RESIDENTIAL MANAGEMENT SYSTEM

DataBaes				
Team Members	Aniket Bhore	Prathmesh Pardeshi	Satadru Debnath	Vedant Deshpande
NU Email ID	bhore.a@northeastern.edu	pardeshi.pr@northeastern.edu	debnath.s@northeastern.edu	deshpande.ve@northeastern.edu

Table of Contents

1. OVERVIEW

According to a survey done by U.S. Department of Housing and Urban Development, we are building more houses now than in the last ten years! This quick rise in building calls for centralized management of apartments and rental properties to be made more efficient. Currently, property managers are using various manual and disparate systems to manage their properties, leading to inefficiencies and errors. A **Residential Management System** is needed to efficiently manage the day-to-day operations of a residential complex, such as apartments, condominiums, or gated communities.

A Residential Management System (RMS) is a software solution for managing the day-to-day operations of a residential complex or community. It typically includes modules for property and unit management, resident information management, maintenance management, and financial management. It allows landlords and property managers to easily track and update this information, as well as generate reports and communicate with tenants. Additionally, it can also be used to automate tasks such as rent collection and maintenance requests.

In this project we aim to create a Residential Management System with database support. This project is a part of our course DAMG6210, here we solved the problem of manual entry of data in apartments by storing data in Oracle database.

2. PROBLEM STATEMENT

The goal is to design and develop a comprehensive Residential Management System using Oracle database that will enable property managers to automate and streamline their operations, including tenant and lease management, rent and bill collection, maintenance, and repair tracking.

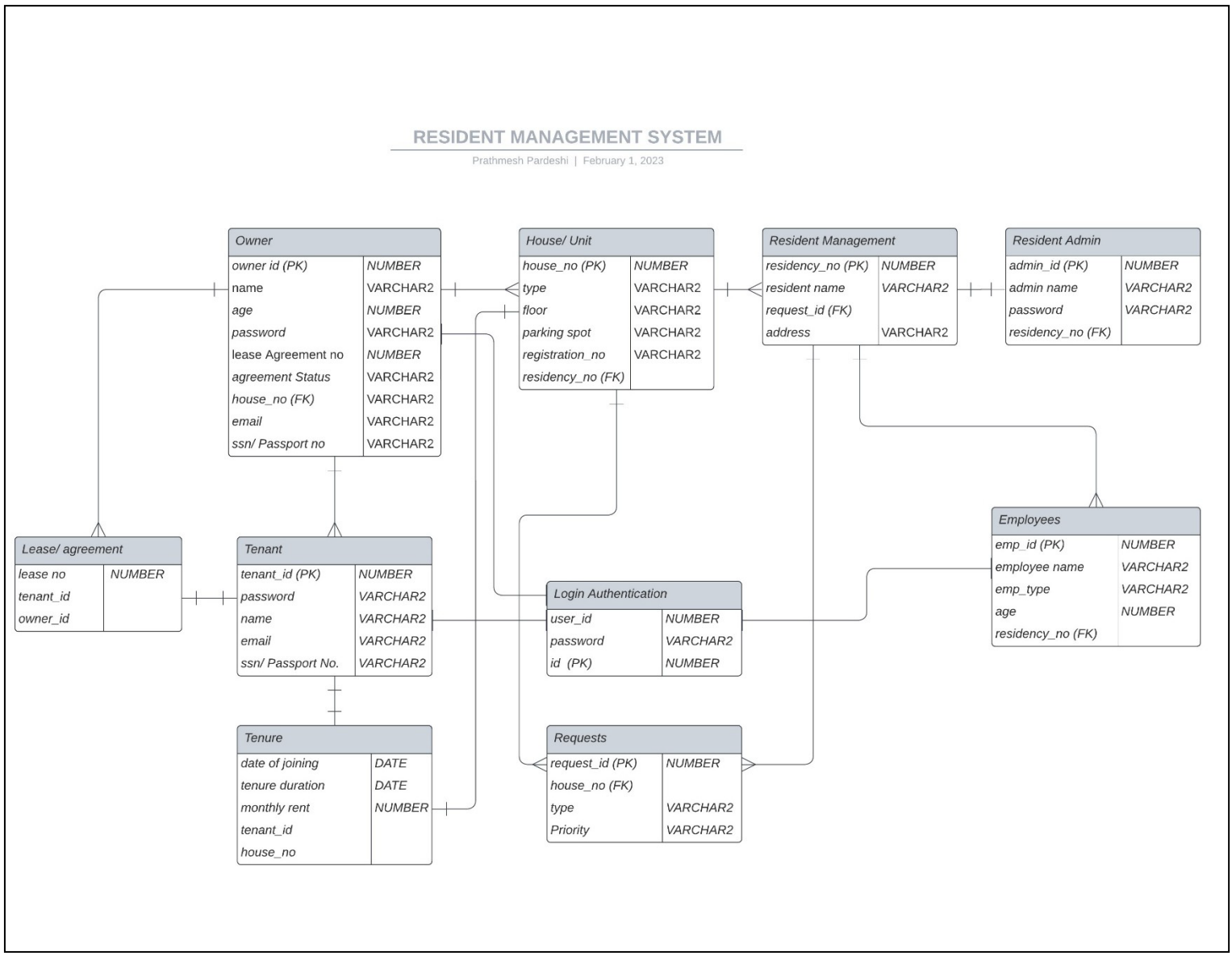
3. BUSINESS PROBLEMS ADDRESSED

1. Provide an easy-to-use platform for landlords and tenants to manage lease agreements, payments, maintenance requests, and security deposits online.
2. Provide a centralized database for lease agreements and tenant information to streamline the process of creating and renewing leases, as well as managing tenant turnover.
3. Enable tenants to submit maintenance requests and track their progress online, allowing for efficient management of repairs and upkeep.
4. Provide a system for landlords to track and manage security deposits, including tracking the amount and reason for deductions, and returning the deposit at the end of the lease term.
5. Generate automated rent invoices and reminders to ensure timely payments and avoid late fees, while also providing a platform for tenants to easily make payments online.
6. Allow for the tracking and management of expenses related to property maintenance and repairs, to help landlords better understand the costs associated with owning and managing their properties.

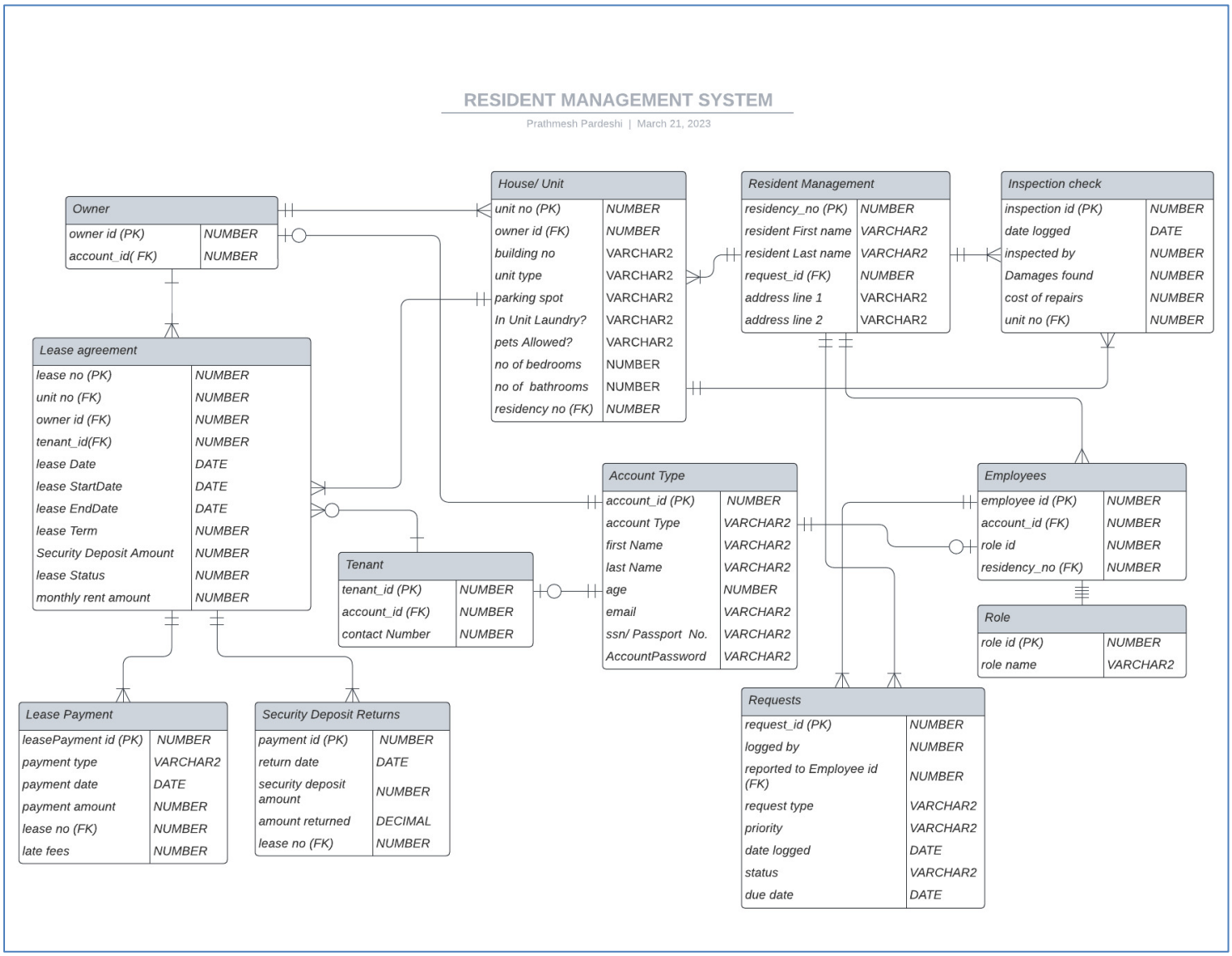
4. ENTITY RELATIONSHIP DIAGRAM

When it comes to designing a database, Entity Relationship (ER) diagrams play a big role. Based on the business idea, we have drafted a representation of the ER model containing tables with relations.

Stage 1 ER diagram -



Final ER Diagram -



5. BUSINESS RULES

There are 4 user types for this Database Management system – Admin, Owner, Tenant, Employee. Following are the Business rules

• Each property may have one or more lease agreements.
• Each lease agreement is associated with only one property.
• Each lease agreement is associated with only one tenant.
• Each tenant may have zero or more lease agreements.
• Each lease agreement may have one or more lease payments.
• Each lease payment is associated with only one lease agreement.
• Each lease payment is associated with only one tenant.
• Each maintenance request is associated with only one property.
• Each maintenance request may be associated with only one employee.
• Each employee may be associated with one or more maintenance requests.
• Each lease agreement may have one security deposit associated with it.
• Each refund request for a security deposit is associated with only one lease agreement.
• Each lease agreement must specify the lease start and end dates, as well as the monthly rent amount

• Late fees may be assessed for any rent payments received after the due date specified in the lease agreement.
• Employees must complete the maintenance requests within a reasonable timeframe and update the status of the request in the system.
• Tenants are responsible for maintaining the property in a clean and safe condition, and any damages caused by the tenant or their guests will be deducted from the security deposit.

6. TABLES

Based on the business, we have created an idea of how many tables this system will consist of. These tables will define the database through column name, data type, constraints, and the description of those entities

1) Account Table

Column Name	Data Type	Constraints
Account id	**NUMBER**	**NOT NULL UNIQUE**
Account type	**VARCHAR2 (50)**	**NOT NULL**
First Name	**VARCHAR2 (50)**	**NOT NULL**
Last Name	**VARCHAR2 (20)**	**NOT NULL**
Age	**NUMBER (20)**	**NOT NULL UNIQUE**
Email	**VARCHAR2 (20)**	**NOT NULL**
ssn/ passport no.	**VARCHAR2 (20)**	**NOT NULL UNIQUE**
Account Password	**VARCHAR2 (20)**	**NOT NULL UNIQUE**

2) Tenant Table

Column Name	Data Type	Constraints
tenant_id	**NUMBER**	**NOT NULL UNIQUE**
Contact Numbwe	**NUMBER**	**NOT NULL UNIQUE**

3) House Table

Column Name	Data Type	Constraints
house_no	**NUMBER**	**NOT NULL UNIQUE**
Building no	**VARCHAR2 (20)**	**NOT NULL**
Unit type	**VARCHAR2 (50)**	**NOT NULL**
parking spot	**VARCHAR2 (20)**	**NOT NULL**
Is Unit Laundery	**VARCHAR2 (20)**	**NOT NULL UNIQUE**
Pets allowed?	**VARCHAR2 (20)**	**NOT NULL UNIQUE**
No of Bedrooms	**NUMBER**	**NOT NULL**
No of Bathrooms	**NUMBER**	**NOT NULL**
Residency no	**NUMBER**	**NOT NULL UNIQUE**
Current Tenant	**NUMBER**	**NOT NULL**

4) Resident Management Table

Column Name	Data Type	Constraints
residency_no	**NUMBER**	**NOT NULL UNIQUE**
resident first name	**VARCHAR2 (20)**	**NOT NULL**
resident first name	**VARCHAR2 (50)**	**NOT NULL**
Address 1	**VARCHAR2 (20)**	**NOT NULL**
Address 2	**VARCHAR2 (20)**	**NOT NULL**
Request id	**NUMBER**	**NOT NULL UNIQUE**

5) Inspection Check

Column Name	Data Type	Constraints
inspection_id	**NUMBER**	**NOT NULL UNIQUE**
Date logged	**DATE**	**NOT NULL**
Inspected by	**NUMBER**	**NOT NULL UNIQUE**
Damages found	**NUMBER**	**NOT NULL UNIQUE**
Cost of repairs	**NUMBER**	**NOT NULL**
Unit no	**NUMBER**	**NOT NULL UNIQUE**

6) Lease Agreement Table

Column Name	Data Type	Constraints
lease no	**NUMBER**	**NOT NULL UNIQUE**
tenant_id	**NUMBER**	**NOT NULL UNIQUE**
Unit no	**NUMBER**	**NOT NULL UNIQUE**
Building no	**NUMBER**	**NOT NULL**
Residency no	**NUMBER**	**NOT NULL**
Lease Date	**DATE**	**NOT NULL**
Lease Start Date	**DATE**	**NOT NULL**
Lease End Date	**DATE**	**NOT NULL**
Lease Term	**NUMBER**	**NOT NULL**
Security Deposit	**NUMBER**	**NOT NULL**
Lease Status	**NUMBER**	**NOT NULL**
Monthly rent amount	**NUMBER**	**NOT NULL UNIQUE**

7) Security Deposit Table

Column Name	Data Type	Constraints
Payment id	**NUMBER**	**NOT NULL**
Return date	**DATE**	**NOT NULL**
Security deposit amount	**NUMBER**	**NOT NULL**
Amount returned	**NUMBER**	**NOT NULL UNIQUE**
Lease no	**NUMBER**	**NOT NULL UNIQUE**

8) Requests Table

Column Name	Data Type	Constraints
request_id	**NUMBER**	**NOT NULL UNIQUE**
Logged by	**NUMBER**	**NOT NULL UNIQUE**
Reported to Emplyee id	**NUMBER**	**NOT NULL**
Request type	**VARCHAR2 (20)**	**NOT NULL**
Priority	**VARCHAR2 (20)**	**NOT NULL**
Date logged	**DATE**	**NOT NULL**
Status	**VARCHAR2 (20)**	**NOT NULL**
Due date	**VARCHAR2 (20)**	**NOT NULL**

9) Employees Table

Column Name	Data Type	Constraints
emp_id	**NUMBER**	**NOT NULL UNIQUE**
Role id	**VARCHAR2 (20)**	**NOT NULL**
Residency_no	**NUMBER**	**NOT NULL UNIQUE**

10) Role Table

Column Name	Data Type	Constraints
Role id	**NUMBER**	**NOT NULL UNIQUE**
Role Name	**VARCHAR2 (20)**	**NOT NULL**

11) Owner Table

Column Name	Data Type	Constraints
Owner id	**NUMBER**	**NOT NULL UNIQUE**

7. VIEWS

1. Lease_Agreement_View :

- This view displays information about lease agreements, including the lease ID, tenant's first and last name, property address, lease start and end dates, monthly rent, and security deposit.

- This view can be used to easily see all the lease agreements in the system and the details associated with each one.

2. Lease_Payment_View:

- This view displays information about lease payments, including the payment ID, tenant's first and last name, property address, payment amount, and payment date.

- This view can be used to track lease payments, see when payments were made, and ensure that tenants are paying their rent on time.

3. Maintenance_Request_View:

- This view displays information about maintenance requests, including the maintenance request ID, tenant's first and last name, property address, and a description of the request.

- This view can be used to track maintenance requests and ensure that they are addressed in a timely manner.

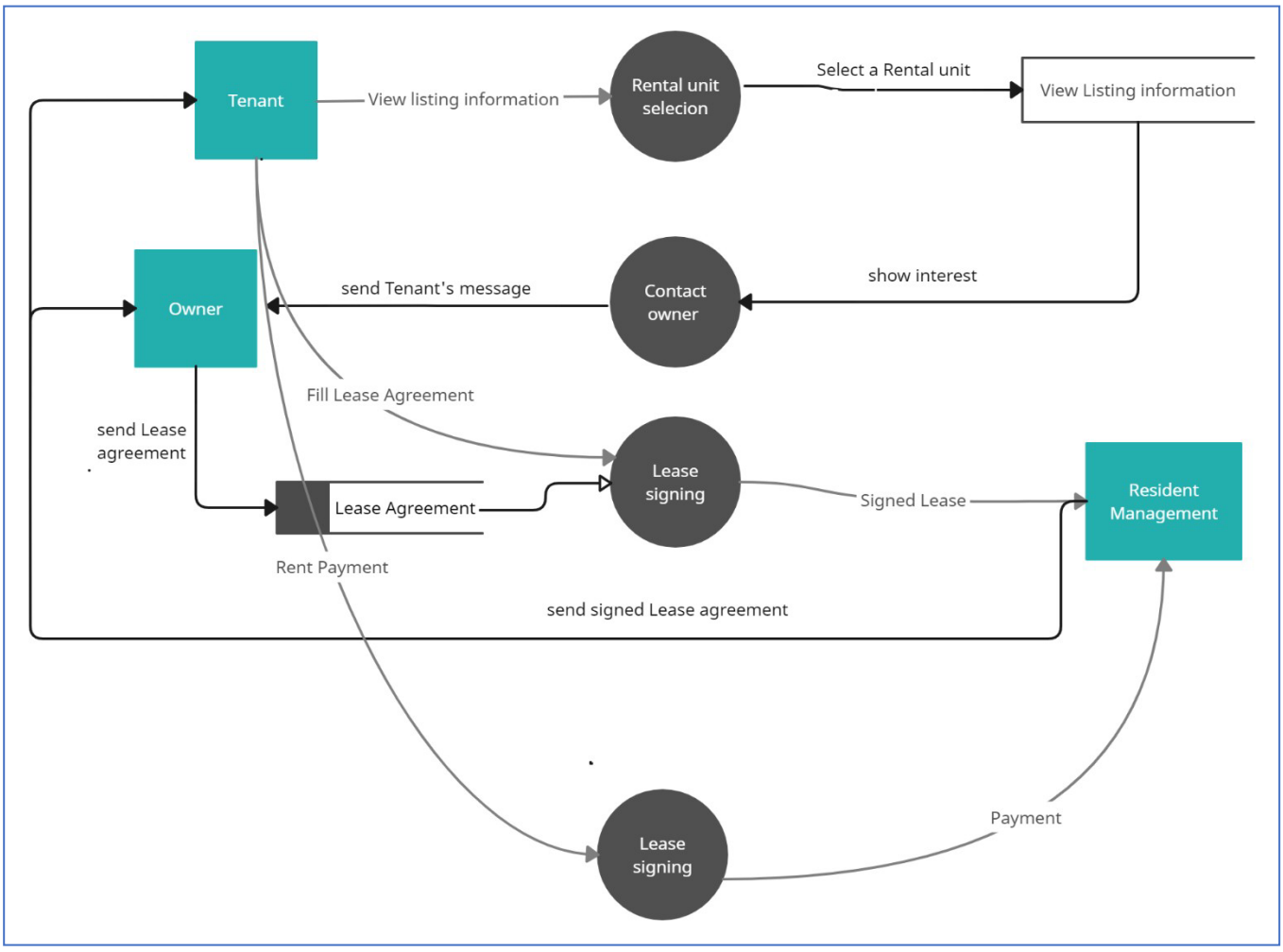
4. Unit_Inspection_View:

- This view tracks the results of unit inspections, including any damages found and the actions taken to repair them. It includes information about the date of the inspection, the unit inspected, and the employee who performed the inspection.

- This view presents this information in a way that's easier to analyze and use in reports. The damages found and repairs made columns provide additional information about the condition of the unit and any necessary maintenance work.

8. DATA FLOW DIAGRAM

Lease Agreement View



9. SECURITY CONSTRAINTS

To ensure data security for a residential management system that keeps track of lease agreements, lease payments, maintenance requests, assignment of maintenance requests to employees, and current tenants, the following SQL data security constraints can be implemented for each role –

1. Tenant:

a. Read access to their own lease agreement and lease payment information.

b. Write access to their own maintenance request information.

c. No access to other tenant's information.

d. No access to owner or employee information.

2. Owner:

a. Read access to their own property information, lease agreement, lease payment information, and maintenance request information.

b. Write access to their own property information and maintenance request information.

c. No access to other owner's information.

d. No access to tenant or employee information.

3. Resident Admin:

a. Read access to all tenant and owner information, lease agreements, lease payment information, maintenance request information, and property information.

b. Write access to tenant and owner information, lease agreements, lease payment information, maintenance request information, and property information.

c. No access to employee personal information.

4. Employee:

a. Read access to their own assigned maintenance requests.

b. Write access to their own assigned maintenance requests.

c. No access to tenant, owner, or resident admin information.