



*Universidad de las Fuerzas Armadas
'ESPE'*



Group 3

Condominium Management System

Software Engineering

Team members: Klever Jami,
Gabriel Manosalvas,
José Leiton
Alexander Maza,

Fecha: 09/12/2024

Condominium Management System: A Comprehensive Solution

This presentation outlines the design and functionalities of a condominium management system. We will delve into the system's architecture, key classes, functionalities, and future enhancements. Our goal is to provide a user-friendly and efficient solution for managing various aspects of condominium operations.

KJ



Project Objectives: Streamlining Condominium Management

Comprehensive Data Management

The system centralizes essential condominium data (resident info, vehicles, payments, utilities, reservations), streamlining operations and eliminating disparate systems.

CRUD-Based Data Management

A robust CRUD framework allows administrators to easily add, modify, and delete resident, vehicle, payment, and other key data.

Condominium Management System Development



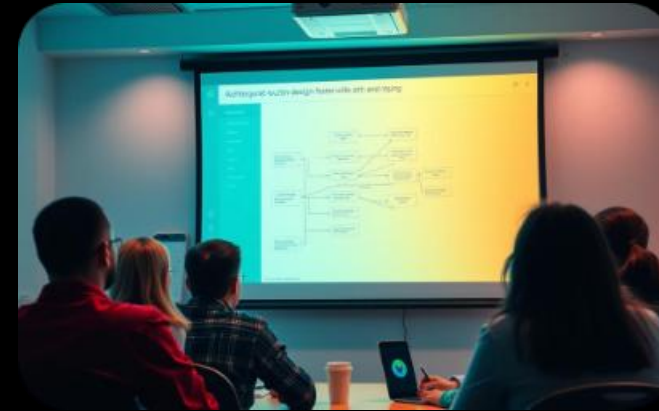
1. Requirements Definition

Gathering condominium needs.



2. Analysis

Identification of key classes and relationships.



3. Design

Creation of a complete UML diagram.



4. Implementation

Java development of the system.



5. Testing

Unit and integration testing.



6. Maintenance

Continuous updates and improvements.

Condominium Management Platform: Requirements Overview

System Goal

To provide a comprehensive platform for managing condominiums, offering convenient and efficient tools for both residents and administrators.

Resident Features

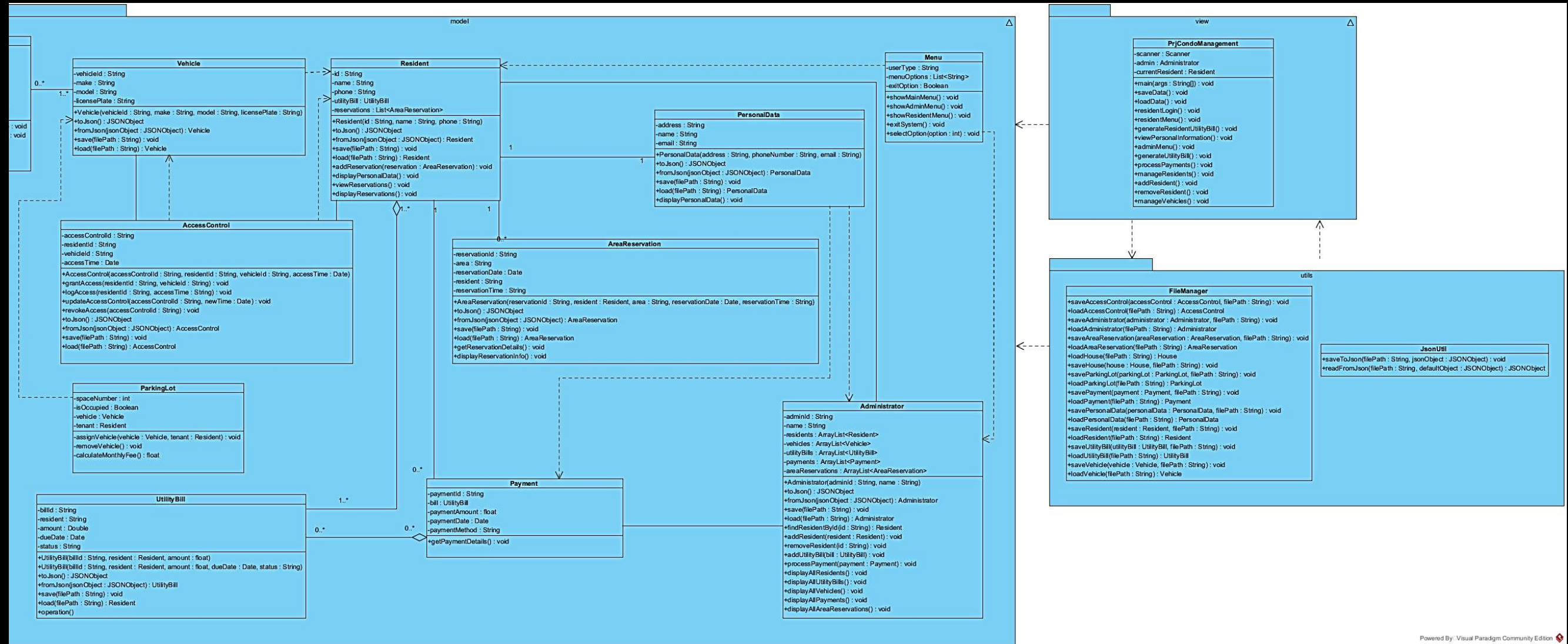
- Bill payments
- Common area reservations
- Access management
- Personal information access

Administrator Features

- Resident, vehicle, reservation, and payment management
- Access control
- Payment and bill status monitoring

UML System Diagram

UML Diagram Explanation:



System Architecture: A Modular Modular Design

Model

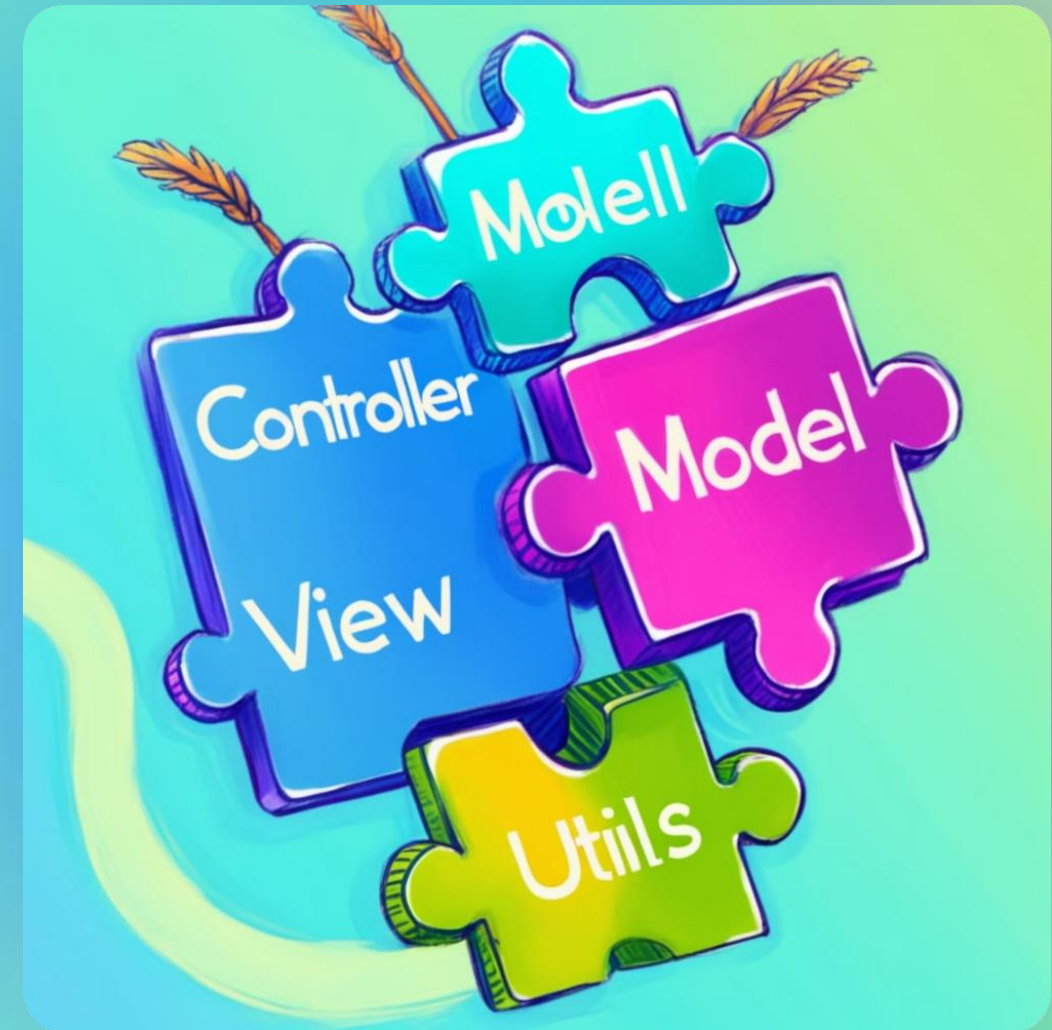
The Model package contains core entity classes (Resident, Vehicle, Payment, etc.), defining data structure and behavior.

View

The View package's Menu class provides the user interface for system interaction and data management.

Utils

The Utils package's FileManager handles efficient JSON data persistence and retrieval.



Classes

Resident

Represents residents with attributes like ID, name, and phone number. Manages resident details, including vehicle registration and billing information.

Vehicle

Represents residents' vehicles with attributes like ID, make, model, and plate number. Facilitates associating vehicles with residents and managing parking assignments.

Payment

Manages payments made by residents, including payment ID, amount, date, and method. Tracks payment history and invoice reconciliation.

Utility Bill

Represents service invoices issued to residents, including invoice ID, resident, amount, due date, and payment status. Enables efficient invoice generation and tracking.

Classes

Access Control

Manages the access of residents and vehicles to the condominium. Attributes: accessControllId, residentId, vehicleId, accessTime. Methods: Grant, register, and revoke access to condominium areas.

Parking Lot

Manages the parking spaces of the condominium. Attributes: spaceNumber, isOccupied, vehicle, tenant. Methods: Assign or release parking spaces.

Area Reservation

Manages reservations of common areas. Attributes: reservationId, resident, area, reservationDate, reservationTime. Methods: Make and manage reservations.

House

Represents the houses in the condominium. Attributes: houseNumber, owner, rooms, bathrooms, price. Methods: Manage house information.

Classes

Administrator

Represents the condominium administrator.

Attributes: adminId, name. Methods: Manage residents, vehicles, payments, etc.

Personal Data

Manages the personal data of the resident.

Attributes: address, phone, email. Methods: Update and manage personal information.

Main System Flow

User Interaction

1. System presents a main menu with options:
 - Login as administrator
 - Login as resident
 - Load/save data from/to JSON files
 - Exit
2. Login:
 - Administrator: Accesses options for managing the system (residents, vehicles, invoices, payments)
 - Resident: Accesses their personal data and can generate invoices.



Core Methods in PrjCondoManagement Class

1

Attributes

scanner: Used to read user input.

admin: Instance of Administrator object, representing the administrator.

currentResident: Maintains the current resident's session.

2

Methods

getIntInput(): Reads and validates user input, ensuring it is an integer.

main(): Displays the main menu with options like login, save data, load data, or exit.

method call

Administrator Menu



Manage Residents

Add, remove, and view resident details.



Manage Vehicles

Add and manage vehicle registrations.



Generate Utility Bills

Create and manage invoices.



Process Payments

Record and track resident payments.



Manage Area Reservations

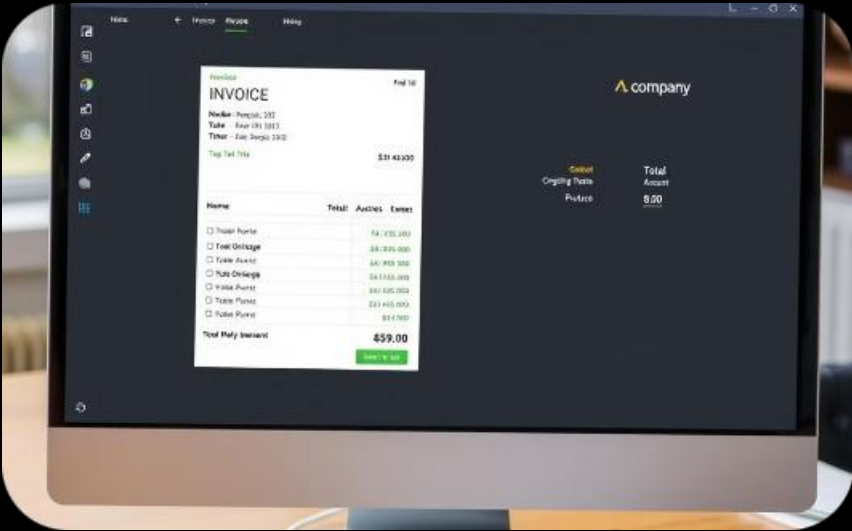
Control reservations of common areas.

Resident Menu



View Personal Information

Check personal details like name, phone, etc.



Generate Utility Bills

Residents can generate bills for services used.



View Bills and Payments

See generated bills and payment history.

File Management (Data Persistence)



Saving Data

Securely store project information in a persistent format.



Loading Data

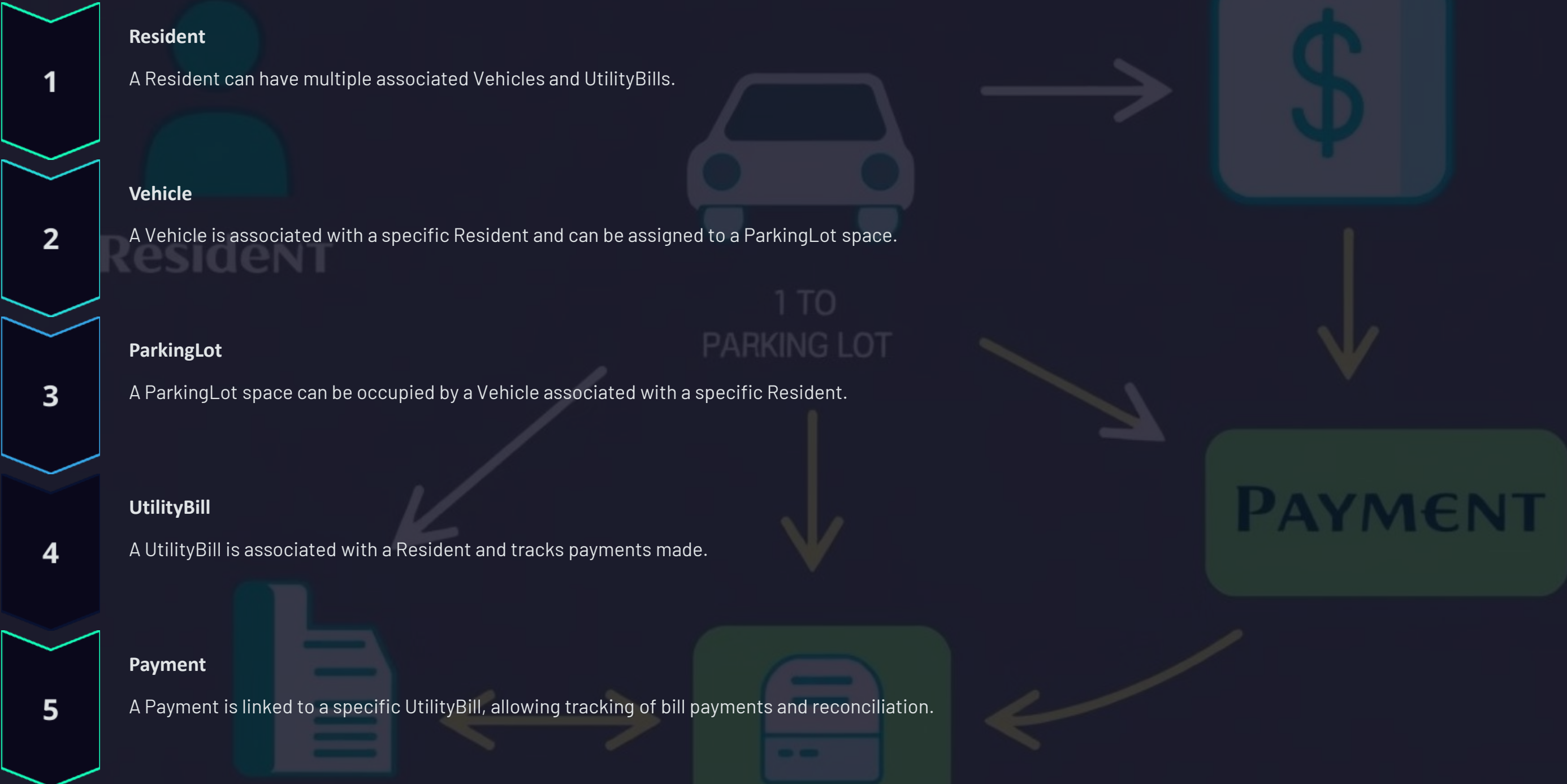
Retrieve data quickly and efficiently.



Data Security

Protect sensitive information with robust security measures.

Class Relationships: Connecting the Dots



System Functionalities: Empowering Management

1 Authentication

The system provides secure login for both residents and administrators, ensuring access control and data security.

2 Resident Management

Administrators can easily add, remove, or modify resident details, including personal information, vehicle registrations, and billing data.

3 Vehicle Management

Administrators can assign and remove vehicles from resident accounts, manage parking lot assignments, and track vehicle access records.

4 Billing and Payments

The system generates utility bills automatically and allows residents to make payments, track payment history, and view invoice details.



JSON File Management: Data Persistence and Retrieval

1

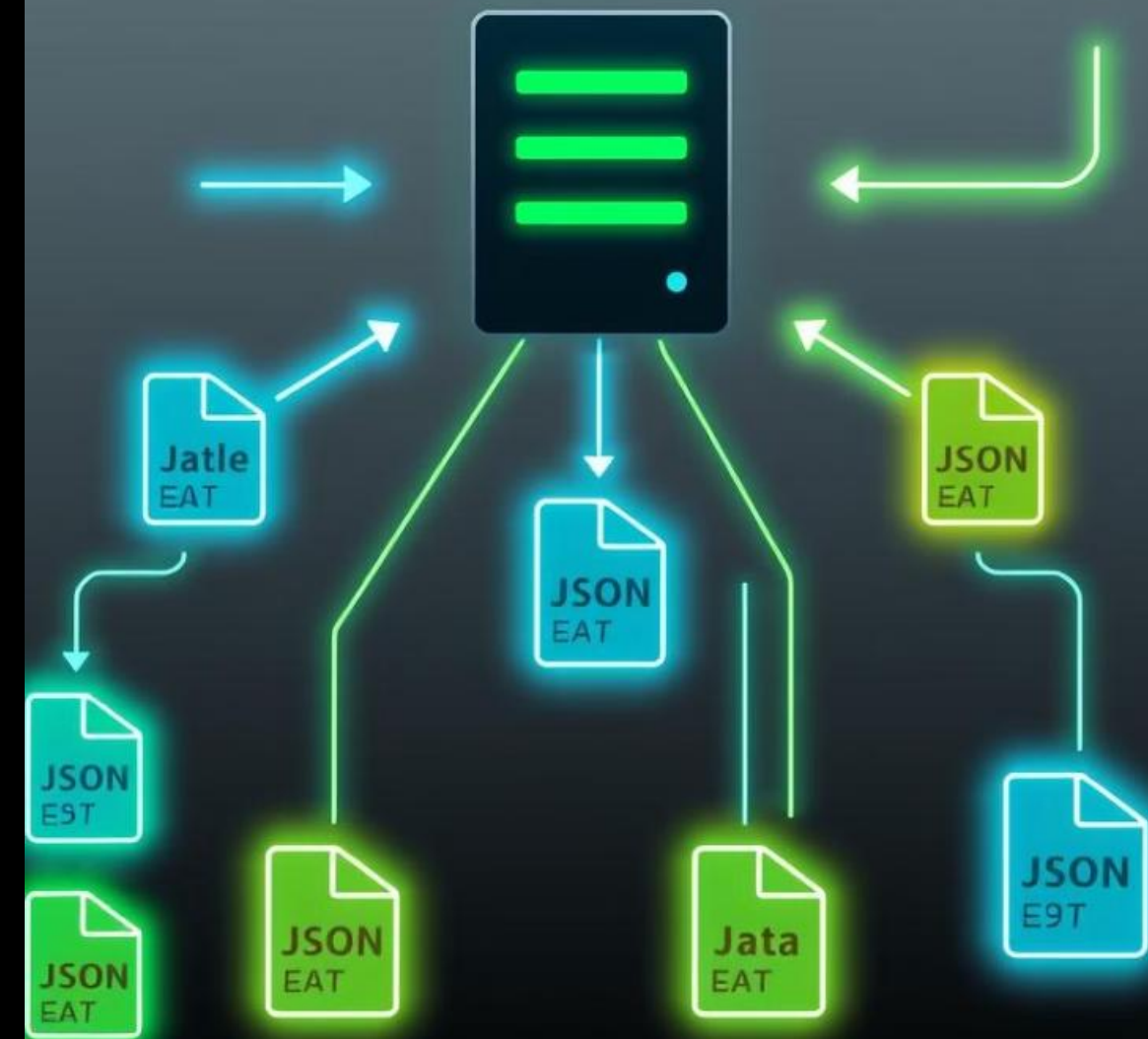
Data Serialization

When data is saved, the system serializes entity data into JSON format, which is then written to individual files. This ensures data consistency and ease of storage.

2

Data Deserialization

Upon system startup, the FileManager class reads the JSON files for each entity and deserializes the data, reconstructing the objects in memory. This allows for fast data loading.

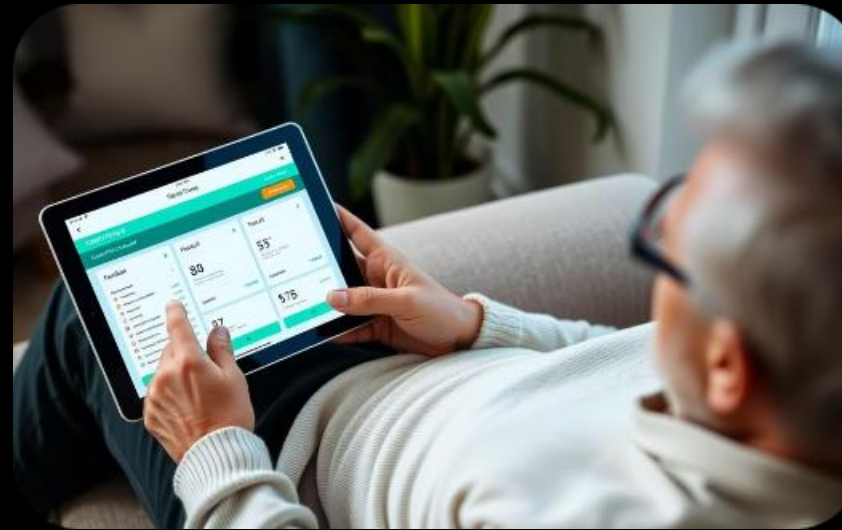


User Interaction



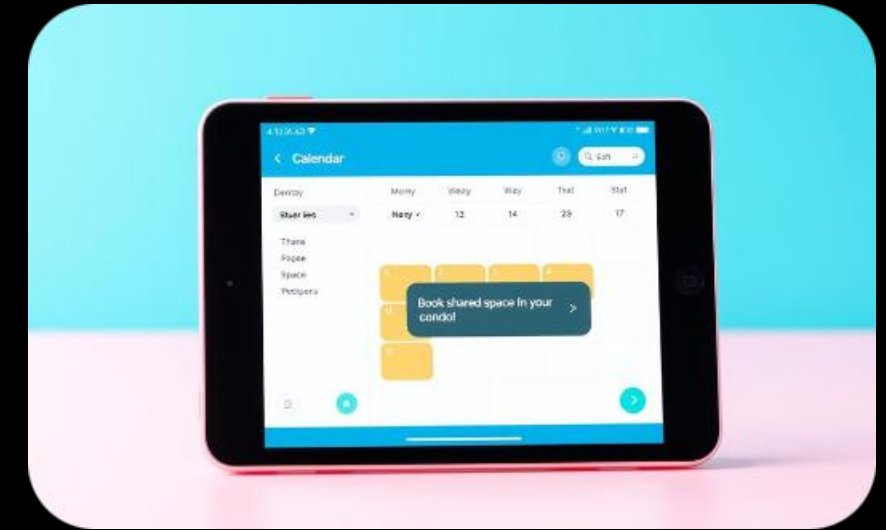
Administrator

Administrators gain access to a comprehensive menu for managing residents, vehicles, bills, payments, parking, and reservations.



Resident

Residents can view their personal information, generate utility bills, track payment history, and make payments.



Reservations

Residents can easily reserve common areas for events or meetings.

Future Enhancements: Expanding Capabilities

