./

Report – Food Order Management System

Course Code: <CODE>



Documented by-

Name: N Prakruthi

SSID: 104288

Version Number:

Team Members :

Team No:

Module: Model Based System Engineering

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ver.Rel. No.** | **Release Date** | **Prepared. By** | **Reviewed By** | **Approved By** | **Remarks/Revision Details** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Document History**

**TABLE OF CONTENTS**

1. Problem Statement         3
2. Project Description         3
3. Requirement specification                                                                                                   4
4. Design                                                                                                                                  6
5. Test Cases                                                                                                                             7
6. Result and Conclusion                                                                                             8

**LIST OF TABLES**

1. Interactive food menu                                                                                                8
2. Admin login                                                                                                                 8
3. Search employee record                                                                                             9
4. Delete employee record                                                                                                   9
5. Add food item to menu                                                                                                   9
6. Delete food item from menu                                                                                               10

**LIST OF DIAGRAMS**

1. Use-Case diagram                                                                                                                7

**1.  Problem statement:**

Long queues are an inevitable part of dining in a fast-casual restaurant where customers have to line up to place their order. Some customers have the patience to wait, while others simply walk away at the sight of a long line and Every restaurant counter needs an employee for taking the order and processing the payment. Labor rates are increasing every now and then and it is difficult to find employees, hence to solve this problem we plan to design a "Food Order Management System."

**2. Project Description:**

The project has two main sections namely admin and customer section

From the admin section, the user can manage employee records by entering his/her name, phone number, address, department, date of birth, and joining date. The admin can also view the last orders of the customers.The user can simply order foods by selecting food items, entering a customer name, quantity and address in the customer section.

The system does not create an external file to store the user’s data permanently. This system is developed using C Programming Language and different variables, strings have been used for the development of it

**3. Requirement specification:**

* **Functional requirements**

1. Admin login

The admin section should have a login facility through which only authorised person can view and edit the system.

1. Add new records

Admin should have the ability to add new records. Admin should add various information such as employee id, employee name, date of birth, year of joining, place and department.

1. Add and delete food items in menu

Admin should be able to add a new food item to menu and delete a food item from menu using its Id.

1. Search, delete and view record

Admin should be able to search through the records using either name or employee id. Deletion of records can also be done using employee id. View of complete records should be provided to the admin.

1. View previous placed orders

The admin should be able to view all the previously placed orders for further usage.

6. Display the menu

In the system all items are displayed with their rates.

1. Modify menu

Admin can make changes in the menu like adding or removing food items which are not available.

1. Select food items

Items are selected by customers which they prefer..

1. Display order amount

After selecting items and their quantity, the total bill amount should be displayed.

10. Provide delivery details

Customer should be asked to enter delivery details. On successful entry of delivery details, order is placed.

* Non-Functional requirements

1. Portability:

System running on one platform can be easily converted to run on another platform

1. Reliability

The ability of the system to behave consistently in a ser-acceptable manner when operating within the environment for which the system was intended.

1. Availability

The system should be available at all times, meaning the user can access it all the time, only restricted by the down time of the server on which the system runs.

1. Security

Secure access of confidential data (customer information)

1. User friendly

System should be easily used by the customers.

1. Performance

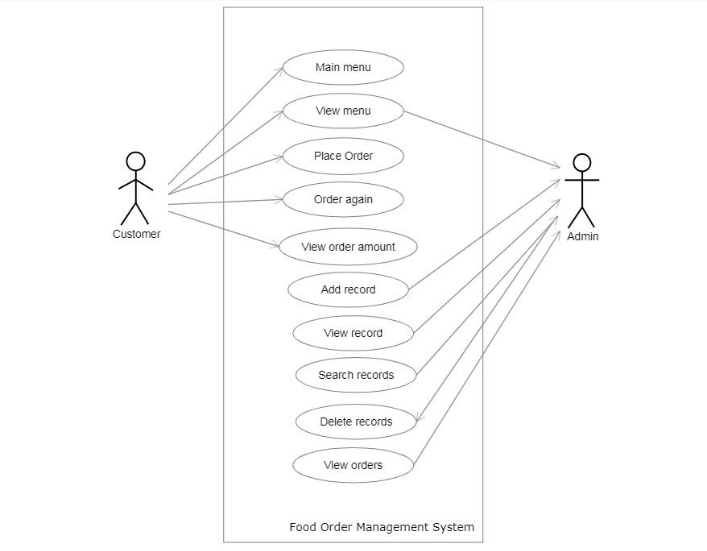
Performance should be fast.

1. Privacy

Personal data of the system should not disclose to anyone.

* Software requirements
* Code::Blocks v17.12
* GNU C compiler
* Operating system
* Windows OS
* Language
* C language

**Use-case diagram**

****

Use-Case diagram

**4. Design:**

The main objective of the Food Order Management system is to ease the food ordering process.

The proposed system increases ease of account maintenance, menu updating and billing process.

The system mainly has two sections:

1. Admin section
2. Customer section.

The admin has a login facility through which only an authorised person can view the admin section. The admin can add new employee details, delete an existing employee details, search for employee details using either name or employee ID. Admin can also view all employee records and view previously placed orders.

The customer can choose a food item from menu. Id,food name and  their prices are displayed. Customer can choose an item using its id and its quantity. Once customer enters delivery details, the order will be placed.

**Main Features:**

1. Order food
2. Add employee records
3. Delete employee records
4. Add food item to menu
5. Delete food item from menu
6. View last orders
7. View all employee records
8. Search employee by ID or Name
9. Remove Employee records

**5. Test cases:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **TEST SCENARIO** | **TEST STEP** | **EXPECTED RESULT** | **ACTUAL OUTPUT** |
| **1** | **Verify that correct option is selected from the menu** | **Selecting any option from the menu.** | **Go to that particular option.** | **The option selected gets displayed.** |
|  | **Verify if different option is selected apart from the menu options** | **Selecting NON menu option.** | **Should give an error** | **Displays saying option doesn’t exist and returns to the menu.** |

#### TABLE 1: INTERACTIVE FOOD MENU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **TEST SCENARIO** | **TEST STEP** | **EXPECTED RESULT** | **ACTUAL OUTPUT** |
| **1** | **Password: Verify if the given password is correct** | **Entering the password.** | **Accepts the password and grants access to admin section** | **Successful login to admin section** |
|  | **Verify if the given password is wrong** | **Entering the password** | **Accepts the password and shows incorrect password** | **Unsuccessful login. Returns back to main menu** |

#### TABLE 2: ADMIN LOGIN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **TEST SCENARIO** | **TEST STEP** | **EXPECTED RESULT** | **ACTUAL OUTPUT** |
| **1** | **Employee name: Verify if the given name is available** | **Entering the name.** | **Accepts the name and gives the output** | **Displays the particular information of the given name.** |
|  | **Verify if the given name is NOT available** | **Entering the name** | **Should return that it doesn’t exist.** | **Displays that it doesn’t exist and asks to re-enter the correct name.** |
| **2** | **Employee ID: Verify if the given ID is available** | **Entering the ID.** | **Accepts the ID and gives the output** | **Displays the particular information of the given ID.** |
|  | **Verify if the given ID is NOT available** | **Entering the ID** | **Should return that it doesn’t exist.** | **Displays that it doesn’t exist and asks to re-enter the correct ID.** |

#### TABLE 3: SEARCH EMPLOYEE RECORD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **TEST SCENARIO** | **TEST STEP** | **EXPECTED RESULT** | **ACTUAL OUTPUT** |
| **1** | **Employee ID: Verify if the given ID is available** | **Entering the ID.** | **Accepts the ID and deletes the record** | **Displays the particular information of the given ID and deletes the record.** |
|  | **Verify if the given ID is NOT available** | **Entering the ID** | **Should return that it doesn’t exist.** | **Displays that it doesn’t exist and asks to re-enter the correct ID.** |

#### TABLE 4: DELETE EMPLOYEE RECORD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **TEST SCENARIO** | **TEST STEP** | **EXPECTED RESULT** | **ACTUAL OUTPUT** |
| **1** | **Food ID: Verify if the given ID is available** | **Entering the ID.** | **Accepts the ID and asks to re-enter  valid ID.** | **Displays that the food record already exists and asks to re-enter valid ID** |
|  | **Verify if the given ID is NOT available** | **Entering the ID** | **Accepts the ID and adds food item to menu** | **Asks to enter the food name and its price. Adds food details to menu** |

#### TABLE 5:  ADD FOOD ITEM TO MENU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **TEST SCENARIO** | **TEST STEP** | **EXPECTED RESULT** | **ACTUAL OUTPUT** |
| **1** | **Food ID: Verify if the given ID is available** | **Entering the ID.** | **Accepts the ID and deletes the record** | **Deletes the record.** |
|  | **Verify if the given ID is NOT available** | **Entering the ID** | **Should return that it doesn’t exist.** | **Displays that it doesn’t exist and asks to re-enter the correct ID.** |

#### TABLE 6: DELETE FOOD ITEM FROM MENU

# 5. Expected outcome:

The system should give proper outcome as per the design. It should handle all the errors. It should pass all the test cases mentioned above. A proper system that displays the food menu to customers and

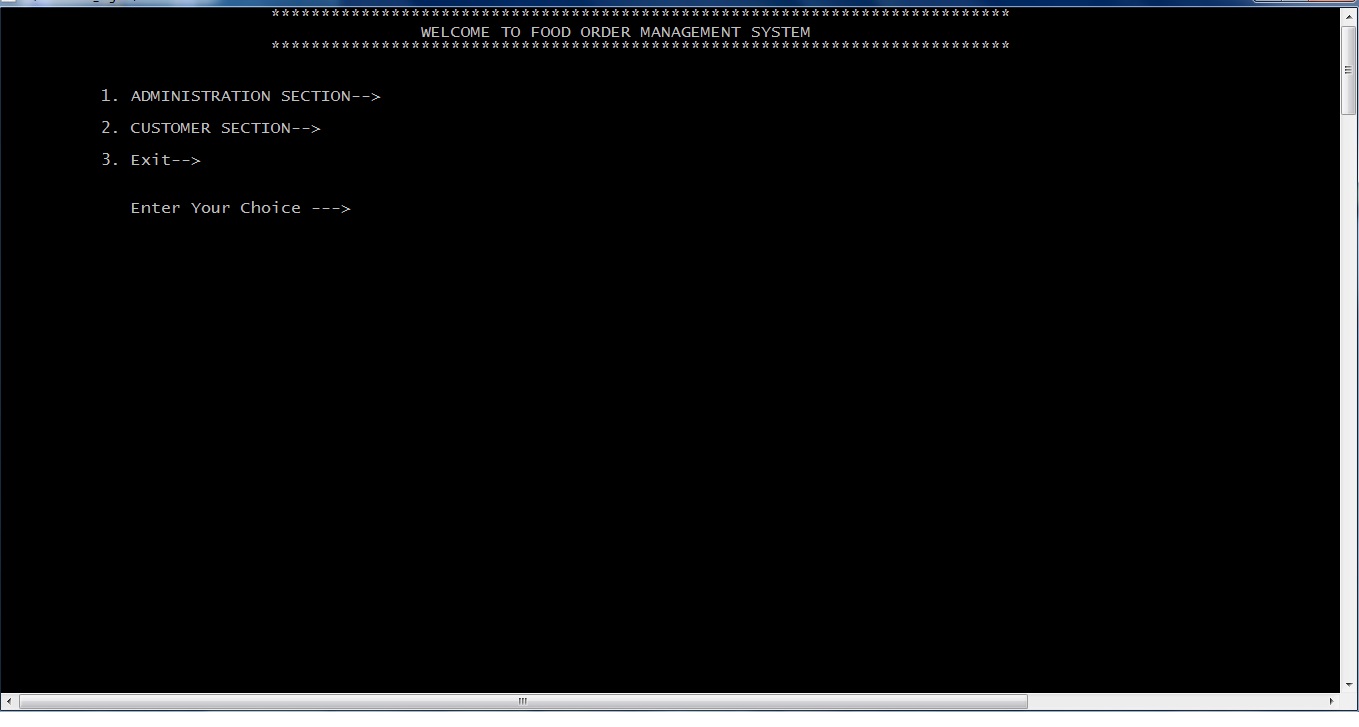


Fig 2: Main menu

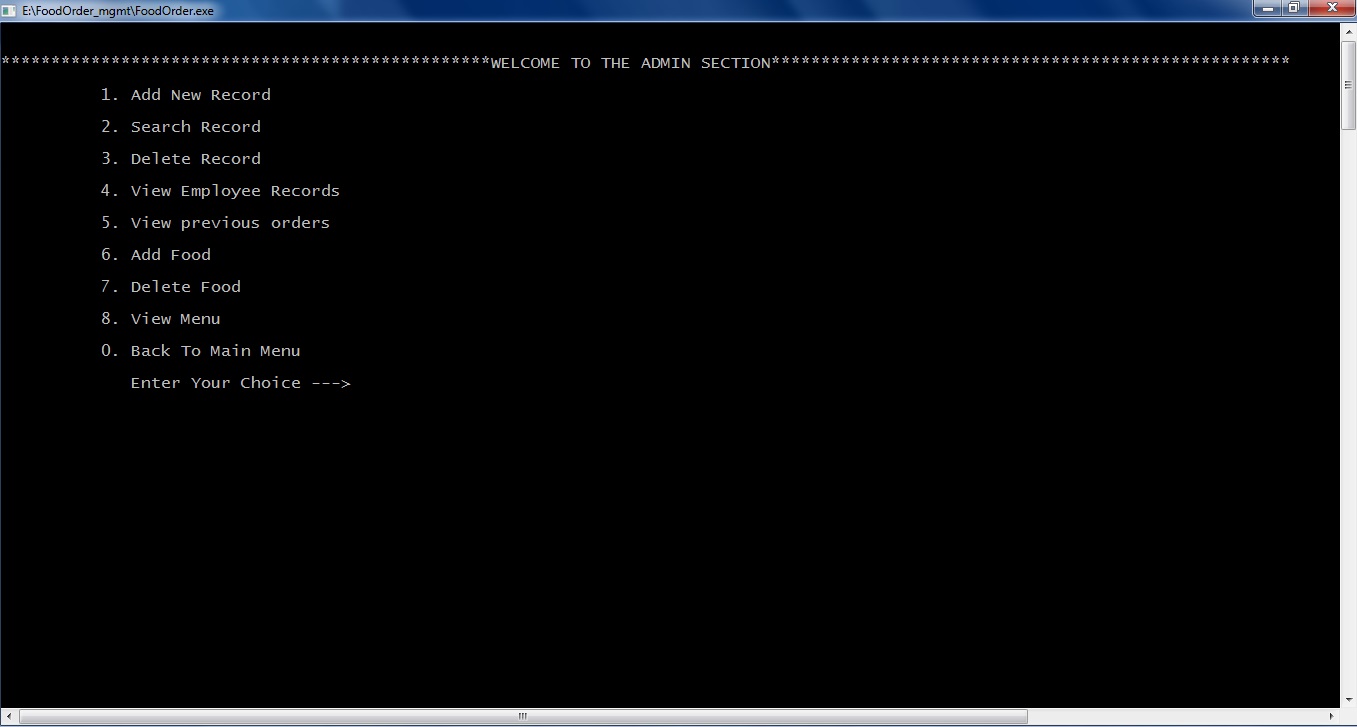


Fig 3. Admin section

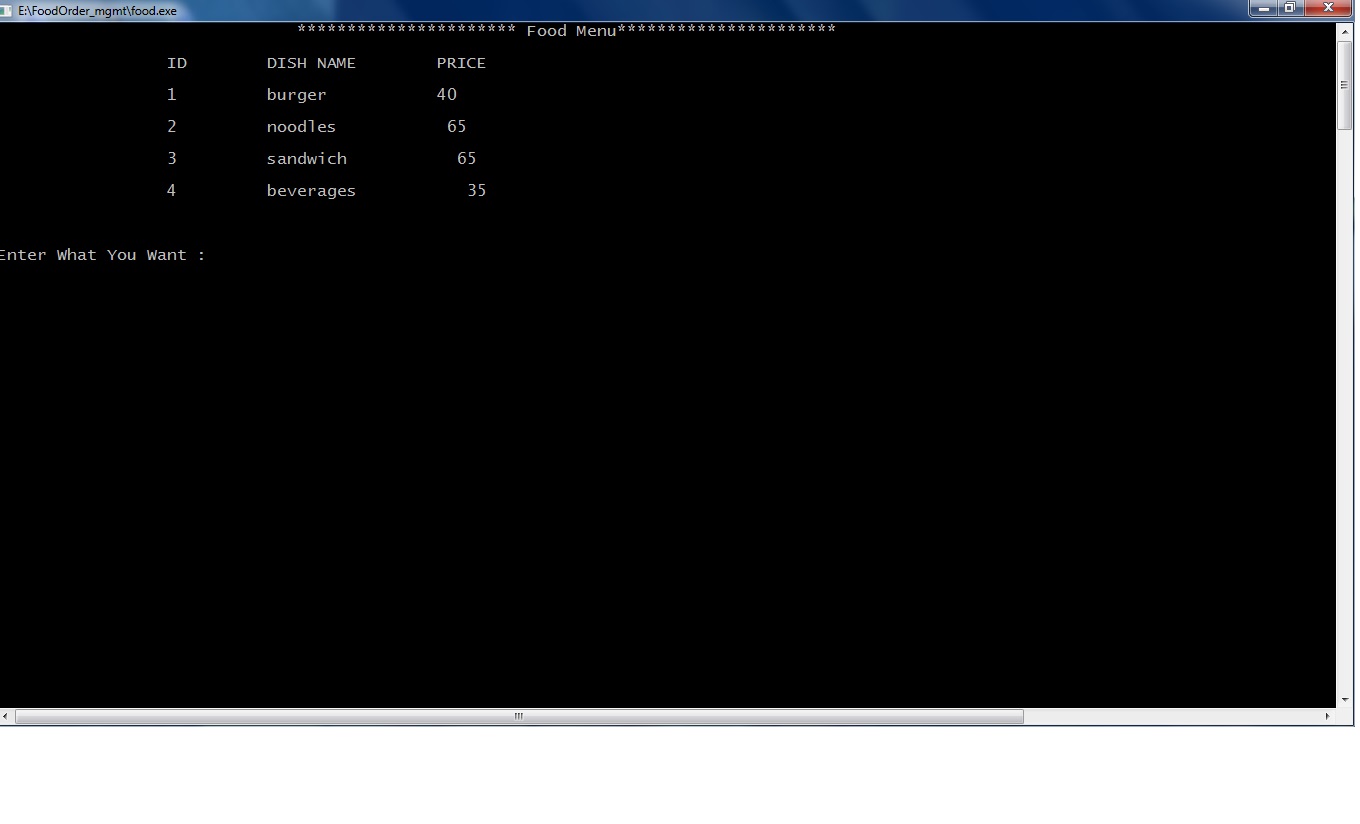


Fig 4. Customer section

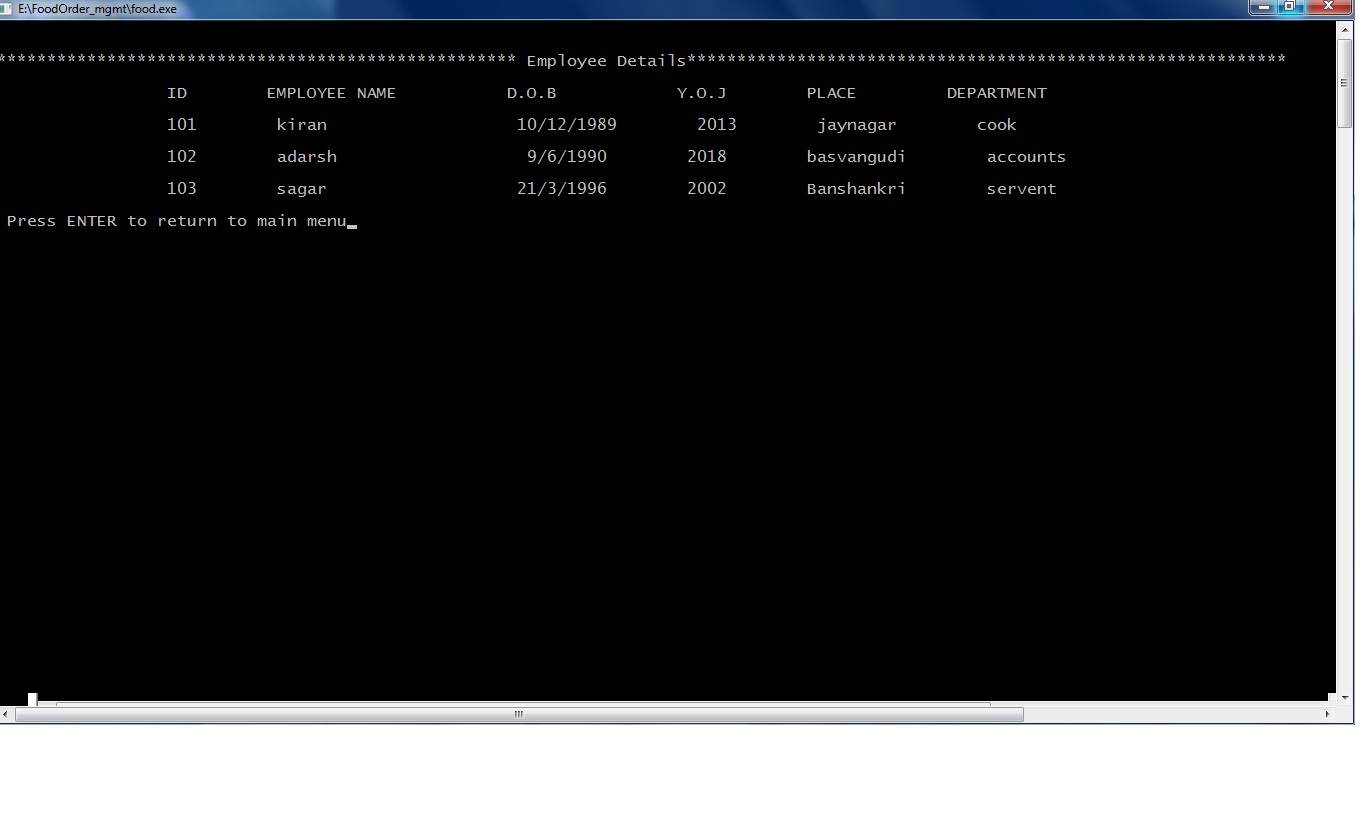


Fig 5. Employee Records

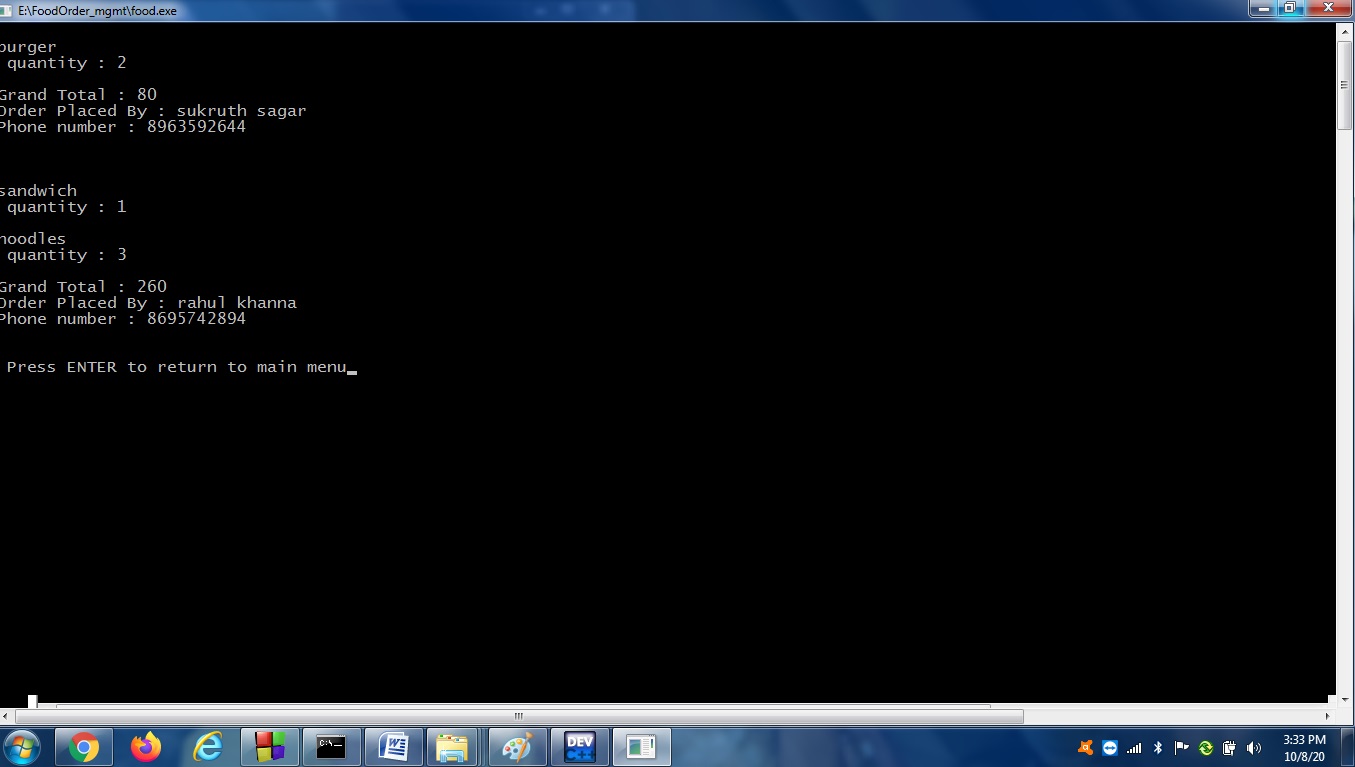


Fig 6. Orders placed