**BAKERY MANAGEMENT SYSTEM**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

PATLOLLA SAI SAKETH-1602-19-737-097

MANCHUKONDA AMRUTH SAI-1602-19-737-066



**Department of Information Technology Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University) Ibrahimbagh, Hyderabad-31 202**

**Vasavi College of Engineering (Autonomous) (Affiliated to Osmania University)**

**2020**

**Hyderabad-500 031**

 **Department of Information Technology**

**DECLARATION BY THE CANDIDATE**

We, PATLOLLA SAI SAKETH and MANCHUKONDA AMRUTH SAI bearing hall ticket numbers, 1602-19-737-097 and 1602-19-737-066 respectively, hereby declare that the project report entitled “BAKERY MANAGEMENT SYSTEM” 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 us 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.

PATLOLLA SAI SAKETH-1602-19-737-097

MANCHUKONDA AMRUTH SAI-1602-19-737-066

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

**Acknowledgements**

The satisfaction that accompanies the successful completion of this project would not be in complete without the mention of the people who made it possible, without whose constant guidance and encouragement would have made efforts go in vain. We consider ourselves privileged to express gratitude and respect towards all those who guided us through the completion of this project.

We convey thanks to my project guide Mrs.DRL.Prasanna of Information technology Department for providing encouragement, constant support and guidance which was of a great help to complete this project successfully. Last but not the least, we wish to thank our parents for financing our studies in this privileged Vasavi College of Engineering as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is gratefully acknowledged.

**Abstract**

In our project, on “Bakery Management System”, we have tried to show how the data/information in Bakery is managed. This is just an overview of management System in Bakeries.

Our main intention is to process the order given by the customer efficiently by displaying the menu with availability of items to the customer and also we are trying to provide the token numbers to them and calculate the approximate time for their order to get ready. And finally we show them the bill and provide required payment options to the customer.

**TABLE OF CONTENTS**

1. Introduction……………………………………………6

2. Technology…………………………………………….8

3. Proposed Work………………………………………...9

4. Results…………………………………………………24

5. Additional Knowledge Acquired………………………29

6. Conclusion and Future Work…………………………..30

7. References……………………………………………...31

**INTRODUCTION**

A Bakery management is delegated to get order form customer. It mainly saves cost, time and energy. It is very secure and time saving. Overall, it requires lower maintenance and services and ensures efficient functioning.

It allows the customer to select the items they desire. In our bakery the menu consists of five sub categories and two items in each sub categories. The five sub categories are Burgers, Pizzas, Puffs, Cakes and Pies. Burgers consists of veg burger and non-veg burger, Pizzas consists of veg pizzas and non-veg pizzas, Puffs consists of veg puffs and egg puffs, Cakes consisits of Blue berry cake and Cherry cake, Pies consists of Cherry pies and Apple pies.

**Owner User**

(At the beginning of the day)

The owner or the employee need to enter the stock available initially. The stock need to be entered daily morning. He must also need to set a four digit pin which is used to exit the program at the end of the day.

(At the end of the day)

The owner or the employee has to exit from the main menu, and need to enter the four digit pin set by him in the morning. The program exits and completes.

**Customer** **User**

The customer has to place the order by selecting the items from the menu. He can only select the items which are available, and the quantity of an items selected must be less than or equal to the quantity available. Before conforming the order, the customer can add any items to his ordered list or delete any items from his ordered list. After he conforms his order the token number and order time will be displayed. The customer need to select a payment option. The six digit bill number, name and phone number has to be entered.

**Example:**

After the owner or employee has entered the stock and sets the pin, the main menu is displayed. The customer need to select one option from sub menu, the items are displayed with availability. The customer need to select any item and has to enter the number of pieces. In this way he can select the items required. After selecting the items he will be asked whether to conform the order or add any items or delete any item from his order.

After conforming the order , the bill is printed and token number and time for his order to be ready is displayed. He has to select a payment method. He then need to enter the six digit bill number, his name and his mobile number.

At the end the owner or employee need to enter the four digit pin and exits the program.

**TECHNOLOGY:**

All compute software needs certain hardware components or other software components resources to be present. In order for computesr to used efficiently these are the primary requestisites. There are two different categories of this section.

1. **Software Requirements**:

Software requirements mainly share out with defining the software resource requirements that need to be installed on a computer to provide optimal functioning of a particular application. These conditions are not included in the software installation package and need to be installed separately.

In order to use Bakery Management System the following are the prerequisties,

a. Operating System – Windows 7 and above

b. C Compiler – GNU Compiler

c. Editor – Any basic editor is preferable (Example: Notepadd++)

**b. Hardware Requirements:**

Hardware requirements refer to the common set requirements defined by any operating system or software application and are usually the physical computer resources. In this section we basically deal with primary memory, secondary memory, processing power.

In order to use Bakery Management System one should have the following,

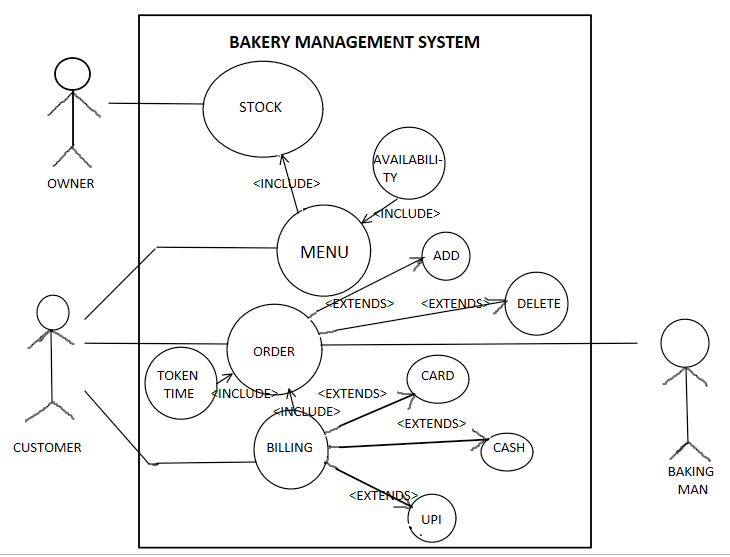
a. Processor - Intel Core i5 and above

b. Memory – 4GB Ram

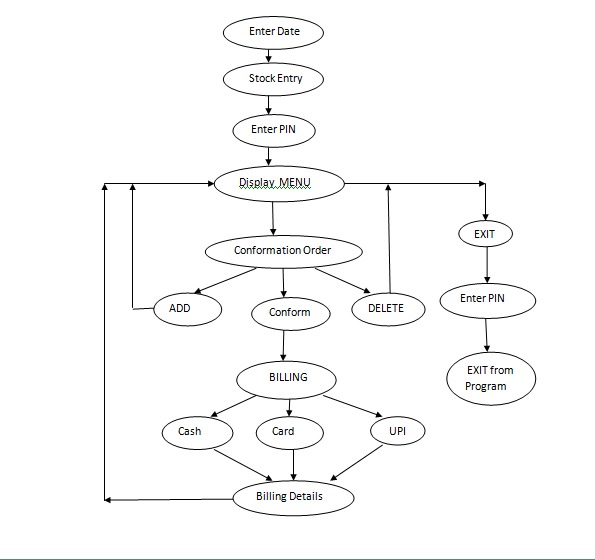
**PROPOSED WORK**

1. **Design**
2. User Case Diagram and descriptions for all the use cases

**USE CASE DIAGRAM**



**FLOW CHART**

****

1. **Implementation**

**USE CASE DESCRIPTION**

Use case ID: UC01

Name: Stock

Actors: Owner

Description: Allows to enter the stock available

Pre Condition: None

|  |  |
| --- | --- |
| **USER** | **SYSTEM** |
|  | 1)Shows the menu of items |
| 2)Enters the stock available |  |
|  | 3) Data is stored |

Post Condition: Stock available is stored

Use case ID: UC02

Name: Order

Actors: Customer

Description: Allows to enter the order

Pre Conditions: Must enter the items which are available

Post Condition: Order list is send to baking room

|  |  |
| --- | --- |
| **USER** | **SYSTEM** |
|  | 1. Menu is shown with availability |
| 1. Selects the wanted items |  |
|  | 1. Stores the list given by customer 2. Asks the customer wether to conform order |
| 1. Finalizes the order |  |
|  | 1. Prints the token number and time for order to be ready. |

Use case ID: UC03

Name: Billing

Actors: Customer

Description: Shows the bill and asks payment method

Pre Conditions: None

Post Conditions: Bill is send to payment counter

|  |  |
| --- | --- |
| **USER** | **SYSTEM** |
|  | 1. Shows the ordered list and total amount |
| 1. Enters the payment option |  |
|  | 1. Bill is sent to payment counter 2. Order is send to baking room after the payment. |

Use case ID: UC04

Name: Menu

Actors: None

Description: Shows the list lo items available

Pre conditions: Availability is checked

Post conditions: None

|  |  |
| --- | --- |
| **USER** | **SYSTEM** |
|  | 1. Availability of items is checked |
|  | 1. If item is available, it is displayed with number of pieces |

Use case ID: UC05

Name: Add

Actors: Customer

Description: Adds the items to the customer’s order

Pre conditions: None

Post condition: Order of customer is updated with new items added

|  |  |
| --- | --- |
| **USER** | **SYSTEM** |
|  | 1. Menu is displayed with availability. |
| 1. Enters the items to be added. |  |
|  | 1. Ordered list is updated. |

Use case ID: UC06

Name: Delete

Actors: Customer

Description: Deletes the items in the ordered list

Pre conditions: None

Post conditions: Order of customer is updated with items deleted

|  |  |
| --- | --- |
| **USER** | **SYSTEM** |
|  | 1. Customers ordered list is displayed. |
| 1. Selects the items to be removed |  |
|  | 1. Ordered list is updated |

**ALGORITHM**

//**Function name: void tokenandtime(int I,int \*z,char date[14])**

// **It calculates the time taken for the order to be readied**

void tokenandtime(int i,int \*z,char date[14])

{

time\_t now;

struct tm \*t;

time(&now);

t=localtime(&now);

strftime(T[i].ordered\_time,80,"%I:%M%p",t);

int k,j;

if(i==0)

{

T[0].token=1;

T[0].ordered\_hour=t->tm\_hour;

T[0].ordered\_minute=t->tm\_min;

k=ordertime(z);

T[0].order=k;

}

else

{

T[i].token=T[i-1].token+1;

T[i].ordered\_hour=t->tm\_hour;

T[i].ordered\_minute=t->tm\_min;

k=ordertime(z);

if(T[i-1].ordered\_minute>T[i].ordered\_minute)

{ T[i].ordered\_minute+=60;

--T[i].ordered\_hour;

}

j=T[i-1].order-(60\*(T[i].ordered\_hour-T[i-1].ordered\_hour))+(T[i].ordered\_minute-T[i-1].ordered\_minute);

T[i].order=k+j;

}

}

int ordertime(int \*z)

{

int sum=0;

if(\*z!=0)

sum=sum+(3\* \*z);

if(\*(z+1)!=0)

sum=sum+(3\* \*(z+1));

if(\*(z+2)!=0)

sum=sum+(4\* \*(z+2));

if(\*(z+3)!=0)

sum=sum+(4\* \*(z+3));

if(\*(z+4)!=0)

sum=sum+(2\* \*(z+4));

if(\*(z+5)!=0)

sum=sum+(2\* \*(z+5));

if(\*(z+6)!=0)

sum=sum+(2\* \*(z+6));

if(\*(z+7)!=0)

sum=sum+(2\* \*(z+7));

if(\*(z+8)!=0)

sum=sum+(3\* \*(z+8));

if(\*(z+9)!=0)

sum=sum+(3\* \*(z+9));

return sum;

}

**Github links and folder structure.**

<https://github.com/SakethSai000/Bakery-management-system>

<https://github.com/Amruth066/Bakery-Management-System>

**C. Testing**

Testing is a method to check whether the actual product matches the expected requirements and to ensure that the product is defect-free. This process involves execution of various parts of the product either using manual or automated tools. The purpose is to identify errors, gaps or missing requirements in contrast to the actual requirements.

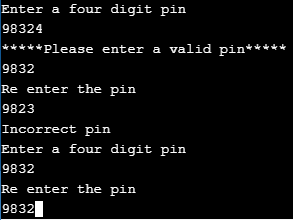
**USER TEST CASES**

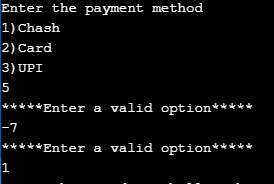
|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID:** TC01 | | **User Case ID:**  UC01 |
| **Test Case Title:** Setting a four digit Pin | |
| **Test Case Description:** owner or employee sets a pin which is used to exit the program at the end of the day. | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays prompt for user to enter four digit pin. And reenter the same pin. | System displays nothing or Invalid | If the pin entered correctly displays nothing and if entered anything incorrectly, then prints Invalid |

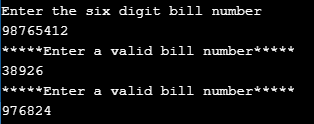
|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID:** TC02 | | **User Case ID:**  UC02 |
| **Test Case Title:** selecting a payment method | |
| **Test Case Description:** user selects a payment method to pay the bill | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays prompt for user to enter the payment method | System display nothing or Invalid | If wrong input is entered it displays invalid or else nothing is displayed |

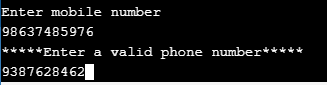
|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID:** TC03 | | **User Case ID:**  UC03 |
| **Test Case Title:** Entering six digit bill number | |
| **Test Case Description:** user enters a specific six digit bill number | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays prompt for user to enter the bill number | System should display Invalid or nothing is displayed | If wrong input is entered it displays invalid or else nothing is displayed |

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID:** TC04 | | **User Case ID:**  UC04 |
| **Test Case Title:** Entering a phone number | |
| **Test Case Description:** user enters a phone number | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays prompt for user to enter the phone number | System should display Invalid or nothing is displayed | If wrong input is entered it displays invalid or else nothing is displayed |

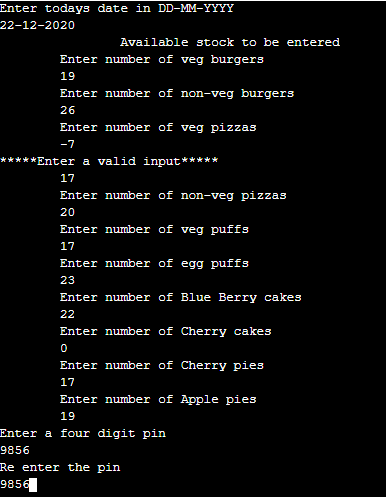
****

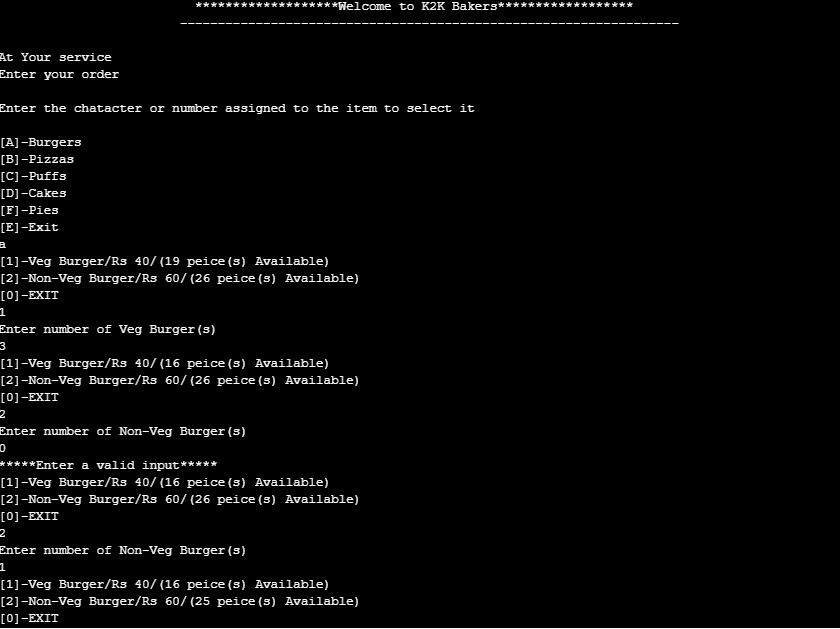
****

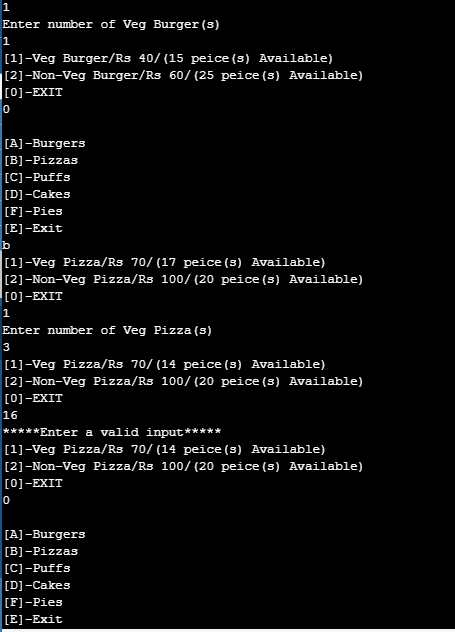
****

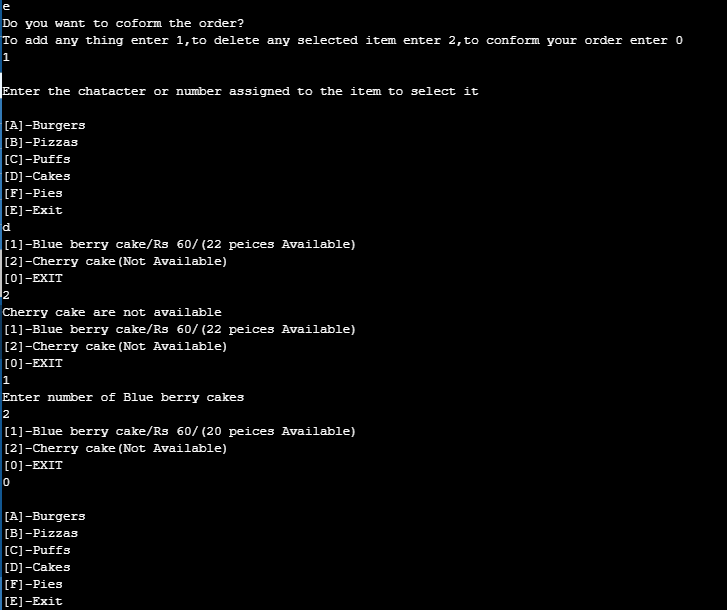
****

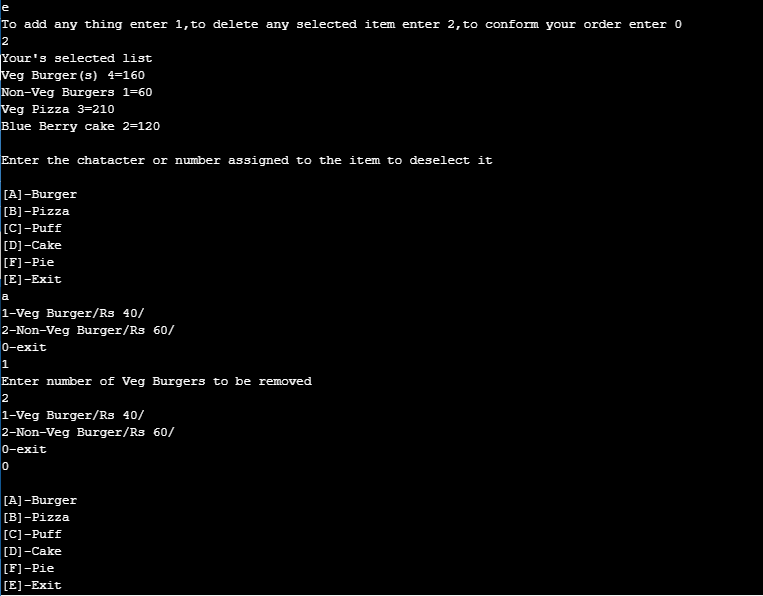
**RESULTS**

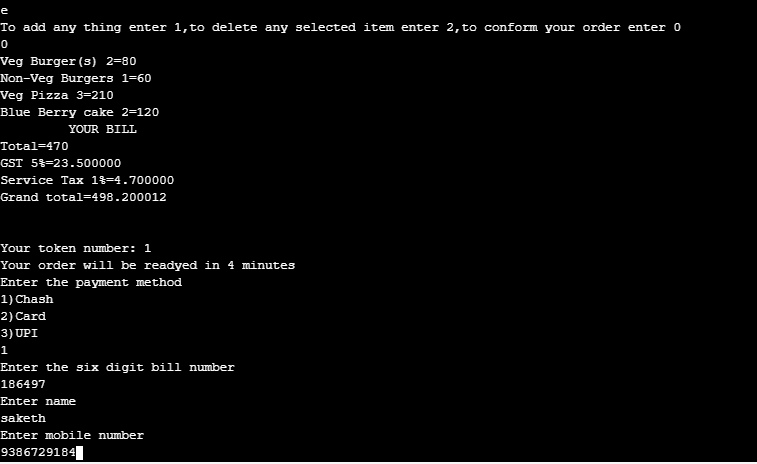


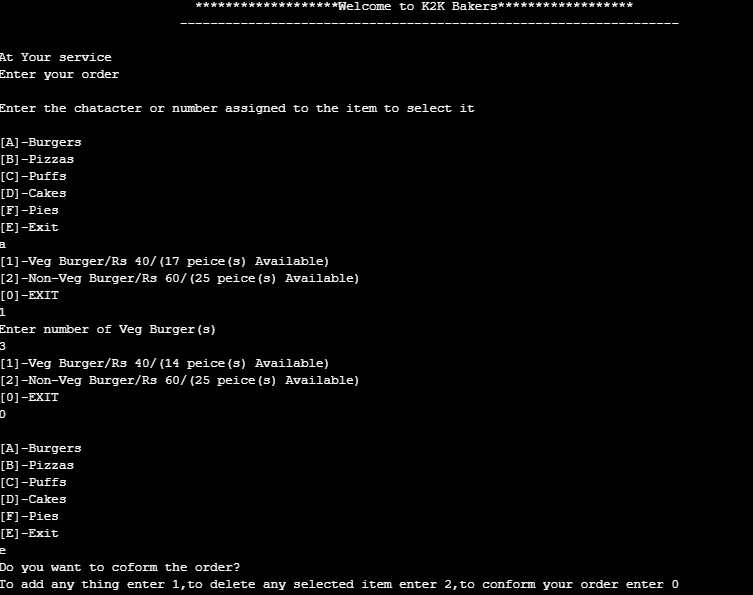


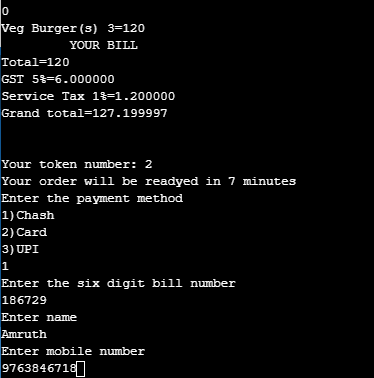


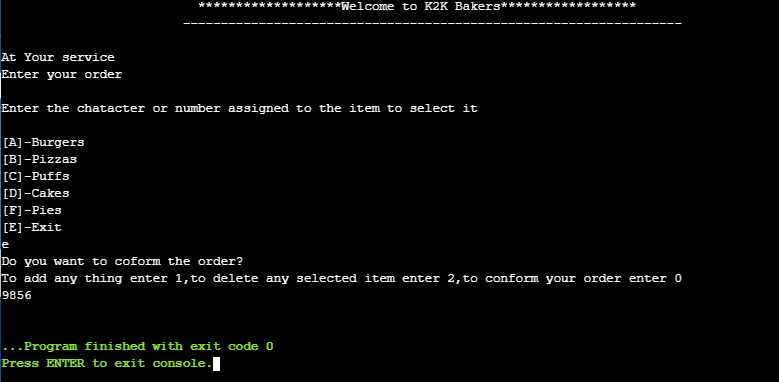


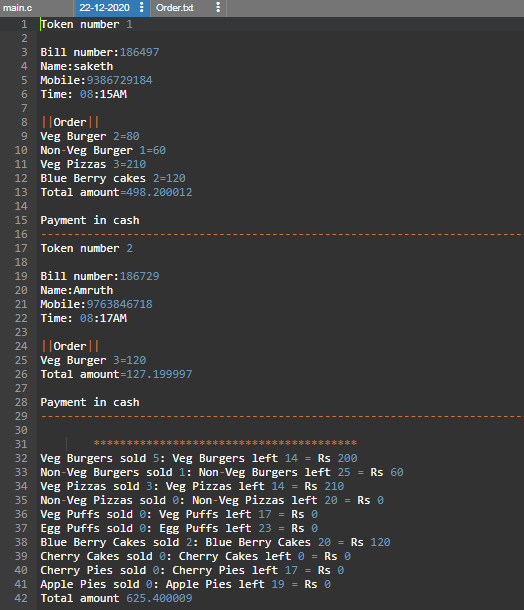


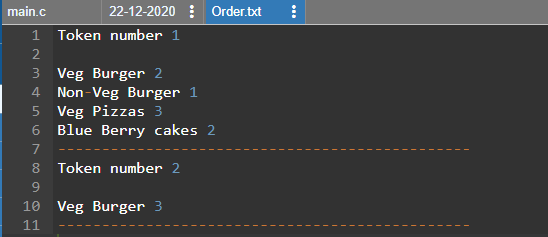












Our bakery management system has some features which are not available in existing systems. One of which is, we display the menu with availability of items. If an item is not available then it displays “Not Available”, if the item is available then it displays the number of pieces of the item available. And as the customers order, the availability of items get updated. In the existing system the employees need to check whether the item is available, manually. But in our system the availability of items is always kept track.

The second feature is, we give to the customer token number as well as the time taken for their order to be ready. This gives the customers an idea of they want to do in that time, they can wait or can come after the order is ready. This feature is not available in existing systems. We also provide the details of the items sold for that day. Items left out and the income from each item on that day. This gives the owner an idea on which items he need to sell or remove them from the menu.

**ADDITIONAL KNOWLEDGE ACQUIRED**

Implementing this project in C Language has introduced us to different libraries such as: ‘conio.h’, ‘time.h’ . We were able to use the knowledge we have on the Linked List Data Structure and execute it as a real-time application. We used the ‘time.h’ library to display time taken to prepare the order given by the customer in a controlled manner( to construct our own time). We explored the ‘string.h’ and ‘stdlib.h’ libraries for various functions for performing general functions and manipulating arrays of characters .

Also, we have further improved in our knowledge in file-handling because of the vast amount of data manipulation we have done using text files.

Other than this, we have learnt the value of team spirit and have understood the intention behind working in teams. We have learnt to be team players.

**CONCLUSION AND FUTURE WORK**

Our main intention is to process the order given by the customer efficiently by displaying the menu with availability of items to the customer and also we are trying to provide the token numbers to them and calculate the approximate time for their order to get ready. And finally we show them the bill and provide required payment options to the customer

Future work is to we develop the application in PHP Or HTML to make the real payment, make this console application as a web application.

**REFERENCES**

Reading and Writing data from and into csv file

https://www.geeksforgeeks.org/csv-file-management-using-c/

Stack Overflow (for debugging errors)

https://stackoverflow.com/