**WATER VENDING MACHINE MANAGEMENT SYSTEM**

***Mini Project Report***

***Submitted in partial fulfilment of the***

***Requirements for the award of the Degree of***

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

BY

**<E KAVYA><1602-19-737-019>**

**<RAMAKISHAN RAO GARI SUSHMA><1602-19-737-049>**

****

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2020**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**

****

**DECLARATION BY THE CANDIDATE**

We, **E KAVYA** and **RAMAKISHAN RAO GARI SUSHMA** , bearing hall ticket numbers, **1602-19-737-019** and **1602-19-737-049** , hereby declare that the project report entitled **”WATER VENDING MACHINE MANAGEMENT SYSTEM”** Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Information Technology

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**E Kavya**

**1602-19-737-019**

**Ramakishan Rao Gari Sushma**

**1602-19-737-049**

**(Faculty In-Charge) (Head,Dept of IT)**

**ACKNOWLEDGEMENT**

It gives us immense pleasure to thank the **department of INFORMATION TECHNOLOGY**, for introducing the subject **“mini project”** in BE third semester that let us learn and explore more features in “C programming language”.

I would also like to show my appreciation to our honourable principal, **Dr S V RAMANA** sir, for supporting us and our beloved mini project lecturer, **Mrs LEELAVATHI** mam, for letting us properly understand the process of doing the mini project using c and for providing insight and expertise that greatly assisted the project.

My parents were my first teachers and they have provided me with such a great exposure that has helped me bloom. My family and friends will always be loved for sticking by me through thick and thin. THANK YOU!

**ABSTRACT**

Water vending machine is a system which dispenses water by showing a smart card or inserting coins into it. Nowadays, vending machines are available and operating on smart cards but the data of card usage is not being stored for further use by machineowner. Through our program, the machineowner will be able to check the time and date of last card usage. Also, the owner and customer will be able to know the balance amount left in the customer's smart card. Customers can also take a new card or recharge his card. So, in case a user loses his card or card doesn’t work, as the machine owner knows how much money is left in his card, he can issue a new card accordingly. This is the aim of our project.

**INTRODUCTION**

Water is a very important resource needed to sustain life, and safe drinking water is a fundamental requirement for human life. In India, Mineral water plants are run by the government to provide clean drinking water to people. In very few places, these water plants are run by people whereas in many parts of our country water plants are completely operated on water vending machines. Not only by the government, but also many private businesses are using these water vending machines. It reduces human effort and also avoids water wastage which is one of the main concerns we have now.

Water vending machines dispense water by showing a smart card or inserting coins into it. So to get water from this machine, one must show their smart card or insert coins into it. Those who want to take water inserting coins don't need smart cards.But, everytime a registered user takes water using his smart card, the amount will get deducted from his card based on the quantity of water he takes. Once the amount left in the smart card is not sufficient to take water, the user must recharge it with the required amount.

**What we wanted to achieve through this project**

The user has to first register inorder to get a smartcard. The owner will give a smartcard to the registered user and the smartcard will have some minimum balance initially. The user can request for recharging the smartcard whenever he wants and the owner will recharge it. The details of all the registered users will be stored in files. When the user wants to take water using his smartcard,he has to login using his registered username and password and he has to enter the number of litres he wants to take. If the amount in the smartcard is sufficient,then the water will be given to the user. Otherwise,the user has to recharge his card or else he can reduce the number of litres and check if the amount in his smartcard is sufficient. If the user enters incorrect login details, an error message will be displayed saying that the user has not been registered. The date and time of the last usage of the smart card will be recorded.

If the user wants to check the balance left in his smartcard before taking water, there is an option in the menu called “check balance in the smartcard”. The user has to login with his username and password. If the login credentials are correct,the balance left in his smartcard will be shown. If the registered user does not want to use the water vending machine system anymore,he can unregister anytime. He needs to select the option “unregister” from the main menu and enter his login credentials. If his login details are correct, he will be successfully unregistered and his details will be removed from the file.

An unregistered user can also take water from the machine. He needs to give coins to the owner in exchange for water. The details of all the users who are taking water using coins will be stored in a separate file.

The owner will have access to the details of all the registered users. He can check the balance left in their smartcards and the date and time of their last usage by just entering their username. The owner has to recharge the smartcard whenever the user requests.

**TECHNOLOGY**

In implementing our project, the technological software and hardware requirements we used are-

**a) Software requirements**:

**Compiler:**

Since our project was supposed to be based on the C programming language, it is a bare necessity to have the knowledge and syntaxes of the language and a proper compiler and text editor to run and write the programs. Among the many available compilers, we have installed and used the GCC compiler to run/execute the code for our mini project that we have implemented.

**Text editor:**

To actually write and complete a code in any language, a text editor is important. We have installed and used a text editor called **“**Sublime Text” and “atom”. Around 700 lines of the source code for “WATER VENDING MACHINE MANAGEMENT SYSTEM” implementation in C has been written in this editor.

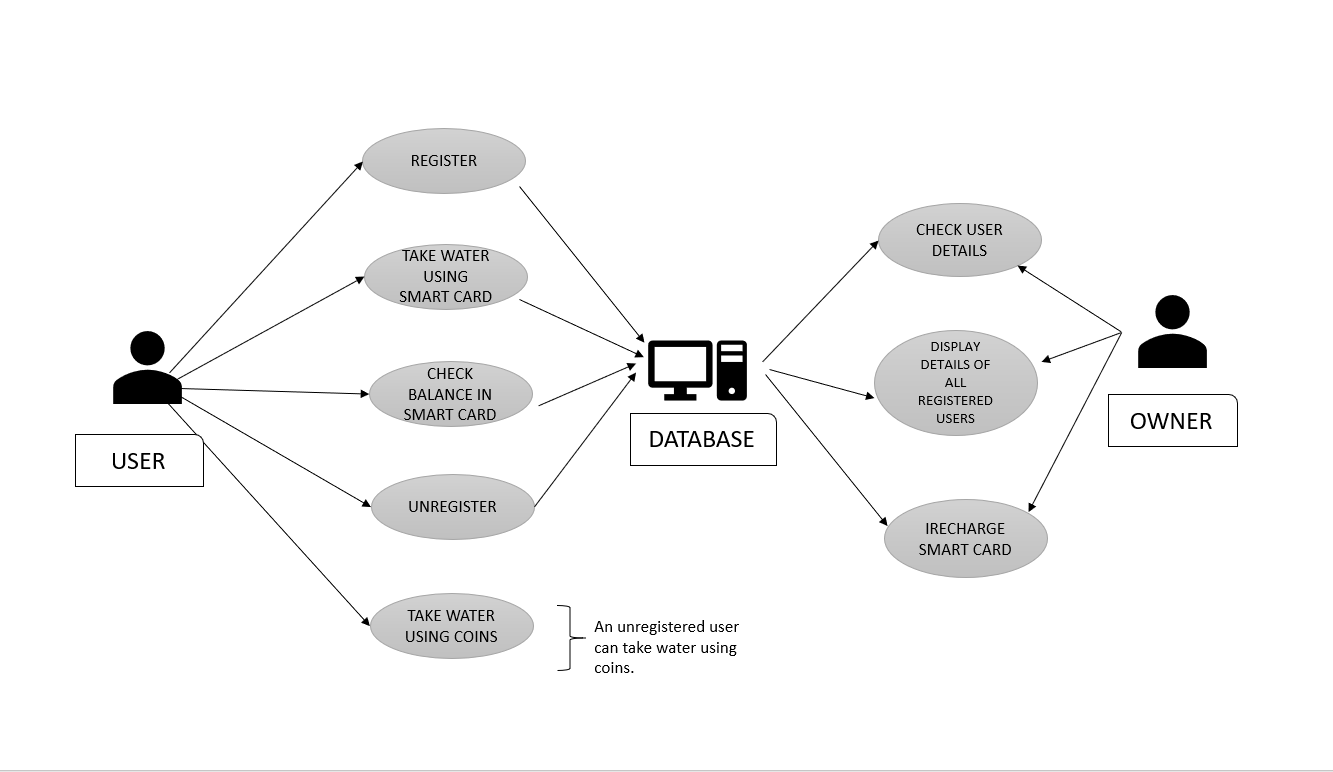
**b) Hardware requirements**:

Personal computer or Laptop(Prefer Windows)

**PROPOSED WORK**

**a.DESIGN**

**Use-case diagram**

****

**b.IMPLEMENTATION**

Our project “WATER VENDING MACHINE MANAGEMENT SYSTEM” has separate use cases for smartcard users, owner and unregistered users also. User has six use cases. They are-

1. Register
2. Take water using smartcard
3. Check balance in smartcard
4. Unregister
5. Take water using coins ( For unregistered users)

Users will have to give their username, password, mobile number to register. Then the owner will issue a smart card. These details will be stored in a file. This is our register module.

void registration()

{

FILE \*fp;

struct user t1;

char num[20];

char uname[20];

fp=fopen(fname,"a+");

printf("\n\n\t\t\t\t\tPLEASE ENTER THE FOLLOWING DETAILS TO REGISTER !!");

pqr:

printf("\n\n\n\t\t\tENTER USERNAME: ");

scanf("%s",uname);

while(1)

{

fread(&t1,sizeof(t1),1,fp);

if(feof(fp))

{

break;

}

if(strcmp(t1.user\_name,uname)==0)

{

printf("\n\n\n\t\t\t USERNAME ALREADY EXISTS. PLEASE ENTER A DIFFERENT USERNAME!!");

goto pqr;

}

}

strcpy(t1.user\_name,uname);

printf("\n\t\t\tENTER PASSWORD :");

scanf("%s",t1.password);

xy:

printf("\n\t\t\tENTER PHONE NUMBER :");

scanf("%s",num);

if(strlen(num)!=10)

{

printf("\n\t\t\t INVALID NUMBER. PLEASE ENTER THE NUMBER AGAIN !! \n");

goto xy;

}

else if(num[0]=='0')

{

printf("\n\t\t\t INVALID NUMBER. PLEASE ENTER THE NUMBER AGAIN !! \n");

goto xy;

}

else

{

strcpy(t1.phone\_no,num);

}

printf("\n\n\n\t\t\tYOUR CARD WILL BE RECHARGED WITH A BALANCE OF Rs 500 ");

t1.min\_balance=500;

fwrite(&t1,sizeof(t1),1,fp);

fclose(fp);

printf("\n\n\t\t\tYOU HAVE SUCCESSFULLY REGISTERED!!\n\n\n");

exit(0);

}

When the registered user wants to take water using his smartcard, he has to enter login credentials and the number of litres he wants to take. The balance in his card before and after using it will be displayed on the screen. The date and time of the last usage of the smart card will be recorded. This is our second use case.

void take\_water\_using\_smartcard()

{

int litres;

char name[20],pass[20];

FILE \*fp,\*fp1;

struct user t,t1;

int found=0,count=0;

fp=fopen(fname,"r");

fp1=fopen("temp.dat","w");

printf("\n\n\n\t\t\t\t\tPLEASE ENTER YOUR LOGIN CREDENTIALS TO TAKE WATER ");

printf("\n\n\n\t\t\tENTER USERNAME: ");

scanf("%s",name);

printf("\n\t\t\tENTER PASSWORD :");

scanf("%s",pass);

while(1)

{

fread(&t,sizeof(t),1,fp);

if(feof(fp))

{

break;

}

if((strcmp(t.user\_name,name)==0)&&(strcmp(t.password,pass)==0))

{

found=1;

fflush(stdin);

printf("\n\n\t\t\t THE COST OF WATER FOR 1 LITRE IS Rs 10\n\n ");

printf("\t\t\t ENTER THE NUMBER OF LITRES YOU WANT TO TAKE :");

scanf("%d",&litres);

printf("\n\t\t\t YOU WILL BE CHARGED : %d",(litres\*10));

if((litres\*10)<=t.min\_balance)

{

printf("\n\n\t\t\t BALANCE IN YOUR SMART CARD BEFORE TAKING WATER IS : %d",t.min\_balance);

printf("\n\n\t\t\t Rs %d WILL BE DEDUCTED FROM YOUR SMART CARD",(litres\*10));

printf("\n\n\t\t\t BALANCE IN YOUR SMART CARD AFTER TAKING WATER IS : %d",(t.min\_balance-(litres\*10)));

time\_t raw;

time(&raw);

struct tm \*time\_ptr;

time\_ptr = localtime(&raw);

char current\_date[11];

char current\_time [20];

strftime(current\_date, sizeof(current\_date), "%m/%d/%Y", time\_ptr);

strftime(current\_time, sizeof(current\_time), "%H:%M:%S", time\_ptr);

strcpy(t.date,current\_date);

strcpy(t.time,current\_time);

printf("\n\n\n\n\t\t\t\t\t\tTHANK YOU\n\n");

t.min\_balance=t.min\_balance-(litres\*10);

fwrite(&t,sizeof(t),1,fp1);

}

else

{

printf("\n\t\t\t BALANCE IN YOUR SMART CARD IS : %d",t.min\_balance);

printf("\n\n\t\t\t THIS AMOUNT IS NOT SUFFICIENT TO TAKE THE WATER YOU REQUESTED. PLEASE RECHARGE YOUR SMART CARD");

fwrite(&t,sizeof(t),1,fp1);

}

}

else

{

fwrite(&t,sizeof(t),1,fp1);

}

}

fclose(fp);

fclose(fp1);

if(found==0)

{

printf("\n\t\t\t SORRY NO RECORD FOUND !! \n\n\n");

}

else

{

fp=fopen(fname,"w");

fp1=fopen("temp.dat","r");

while(1)

{

fread(&t,sizeof(t),1,fp1);

if(feof(fp1))

{

break;

}

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

}

}

fclose(fp);

fclose(fp1);

exit(0);

}

Similarly, we have separate modules for checking balance, unregister use cases.

Unregistered users should also enter their name, mobile number and number of litres of water they want to take. These details of all unregistered users are stored in a separate file.Then, the amount given by the user will be displayed on the screen. After giving that money, he can take water.

Owner has four use cases. They include-

1. Check user details
2. Recharge smartcard
3. Display details of registered users
4. Display less balance users

Owner has to enter his password. If the password is correct, he can check details of every user by entering the username of the card user. The details displayed are-

1. Mobile number
2. Balance
3. Date and time of last card usage

void check\_user\_details(){

int found=0,count=0,amount,c=0;

char name[20];

FILE \*fp;

struct user t4;

fp=fopen(fname,"r");

printf("\n\n\n\t\t\t\t\tPLEASE ENTER USERNAME OF ANY CARD USER : ");

scanf("%s",name);

while(1)

{

fread(&t4,sizeof(t4),1,fp);

if(feof(fp))

{

break;

}

if((strcmp(t4.user\_name,name)==0))

{

found=1;

printf("\n\n\t\t\t\t MOBILE NUMBER : %s",t4.phone\_no);

printf("\n\n\t\t\t\t BALANCE LEFT IN %s 's SMART CARD IS : %d",name,t4.min\_balance);

printf("\n\n\t\t\t\t CARD WAS LAST USED ON : ");

printf("\n\n\t\t\t\t DATE : %s",t4.date);

printf("\n\n\t\t\t\t TIME : %s\n\n\n",t4.time);

break;

}

}

fclose(fp);

if(found==0)

{

printf("\n\n\t\t\t\t\tSORRY, NO RECORD FOUND !!\n\n\n\n");

}

exit(0);

}

Owner can also recharge smartcards by entering just the username of any card user.

Balance left in the smartcard before and after using it will be displayed on the screen. These changes will be updated in the file records. This is our recharge module.

void recharge()

{

int found=0,count=0,amount;

char name[20];

FILE \*fp,\*fp1;

struct user t,t1;

fp=fopen(fname,"r");

fp1=fopen("temp.dat","w");

system("cls");

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

printf("\n\n\n\t\t\t\t\t\t\tWELCOME TO RECHARGE ZONE !! ");

printf("\n\n\n\t\t\t\t ENTER USERNAME: ");

scanf("%s",name);

while(1)

{

fread(&t,sizeof(t),1,fp);

if(feof(fp))

{

break;

}

if((strcmp(t.user\_name,name)==0))

{

found=1;

fflush(stdin);

printf("\n\t\t\t\t BALANCE IN YOUR SMART CARD BEFORE RECHARGING : %d",t.min\_balance);

printf("\n\n\t\t\t\t ENTER THE AMOUNT TO RECHARGE :");

scanf("%d",&amount);

t.min\_balance=t.min\_balance+amount;

printf("\n\n\t\t\t\t AMOUNT IN THE SMART CARD AFTER RECHARGING : %d",t.min\_balance);

printf("\n\n\n\n\t\t\t\t\t\t\t THANK YOU\n\n\n");

fwrite(&t,sizeof(t),1,fp1);

}

else

{

fwrite(&t,sizeof(t),1,fp1);

}

}

fclose(fp);

fclose(fp1);

if(found==0)

{

printf("\n\n\t\t\t\t\t SORRY NO RECORD FOUND !! \n\n\n\n");

}

else

{

fp=fopen(fname,"w");

fp1=fopen("temp.dat","r");

while(1)

{

fread(&t,sizeof(t),1,fp1);

if(feof(fp1))

{

break;

}

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

}

}

fclose(fp);

fclose(fp1);

exit(0);

}

We have another use case for owner where the details of all registered users having balance less than the amount entered by the owner will be displayed on the screen.

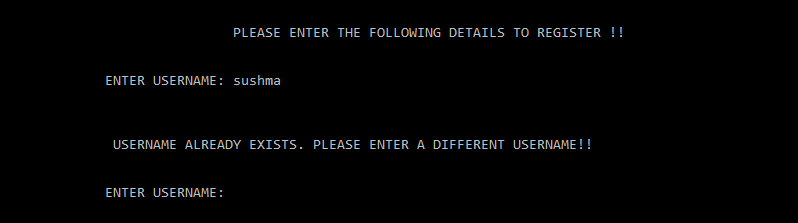
This is the last module.

**GITHUB LINKS-**

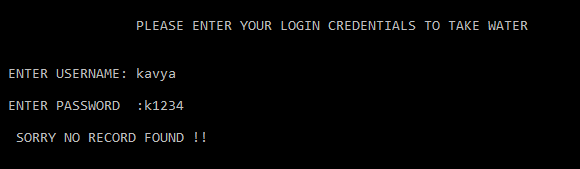
[**https://github.com/kavya-e/water-wending-machine-management-system**](https://github.com/kavya-e/water-wending-machine-management-system)

**c. TESTING**

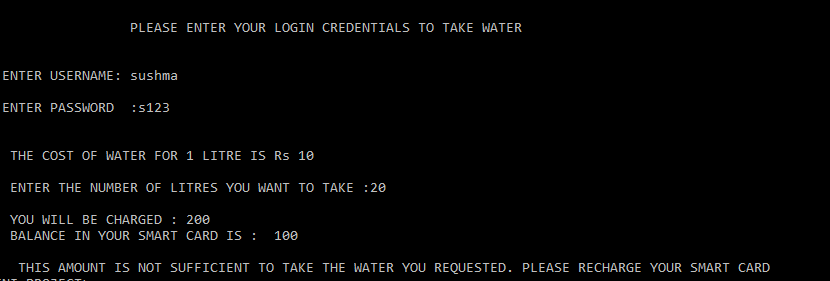
1. In the register use case if the username entered by a new user already exists, an error message is displayed on the screen saying the username already exists and asks the user to enter a different username.



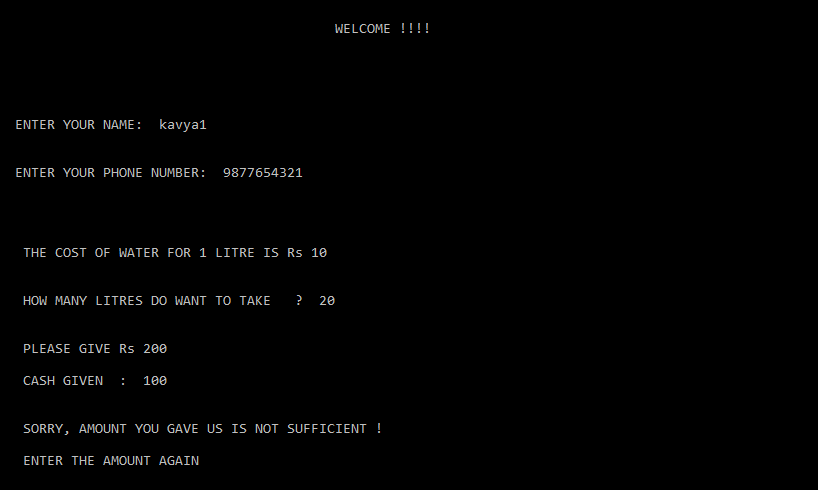
1. While entering login credentials in any other use case, if a user enters incorrect password, an error message is displayed saying no record found.



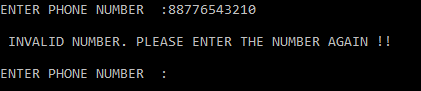
1. In taking water using the smartcard use case, If the balance left in the user's smartcard is not sufficient to take water he requested, an error message is displayed requesting the user to recharge his smartcard.

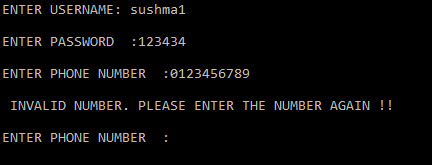


1. Similarly, in taking water using coins use case if the amount given by an unregistered user is not sufficient to take water he requested, an error message is displayed requesting the user to enter the required amount.



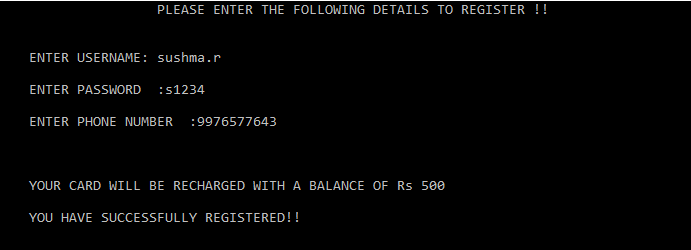
1. In register use case if new user enters invalid mobile number i.e., number having more than 10 digits or a number starting with zero, an error message is displayed on the screen asking him to enter a valid phone number.



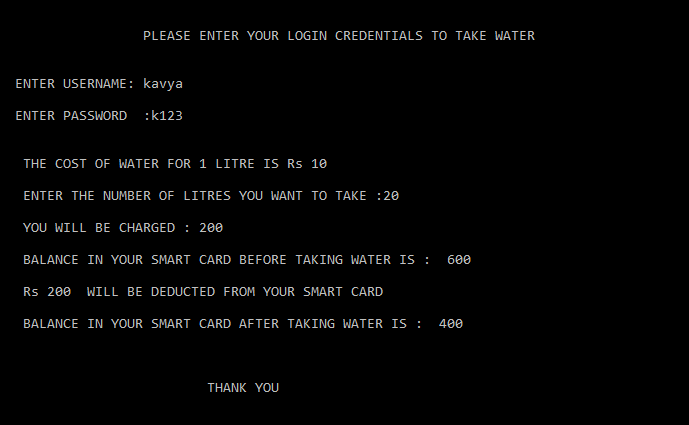


**RESULTS**

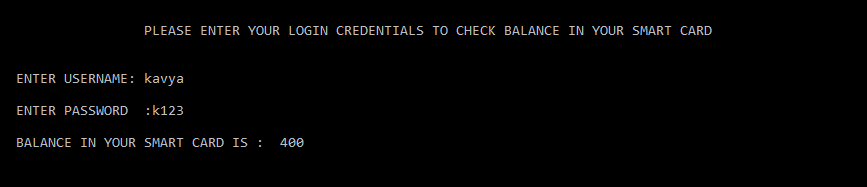
1. Output of Register use case-



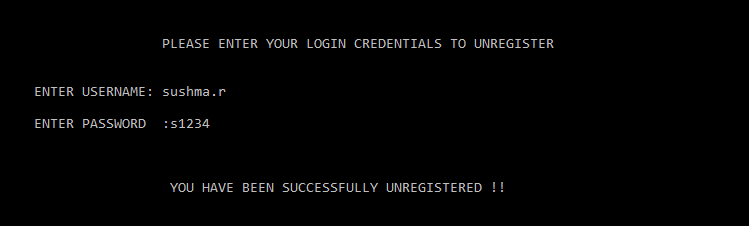
2. Output of take water using smartcard use case-



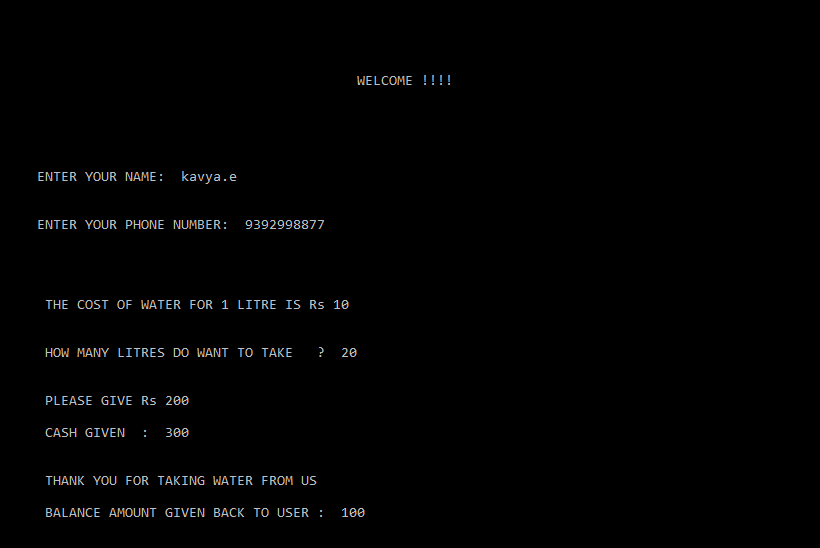
3. Output of show balance use case-



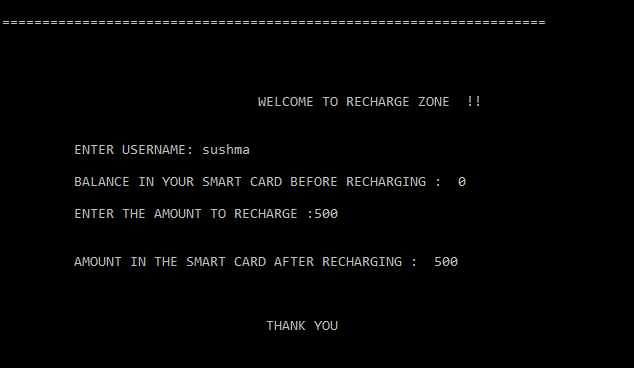
4. Output of Unregister use case-



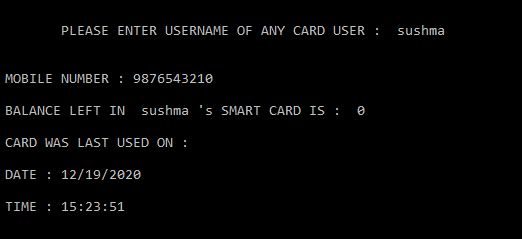
5. Output of Take water using coins use case-



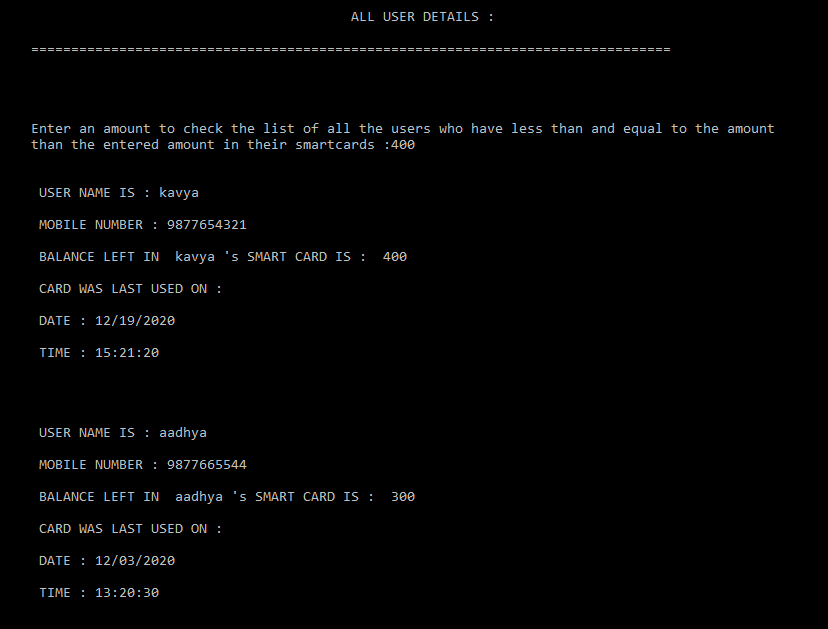
6. Output of Register use case-



7. Output of check details use case-



8. Output of check balance less balance use case-



**ADDITIONAL LEARNING**

The first thing we learnt through this project is how to be a good team player and contribute your part effectively while working together with someone in a group. Both of us contributed equally to the project and we made sure that both of us understood the implementation well. We helped each other understand and proceed with the code whenever we got struck in between. We were both very responsible for our project and managed to do the best we could. Working together really helped us to manage our time in an effective way as we could share opinions with each other.

We learned to manage our time effectively. Very often,both of us were busy with other assignments and quizzes,yet we dedicated some time everyday for the mini project and made progress.This project helped both of us to put our interest in coding in a certain work and really helped it to grow. We made plans beforehand and communicated each other’s ideas. We scheduled meetings regularly and made sure we made progress in the project every time. Working together as a team also made it possible for us to finish the work in the given time.

Through this project, we were able to explore and learn more concepts other than those which were already taught to us. We understood the importance of self learning. We taught ourselves and helped each other understand the concept better. It helped us to be more creative. It has been quite some time since we used “c” programming language and this project made us revise all the concepts again which is very helpful.

Finally, I would say we thoroughly enjoyed doing this project. As this is our first mini project, we are really glad that we could successfully complete it. It really boosted our confidence and we are looking forward to working in many more programming projects in future.

**DISCUSSION AND FUTURE WORK**

We had few other ideas about what else to add in this project but couldn't add since we did not have sufficient time and there are few concepts which we need to learn before implementing the code. We did not manage to learn many concepts as we were busy with other works. But, in future, we would definitely learn more about whatever is necessary to implement more features in our project if possible. We will not miss a chance to make our project much better than it is now.

Some of the features which we had on our mind:

2 step verification system for registration

Giving points/discount coupons to regular customers

Disabling the user account in the records if he is not a regular user

**REFERENCES**

Youtube tutorials and google. We used stack overflow whenever we got struck with errors in our code.