Workshop #1: Apartament complex aplication
Juan David Páez Olivares – 20231020028
Juan Esteban Martinez Buritica - 20222020039

Wordshop #1: Apartament complex aplication

In the following report we present a database design for an application of an apartment complex, in which different functionalities were presented based on an analysis of different interviews conducted with 15 students, from which we were able to create the following user stories, focused on a user who is a tenant within the apartment complex.

User Stories:

- As a tenant, I want to be able to pay my rent online securely and easily to avoid delays and additional fees.
- As a tenant, I want to be able to request repairs and maintenance through an online platform so that my requests are handled efficiently.
- As a tenant, I want to have access to important documents such as the lease and receipts so that I can refer to them at any time.
- As a tenant, I want to receive reminders about important dates such as rent expiration and lease renewal to avoid inconveniences.

Technical and design considerations/decisions:

For the design of databases for this software, we have decided to use SQL databases, since the data needed to carry out the components that have been extracted from the user stories can be processed using relational tables.

Database Design

STEP 1. Define components

Online Payment: Allows tenants to make payments for rent and other charges associated with their tenancy securely and conveniently over the Internet. By eliminating the need for cash or check payments, it reduces the risk of payment errors, and offers greater flexibility to tenants.

Maintenance requests: A system that allows tenants to report problems or request repairs to their units quickly and easily. Streamlining the process of requesting and tracking repairs, improves communication between tenants and the maintenance team, and helps prioritize tasks.

Documents: A digital repository where all important documents related to the apartment complex and tenants are stored. Facilitating access to documents such as leases, receipts, internal regulations, etc., and reduces the use of paper.

Reminders: A system that sends automatic notifications to tenants and complex staff about important dates, such as rent due dates, lease renewals, maintenance appointments, etc. Helps avoid payment delays, facilitates task planning and improves organization.

STEP 2. Define entities

- User
- Payments
- Maintenance Requests
- Contracts_Bills
- Reminders
- Unit

STEP 3. Define attributes by entities

Entity: User

Attributes:

- ID_User (PK)
- name

- last_name
- email
- phone
- user_type
- password
- registration date

Entity: Payments

Attributes:

- ID_payment (PK)
- payment_date
- amount
- method of payment
- description
- ID_User (FK user)
- ID_Contract_bill (FK a Contract_bill)

Entity: Maintenance Requests

Attributes:

- Request_ID (PK)
- Request_Date
- Description
- Status (pending, in process, resolved)
- User_ID (FK to User)
- Unit_ID (FK to Unit)

entity: Contracts_Receipts

Attributes:

- Contract_bill_ID (PK)
- Start_date
- End_date
- Rent_amount
- User_ID (FK to User)
- Unit_ID (FK to Unit).

entity: Reminders

Attributes:

- Reminder_ID (PK)
- Reminder_date
- Description
- User_ID (FK to User)
- Reminder_type (rent payment, contract renewal, etc.).

entity: Unit

Attributes:

- Unit_ID (PK)
- Unit_number
- Unit_type (studio, one bedroom, etc.)
- Size
- Status (occupied, unoccupied).

STEP 4. Define Relationships

Workshop #1: Apartament complex aplication
Juan David Páez Olivares – 20231020028
Juan Esteban Martinez Buritica - 20222020039

🟩	***E1***	***E2***	***E3***	***E4***	***E5***	***E6***
E1	🟩	🟥	🟥	🟥	🟥	
E2	🟥	🟩		🟥		
E3	🟥		🟩			🟥
E4	🟥	🟥		🟩		🟥
E5	🟥				🟩	
E6			🟥	🟥		🟩

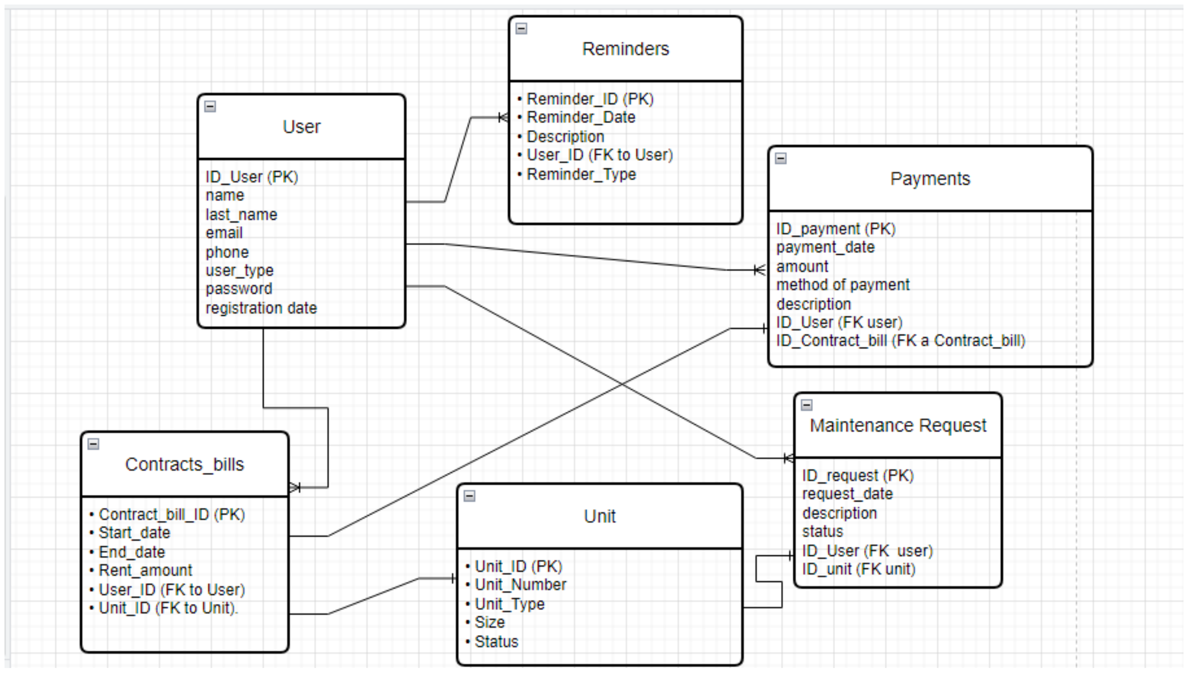
- User (e1)
- Payments (e2)
- Maintenance requests (e3)
- Contracts_Receipts (e4)
- Reminders (e5)
- Unit (e6)

STEP 5. Define relationship types

- User and Payments: A user can make many payments.
- User and Contracts_Receipts: A user can have several contracts or receipts (if he has rented different units or renewed).
- User and Maintenance requests: A user can make many maintenance requests.
- Contracts_Receipts and Unit: A contract or receipt is associated to a single unit.
- Payments and Contracts_Receipts: A payment is associated to a specific contract or receipt.

- Maintenance Requests and Unit: A maintenance request is associated to a specific unit.
- User and Reminders: A user can have many reminders.

STEP 6. First Entity-Relationship Diagram



STEP 9 and 10. Obtain ER Model Data Structure and Define Constraints and Properties of Data

Entity: User

- **ID_User** (INT, PK, AUTO_INCREMENT)
- **First_Name** (VARCHAR(50))
- **Last_Name** (VARCHAR(50))
- **Email** (VARCHAR(100))
- **Phone** (VARCHAR(15))
- **User_Type** (ENUM('tenant', 'owner', 'administrator'))

- **Password** (VARCHAR(255))
- **Registration_Date** (DATETIME)

Entity: Payments

- **ID_Payment** (INT, PK, AUTO_INCREMENT)
- **Payment_Date** (DATETIME)
- **Amount** (DECIMAL(10, 2))
- **Payment_Method** (VARCHAR(50))
- **Description** (TEXT)
- **ID_User** (INT, FK to User)
- **ID_Contract_Receipt** (INT, FK to Contracts_Receipts)

Entity: Maintenance_Requests

- **ID_Request** (INT, PK, AUTO_INCREMENT)
- **Request_Date** (DATETIME)
- **Description** (TEXT)
- **Status** (ENUM('pending', 'in process', 'resolved'))
- **ID_User** (INT, FK to User)
- **ID_Unit** (INT, FK to Unit)

Entity: Contracts_Receipts

- **ID_Contract_Receipt** (INT, PK, AUTO_INCREMENT)
- **Start_Date** (DATE)
- **End_Date** (DATE)
- **Rent_Amount** (DECIMAL(10, 2))
- **ID_User** (INT, FK to User)
- **ID_Unit** (INT, FK to Unit)

Entity: Reminders

- **ID_Reminder** (INT, PK, AUTO_INCREMENT)
- **Reminder_Date** (DATETIME)
- **Description** (TEXT)
- **ID_User** (INT, FK to User)
- **Reminder_Type** (ENUM('rent payment', 'contract renewal', 'other'))

Entity: Unit

- **ID_Unit** (INT, PK, AUTO_INCREMENT)
- **Unit_Number** (VARCHAR(10))
- **Unit_Type** (ENUM('studio', 'one bedroom', 'two bedrooms', 'more'))
- **Size** (DECIMAL(5, 2))
- **Status** (ENUM('occupied', 'vacant'))