**Report for Food Ordinary System**

**Name OsamaKhubaib id 24114**

**Name Muhammad Wassam Baig id :35456**

**Table of Contents**

1. Introduction

2. System Requirements

3. Entity-Relationship **(E-R)** Model

4. Data Flow and Processes

5. Implementation Detail

6. Challenges and Solution

7. Conclusion

------------------------------------------------------------------

***1. Introduction***

The Food Ordinary System (FOS) is designed to streamline and optimize the entire food supply chain process, from production to consumption. This system aims to reduce waste, enhance food safety, improve inventory management, and ensure the efficient distribution of food resources. The core of the FMS is its robust database, which stores and manages all relevant data efficiently and securely.

----------------------------------------------------------------

***2. System Requirements***

To design a functional and efficient Food Management System, the following requirements were identified:

*- Data Storage*: Efficient storage of food item information, supplier details, inventory data, and distribution records.

***- User Management*:** Secure management of user roles, including administrators, suppliers, and customers.

- Reporting: Generation of detailed reports on food stock, usage, waste, and distribution.

------------------------------------------------------------------

***3. Entity-Relationship (E-R) Model***

The E-R model for the Food Management System consists of the following key entities and relationships:

1. *Customer*: name, phone, address, reservation.
2. *Staff*: receptionist, waiters, chef, guards.
3. *Utilities*: Menu, tables, rooms, music.

--------------------------------------------------------------

*Relationships*:

- Staff supplies Food Item.

- Food Item is stored in Inventory.

- Food Item is distributed through Staff.

-----------------------------------------------------------------

***5. Data Flow and Processes:***

The data flow and processes within the FMS are as follows:

**1. *Data Entry:*** Suppliers provide details of food items, which are entered into the Food Item table.

**2. *Inventory Management*:** Upon receiving food items, inventory records are created, updating the Inventory table.

**4. *Distribution*:** Food items are distributed based on inventory data, and distribution records are updated.

5. *Reporting*: Reports on stock levels, usage patterns, and waste are generated from the database.

---

**6. Implementation Details**:

The database for the Food Management System was implemented using MySQL due to its robustness and support for complex queries. Key considerations during implementation included:

- Ensuring data integrity through foreign key constraints.

- Optimizing queries for real-time inventory tracking.

- Implementing user authentication and role-based access control.

-------------------------------------------------------------------

***8. Challenges and Solutions:***

Several challenges were encountered during the project:

**- *Data Integrity*:** Ensuring no duplicate or inconsistent data. Solution: Implemented strict validation rules and constraints.

**- *Scalability***: Handling large volumes of data. Solution: Optimized database schema and indexing strategies.

*- Real-Time Tracking*: Maintaining up-to-date inventory information. Solution: Implemented efficient real-time data processing mechanisms.

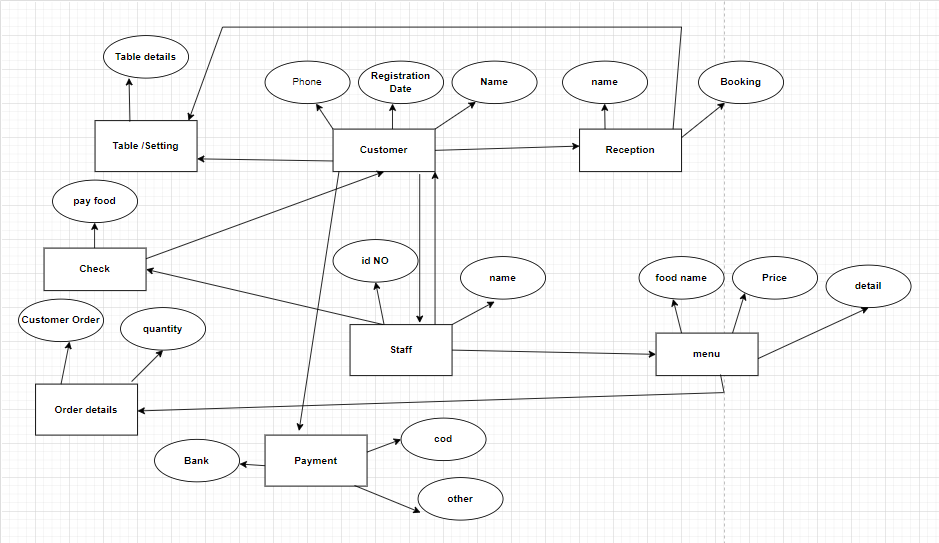
---------------------------------------------------------------

***9. Conclusion***

The Food Ordinary System, with its robust database, successfully addresses the complexities of managing the food supply chain. By leveraging advanced database design and efficient data processing, the system enhances food safety, reduces waste, and ensures efficient distribution. This report has outlined the critical components of the database, from design to implementation, highlighting the significance of a well-structured database in achieving these goals.

The FMS stands as a testament to the power of technology in solving real-world challenges, promising a more sustainable and efficient future for food management.

**Erd**

****

**CODE**

SQL> create user Order\_food3

2 system

3

SQL> create user Order\_food4

2 identified by Abc;

User created.

SQL> GRANT CREATE SESSION to Order\_food4;

Grant succeeded.

SQL> grant create table to Order\_food4;

Grant succeeded.

SQL> grant unlimited tablespace to Order\_food4;

Grant succeeded.

SQL> Create table Customers

2 (

3 Name varchar(10) Primary key,

4 Phone varchar(15) not Null,

5 Registration varchar(16) Not Null);

Table created.

SQL>

SQL> Create table Staffs(

2 Staff\_id number(12),

3 Staff\_name varchar2(29),

4 Staff\_jobtype varchar2(30));

Table created.

SQL> Create table Orderdetail(

2 ordertype varchar2(9),

3 description varchar2(14),

4 price number(10),

5 type varchar2(10));

Table created.

SQL> create table payments(

2 amount number(10));

Table created.

SQL> Create table receptions (

2 Name varchar(14) Not Null,

3 Booking varchar2(13) Not Null) ;

Table created.

SQL> INSERT INTO Customers (name, Registration, phone)

2 Values('John Doe', 'j34', '123-5');

1 row created.

SQL> INSERT INTO Orderdetail ( ordertype, description, price)

2 VALUES ( 1001, 2, 10.99);

1 row created.

"

SQL> DESCRIBE Staffs;

Name Null? Type

----------------------------------------- -------- ----------------------------

STAFF\_ID NUMBER(12)

STAFF\_NAME VARCHAR2(29)

STAFF\_JOBTYPE VARCHAR2(30)

.

SQL> CREATE TABLE staffsdata (

2 id INT PRIMARY KEY,

3 name VARCHAR(255));

Table created.

SQL> INSERT INTO staffsdata (id,name) VALUES (1,'Osama');

1 row created.

SQL> INSERT INTO staffsdata (id,name) VALUES (2,'WASAM');

1 row created.