CS 387 Project - Restaurant Management System

Project Report

30 April, 2022

IIT-Bombay

Team YARA

190050107	Satwik M
190050124	Thivesh M
190050038	Akash G
190050026	Karthikeya B

Github Repository : https://github.com/Akash-116/CS387-Project

Table of contents :-

1. Problem Description :

A typical Restaurant Management System has to support a complete restaurant setting, from the owner's endpoint of managing a business to the customer's endpoint of having a good dining experience. It should be neat, fast, easy-to-use, resilient to heavy loads. Analyzing the myriad of data captured and providing insights into them is also a possibility. Such predictive analytics is an important feature for businesses, helping them in making key decisions.

YARA a.k.a. Yet-Another-Restaurant-App, is a one-stop solution for managing your restaurant business. It provides an interface to both the managers and customers. Owners and personnel at various levels - waiters, chefs, inventory managers, billing managers, delivery managers can easily monitor their responsibilities such as customer orders, inventory levels, revenue for the day, online deliveries among others. Customers can login into the website and place online orders on the website after going through the menu of the restaurant.

The application is supposed to capture and deliver large amounts of data of various types, in a quick and organized fashion. As such, databases are the ideal choice over file systems. Databases provide many advantages over file system storage such as accessibility, maintaining data integrity, concurrent access to the data, data security, and data resilience. These features are necessary for a practical, day-to-day business application.

Relational databases serve our purpose. In a restaurant setting, we typically have a lot of abstraction as entities and relationships between these entities. And as such, relational databases are a good choice for modeling this abstraction. GraphDBs don't serve much purpose here since we don't have a network of relationships among entities and rather have mostly single-level relationships. Time Series DBs also don't fit with our problem statement since we don't have extensive usage of time attributes.

For developing the frontend, we are planning to use ReactJS and for the backend, NodeJS. To implement and manage databases, we are using PostgreSQL API from NodeJS. To incorporate analytics into our databases, we plan to use Apache Spark engine. Leveraging this analytics, we plan to provide various observations and patterns to both customers and managers.

2. User classes and use cases :

The classes of users are - Restaurant Manager, Chefs, Head Waiter, Waiters, Inventory Manager, Billing Manager, Delivery Manager, Delivery Persons, Customers

The use cases with the corresponding primary actors are as follows -

#1

Title: Logging on to the system

Description:

It will allow the user to log onto the website to access it.

Trigger event:

It will be both automatically Triggered.

Primary actor:

Everyone.

Inputs:

Login details such as email/username and password.

Preconditions:

No user is logged in or the previous user logged out from this computer.

Main Success Path:

The correct email and password are entered and the login is successful.

Exceptions:

Wrong password:

We will redirect to the login page and notify the user that their password is Wrong.

Postconditions:

The system will take the user to the home page.

#2

Title: Changing/Adding Dishes and Menu

Description:

It allows the user to change the available dishes and change the menu accordingly.

Trigger event:

It is Human Triggered.

Primary Actor:

Restaurant Manager, Chef

Inputs:

Dish name, new ingredients, new menu

Preconditions:

The user is logged in and has appropriate privileges.

Main Success Path:

If a new dish is to be added, then we add a new row to the dishes table and add appropriate rows to the dishes_items table

If a dish is being updated then the corresponding row is updated in the dishes table and correspondingly rows are added or deleted in the dishes_items table

Exceptions:

If some constraint fails:

Update or Adding is unsuccessful

Postconditions:

The menu will be updated and the dish added/updated.

#3

Title: Manage Table Status

Description:

Allows the user to manage the status of tables (Occupied/Empty)

Trigger event:

Human Triggered

Primary Actor:

Head waiter, Restaurant Manager

Inputs:

The new status of the table

Preconditions:

The user is logged in and has appropriate privileges.

Main Success Path:

The status column in the row corresponding to the table gets updated

Exceptions:

The Table in the question may not be present, So we will give out an error message.

Postconditions:

Table status gets changed.

#4

Title: Manage Orders and Customers

Description:

It allows users to add/remove/edit orders by customers and manage customer details.

Trigger event:

It is Human Triggered.

Primary Actor:

Restaurant Manager, Billing Manager

Inputs:

Order by customers, Details of customers.

Preconditions:

The user should have the necessary privileges and the input data should be of appropriate types.

Main Success Path:

If a new order is to be added, then we add a new row to the Orders table and add appropriate rows to the orders_dishes table

If a new customer is to be added, then we add a new row to the Customers table.

Exceptions:

If the input data types are not proper or the user does not have the required permissions we will give an error message.

Postconditions:

Order is added/updated to Orders Table or customer is added/updated to Customers Table.

#5

Title: Manage delivery People

Description:

Helps manage the delivery personnel

Trigger event:

Human Triggered

Primary Actor:

Delivery Manager, Restaurant Manager

Inputs:

Name of the person, their details to be updated

Preconditions:

The user should have the necessary privileges and the input data should be of appropriate types.

Main Success Path:

If a new person is to be added, we check if the person already exists.

If a person's details are being changed, we check that the details are valid

Exceptions:

If the input is invalid/wrong we notify the user through an error message.

If the user doesn't have the necessary permissions we redirect them back to the home page.

Postconditions:

A new delivery person is added/ the details of that person are updated

#6

Title: Manage Items

Description:

It allows users to add/remove/edit items.

Trigger event:

It is Human Triggered.

Primary Actor:

Restaurant Manager, Head Waiter

Inputs:

Details of items to modify/add.

Preconditions:

The user should have the necessary privileges and the input data should be of appropriate types.

Main Success Path:

If a new item is to be added, then we add a new row to the Items table

If an item is being updated then the corresponding row is updated in the Items table.

Exceptions:

If the input data types are not proper or the user does not have the required permissions we will give an error message.

Postconditions:

Item is added/updated to Items Table

#7

Title: Manage dishes in Cart

Description:

It allows a customer to remove/add dishes to the cart.

Trigger event:

It is Human Triggered.

Primary Actor:

Customer

Inputs:

Dish to be added/removed in cart.

Preconditions:

The customer has to be logged in.

Main Success Path:

If a new dish is to be added to the cart, then we add a new row to the Cart table

If a dish is being updated/removed then the corresponding row is updated/deleted in the Cart table.

Exceptions:

If the customer is not logged in we will redirect him/her to the login page.

Postconditions:

Dish is added/updated to Cart Table

#8

Title: View Employee details

Description:

It allows a manager to view the details of an employee

Trigger event:

It is Human Triggered.

Primary Actor:

Manager

Inputs:

Employee ID whose details he wants to view

Preconditions:

The manager has to be logged in, as only he can access these details

Main Success Path:

Get the row corresponding to the employee and show his details on the page

Exceptions:

If there is no employee with that ID then we print an error message

Postconditions:

Employee details are shown on the webpage

#9

Title: Manage Employees

Description:

It allows a manager to edit the details of an employee or delete and add employees

Trigger event:

It is Human Triggered.

Primary Actor:

Manager

Inputs:

Employee ID whose details he wants to edit or remove and the details to update

Details of the new employee he wants to add

Preconditions:

The manager has to be logged in, as only he can access these details

Main Success Path:

Update the row corresponding to the employee

Add row corresponding to the new employee

Exceptions:

If there is no employee with that ID then we print an error message

Postconditions:

Employee details are shown on the webpage

#10

Title: View menu

Description:

It allows users to view the dishes and their details and cost

Trigger event:

It is Human Triggered.

Primary Actor:

All users can view the menu

Inputs:

Dish ID if he wants to view details of one dish, otherwise look at the whole menu

Preconditions:

The user has to be logged in

Main Success Path:

Get the details of the row corresponding to all the dishes or a specific dish

Exceptions:

If there is no dish with that ID then we print an error message

Postconditions:

Menu or dish details will be given on the webpage

#11

Title: View and Edit Restaurant details

Description:

It allows the manager to view and edit restaurant details

Trigger event:

It is Human Triggered.

Primary Actor:

Manager

Inputs:

Various details of our Restaurant

Preconditions:

The user has to be logged in

Main Success Path:

Get the details of the row corresponding to all the dishes or a specific dish

Exceptions:

If there is no dish with that ID then we print an error message

Postconditions:

Menu or dish details will be given on the webpage

#12

Title: Best Waiter

Description:

It allows a manager to see the best waiter

Trigger event:

It is an Automatic Trigger

Primary Actor:

Manager

Inputs:

Nothing

Preconditions:

The manager has to be logged in, as only he can access these details

Main Success Path:

Getting data about waiter and displaying best waiter

Exceptions:

If the manager is not logged in, we redirect to the login page

Postconditions:

The best waiter is shown

#13

Title: Best Dishes according to rating

Description:

It allows everyone to see the best dish.

Trigger event:

It is an Automatic Trigger

Primary Actor:

All users

Inputs:

Nothing

Preconditions:

The user has to be logged in, as only he can access these details

Main Success Path:

Getting data about the dishes and displaying dishes with the best rating

Exceptions:

If the user is not logged in, we redirect to the login page

Postconditions:

Best rating dish is shown

#14

Title: Best Customer

Description:

It allows a manager to see the best customer

Trigger event:

It is an Automatic Trigger

Primary Actor:

Manager

Inputs:

Nothing

Preconditions:

The manager has to be logged in, as only he can access these details

Main Success Path:

Getting data about customers and displaying the best customer

Exceptions:

If the manager is not logged in, we redirect to the login page

Postconditions:

The best customer is shown

#15

Title: Most ordered dish

Description:

It allows a manager to see the most ordered dish

Trigger event:

It is an Automatic Trigger

Primary Actor:

Manager

Inputs:

Nothing

Preconditions:

The manager has to be logged in, as only he can access these details

Main Success Path:

Getting data about dishes and displaying the most ordered dish.

Exceptions:

If the manager is not logged in, we redirect to the login page

Postconditions:

The most ordered dish is shown

#16

Title: Best day of the week

Description:

It allows a manager to see the best day of the week

Trigger event:

It is an Automatic Trigger

Primary Actor:

Manager

Inputs:

Nothing

Preconditions:

The manager has to be logged in, as only he can access these details

Main Success Path:

Getting data about orders and displaying the best day of week

Exceptions:

If the manager is not logged in, we redirect to the login page

Postconditions:

The best day of the week is shown.

#17

Title: The best delivery person

Description:

It allows a manager to see the best delivery person

Trigger event:

It is an Automatic Trigger

Primary Actor:

Manager

Inputs:

Nothing

Preconditions:

The manager has to be logged in, as only he can access these details

Main Success Path:

Getting data about delivery persons and displaying the best delivery person

Exceptions:

If the manager is not logged in, we redirect to the login page

Postconditions:

The best delivery person is shown

3. User interfaces for use cases :

The user interfaces for different user types are as follows -

For Everyone:-

Use Case : Login
- form:
  - Username <text>
  - password <password>
  - New User?
  - Forgot Password?
  - Login with google
- Input:
  - Login credentials
- Output:
  - Entry into website

For Owners:-

Use Case: View Employee details
- Input:
  - Select an employee to view
- Output:
  - Employee details visible

Use Case: Delete Employee details
- Input:
  - Select the employee to Delete
- Output:
  - Employee record deleted

Use Case: Add Employee details
- Form:
  - Name
  - Address
  - Ph. No.
  - Status
  - Type
  - Salary
- Input:
  - Fill the form for adding an employee
- Output:
  - Success/Fail

Use Case: View Dish details
- Input:
  - Select dish to view details
- Output:
  - Dish details visible

Use Case: Delete dish
- Input:
  - Select dish to Delete
- Output:
  - dish record deleted

Use Case: Add Dish details
- Form:
  - Name
  - Recipe
  - Time taken
  - Cost
  - Type
  - Photo
- Input:
  - Fill the form for adding a dish
- Output:
  - Success/Fail

Use Case: View and edit Restaurant details
- Form:
  - Name
  - Address
  - Ph.No.
  - Description
  - Photo
- Input:
  - Various details of our Restaurant
- Output:
  - Success/Fail

Use Case: View Item details
- Input:
  - Select Item to view details
- Output:
  - Item details visible

Use Case: View Items purchased and used up on that day

Input:

None

Output:

Per day information on items consumed and bought

Use Case: View Table Status

Input:

Select table to view Status

Output:

Status of the table - location, status are displayed

Use Case: View order history details

Input:

Select order for details

Output:

Order details - cost, customer, Items, Mode - are displayed

Use Case: View customer details

Input:

Select Customer to view

Output:

Customer details - name, ph.no, address, no. of orders, no. of dishes ordered -are displayed

Use Case: View offer details

Input:

Select offer to display

Output:

Offer details - offerID, discount given - displayed

Use Case: View delivery person details

Input:

Select delivery person to display

Output:

Display delivery person details - Primary area, Secondary area, EmployeeID

For Customers:-

Use Case: View menu and select items

Form:

Various items are selected

Input:

User selects needed items and clicks "Next"

Output:

If successful, the user is shown the CART page

Use Case: View cart and confirm the order

Input:

Just confirms the order

Output:

Success/Failure

Use Case: View previous orders

Input:

Select order to view

Output:

Previous order details shown

Use Case: Change account details

Form:

Name

Address

Ph. No.

Input:

Input required details

Output:

Success/Failure

For Chef:-

The relevant subset of Owner use cases - (Manage Dishes,Menu,Recipe of dishes)

For Head Waiter:-

The relevant subset of Owner use cases - (Manage Table Status, Manage Waiters)

Use Case: Change table status

- Form:
  - tableID
  - location
  - Status
- Input:
  - Fill the form
- Output:
  - success/failure

For Inventory Manager :-

The relevant subset of Owner use cases - (Manages inventory of items)

Use case : Update about items

- form:
  - itemID
  - Name
  - Market Cost
  - Quantity of inventory
- Input :
  - fill form
- Output:
  - Success/failure

For Billing Manager:-

The relevant subset of Owner use cases - (Managers Orders and Customers)

Use Case: Create an order

form:

customer

dishes

Type

Offer

Input:

fill the form

Output:

Success/Failure

For Delivery Manager:-

The relevant subset of Owner use cases - (Manages Delivery Persons and Area Codes)

Use Case: Assign a delivery person

form:

Order

Customer

Delivery Person

Address

Input:

Fill the form

Output:

Success/Failure

4. Entities and ER diagram :

We have included the main entities and the information about them as their attributes in the following ER Diagram on the next page -

We represented the relations also as rectangular boxes similar to entities. For Entities, the header is darker, for relations the header is white.