DataBase Foundations

Season 2024-III

Workshop No. 1 — DataBase Design

Oscar Santiago Montoya Amaya

20212020076

osmontoyaa@udistrital.edu.co

Computer Engineering

Universidad Distrital Francisco José de Caldas

Once the responses have been collected, we can make user stories to get started.

Most important user stories:

- As a potential resident, I want to have clear payment options (online, credit card), then you will be able to pay your rent in the easiest and most convenient way.
- As a future resident, I want to obtain detailed information about the services (gym, swimming pool), then I will be able to know if these services cover my wellness and leisure needs.
- As an interested resident, I want a tool to calculate the total cost (rent + utilities), then I will be able to better plan my budget and avoid surprises.

Less important user stories:

- As an applicant, I want to have access to a live chat with the administrator, then I will be able to resolve any questions in real time without having to wait.
- As a future resident, I want to be able to chat with other residents so I can get to know the community better and get opinions about life in the community.
- As an animal lover, I want to know the complex's pet policies so I can know if I can move in with my pet without problems.
- As a potential resident, I want to be informed about social or community events, then I will be able to better integrate into the community and participate in activities.

Now with the user stories we can define the important components of the application

Most important

- Shows a list of all blocks and apartments.
- Receive payments for administration services and everything related.
- Make reservations for common spaces.
- Wall where it shows all the updated information about all the information of common interest (status of the apartments, and social and community events)
- A place where all the rights and duties of the residents and the administration are shown.

Less important

- A chat to talk with the administrator, with the residents themselves and/or both

Once the components have been defined, separated by the most important ones, we can begin to create the different entities in the database with their respective attributes:

1. **Blocks :Represents the different blocks in the apartment complex:**
   - 1.1. Block ID
   - 1.2. Block Name (e.g., A, B, C)
   - 1.3. Total Apartments (Number of apartments in the block)
2. **Apartments: Represents individual apartments within each block.**
   - 2.1. Apartment ID
   - 2.2. Apartment Number (e.g., 101, 102)
   - 2.3. ApartmentStatus (Occupied, Vacant, etc.)
   - 2.4. Size (Square meters, etc.)
   - 2.5. RentStatus (Paid, Pending)
3. **Residents: Represents the residents living in the apartments.**
   - 3.1. Resident ID
   - 3.2. FirstName

3.3. LastName

3.4. Phone Number

3.5. Email

3.6. Role (Owner, Tenant, etc.)

4. **Payments: Tracks payments made by residents for administrative services.**

4.1. Payment ID

4.2. Amount

4.3. Payment Date

4.4. Payment Type (Service, Admin, etc.)

4.5. Status (Paid, Pending)

5. **Reservations: Represents reservations for common spaces such as gyms, event rooms, etc.**

5.1. Reservation ID

5.2. Reservation Date

5.3. Start Time

5.4. End Time

6. **CommonSpaces: Represents common areas that can be reserved by residents.**

6.1. Space ID

6.2. Space Name (e.g., Gym, Pool, Event Room)

6.3. Capacity

6.4. Availability Status (Available, Maintenance, etc.)

7. **Announcements: A wall to post updates on community events, apartment statuses, or news.**

7.1. Announcement ID

7.2. Title

7.3. Message

7.4. Date Posted

7.5. Posted By (Admin, Resident, etc.)

8. RightsAndDuties:Contains the rights and duties of residents and administrators.

8.1. Right Duty ID

8.2. Title

8.3. Description

8.4. Type (Right, Duty)

8.5. Audience (Resident, Administration)

To represent the relationships between the entities using a Relationships Matrix, we'll use e1, e2, etc., to refer to each entity. Here's the assignment of the entities:

e1 = Blocks

e2 = Apartments

e3 = Residents

e4 = Payments

e5 = Reservations

e6 = CommonSpaces

e7 = Announcements

e8 = RightsAndDuties

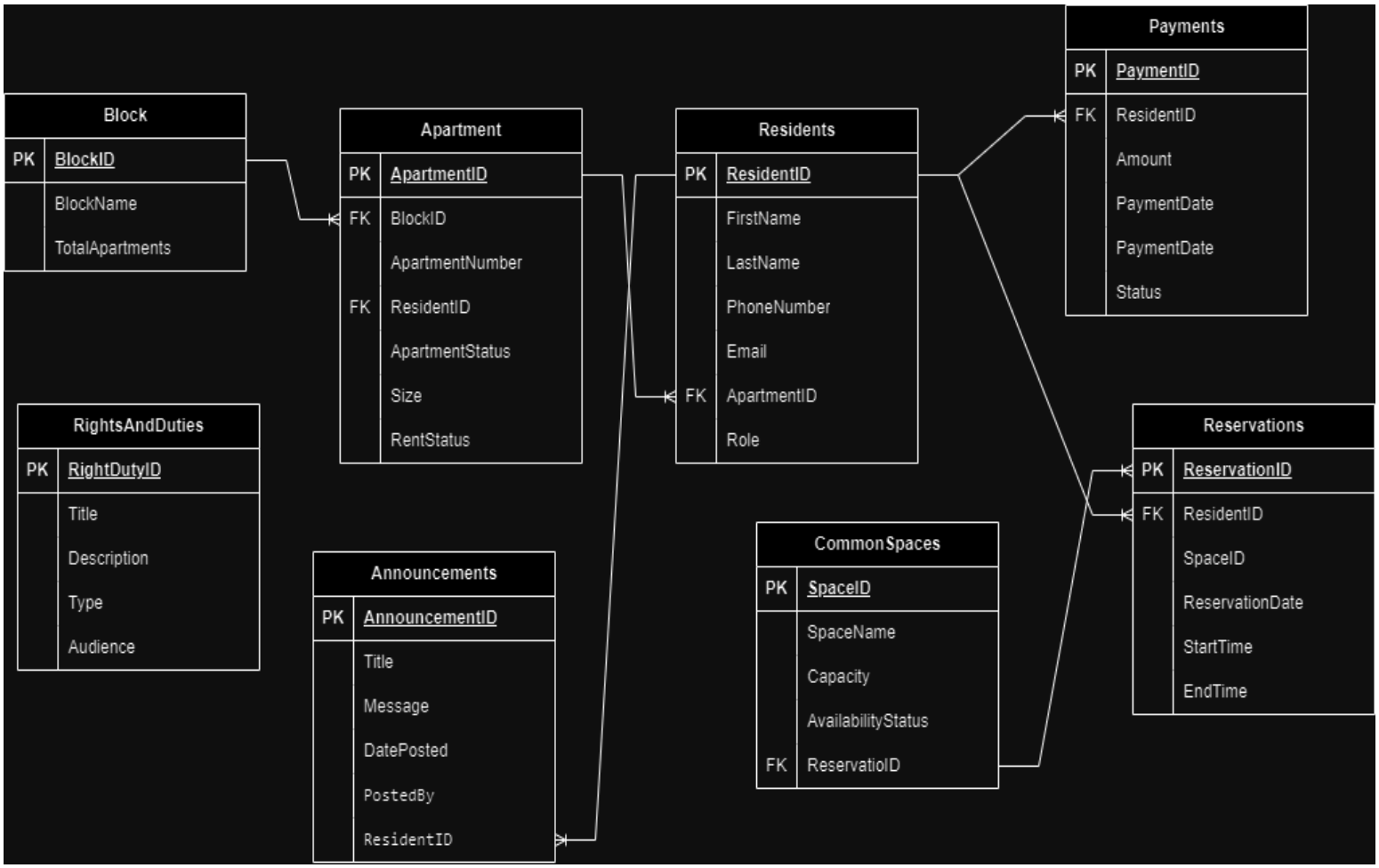
	e1	e2	e3	e4	e5	e6	e7	e8
e1		x						
e2	x		x					
e3		x		x	x		x	
e4			x					
e5			x			x		
e6					x			
e7			x					
e8								

Relationships:

- ☐ Blocks and Apartments have a one-to-many relationship (one block can have many apartments).

- [ ] Apartments and Residents have a one-to-one relationship (one apartment has one resident).
- [ ] Residents and Payments have a one-to-many relationship (one resident can make multiple payments).
- [ ] Residents and Reservations have a one-to-many relationship (one resident can make multiple reservations).
- [ ] Reservations and CommonSpaces have a many-to-one relationship (many reservations can be made for one common space).
- [ ] Announcements are related to Residents or Administrators (can be posted by both).
- [ ] RightsAndDuties are visible to all residents and administration.

Next with the relationships and the type i made the entity-relationship diagram



Once with the relationships we can define the data type of each attribute and whether it is null or not to specify whether the data in the attribute itself is mandatory or not.