRY
AGAIN\n");
            break;
         }
      }
      while(num<1||num>8);
      fseek(fpte,-sizeof(customer),SEEK_CUR);
      fwrite(&customer,sizeof(customer),1,fpte);
      fseek(fpte,-sizeof(customer),SEEK_CUR);
      fread(&customer,sizeof(customer),1,fpte);
      choice=5;
```

```
            break;
        }
    }
    if(choice==5)
    {
        system("cls");
        printf("\n\t\tEDITING COMPLETED...\n");
        printf("--------------------\n");
        printf("THE NEW RECORD IS:\n");
        printf("--------------------\n");
        printf("\nTIME: %s",customer.time);
        printf("\nMEETING WITH: %s",customer.name);
        printf("\nMEETING AT: %s",customer.place);
        printf("\nDURATION: %s",customer.duration);
        printf("\nNOTE: %s",customer.note);
        fclose(fpte);
        printf("\n\n\tWOULD YOU LIKE TO EDIT ANOTHER
RECORD.(Y/N)");
        scanf("%c",&choice);
        count++;
    }
    else
```

```
        {
            printf("\nTHE RECORD DOES NOT EXIST::\n");
            printf("\nWOULD YOU LIKE TO TRY AGAIN...(Y/N)");
            scanf("%c",&choice);
        }
    }
    while(choice=='Y'||choice=='y');
    fclose ( fpte ) ;
    if(count==1)
        printf("\n%d FILE IS EDITED...\n",count);
    else if(count>1)
        printf("\n%d FILES ARE EDITED..\n",count);
    else
        printf("\nNO FILES EDITED...\n");
    printf("\tPRESS ENTER TO EXIT EDITING MENU.");
    getch();
}
int passwrd()
{
    char pass[15]= {0},check[15]= {0},ch;
    FILE *fpp;
```

```
int i=0,j;
printf("::FOR SECURITY PURPOSE::");
printf("::ONLY THREE TRIALS ARE ALLOWED::");
for(j=0; j<3; j++)
{
   i=0;
   printf("\n\n\tENTER THE PASSWORD:");
   pass[0]=getch();
   while(pass[i]!='\r')
   {
      if(pass[i]=='\b')
      {
         i--;
         printf("\b");
         printf(" ");
         printf("\b");
         pass[i]=getch();
      }
      else
      {
         printf("*");
```

```
            i++;
            pass[i]=getch();
        }
    }
    pass[i]='\0';
    fpp=fopen("SE","r");
    if (fpp==NULL)
    {
        printf("\nERROR WITH THE SYSTEM FILE...[FILE
MISSING]\n");
        getch();
        return 1;
    }
    else
        i=0;
    while(1)
    {
        ch=fgetc(fpp);
        if(ch==EOF)
        {
            check[i]='\0';
            break;
```

```
        }
        check[i]=ch-5;
        i++;
      }
      if(strcmp(pass,check)==0)
      {
        printf("\n\n\tACCESS GRANTED...\n");
        return 0;
      }
      else
      {
        printf("\n\n\tWRONG PASSWORD..\n\n\tACCESS
DENIED...\n");
      }
   }
   printf("\n\n\t::YOU ENTERED WRONG PASSWORD::YOU ARE
NOT ALLOWED TO ACCESS ANY FILE::\n\n\tPRESS ANY KEY TO GO
BACK...");
   getch();
   return 1;
}
void editpasswrd()
```

```
{
   system("cls");
   printf("\n");
   char pass[15]= {0},confirm[15]= {0},ch;
   int choice,i,check;
   FILE *fp;
   fp=fopen("SE","rb");
   if(fp==NULL)
   {
      fp=fopen("SE","wb");
      if(fp==NULL)
      {
         printf("SYSTEM ERROR...");
         getch();
         return ;
      }
      fclose(fp);
      printf("\nSYSTEM RESTORED...\nYOUR PASSWORD IS
'ENTER'\n PRESS ENTER TO CHANGE PASSWORD\n\n");
      getch();
   }
   fclose(fp);
```

```
check=passwrd();
if(check==1)
{
   return ;
}
do
{
   if(check==0)
   {
      i=0;
      choice=0;
      printf("\n\n\tENTER THE NEW PASSWORD:");
      fflush(stdin);
      pass[0]=getch();
      while(pass[i]!='\r')
      {
         if(pass[i]=='\b')
         {
            i--;
            printf("\b");
            printf(" ");
```

```
        printf("\b");
        pass[i]=getch();
    }
    else
    {
        printf("*");
        i++;
        pass[i]=getch();
    }
}
pass[i]='\0';
i=0;
printf("\n\tCONFIRM PASSWORD:");
confirm[0]=getch();
while(confirm[i]!='\r')
{
    if(confirm[i]=='\b')
    {
        i--;
        printf("\b");
        printf(" ");
```

```
        printf("\b");
        confirm[i]=getch();
    }
    else
    {
        printf("*");
        i++;
        confirm[i]=getch();
    }
}
confirm[i]='\0';
if(strcmp(pass,confirm)==0)
{
    fp=fopen("SE","wb");
    if(fp==NULL)
    {
        printf("\n\t\tSYSTEM ERROR");
        getch();
        return ;
    }
    i=0;
```

```
            while(pass[i]!='\0')
            {
               ch=pass[i];
               putc(ch+5,fp);
               i++;
            }
            putc(EOF,fp);
            fclose(fp);
         }
         else
         {
            printf("\n\tTHE NEW PASSWORD DOES NOT MATCH.");
            choice=1;
         }
      }
   }
   while(choice==1);
   printf("\n\n\tPASSWORD CHANGED...\n\n\tPRESS ANY KEY TO
GO BACK...");
   getch();
}
void deleterec( )
```

```
{
   system("cls");
   FILE *fp,*fptr ;
   struct record file ;
   char filename[15],another = 'Y' ,time[10];;
   int choice,check;
   printf("\n\n\t\t************************\n");
   printf("\t\t* WELCOME TO DELETE MENU*");
   printf("\n\t\t************************\n\n");
   check = passwrd();
   if(check==1)
   {
      return ;
   }
   while ( another == 'Y' )
   {
      printf("\n\n\tHOW WOULD YOU LIKE TO DELETE.");
      printf("\n\n\t#DELETE WHOLE RECORD\t\t\t[1]");
      printf("\n\t#DELETE A PARTICULAR RECORD BY TIME\t[2]");
      do
      {
```

```
        printf("\n\t\tENTER YOU CHOICE:");
        scanf("%d",&choice);
        switch(choice)
        {
        case 1:
           printf("\n\tENTER THE DATE OF RECORD TO BE
DELETED:[yyyy-mm-dd]:");
           fflush(stdin);
           gets(filename);
           fp = fopen (filename, "wb" ) ;
           if ( fp == NULL )
           {
              printf("\nTHE FILE DOES NOT EXISTS");
              printf("\nPRESS ANY KEY TO GO BACK.");
              getch();
              return ;
           }
           fclose(fp);
           remove(filename);
           printf("\nDELETED SUCCESFULLY...");
           break;
        case 2:
```

```
        printf("\n\tENTER THE DATE OF RECORD:[yyyy-mm-dd]:");
        fflush(stdin);
        gets(filename);
        fp = fopen (filename, "rb" ) ;
        if ( fp == NULL )
        {
           printf("\nTHE FILE DOES NOT EXISTS");
           printf("\nPRESS ANY KEY TO GO BACK.");
           getch();
           return ;
        }
        fptr=fopen("temp","wb");
        if(fptr==NULL)
        {
           printf("\nSYSTEM ERROR");
           printf("\nPRESS ANY KEY TO GO BACK");
           getch();
           return ;
        }
        printf("\n\tENTER THE TIME OF RECORD TO BE
DELETED:[hh:mm]:");
        fflush(stdin);
```

```
            gets(time);
            while(fread(&file,sizeof(file),1,fp)==1)
            {
               if(strcmp(file.time,time)!=0)
                  fwrite(&file,sizeof(file),1,fptr);
            }
            fclose(fp);
            fclose(fptr);
            remove(filename);
            rename("temp",filename);
            printf("\nDELETED SUCCESFULLY...");
            break;
         default:
            printf("\n\tYOU ENTERED WRONG CHOICE");
            break;
         }
      }
      while(choice<1||choice>2);
      printf("\n\tDO YOU LIKE TO DELETE ANOTHER
RECORD.(Y/N):");
      fflush(stdin);
      scanf("%c",&another);
```

```
    }
    printf("\n\n\tPRESS ANY KEY TO EXIT...");
    getch();
}
int main()
{
    int ch;
    printf("\n\n\t**********************************\n");
    printf(    "\t*PASSWORD PROTECTED PERSONAL DIARY*\n");
    printf(    "\t**********************************");
    while(1)
    {
        printf("\n\n\t\tMAIN MENU:");
        printf("\n\n\tADD RECORD\t[1]");
        printf("\n\tVIEW RECORD\t[2]");
        printf("\n\tEDIT RECORD\t[3]");
        printf("\n\tDELETE RECORD\t[4]");
        printf("\n\tEDIT PASSWORD\t[5]");
        printf("\n\tEXIT\t\t[6]");
        printf("\n\n\tENTER YOUR CHOICE:");
        scanf("%d",&ch);
```

```
switch(ch)
{
case 1:
   addrec();
   break;
case 2:
   viewrec();
   break;
case 3:
   editrec();
   break;
case 4:
   deleterec();
   break;
case 5:
   editpasswrd();
   break;
case 6:
   printf("\n\n\t\tTHANK YOU FOR USING OUR SOFTWARE ");
   getch();
   exit(0);
```

```
        default:
            printf("\nYOU ENTERED WRONG CHOICE..");
            printf("\nPRESS ANY KEY TO TRY AGAIN");
            getch();
            break;
        }
        system("cls");
    }
    return 0;
}
```

Conclusion

The console-based personal diary management application developed using the C programming language provides a simple and efficient way to manage personal diary entries. By implementing basic CRUD operations and ensuring data persistence through file handling, the application meets the primary objectives of the project. The user-friendly interface makes it easy for users to navigate and operate the application, making it a valuable tool for personal information management.

ture enhancements could include adding encryption for data security, implementing a graphical user interface (GUI), and providing cloud storage options for data backup and synchronization.

NORTH SOUTH UNIVERSITY

Center of Excellence in Higher Education

Lab Report

Course code : CSE115L

Submitted By

Student name(s) :

Rafi Saad (2422435642)
Rashad Rafid Alamgir (2421190642)
Riyadh Ahammed Desh (2423292642)
Yasir Obayed (2423524642)

Section : 07

Dept. of Computer Science and Engineering, NSU

Submitted on 5 December,2024