(Database Management Project)

ONLINE RESTAURANT

UCE2021521:Koshin Gautam
UCE2021532:Gungan Khatri
UCE2021533 :Bhakti Khokad
UCE2021535:Prerna Kothari

Problem Statement

Our proposed system is an online Restaurant system that enables ease for customers. It overcomes the disadvantages of the traditional queueing system. Our proposed system is a medium to order hassle-free online food from restaurants and mess service. This system improves the method of taking orders from customers. The online food ordering system sets up a food menu online and customers can easily place the order as per their wish. Also with a food menu, customers can easily track their orders. For more secure ordering separate accounts are maintained for each user by providing them an ID and a password.

“The objective of this online restaurant project is to design a database to centrally handle the details of all the customers and to provide access to this information with an easy-to-use Graphic user interface that can be accessed by any device with basic Java (swing) rendering capabilities.”

Key features:

- An online food ordering system is software that lets restaurants, coffee shops, or bars accept orders online. It typically allows customers to choose and pay for food, then alerts the kitchen when an order is made. This happens without contact between staff and customers

- Here we have inculcated the basic and advanced feature of mysql

Our database dbms_miniproject has in total 5 tables namely Admin , Customer, Menu, Cart, Order using which we have made our login signup frames.

Using jdbc connect in java we have given admin the facility of altering the menu and viewing the list of customers

And customers with the facity of adding items from menu to the cart.

Key Features :

- In our first frame we get options to login as ADMIN, NEW USER and OLD USER

Where clicking on ADMIN the login page of admin pops up

clicking on NEW USER the signup page of the new user opens up

clicking on OLD USER the login page of user opens up

- After a successfully login as ADMIN we get to the page showing the functionalities of the admin where we can choose if we want to alter the menu or view the details of the customers.
- After a successful login as the USER we get to the page showing the functionalities of the user where we can choose to alter the cart of simply just view the menu.

Key Features :

- Here we used the swing GUI where we inculcated buttons, frames, panels, labels and text boxes into our program .
- The connectivity of a button to another frame was carries out by the setVisible function.

  Eg : new login().setVisible(true);

We have also made an exit button which terminates our entire project using the simple function of System.exit(0)

- We used a wide variations of colours and font which makes our project more user interactive.

DataBase Requirements

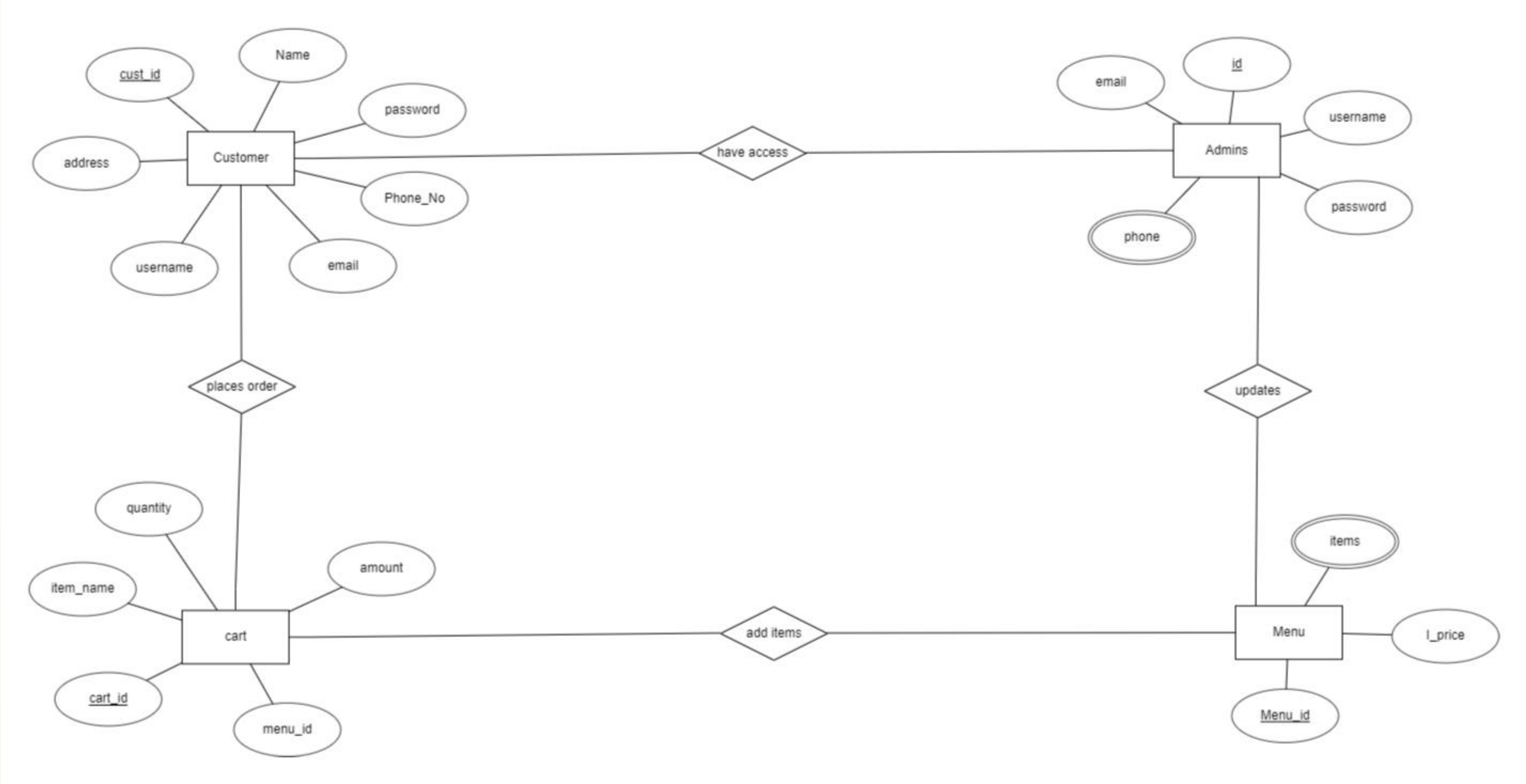
1. **Admin Table**: The table stores the details of the admin including admin id, username, password, email, phone number, and other relevant information

2. **Menu Table**: This table contains information about menu_id, available items, and Item prices.

3. **Customer Table**: The Customer Table stores the details of all customers including their Name, cust_id, email,phone_no, username, password, address, and other relevant details.

4. **Cart Table**: The table stores info about the ordered item_name, quantity of the item, amount, menu_id, and cart_id.

Online Restaurant Management System(ERD)



Normalization

- Normalization is the process of organizing the data in the database.

"The main objective of database normalization is to eliminate redundancy, minimize data modification errors, increase integrity, and simplify the query process."

After Normalization:

1. Improved overall database organization
2. Data consistency
3. Reduces redundancy
4. Cost reduction
5. Increased security

First normal form (1NF)

First Normal Form (1NF) is the first step in the normalization process in SQL. It requires that a table meets certain criteria to ensure that each column contains atomic values and each row is unique. Specifically, a table is in first normal form if:

1. **Each table cell contains a single, indivisible value (i.e., atomic value).**
2. **Each column in a table has a unique name.**
3. **The order of rows and columns is irrelevant.**
4. **Each row in a table is unique; no two rows contain exactly the same data.**

To convert a table that is not in 1NF into 1NF, you may need to split columns that contain multiple values into separate columns, or create additional tables to represent relationships between entities.

Second normal form (2NF)

Second Normal Form (2NF) is the second step in the normalization process in SQL. It requires that a table meets certain criteria to ensure that each non-key column in the table is functionally dependent on the entire primary key. Specifically, a table is in second normal form if:

1. **It is already in first normal form (1NF).**
2. **All non-key columns in the table are dependent on the entire primary key.**

To bring a table that is not in 2NF into 2NF, we may need to create additional tables to represent relationships between entities, or split columns that contain multiple values into separate tables

Third normal form (3NF)

Third Normal Form (3NF) is the next step in the normalization process in SQL. It requires that a table meets certain criteria to ensure that each non-key column in the table is only dependent on the primary key and not on any other non-key column. Specifically, a table is in third normal form if:

1. **It is already in second normal form (2NF).**
2. **All non-key columns in the table are dependent only on the primary key.**

In other words, a table should not contain any transitive dependencies, where a non-key column is dependent on another non-key column that is not part of the primary key. To bring a table that is not in 3NF into 3NF, we may need to further split the table into smaller tables, or remove columns that contain redundant information

Fourth normal form (Boyce Codd Normal Form) and beyond

- While normalizing your database in accordance with 4NF, 5NF, and 6NF, is recommended, most relational databases do not require more than 3NF to be satisfied to be considered normalized. The benefits of data normalization beyond 3NF don’t always cause significant errors when there are updates, deletions, or insertions of data. However, if your company utilizes complex datasets that get changed frequently, it is recommended that you also satisfy the remaining normal forms.