**RDBMS MINI PROJECT**

**(SPRINT 1 &2)**

**FOOD ORDERING SYSTEM**

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| Serial No. | Topic Name | Content | Page No. |
| 1. | Introduction | 1.1 Setup checklist for mini project | 3 |
|  |  | 1.2 Instructions | 3 |
| 2. | Problem statement | 2.1 Objective | 4 |
|  |  | 2.2 Abstract of the project | 4 |
|  |  | 2.3 Functional components of the project | 4 |
|  |  | 2.4 Technology used | 5 |
| 3. | Implementation in RDBMS LOT | 3.1 Guidelines on the functionality to be built | 6 |
| 4. | Evaluation | * 1. Evaluation | 7 |

1. **Introduction**

This document outlines a mini project for the RDBMS LOT which will be carried out in 2 sprints . The project is to design the database, reports and queries related to Food Ordering System. This document contains information about the attributes that will be participating in the system and guidelines about reports.

**1.1** **Setup Checklist for Mini Project**

Minimum System Requirements

* Physical Memory (RAM) - 1GB Minimum
* Virtual Memory - Double the amount of RAM
* Disk space - Total 5 GB
* Processor - 550 MHz minimum
* Video Adapter - 256 colors
  1. **Instructions**
* Follow standards while coding
* Create a directory by your name in drive <drive>. In this directory, create a subdirectory Mini Project. Store your Project here.
* You can refer to your course material.
* The total time required to complete this mini project is 25 days for Sprint 1 and 14 days for Sprint 2
* Maintain the code.

1. **Problem Statement**
   1. **Objective:**

Designing the database, developing the queries and basic reports required for Food Ordering System.

* 1. **Abstract of the Project**

This database is used as Food Ordering System. Also some basic reports have to be developed. Since this database is to be used by front end systems, it also requires some backend query and procedure designing.

* 1. **Functional components of the project**

Design the normalized relational database using the following details. You can make appropriate assumptions wherever required. Some of the attributes are given below with the restrictions on data it can contain. Find the required attributes for all the tables and create appropriate constraints on it. (For Ex. Primary key, Foreign key, etc.)

Some of the entities and attributes are as follows:

* CUSTOMER – Mobile number, Customer name, Customer Address, Customer City, Customer Password
* USER\_LOGIN – Login ID, Login Password, User Type
* RETAURANT – Restaurant ID, Restaurant name, Restaurant address, Restaurant city, Restaurant contact number, Cost for 2, details regarding cusines, rating
* MENU – Restaurant ID, Item Name, Item Cost, Quantity Available, Status
* ORDERS – Order number, Restaurant ID, customer Mobile number, order amount, order status, payment status
* TRANSACTION -- Transaction ID, Order number, Customer Mobile Number, Restaurant ID, Order amount, Payment Mode.

Some of the guidelines/protocols are given below:

* Normalize the tables.
* Create additional tables, if necessary.
* One Restaurant can have many items and many restaurants can have same item in their menu.
* Many customer can order same items from same restaurant.
* One restaurant can have multiple customer at the same time.
* One customer can order only from one restaurant at a particular time.
  1. **Technology Used**

Databases:

Oracle 11G Express Edition

1. **Implementation in RDBMS LOT:**
   1. **Guidelines on the functionality to be built:**

**SPRINT 1 -**

1. Create a procedure which gets customer details to register them into the application and restrict the user who is already registered.
2. Create a procedure which gets Login ID and Password from the user and if they match from the data in database, show corresponding path. For example, if the login user is a customer show the restaurant details and ask him to choose from the given restaurants and if the user is admin, give him the access to see into the database and do modifications as required.
3. Create a procedure to take restaurant id as input from user and show the menu of that particular restaurant. Lead the customer to select the items he wants to order and make an order.
4. Create a procedure to take order from the customer and generate a bill with the payment status and delivery status as ‘pending’. Lead the customer to make the payment for his order.
5. Create a procedure to make transaction for the particular order. Make sure that there is only one transaction per order.
6. Create a trigger which gets triggered when there is any insertion in the transaction table for a particular order. This trigger has to update the Payment status and delivery status of the order to ‘paid’ and ‘delivered’ respectively.

**SPRINT 2**

1. Perform performance tuning on the database and the pl/sql code built.
2. Create the entire database in mongo db.
3. **Evaluation and assessment parameters:**
   1. **Evaluation**

* Evaluation will be done at the end of Oracle training
* Total Marks: 100
* Marks Distribution mentioned below.

This Mini project will be done individually. Implement the Software development life cycle for the project using agile methodology and develop code for the respective functionality. Evaluation will be done using online presentation mode, where participant will present their work.

This project shall be evaluated in two parts:

* Marks distribution is for one part of project evaluation (Marks: 90)
* Project Presentation is another part of project evaluation (Marks: 10)