

**Meal Management System**

**by**

**Student Hasan Mahmud Nayem(162-35-1627)**

A project (SWE 231) submitted in fulfillment of the requirements for the degree of BSc in Software Engineering

**Department of Software Engineering**

**DAFFODIL INTERNATIONAL UNIVERSITY**

**Spring – 2017**

**DAFFODIL INTERNATIONAL UNIVERSITY**

DECLEARATION OF PROJECT

|  |  |
| --- | --- |
| Author Full Name | : Hasan Mahmud Nayem |
| Date of Birth | : 21 January, 1997 |
| Title of Project | : Schedule Management System |
| Academic Session | : 2017 |
|  |  |

|  |  |
| --- | --- |
|  | Certified by: |
| SIGNATURE | SIGNATURE OF SUPERVISOR |
|  |  |
| STUDENT ID | NAME OF SUPERVISOR |
|  |  |
| Date: | Date: |

**Acknowledgement**

First of all I would like to thanks our respected teacher Dr. Md. ASRAF ALI. He is our course teacher. He was explained all the materials I use in this project. He covered all basic and advance concept in C programming. This is a project where we are being tested about our programming knowledge. There are lot of project available but I think this project is perfect for me. Because here I can apply all I have learn during the course.

My project is based on Mess and Meal Management System. I feel the necessity of this project from my own life. I think this project will help all the students and bachelors who lived in a hostel or mess. And also in this project I can apply properly what I learned during the whole course. Like pointer, file, data structures and lot more. This project make me more confident. Now I can apply programming knowledge to solve real problems. During completing the project I face lot of problems and this problem gain more knowledge.

I have also a plan to make this apps for android platform.

Thank you.

**TABLE OF CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PROJECT DECLARATION | | | | i |
| ACKNOWLEDGEMENT | | | | ii |
| TABLE OF CONTENTS | | | | iii |
| LIST OF TABLES | | | | iv |
| LIST OF FIGURES | | | | v |
| LIST OF ABBREVIATIONS | | | | vi |
| LIST OF SYMBOLS | | | | vii |
| ABSTRACT | | | | Viii |
| CHAPTER 1 INTRODUCTION | | | | 1 |
|  | 1.1 | Background | | 2 |
|  | 1.2 | Motivation of the Project | | 3 |
|  | 1.3 | Problem Statement | | 5 |
|  | 1.4 | Project Objectives | | 6 |
|  | 1.5 | Project Organization | | 6 |
| CHAPTER 2 PROJECT METHODOLOGY | | | | 7 |
|  | 2.1 |  | | 10 |
|  | 2.2 |  | | 12 |
|  | 2.3 |  | | 15 |
| CHAPTER 3 RESULTS AND DISCUSSION | | | | 16 |
|  | 3.1 |  | | 16 |
|  | 3.2 |  | | 18 |
|  | 3.3 |  | | 18 |
|  | 3.4 |  | | 20 |
| CHAPTER 4 CONCLUSIONS AND RECOMMENDATIONS | | | | 20 |
|  | 4.1 | Findings and Contributions | | 24 |
|  | 4.2 | Recommendations for Future Works | | 25 |
| REFERENCES | | | | 27 |
| APPENDIX A | | |  | 30 |
| APPENDIX B | | |  | 31 |
|  | | |  |  |

**List of Tables:**

**List of Figures:**

**List of Abbreviation:**

**List of Symbols:**

**Abstract:**

**There** are lot of students who do not live with family during their study session. They lived in a hostel or mess. In hostel or mess there are a familiar system to control food system called meal. My project is based on this meal management system.

I have found this traditional system can take lot of time to calculate each and everything which is necessary to do daily. I want to do this system automatically. Each and every day we have to calculate total cost for food and other things. We also have track members and about their food, how times they take food. And end of the month we have to calculate total cost of the mess and what is the cost of each person. I think why do we have to calculate it manually? We can make a software and it will help me to track members and do all the calculation we need for us. That’s why I make this software.

**Chapter 1**

**INTRODUCTION**

* 1. **Background:** In bachelor life we have lot of problem. One of the most common problem is live without family and taking healthy food. Another problem is to save our money. That’s why people use a traditional system which will make sure we take healthy food and save our money. Also if we don’t take food we don’t have to pay. This is a traditional meal system. But this system kill lot of time to calculate everything and track members. That’s why I think I can make a software project which can save our time and do everything we need silently. We don’t have to pay attention how to calculate. This software will do so as we need.
  2. **Motivation of the Project:** Who is manager of a mess or hostel have to do a lot more things. One of critical part is the track all information about mess members, how much they pay, who will take food, what is the cost of food and lot more. This task are very time consuming whether we can make it automate. Last month I was a manager of our mess. And then I think I have to make a software which can reduce our work. That’s why I make this project. I think this will help lot of people.
  3. **Problem Statement:** This is meal management system. There are lot section. Like member section, update section, report section etc. Our main task is to update cost and meal every day. It will calculate and make a report for us. For doing this at first we have to add some members to the system.
  4. **Project Objectives:**
* Create Meal Table
* Login to Meal Table
* Members Section
* Add, Delete, Search, Add balance to members account
* Update daily meal and cost
* Generate and View report
* Create a report text file
  1. **Project Organization**

**Chapter 2**

**PROJECT METHODOLOGY**

**2.1 Meal Table:** Meal Table is the hurt of the project. When someone start this project this is where they will land. So this the landing of the project

2.1.1 Create Meal Table: This function will create a meal table for the user. One can add member and all information they need. All information under a meal table is separate from others. So this is very important for us if we want to manage more than one meal system at a time.

2.1.2 Login to Meal Table: If we already have a meal table we can login it with user name and password. After login we can see our main menu from where we can add members and do whatever we to perform.

2.1.3 Exit: This is where we can exit from the system.

**2.2 Linked List:** I use linked list for my project to store schedule data and operate on it. I have write some function for linked list. Like

2.2.1 CreateNode: This function will take an argument of Schedule type data and then allocate memory for it then create a node and return.

2.2.2 Add: This function also take an argument of Schedule type and add schedule to the list.

2.2.3 Delete: This function will delete the schedule from the list

2.2.4 Display: This function will display the list of the schedule.

2.2.5 Search: This function will search the list of members.

**2.3 Members:** Our all tasks based upon members. So I can say that this is main part of the project. There are several function which will create and manage members and also update their data according our needs.

2.3.1 Add Member: This function will get some necessary information from the user and create a new member. Then add this member to the list.

2.3.2 Delete Member: This function will delete the member we don’t need to track the information any more.

2.3.3 Search Member: This function will search member from the list and also display some basic information about the member if found.

2.3.4 Update Balance: This function will update the balance of a specific user.

2.3.5 Display Members: This function will display full member list.

**2.4 Update Daily Tasks:** In this section some daily task goes. We update daily cost and meal here.

2.4.1 Update Daily Cost: This function will help to update daily cost. If once updated no one can modify this data except admin. And also have to provide password.

2.4.2 Update Daily Meal: This function will help to update daily meal according to users. If once updated no one can modify this data except admin. And also have to provide password.

**2.5 Report:** This section is all about report. It can generate and display report according to our needs. Also it can make report file.

2.5.1 View Full Report: This function is responsible to calculate and display the full report of the table. All member data can also see here. It will display a formatted text to the screen.

2.5.2 Report Log: This is a log system. It will display the log of the meal.

2.5.3 Members Meal by Date: We can search member meal information by putting a specific date according our needs.

2.5.4 Generate Report File: This function will create a text file based on the member data. It is similar to full report.

**2.6 File:** In my project there are several task which I have to write an external file. I was write all members information to the file and read it while starting the project.

**2.7 Others:** There are so many other helper function exits. Which will help to design the display, or help to interact with each other.

**CHAPTER 3**

**RESULT AND DISCUSSION**

**3.1 Menus:** There are 3 menus exist on the screen when anyone launch this program. The menus are Login to Meal Table, Create a New Meal Table and Exit. I will describe all the feature.

**3.2 Login to Meal Table:** Here People can login their existing meal table with month name and password. After login there are mainly three menus, Members, Daily Task and Check Report with a Logout Menus.

**3.2.1 Members:** All Members information and operation goes here. We can perform several task like

1. Add New Member: This menu will get input some information from users and create a new member based on these data.

2. Delete Member: This function will delete member from the system which member don’t need to track.

3. Search Member: This function will search members by their name and display some basic information about them if found.

4. View All Member: This function will display all available member list to the screen with some basic information.

5. Add Balance: This function will add balances to the user account.

6. Back to Main Menu: This function is responsible to get back the main menu.

**3.2.2 Daily Task:** Our daily task function goes here.

1. Update Meal: This function will update daily meal according to member.

2. Update Cost: This function will update daily cost.

3. Back to Main Menu: This function is responsible to get back the main menu.

**3.2.3 Report:** All Report information and operation goes here. We can perform several task like

1. View Full Report: This menu will display full report to the screen.

2. Members Meal by Date: This function will search and print members meal information on a particular day.

3. Monthly Log: This function will display the monthly log.

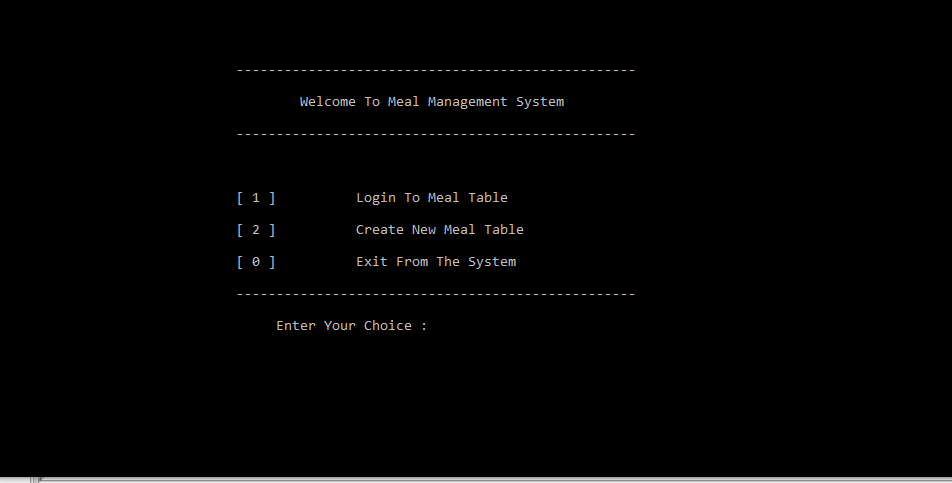
4. Make a File: This function will create file based on full report.

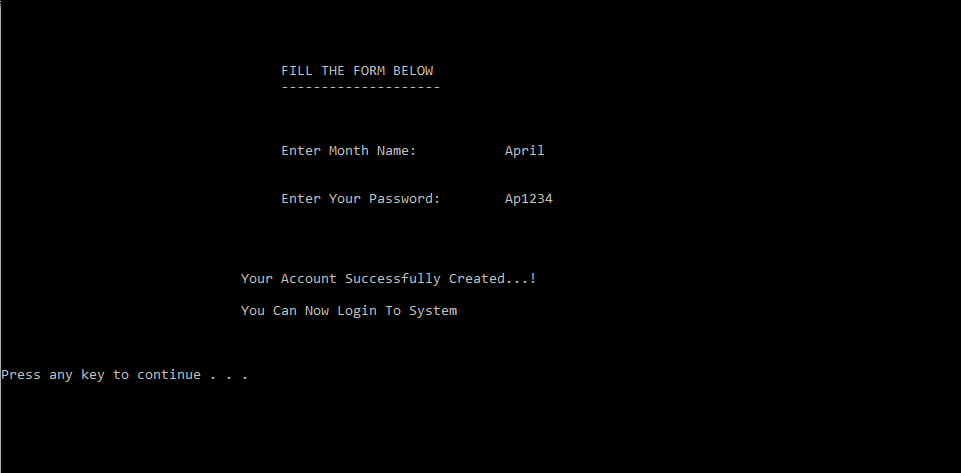
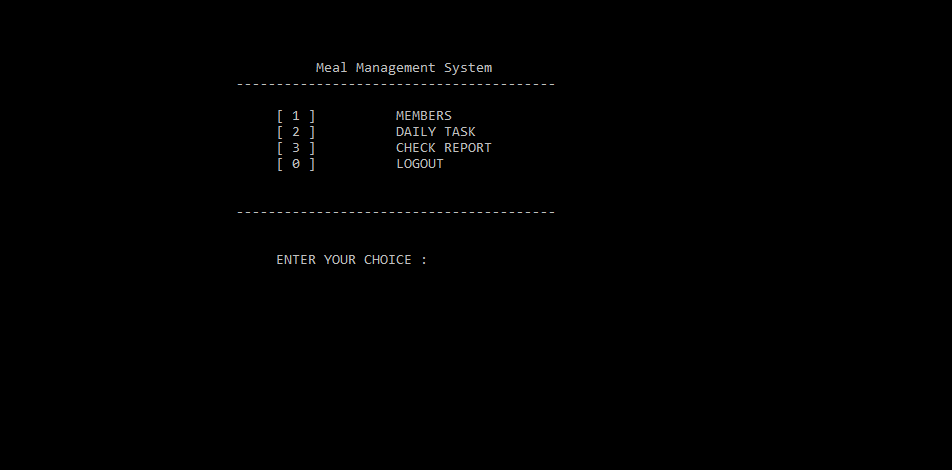
5. Back to Main Menu: This function is responsible to get back the main menu.

**3.2.4 Logout:** This function will logout a user from the system. Then we can login to different accounts too.

**3.3 Create A Meal Table:** We have to create a meal table if we don’t already create it to perform any operation. Here we just have to put a month name and a password. It will then store by the system for further use.

**3.4 Exit:** By pressing on this menu Exit user will terminate the program and will not able to do any perform before starts the program again. When user exit from the program all data are being write to the data file. So it’s important to exit properly from the system.

 Figure 1: Welcome Menu

 Figure 2: Create Meal Table****

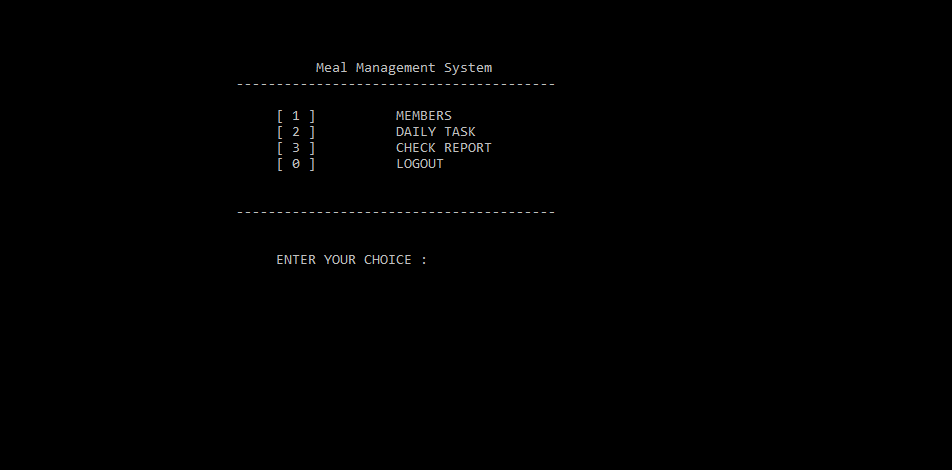
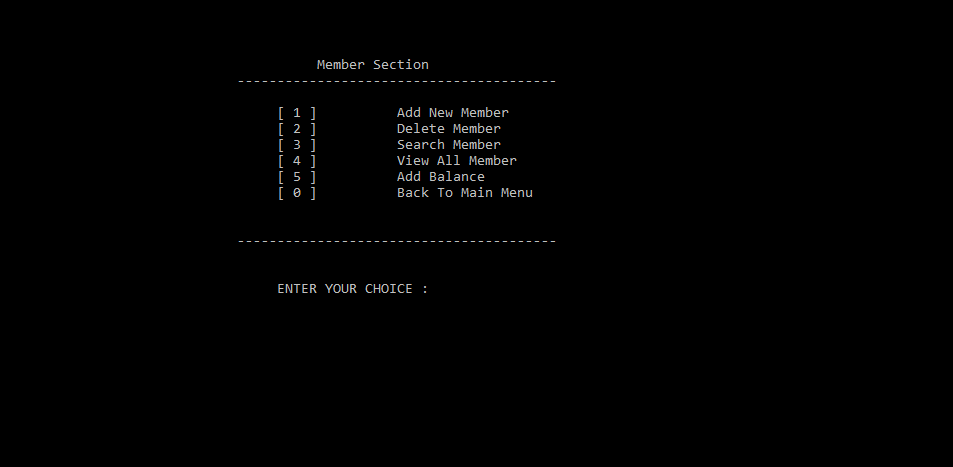
** Figure 3:** Main Menu

Figure 4: Member Section

****

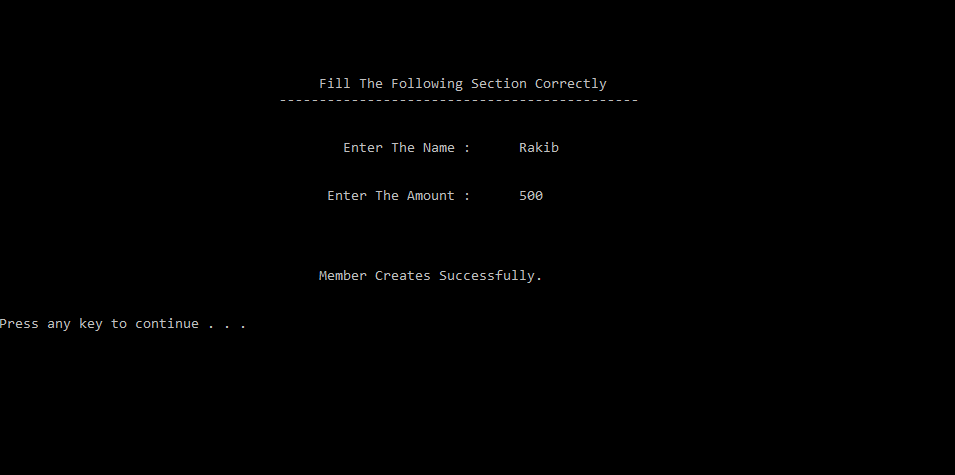
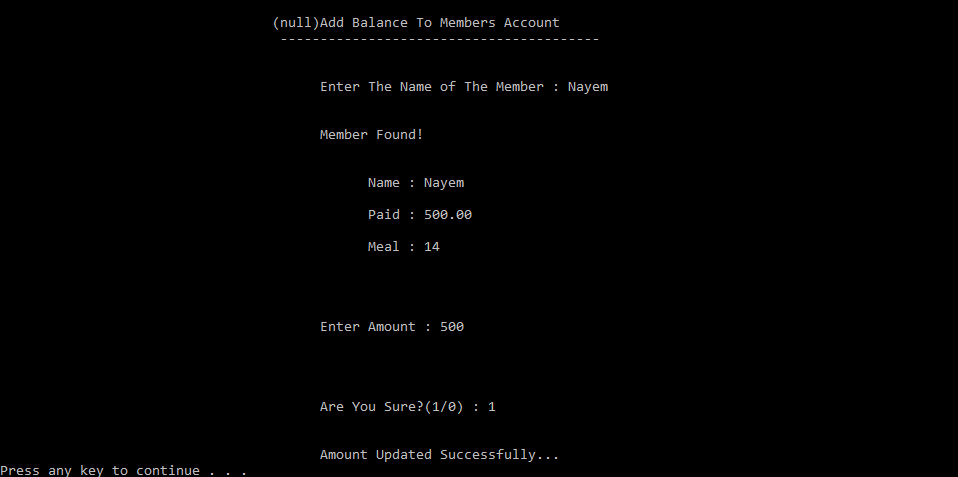
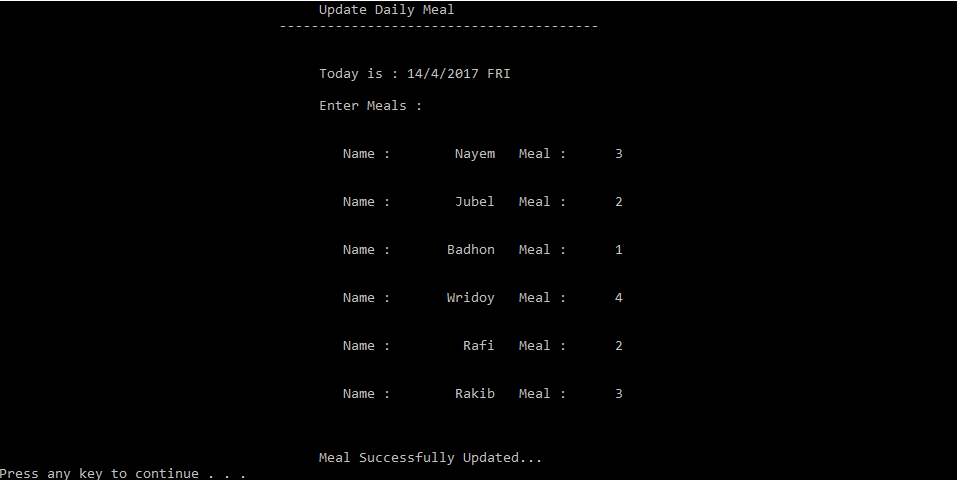
** Figure 5:** Creating Members

Figure 6: Searching Members

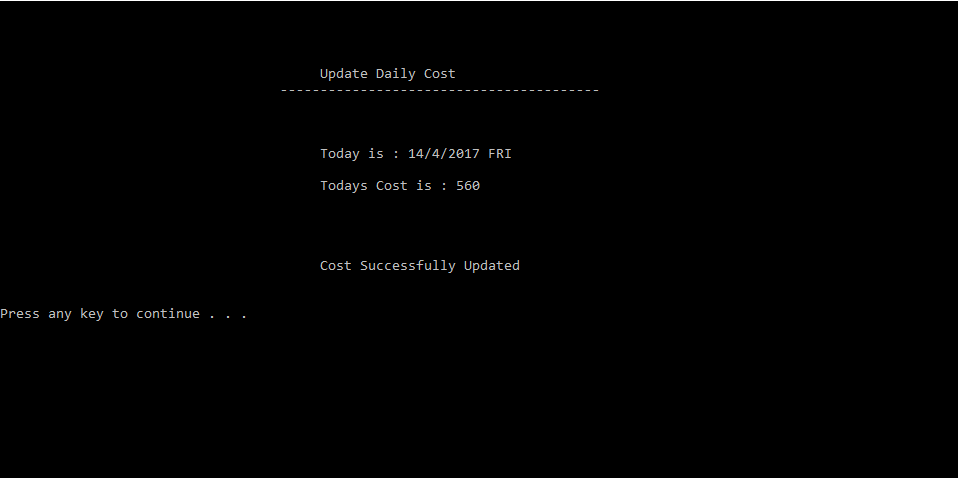
** Figure 7:** Displaying Full Member list

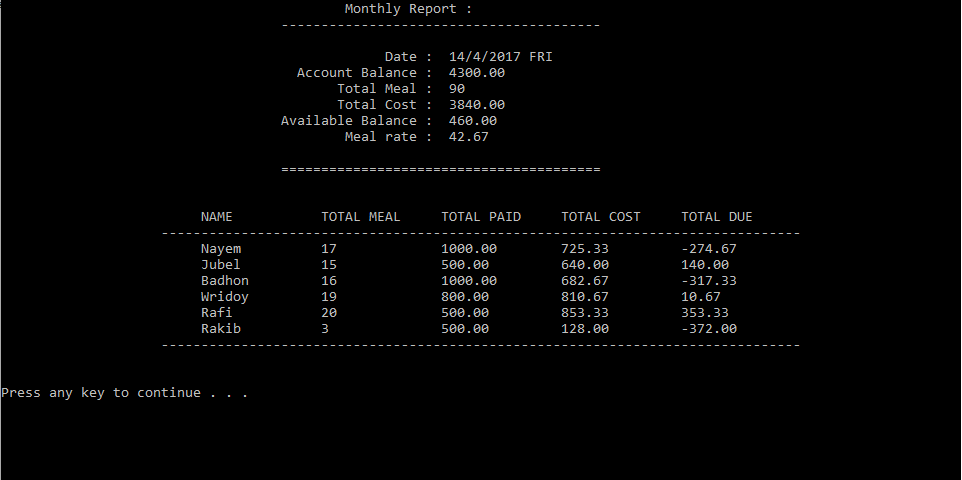
**Figure 8:** Adding Balance to User Account

****

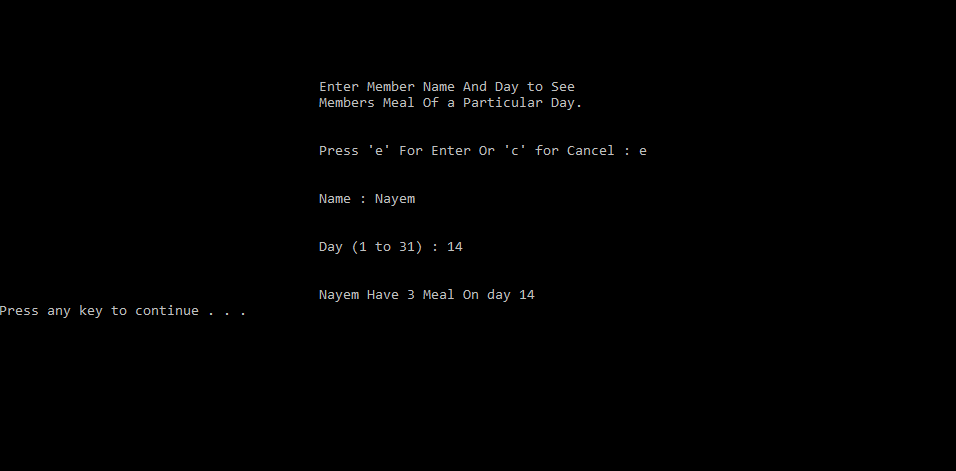
** Figure 9:** Updating Daily Meal

**Figure 10:** Update Daily Cost



 **Figure 11:** Viewing Monthly Report

**Figure 12:** Search by Name and Date



**CHAPTER 4**

**CONCLUTIONS AND RECOMMANDATIONS**

**4.1 Findings and Contributions**

**4.2 Recommendations for future work**

* I will add some more feature
* I will add Bazar Section
* I will automate this system for Bazar too
* I will Make an Android apps based on this concept
* I will add notification system to my project

**Appendix A**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <time.h>**

**#include <windows.h>**

**#define DAY 32**

**struct member**

**{**

**char name[30];**

**double pay[2];**

**int meals[DAY];**

**};**

**typedef struct member Members;**

**struct lists**

**{**

**Members info;**

**struct lists \*next;**

**};**

**struct times**

**{**

**WORD wYear;**

**WORD wMonth;**

**WORD wDayOfWeek;**

**WORD wDay;**

**WORD wHour;**

**WORD wMinute;**

**WORD wSecond;**

**WORD wMilliSecond;**

**};**

**typedef struct times Times;**

**struct dayB**

**{**

**char bazars[30][30];**

**int len;**

**};**

**typedef struct dayB Bazar;**

**void displayWelCome(void);**

**void displayMainMenu(void);**

**void displayMemberScreen(void);**

**void displayDailyTaskSection(void);**

**void displayReportSection(void);**

**void createMealTable(void);**

**int login(void);**

**int isMatchID(char name[], char pass[]);**

**MemberList\* createNode(Members data);**

**void addMember(Members data);**

**void removeMember(int index);**

**void displayMembers(void);**

**Members get(int index);**

**int search(char \*name);**

**void initalZero(Members mem);**

**Members createMember(void);**

**void deleteMembers(void);**

**void searchMember(void);**

**void viewMemberInfo(int index);**

**void addBalance(void);**

**double sumOfPayments(Members mem);**

**Times getTimes(void);**

**char\* getTimeString(Times tm);**

**char\* getWeekDay(int day);**

**int membersTotalMeal(Members mem);**

**void viewDailyReport(void);**

**int totalMealDay(int day);**

**int totalMeal(void);**

**double accountBalance(void);**

**double totalCost(void);**

**double getBalance(void);**

**double mealRate(void);**

**double costForMeal(int meal);**

**void viewDailyReport(void);**

**void generateReportFile(void);**

**void reportLog(void);**

**void membersMealByDate(void);**

**void updateDailyMeal(void);**

**void updateDailyCost(void);**

**void readLoginFile();**

**void writeLoginFile();**

**void readMembersInfo();**

**void writeMembersInfo();**

**void readCostInfo();**

**void writeCostInfo();**

**void reset();**

**double costTable[DAY] = {0};**

**char monthNames[50][15];**

**char password[50][15];**

**MemberList \*head;**

**int listSize = 0;**

**int tableSize = 0;**

**char currentPass[15], currentMonth[15];**

**int main()**

**{**

**int choice, menu, submenu, mealDays, islogin;**

**system("color 1f");**

**readLoginFile();**

**do**

**{**

**system("CLS");**

**displayWelCome();**

**scanf("%d", &choice);**

**switch(choice)**

**{**

**case 0:**

**writeLoginFile();**

**printf("Exiting...!\n");**

**break;**

**case 1:**

**islogin = login();**

**if(islogin == 1)**

**{**

**do**

**{**

**system("CLS");**

**displayMainMenu();**

**scanf("%d", &menu);**

**switch(menu)**

**{**

**case 0: printf("%\n\n\n%40sLogoutSuccessful\n\n\n");**

**writeMembersInfo();**

**writeCostInfo();**

**reset();**

**break;**

**case 1:**

**do**

**{**

**system("CLS");**

**displayMemberScreen();**

**scanf("%d",&submenu);**

**switch(submenu)**

**{**

**case 0:**

**break;**

**case 1:**

**addMemberToList();**

**break;**

**case 2:**

**deleteMembers();**

**break;**

**case 3:**

**searchMember();**

**break;**

**case 4:**

**system("CLS");**

**printf("\n\n%50s", " ");**

**printf("Displaying All Members");**

**displayMembers();**

**break;**

**case 5:**

**addBalance();**

**break;**

**default:**

**printf("Not A Valid Entry\n");**

**break;**

**}**

**system("pause");**

**}while(submenu != 0);**

**break;**

**case 2:**

**do**

**{**

**system("CLS");**

**displayDailyTaskSection();**

**scanf("%d", &submenu);**

**switch(submenu)**

**{**

**case 0:**

**break;**

**case 1:**

**updateDailyMeal();**

**break;**

**case 2:**

**updateDailyCost();**

**break;**

**default:**

**printf("Not A Valid Entry\n");**

**break;**

**}**

**system("pause");**

**}while(submenu != 0);**

**break;**

**case 3:**

**do**

**{**

**system("CLS");**

**displayReportSection();**

**scanf("%d", &submenu);**

**switch(submenu)**

**{**

**case 0:**

**break;**

**case 1:**

**viewDailyReport();**

**break;**

**case 2:**

**membersMealByDate();**

**break;**

**case 3:**

**reportLog();**

**break;**

**case 4:**

**generateReportFile();**

**printf("\n\n\n%40sSuccessfully Generated\n\n\n", " ");**

**break;**

**default:**

**printf("Not A Valid Entry\n");**

**break;**

**}**

**system("pause");**

**}while(submenu != 0);**

**break;**

**default:**

**printf("Not A Valid Entry\n");**

**break;**

**}**

**system("pause");**

**}while(menu != 0);**

**}**

**else**

**{**

**printf("\n%40sMonth And Password Not Match\n", " ");**

**printf("%40sPlease Try Again With Proper Password\n", " ");**

**printf("\n\n%40sIf You Don't have Registered Yet, Please Create An Table First\n\n", " ");**

**}**

**break;**

**case 2:**

**createMealTable();**

**break;**

**}**

**system("pause");**

**}**

**while(choice != 0);**

**}**

**MemberList\* createNode(Members data)**

**{**

**MemberList\* temp;**

**temp = (MemberList \*)malloc(sizeof(MemberList));**

**if(temp == NULL)**

**{**

**printf("Memory Not Allocated\n");**

**return 0;**

**}**

**temp->info = data;**

**temp->next = NULL;**

**listSize++;**

**return temp;**

**}**

**createNode function**

**void addMember(Members data)**

**{**

**MemberList \*temp = createNode(data);**

**if(head == NULL)**

**{**

**head = temp;**

**head->next = NULL;**

**}**

**else**

**{**

**MemberList \*current = head;**

**while(current->next != NULL)**

**{**

**current = current->next;**

**}**

**current->next = temp;**

**}**

**}**

**void removeMember(int index)**

**{**

**if(index < 0 || index > listSize-1)**

**{**

**printf("Not A Valid Index\n");**

**return;**

**}**

**if(index == 0)**

**{**

**head = head->next;**

**listSize--;**

**return;**

**}**

**else**

**{**

**MemberList \*current = head;**

**while(index-1>0)**

**{**

**current = current->next;**

**index--;**

**}**

**current->next = current->next->next;**

**listSize--;**

**}**

**}**

**void displayMembers()**

**{**

**int i;**

**printf("\n\n\n\n%50s", " ");**

**printf("Member List\n");**

**printf("%38s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

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

**if(head == NULL)**

**{**

**printf("\n%40s", " ");**

**printf("There Is No Member\n\n\n");**

**return;**

**}**

**MemberList \*current = head;**

**int index = 0;**

**printf("\n%40s", " ");**

**printf("%10s%15s%10s\n", "Index", "Name", "Amount");**

**while(current != NULL)**

**{**

**printf("\n%40s", " ");**

**printf("%10d%15s%10.2lf\n", index++, current->info.name, sumOfPayments(current->info));**

**current = current->next;**

**}**

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

**}**

**Members get(int index)**

**{**

**Members m;**

**if(listSize == 0)**

**{**

**printf("There Are No Member Available\n");**

**return m;**

**}**

**if(index <0 && index > listSize)**

**{**

**printf("Invalid Index\n");**

**return m;**

**}**

**MemberList \*current = head;**

**while(current != NULL && index > 0)**

**{**

**current = current->next;**

**index--;**

**}**

**m = current->info;**

**return m;**

**}**

**int search(char \*name)**

**{**

**int ind = 0;**

**MemberList \*current = head;**

**while(current != NULL)**

**{**

**if(strcmp(current->info.name, name) == 0)**

**{**

**return ind;**

**}**

**else**

**{**

**current = current->next;**

**ind++;**

**}**

**}**

**return -1;**

**}**

**Members createMember(void)**

**{**

**char name[30];**

**double pay;**

**Members mem;**

**int i;**

**system("CLS");**

**printf("\n\n\n\n\n%40s", " ");**

**printf("Fill The Following Section Correctly\n");**

**printf("%35s", " ");**

**for(i=0; i<45; i++)**

**{**

**printf("%c", '-');**

**}**

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

**printf("%40s%20s%5s", " ", "Enter The Name : ", " ");**

**scanf("%s", &name);**

**printf("\n\n%40s%20s%5s", " ", "Enter The Amount : ", " ");**

**scanf("%lf", &pay);**

**strcpy(mem.name, name);**

**for(i=0; i<DAY; i++)**

**{**

**mem.meals[i] = -1;**

**}**

**mem.pay[0] = pay;**

**mem.pay[1] = 0;**

**printf("\n\n\n\n%40s", " ");**

**printf("Member Creates Successfully.\n\n\n");**

**return mem;**

**}**

**void addMemberToList(void)**

**{**

**Members m = createMember();**

**addMember(m);**

**}**

**void deleteMembers()**

**{**

**system("CLS");**

**int ind, conf, i;**

**printf("\n\n\n%40s", " ");**

**printf("Select Index From Below To Delete (-1 for cancel)\n");**

**displayMembers();**

**printf("\n\n%40s", " ");**

**printf("Enter Index : ");**

**scanf("%d", &ind);**

**if(ind == -1)**

**{**

**printf("\n\n\n%40sCancel Deleting\n\n\n", " ");**

**return;**

**}**

**printf("\n\n\n%40s", " ");**

**printf("Are Your Sure ? (1/0) : ");**

**scanf("%d", &conf);**

**if(conf == 1)**

**{**

**removeMember(ind);**

**printf("\n\n%40s", " ");**

**printf("Member Deleted Successfully\n");**

**}**

**else**

**{**

**printf("\n\n%40s", " ");**

**printf("Cancel Deletion\n");**

**}**

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

**}**

**void searchMember()**

**{**

**system("CLS");**

**char name[30];**

**int i;**

**printf("\n\n\n\n%40s", " ");**

**printf("Search Member Info\n");**

**printf("%35s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

**printf("\n\n\n%40s", " ");**

**printf("Enter Name To Search : ");**

**scanf("%s", &name);**

**int sear = search(name);**

**if(sear != -1)**

**{**

**printf("\n\n%45s", " ");**

**printf("Members Found !\n\n\n");**

**viewMemberInfo(sear);**

**}**

**else**

**{**

**printf("\n\n%40s", " ");**

**printf("Members Not Found!\n");**

**}**

**}**

**void viewMemberInfo(int index)**

**{**

**Members m = get(index);**

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

**printf("%40s%10s : %-20s\n\n", " ", "Name", m.name);**

**printf("%40s%10s : %-20.2lf\n\n", " ", "Paid", sumOfPayments(m));**

**printf("%40s%10s : %-20d\n\n", " ", "Meal", membersTotalMeal(m));**

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

**}**

**void addBalance(void)**

**{**

**system("CLS");**

**char name[30];**

**int ind = 0, conf, ch=1, i;**

**double amt, previous;**

**printf("\n\n\n\n%40s");**

**printf("Add Balance To Members Account\n");**

**printf("%35s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

**printf("\n\n\n%40s", " ");**

**printf("Enter The Name of The Member : ");**

**scanf("%s", &name);**

**MemberList \*current = head;**

**while(current != NULL)**

**{**

**if(strcmp(current->info.name, name) == 0)**

**{**

**printf("\n\n%40s", " ");**

**printf("Member Found!\n");**

**viewMemberInfo(ind);**

**printf("\n\n%40s", " ");**

**printf("Enter Amount : ");**

**scanf("%lf", &amt);**

**printf("\n\n\n\n%40s", " ");**

**printf("Are You Sure?(1/0) : ");**

**scanf("%d", &conf);**

**if(conf == 1)**

**{**

**current->info.pay[0] = amt+current->info.pay[0];**

**printf("\n\n%40s", " ");**

**printf("Amount Updated Successfully...\n");**

**ch = 0;**

**return;**

**}**

**}**

**else**

**{**

**current = current->next;**

**ind++;**

**}**

**}**

**if(ch)**

**{**

**printf("\n\n\n%40s", " ");**

**printf("Members Not Found!\n");**

**}**

**}**

**double sumOfPayments(Members m)**

**{**

**return m.pay[0];**

**}**

**Times getTimes()**

**{**

**Times tm;**

**GetSystemTime(&tm);**

**return tm;**

**}**

**char\* getTimeString(Times tm)**

**{**

**char timeString[12];**

**sprintf(timeString, "%d/%d/%d %s", tm.wDay, tm.wMonth, tm.wYear, getWeekDay(tm.wDayOfWeek));**

**return timeString;**

**}**

**char\* getWeekDay(int day)**

**{**

**char weekDay[5];**

**switch(day)**

**{**

**case 0:**

**strcpy(weekDay, "SUN");**

**break;**

**case 1:**

**strcpy(weekDay, "MON");**

**break;**

**case 2:**

**strcpy(weekDay, "TUE");**

**break;**

**case 3:**

**strcpy(weekDay, "WED");**

**break;**

**case 4:**

**strcpy(weekDay, "THU");**

**break;**

**case 5:**

**strcpy(weekDay, "FRI");**

**break;**

**case 6:**

**strcpy(weekDay, "SAT");**

**}**

**return weekDay;**

**}**

**void updateDailyCost(void)**

**{**

**Times tm = getTimes();**

**char dat[30], passw[15], choice;**

**double cost;**

**int i, editable = 1;**

**system("CLS");**

**printf("\n\n\n\n%40s", " ");**

**printf("Update Daily Cost");**

**printf("\n%35s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

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

**if(head == NULL)**

**{**

**printf("\n%40s", " ");**

**printf("There is No Member Available. So You Can't Update Cost\n\n");**

**return;**

**}**

**if(costTable[tm.wDay] != 0)**

**{**

**printf("\n%40s", " ");**

**printf("Already Updated\n%40sOnly Admin Can Change Data After Edited Once\n\n", " ");**

**printf("%40sPress (e) to Edit Or (c) to Cancel : ", " ");**

**scanf(" %c", &choice);**

**if(choice == 'e')**

**{**

**printf("\n%40sEnter Password : ", " ");**

**scanf("%s", &passw);**

**if(strcmp(currentPass, passw) == 0)**

**{**

**editable = 1;**

**}**

**else editable = 0;**

**}**

**else**

**{**

**printf("\n%40sUpdate Canceled\n", " ");**

**return;**

**}**

**}**

**if(editable == 1)**

**{**

**strcpy(dat, getTimeString(tm));**

**printf("\n\n%40s", " ");**

**printf("Today is : %s\n\n", dat);**

**printf("%40s", " ");**

**printf("Todays Cost is : ");**

**scanf("%lf", &cost);**

**costTable[tm.wDay] = cost;**

**printf("\n\n\n\n%40sCost Successfully Updated\n\n\n", " ");**

**}**

**}**

**void updateDailyMeal(void)**

**{**

**int m, i, editable = 1;**

**MemberList \*current = head;**

**Times tm = getTimes();**

**char ch[30], passw[15], choice;**

**strcpy(ch, getTimeString(tm));**

**system("CLS");**

**printf("\n\n\n\n%40s", " ");**

**printf("Update Daily Meal");**

**printf("\n%35s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

**printf("\n\n\n%40s", " ");**

**printf("Today is : %s\n", ch);**

**if(head == NULL)**

**{**

**printf("\n\n%40s", " ");**

**printf("There Are No Members Available\n");**

**return;**

**}**

**if(current->info.meals[tm.wDay] != -1)**

**{**

**printf("\n%40s", " ");**

**printf("Already Updated\n%40sOnly Admin Can Change Data After Edited Once\n\n", " ");**

**printf("%40sPress (e) to Edit Or (c) to Cancel : ", " ");**

**scanf(" %c", &choice);**

**if(choice == 'e')**

**{**

**printf("\n%40sEnter Password : ", " ");**

**scanf("%s", &passw);**

**if(strcmp(currentPass, passw) == 0)**

**{**

**editable = 1;**

**}**

**else editable = 0;**

**}**

**else**

**{**

**printf("\n%40sUpdate Canceled\n", " ");**

**return;**

**}**

**}**

**if(editable)**

**{**

**printf("\n%40s", " ");**

**printf("Enter Meals : \n");**

**while(current != NULL)**

**{**

**printf("\n\n%40s", " ");**

**printf("%10s%12s%10s%5s", "Name : ", current->info.name, "Meal : ", " ");**

**scanf("%d", &m);**

**current->info.meals[tm.wDay] =m;**

**current = current->next;**

**}**

**printf("\n\n\n%40s", " ");**

**printf("Meal Successfully Updated...\n");**

**}**

**}**

**int totalMealDay(int day)**

**{**

**MemberList \*current = head;**

**int total = 0;**

**while(current != NULL)**

**{**

**if(current->info.meals[day] != -1)**

**{**

**total = total + current->info.meals[day];**

**current = current->next;**

**}**

**}**

**return total;**

**}**

**int membersTotalMeal(Members mem)**

**{**

**int i, total = 0;**

**for(i=0; i<DAY; i++)**

**{**

**if(mem.meals[i] != -1)**

**{**

**total += mem.meals[i];**

**}**

**}**

**return total;**

**}**

**int totalMeal(void)**

**{**

**MemberList \*current = head;**

**int total = 0;**

**while(current != NULL)**

**{**

**total += membersTotalMeal(current->info);**

**current = current->next;**

**}**

**return total;**

**}**

**double accountBalance(void)**

**{**

**MemberList \*current = head;**

**double balance = 0;**

**while(current != NULL)**

**{**

**balance += sumOfPayments(current->info);**

**current = current->next;**

**}**

**return balance;**

**}**

**double totalCost(void)**

**{**

**int i;**

**double cost = 0;**

**for(i=1; i<DAY; i++)**

**{**

**cost += costTable[i];**

**}**

**return cost;**

**}**

**double getBalance(void)**

**{**

**return accountBalance() - totalCost();**

**}**

**double mealRate(void)**

**{**

**return (totalCost()/totalMeal());**

**}**

**double costForMeal(int meal)**

**{**

**return mealRate()\*meal;**

**}**

**void viewDailyReport(void)**

**{**

**int i, d1;**

**double d2, d3, d4;**

**Times tm = getTimes();**

**char dt[15];**

**strcpy(dt, getTimeString(tm));**

**MemberList \*current = head;**

**system("CLS");**

**printf("%35s%25s\n", " ", "Monthly Report : ");**

**printf("%35s" , " ");**

**for(i=0; i<40; i++)**

**{**

**printf("-");**

**}**

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

**printf("%27s%25s : %-15s\n", " ", "Date", dt);**

**printf("%27s%25s : %-10.2lf\n", " ", "Account Balance", accountBalance());**

**printf("%27s%25s : %-10d\n", " ", "Total Meal", totalMeal());**

**printf("%27s%25s : %-10.2lf\n", " ", "Total Cost", totalCost());**

**printf("%27s%25s : %-10.2lf\n", " ", "Available Balance", getBalance());**

**printf("%27s%25s : %-10.2lf\n", " ", "Meal rate", mealRate());**

**printf("\n%35s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("=");**

**}**

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

**printf("%-25s%-15s%-15s%-15s%-15s%-15s\n", " ", "NAME", "TOTAL MEAL", "TOTAL PAID", "TOTAL COST", "TOTAL DUE");**

**printf("%20s", " ");**

**for(i=0; i<80; i++)**

**{**

**printf("-");**

**}**

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

**while(current != NULL)**

**{**

**d1 = membersTotalMeal(current->info);**

**d2 = sumOfPayments(current->info);**

**d3 = costForMeal(d1);**

**d4 = d3-d2;**

**printf("%-25s%-15s%-15d%-15.2lf%-15.2lf%-15.2lf\n", " " , current->info.name, d1, d2, d3, d4);**

**current = current->next;**

**}**

**printf("%20s", " ");**

**for(i=0; i<80; i++)**

**{**

**printf("-");**

**}**

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

**}**

**void generateReportFile(void)**

**{**

**FILE \*files;**

**int i, d1;**

**double d2, d3, d4;**

**Times tm = getTimes();**

**char dt[15];**

**strcpy(dt, getTimeString(tm));**

**MemberList \*current = head;**

**files = fopen("MonthlyReport.txt", "w");**

**if(files == NULL)**

**{**

**exit(0);**

**}**

**fprintf(files, "%35s%25s\n", " ", "Monthly Report : ");**

**fprintf(files, "%35s" , " ");**

**for(i=0; i<40; i++)**

**{**

**fprintf(files, "%s", "-");**

**}**

**fprintf(files, "%s", "\n\n");**

**fprintf(files, "%27s%25s : %-15s\n", " ", "Date", dt);**

**fprintf(files, "%27s%25s : %-10.2lf\n", " ", "Account Balance", accountBalance());**

**fprintf(files, "%27s%25s : %-10d\n", " ", "Total Meal", totalMeal());**

**fprintf(files, "%27s%25s : %-10.2lf\n", " ", "Total Cost", totalCost());**

**fprintf(files, "%27s%25s : %-10.2lf\n", " ", "Available Balance", getBalance());**

**fprintf(files, "%27s%25s : %-10.2lf\n", " ", "Meal rate", mealRate());**

**fprintf(files, "\n%35s", " ");**

**for(i=0; i<40; i++)**

**{**

**fprintf(files, "%s", "=");**

**}**

**fprintf(files, "%s", "\n\n\n");**

**fprintf(files, "%-25s%-15s%-15s%-15s%-15s%-15s\n", " ", "NAME", "TOTAL MEAL", "TOTAL PAID", "TOTAL COST", "TOTAL DUE");**

**fprintf(files, "%20s", " ");**

**for(i=0; i<80; i++)**

**{**

**fprintf(files, "%s", "-");**

**}**

**fprintf(files, "%s", "\n");**

**while(current != NULL)**

**{**

**d1 = membersTotalMeal(current->info);**

**d2 = sumOfPayments(current->info);**

**d3 = costForMeal(d1);**

**d4 = d3-d2;**

**fprintf(files, "%-25s%-15s%-15d%-15.2lf%-15.2lf%-15.2lf\n", " " , current->info.name, d1, d2, d3, d4);**

**current = current->next;**

**}**

**fprintf(files, "%20s", " ");**

**for(i=0; i<80; i++)**

**{**

**fprintf(files, "%s", "-");**

**}**

**fclose(files);**

**}**

**void reportLog(void)**

**{**

**int i;**

**Times tm = getTimes();**

**MemberList \*temp = head;**

**system("CLS");**

**printf("\n\n\n%40sMonthly Meal Log\n\n", " ");**

**printf("%30s%10s", " ", "DAY");**

**while(temp != NULL)**

**{**

**printf("%10s", temp->info.name);**

**temp = temp->next;**

**}**

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

**for(i=1; i<=tm.wDay; i++)**

**{**

**printf("\n%30s%10d", " ", i);**

**temp = head;**

**while(temp != NULL)**

**{**

**printf("%10d", temp->info.meals[i]);**

**temp = temp->next;**

**}**

**}**

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

**}**

**void membersMealByDate(void)**

**{**

**char ch, name[15];**

**int day, index;**

**Members temp;**

**do**

**{**

**system("CLS");**

**printf("\n\n\n\n\n%40s", " ");**

**printf("Enter Member Name And Day to See\n");**

**printf("%40sMembers Meal Of a Particular Day.\n", " ");**

**printf("\n\n%40sPress 'e' For Enter Or 'c' for Cancel : ", " ");**

**scanf(" %c", &ch);**

**if(ch == 'e')**

**{**

**printf("\n\n%40sName : ", " ");**

**scanf("%s", name);**

**printf("\n\n%40sDay (1 to 31) : ", " ");**

**scanf("%d", &day);**

**if(day < 1 || day > 31)**

**{**

**printf("Invalid Day Input\n");**

**}**

**else**

**{**

**index = search(name);**

**if(index != -1)**

**{**

**temp = get(index);**

**printf("\n\n%40s%s Have %d Meal On day %d\n", " ", name, temp.meals[day], day);**

**}**

**}**

**}**

**else if(ch != 'c')**

**{**

**printf("Invalid Command\n");**

**}**

**system("pause");**

**}while(ch != 'c');**

**}**

**void displayWelCome(void)**

**{**

**int i;**

**printf("\n\n\n\n%30s", " ");**

**for(i=0; i<50; i++)**

**{**

**printf("%c", '-');**

**}**

**printf("\n\n%31s%40s\n\n", " ", "Welcome To Meal Management System");**

**printf("%30s", " ");**

**for(i=0; i<50; i++)**

**{**

**printf("%c", '-');**

**}**

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

**printf("%35s%10s%-30s\n\n", "[ 1 ]", " ", "Login To Meal Table");**

**printf("%35s%10s%-30s\n\n", "[ 2 ]", " ", "Create New Meal Table");**

**printf("%35s%10s%-30s\n\n", "[ 0 ]", " ", "Exit From The System");**

**printf("%30s", " ");**

**for(i=0; i<50; i++)**

**{**

**printf("%c", '-');**

**}**

**printf("\n\n%35s%-30s", " ", "Enter Your Choice : ");**

**}**

**void displayMainMenu(void)**

**{**

**int i;**

**printf("\n\n\n\n%40s", " ");**

**printf("Meal Management System\n");**

**printf("%30s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

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

**printf("%40s%10s%-30s\n", "[ 1 ]", " ", "MEMBERS" );**

**printf("%40s%10s%-30s\n", "[ 2 ]", " ", "DAILY TASK" );**

**printf("%40s%10s%-30s\n", "[ 3 ]", " ", "CHECK REPORT" );**

**printf("%40s%10s%-30s\n", "[ 0 ]", " ", "LOGOUT" );**

**printf("\n\n%30s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

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

**printf("%35s%-25s%2s", " ", "ENTER YOUR CHOICE : " , " ");**

**}**

**void displayMemberScreen(void)**

**{**

**int i;**

**printf("\n\n\n\n%40s", " ");**

**printf("Member Section\n");**

**printf("%30s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

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

**printf("%40s%10s%-30s\n", "[ 1 ]", " ", "Add New Member" );**

**printf("%40s%10s%-30s\n", "[ 2 ]", " ", "Delete Member" );**

**printf("%40s%10s%-30s\n", "[ 3 ]", " ", "Search Member" );**

**printf("%40s%10s%-30s\n", "[ 4 ]", " ", "View All Member" );**

**printf("%40s%10s%-30s\n", "[ 5 ]", " ", "Add Balance" );**

**printf("%40s%10s%-30s\n", "[ 0 ]", " ", "Back To Main Menu" );**

**printf("\n\n%30s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

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

**printf("%35s%-25s%2s", " ", "ENTER YOUR CHOICE : " , " ");**

**}**

**void displayDailyTaskSection(void)**

**{**

**int i;**

**printf("\n\n\n\n%40s", " ");**

**printf("Meal Management System\n");**

**printf("%30s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

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

**printf("%40s%10s%-30s\n", "[ 1 ]", " ", "Update Meal" );**

**printf("%40s%10s%-30s\n", "[ 2 ]", " ", "Update Cost" );**

**printf("%40s%10s%-30s\n", "[ 0 ]", " ", "Back To Main Menu" );**

**printf("\n\n%30s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

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

**printf("%35s%-25s%2s", " ", "ENTER YOUR CHOICE : " , " ");**

**}**

**void displayReportSection(void)**

**{**

**int i;**

**printf("\n\n\n\n%40s", " ");**

**printf("Meal Management System\n");**

**printf("%30s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

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

**printf("%40s%10s%-30s\n", "[ 1 ]", " ", "View FUll Report" );**

**printf("%40s%10s%-30s\n", "[ 2 ]", " ", "Members Meal By Date" );**

**printf("%40s%10s%-30s\n", "[ 3 ]", " ", "Monthly Log" );**

**printf("%40s%10s%-30s\n", "[ 4 ]", " ", "Make A File" );**

**printf("%40s%10s%-30s\n", "[ 0 ]", " ", "Back To Main Menu" );**

**printf("\n\n%30s", " ");**

**for(i=0; i<40; i++)**

**{**

**printf("%c", '-');**

**}**

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

**printf("%35s%-25s%2s", " ", "ENTER YOUR CHOICE : " , " ");**

**}**

**void createMealTable(void)**

**{**

**char monthName[15], pass[15];**

**int i;**

**system("CLS");**

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

**printf("%35s%-30s\n", " ", "FILL THE FORM BELOW");**

**printf("%35s", " ");**

**for(i=0; i<20; i++)**

**{**

**printf("%c", '-');**

**}**

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

**printf("%35s%-25s%3s", " ", "Enter Month Name: ", " ");**

**scanf("%s", &monthName);**

**printf("\n\n%35s%-25s%3s", " ", "Enter Your Password: ", " ");**

**scanf("%s", &pass);**

**strcpy(monthNames[tableSize], monthName);**

**strcpy(password[tableSize], pass);**

**tableSize++;**

**printf("\n\n\n\n%30s", " ");**

**printf("Your Account Successfully Created...!\n\n");**

**printf("%30s", " ");**

**printf("You Can Now Login To System\n\n\n\n");**

**}**

**int login(void)**

**{**

**int i;**

**char name[15], pass[15];**

**system("CLS");**

**printf("\n\n\n\n%35s", " ");**

**printf("Login To Meal Management System");**

**printf("\n\n%35s", " ");**

**printf("Enter Month Name And Password Correctly");**

**printf("\n\n%30s", " ");**

**for(i=0; i<50; i++)**

**{**

**printf("%c", '-');**

**}**

**printf("\n\n\n%35s", " ");**

**printf("%-15s%3s", "Month Name : ", " ");**

**scanf("%s", &name);**

**printf("\n\n%35s", " ");**

**printf("%-15s%3s", "Password : ", " ");**

**scanf("%s", &pass);**

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

**if(isMatchID(name, pass) == 1)**

**{**

**printf("\n%30s%-30s\n\n", " ", "Access Granted");**

**strcpy(currentMonth, name);**

**strcpy(currentPass, pass);**

**readMembersInfo();**

**readCostInfo();**

**return 1;**

**}**

**return -1;**

**}**

**int isMatchID(char name[], char pass[])**

**{**

**int i;**

**for(i=0; i<tableSize; i++)**

**{**

**if(strcmp(monthNames[i], name) == 0 && strcmp(password[i], pass) == 0)**

**{**

**return 1;**

**}**

**}**

**return -1;**

**}**

**void readLoginFile()**

**{**

**FILE \*loginFile;**

**char names[15], passw[15];**

**loginFile = fopen("logins.dat", "r+");**

**if(loginFile == NULL)**

**{**

**loginFile = fopen("logins.dat", "w");**

**if(loginFile == NULL)**

**{**

**exit(0);**

**}**

**}**

**while(fscanf(loginFile, "%s %s", &names, &passw) != EOF)**

**{**

**strcpy(monthNames[tableSize], names);**

**strcpy(password[tableSize], passw);**

**tableSize++;**

**}**

**fclose(loginFile);**

**}**

**void writeLoginFile()**

**{**

**FILE \*loginFile;**

**int i;**

**loginFile = fopen("logins.dat", "w+");**

**if(loginFile == NULL)**

**{**

**exit(0);**

**}**

**for(i=0; i<tableSize; i++)**

**{**

**fprintf(loginFile, "%s %s\n", monthNames[i], password[i]);**

**}**

**fclose(loginFile);**

**}**

**void readMembersInfo()**

**{**

**FILE \*memberFile;**

**char filename[30];**

**char name[30];**

**int meal, i;**

**Members temp;**

**strcpy(filename, currentMonth);**

**strcat(filename, ".dat");**

**memberFile = fopen(filename, "r+");**

**if(memberFile == NULL)**

**{**

**memberFile = fopen(filename, "w");**

**if(memberFile == NULL)**

**{**

**exit(0);**

**}**

**}**

**while(fscanf(memberFile, "%s", &name) != EOF)**

**{**

**strcpy(temp.name, name);**

**fscanf(memberFile, "%lf %lf", &temp.pay[0], &temp.pay[1]);**

**for(i=0; i<32; i++)**

**{**

**fscanf(memberFile, "%d", &meal);**

**temp.meals[i] = meal;**

**}**

**addMember(temp);**

**}**

**fclose(memberFile);**

**}**

**void writeMembersInfo()**

**{**

**FILE \*memberFile;**

**char fileName[30];**

**int i;**

**MemberList \*temp = head;**

**strcpy(fileName, currentMonth);**

**strcat(fileName, ".dat");**

**memberFile = fopen(fileName, "w+");**

**if(memberFile == NULL)**

**{**

**exit(0);**

**}**

**while(temp != NULL)**

**{**

**fprintf(memberFile, "%s\n", temp->info.name);**

**fprintf(memberFile, "%.2lf %.2lf\n", temp->info.pay[0], temp->info.pay[1]);**

**for(i=0; i<32; i++)**

**{**

**fprintf(memberFile, "%d ", temp->info.meals[i]);**

**}**

**fprintf(memberFile, "%s", "\n");**

**temp = temp->next;**

**}**

**fclose(memberFile);**

**}**

**void readCostInfo()**

**{**

**FILE \*file;**

**int i = 0;**

**double cost;**

**char fileName[30];**

**strcpy(fileName, currentMonth);**

**strcat(fileName, "costs.dat");**

**file = fopen(fileName, "r+");**

**if(file == NULL)**

**{**

**file = fopen(fileName, "w");**

**if(file == NULL)**

**{**

**exit(0);**

**}**

**}**

**while(fscanf(file, "%lf", &cost) != EOF)**

**{**

**costTable[i] = cost;**

**i++;**

**}**

**fclose(file);**

**}**

**void writeCostInfo()**

**{**

**int i;**

**FILE \*file;**

**char fileName[30];**

**strcpy(fileName, currentMonth);**

**strcat(fileName, "costs.dat");**

**file = fopen(fileName, "w+");**

**if(file == NULL)**

**{**

**exit(0);**

**}**

**for(i=0; i<DAY; i++)**

**{**

**fprintf(file, "%.2lf\n", costTable[i]);**

**}**

**fclose(file);**

**}**

**void reset()**

**{**

**int i;**

**head = NULL;**

**listSize = 0;**

**currentMonth[0] = '\0';**

**currentPass[0] = '\0';**

**for(i=0; i<DAY; i++)**

**{**

**costTable[i] = 0.0;**

**}**

**}**

**REFERENCES**