Anudip Foundation

Software Requirement Specification

Name: - Prajval Yogesh Wachpe

Topic: - Real Estate Management System.

Guide: - Mrs. Padmaja Rokade

➢ Introduction

The goals of the Real Estate Management System (REMS) are aligned with addressing the needs and challenges within the real estate domain. Streamline the process of managing and tracking real estate properties, including details such as property types, locations, sizes, and associated amenities. Facilitate seamless transactions by providing features for listing, viewing, and purchasing properties. The system should ensure transparent and secure transactions for buyers, sellers, and agents. Foster user engagement by providing a user-friendly interface for clients, agents, and administrators. The system should support functionalities such as property searches, appointments and feedback. Enhance the efficiency of real estate agents by providing tools for managing appointments, client interactions, and property listings. The system should enable agents to effectively market and sell properties.

➢ Problem Definition

The current real estate management processes are confronted with several challenges that impact the efficiency and effectiveness of operations.

Real Estate Management System.

➢ System Overview

The Real Estate Management System (REMS) is a robust platform designed for efficient property management, featuring secure user authentication and authorization with a role-based access control mechanism. The centralized dashboard provides a quick overview of critical metrics and activities, allowing users to access property listings, vacancy status, pending tasks, and financial summaries rapidly. Core functionalities include detailed property listings with advanced search and filter capabilities, ensuring up-to-date information on property status. Seamless integration of tenant management encompasses tenant profiles, lease agreement management, and rent payment tracking, with automated reminders for lease renewals.

The system facilitates efficient tracking of property maintenance and repairs, including streamlined processes for maintenance requests and comprehensive financial oversight. The communication module enhances interaction with a messaging system and timely notifications. Integration and API capabilities ensure adaptability, allowing connection with external systems, while robust data security measures and compliance with regulations are prioritized.

Mobile accessibility is emphasized, providing a user-friendly interface and key feature access through mobile apps, ultimately optimizing real estate management processes for professionals.

➢ Introduction To MySQL

MySQL is an open-source relational database management system (RDBMS) that uses structured query language (SQL) to manage and manipulate data in a database. It is widely used for various applications, from small web applications to large enterprise systems.

MySQL's key features include:

- Scalability: Capable of handling large amounts of data and concurrent connections.
- Flexibility: Supports various data types and storage engines.
- Performance: Optimized for speed and efficiency.
- Reliability: Known for its stability and robustness.

➢ Installation Of MySQL

MySQL can be installed on various operating systems, including Windows, macOS, and Linux. Here are the general steps to install MySQL:

Windows:

- Download the MySQL installer from the official website.

https://dev.mysql.com/downloads/installer/

- Run the installer and follow the on-screen instructions.
- Choose the installation type (Typical, Complete, or Custom). Recommended Custom.
- Set a root password for the MySQL server.

➢ E-R Diagram (ERD)

An Entity-Relationship Diagram (ERD) is a visual representation of the data model that shows the entities, attributes, relationships between entities, and cardinality. ERDs are commonly used in database design to help developers and stakeholders understand the structure and relationships within a database.

➢ Identify Entities

- Start by identifying the main entities in your system. These are the objects or concepts about which you want to store data.
- Each entity should correspond to a table in your database.

➢ Define Attributes

- For each entity, list the attributes (properties or fields) that describe it.
- These attributes will become columns in the corresponding database table.

➢ Identify Relationships

- Determine how entities are related to each other. There are three types of relationships: one-to-one (1:1), one-to-many (1:N), and many-to-many (N:M).
- Represent these relationships using lines connecting the entities.

➢ Cardinality Notation

Cardinality represents the number of times an entity of an entity set participates in a relationship set. Or we can say that the cardinality of a relationship is the number of tuples (rows) in a relationship.

- Use notation (such as Crow's Foot Notation or Chen Notation) to indicate the cardinality of each relationship.
- Cardinality describes how many instances of one entity are related to how many instances of another entity.
- Common notations include:
- One (1)
- Zero or one (0-1)

➢ Optional

- Include additional information in your ERD, such as primary keys, foreign keys, and constraints (e.g., unique constraints).

➢ Create the Diagram

- Use specialized diagramming software or tools (e.g., Lucidchart, draw.io, or even pen and paper) to create your ERD.

- **Refine and Review:**
  - Review your ERD with stakeholders and team members to ensure it accurately represents the data model and relationships. Make any necessary refinements.

- Let's identify the entities of the Real Estate Management System:
  1. Property
  2. Agent
  3. Buyer
  4. Transaction
  5. Feedback

Attributes And Relationships:

- **Property:**
  - PropertyID (Primary Key)
  - Address
  - Type (e.g., Residential, Commercial)
  - Price
  - SquareFootage
  - Bedrooms
  - Bathrooms
- **Relationships:**
  1. One Property can be managed by one or more Agents (One-to-Many)

- **Agent:**
  - AgentID (Primary Key)
  - FirstName
  - LastName
  - Email
  - Phone
- **Relationships:**
  - One Agent can manage multiple Properties (Many-to-One)
  - One Agent can be associated with multiple Transactions (One-to-Many)

- **Buyer:**
  - BuyerID (Primary Key)
  - FirstName
  - LastName
  - Email
  - Phone
- **Relationships:**
  - One Buyer can be associated with multiple Transactions (One-to-Many)

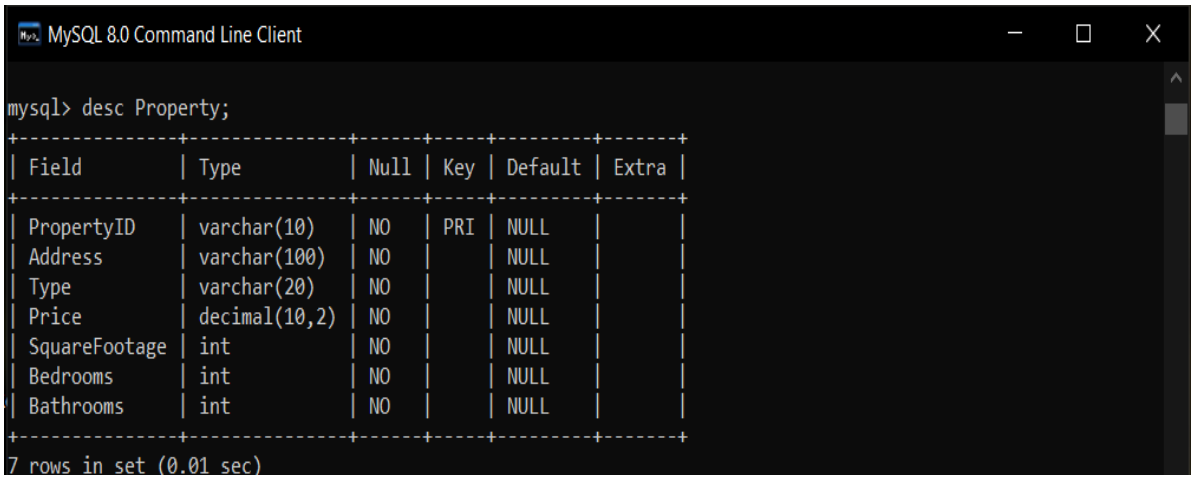
- **Transaction:**
  - TransactionID (Primary Key)
  - PropertyID (Foreign Key)
  - AgentID (Foreign Key)
  - BuyerID (Foreign Key)
  - TransactionDate
  - TransactionAmount
- **Relationships:**
  - One Transaction is associated with one Property, one Agent, and one Buyer (Many-to-One)

- **Inspection:**
  - InspectionID (Primary Key)
  - PropertyID (Foreign Key)
  - InspectorName
  - InspectionDate
  - InspectionResult
- **Relationships:**
  - One Inspection is associated with one Property (Many-to-One)

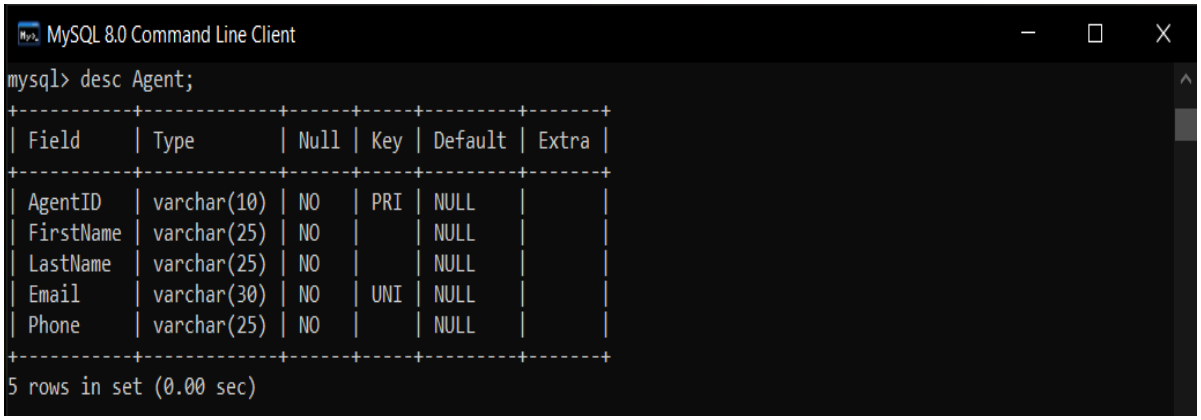
- **Feedback:**
  - FeedbackID (Primary Key)
  - BuyerID (Foreign Key)
  - Date
  - AgentName
  - Feedback
- **Relationships:**
  - One Buyer can provide feedback on multiple occasions (One-to-Many)

Table Structure

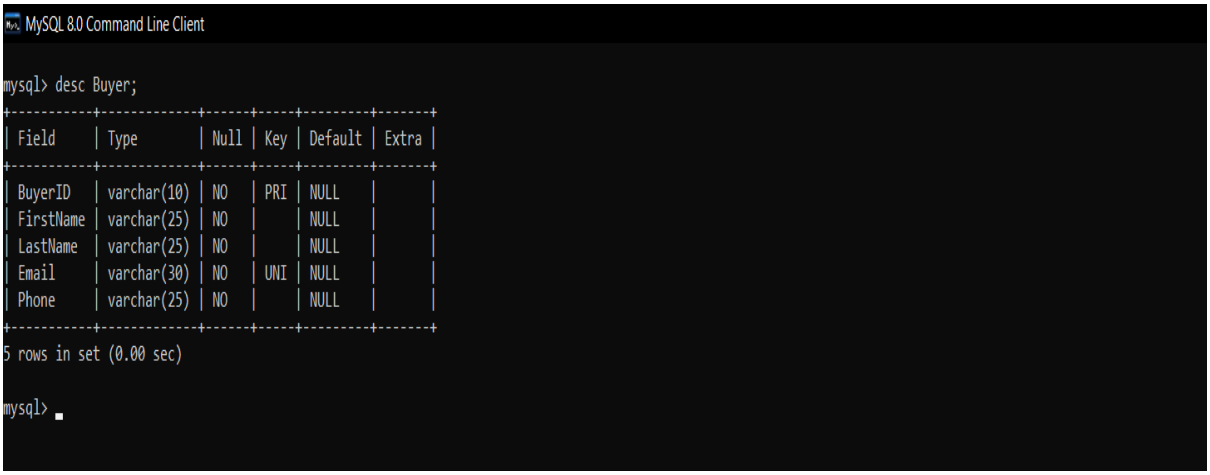
1. Property



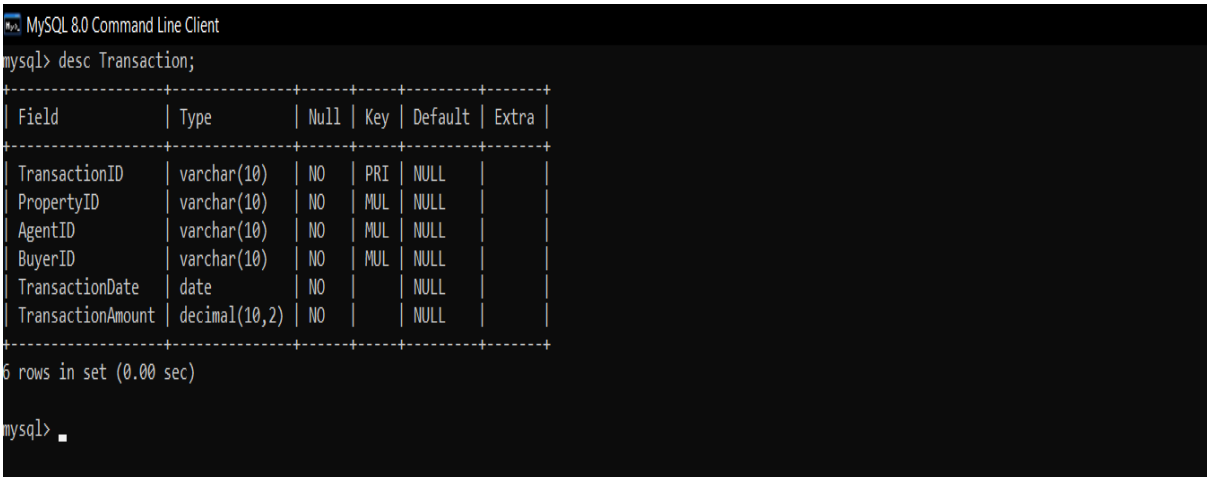
2. Agent



3. Buyer



4. Transaction



5. Inspection

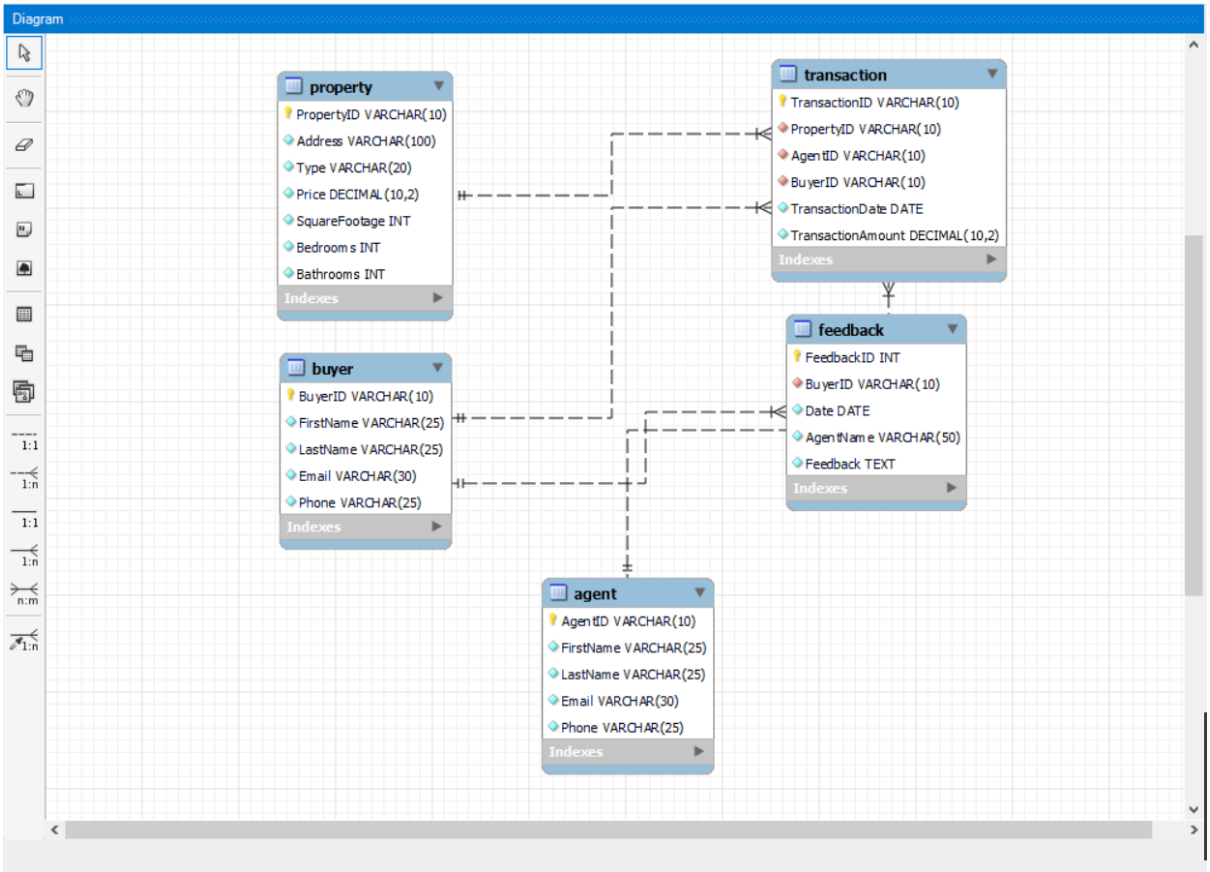
6. Feedback

```
MySQL 8.0 Command Line Client

mysql> desc Feedback;
+------------+------------+------+-----+---------+----------------+
| Field      | Type       | Null | Key | Default | Extra          |
+------------+------------+------+-----+---------+----------------+
| FeedbackID | int        | NO   | PRI | NULL    | auto_increment |
| BuyerID    | varchar(10)| NO   | MUL | NULL    |                |
| Date       | date       | NO   |     | NULL    |                |
| AgentName  | varchar(50)| NO   |     | NULL    |                |
| Feedback   | text       | NO   |     | NULL    |                |
+------------+------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql>
```

➢ Entity Relationship Diagram



➢ In This E-R Diagram

1. Property-Agent Relationship:
   - One Property is managed by one or more Agents (One-to-Many).

2. Agent-Property Relationship:
   - One Agent can manage multiple Properties (Many-to-One).

3. Agent-Transaction Relationship:
   - One Agent can be associated with multiple Transactions (One-to-Many).

4. Transaction-Property Relationship:
   - One Transaction is associated with one Property (Many-to-One).
5. Transaction-Agent Relationship:
   - One Transaction is associated with one Agent (Many-to-One).

6. Transaction-Buyer Relationship:
   - One Transaction is associated with one Buyer (Many-to-One).

7. Inspection-Property Relationship:
   - One Inspection is associated with one Property (Many-to-One).

8. Feedback-Buyer Relationship:
   - One Buyer can provide feedback on multiple occasions (One-to-Many).

➢ Creating And Using Database

Using MySQL server, create a new database for your Real Estate Management System. You can do this with SQL commands or through the graphical interface.

Before performing any operations on a database, you need to select it using the USE statement:

- ***CREATE DATABASE RealEstate;***
- ***USE RealEstate;***

➢ Creating Tables For Each Entity

- **Property :**

```
CREATE TABLE Property (
PropertyID VARCHAR(10) PRIMARY KEY,
Address VARCHAR(100) NOT NULL,
Type VARCHAR(20) NOT NULL,
Price DECIMAL(10, 2) NOT NULL,
SquareFootage INT NOT NULL,
Bedrooms INT NOT NULL,
Bathrooms INT NOT NULL
);
```

- **Agent :**

```
CREATE TABLE Agent (
AgentID VARCHAR(10) PRIMARY KEY,
FirstName VARCHAR(25) NOT NULL,
LastName VARCHAR(25) NOT NULL,
Email VARCHAR(30) UNIQUE NOT NULL,
Phone VARCHAR(25) NOT NULL
);
```

- **Buyer :**

```
CREATE TABLE Buyer (
BuyerID VARCHAR(10) PRIMARY KEY,
FirstName VARCHAR(25) NOT NULL,
LastName VARCHAR(25) NOT NULL,
Email VARCHAR(30) UNIQUE NOT NULL,
Phone VARCHAR(25) NOT NULL
);
```

- **Transaction :**

```
CREATE TABLE Transaction (
TransactionID VARCHAR(10) PRIMARY KEY,
PropertyID VARCHAR(10) NOT NULL,
AgentID VARCHAR(10) NOT NULL,
BuyerID VARCHAR(10) NOT NULL,
TransactionDate DATE NOT NULL,
TransactionAmount DECIMAL(10, 2) NOT NULL,
 FOREIGN KEY (PropertyID) REFERENCES Property(PropertyID),
FOREIGN KEY (AgentID) REFERENCES Agent(AgentID),
FOREIGN KEY (BuyerID) REFERENCES Buyer(BuyerID) );
```

- **Inspection :**

```
CREATE TABLE Inspection (
InspectionID VARCHAR(10) PRIMARY KEY,
PropertyID VARCHAR(10) NOT NULL,
 InspectorName VARCHAR(50) NOT NULL,
 InspectionDate DATE NOT NULL,
InspectionResult VARCHAR(50) NOT NULL,
FOREIGN KEY (PropertyID) REFERENCES Property(PropertyID)
);
```

- **Feedback :**

```
CREATE TABLE Feedback (
 FeedbackID INT AUTO_INCREMENT PRIMARY KEY,
BuyerID VARCHAR(10) NOT NULL,
 Date DATE NOT NULL,
AgentName VARCHAR(50) NOT NULL,
Feedback TEXT NOT NULL,
FOREIGN KEY (BuyerID) REFERENCES Buyer(BuyerID)
);
```

➢ Hardware Interface

Hardware devices which can be used for this project are the laptops as well as computers. Computers/PC's as well as laptops are used in order to access the web browsers.

Hardware specifications are listed below:

• Processor: Intel core 5

• Ram size: 8GB / 4GB

• Hard disk capacity: 500 GB/ 1TB

➢ Software Interface

• IDE: MySQL Command Line

• Language: SQL

• Documentation: Ms-Office

Platform: - Linux/windows/ubuntu

➢ Constraint

- The real estate management system should seamlessly integrate with MySQL, using its command line for database management. All data, including properties, transactions, users, and relevant information, must be stored in the MySQL database.
- The system, following open-source principles, allows developers to freely access, modify, and distribute the source code. Leveraging MySQL's scalability features is essential to accommodate the growing volume of real estate data, users, and transactions.
- The command line interactions should be user-friendly, with easy-to-use commands or scripts for non-technical users. Robust security measures must be in place to protect sensitive data, and data integrity constraints, including foreign key relationships, should be enforced.
- The system should allow customization and extension of data models within MySQL to meet specific business requirements. Utilizing MySQL's features like stored procedures, triggers, and views enhances the system's functionality.
- Regular automated backups are crucial to prevent data loss, and the system should offer efficient data restoration using MySQL command line tool

➢ Safety Requirement

- Implement robust authentication for authorized access to sensitive real estate data.
- Securely store user passwords using industry-standard encryption.
- Encrypt all system communication with external entities using secure protocols like HTTPS.
- Utilize Role-Based Access Control (RBAC) to limit user access based on roles.
- Maintain an audit trail for user activities, system events, and critical data changes.
- Perform regular automated database backups for data recovery in emergencies.
- Employ checksums or hash functions to prevent data corruption during storage or transmission.
- Define procedures for an emergency shutdown in the event of a security breach or system malfunction.
- Implement mechanisms for continuous system availability, even during hardware failures.
- Ensure compliance with relevant privacy regulations, such as GDPR or HIPAA.
- Develop and document an incident response plan, detailing steps for security incidents.

➢ Team Structure

The Real Estate Management System team at Anudip Foundation consists of a single dedicated member, Prajval Yogesh Wachpe, serving as the Project Lead and Developer. In this role, you bear the responsibility for the entire development and implementation of the system. Under the supervision of Mrs. Padmaja Rokade, Project Assistant, you collaborate closely to ensure project guidance and regular updates. Task allocation, encompassing system design, development, testing, and deployment, is entirely managed by you. This streamlined structure reflects the unique nature of a solo endeavor, highlighting your central role as the key contributor for the Real Estate Management System project at Anudip Foundation. Regular meetings with Mrs. Padmaja Rokade facilitate project reviews, discussions, and progress updates, ensuring effective communication and guidance throughout the development process.

➢ Conclusion

In conclusion, the Real Estate Management System represents a comprehensive and efficient solution for the dynamic and intricate challenges within the real estate domain. Developed under the dedicated leadership and expertise of the project lead, Prajval Yogesh Wachpe, at Anudip Foundation, the system aims to streamline and enhance the management of real estate data. By leveraging robust authentication mechanisms, secure communication protocols, and meticulous access controls, the system ensures the safeguarding of sensitive information. Regular meetings and collaboration with the Project Assistant, Mrs. Padmaja Rokade, provide valuable guidance and support throughout the development process. The incorporation of features such as automated backups, data integrity checks, and an audit trail underscores the commitment to data security and system reliability. As this project progresses, the Real Estate Management System is poised to make a significant impact, offering a user-friendly interface and adhering to the highest standards of privacy and compliance. Overall, the system embodies a forward-looking approach to real estate management, providing a scalable, secure, and adaptable solution for the evolving needs of the real estate industry.