WorkShop 1 - Apartment Complex

2) List of stories

1. As a resident, I want to reserve common areas to ensure space is available when I need it.
2. As a resident, I want to receive notifications when packages arrive, so I can stay informed of my deliveries.
3. As a resident, I want to report maintenance issues, so they can be resolved quickly.
4. As a resident, I want to see my utility and fee payment balance, so I can stay up to date with my payments.
5. As a resident, I want to communicate with my neighbors, to foster a better community.
6. As a resident, I want to access the community's calendar of activities, so I can stay informed of available events and classes.
7. As a resident, I want to pay for utilities from the app, to make my payments easier and faster.
8. As a resident, I want to have access to a list of emergency numbers for the community, so I can quickly contact them in case of an emergency.
9. As a resident, I want to receive notifications about important utility outages, so I can be prepared and take precautions.
10. As a resident, I want to see how many spaces are available in the visitor parking lot, so I can better manage visits.
11. As a resident, I want to access an interactive map of the community, so I can easily locate important areas.
12. As a resident, I want to schedule guided tours for my friends or family, to facilitate their access to the complex.
13. As a resident, I want to see the security cameras in real time, to monitor the security of the complex.
14. As a resident, I want to receive automatic reminders about payment due dates, to avoid delays or penalties.
15. As a resident, I want to reserve shifts at the complex's gym, to ensure I have access at the desired time.

3) **Quantity of blocks and apartments**

Number of blocks: 6
Number of apartments per block: 12
Total apartaments: 72

Administration services (gardening, maintenance, etc.): $250.000
Parking (including visitor parking): $150.000
TOTAL: $400.000

Quantity of common spaces: 1

4) **Extra functionalities:**

- **Receive notifications when packages arrive**
- **Report maintenance issues**
- **View my utility and fee payment balance**
- **Communicate with my neighbors**
- **Community activity calendar**
- **Pay utilities from the app**
- **Have access to a list of community emergency numbers**
- **Receive notifications about important utility outages**
- **See how many spaces are available in the visitor parking lot**
- **An interactive map of the complex**
- **Schedule guided tours for my friends or family, to facilitate their access to the complex.**
- **View security cameras in real time**
- **Receive automatic reminders about payment due dates**
- **Book appointments at the complex gym**

5)Diagram of MER

1. Identification of key elements

- **Apartment blocks**
- **Apartments**
- **Residents**
- **Management services**
- **Parking (including visitors)**
- **Packages**
- **Maintenance reports**
- **Payments**
- **Reservations (for common areas or gym)**
- **Community events or activities**
- **Security cameras**
- **Guided tours**
- **Emergency numbers**

2. Definition of entities

- **Block**
- **Apartment**
- **Resident**

- **Administration service**
- **Parking**
- **Package**
- **Maintenance report**
- **Payment**
- **Reservation**
- **Community event**
- **Security camera**
- **Guided tour**
- **Emergency number**

3. Entity Attributes

1. Block

- **Block ID (PK)**
- **Block name or number**
- **Address or location within the complex**

2. Apartment

- **Apartment ID (PK)**
- **Apartment number**
- **Block ID (FK)**
- **Apartment size (square meters)**
- **Status (occupied or available)**

3. Resident

- **Resident ID (PK)**
- **Full name**
- **Email**
- **Phone number**
- **Apartment ID (FK)**
- **Contract start date**
- **Resident type (owner or tenant)**

4. Management service

- **Service ID (PK)**
- **Service type (gardening, maintenance, etc.)**
- **Monthly cost**

5. Parking

- **Parking ID (PK)**

- **Parking type (resident or visitor)**
- **Availability (number of spaces)**
- **Block ID (FK)**

6. Package

- **Package ID (PK)**
- **Arrival date**
- **Package status (delivered, pending)**
- **Parking ID (PK)**
- **Arrival date**
- **Parking ... Resident (FK)**

7. Maintenance Report

- **Report ID (PK)**
- **Issue Description**
- **Report Date**
- **Report Status (Pending, Processing, Resolved)**
- **Resident ID (FK)**
- **Apartment ID (FK)**

8. Payment

- **Payment ID (PK)**
- **Payment Date**
- **Payment Amount**
- **Payment Type (Utilities, Management, Other)**
- **Resident ID (FK)**
- **Apartment ID (FK)**

9. Reservation

- **Reservation ID (PK)**
- **Reservation Date**
- **Reserved Space (Gym, Event Hall, Pool, etc.)**
- **Resident ID (FK)**
- **Block ID (FK)**
- **10. Community Event**
- **Event ID (PK)**
- **Event Name**
- **Event Date and Time**
- **Event Description**
- **Block ID (FK)**

11. Security Camera

- **Camera ID (PK)**
- **Camera Location**
- **Camera Status (Active, Inactive)**

12. Guided tour

- **Visit ID (PK)**
- **Date of visit**
- **Resident ID (FK)**
- **Guide or caretaker ID (FK)**

13. Emergency number

- **Number ID (PK)**
- **Type of emergency (security, fire, ambulance, etc.)**
- **Phone number**

For the fourth step I decided to make an image representing how the relationships and connections between each type of entity work.

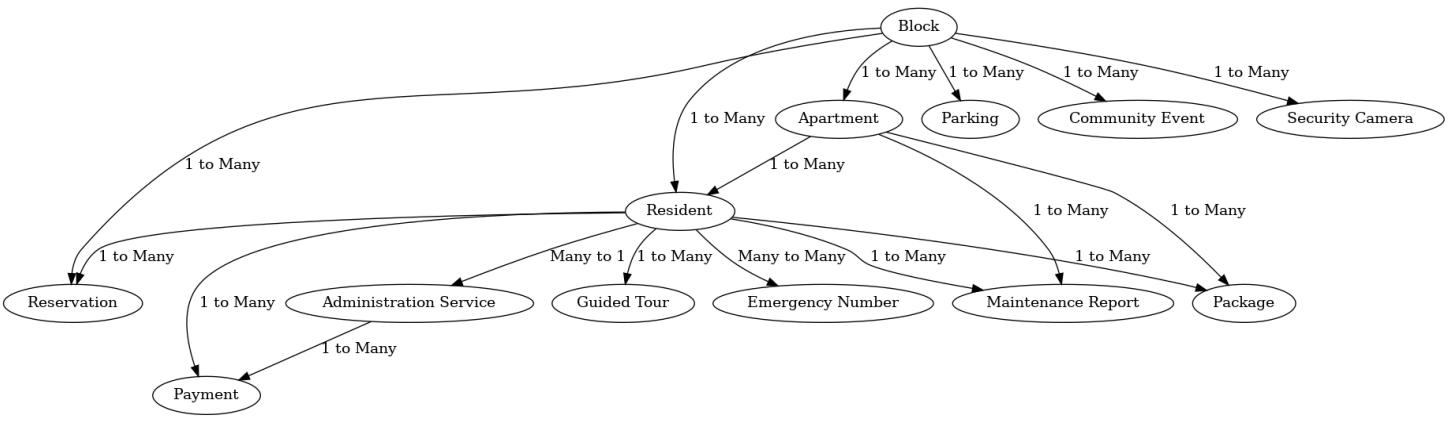
Entidad \ Relación	Bloque	Apartamento	Residente	Servicio Adm.	Estacionamiento	Paquete	Reporte Mant.	Pago	Reserva	Evento	Cámara	Visita Guiada	Número Emerg.
Bloque	-	✓	✓		✓				✓	✓	✓		
Apartamento	✓	-	✓			✓	✓	✓					
Residente	✓	✓	-	✓	✓	✓	✓	✓	✓			✓	✓
Servicio de administración			✓	-				✓					
Estacionamiento	✓		✓		-								
Paquete		✓	✓			-							
Reporte de mantenimiento		✓	✓				-						
Pago		✓	✓	✓				-					
Reserva	✓		✓						-				
Evento comunitario	✓									-			
Cámara de seguridad	✓										-		
Visita guiada			✓									-	
Número de emergencia			✓										-

The fifth step I will do in a written and graphic way demonstrating the reason for the relationship.

- **Block - Apartment: 1 to many (A block has many apartments, one apartment belongs to only one block).**
- **Block - Resident: 1 to many (A block has many residents, but one resident lives in one apartment in only one block).**
- **Block - Parking: 1 to many (A block can have multiple parking spaces assigned to it, one parking lot belongs to one block).**
- **Block - Reservation: 1 to many (A block can have multiple reservations for its common areas).**
- **Block - Community Event: 1 to many (A block can host multiple community events).**
- **Block - Security Camera: 1 to many (A block has multiple security cameras assigned to it).**
- **Apartment - Resident: 1 to many (An apartment can have multiple residents, but one resident lives in only one apartment).**
- **Apartment - Package: 1 to many (An apartment can receive multiple packages, one package belongs to only one apartment).**
- **Apartment - Maintenance Report: 1 to many (An apartment can have multiple maintenance reports).**
- **Resident - Management Service: Many to 1 (Many residents use the same management services, such as gardening or maintenance.)**
- **Resident - Parking: 1 to many (A resident may have access to one or more parking spaces, but a space may only be assigned to one resident at a time.)**
- **Resident - Package: 1 to many (A resident may receive multiple packages.)**
- **Resident - Maintenance Report: 1 to many (A resident may generate multiple maintenance reports.)**
- **Resident - Payment: 1 to many (A resident may make multiple payments for services and fees.)**
- **Resident - Reservation: 1 to many (A resident may make multiple reservations for common spaces.)**
- **Resident - Guided Tour: 1 to many (A resident may schedule multiple guided tours for guests.)**
- **Resident - Emergency Number: Many to many (Multiple residents may access the same emergency numbers.)**
- **Management Service - Payment: 1 to many (A management service may generate many resident-related payments.)**

Entity 1	Relationship	Entity 2	Relationship Type
Block	has	Apartment	1 to many
Block	has	Resident	1 to many
Block	has	Parking	1 to many
Block	has	Reservation	1 to many
Block	organizes	Community Event	1 to many
Block	has	Security Camera	1 to many
Apartment	houses	Resident	1 to many
Apartment	receives	Package	1 to many
Apartment	generates	Maintenance Report	1 to many
Resident	uses	Administration Service	many to 1
Resident	uses	Parking	1 to many
Resident	receives	Package	1 to many
Resident	generates	Maintenance Report	1 to many
Resident	makes	Payment	1 to many
Resident	makes	Reservation	1 to many
Resident	schedules	Guided Tour	1 to many
Resident	accesses	Emergency Number	many to many
Administration Service	generates	Payment	1 to many

As a sixth step This is a preview of the diagram I developed (Can't find the arrow type in paw shape so I wrote the relationship type just like that)

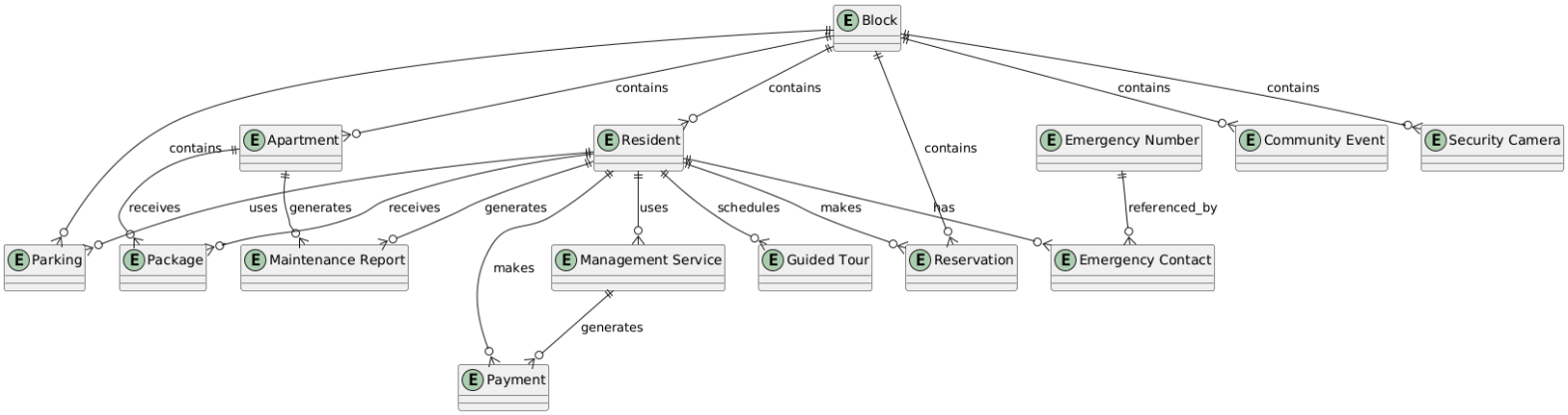


Since the project has a one-to-many relationship, which is the relationship between resident and emergency number, we will move this relationship to a different relationship like this.This is the seventh step (I found that PlantUML can generate this type of arrows so I changed the format in which I made the diagram)

Resident - Emergency Number: To simplify this relationship, we could introduce an intermediate entity called "Emergency Contact" to act as a bridge. Thus, the relationship would be broken down into:

Residents have one or many emergency contacts.

Emergency Number is referenced by one or many emergency contacts.



In step number nine, I will define the variable type for each component of each entity. I will display it in text form

Block

- **Block ID (PK): int**
- **Block name or number: string**
- **Address or location within the complex: string**

Apartment

- **Apartment ID (PK): int**
- **Apartment number: string**
- **Block ID (FK): int**
- **Apartment size (square meters): float**
- **Status (occupied or available): string**

Resident

- **Resident ID (PK): int**
- **Full name: string**
- **Email: string**
- **Phone number: string**
- **Apartment ID (FK): int**
- **Contract start date: date**
- **Resident type (owner or tenant): string**

Management Service

- **Service ID (PK): int**
- **Service type (gardening, maintenance, etc.): string**
- **Monthly cost: float**

Parking

- **Parking ID (PK): int**
- **Parking type (resident or visitor): string**
- **Availability (number of spaces): int**
- **Block ID (FK): int**

Package

- **Package ID (PK): int**
- **Arrival date: date**
- **Package status (delivered, pending): string**
- **Parking ID (FK): int**
- **Resident ID (FK): int**

Maintenance Report

- **Report ID (PK): int**
- **Issue Description: string**
- **Report Date: date**
- **Report Status (Pending, Processing, Resolved): string**
- **Resident ID (FK): int**
- **Apartment ID (FK): int**

Payment

- **Payment ID (PK): int**
- **Payment Date: date**
- **Payment Amount: float**
- **Payment Type (Utilities, Management, Other): string**
- **Resident ID (FK): int**
- **Apartment ID (FK): int**

Reservation

- **Reservation ID (PK): int**

- **Reservation Date: date**
- **Reserved Space (Gym, Event Hall, Pool, etc.): string**
- **Resident ID (FK): int**
- **Block ID (FK): int**

Community Event

- **Event ID (PK): int**
- **Event Name: string**
- **Event Date and Time: datetime**
- **Event Description: string**
- **Block ID (FK): int**

Security Camera

- **Camera ID (PK): int**
- **Camera Location: string**
- **Camera Status (Active, Inactive): string**

Guided Tour

- **Visit ID (PK): int**
- **Date of visit: date**
- **Resident ID (FK): int**
- **Guide or caretaker ID (FK): int**

Emergency Number

- **Number ID (PK): int**
- **Type of emergency (security, fire, ambulance, etc.): string**
- **Phone number: string**

Now for the 10th and final step defining the restrictions and properties of the data I will give in text a form in which the data can be and finally I will show an image of how the completed database design looks.

Block

- **Block ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Block name or number: varchar(50) (NOT NULL)**
- **Address or location within the complex: varchar(255) (NOT NULL)**

Apartment

- **Apartment ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Apartment number: varchar(10) (NOT NULL)**
- **Block ID (FK): int (FOREIGN KEY, NOT NULL)**
- **Apartment size (square meters): float (NOT NULL, CHECK (size >= 0))**

- **Status (occupied or available): varchar(20) (NOT NULL, CHECK (status IN ('occupied', 'available')))**

Resident

- **Resident ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Full name: varchar(100) (NOT NULL)**
- **Email: varchar(100) (NOT NULL, UNIQUE)**
- **Phone number: varchar(15) (NULL)**
- **Apartment ID (FK): int (FOREIGN KEY, NULL)**
- **Contract start date: date (NOT NULL)**
- **Resident type (owner or tenant): varchar(20) (NOT NULL, CHECK (resident_type IN ('owner', 'tenant')))**

Management Service

- **Service ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Service type (gardening, maintenance, etc.): varchar(50) (NOT NULL)**
- **Monthly cost: float (NOT NULL, CHECK (cost >= 0))**

Parking

- **Parking ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Parking type (resident or visitor): varchar(20) (NOT NULL, CHECK (parking_type IN ('resident', 'visitor')))**
- **Availability (number of spaces): int (NOT NULL, CHECK (availability >= 0))**
- **Block ID (FK): int (FOREIGN KEY, NOT NULL)**

Package

- **Package ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Arrival date: date (NOT NULL)**
- **Package status (delivered, pending): varchar(20) (NOT NULL, CHECK (package_status IN ('delivered', 'pending')))**
- **Parking ID (FK): int (FOREIGN KEY, NULL)**
- **Resident ID (FK): int (FOREIGN KEY, NULL)**

Maintenance Report

- **Report ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Issue Description: text (NOT NULL)**
- **Report Date: date (NOT NULL)**
- **Report Status (Pending, Processing, Resolved): varchar(20) (NOT NULL, CHECK (report_status IN ('Pending', 'Processing', 'Resolved')))**
- **Resident ID (FK): int (FOREIGN KEY, NULL)**
- **Apartment ID (FK): int (FOREIGN KEY, NULL)**

Payment

- **Payment ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Payment Date: date (NOT NULL)**
- **Payment Amount: float (NOT NULL, CHECK (payment_amount >= 0))**
- **Payment Type (Utilities, Management, Other): varchar(20) (NOT NULL, CHECK (payment_type IN ('Utilities', 'Management', 'Other')))**
- **Resident ID (FK): int (FOREIGN KEY, NOT NULL)**
- **Apartment ID (FK): int (FOREIGN KEY, NULL)**

Reservation

- **Reservation ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Reservation Date: date (NOT NULL)**
- **Reserved Space (Gym, Event Hall, Pool, etc.): varchar(50) (NOT NULL)**
- **Resident ID (FK): int (FOREIGN KEY, NOT NULL)**
- **Block ID (FK): int (FOREIGN KEY, NOT NULL)**

Community Event

- **Event ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Event Name: varchar(100) (NOT NULL)**
- **Event Date and Time: datetime (NOT NULL)**
- **Event Description: text (NULL)**
- **Block ID (FK): int (FOREIGN KEY, NULL)**

Security Camera

- **Camera ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Camera Location: varchar(100) (NOT NULL)**
- **Camera Status (Active, Inactive): varchar(20) (NOT NULL, CHECK (camera_status IN ('Active', 'Inactive')))**

Guided Tour

- **Visit ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Date of visit: date (NOT NULL)**
- **Resident ID (FK): int (FOREIGN KEY, NOT NULL)**
- **Guide or caretaker ID (FK): int (FOREIGN KEY, NULL)**

Emergency Number

- **Number ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Type of emergency (security, fire, ambulance, etc.): varchar(50) (NOT NULL)**
- **Phone number: varchar(15) (NOT NULL)**

Emergency Contact

- **Contact ID (PK): int (PRIMARY KEY, NOT NULL, AUTO_INCREMENT)**
- **Resident ID (FK): int (FOREIGN KEY, NOT NULL)**
- **Number ID (FK): int (FOREIGN KEY, NOT NULL)**

The graphical view of the MER diagram of the complete database looks like this: