Software Modeling I

Season 2024-III

Workshop No.1 Object- Oriented Programming

Report From:

User Histories

By: Kevin Estiven Lozano Duarte

Code: 20221020152

Eng. Carlos Andrés Sierra, M.Sc.

Computer Engineering

Universidad Distrital Francisco José de Caldas

Index

1. User Stories

Based on student responses and the minimum required functionalities, here are the user stories:

1. As a resident, I want to see a list of all the blocks and apartments, so what I can understand the layout of the complex.
2. As a resident, I want to make online payments for administrative services, so what I can conveniently manage my financial obligations.
3. As a resident, I want to make reservations for common spaces, so what I can plan social gatherings or personal activities.
4. As a resident, I want to pay rent online to avoid the hassle of writing and mailing checks.
5. As a resident, I want to report maintenance issues through the app, so what I can quickly notify management about problems in my apartment.
6. As a resident, I want to see available parking spaces in real time, so what I can easily find a spot for my vehicle.
7. As a resident, I want to reserve laundry machines so what I can guarantee availability when I need to do laundry.
8. As a resident, I want to receive notifications about package deliveries, so what I can pick them up on time.
9. As a resident, I want to have access to a community bulletin board for events and announcements so what I can stay informed about complex activities.
10. As a resident, I want to communicate with management easily through the app, so what I can get quick responses to my inquiries.
11. As a resident, I want to split bills with my roommates, so what I can fairly divide shared expenses.
12. As a resident, I want to see my lease and other documents, so what I can refer to them as needed without having to search for physical copies.
13. As a resident, I want to file noise complaints, so what I can report nuisances and maintain a quiet living environment.
14. As a resident, I want to see my apartment's energy usage statistics, so what I can monitor and potentially reduce my energy consumption.
15. As a resident, I want to apply for utility access cards, so what I can easily access the complex's facilities.
16. As a resident, I want to renew my lease through the app, so what I can conveniently extend my stay without paperwork.

17. As a resident, I want to access a marketplace to buy and sell items, so what I can transact with other residents.
18. As a resident, I want to see a calendar of maintenance schedules for my apartment, so what I can plan for upcoming maintenance work.

2. Technical and Design Considerations/Decisions

1. ***Architecture***
    - We will use a microservices architecture to ensure scalability and easier maintenance of individual functions.
2. ***Database***
    - We will use a relational database (PostgreSQL) for structured data such as user, apartment and transaction information. For real-time functions such as parking availability, we will use a NoSQL database (MongoDB) for better performance.
3. ***Authentication***
    - We will implement JWT (JSON Web Tokens) for secure authentication and authorization.
4. ***Payment integration***
    - we will use a third-party payment gateway (e.g. Stripe) to handle rental and other payments securely.
5. ***Real-time updates***
    - we will use WebSockets for real-time functions such as parking availability and community board updates.
6. ***File storage***
    - to store documents such as lease agreements, we will use cloud storage (e.g., AWS S3) for scalability and reliability.
7. ***Notifications***
    - We will implement push notifications for package deliveries and other important alerts.
8. ***Mobile responsiveness***
    - The application will be designed with a mobile focus to ensure a good user experience on all devices.
9. ***API design***
    - We will create a RESTful API for communication between frontend and backend services.
10. ***Security***
    - We will implement HTTPS for all communications and use encryption for sensitive data storage.

3.Database design (10-step methodology)

1. ***Determine the purpose of the database***

The purpose is to manage all aspects of an apartment complex, including resident information, apartments, payments, reservations, and various resident services.

2. **Find and organize the required information**

Information includes residents, apartments, blocks, payments, reservations, maintenance requests, parking spaces, packages, community events, energy usage, amenities, and leases.

3. ***Divide the information into tables.***

Tables: Users, Apartments, Blocks, Payments, Reservations, Maintenance requests, Parking spaces, Packages, Community events, Energy use, Amenities, Leases.

4. ***Convert data elements into columns***

Example for the Users table: User ID, First Name, Last Name, Email, Phone, Apartment ID

5. ***Specify primary keys***

Example: UserID for table Users, ApartmentID for table Apartments

6. ***Establish table relationships***

Example: Users to apartments (one to many), apartments to blocks (many to one)

7. ***Refine the layout***

Add indexes for frequently searched columns

Normalize tables to reduce data redundancy

8. **Apply normalization rules**

Ensure tables are at least in 3NF (third normal form)

9. ***Create the physical database design***

Define data types for each column

Set constraints (e.g. NOT NULL, UNIQUE)

10. ***Create views and implement security mechanisms***

Create views for common queries (e.g. ApartmentDetails, UserPayments)

Implement role-based access control