**Online Restaurant Management System**

A project submitted to

**UKA TARSADIA UNIVERSITY**

in partial fulfillment of the requirements for the degree of

**Bachelor of Science**

in

**Information Technology**

for

**5 Years Integrated M.Sc.(IT)**

By

**Hasti Ghelani**

(202006100110065)

and

**Vatsal Mangukiya**

(202006100110076)

Guided by

**Mr. Bhavik Sarang**

**Assistant Professor**



**Babu Madhav Institute of Information Technology**

**Uka Tarsadia University**

**Bardoli – 394350**

**December 2020**

**CERTIFICATE**

****This is to certify that **Hasti Ghelani(202006100110065) & Vatsal Mangukiya(202006100110076)**has submitted project entitled **“Project Title”** as the partial fulfillment for the award of the degree of Bachelor of Science in Information Technology for 5 Years Integrated M.Sc.(IT) in 2020 – 2021.

**Date:**

**Place:**

**Mr. Bhavik Sarang**  **Dr. Jitendra Nasriwala**

Guide Programmer Coordinator,

**External Examiner**



**Babu Madhav Institute of Information Technology,**

**Uka Tarsadia University,**

**Bardoli – 394350**

**Table of Content**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Chapters** | **Particulars** | | | **Page no.** |
| **1** | **Introduction** | | |  |
| 1.1 | Problem Definition | |  |
| 1.2 | Project Objective | |  |
| 1.3 | Product Scope | |  |
|  | | | | |
| **2** | **Overall Description** | | |  |
| 2.1 | Product Perspective/ Environment Description | |  |
|  | 2.1.1 | Hardware Interface/ Hardware Specification |  |
|  | 2.1.2 | Software Interface/ Software Specification |  |
|  | | | | |
| **3** | **System Specific Requirements** | | |  |
| 3.1 | Functional Requirement | |  |
| 3.2 | Non-functional Requirement | |  |
|  | | | | |
| **4** | **System Design** | | |  |
| 4.1 | Database Design | |  |
| 4.1.1 | Database Schema |  |
| 4.1.2 | Data Dictionary |  |
|  | | | | |
| **5** | **System Implementation** | | |  |
| 5.1 | Screenshots | |  |
|  | | | | |
| **6** | **Testing** | | |  |
| 6.1 | Test Cases | |  |
|  | | | | |
| **9** | **Future Enhancement** | | |  |
|  |  | | |  |
|  | **Conclusion** | | |  |
|  |  | | |  |
|  | **Bibliography** | | |  |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Figure Description** | **Page no.** |
| **5.1.1** | **Use Case Diagram for Owner** |  |
| **5.1.2** |  |  |
| **5.1.3** |  |  |
| **5.2.1** | **Activity Diagram for Manage Product** |  |
| **5.2.2** |  |  |
| **5.2.3** |  |  |
| **5.2.4** |  |  |
| **5.2.5** |  |  |
| **5.3.1** | **Sequence Diagram for “adding a product”** |  |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table no.** | **Table Description** | **Page no.** |
|  | Data Dictionary Tables | 50 - 60 |
| 8.1.1 | Test Cases for Login |  |
| 8.1.2 | Test Cases for Product |  |

**Chapter 1**

**Introduction**

* 1. **Problem Definition**
  2. **Initial Requirement Document**

|  |  |
| --- | --- |
| Title of the project | Online Restaurant Management System |
| Stakeholders involved in capturing requirements | Restaurant Owner, Restaurant Staff and Developer |
| Techniques used for requirement capturing | Interviewing, Brainstorming |
| Name of the person along with designation | - |
| Date | July, 2022 |
| Users of the system | Owner, Branch Manager, Cashier, Order Man, Customer |
| Version | 2.0 |
| Consolidated list of initial requirements: | |
| 1. The owner has to login first to use the system. 2. The owner shall be able to manage branches. 3. The owner shall be able to manage menu. 4. The owner shall be able to manage gallery. 5. The owner shall be able to manage staff branch wise. 6. The owner/branch manager shall be able to manage events orders. 7. The owner shall be able to manage tables. 8. The branch manager/order man shall be able to manage the status of the tables (like Occupied/ Unoccupied/ Reserved). 9. The branch manager/order man shall be able to manage customer, when the customer physically visits and order for the item(s) from the restaurant. 10. The branch manager/order man shall be able to manage customer order. 11. The customer shall be able to view the details about restaurant like its profile, location, etc. 12. The customer shall be able to view and search (filter) the item category wise. 13. The customer shall be able to manage cart by adding the item(s) in the cart, by updating the quantity or by removing the item. 14. The customer shall be able to register and login into the system; and make payment before the order confirmation. 15. The customer shall be able to make payment using one of the online payment gateways for the online and offline orders or through cash on delivery only for offline. 16. The customer shall be able to manage pick-up timings of the order. 17. The system shall be able to generate bill. 18. The cashier shall be able to update the status of order, while received the payment of the order make by the customers. 19. The customer shall be able to view order history. 20. The customer shall be able to make cancellation of order before delivery, within 5 hours. 21. The system shall be able provide the availability of a particular item to the customer. 22. The customer shall be able to give feedback for the order. 23. The owner shall be able to view all the feedbacks given by the customers. 24. The system users shall be able to update their profile including their password. 25. The owner/branch manager shall be able to manage expenses. 26. The system should be able to generate reports like:     1. Details of all the orders:        1. Table wise        2. Item Order Category wise        3. Event wise     2. Details of all the customers     3. Status of all the event orders (Day/Date, Weekly, Monthly, Yearly)     4. Purchases and Sales Expenses | |

* 1. **Project Objective**

The restaurant management system facing issues in traditional methods so the main objective of system is provided automation to the business. The system will manage finance of restaurant, which will bring transparency in restaurant. System will reduce the tedious work of employees and decrease manual work and errors. System will make task easier for the both side of user.

* 1. **Product Scope**

Restaurant web application will be available 24\*7.

* + This system will be implemented for one organization.
  + Waiter, Cashier, Admin have their individual login with different functionalities.
  + Visitor can only view menu online.

**Chapter 2**

**Overall Description**

**2.1 Product perspective/Environment Description**

***Hardware Requirements***

* Processor : Intel(R) Core(TM) i3 or more
* RAM: 512MB or more
* Hard Disk: 50MB or more
* Input Device: PC/Laptop/Mobile/Tablet
* Output Device: Printer/PC/Screen
* Communication Interface: LAN cable for printer

***Software Requirements***

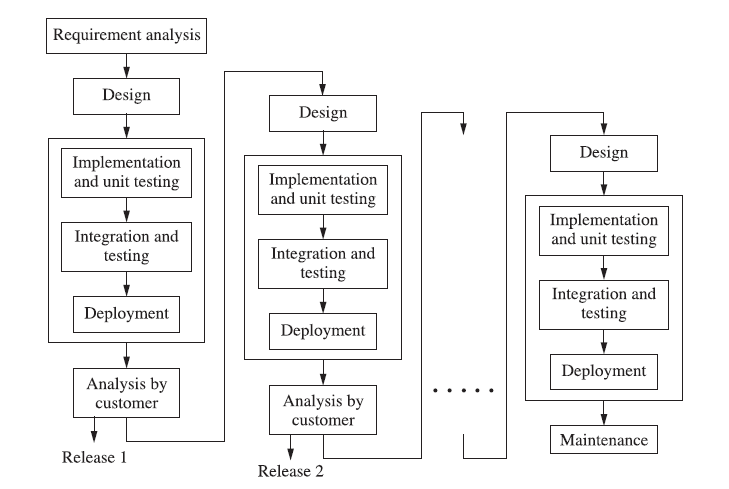
* Operating System : Microsoft Windows Platform, Linux Platforms, MAC OS
* Front-End : Html, CSS, JavaScript, Bootstrap
* Back-End : PHP, MySQL Server
* Tools Used : XAMPP(Apache), VS Code

**Chapter 3**

**System Planning**

**3.1 Software Engineering Model:**

* Iterative Enhancement Model is used to develop this project.

****

**Chapter 4**

**System Specific Requirement**

**4.1 Functional Requirement:**

Functional requirements define the fundamental action that system must perform.

***Module 1: Manage User Login***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR 1** | Owner, Cashier, Branch Manager, order man and Customer will be authenticated with their Username and password to login into the system. | Login page |
| **FR 2** | Enable a registered user to reset password. And also, if user forgets password, then OTP is sent through mail for user’s verification. | Login Page  Change Password Page  Forgot Password page |

***Module 2: Manage staff***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR 1** | This module will be managed by Owner and Branch Manager.  The Owner and Branch Manager will be able to perform following operation like:   * Add new staff * Update an existing staff information * View/Search staff details   The system will assign a unique id (staff id) to each staff. | Staff\_View\_Update\_Search Page |

***Module 3: Manage Items***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR 1** | This module will be managed by Owner and Branch Manager.  The Owner and branch manager will be able to manage many other modules like:   * Manage Item description * Manage Item Images * Manage Item price | Item\_Add\_View\_Update\_Search Page |
| **FR 2** | The system will assign a unique id (Item id) to each Item.  Whenever any new product arrives, the Owner and branch manager will be allow to record following details of the product like . . .  Item name with its Category, Images, Price and Description etc. | Item\_Add\_View\_Update\_Search Page |
| **FR 3** | **Manage Item:**  To manage this module, the owner and branch manager will be able to perform following operations like:   * Add new item * Update an existing item * Active/De-active item   The Owner and customer, manager will be able to:   * View/Search Items | Item\_Add\_View\_Update\_Search Page |
| **FR 4** | The owner and branch manager will be able to:   * View/Search item (by Category wise, Item\_id wise, Name wise) | Item\_Add\_View\_Update\_Search\_Active\_Deactive Page |
| **FR 5** | **Manage Category:**  In Manage Category module, the owner will be able to perform following operations like:   * Add new category(Branch wise) * Update an existing category * Active/De-active   The Owner customer and manager will be able to:   * View/Search categories   The system will assign a unique id (Category id) to each category. | catagory\_Add\_View\_Update\_Search Page |
| **FR 6** | **Manage Description:**  In manage item description module the Owner and branch manager will be able to perform following operations like:   * Add description * Update Description | Item\_Add\_View\_Update\_Search Page |
| **FR 7** | **Manage Image:**  In manage Image module Owner and branch manager will be able to perform following operations like:   * Add Item Image * Update Item Image | Item\_Add\_View\_Update\_Search Page |
| **FR 8** | **Manage Price:**  In manage price module Owner and branch manager will be able to perform following operations like:   * Add Item price * Update Item price | Item\_Add\_View\_Update\_Search Page |
| **FR 9** | Customer and Order man will be able to view items menu from their login. | Customer\_menu page  Order\_man menu page |

***Module 4: Manage Branches***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR 1** | This module will be managed by Owner.  The Owner will be able to manage many other modules like:   * Manage Branch Details * Manage Branch Manager | Item\_Add\_View\_Update  \_Search Page |
| **FR 2** | **Manage Branch Details:**  In manage branch module the Owner will be able to manage other modules like:   * Add branch address | Item\_Add\_View\_  Search Page |
| **FR 3** | **Manage Branch manager:**  The Owner will be able to manage managers of different branches.   * Add branch manager * Manage the details of branch manager. * View/Search manager information | Branch manager\_Add\_View\_Update  \_Search Page |
| **FR 4** | Customer will be able to view every branch’s address. |  |

***Module 5: Manage Tables***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR 1** | This module will be managed by branch manager and order man.  The branch manager will be able to manage Table Details. | Table\_add\_view\_update page |
| **FR 2** | **Manage Table Details:**  The system will assign a unique id (table id) to each Table.  To manage this module, the Owner and branch manager will be able to perform following operations like:   * Add new Table * Update Table * Active/De-active Table | Table\_add\_view\_update page |
| **FR 3** | The order man will be able to manage table order status like occupied/unoccupied  The branch manager and cashier will be able to view table order status. | Table\_order page |

***Module 6: Manage Customer***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR1** | This module will be managed by Branch Manager, Order man, Cashier and Customer.  The Branch manager and Order man will be able to manage customer, when the customer physically visits and order the items.  The Order man, Cashier will be able to perform following operations like:   * Add new customer ( order )   The customer shall also be able to register himself and edit his/her profile.  The system will assign a unique id (customer id) to each customer. | Customer\_Registration(Add)\_ Page  Customer\_ View\_Update\_Search\_Active\_De-active Page |
| **FR2** | The owner will be able to:   * View/Search customer details | Customer\_ View\_Update\_Search\_Active\_De-active Page |
| **FR3** | The customer will be able to view the details about restaurant like its profile, location, etc. | AboutUs Page |

***Module 7: Manage Cart***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR1** | This module will be managed by Customer.  The customer will be able to perform following operations like:   * Add to cart * Update cart items * Removed cart items * View Cart history | Add\_To\_Cart Page |

***Module 8: Manage Customer Order***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR1** | This module will be managed by Branch manager, Cashier and Customer.  When the customer physically visits the restaurant and if he/she wants to order the items(s), then after registering the customer, the Orderman will be able to perform following operations like:   * Add customer’s order * Update customer’s order * Cancel (Delete) customer’s order   The system will assign a unique id (order id) to each customer’s order. | Customer\_Order\_Add Page  Customer\_Order\_ View\_Update\_Search\_  Delete Page |
| **FR2** | The Branch Manager will be able to:   * View/Search customer order(s) | Customer\_Order\_ View\_Update\_Search\_  Delete Page |
| **FR3** | After adding the product into cart, if the customer confirms his/her order, then after login into the system the customer will be able to perform the following operations like:   * Confirm order * Cancel (Delete) order * View order | Customer\_Order\_Add Page  Customer\_Order\_ View\_Update\_Search\_  Delete Page |
| **FR4** | The customer will be able to view order history. | Customer\_ Order\_History Page |
| **FR5** | The customer will be able to cancel the order before delivery, within limited time period. | Customer\_Order\_  Cancel Page |

***Module 9: Manage Event***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR1** | This module will be managed by Branch manager and Owner.  When the customer gives event order the Owner and Branch Manager following operations like:   * Add Event * Update Event * Cancel Event   The system will assign a unique id (event id) to each customer’s order. | Owner\_event page |
| **FR2** | The Branch Manager and Owner will be able to view event details like:   * Event Date * Event Time | Owner\_Ecent page  Branch Manager\_event pae  Cashier\_Dashboard |

***Module 10: Generate Bill***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR1** | After the confirmation of order placed by the customer the system will be able to generate bill in “.pdf” with all the necessary details of the Customer. | Customer\_Order\_Add Page  “.pdf” Format  (Open-source plugin) |
| **FR2** | The system will also allow the Restaurant(owner) to print the bill with all the necessary details of the Customer. | Customer\_Order\_Add Page  “.pdf” Format  (Open-source plugin) |
| **FR3** | The system will assign a unique id (bill no) to each customer’s order bill. | Customer\_Order\_Add Page  “.pdf” Format  (Open-source plugin) |

***Module 11: Manage Payment***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR1** | After the generation of bill, the customer shall be able to make payment using one of the online payment gateways (compulsory in online order ) or through cash on delivery.  The system will assign a unique id (payment id) to each customer’s order bill. | Payment Page |

***Module 12: Manage Gallery***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR1** | This module will manage by Owner.  The owner will be able to manage following operation category wise, like:   * Insert Image * Remove Image | Admin\_Gallery Page |

***Module 13: Manage Expenses***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR1** | This module will be managed by Owner and Branch Manager.  The owner and Manager will be able to do operations like:   * Insert Expense * Update Expenses * Delete Expenses | Expenses Page |
| **FR2** | The owner and branch manager will be able to see information like:   * Expense Value * Expense Type |  |

***Module 14: Manage Feedback***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR1** | The customer will also be able to give feedback for the Restaurant. The system will assign a unique id (feedback id) to each customer’s feedback. | Feedback Page |
| **FR2** | The feedback which is given by the customer and also be viewed by the Owner. | Feedback\_View Page |

***Module 15:Generate Report***

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **FR1** | The system should be able to generate reports like:   * + Details of all the orders:     1. Table wise     2. Item Order Category wise     3. Event wise   + Details of all the customers   + Status of all the event orders (Day/Date, Weekly, Monthly, Yearly)   + Purchases and Sales Expenses | Generate\_Report Page |

**4.2 Non-Functional Requirement:**

Non-functional requirements are often called qualities of a system.

|  |  |  |
| --- | --- | --- |
| **RN** | **Description** | **Comments** |
| **NFR1** | The application will be user-friendly and easy to operate, the functions will be easily understandable. | Usability |
| **NFR2** | The application will be Password protected. The Users will have to enter correct username and password to access the system. | Security |
| **NFR3** | The system will be designed in a maintainable manner. It will be easy to add new requirements in the any individual modules. | Maintainability |
| **NFR4** | The application will be simple in operations. | Simplicity |

**Chapter 5**

**System Analysis**

**5.1 Use case Diagram:**

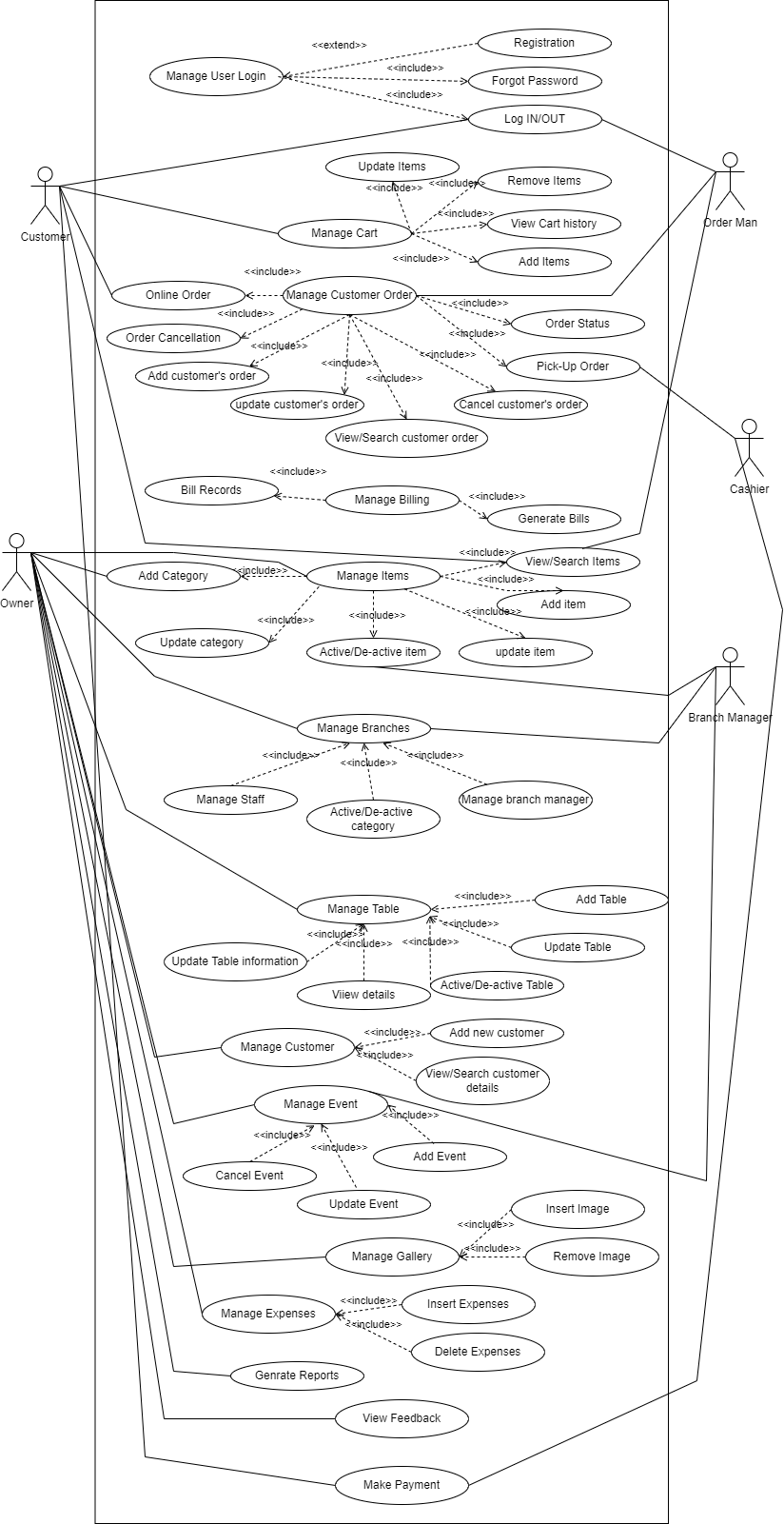


Fig 5.1.1 Use Case Diagram for Online Boutique Management System

**Chapter 6**

**System Design**

* 1. **Database Design:**
     1. **Database Schema:**

**1) Manage User Login:**

* tblUser(Userid[PK], username(email), password)

FD: Userid-> {username(email),password}

**2) Manage Branch:**

* tblBranch(Branchid[PK], address)

FD: Branchid-> {address}

* TblBranchStaff(SerialNo[PK],Staffid[FK],BranchId[FK])

FD: SerialNo-> {Staffid[FK],Branchid[FK]}

**3) Manage Staff:**

* tblStaff(staffid[PK], fname, lname, contactNo, Email, Address, DateOfJoin, designation[Branch Manager, Cashier, Order Man], salary, Branchid[FK][tblBranch]))

FD: Staffid-> {fname, lname, contactNo, Email, Address, DateOfJoin, designation, salary, Branchid[FK]}

**4) Manage Item Category:**

* tblCategory(Categoryid[PK], name,Branchid[FK][tblBranch])

FD: Categoryid-> {name,Branchid[FK]}

* tblItem(Itemid[PK], name, Catid[FK][tblCategory], price, description,imageid[FK])

FD: Itemid-> {name, Catid[FK], price, description,imageid}

* tblItemImage(Imageid[PK], imagepath, Itemid[FK])

FD: Imageid-> {imagepath}

**5) Manage Table:**

* tblTable(Tableid[PK])
* tblCustomerTable(SerialID[PK],Tableid[FK][Tableid], customerName, contactNumber)

FD: Tableid-> { customerName, contactNumber }

**6) Manage Customer:**

* tblCustomerOffline(CustomeroffID[PK], customerName, contactNumber)

FD: CustomeroffID-> { customerName, contactNumber }

* tblCustomerPassword(Customerid[PK][tblCustomerOnline],Cpassword)

FD: Customerid-> { Cpassword}

* tblCustomerDetail(Customerid[PK],FirstName,LastName,contactNumber,Address,pincode,emailid)

FD:Customerid->{FirstName,LastName,contactNumber,Address,pincode,

emailid}

**7) Manage Cart:**

* tblCart(cartId[PK],customerId[FK][tblCustomerDetail])  
   FD:cartID->{customerId[FK]}
* tblCartDetail(cartDetailId[PK], cartId[FK][tblCart], itemId[FK][tblItem], Quantity)  
   FD:cartDetailId-> {cartId[FK][tblCart], itemId[FK][tblItem]}

**8) Manage Order:**

* tblOrderOffline(Orderid[PK], serialId[FK][tblCustomerOffline],itemID[FK][iblItem], Quantity, Datetime)  
   FD:Cartid->{serialId[FK],itemId[FK],quantity, DateTime}
* tblOrderOnline(Orderid[PK], customerID[FK][tblCustomerOnline], Cartid[FK][tblCart], DateTime)  
   FD:Cartid->{Cartid[FK], Itemid, quantity}

**9) Manage Event:**

* tblEvent (Eventid[PK], Eventype, EventDateTime, Userid[FK][tblUser])

FD: Eventid-> { Eventype, Eventtime, EventDateTime, uid}

**10) Genrate Bill:**

tblOfflineBill(BillId[PK], OrderId[FK][tblOrderOffline], billDateTime, totalAmount)  
 FD: BillId-> { Orderid[FK], billDateTime, totalAmount}  
  
 tblOnlineBill(BillId[PK], OrderId[FK][tblOrderOnline], billDateTime, Itemid, name, price, quantity, totalAmout)  
 FD: BillId-> {Orderid[FK], Itemid, name, price, quantity, totalAmout}

**11) Manage Payment:**

* tblPaymentOffline(offPaymentid[PK], Customerid[FK][tblCustomerOffline],Orderid[Fk][tblOrderOffline], paymentMode)  
   FD:offPaymentId->{Customerid[FK][tblCustomerOnline],Orderid[Fk][tblOrder], paymentMode}  
    
   tblPaymentOnline(onPaymentId[PK], Customerid[FK][tblCustomerOnline],Orderid[Fk][tblOrderOnline], paymentAmount,paymentStatus)  
   FD: onPaymentId-> {Customerid[FK],Orderid[Fk],paymentAmount, paymentStatus}

**12) Manage Gallery:**

* tblImageCategory(ImageCatid[Pk],imageCategory)

FD: ImageCatid -> { ImageCategory}

* tblImageGallery(Imageid[Pk],Imagepath,tblImageCategory[PK][tblImage

Category])

FD: Imageid-> { Customerid, tblImageCategory}

**13) Manage Expenses:**

* tblExpense(expanceType,expanceValue)

**14) Manage Feedback:**

* tblFeedback(Feedbackid[PK], Customerid[FK][tblCustomerOnline], )

FD: Feedbackid -> { Customerid}

**6.1.2 Data Dictionary:**

**1) Manage User login:**

* tblUser :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Userid | int | - | Primary Key | Identifies Unique id of User. |
| 2 | username | Varchar | 50 | Not Null, Unique | Specifies Unique username (emailid) of User. |
| 3 | password | Varchar | 15 | Not Null | Password of User. |

**2) Manage Staff:**

* tblDesignation :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Desid | int | - | Primary Key, Auto increment | Identifies Unique id of Designation Category. |
| 2 | Designationtype | Varchar | 15 | Not null, Unique | Specifies Unique  Designation category type. |

* tblStaff :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | staffid | int | - | Primary Key,  Auto increment | Identifies Unique id of Product Size. |
| 2 | name | Varchar | 50 | Not Null, Unique | Specifies the size of Product like S, M, L, XL, XXL. |
| 3 | Desid | int | - | Foreign Key | Referenced form  tblDesignation(Desid) |
| 4 | salary | float | - | Not Null, Unique | Specifies the salary of staff |
| 5 | Branchid | int | - | Foreign Key | Referenced from tblBranch(tblBranch) |

**3) Manage item:**

* tblCategory :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Categoryid | int | - | Primary Key, Auto increment | Identifies Unique id of item category. |
| 2 | name | Varchar | 50 | Not null, Unique | Specifies name of category. |
| 3 | Branchid | Int | - | Foreign Key | Referenced from tblBranch(Branchid) |

* tblItem :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Itemid | int | - | Primary Key,  Auto increment | Identifies Unique id of item. |
| 2 | name | Varchar | 15 | Not Null, Unique | Specifies name of item |
| 3 | Catid | Int | - | Foreign Key | Referenced from tblCategory(Catid) |
| 4 | price | int | - | Not null | Specifies price of item |
| 5 | description | Varchar | 50 | - | Specified description of an item |

* tblItemImage :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Imageid | int | - | Primary Key,  Auto increment | Identifies Unique id of Item Image. |
| 2 | imagepath | Varchar |  | Not Null, Unique | Specified path of image. |
| 3 | Itemid | Int | - | Foreign Key | Referenced from tblItem(Itemid) |

**4) Manage Branch:**

* tblBranch :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Branchid | int | - | Primary Key, Auto increment | Identifies Unique id of Branch. |
| 2 | addrass | Varchar | 50 | Not null, Unique | Specifies address of specific branch. |
| 3 | Staffid | Int | - | Foreign Key | Referenced from tblStaff(staffid) |

**5) Manage Table:**

* tblTable :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Tableid | varchar | 10 | Primary Key | Identifies Unique id of Table. |

* tblCustomerTable :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Itemid | int | - | Primary Key,  Auto increment | Identifies Unique id of item. |
| 2 | Tableid | Varchar | 10 | Foreign Key | Referenced from tblTable(Tableid) |
| 3 | Customer  Name | Varchar | 15 | Not Null, Unique | Specifies name of offline customer |
| 4 | Contact  Number | numeric | 10 | Not Null, Unique | Specifies offline Customer contact number |
| 5 | Orderid | int | - | Foreign Key | Refrrenced from table tblOrder(Orderid) |

**6) Manage Customer:**

* tblCustomerOffline:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Customerid | int | - | Primary Key, Auto increment | Identifies Unique id of offline customer. |
| 2 | customerName | Varchar | 50 | Not null | Specifies customer name. |
| 3 | contactNumber | numeric | 10 | Not null | Specifies customer  Contact number. |
| 4 | userId | int | - | Foreign Key | Referenced from tblUser(Userid) |

* tblCustomerPassword:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Customerid | int | - | Primary Key  [FK] | Identifies Unique id of user.  Referenced from tbl(Customerid) |
| 2 | password | Varchar | 20 | Not null | Password of customer |

* tblCustomerOnline :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Customerid | int | - | Primary Key,  Auto increment | Identifies Unique id of online customer. |
| 2 | customerName | Varchar | 50 | Not null | Specifies customer name. |
| 3 | contactNumber | numeric | 10 | Not null | Specifies customer  Contact number. |
| 4 | userId | int | - | Foreign Key | Referenced from tblUser(Userid) |

**7) Manage Cart:**

* tblCart:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Cartid | int | - | Primary Key, Auto increment | Identifies Unique id of offline customer. |
| 2 | Customerid | Varchar | 50 | Not null | Referenced from tblCustomerOnline  (Customerid) |
| 3 | Itemid | numeric | 10 | Foreign Key | Referenced from tblItem(Itemid). |
| 4 | quantity | int | - | Not null | Specifies Quantity of  item |
| 5 | amount | float | - | Not null | Specifies amount of an item |

**8) Manage Order:**

* tblOrderOffline:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Orderid | int | - | Primary Key, Auto increment | Identifies Unique id of offline order. |
| 2 | Itemid | int | - | Foreign Key | Referenced from tblItem  (Itemid) |
| 3 | quantity | int | - | Not null | Specifies order  Item quantity. |

* tblOrderOnline:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Orderid | int | - | Primary Key, Auto increment | Identifies Unique id of offline order. |
| 2 | Cartid | Int | - | Foreign Key | Referenced from tblCart(Cartid) |
| 3 | Itemid | int | - | Foreign Key | Referenced from tblItem  (Itemid) |
| 3 | quantity | int | - | Not null | Specifies order  Item quantity. |

**9) Manage Event:**

* tblEvent:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Eventid | int | - | Primary Key, Auto increment | Identifies Unique id of Event. |
| 2 | eventType | Varchar | 50 | Not null | Specifies type of an event. |
| 3 | eventDate | date | 10 | Not null | Specifies date of an event. |
| 4 | eventTime | time | - | Not null | Specifies time of an event. |
| 5 | Userid | float | - | Foreign Key | Referenced from tblUser(Userlid). |

**10) Manage Bill:**

* tblOfflineBill:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Billid | int | - | Primary Key, Auto increment | Identifies Unique id of Offline Order Bill. |
| 2 | OrderID | Int | - | Foreign Key | Referenced from tblOrderOffline  (Orderid) |
| 3 | Itemid | Int | - | Foreign Key | Referenced from tblItem(Itemid) |
| 4 | name | Varchar | 50 | Not null | Specified name of  item |
| 5 | price | float | - | Not null | Specified Price of item. |
| 6 | Quantity | int | - | Not null | Quantity of item. |
| 7 | Amount | Float | - | Not null | Specified amount of items(Derived Price\*Quantity) |

* tblOnlineBill:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Billid | int | - | Primary Key, Auto increment | Identifies Unique id of Online Order Bill. |
| 2 | OrderID | Int | - | Foreign Key | Referenced from tblOrderOnline  (Orderid) |
| 3 | Itemid | Int | - | Foreign Key | Referenced from tblItem(Itemid) |
| 4 | name | Varchar | 50 | Not null | Specified name of  item |
| 5 | price | float | - | Not null | Specified Price of item. |
| 6 | Quantity | int | - | Not null | Quantity of item. |
| 7 | Amount | Float | - | Not null | Specified amount of items(Derived Price\*Quantity) |

**11) Manage Payment:**

* tblPayment:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Paymentid | int | - | Primary Key, Auto increment | Identifies Unique id of bill payment. |
| 2 | Itemid | int | - | Foreign Key | Referenced from tblItem  (Itemid) |
| 3 | quantity | int | - | Not null | Specifies order  Item quantity. |

**12) Manage Gallery:**

* tblImageCategory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | ImageCatid | int | - | Primary Key, Auto increment | Identifies Unique id of image category. |
| 2 | Image  Category | int | - | Foreign Key | Specified Category name. |

* tblImagegallery:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Imageid | int | - | Primary Key, Auto increment | Identifies Unique id of image. |
| 2 | Imagepath | varchar | 100 | Not null | Specified image path |
| 3 | tblImageCategory | int | - | Foreign Key | Referenced from tblImageCategory  (tblImageCategory) |

**13) Manage Expanse:**

* tblImageCategory

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | expanceType | varchar | 50 | Not null | Specified Expanse type |
| 2 | expanceValue | int | - | Not null | Specified Expance value. |

**14) Manage Feedback:**

* tblFeedback

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| 1 | Feedbackid | int | 50 | Primary Key, Auto increment | Specified Expanse type |
| 2 | Userid | int | - | Foreign Key | Referenced from tblUser(Userid). |
| 3 | feedback | varchar | 200 | Not Null | Specified customer feedback |

**Chapter 7**

**System Implementation**

* 1. **Screenshots:**

**Chapter 8**

**Testing**

**8.1 Test Cases:**

**Chapter 9**

**Future Enhancement:**

The Project has a very vast scope in future. The project can be implemented on intranet in future. It use in automation System in future.

**Conclusion:**

Our System Provides Easy and user-friendly Software to manage **entry of the** Products, Stock **maintain,** and **easily manage the customer etc..,** It reduce all paperwork. It provides more reliability to organize work. Our system also provide easy interaction between Admin and Customer.

**Bibliography:**

* <https://www.freeprojectz.com/paid-projects/online-boutique-store-0>
* https://www.studentprojectguide.com/projectreport/boutique-management-system/