**RESTAURANT MANAGEMENT SYSTEM**

**OBJECTIVE:**

The main purpose is to improve the performance of the restaurant by eradicating the daily paperwork. With this system the tasks would be performed in less amount of time and more efficiently. An additional benefit of this software is that during the rush hours the load can be balanced effectively, and restaurants would perform better than usual. In addition to this, human error that occurs when performing tasks manually is also minimized and presence of queues in the system to assign tasks to chefs can reduce congestion in the kitchen. The system would also result in reduction of labor which would result in the reduction of expenses of the restaurant. Feedback module would help the restaurant check for how well they are performing, and monthly/yearly figures can be checked by the billing module to see the trends in sales and profits. These benefits can potentially result in generations of more revenue for the restaurant.

Diagram

Description automatically generated

SYSTEM PERSPECTIC DIAGRAM

**FEATURES**

**ADMIN LOGIN:**

The project has an admin login that controls all the online activities in the system. Admin can check and verify various member details and manage the inventory and other information related to menu and chefs in the system.

**CUSTOMER:**

Customers interact with the system directly in order to place order, modify order, get bill and give feedback. We do not store any information related to customers in our system. The process of order taking starts with customers placing orders and then the other series of events begin.

**HEAD CHEF/KITCHEN MANAGER:**

He can approve the cancellation of an order whenever a customer edits or removes a dish from his order. He can also assign a dish to a particular chef based on the specialty of the chef.

**CHEF:**

Chefs don’t interact with the system. They just must look at the dishes present in their queues and prepare the dishes accordingly. The chef’s name, address and specialty etc. are stored in the database.

**HALL MANAGER:**

Hall Managers will provide its input when he marks the bill as paid when customers pay for their order or get the bill printed. Moreover, he gets a notification whenever a particular order is complete, or some customer asks for help through the system. Hall manager can also see tables in the hall and their status i.e., empty or filled.

**PRODUCT FUNCTIONS:**

The system will:

* + - Allow Customers to scroll through the menu and select the dishes he/she wants.
    - Allow the Customers to cancel/edit the order any time before its prepared.
    - Allow Customers to provide feedback regarding the food and overall service of the restaurant.
    - Allow Customers to request for bill.
    - Allow Customers to ask for help through the system.
    - Assign Head Chef to assign the dishes in an order to chefs according to their specialties.
    - Show dish queues and their status for Chefs.
    - Allow admin to perform CRUD (create, retrieve, update and delete) operations on Staff Members, Menu Items and Inventory.
    - Allow Head Chef to mark orders complete.
    - Allow the Head Chef to approve cancellation of dish or order.
    - Allow the Hall Manager to mark the bill as paid.
    - Notify the Hall Manager when a particular order is complete.
    - Allow the Hall Manager to see/edit status of tables reserved and available and their capacities.

**OPERATING ENVIRINMENT:**

It is an android application running on a tablet and the tablets are present in a restaurant. Firstly, the manager would be present at the entrance and the system in his tab would show the tables that are empty/reserved. There would be a tab present at every table for customers which they will use to place orders. When an order is placed the server would notify the head chef/ kitchen manager who would be in the kitchen. The head chef would use his tab which also would have the system installed and would add the order to the appropriate queues of the chefs. The chefs would be present in the kitchen area and their interface would allow them to check the dishes they have to prepare. So, the system is running on various tablets but the operating environment and purpose of each is different for each user.

**FUNCTIONALITIES:**

**CUSTOMER INTERFACE:**

The customer interface contains three screens. All three screens have a consistent layout.

**Place Order:** In this screen, the system shows a list of cards (UI Elements) of dishes. Each dish will have an image, its price per serving.

**Timer and Edit/Cancel Order:** After confirming the order, the user will be shown a timer screen. In this screen customer will be shown “Edit Order” and “Cancel Order” buttons and a timer which shows the completion time of the order. There will also be a button to request for bill.

###### Feedback: In the feedback screen, at the top right corner a button for “Request Bill” will be shown. Beneath this button we will display a form which will have different multiple-choice questions and a submit feedback button.

###### HEAD CHEF INTERFACE:

###### In the head chef interface, the system shows all the current orders in detail i.e., all the dishes of a particular order. In each order, there is a button which will be used to mark that dish cooked. Moreover, when a customer wants to remove a dish from his order, the system will show the head chef a notification to approve the removal of the dish.

###### HALL MANAGER:

###### Hall manager will have a screen where he will get notification whenever an order is completed. The system will notify the hall manager about the order number and table number. The Hall Manager also has a screen where all orders are listed, and a status button to mark the order as paid. Moreover, he also has an interface screen to see and the status of tables in the restaurant as free/available.

**ADMIN INTERFACE:**

Admin is authorized to perform *CRUD* operations on Staff Members, Menu Items and Inventory Items. He’ll be having three different screens for Staff Members, Menu Items, and Inventory.

**USE-CASE DIAGRAM**

Diagram

Description automatically generated

**DATA DIRECTORY**

**Table name: Employee**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINTS** |
| **EmpId** | **int** | **PK, not null** |
| **EmpName** | **string** | **nvarchar(max), not null** |
| **EmpEmail** | **String [25]** | **nvarchar (25), not null** |
| **EmpPassword** | **String [12]** | **nvarchar (12), not null** |
| **EmpDesignation** | **string** | **nvarchar (max), not null** |
| **EmpSpeciality** | **string** | **nvarchar(max), not null** |
| **EmpGender** | **char** | **nvarchar (1), not null** |
| **EmpPhone** | **double** | **float, not null** |

**Table name: FOOD**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINTS** |
| **FoodId** | **int** | **PK, not null** |
| **FoodName** | **string** | **nvarchar(max), null** |
| **FoodType** | **double** | **nvarchar(max), null** |
| **FoodCost** | **string** | **float, not null** |
| **FoodCuisine** | **string** | **nvarchar(max), null** |
| **FoodStatus** | **bool** |  |
| **FoodImage** | **byte []** | **varbinary(max),null** |

**Table name: Assign Work**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINTS** |
| **AssignId** | **int** | **PK, not null** |
| **OrderId (FK)** | **int** | **FK, not null** |
| **EmpId (FK)** | **int** | **FK, not null** |
| **WorkStatus** | **bool** | **Bit, not null** |

**Table name: Bill**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINTS** |
| **Bill Id** | **int** | **PK, not null** |
| **HallTableId (FK)** | **int** | **FK, int, not null** |
| **UserNmae** | **string** | **nvarchar(max), null** |
| **UserEmail** | **string** | **nvarchar(max), null** |
| **BillDate** | **string** | **datetime (7),not null** |
| **BillTotal** | **double** | **float, not null** |
| **BillStatus** | **bool** | **bit, notnull** |

**Table name: Hall Table**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINTS** |
| **HallTableId** | **int** | **PK, int, not null** |
| **HallTableSize** | **int** | **int, not null** |
| **HallTableStatus** | **bool** | **Bit, not null** |

**Table name: Help**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINTS** |
| **HelpId** | **int** | **PK, int, not null** |
| **HallTableId** | **int** | **FK, int, not null** |
| **UserName** | **string** | **nvarchar(max), null** |
| **UserEmail** | **string** | **nvarchar(max), null** |
| **issue** | **string** | **nvarchar(max), null** |

**Table name: Feedback**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINTS** |
| **FeedbackId** | **int** | **PK, int, not null** |
| **HallTableId** | **int** | **FK, int, not null** |
| **Rating** | **int** | **int, not null** |
| **Comments** | **string** | **nvarchar(max), null** |
| **FeedbackStatus** | **bool** | **Bit, not null** |

**Table name: Order**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **DATA TYPE** | **CONSTRAINTS** |
| **OrderId** | **int** | **PK, int, not null** |
| **FoodId** | **int** | **FK, int, not null** |
| **HallTableId** | **int** | **FK, int, not null** |
| **Quantity** | **int** | **int, not null** |
| **OrderTotal** | **int** | **int, not null** |
| **Orderstatus** | **bool** | **bit, not null** |
| **OrderDate** | **DateTime** | **datetime(7),not null** |

**MY SQL DESCRIPTION**

**Diagram

Description automatically generated**

## **External Interface Requirements**

**User Interfaces:**

###### Customer Interface:

###### The customer interface will contain three screens. All three screens will have a consistent layout.

**Place Order:** In this screen, system shows a list of cards of dishes. Each dish will have an image, its price per serving.

**Timer and Edit/Cancel order:** After confirming the order, the user will be shown a timer screen. In this screen customer will be shown “Edit Order” and “Cancel Order” buttons and a timer which shows the completion time of the order. There will also be a button to request for bill**.**

**Feedback:** in feedback screen, at the top right corner a button for “Request Bill” will be shown. Beneath this button we will display a form which will have different multiple-choice questions and a submit feedback button.

###### Head Chef Interface:

In head chef interface, system will show all the current orders in detail i.e. all the dishes of a particular order. In each order, there is a button which will be used to mark that dish cooked. Moreover, when customer wants to remove a dish from his order, system will show head chef a notification to approve the removal of the dish.

###### Hall Manager Interface

Hall manager will have a screen where he will get notification whenever an order is completed. System will notify the hall manager about the order number and table number. Hall Manager also has a screen where all orders are listed, and status button to mark the order as paid. Moreover, he also has an interface screen to see and the status of tables in the restaurant as free/available.

###### Admin Interface

As Admin is authorized to perform *CRUD* operations on Staff Members, Menu Items and Inventory Items. He’ll be having three different screens for Staff Members, Menu Items, and Inventory.

**HARDWARE INTERFACES**

Our system can interact with a hardware device directly. We have to connect our system to the bill printer for handing the hard copy of the bill to the customer. For billing module, we may have to use a credit card reader for payment, but the interaction and the results generated by that reader are just entered into our system manually by the user. Moreover, the central screen in the kitchen will be displaying the status of order queues.

###### SYSTEM REQUIREMENTS

**Place Order**

***Description and Priority***

The system will give customers the ability to place their orders using our product. It will display a list of available and unavailable dishes in the menu where unavailable dishes will be grayed out. Customer will be able to select multiple dishes and their quantity for a particular order.

###### *Stimulus/Response sequences*

When user enters the order activity/page, initially system displays a list of available and unavailable dishes along with their prices.

|  |  |
| --- | --- |
| **Stimulus** | **Response** |
| Customer taps on an unavailable dish. | Nothing happens. |
| Customer enters the quantity and press OK button. | System closes the popup, shows a small green tick mark at the side of dish. Below the tick mark shows quantity selected and total price of that dish. |
| If Customer taps on cancel button after 1. | the system closes the popup, and nothing happens. |

**FUNCTIONAL REQUIREMENTS**

REQ-1: The system will show a list of cards (UI element) of dishes. Each card will have a picture of the dish. Below the dish it shows the price in Rupees per serving.

REQ-2: The system must show all available and unavailable dishes to the Customer.

REQ-3: Tap on any of the displayed dish will result in a popup for quantity and a green mark after quantity has been selected.

REQ-4: The popup for quantity input will not allow the user to enter letters, negative numbers or any invalid characters.

REQ-5: After completing the order the system will display a timer “Time to complete the order” and it is the total time required to serve the dish keeping in view the previously queued orders. Moreover, it also shows a cancel order button.

REQ-6: Unavailable dishes must be displayed but their operations must be disabled.

**Customer Help**

**Description and Priority**

Our system will provide help for the customer if the customer faces issues in using the tab, there will be a ‘help’ option in his interface. If he faces issues in using the tab or want some other assistance, he can notify the hall manager through the system.

###### *Stimulus/Response sequences*

###### The home screen for the customer shows a help button on top right corner of the screen

|  |  |
| --- | --- |
| **Stimulus** | **Response** |
| Customer taps on the help button. | System shows a popup with two buttons, “Call a waiter to manage order”, “Call a waiter for help” |
| Customer taps on any of the button. | System closes the popup and sends a notification to the Hall Manager. Notification will include the table number of the Customer. |

###### Functional Requirements

REQ-2: The system must give Customer the ability to ask for help.

REQ-1: When the customer taps on “Call the waiter to manage order”. the system must store that the above order was given by the waiter.

###### Chef Order and Queue

###### Description and Priority

Whenever a new order is placed by the Customer, the dishes in the orders are classified into categories. The system has the information of specialty of each chef, it will assign each dish to a corresponding chef and place it in the order queue of that chef. There is a centralized screen in the kitchen which displays queues for each chef. Each item in the queue is labeled with the name of the dish.

###### *Stimulus/Response sequences*

**Stimulus:**

Customer taps the “Confirm Order” button in “Place Order screen”.

###### Response:

Displays the dishes on kitchen screen in corresponding chef’s queue.

###### Functional Requirements

REQ-1: System will classify the dishes in the order according to category and add this dish on a particular chef’s queue in the kitchen screen.

**Edit Order**

###### Description and Priority

Customer can edit the order any time before the serving. In editing mode, the customer can change the quantity of the of the food ordered, add and remove dishes from the order.

###### *Stimulus/Response sequences*

The timer screen shows two buttons “Cancel Order” and “Edit Order” button

|  |  |
| --- | --- |
| **Stimulus** | **Response** |
| Customer taps on “Edit Order” button. | System shows the previous menu screen where selected dishes are already marked with green tick. |
| Customer taps on any of the selected dish. | System opens a popup with previous quantity pre-filled. This popup will also contain a button “Remove Dish”. |
| Customer enters new quantity and press “OK” | System shows an error “Cannot edit <Name> dish” or System closes the popup and new quantity will bec System shows an error “Cannot edit <Name> dish” or System closes the popup and new quantity will be displayed on that dish in the list. |
| Customer taps on “Remove Dish” | system responds with “Dish <Name> removed” or “Dish <Name> cannot be removed” |
| Customer taps on any new dish which was not previously selected | stimuli/Responses of “Place Order” feature will be followed. |

###### Functional Requirements

REQ-1:

System must allow the Customer to increase, decrease or even remove the dish from the order any time before serving.

REQ-2:

System must remove the dish or decrease quantity of the dish with the approval of head chef.

###### CANCEL ORDER

###### Description and Priority

Our system will also provide an option to cancel the current order. When the customer taps on the “Cancel Order” button. Customer can cancel the order at any time before serving.

###### *Stimulus/Response sequences*

**Stimulus:**

Customer taps on the “Cancel Order” button

###### Response:

system responds with a popup “Order canceled successfully” or “Order cannot be cancelled”

###### Functional Requirements

###### REQ-1: System must allow the customer to cancel order at any time before serving.

REQ2: In cancel order, all the dishes will be presented for approval to the head chef. Only approved dishes will be dropped.

##### **MARK DISH AS COOKED**

###### Description and Priority

The head chef can mark the dish of a particular order complete when notified by the chef.

###### *Stimulus/Response sequences*

The system will show a list of current orders in earliest first order in head chef screen. Each order also shows order no and table no associated with the order. Moreover, it also shows a list of dishes for each order. Alongside each dish there is a button saying, “Marked Cooked”.

###### Stimulus:

Head chef taps on the “Mark Cooked” button on a dish in an order.

###### Response:

System changes that button to a green tick.

###### Stimulus:

All the dishes of a particular order have been marked “cooked”

###### Response:

System shows a notification to the hall manager saying, “Order of Table No

<Table No> is ready for serving”. The system shows a new screen having Title “Food Ready” to the Customer showing a button “Request Bill” and MCQ’s for customer feedback.

###### Functional Requirements

###### REQ-1:

The system must send a notification to the hall manager once all the dishes of a particular order has been marked “cooked”.

###### REQ-2:

The system must replace the timer screen with a new screen having feedback and request bill options.

##### **REQUEST BILL**

###### Description and Priority

Request bill option gives the ability to the customer to ask for receipt and pay the bill.

###### *Stimulus/Response sequences*

**Stimulus:**

Customer taps on the request bill button

###### Response:

The system prints the bill through a printer. System will add a bill to the hall manager’s view with the button that says “paid”.

###### Functional Requirements

REQ-1: The system must notify the hall manager that a customer has request for a bill

REQ-2: The system must show Hall manager the order no, table no and total payable amount

REQ-3: The system must give ability to the hall manager to change the status of the bill to paid.

##### **Customer Feedback**

###### Description and Priority

The system will give customers the ability to give a feedback for the food or overall services. In the feedback screen there are multiple choice questions each having two options “Satisfactory” and “Unsatisfactory”. At the end there is a submit button.

###### *Stimulus/Response sequences*

**Stimulus:**

The customer taps on request bill option

###### Response:

The system shows a feedback screen with multiple choice questions and a submit button.

###### Functional Requirements

REQ-1: System must show the feedback screen to the user.

###### REQ-2: System must display multiple choice questions for feedback

##### **Add/Edit/Delete Staff Members**

###### Description and Priority

The system gives the ability to the admin to add, edit and delete staff members. Using this feature an admin can add chefs, waiters, managers.

###### *Stimulus/Response sequences*

Admin/Manage screen shows a grid of staff members. There is a button at the top of grid which says Add Member. In the grid after every entry there is a “Edit” and “Remove” button.

responds with a “<Staff Name> removed successfully”

|  |  |
| --- | --- |
| **Stimulus** | **Response** |
| Admin taps on “Add Staff” button | System opens another screen with a form |
| Admin fills the information and hit submit | System responds with “<Staff Member> added successfully” |
| Admin taps on edit button | The system opens a screen with a form prefilled with the existing values. |
| Admin edits the information and hit submit | System responds with “<Staff Member> edited successfully” |
| Admin taps on remove button on a particular row | responds with a “<Staff Name> removed successfully” |

###### Functional Requirements

REQ-1: Admin should be able to add all necessary information about the staff member

REQ-2: System must give admin the ability to edit information about all staff members

REQ-3: System must give admin the ability to remove staff members.

##### **Add/Edit/Delete Menu Items**

###### Description and Priority

The system gives the ability to the admin to add, edit and delete staff members. Using this feature an admin can add chefs, waiters, managers.

###### *Stimulus/response sequences*

The admin screen shows all the previously added dishes. It also shows a “Add Dish” button along with “Edit” and “Remove” with all the available dishes

|  |  |
| --- | --- |
| **Stimulus** | **Response** |
| Admin taps on “Add Dish” button | System opens another screen with a form |
| Admin fills the information and hit submit | System responds with “<Dish> added successfully” |
| Admin taps on edit button | System opens a screen with a form prefilled with the existing values |
| Admin edits the information and hit submit | System responds with “<Dish Member> edited successfully” |
| Admin taps on remove button on a particular row | responds with a “<Dish> removed successfully” |

###### Functional Requirements

REQ-1: Admin should be able to add all necessary information about the staff member

REQ-2: System must give admin the ability to edit information about all staff members

REQ-3: System must give admin the ability to remove staff members.

###### DATA FLOW DIAGRAM

###### Diagram Description automatically generated

###### SAMPLE SCREENS

###### This is the main screen of the application which allows the user to LOGIN as an ADMIN /HALL MANAGER.HEADCHEF/CHEF.

###### A picture containing website Description automatically generated

###### Using forget password option we can reset the password and create a new password

###### Graphical user interface, application Description automatically generated

###### CUSTOMER INTERFACE SCREEN: where customers can add items to cart, see the availability of dishes.

###### Graphical user interface, website Description automatically generated

###### Here the screen shows the list of ordered items, customer can edit their order and cancel the order

###### A screenshot of a computer Description automatically generated

###### This is the confirm order screen for the customer where he/she has added some items in the cart and has the capability to remove some items from the cart and confirm order.

###### Graphical user interface, website Description automatically generated

###### Bill generated for the ordered items.

###### Graphical user interface Description automatically generated

###### This is the main screen of the Hall manager’s dashboard showing all the functionalities that he can perform example: reservation of tables and accept the bills to be paid .

###### This is the layout where hall manager can reserve tables of different capabilities for a customer and check the status of all tables

###### 

###### Graphical user interface Description automatically generated

###### This is the feedback screen where the customer can give his feedback about the services of the restaurant once he has paid the order bill.

###### Graphical user interface, application Description automatically generated