Property Management System

Int M.Sc./III Year DAS/V Semester

22CSC381 - Database Management Systems Lab

Project Review

Name	Roll No
Hari Prasath M	CB.PS.I5DAS22119
Rajeswari Alagappan	CB.PS.I5DAS22139
Tejesshvi V S	CB.PS.I5DAS22166

Amrita School of Physical Sciences, Coimbatore

Department of Mathematics

2023-2024 Odd Semester

Table of Contents

Abstract

The **Property Management System (PMS)** is designed to facilitate secure and efficient property transactions including buying and selling. The system caters to a global market, involving multiple people such as **buyers**, **sellers**, and **administrators**, each with distinct roles. The system has transparency and security and ensures that all property listings are validated and cross-referenced with government databases to maintain authenticity and legality.

The PMS is built around several key features. At its core, it manages the entire process of a property transaction, from listing properties and handling user inquiries to finalizing payments. Sellers can upload property details such as location, images, descriptions, and proof of ownership, which are verified by admins. Only properties that pass this verification are made available for potential buyers, ensuring that all transactions are secure and transparent.

A systematic **enquiry system** allows buyers and sellers to communicate and negotiate directly within the platform. All conversations and enquiries are tracked and managed by the admin, who is responsible for approving property transactions. Payments made for property sales or leases are monitored by the admin to ensure authenticity, adding another layer of security to the process. In addition, all property registrations are completed entirely online, reducing the need for physical visits to regional offices.

The **user roles** are clearly defined: sellers create property listings, buyers search and make inquiries, and admins oversee the platform's complete operations, including **user verification** and transaction approvals. The admin also ensures that users comply with the system's guidelines by validating their credentials before granting them access to the platform. This verification process safeguards the integrity of the user base.

A key functionality of the system is its **payment processing** module, which ensures that all monetary transactions are tracked and recorded. The admin's role in overseeing these payments ensures a smooth and secure financial process for both buyers and sellers.

Overall, this project delivers a diverse platform for managing real estate transactions in a secure, user-friendly, and efficient manner. With cross-referenced government verification, proof of ownership checks, and purely online registrations, the Property Management System ensures a high level of trust and transparency in the property market. And ensures the property is legal and verified.

Business Rules

1. **Property Validation: P**roperty Validation has a set of rules to be followed to validate a property thoroughly. The property details uploaded by the seller is checked property and is also cross verified with government databases to maintain the authenticity. This business rule is a crucial step to be followed.

2. **User roles:** The system classifies user roles into 3 categories – Admin, Seller, and buyer. Each User role has a specific role and responsibilities within their platforms. Admins takes over control on system management, Sellers handle the sales and Buyers focus on purchasing properties.

3. **Admin Approval:** The admin plays a critical role in managing the entire system management. He oversees the property related activities like transaction system, sales and purchases within the system. He is responsible for managing property listings that include new properties, excluding sold properties and accurate information of the available property.

4. **Proof of ownership:** Sellers must submit their property-related documents to the admin in order to prove their ownership of this property. The admin is responsible for reviewing and approving the documentation proof given by the seller and makes sure that property becomes visible for the potential buyers.

5. **Payment processing:** Payment Processes are taken care of by the admin and ensure all the payment transaction has proof. Here payments are securely processed, tracked, and accurately recorded within a system to ensure authenticity.

6. **Online Registration:** All Property registration is done online without the need to visit the regional offices. This makes the process faster and more convenient, allowing users to submit documents and register properties from any location.

7. **User Verification:** All users must have their own credentials before they can access the system. This ensures that only trusted individuals can use the platforms with their own credentials.

Preview of Project

The urgent demand for safe, easy, and transparent property transactions is of utmost importance in the digital world. As Data Science students, we use our skills in data analysis, validation, and user interaction to create a platform that makes property management simpler. By adding real-time data verifications and smart search tools, we aim to improve the user experience, prevent fraud, and help people make better decisions. This will make property transactions easier and more trustworthy in this modern, online-focused environment.

Tools Used:

1. My SQL.
2. Python.

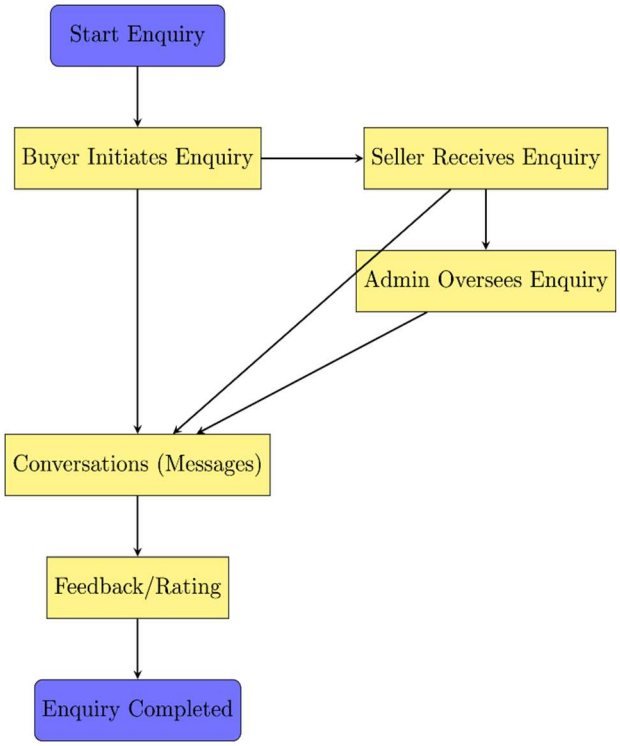
Tables:

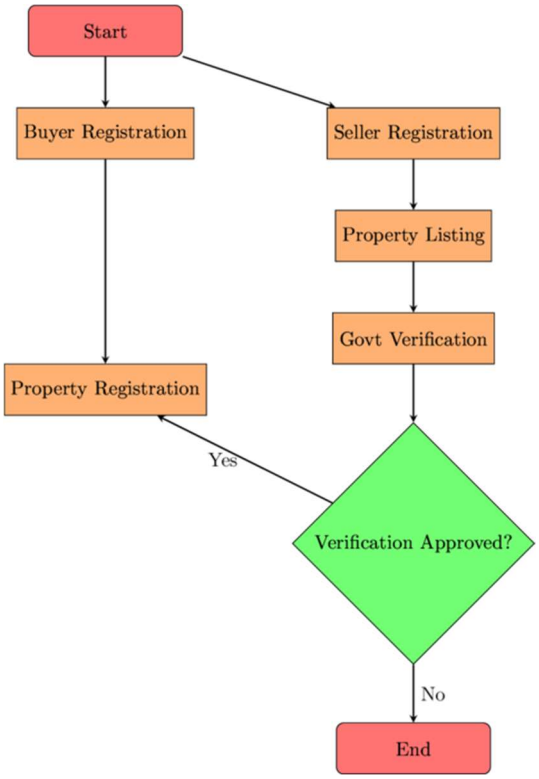
1. Admin.
2. Buyer.
3. Seller.
4. Property.
5. Property Photos.
6. Payment.
7. Registrations.
8. Registration Offices
9. Enquiry
10. Conversations.
11. Feedback
12. Govt Verification

Output:

1. Buyers to be provided with search and filter options.
2. Sellers with an option to upload the property details and images with proof which admin would verify.
3. Property details report.
4. Registration report.
5. Payment reports.

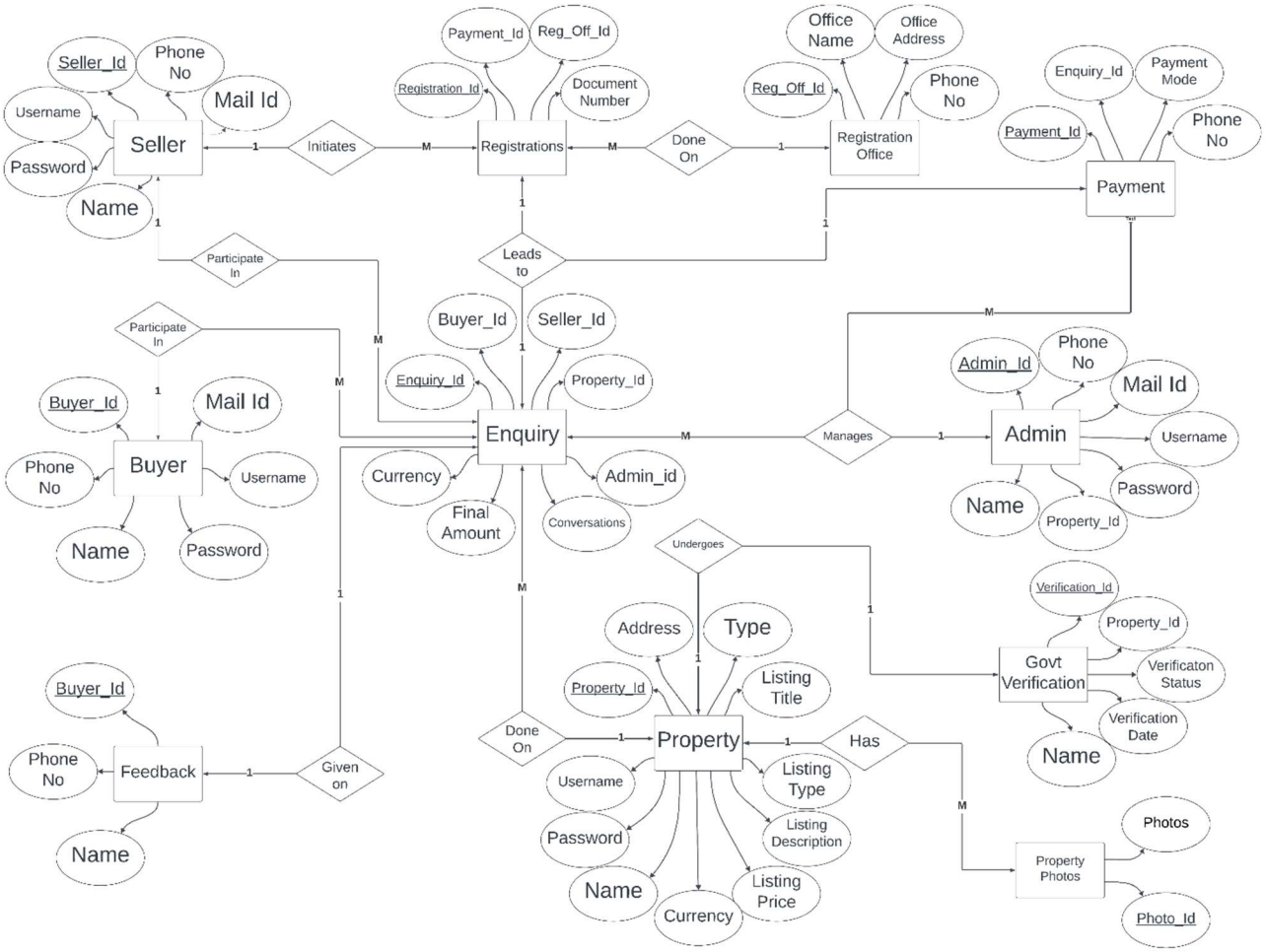
Project Design



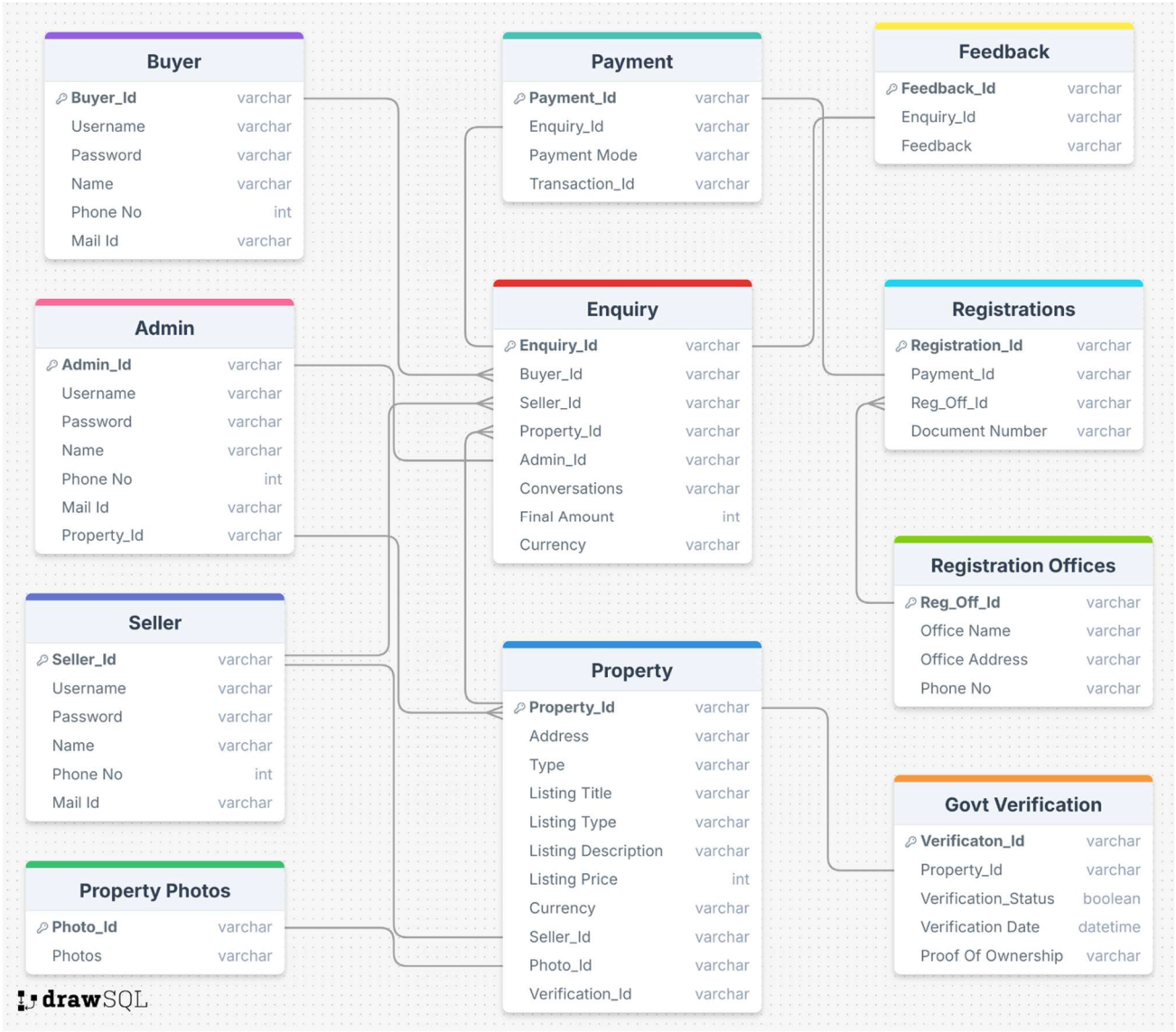


Back End Design

ER Diagram:



Schema Diagram:



Normalization

Functional Dependencies for all the tables:

Buyer:

Buyer_Id → Username, Password, Name, Phone_No, Mail_Id

Username → Buyer_Id, Password, Name, Phone_No, Mail_Id

Seller:

Seller_Id → Username, Password, Name, Phone_No, Mail_Id

Username → Seller_Id, Password, Name, Phone_No, Mail_Id

Admin:

Admin_Id → Username, Password, Name, Phone_No, Mail_Id, Property_Id

Username → Admin_Id, Password, Name, Phone_No, Mail_Id, Property_Id

Property:

Property_Id → Address, Type, Listing_Title, Listing_Type, Listing_Description, Listing_Price, Currency, Seller_Id, Photo_Id, Verification_Id

Property_Photos:

Photo_Id → Photos

Payment:

Payment_Id → Enquiry_Id, Payment_Mode, Transaction_Id

Feedback:

Feedback_Id → Enquiry_Id, Feedback, Rating

Enquiry:

Enquiry_Id → Buyer_Id, Seller_Id, Property_Id, Admin_Id, Conversations, Final_Amount, Currency

Registrations:

Registration_Id → Payment_Id, Reg_Off_Id, Document_Number

Registration_Offices:

Reg_Off_Id → Office_Name, Office_Address, Phone_No

Govt_Verification:

Verification_Id → Property_Id, Verification_Status, Verification_Date, Proof_Of_Ownership

First-Normal Form:

To satisfy the First Normal Form, the relations must:

- Have Atomic values.
- Have Unique Rows.
- Have a Primary Key.

Admin table doesn't satisfy the 1st condition as one Admin_Id can have multiple entries of Property_Id as one Admin can manage multiple Properties. To solve this, we can remove the Property_Id attribute from the admin table and instead, we can link the Property table to the admin table using Admin_id as the Foreign Key.

The Modified FDs will be:

Admin_Id → Username, Password, Name, Phone_No, Mail_Id Username → Admin_Id, Password, Name, Phone_No, Mail_Id

Property_Id → Address, Type, Listing_Title, Listing_Type, Listing_Description, Listing_Price, Currency, Seller_Id, Property_Id, Photo_Id, Verification_Id.

Property table doesn't satisfy the 1st condition as one Property can have multiple Photos and hence, multiple Photo_Ids. To solve this, we can remove Photo_Id from the Property table and instead link the Property_Photos table to the Property table using Property_id as the foreign key.

The Modified FDs will be:

Property_Id → Address, Type, Listing_Title, Listing_Type, Listing_Description, Listing_Price, Currency, Seller_Id, Property_Id, Verification_Id.

Property_Id, Photo_Id → Photo

Enquiry table doesn't satisfy the 1st condition as there can be multiple conversations i.e. messages in a single enquiry. To solve this, we can remove the conversations attribute from the enquiry table and instead create a new table called Conversations and link it to the Enquiry table using Enquiry_Id as the foreign key.

The Modified FDs will be:

Enquiry_Id → Buyer_Id, Seller_Id, Property_Id, Admin_Id, Final_Amount, Currency.

Conversations:

Enquiry_Id, Message_Id → Buyer_Id, Seller_Id, Admin_Id, Message, Time_Stamp

Except **Admin** table, **Property** table and **Enquiry** table, all other tables satisfy, all the 3 conditions. So, 1NF is satisfied by all the tables

Second-Normal Form:

To satisfy the second normal form, the relations must:

- Satisfy 1NF.
- Not have Partial Dependencies.

Property_Photos:

FD => Property_Id, Photo_Id → Photo.

Candidate Primary Key => (Property_Id, Photo_Id)

None of the attributes depend partially on any part of the Candidate Primary Key.

So, 2NF is satisfied.

Conversations:

FD => Enquiry_Id, Message_Id → Buyer_Id, Seller_Id, Admin_Id, Message, Time_Stamp.

Candidate Primary Key => (Enquiry_Id, Sender_Id)

None of the attributes depend partially on any part of the Candidate Primary Key.

So, 2NF is satisfied.

Except **Property_Photos** table and **Conversations** table, all other tables only have a Singular Primary key. Since there aren't any Composite Primary Keys, we can conclude that, there are no Partial dependencies and thus 2NF is satisfied by all the tables.

Third-Normal Form:

To satisfy the Third Normal Form, the relations must:

- Satisfy 2NF.
- Not have Transitive Dependencies.

Buyer:

Buyer_Id → Username

Username → Buyer_Id, Password, Name, Phone_No, Mail_Id

Here, we have a transitive dependency between Buyer_Id and Username. As Username and Buyer_Id both are Unique and it can act as a primary key individually, we can remove the Username attribute to solve the Transitive Dependency as well as remove the redundancy in the table.

So, FD => Buyer_Id → Password, Name, Phone_No, Mail_Id

Seller:

Seller_Id → Username

Username → Seller_Id, Password, Name, Phone_No, Mail_Id

Here, we have a transitive dependency between Seller_Id and Username. As Username and Seller_Id both are unique and they can individually act as a primary key for the given table, we can remove the Username attribute to solve the Transitive Dependency as well as remove the redundancy in the table.

So, FD => Seller_Id → Password, Name, Phone_No, Mail_Id

Admin:

Admin_Id → Username

Username → Admin_Id, Password, Name, Phone_No, Mail_Id

Here, we have a transitive dependency between Admin_Id and Username. As Username and Admin_Id both are unique and they can individually act as a primary key for the given table, we can remove the Username attribute to solve the Transitive Dependency as well as remove the redundancy in the table.

So, FD => Admin_Id → Password, Name, Phone_No, Mail_Id

Other than **Buyer**, **Seller** and **Admin** tables, all other relations have only one Functional Dependency. Thus, there aren't any transitive dependency. So, we conclude that 3NF is satisfied by all tables.

Boyce-Codd Normal Form:

To satisfy the Boyce-Codd Normal Form, the relations must:

- Satisfy 3NF.
- For every Functional Dependency, the LHS must be a candidate key.

The functional dependencies of all the relations, either have a singular primary key or composite primary key on its LHS, satisfying the conditions for BCNF.

Thus, we can conclude that all tables are Fully Normalized till BCNF.

Conclusion

In conclusion, the Property Management System stands as a robust and comprehensive solution designed to revolutionize the real estate market. By integrating stringent verification processes with a user-friendly interface, the system ensures that every property transaction—from listing to final payment—is conducted securely and transparently. Key features such as government-verified property details, clearly defined roles for buyers, sellers, and administrators, and an efficient online registration process collectively contribute to a trustworthy digital environment.

From a technical perspective, the project demonstrates meticulous attention to detail in both design and implementation. The underlying database has been carefully structured and normalized up to Boyce-Codd Normal Form, ensuring data integrity, reducing redundancy, and enhancing system performance. This rigorous approach not only improves the reliability of the system but also lays a solid foundation for future scalability and enhancements.

Overall, this project reflects our commitment to creating innovative, secure, and efficient solutions in property management. We invite users and collaborators to explore, contribute, and provide feedback, as we continue to refine the system to meet the evolving needs of the digital property marketplace.