



MINI Project dairy management

Computer Applications (Uttaranchal University)



Scan to open on Studocu

MINI PROJECT SYNOPSIS

ON

“DAIRY MANAGEMENT SYSTEM”

Submitted in the partial fulfilment for the award of degree of
Bachelor of Science (Information Technology)



UTTARANCHAL UNIVERSITY
Session 2021-22

Under the Supervision of
Mr. Praveen Shah
Assistant Professor
(Department of Computer Applications)

Submitted By:
Saurabh Singh
B.Sc.(IT) Ist Semester
Roll No.: 39

ACKNOWLEDGEMENT

First and for most I am ever grateful to God to whom I owe my life. I would also like to thank my parents for giving me the opportunity to study at Uttarakhand University, Dehradun. I wish to express my deep sense of gratitude to our Project Mentor **Mr. Praveen Shah** for his valuable guidance to prepare the project and in assembling the project material. I am very thankful for his faithful blessings and for providing necessary and related facilities required for our computer project file. In last I also want to thank those directly or indirectly took interest to complete my project file.

Saurabh Singh
B. Sc(IT)
Ist Semester

DECLARATION

I hereby declare that the project report titled **"Dairy management"** is submitted by **Saurabh Singh** to Uttaranchal Institute of Management. The project was done under the guidance of Mr. Praveen Shah. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this university or any other university or institute.

Saurabh Singh
B. Sc-IT
Ist Semester

CERTIFICATE OF ORIGINALITY

This is to certify that the project entitled **“Dairy Management System”** by **Saurabh Singh** has been submitted in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Science (Information Technology)** from Uttarakhand University, Dehradun. The results embodied in this project have not been submitted to any other University or Institution for the record of any degree.

Mr. Praveen Shah
Assistant Professor
(Project Mentor)

INDEX

S.NO	TITLE	PAGE NO.
1.	ACKNOWLEDGEMENT	ii
2.	DECLARATION	iii
3.	CERTIFICATE OF ORIGINALITY	iv
4.	INTRODUCTION	06
5.	OBJECTIVE	08
6.	SYSTEM ANALYSIS AND REQUIREMENT	09
a.	○ EXISTING SYSTEM	09
b.	○ NEED FOR NEW SYSTEM	10
c.	○ SOFTWARE AND HARDWARE USED	
7.	MODULES	11
8.	SYSTEM DESIGN	17
9.	CODING	18
10.	OUTPUTS	19
11.	TESTING & SECURITY TESTING	
12.	FUTURE SCOPE	
13.	BIBLIOGRAPHY	

INTRODUCTION

We are creating a Window application for “**Dairy management system**” In C language by using turbo C7 & Dev C++.

- ❖ The mission of the Milk Production House Project is to create to communication between rural area people and dairy management.
- ❖ Our goal is to develop this application to encourage our dairy industry.
- ❖ Dairy milk management system is a software application to maintain day to day transactions in a Milk dairy.
- ❖ This software helps to register all the suppliers, Buyer details, purchase, Sales details.
- ❖ Our plan is to collect the Milk from many different sources and distribute the same to the Milk buyers.

OBJECTIVE

- ❖ Dairy management system is user friendly software.
- ❖ Reduces the time and work efforts by humans and to automate the dairy.
- ❖ User can see the previous transactions and view milk rates.
- ❖ Easy to maintain and store records and to generate bills.
- ❖ Records or the data of a user can be easily fetched.

SYSTEM ANALYSIS

EXISTING SYSTEM

When we analysed that in the management of dairy is working with the manual method. And we all know that the manual system has many disadvantages. Some are mentioned below

- ❖ The manual system requires more time for processing.
- ❖ The manual system has more chances of error.
- ❖ Difficult to maintain the records.
- ❖ Manual system is costly.
- ❖ Immediate response to the queries of the customer is difficult and time consuming.
- ❖ More man power is needed.

PROPOSED SYSTEM

As we have analysed the

- ❖ Requires less time for completion of any work.
- ❖ Decrease the chances of error.
- ❖ Should work smoothly and very fast.
- ❖ Time saving and man power saving.
- ❖ User friendly anyone having computer knowledge can handle it easily.
- ❖ Suitable for computerised data entry. Maintaining dairy information, staff & customer information, Milk rates information.

SOFTWARE & HARDWARE USED

Software used

Dev-C++

Hardware used

Processor – intel core i5 11600KF 4.2ghz

RAM – 8GB

Hard Disk – minimum space available 2-3 Gb(512g SSD used)

Operating System –Windows/Linux

Technology Used

C Programming

MODULES

We are categorising the modules in the three modules

❖ ADMIN

The admin has the unique login and password which completely controls the working .Admin will verify the reports entered by staff.

- 1.Login**
- 2.Manage Milk Rates**
- 3.Manage Staff**
- 4.Manage Seller**

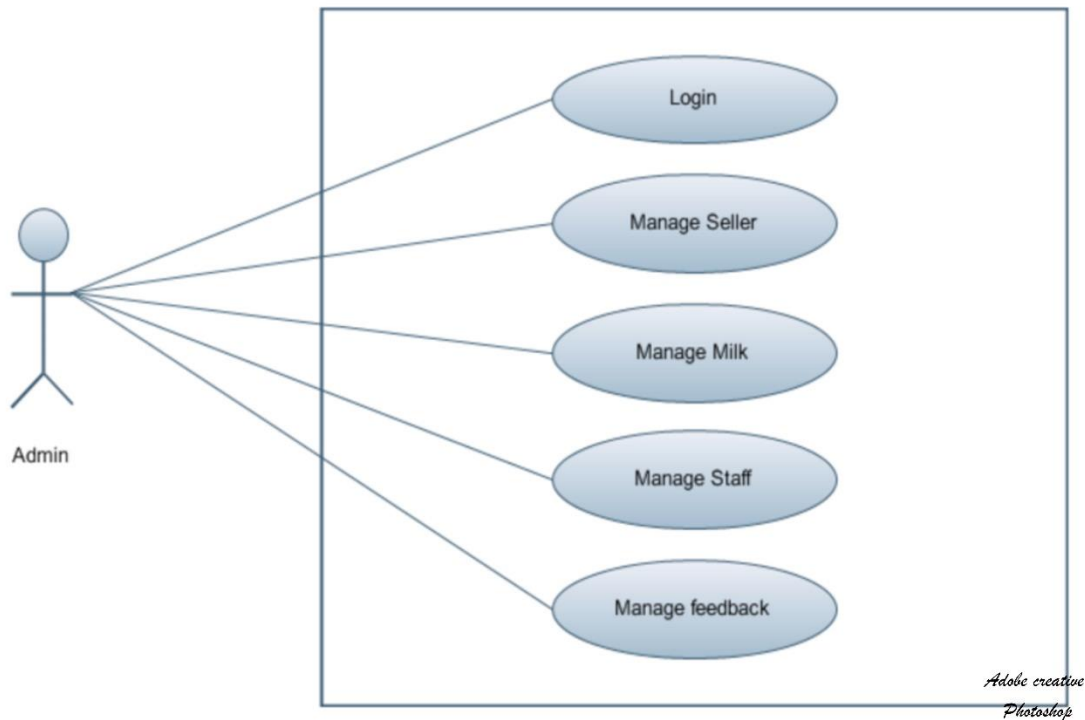
❖ STAFF

- 1.Login**
- 2.Seller Registration**
- 3.Manage Feedback**
- 4.Insert milk details**
- 5.Seller**

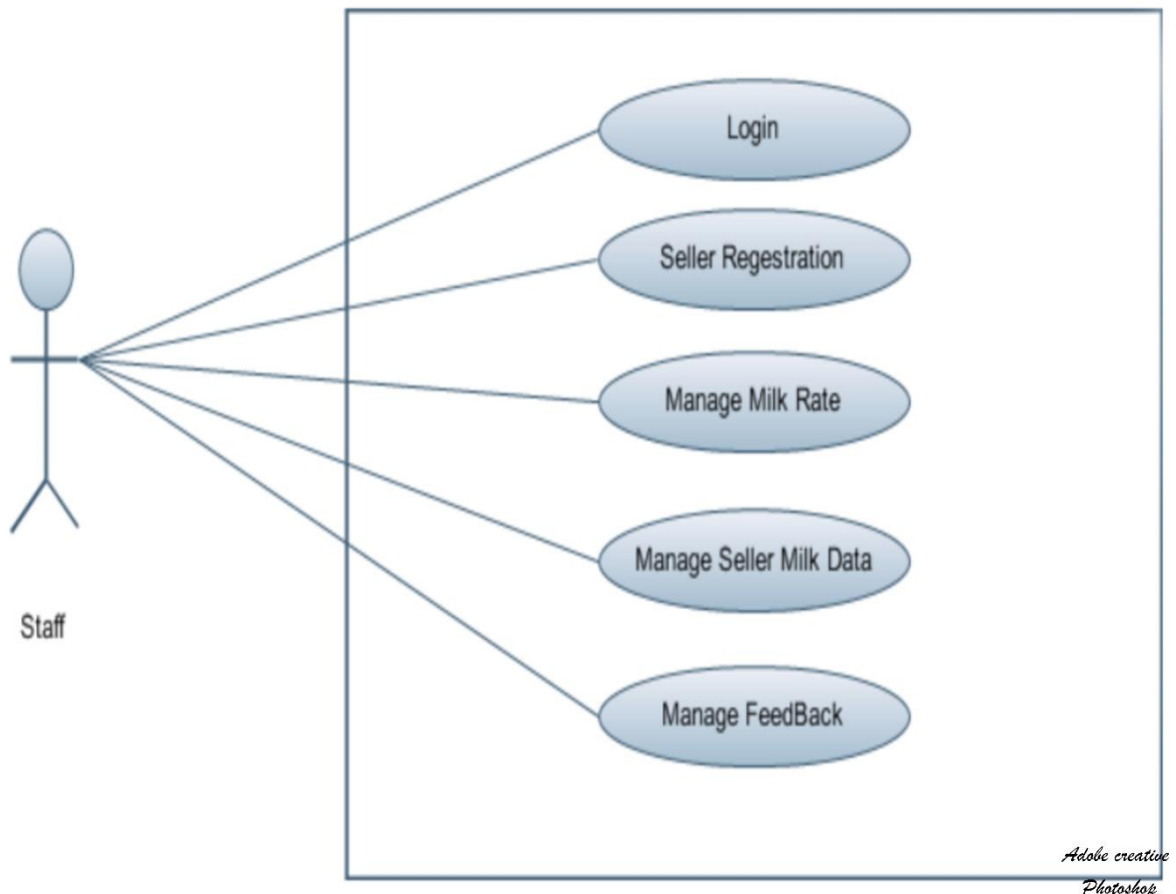
❖ USER

- 1.Registration**
- 2.Login**
- 3.View Milk Rates**
- 4.View Service Facility**
- 5.View Milk Sold**
- 6.Feedback**

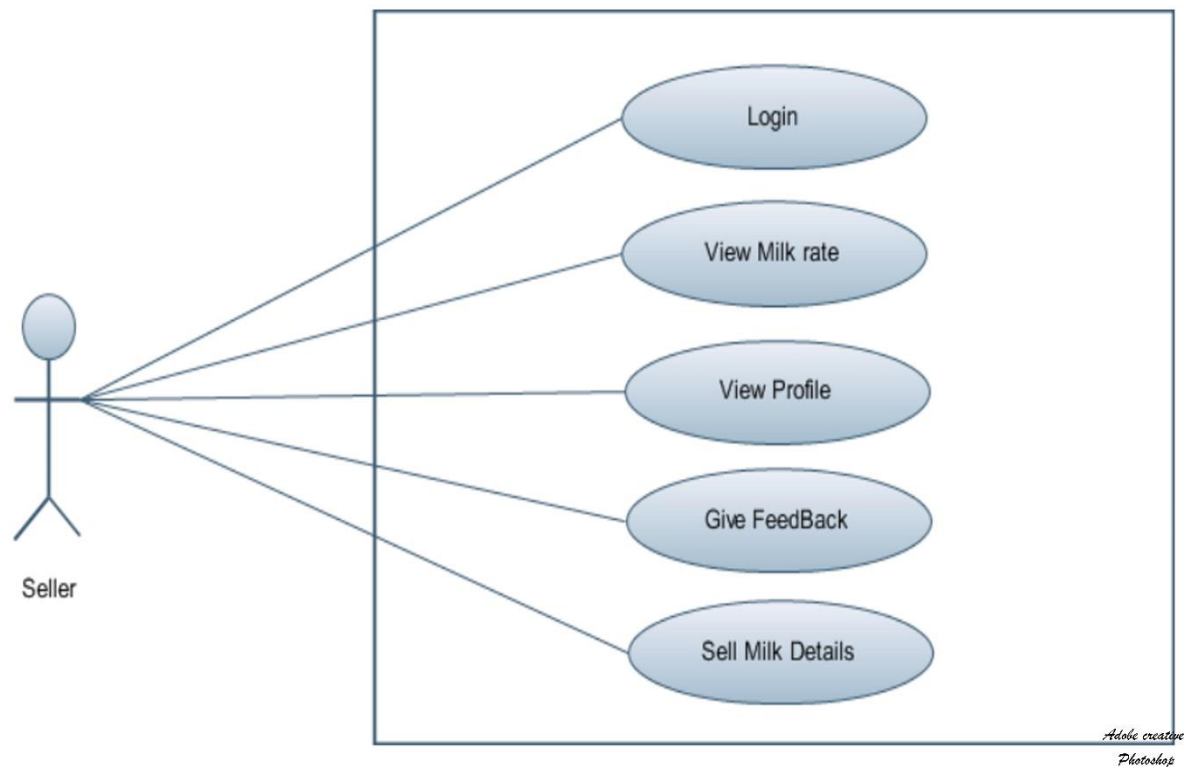
ADMIN MODULES



STAFF MODULE



USER MODULE



SYSTEM DESIGN

SYSTEM ANALYSIS AND DESIGN

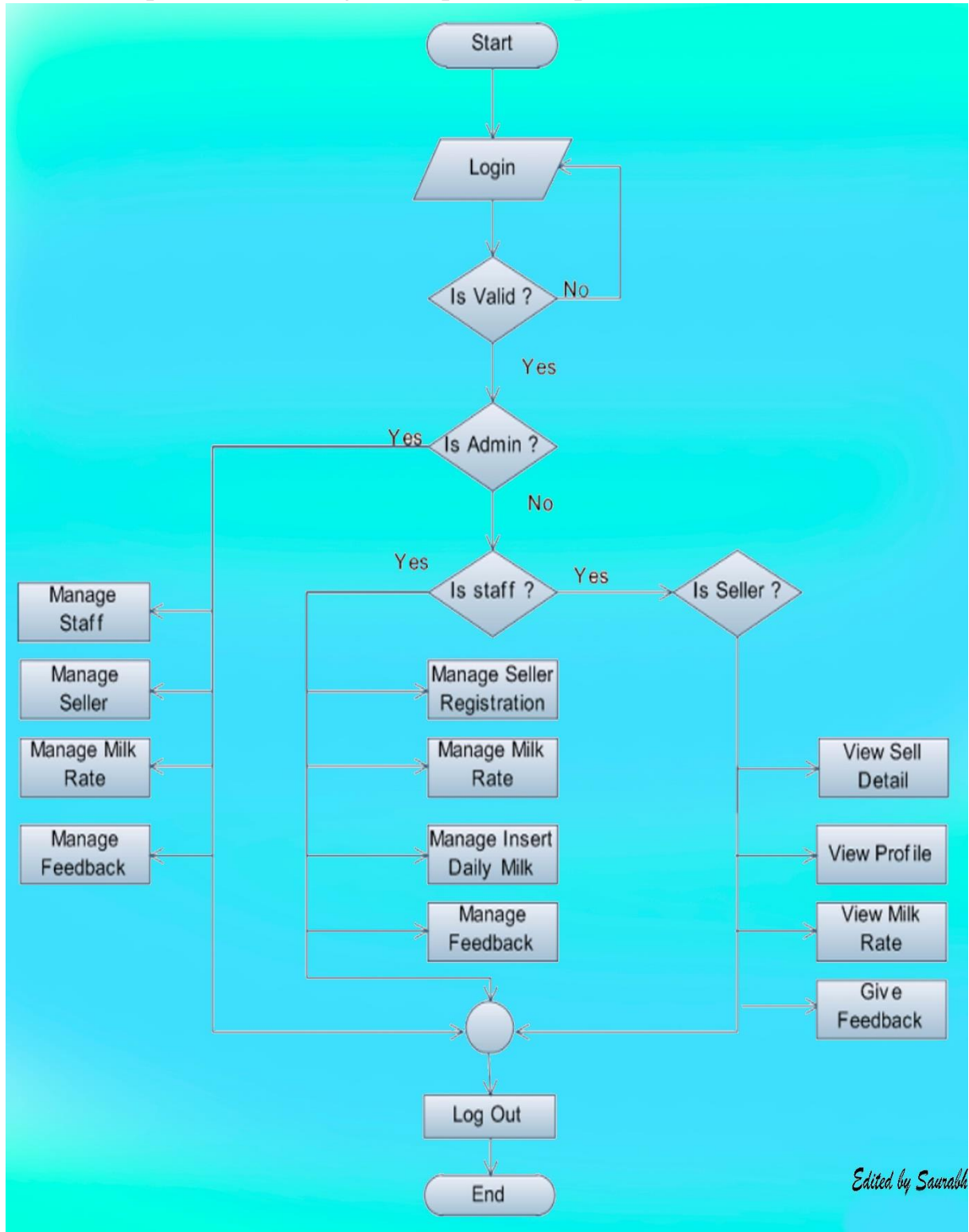
The way that is followed by carrying on with the development of application is as followed

DEFINING A PROBLEM

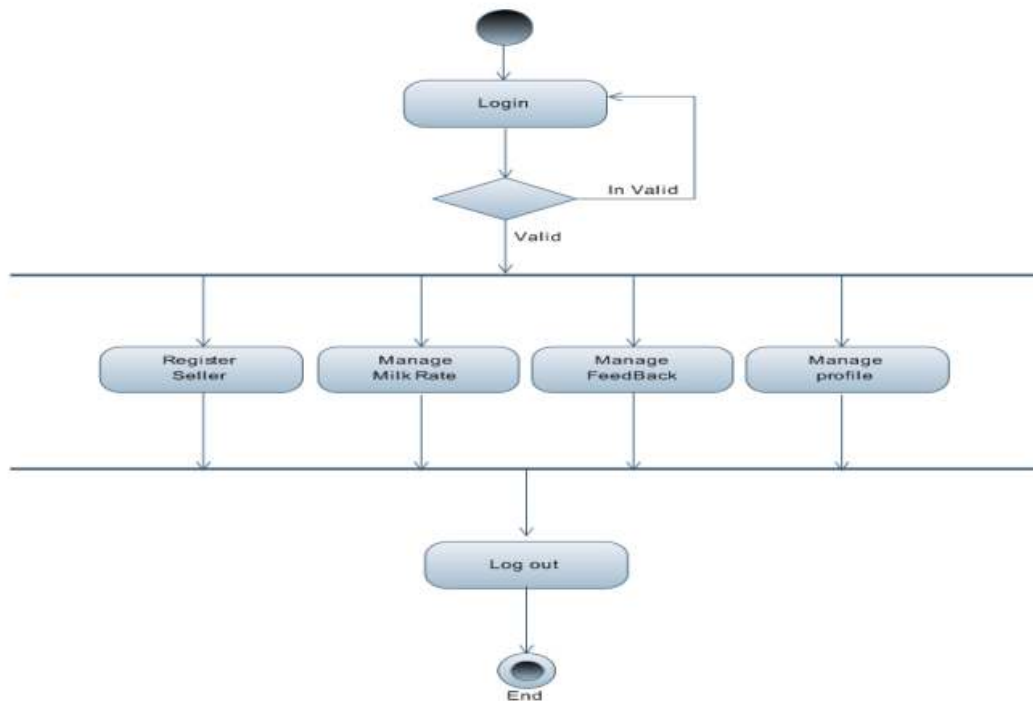
Defining a problem is one of the important activities of the project. The objective is to define precisely the business problem to be solved and there by determined the scope of the new system. The phase consists of two main tasks. The first task within this activity is to review the organization needs to originally initiated the project. The second task is to identify, at an abstract or general level, the expected capability of the new system. Thus, it helps to define the goal to be achieved and the boundaries of your system. A clear understanding of the problem will help us in building a better system and reduce the risk of project failure. It also specifies the resource that have to be made available to the project. Three important factors project goal, project budget and the resource limits are sometimes called the projects in terms of reference.

FLOW CHART

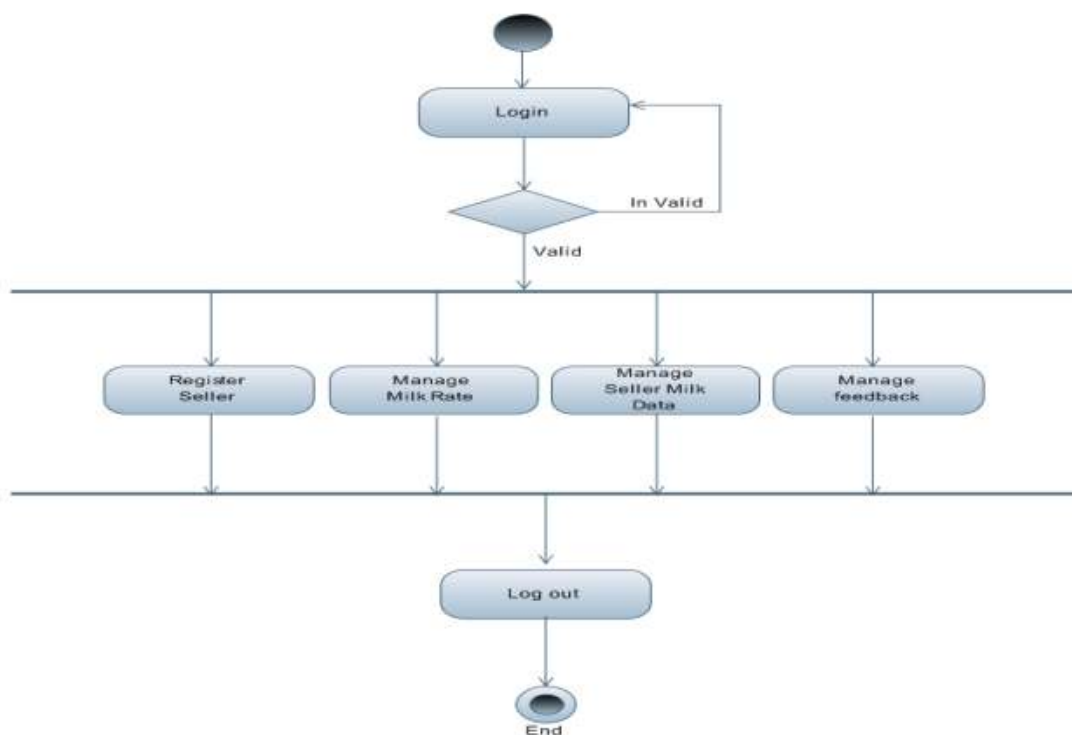
A flow chart is a diagram of the steps in a process and their sequences. Also known as a process flow diagram or process map.



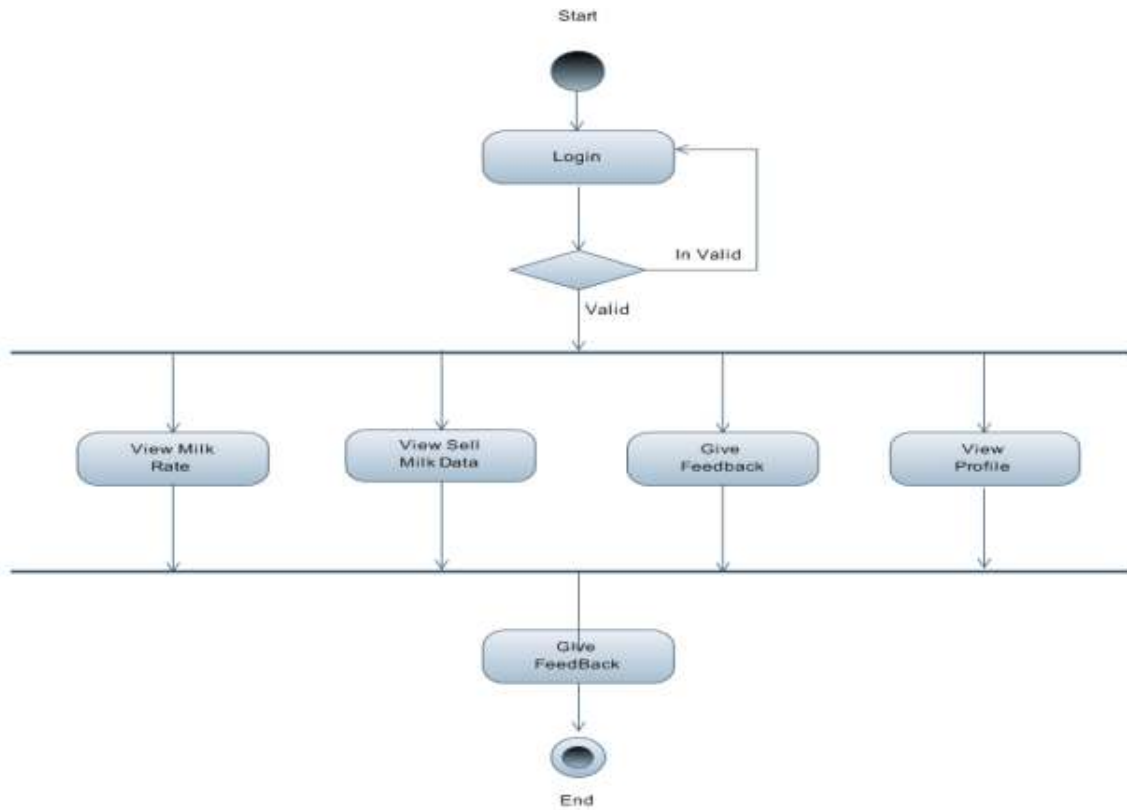
ACTIVITY DIAGRAM FOR ADMIN



ACTIVITY DIAGRAM FOR STAFF



ACTIVITY DIAGRAM FOR SELLER



CODING

```
#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<string.h>

int password();

void addrecord();

void viewrecord();

void editrecord();

void editpassword();

void deleterecord();

struct record

{

    char time[6];

    char name[30];

    char place[25];

    char amount[10];

    char note[500];

};
```

```

int main()

{

    int ch;

    printf("\n\n\t*****\n");

    printf("\t-PASSWORD PROTECTED PERSONAL DIARY-\n");

    printf("\t*****");

    while(1)

    {

        printf("\n\n\t*****MAIN MENU*****");

        printf("\n\t[1]ADD RECORD");

        printf("\n\t[2]VIEW RECORD");

        printf("\n\t[3]EDIT RECORD");

        printf("\n\t[4]DELETE RECORD");

        printf("\n\t[5]EDIT PASSWORD");

        printf("\n\t[6]EXIT");

        printf("\n\n\tENTER YOUR CHOICE:");

        scanf("%d",&ch);

        switch(ch)

        {

            case 1:

```

addrecord();

break;

case 2:

viewrecord();

break;

case 3:

editrecord();

break;

case 4:

deleterecord();

break;

case 5:

editpassword();

break;

case 6:

printf("\n\n\tTHANK YOU FOR USING THE SOFTWARE ");

getch();

exit(0);

default:

```

        printf("\n\tYOU ENTERED WRONG CHOICE..");

        printf("\n\tPRESS ANY KEY TO TRY AGAIN");

        getch();

        break;

    }

    system("cls");

}

return 0;

}

void addrecord( )

{

    system("cls");

    FILE *fp ;

    char another = 'Y' ,time[10];

    struct record e ;

    char filename[15];

    int choice;

    printf("\n\n\t*****\n");

    printf("\t**WELCOME TO THE ADD MENU**");

    printf("\n\t*****\n\n");

```

```

printf("\n\n\tENTER DATE OF YOUR RECORD:[dd-mm-yyyy]:");

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.amount);

    fflush(stdin);

    printf("\tNOTE:");

    gets(e.note);

    fwrite ( &e, sizeof ( e ), 1, fp ) ;

    printf("\n\tYOUR 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 viewrecord()

{

```

```

FILE *fpte ;

system("cls");

struct record customer ;

char time[6],choice,filename[14];

int ch;

printf("\n\n\t*****\n");

printf("\t\t* HERE IS THE VIEWING MENU *");

printf("\n\t\t*****\n\n");

choice=password();

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 ( "\n\tTHE RECORD DOES NOT EXIST...\n" ) ;

    printf("\tPRESS 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("\n\tTHE 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("\nAmount: %s",customer.amount);

    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("\nAmount: %s",customer.amount);

        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 editrecord()

{

```

```

system("cls");

FILE *fpte ;

struct record customer ;

char time[6],choice,filename[14];

int num,count=0;

printf("\n\n\t*****\n");

printf("\t**WELCOME TO THE EDITING MENU**");

printf("\n\t*****\n\n");

choice=password();

if(choice!=0)

{

    return ;

}

do

{

    printf("\n\tENTER THE DATE OF RECORD TO BE EDITED:[dd-
mm-yy]:");

    fflush(stdin);

    gets(filename);

    printf("\n\tENTER TIME:[hh:mm]:");

```

```

gets(time);

fpte = fopen ( filename, "rb+" ) ;

if ( fpte == NULL )

{

    printf( "\n\t\t\tRECORD DOES NOT EXISTS:" ) ;

    printf("\n\t\t\tPRESS 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.amount);

        printf("\nNOTE: %s",customer.note);

        printf("\n\n\t\t\tWHAT WOULD YOU LIKE TO EDIT..");

```

```
printf("\n\t1.TIME.");

printf("\n\t2.MEETING PERSON.");

printf("\n\t3.MEETING PLACE.");

printf("\n\t4.Amount.");

printf("\n\t5.NOTE.");

printf("\n\t6.WHOLE RECORD.");

printf("\n\t7.GO BACK TO MAIN MENU.");

do

{

    printf("\n\tENTER YOUR CHOICE:");

    fflush(stdin);

    scanf("%d",&num);

    fflush(stdin);

    switch(num)

    {

        case 1:

            printf("\n\tENTER THE NEW DATA:");

            printf("\n\tNEW TIME:[hh:mm]:");

            gets(customer.time);

            break;
```

case 2:

```
printf("\n\tENTER THE NEW DATA:");
```

```
printf("\n\tNEW MEETING PERSON:");
```

```
gets(customer.name);
```

```
break;
```

case 3:

```
printf("\n\tENTER THE NEW DATA:");
```

```
printf("\n\tNEW MEETING PLACE:");
```

```
gets(customer.place);
```

```
break;
```

case 4:

```
printf("\n\tENTER THE NEW DATA:");
```

```
printf("\n\tAmount:");
```

```
gets(customer.amount);
```

```
break;
```

case 5:

```
printf("\n\tENTER THE NEW DATA:");
```

```
printf("\nNOTE:");
```

```
gets(customer.note);
```

```
break;
```

case 6:

```
printf("\n\tENTER THE NEW DATA:");
```

```
printf("\n\tNEW TIME:[hh:mm]:");

gets(customer.time);

printf("\n\tNEW MEETING PERSON:");

gets(customer.name);

printf("\n\tNEW MEETING PLACE:");

gets(customer.place);

printf("\n\tAmount:");

gets(customer.amount);

printf("\n\tNOTE:");

gets(customer.note);

break;

case 7:
    printf("\n\tPRESS ANY KEY TO GO BACK...\n");

    getch();

    return ;

    break;

default:
    printf("\n\tYOU 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("\t-----\n");

    printf("\t\tTHE NEW RECORD IS:\n");

    printf("\t-----\n");

    printf("\n\t\tTIME: %s",customer.time);

    printf("\n\t\tMEETING WITH: %s",customer.name);

    printf("\n\t\tMEETING AT: %s",customer.place);

```

```

printf("\n\t\tAmount: %s",customer.amount);

printf("\n\t\tNOTE: %s",customer.note);

fclose(fpte);

printf("\n\n\tWOULD YOU LIKE TO EDIT ANOTHER
RECORD.(Y/N)");

scanf("%c",&choice);

count++;

}

else

{

printf("\n\t\tTHE RECORD DOES NOT EXIST::\n");

printf("\n\t\tWOULD YOU LIKE TO TRY AGAIN...(Y/N)");

scanf("%c",&choice);

}

}

while(choice=='Y'||choice=='y');

fclose ( fpte ) ;

if(count==1)

printf("\n\t\t%d FILE IS EDITED...\n",count);

else if(count>1)

```

```

    printf("\n\t\t%d FILES ARE EDITED..\n",count);

else

    printf("\n\tNO FILES EDITED...\n");

printf("\tPRESS ENTER TO EXIT EDITING MENU.");

getch();

}

int password()

{

    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("\n\tERROR 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\n\n\t**WARNING SYSTEM OVERRIDE**\n");

}

}

    printf("\n\n\t*YOU ENTERED WRONG PASSWORD*\n\n*YOU ARE
NOT ALLOWED TO ACCESS ANY FILE*\n\n\tPRESS ANY KEY TO
GO BACK...");

    getch();

    return 1;

}

void editpassword()

{

    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=password();

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\t**SYSTEM FLUSHED**");

        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 deleterecord( )

{

    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*\n\tDELETED
RECORDS CAN NOT BE RESTORED\t");

    printf("\n\t\t*****\n\n");

    check = password();

    if(check==1)

    {

        return ;

    }

    while ( another == 'Y' )

    {

```

```

printf("\n\nHOW WOULD YOU LIKE TO DELETE.");

printf("\n\n[1]#DELETE WHOLE RECORD\t\t");

printf("\n\n[2]#DELETE A PARTICULAR RECORD BY TIME\t");

do

{

    printf("\n\nENTER YOUR CHOICE:");

    scanf("%d",&choice);

    switch(choice)

    {

        case 1:

            printf("\n\nENTER THE DATE OF RECORD TO BE
DELETED:[dd-mm-yyyy]:");

            fflush(stdin);

            gets(filename);

            fp = fopen (filename, "wb" ) ;

            if ( fp == NULL )

            {

                printf("\n\nTHE FILE DOES NOT EXISTS");

                printf("\n\nPRESS ANY KEY TO GO BACK.");

                getch();

```

```

        return ;

    }

    fclose(fp);

    remove(filename);

    printf("\n***RECORDS DELETED SUCCESSFULLY***");

    break;

case 2:

    printf("\n\tENTER THE DATE OF RECORD:[dd-mm-yyyy]:");

    fflush(stdin);

    gets(filename);

    fp = fopen (filename, "rb" ) ;

    if ( fp == NULL )

    {

        printf("\n\tTHE FILE DOES NOT EXISTS");

        printf("\n\tPRESS ANY KEY TO GO BACK.");

        getch();

        return ;

    }

    fptr=fopen("temp","wb");

    if(fptr==NULL)

```

```

{

    printf("\n\tSYSTEM FALIURE");

    printf("\n\tPRESS 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("\n\t**DELETED 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);

}

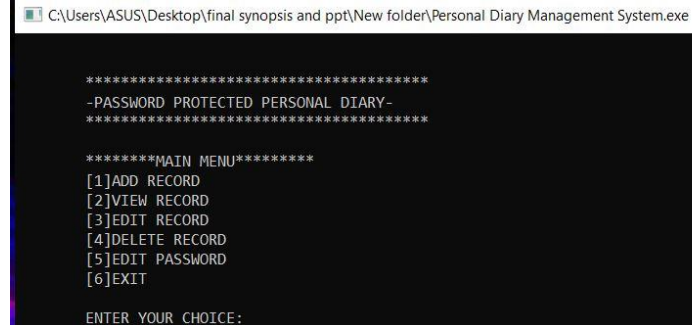
printf("\n\n\tPRESS ANY KEY TO EXIT...");

getch();

}
```

OUTPUT

1.MAIN MENU



A screenshot of a Windows application window titled "C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Personal Diary Management System.exe". The window has a black background with white text. It displays a main menu with the following options: [1]ADD RECORD, [2]VIEW RECORD, [3]EDIT RECORD, [4]DELETE RECORD, [5]EDIT PASSWORD, and [6]EXIT. The prompt "ENTER YOUR CHOICE:" is at the bottom.

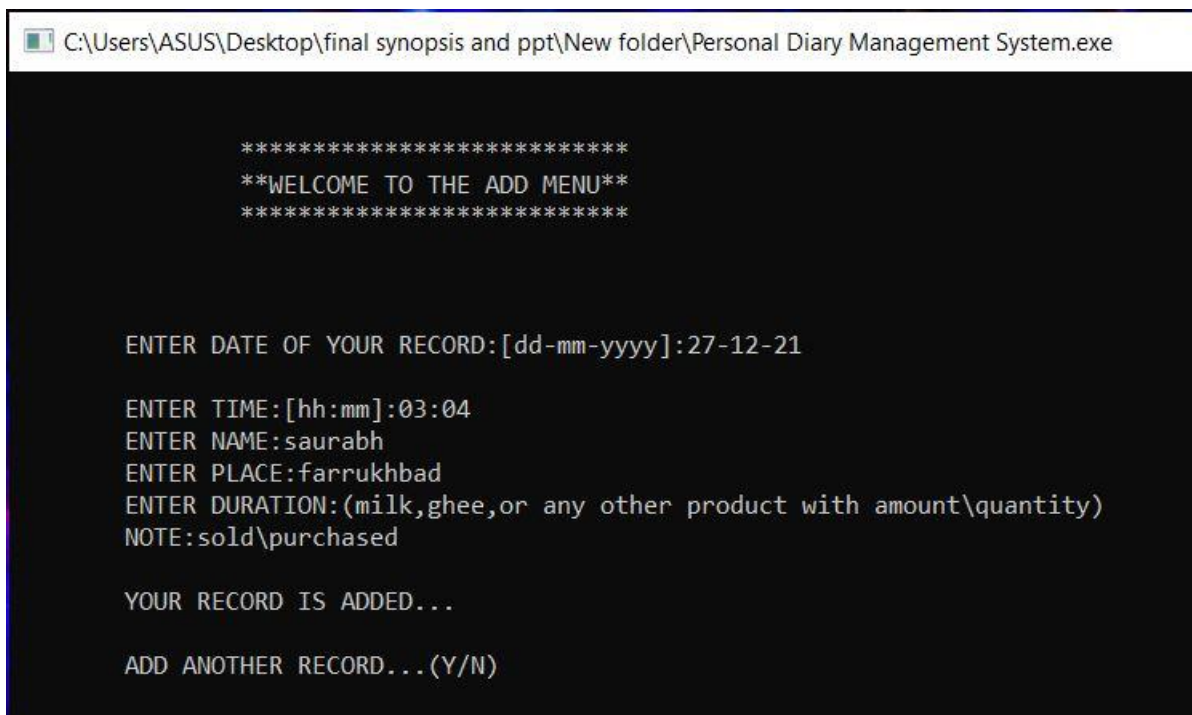
```
C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Personal Diary Management System.exe

*****
-PASSWORD PROTECTED PERSONAL DIARY-
*****

*****MAIN MENU*****
[1]ADD RECORD
[2]VIEW RECORD
[3]EDIT RECORD
[4]DELETE RECORD
[5]EDIT PASSWORD
[6]EXIT

ENTER YOUR CHOICE:
```

2.TO ADD RECORDS



A screenshot of a Windows application window titled "C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Personal Diary Management System.exe". The window has a black background with white text. It displays a welcome message and prompts for adding a record. The user has entered the following information: Date: 27-12-21, Time: 03:04, Name: saurabh, Place: farrukhbad, Duration: (milk,ghee,or any other product with amount\quantity), and Note: sold\purchased. The screen shows "YOUR RECORD IS ADDED..." and "ADD ANOTHER RECORD...(Y/N)".

```
C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Personal Diary Management System.exe

*****
**WELCOME TO THE ADD MENU**
*****

ENTER DATE OF YOUR RECORD:[dd-mm-yyyy]:27-12-21

ENTER TIME:[hh:mm]:03:04
ENTER NAME:saurabh
ENTER PLACE:farrukhbad
ENTER DURATION:(milk,ghee,or any other product with amount\quantity)
NOTE:sold\purchased

YOUR RECORD IS ADDED...

ADD ANOTHER RECORD...(Y/N)
```

3.TO VIEW RECORD

C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Personal Diary Management System.exe

```
*****  
* HERE IS THE VIEWING MENU *  
*****  
:::::FOR SECURITY PURPOSE::::::::::ONLY THREE TRIALS ARE ALLOWED:::::  
  
ENTER THE PASSWORD:  
  
ACCESS GRANTED...  
  
ENTER THE DATE OF RECORD TO BE VIEWED:[yyyy-mm-dd]:27-12-27
```

C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Per

```
HOW WOULD YOU LIKE TO VIEW:  
  
1.WHOLE RECORD OF THE DAY.  
2.RECORD OF FIX TIME.  
ENTER YOUR CHOICE:
```

FOR WHOLE DAY

```
C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Personal I

HOW WOULD YOU LIKE TO VIEW:

1.WHOLE RECORD OF THE DAY.
2.RECORD OF FIX TIME.
    ENTER YOUR CHOICE:1

THE WHOLE RECORD FOR 27-12-21 IS:

TIME: 10:21
MEETING WITH: abc
MEETING AT: abc
Amount: 12
NOTE: ff

TIME: 03:04
MEETING WITH: saurabh
MEETING AT: farrukhbad
Amount: (milk,gheesold\purchased
NOTE: sold\purchased

WOULD YOU LIKE TO CONTINUE VIEWING...(Y/N):
```

TO VIEW FOR PARTICULAT TIME

```
C:\Users\ASUS\Desktop\final synopsis and ppt\New folder

HOW WOULD YOU LIKE TO VIEW:

1.WHOLE RECORD OF THE DAY.
2.RECORD OF FIX TIME.
    ENTER YOUR CHOICE:2

ENTER TIME:[hh:mm]:03:04

YOUR RECORD IS:
TIME: 03:04
MEETING WITH: saurabh
MEETING AT: farrukhbad
Amount: (milk,gheesold\purchased
NOTE: sold\purchased

WOULD YOU LIKE TO CONTINUE VIEWING...(Y/N):
```

3.TO EDIT RECORDS (PASSWORD NEEDED)

C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Personal Diary Management System.exe

```
*****
**WELCOME TO THE EDITING MENU**
*****

:::::FOR SECURITY PURPOSE::::::::::ONLY THREE TRIALS ARE ALLOWED:::::

ENTER THE PASSWORD:

ACCESS GRANTED...

ENTER THE DATE OF RECORD TO BE EDITED:[dd-mm-yy]:27-12-21

ENTER TIME:[hh:mm]:03:04

YOUR OLD RECORD WAS AS:
TIME: 03:04
MEETING WITH: saurabh
MEETING AT: farrukhbad
DURATION: (milk,gheesold\purchased
NOTE: sold\purchased

WHAT WOULD YOU LIKE TO EDIT..
1.TIME.
2.MEETING PERSON.
3.MEETING PLACE.
4.Amount.
5.NOTE.
6.WHOLE RECORD.
7.GO BACK TO MAIN MENU.
ENTER YOUR CHOICE:
```

UNDER CONSTRUCTION

3.TO DELETE RECORDS (FOR WHOLE DAY)

C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Personal Diary Management System.exe

```
*****
* WELCOME TO DELETE MENU*
DELETED RECORDS CAN NOT BE RESTORED
*****

:::::FOR SECURITY PURPOSE::::::::::ONLY THREE TRIALS ARE ALLOWED:::::

ENTER THE PASSWORD:

ACCESS GRANTED...

HOW WOULD YOU LIKE TO DELETE.

[1]#DELETE WHOLE RECORD
[2]#DELETE A PARTICULAR RECORD BY TIME
ENTER YOU CHOICE:1

ENTER THE DATE OF RECORD TO BE DELETED:[dd-mm-yyyy]:
```

FOR PARTICULAR TIME

C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Personal Diary Management System.exe

```
*****
* WELCOME TO DELETE MENU*
DELETED RECORDS CAN NOT BE RESTORED
*****

:::::FOR SECURITY PURPOSE::::::::::ONLY THREE TRIALS ARE ALLOWED:::::

ENTER THE PASSWORD:

ACCESS GRANTED...

HOW WOULD YOU LIKE TO DELETE.

[1]#DELETE WHOLE RECORD
[2]#DELETE A PARTICULAR RECORD BY TIME
    ENTER YOUR CHOICE:2

ENTER THE DATE OF RECORD:[dd-mm-yyyy]:27-12-21

ENTER THE TIME OF RECORD TO BE DELETED:[hh:mm]:03-04

**DELETED SUCCESFULLY**
DO YOU LIKE TO DELETE ANOTHER RECORD~(Y/N):
```

4.TO EDIT\CHANGE PASSWORD

C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Personal Diary Management System.exe

```
:::::FOR SECURITY PURPOSE::::::::::ONLY THREE TRIALS ARE ALLOWED:::::

ENTER THE PASSWORD:

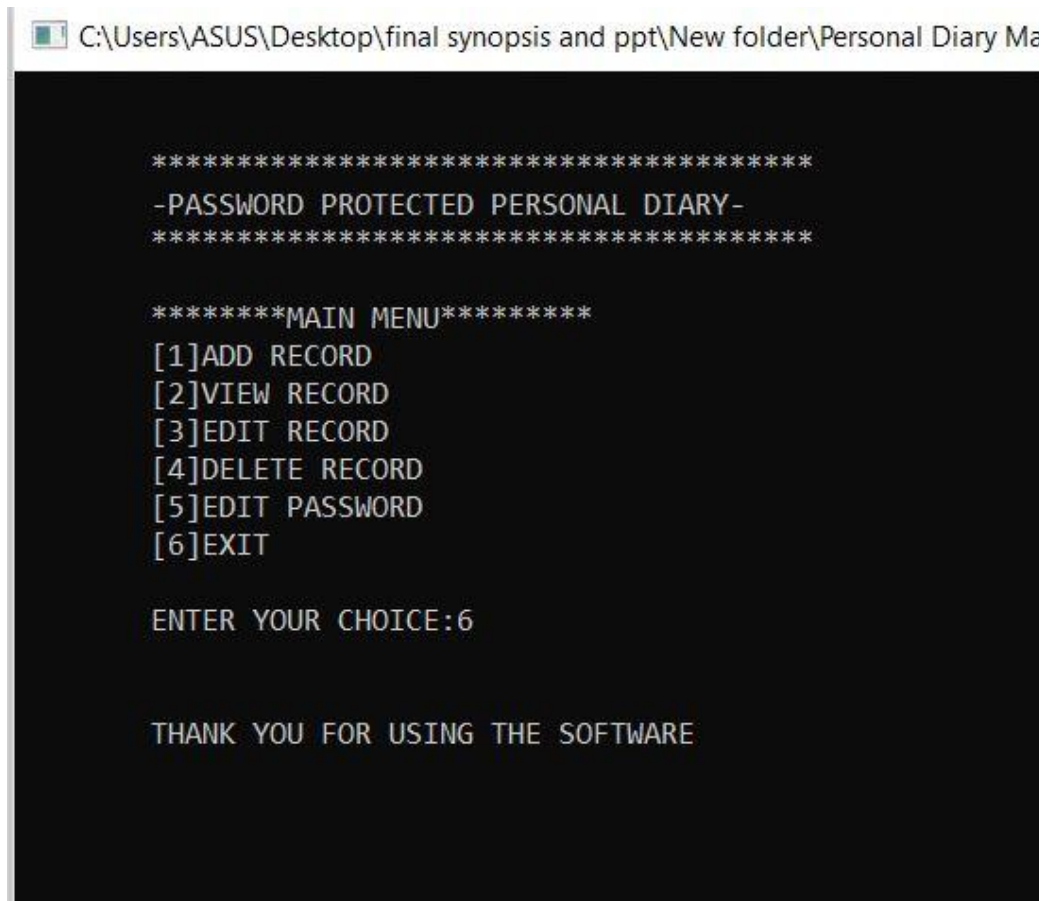
ACCESS GRANTED...

ENTER THE NEW PASSWORD:
CONFIRM PASSWORD:

PASSWORD CHANGED...

PRESS ANY KEY TO GO BACK...
```

5.EXIT



```
C:\Users\ASUS\Desktop\final synopsis and ppt\New folder\Personal Diary Ma

*****
-PASSWORD PROTECTED PERSONAL DIARY-
*****

*****MAIN MENU*****
[1]ADD RECORD
[2]VIEW RECORD
[3]EDIT RECORD
[4]DELETE RECORD
[5]EDIT PASSWORD
[6]EXIT

ENTER YOUR CHOICE:6

THANK YOU FOR USING THE SOFTWARE
```

TESTING

Testing System testing is designed to uncover the weaknesses that were not found in earlier test. In the testing phase, the program is executed with the explicit intention of finding errors. This includes forced system failures and validation of the system, as its user in the operational environment will implement it. For this purpose, test cases are developed. When a new system replaces the old one, such as in the present case, the organization can extract data from the old system to test them on the new. Such data usually exist in sufficient volume to provide sample listings and they can create a realistic environment that ensures eventual system success. Regardless of the source of test data, the programmers and analyst will eventually conduct four different types of tests.

SECURITY TESTING OF THE PROJECT

Unit testing focuses verification efforts on the smallest unit of the software design, the module. This is also known as Module Testing. The modules are tested separately. This testing is carried out during programming stage itself.

Validation Testing

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the users. After validation test has been conducted one of the two possible conditions exists

1. The function or the performance characteristics confirm to specification and are accepted.
2. A deviation from specification is uncovered and a deficiency list is created.

Output Testing

After performing the validation testing the next step is output testing of the proposed system since no system is useful if it does not produce the required output in the specific format. The outputs generated or displayed by the system under consideration are tested by asking the users about the formats required by them.

User Acceptance Testing

User acceptance of a system is a key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system users at the time development and making changes whenever required.

System verification and validation

System verification and validation is done to check the quality of the software in simulated and live environment. A number of different transactions are used to perform verification. Validation is the process of demonstrating that the implemented software does satisfy the system requirements. One aspect of software validation is to statistically analyse the program without resorting to actual execution. The system validation done in such-a-way that the system response time will not cause any hardship to the user.

White Box Testing

White box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, we can derive test cases that

- ❖ Guarantee that all independent paths within a module have been exercised at least once.
- ❖ Exercise all logical decisions on their true and false sides.
- ❖ Execute all loops at their boundaries and within their operational bounds.
- ❖ Exercise internal data structures to ensure their validity.

Black Box Testing

Black box testing methods focus on the functional requirements of the software. That is, black box testing enables us to derive sets of input conditions that will fully exercise all functional requirements of the program. Black box testing attempts to find errors in following categories:

- Incorrect or missing functions
- Interface errors
- Performance errors
- Initialization and termination error.

FUTURE SCOPE

As now the project/software will work upon command prompt in future we will make GUI (graphical User Interface) or web-based software/system so that every user or employ can access it remotely.

Seller can get his/her payment directly into account.

Customer also be a user of system.

Performance will be improved day to day users increment.

In future we can convert it in android application (APP).

BIBLIOGRAPHY

- <https://www.javatpoint.com/c-preprocessor-include>
- <https://www.javatpoint.com/printf-scanf>
- https://en.cppreference.com/w/c/language/main_function
- <https://www.javatpoint.com/data-types-in-c>
- <https://www.javatpoint.com/variables-in-c>
- <https://www.javatpoint.com/keywords-in-c>
- <https://www.javatpoint.com/c-switch>
- <https://www.javatpoint.com/c-loop>
- <https://www.javatpoint.com/c-break>
- <https://www.javatpoint.com/c-continue>
- <https://www.javatpoint.com/c-if-else>
- <https://www.javatpoint.com/c-format-specifier>
- <https://www.javatpoint.com/functions-in-c>
- <https://www.javatpoint.com/c-strings>