WORKSHOP #1

SANTIAGO ZAMUDIO DIAZ – 20231020128

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

BASEDATA FUNDAMENTALS
GROUP 020-82
CARLOS ANDRÉS SIERRA VIRGUEZ

2024

User Stories:

1. As a resident, I want to reserve common areas so that I can ensure availability for personal events. So what it allows users to plan events without conflicts over shared spaces.
2. As a resident, I want to receive maintenance notifications so that I am aware of any repairs or issues affecting my apartment or the building. So, what ensures residents stay informed about necessary maintenance and avoid inconveniences.
3. As a resident, I want to track my payment history so that I can stay on top of my bills and avoid late fees. So, what helps residents manage their finances and ensure they do not miss any payments.
4. As a resident, I want to receive notification when a visitor arrives so that I can be informed of their presence and ensure a secure living environment. So, what enhances resident security and awareness of visitor activity.

There are not 15 user stories because these were the four different responses given by the 15 students; therefore, I do not see the need to repeat the user stories.

Technical and design considerations/desicions:

For the technical and design considerations, I have created a concise list of potential functionalities that the application could include, considering Considering the proposed design, I have created a concise list of potential functionalities that the application could include for the technical and design considerations.

- **Billing and Payments Management**:

  The Bills table handles resident billing, including the issuance date, due date, amount, and payment status. This data is directly linked to the Type - Bill table, which helps classify and generate reports on different types of bills.

  Each bill is directly associated with a specific apartment and is automatically updated when payments are made. This structure ensures real-time tracking of payments, allowing administrators to monitor financial obligations accurately.
- **Maintenance Request Handling**:

  A dedicated Maintenance Request table was created to manage maintenance requests. Each request is linked to a specific apartment, enabling administrators to quickly identify which apartments require attention, track the status of each request, and note resolution dates.

Additionally, residents can submit maintenance requests and check their status, ensuring transparency in the process and providing clear communication between residents and management.

- **Visitor and Parking Management**:

  The system includes a Visitors table that allows for the registration of visitors, their vehicles, and the assignment of parking spaces. This is vital for the security of the complex and ensures efficient management of parking space usage.

  The Parking Lot table tracks parking slot assignments for both residents and visitors, preventing conflicts over parking spaces and providing clear records of parking slot occupancy.

- **Operation of Common Areas**:

  Gym and Sauna tables were designed to manage common spaces. The Capacity and Operating Hours fields allow for the efficient use of these shared resources, preventing overcrowding and providing residents with information about when they can access these amenities.

  An additional reservation system could be implemented, allowing residents to book these spaces in advance, ensuring that usage remains within the allowed capacity and promoting fair use of these facilities.

Design by steps:

Step 1 (Components):

- Apartments

- Parking

- Services

Step 2 (Entities):

1. Holder
2. Parking lot
3. Tower
4. Apartment
5. Bill
6. Maintenance request
7. Gym
8. Sauna
9. Vehicles
10. Type – Vehicle
11. Type – Bill
12. PQR
13. Type – PQR

14. Administration
15. Type – PL
16. Visitor

Step 3 (Attributes):

Holder

- ID
- Name
- Phone Number
- Email

Parking lot

- Space Number: Specific parking spot number.
- Type
- Name Owner

Tower

- Name or Number: Identification for the tower.
- Number of Floors: Total floors in the tower.

Apartment

- Number: Apartment number.
- Floor: The floor where the apartment is located.
- Size: Square footage or size of the apartment.
- Number of Rooms: How many rooms the apartment has.

Bill

- ID: Unique identifier for the bill.
- Date Issued
- Amount: Total bill amount.
- Due Date
- Payment Status: Paid, unpaid, or overdue.
- Type of Bill: (Linked to Type - Bill entity).

Maintenance Request

- ID: Number of the maintenance
- Description: Details of the maintenance issue.

- Request Date
- Status: Pending, in progress, or completed.
- Resolution Date

Gym

- Name: Name of the gym.
- Capacity: Maximum number of people.
- Operating Hours: Times the gym is open.

Sauna

- Capacity: Maximum number of people.
- Operating Hours: Times the sauna is open.
- Availability: Whether the sauna can be used or reserved.

Vehicles

- ID: Unique identifier for the vehicle.
- License Plate Number
- Model: The vehicle's make and model.
- Color
- Type: (Linked to Type - Vehicle entity).

Type - Vehicle

- ID: Unique identifier for the vehicle type.
- Description: Defines the type of vehicle (e.g., car, motorcycle, bicycle).

Type - Bill

- ID: Unique identifier for the bill type.
- Description: Defines the type of bill (e.g., rent, utilities, maintenance fees).
- Frequency: How often the bill is issued (monthly, annually, one-time).

PQR (Petitions, Complaints, and Requests)

- ID: Unique identifier for the PQR.
- Type: The category of the PQR (e.g., Petition, Complaint, Request).
- Description: Detailed explanation of the issue or request.
- Date Submitted: The date when the PQR was submitted.
- Status: The current status of the PQR (e.g., Pending, In Progress, Resolved).
- Resolution Date: The date when the PQR was resolved (if applicable).
- Holder ID: Reference to the holder (resident) who submitted the PQR.
- Response: Any response or action taken by the administration.

Type - PQR

- Name: Name of the type (e.g., Petition, Complaint, Request).
- Description: A brief description of what this type covers (e.g., "Petitions for service improvements" or "Complaints about maintenance issues").

Administration

- ID: Unique identifier for the administration.
- Name
- Contact Info: Phone and email.
- Operating Hours: Hours when the administration office is open.

Type - PL

- Name: Name of the type.
- Description.

Visitor

- Name
- ID
- Date
- Vehicle
- Parking Slot

Step 4 (Relationships):

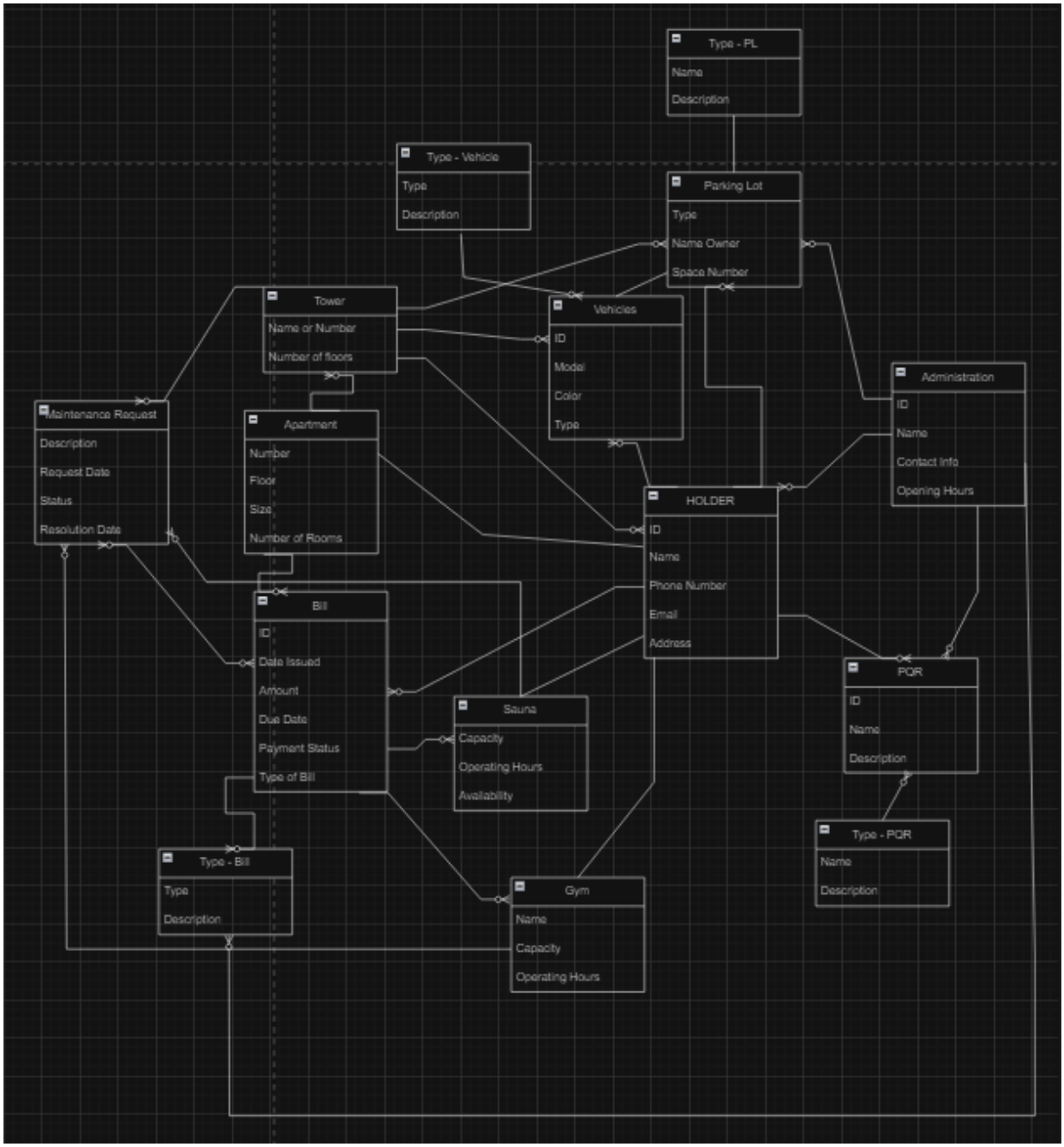
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1		✓	✓	✓	✓		✓	✓	✓			✓		✓		✓
2	✓		✓	✓					✓						✓	✓
3	✓	✓		✓		✓			✓							✓
4	✓	✓	✓		✓	✓										✓
5	✓			✓			✓	✓			✓			✓		
6			✓	✓			✓	✓					✓			
7	✓				✓	✓										✓
8	✓				✓	✓										✓
9	✓	✓	✓							✓						✓
10									✓							
11					✓											
12	✓					✓							✓	✓		
13						✓						✓				

14	✓				✓							✓				
15		✓														
16	✓	✓	✓	✓			✓	✓	✓							

Step 5 (Define relations):

- Holder – Parking Slot (One - Many)
- Holder – Tower (Many - One)
- Holder – Apartment (One - One)
- Holder – Bill (One - Many)
- Holder – Gym (One - One)
- Holder – Sauna (One – One)
- Holder – Vehicles (One - Many)
- Holder – PQR (One - Many)
- Holder – Administration (Many - One)
- Parking Slot – Tower (Many - One)
- Parking Slot – Apartment (Many - One)
- Parking Slot – Vehicles (One - One)
- Tower – Apartment (One - Many)
- Tower – Maintenance Request (One - Many)
- Tower – Vehicles (One - Many)
- Apartment – Bill (One - Many)
- **Apartment – Maintenance Request (Many - Many)**
- Bill – Gym (One - Many)
- Bill – Sauna (One - Many)
- Bill – Type – Bill (One - Many)
- Maintenance Request – Gym (Many - One)
- Maintenance Request – Sauna (Many - One)
- Maintenance Request – Type - PQR (One - One)
- Vehicles – Type - Vehicle (Many - One)
- PQR – Type - PQR (Many - One)
- PQR – Adminitration (Many - One)
- Type - PL – Parking Lot (One - One)
- Bill – Administration (Many - One)
- Visitor – Holder (Many - One)
- Visitor – Parking Lot (One - One)
- Visitor – Tower (Many - One)
- Visitor – Apartment (Many - One)
- Visitor – Gym (Many - One)
- Visitor – Sauna (Many - One)
- Visitor – Vehicles (One - One)

Step 6 (First design):



Step 7 (Many-Many consideration):

New entity: Apartment – Maintenance.

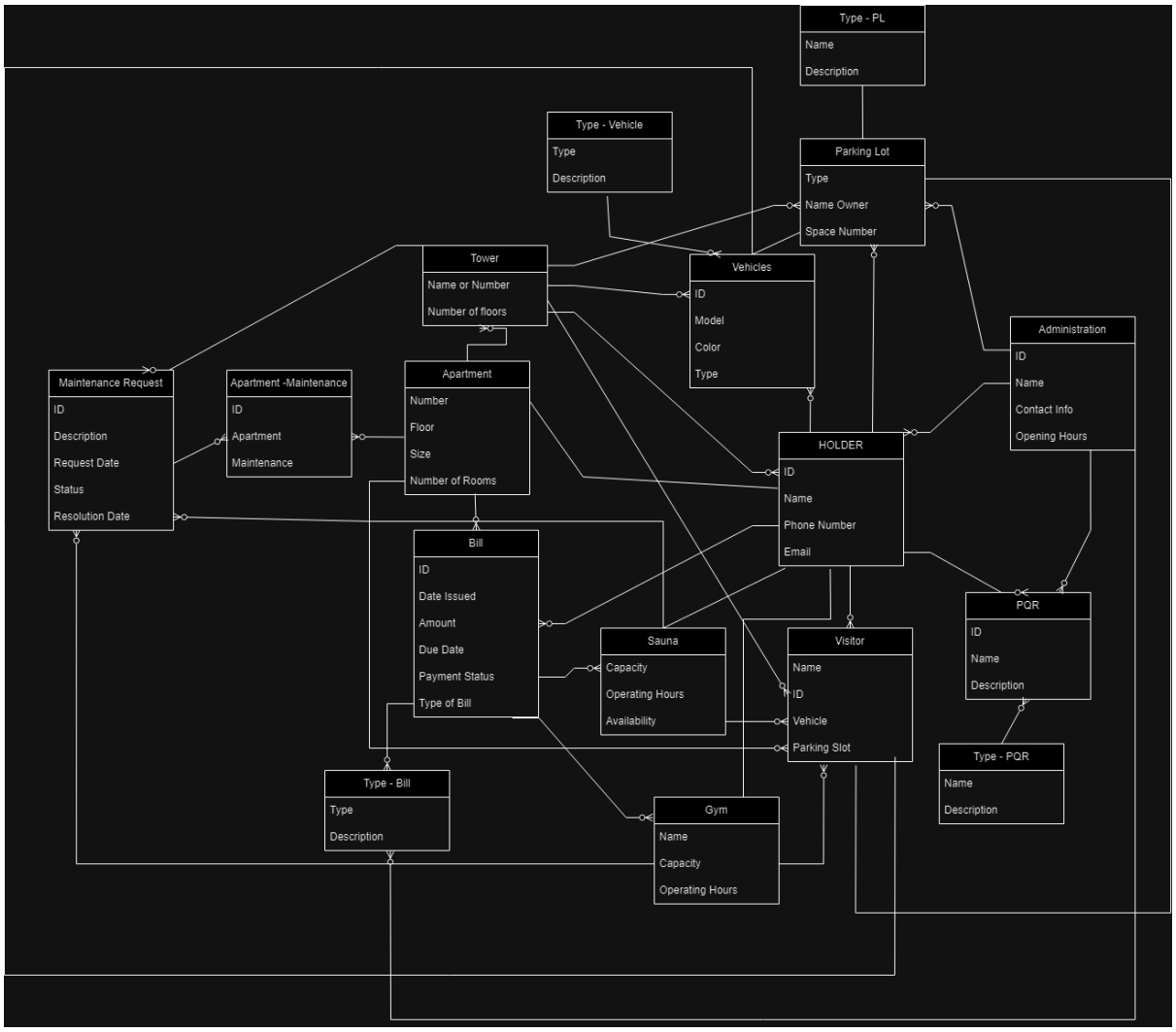
Apartment – Maintenance:

- ID (PK)
- Apartment (FK)
- Maintenance Request (FK)

Apartment - Maintenance – Apartment (Many - One)

Apartment - Maintenance – Maintenance Request (Many - One)

Step 8 (Final Design):



Step 9 (Get data-structured):

Holder

- ID: int (Unique identifier for the holder).
- Name: string (Full name of the holder).
- Phone Number: string (Contact phone number).
- Email: string (Email address).

Parking Lot

- Space Number: int (Specific parking spot number).
- Type: string (Linked to the Type - PL entity).
- Name Owner: string (Name of the holder owning the parking space).

Tower

- Name or Number: string (Identification for the tower, could be a name or a number).
- Number of Floors: int (Total number of floors in the tower).

Apartment

- Number: int (Apartment number).
- Floor: int (Floor where the apartment is located).
- Size: float (Square footage or size of the apartment).
- Number of Rooms: int (Number of rooms in the apartment).

Bill

- ID: int (Unique identifier for the bill).
- Date Issued: date (Date when the bill was issued).
- Amount: float (Total bill amount).
- Due Date: date (Date the bill is due).
- Payment Status: string (Paid, unpaid, or overdue).
- Type of Bill: string (Linked to the Type - Bill entity).

Maintenance Request

- ID: int (number of the maintenance request).
- Description: string (Details of the maintenance issue).
- Request Date: date (Date when the request was made).
- Status: string (Pending, in progress, or completed).
- Resolution Date: date (Date when the issue was resolved, if applicable).

Gym

- Name: string (Name of the gym).
- Capacity: int (Maximum number of people).

- Operating Hours: string (Operating hours of the gym).

Sauna

- Capacity: int (Maximum number of people).
- Operating Hours: string (Operating hours of the sauna).
- Availability: string (Whether the sauna is available or reserved).

Vehicles

- ID: int (Unique identifier for the vehicle).
- License Plate Number: string (License plate number).
- Model: string (Vehicle make and model).
- Color: string (Color of the vehicle).
- Type: string (Linked to the Type - Vehicle entity).

Type - Vehicle

- ID: int (Unique identifier for the vehicle type).
- Description: string (Description of the vehicle type, e.g., car, motorcycle, bicycle).

Type - Bill

- ID: int (Unique identifier for the bill type).
- Description: string (Description of the bill type, e.g., rent, utilities, maintenance fees).
- Frequency: string (How often the bill is issued: monthly, annually, one-time).

PQR (Petitions, Complaints, and Requests)

- ID: int (Unique identifier for the PQR).
- Type: string (Category of the PQR: Petition, Complaint, Request).
- Description: string (Detailed explanation of the issue or request).
- Date Submitted: date (Date the PQR was submitted).
- Status: string (Current status: Pending, In Progress, Resolved).
- Resolution Date: date (Date the PQR was resolved, if applicable).
- Holder ID: int (Reference to the holder who submitted the PQR).
- Response: string (Any response or action taken by the administration).

Type - PQR

- Name: string (Name of the PQR type: Petition, Complaint, Request).
- Description: string (Brief description of what this type covers).

Administration

- ID: int (Unique identifier for the administration).
- Name: string (Name of the administration).
- Contact Info: string (Phone and email contact information).
- Operating Hours: string (Hours when the administration office is open).

Type - PL

- Name: string (Name of the parking lot type).
- Description: string (Description of the parking lot type).

Apartment – Maintenance:

- ID: int
- Apartment: int (Apartment number)
- Maintenance Request: int (Mainteinance ID)

Visitor

- Name: String
- ID: Int
- Date: Datetime
- Vehicle: String
- Parking Slot: Int

Step 10 (Constraints and Properties):

Holder

- ID: int
    - Constraints: PRIMARY KEY, NOT NULL, AUTO_INCREMENT.
    - Properties: Must be unique, automatically generated.
- Name: string
    - Constraints: NOT NULL, VARCHAR(100).
    - Properties: Must contain only alphabetic characters and spaces, with a minimum length of 2 characters.

- Phone Number: string
    - Constraints: NOT NULL, VARCHAR(15).
    - Properties: Can contain digits, dashes, or spaces. Must be a valid phone number format (e.g., regex check).
- Email: string
    - Constraints: NOT NULL, UNIQUE, VARCHAR(100).
    - Properties: Must follow valid email format (e.g., name@domain.com).

Parking Lot

- Space Number: int
    - Constraints: NOT NULL, UNIQUE.
    - Properties: Must be a positive integer.
- Type: string
    - Constraints: NOT NULL, VARCHAR(50).
    - Properties: Linked to the Type - PL entity. Must correspond to a valid parking lot type (e.g., covered, uncovered).
- Name Owner: string
    - Constraints: NOT NULL, VARCHAR(100).
    - Properties: Must reference an existing holder's name.

Tower

- Name or Number: string
    - Constraints: NOT NULL, VARCHAR(50).
    - Properties: Can be a name or numeric value to identify the tower.
- Number of Floors: int
    - Constraints: NOT NULL, CHECK (Number of Floors > 0).
    - Properties: Must be a positive integer.

Apartment

- Number: int
    - Constraints: NOT NULL, UNIQUE.
    - Properties: Must be a positive integer.
- Floor: int
    - Constraints: NOT NULL, CHECK (Floor >= 0).
    - Properties: Must be a non-negative integer.
- Size: float
    - Constraints: NOT NULL, CHECK (Size > 0).
    - Properties: Must be a positive floating-point number representing square footage.

- Number of Rooms: int
    - Constraints: NOT NULL, CHECK (Number of Rooms > 0).
    - Properties: Must be a positive integer.

Bill

- ID: int
    - Constraints: PRIMARY KEY, NOT NULL, AUTO_INCREMENT.
    - Properties: Must be unique, automatically generated.
- Date Issued: date
    - Constraints: NOT NULL.
    - Properties: Must be a valid date and cannot be in the future.
- Amount: float
    - Constraints: NOT NULL, CHECK (Amount > 0).
    - Properties: Must be a positive floating-point value.
- Due Date: date
    - Constraints: NOT NULL.
    - Properties: Must be a valid date and cannot be in the past relative to the issue date.
- Payment Status: string
    - Constraints: NOT NULL, VARCHAR(10), CHECK (Payment Status IN ('Paid', 'Unpaid', 'Overdue')).
    - Properties: Restricted to these three possible values.
- Type of Bill: string
    - Constraints: NOT NULL, VARCHAR(50).
    - Properties: Linked to the Type - Bill entity.

Maintenance Request

- Description: string
    - Constraints: NOT NULL, VARCHAR(255).
    - Properties: Brief description of the issue.
- Request Date: date
    - Constraints: NOT NULL.
    - Properties: Must be a valid date, cannot be in the future.
- Status: string
    - Constraints: NOT NULL, VARCHAR(15), CHECK (Status IN ('Pending', 'In Progress', 'Completed')).
    - Properties: Limited to these possible values.
- Resolution Date: date
    - Constraints: NULLABLE.
    - Properties: Must be a valid date, and only filled after the issue is resolved.

Gym

- Name: string
    - Constraints: NOT NULL, VARCHAR(100).
    - Properties: Must be unique to avoid name conflicts.
- Capacity: int
    - Constraints: NOT NULL, CHECK (Capacity > 0).
    - Properties: Must be a positive integer.
- Operating Hours: string
    - Constraints: NOT NULL, VARCHAR(50).
    - Properties: Should follow a valid time range format (e.g., "6 AM - 10 PM").

Sauna

- Capacity: int
    - Constraints: NOT NULL, CHECK (Capacity > 0).
    - Properties: Must be a positive integer.
- Operating Hours: string
    - Constraints: NOT NULL, VARCHAR(50).
    - Properties: Same as gym operating hours.
- Availability: string
    - Constraints: NOT NULL, VARCHAR(10), CHECK (Availability IN ('Available', 'Reserved')).
    - Properties: Limited to these two values.

Vehicles

- ID: int
    - Constraints: PRIMARY KEY, NOT NULL, AUTO_INCREMENT.
    - Properties: Must be unique, automatically generated.
- License Plate Number: string
    - Constraints: NOT NULL, VARCHAR(10), UNIQUE.
    - Properties: Must follow a valid license plate format.
- Model: string
    - Constraints: NOT NULL, VARCHAR(50).
    - Properties: Vehicle’s make and model.
- Color: string
    - Constraints: VARCHAR(20).
    - Properties: Optional, can be a common color name.
- Type: string

    - Constraints: NOT NULL, VARCHAR(50).
    - Properties: Linked to the Type - Vehicle entity.

Type - Vehicle

- ID: int
    - Constraints: PRIMARY KEY, NOT NULL, AUTO_INCREMENT.
    - Properties: Must be unique, automatically generated.
- Description: string
    - Constraints: NOT NULL, VARCHAR(50).
    - Properties: Brief description of the vehicle type (e.g., car, motorcycle, bicycle).

Type - Bill

- ID: int
    - Constraints: PRIMARY KEY, NOT NULL, AUTO_INCREMENT.
    - Properties: Must be unique, automatically generated.
- Description: string
    - Constraints: NOT NULL, VARCHAR(50).
    - Properties: Brief description of the bill type.
- Frequency: string
    - Constraints: NOT NULL, VARCHAR(10), CHECK (Frequency IN ('Monthly', 'Annually', 'One-time')).
    - Properties: Restricted to these values.

PQR (Petitions, Complaints, and Requests)

- ID: int
    - Constraints: PRIMARY KEY, NOT NULL, AUTO_INCREMENT.
    - Properties: Must be unique, automatically generated.
- Type: string
    - Constraints: NOT NULL, VARCHAR(10), CHECK (Type IN ('Petition', 'Complaint', 'Request')).
    - Properties: Limited to these values.
- Description: string
    - Constraints: NOT NULL, VARCHAR(255).
    - Properties: Detailed explanation of the issue or request.
- Date Submitted: date
    - Constraints: NOT NULL.
    - Properties: Valid date, cannot be in the future.

- Status: string
    - Constraints: NOT NULL, VARCHAR(15), CHECK (Status IN ('Pending', 'In Progress', 'Resolved')).
    - Properties: Limited to these statuses.
- Resolution Date: date
    - Constraints: NULLABLE.
    - Properties: Only filled after the issue is resolved.
- Holder ID: int
    - Constraints: FOREIGN KEY, NOT NULL.
    - Properties: Must reference an existing holder ID.
- Response: string
    - Constraints: VARCHAR(255).
    - Properties: Optional response from the administration.

Type - PQR

- Name: string
    - Constraints: PRIMARY KEY, NOT NULL.
    - Properties: Must be unique.
- Description: string
    - Constraints: NOT NULL, VARCHAR(40).
    - Properties: Detalied explenation of type.

Type - PQR

- Name: string
    - Constraints: PRIMARY KEY, NOT NULL.
    - Properties: Must be unique.
- Name: string
    - Constraints: PRIMARY KEY, NOT NULL.
    - Properties: Must be unique.
- Name: string
    - Constraints: PRIMARY KEY, NOT NULL.
    - Properties: Must be unique.

Apartment - Maintenance

- ID: int
    - Constraints: PRIMARY KEY, NOT NULL, AUTO_INCREMENT.
    - Properties: Must be unique, automatically generated.
- Apartment: int

    - Constraints: NOT NULL, FOREIGN KEY (references the Apartment entity, Apartment.Number).
    - Properties: Must correspond to an existing apartment number, representing the apartment where the maintenance request applies.
- Maintenance Request: int
    - Constraints: NOT NULL, FOREIGN KEY (references the Maintenance Request entity, Maintenance Request.ID).
    - Properties: Must correspond to a valid maintenance request ID, linking the request to the specific apartment.

Visitor

- Name: String
    - Constraints: NOT NULL, VARCHAR (100).
    - Properties: Must contain only alphabetic characters and spaces, with a minimum length of 2 characters.
- ID: int
    - Constraints: PRIMARY KEY, NOT NULL, AUTO_INCREMENT.
    - Properties: Must be unique.
- Vehicle: int
    - Constraints: NOT NULL, FOREIGN KEY (references the type of vehicle).
    - Properties: Must correspond to a valid type of vehicle.
- Parking Slot: int
    - Constraints: NULL, FOREIGN KEY (references the number of parking slot, if Visitor has a valid vehicle type for a lot).
    - Properties: Must be a valid number.