RESTAURANT

Management System

Prepared By:

[أنس عصام محم ود السيد - 201900194]

[عبد ال رحمن حس ي حمدى عبدال رحمن - 201900412]

[إس لم شعبان عيد شعبان - 201900142]

[بثينة سيد عبد الس لم محم ود - 201900221]

[شين محمد ر زق محمد - 201900370]

DATE

[2022/5]

Copyright @Template.net

CONTENT

- Description
- Class Diagram
- Database Schema
- Design Pattern

Copyright @Template.net

DESCRIPTION

Our Software is a Restaurant Management System. We made it to help restaurants:

1. **Reduce the order processing time.**
2. **Automate redundant work.**
3. **Deliver a great customer experience.**
4. **Determine Profits & Costs**

This Software has two modules:

1. **Admin**
2. **Employee**

You can log in as Admin or Employee, If
you logged in as admin :

Admin managed to add new employees, list, update and delete them and does the same operations in meals and makes special offers. He can make reports about employees that report contains all information about all employees.

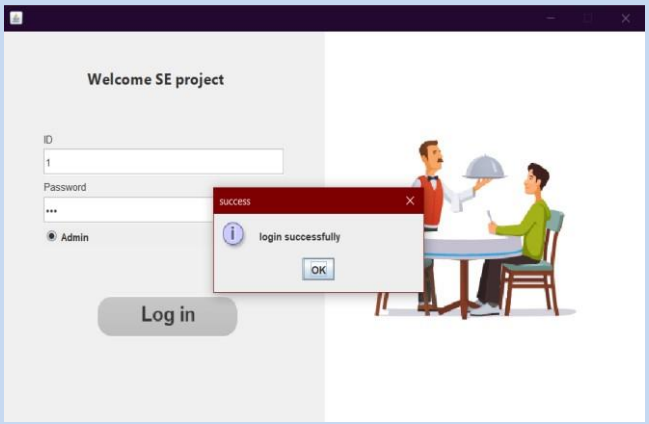
If you logged in as an employee :

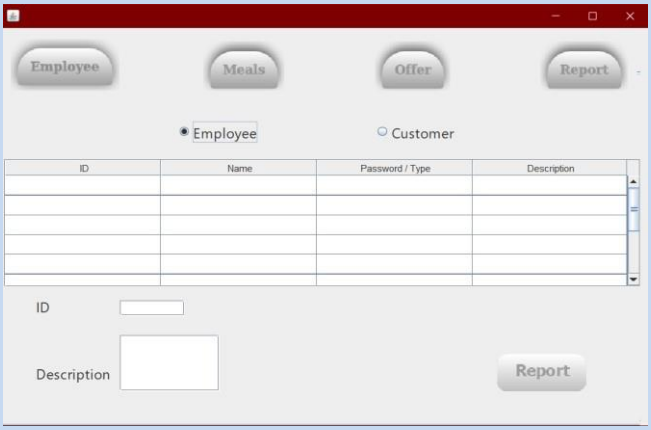
The employee managed to add new Customers, list, update and delete them. He can make and cancel orders and show offers.

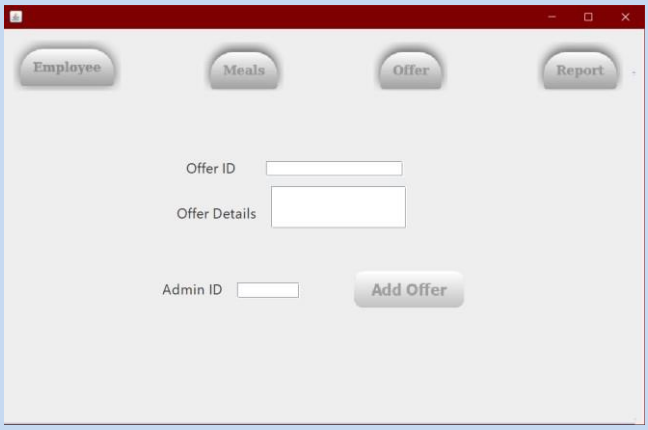
Copyright @Template.net

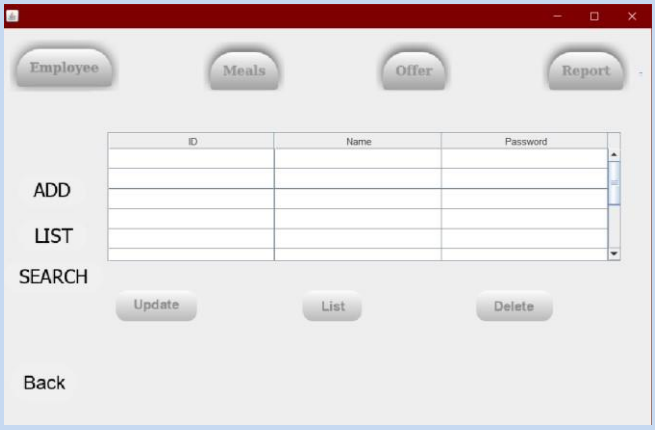
GUI Route

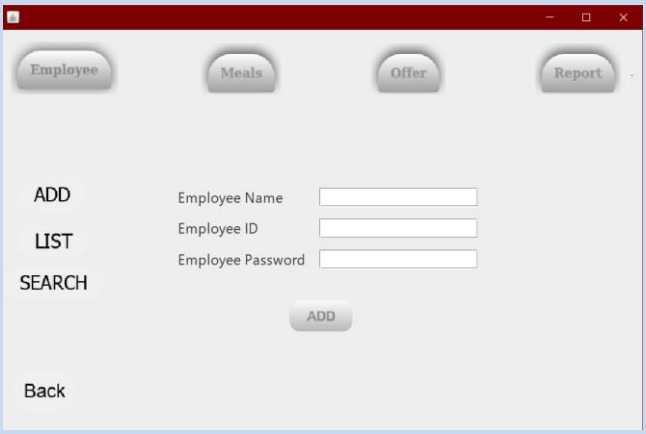
ADMIN VIEW*

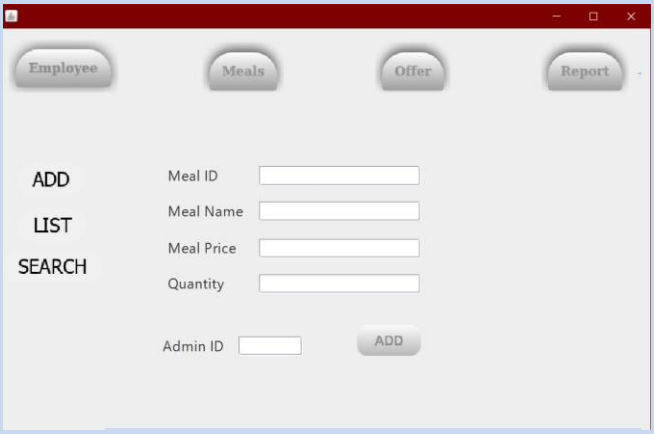






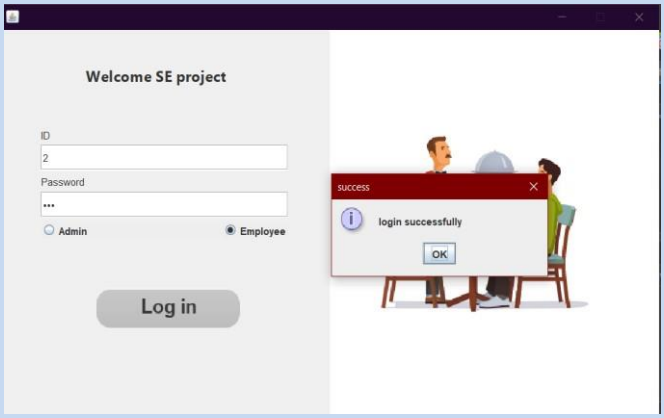


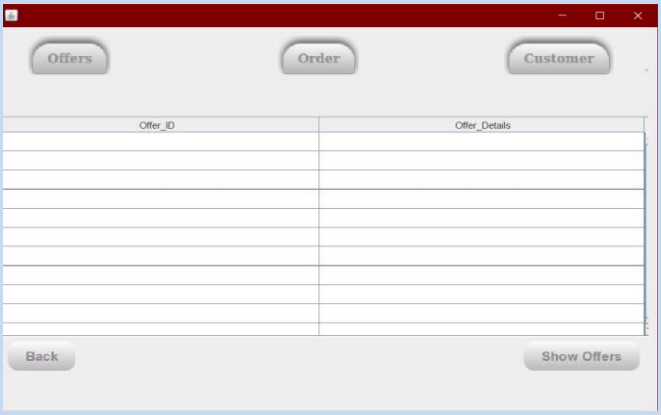


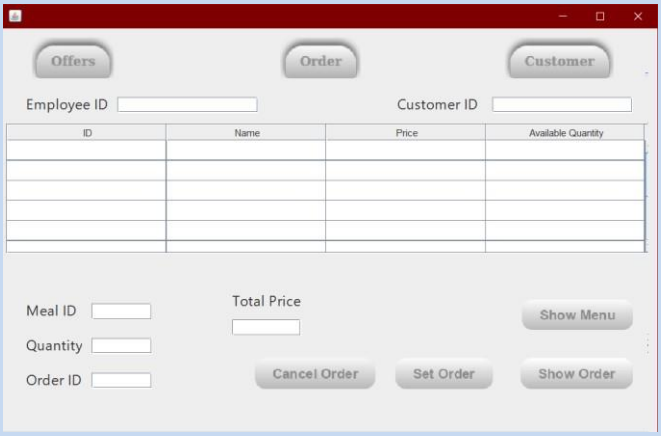


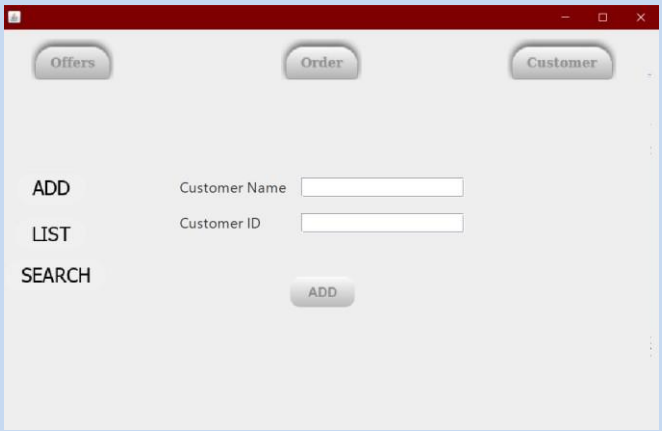
Copyright @Template.net

EMPLOYEE VIEW



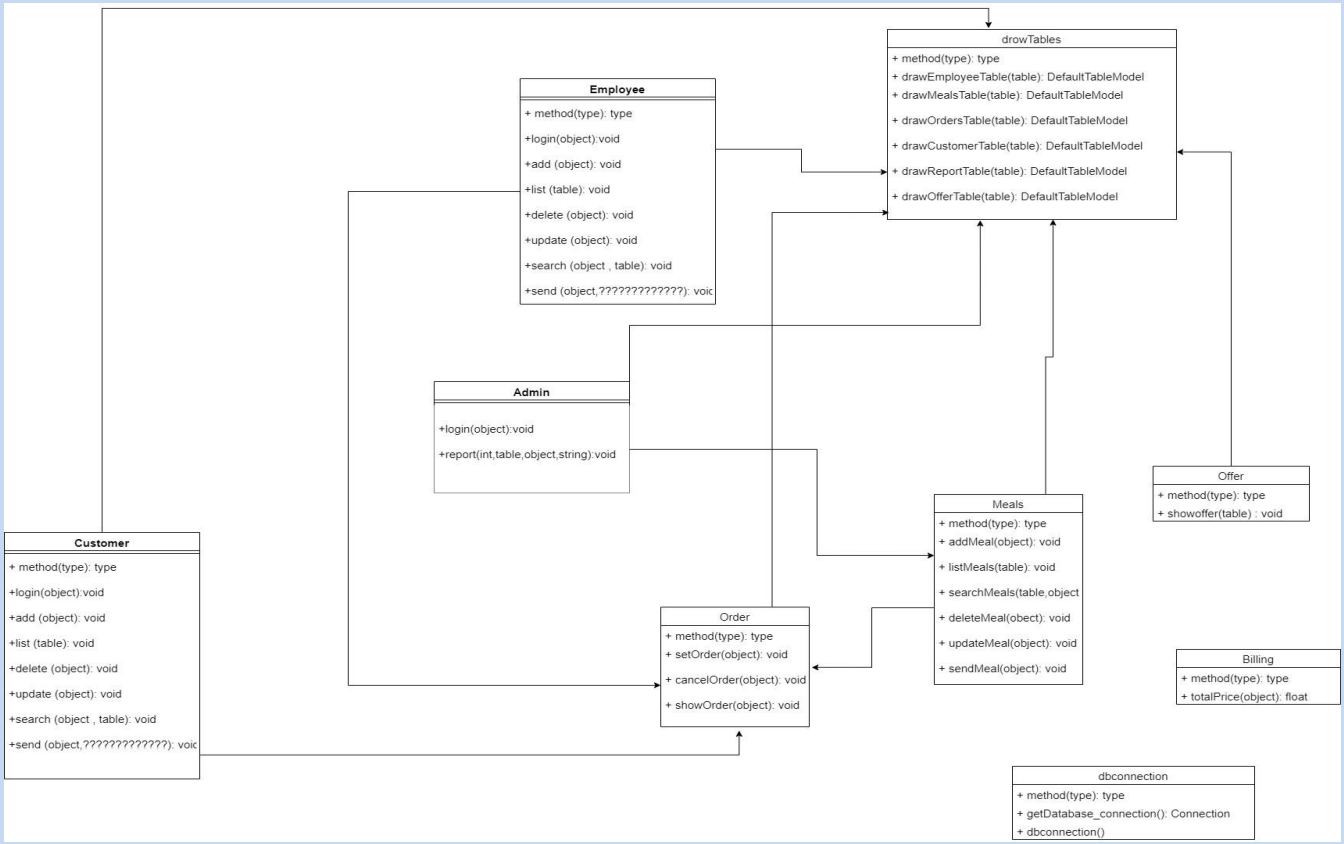






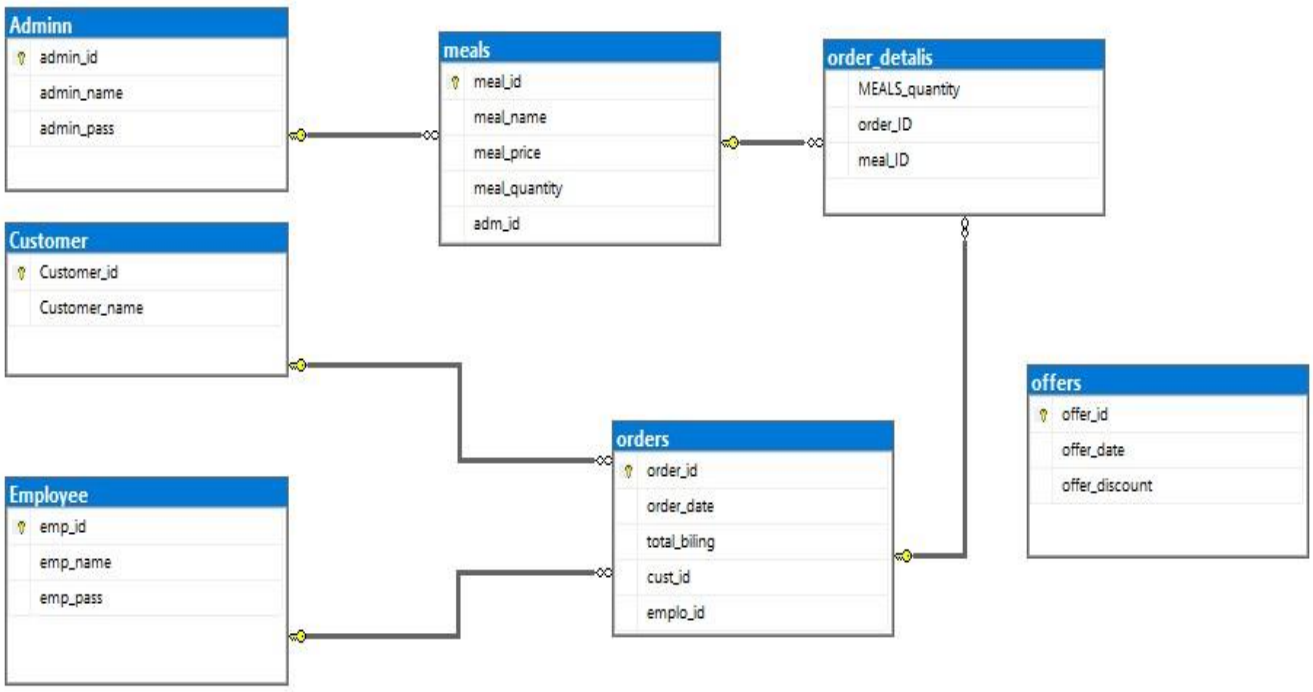
Copyright @Template.net

CLASS DIAGRAM



Copyright @Template.net

DATABASE SCHEMA



Copyright @Template.net

Design Patterns

1-SINGLETON PATTERN

Type: Creational Pattern.

Context: this pattern used to make only one instance from class and avoid creation of any farther instances.

Problem: creation of only one instance from certain class and create a global point to access to it.

Solution: make the constructor of this class as a private contractor to ensure that no other class will be able to create an instance form it & and make a static public method creates the single instance of this class (playing the role of global access point) and stores a reference to that object in a static private variable.

Example: We used this design pattern to make one object database and system classes.

References: The singleton design pattern is one of the twenty-three well-known "Gang of Four" design patterns written by Erich Gamma, Richard Helm, Ralph Johnson, and John Glissades, with a foreword by Grady Brooch.

Copyright @Template.net

2-IMMUTABLE PATTERN

Type: Creational Pattern. 72

Context: An immutable object is an object that has a state that never changes after creation.

Problem: create class with variables does not change after creation.

Forces: There must be no cases that would allow 'illegal' modification of an immutable object.

Solution: ensure that no way to set or modify instant variables except constructor & this class shouldn’t have any setter or method can modify for this variables.

Example: We used this design pattern in database connect class which contain connection configurations.

Copyright @Template.net