Architecture Description of
Layered Architecture (Three-Tier Architecture) for
CondoCare Management System

Contents

Contents......2
1 Introduction...... 6
1.1 Identifying information...... 6
1.2 Supplementary information......6
1.3 Other information...... 6
1.3.1 Overview...... 6
1.3.2 Architecture evaluations...... 6
1.3.3 Rationale for key decisions......6
2 Stakeholders and concerns...... 7
3 Viewpoints...... 9
3.1 Profile Management Viewpoint:......9
3.2 Overview
This viewpoint focuses on the efficient management of profiles for public users, including condo owners and rental users...... 9
3.3 Concerns and Stakeholders...... 9
3.3.1 Concerns
Efficient profile management for condo owners and rental users......9
3.3.2 Typical Stakeholders...... 9
3.4 Model kinds...... 9
3.5 User Profile Model:......10
3.5.1 User Profile Model conventions:......10
3.5.2 User Profile Model correspondence rules:...... 10
3.6 Operations on views......10
3.2 Overview......11
3.3 Concerns and Stakeholders...... 11
3.3.1 Concerns...... 11
3.3.2 Typical Stakeholders......11
3.4 Model kinds...... 11
3.5 Property Model:...... 11
3.5.1 Property Model conventions:...... 11
3.5.2 Property Model correspondence rules:...... 12
3.5 Financial Overview Model:...... 12
3.5.1 Financial Overview Model conventions:...... 12
3.5.2 Financial Overview Model correspondence rules:......12
3.6 Operations on views......12
3.2 Overview...... 13
3.3 Concerns and Stakeholders......13
3.3.1 Concerns...... 13
3.3.2 Typical Stakeholders...... 13
3.4 Model kinds...... 14
3.5 Property Profile Model:......14
3.5.1 Property Profile Model conventions:......14
3.5.2 Property Profile Model correspondence rules:...... 14

3.5 Condo Files Model:... 15
3.5.1 Condo Files Model conventions:... 15
3.5.2 Condo Files Model correspondence rules:...15
3.5 Unit Information Model:.. 15
3.5.1 Unit Information Model conventions:.. 15
3.5.2 Unit Information Model correspondence rules:...15
3.6 Operations on views..15
3.2 Overview.. 16
3.3 Concerns and Stakeholders...16
3.3.1 Concerns.. 16
3.3.2 Typical Stakeholders.. 16
3.4 Model kinds.. 17
3.5 Condo Fees Calculation Model:.. 17
3.5.1 Condo Fees Calculation Model conventions:... 17
3.5.2 Condo Fees Calculation Model correspondence rules:.....................................17
● Correspondence rules ensure accurate links between fee components and their representation in the model...17
3.5 Operational Budget Model:..18
3.5.1 Operational Budget Model conventions:... 18
3.5.2 Operational Budget Model correspondence rules:.. 18
3.5 Costs Recording Model:.. 18
3.5.1 Costs Recording Model conventions:...18
3.5.2 Costs Recording Model correspondence rules:... 18
3.5 Annual Financial Reports Model:.. 18
3.5.1 Annual Financial Reports Model conventions:.. 18
3.5.2 Annual Financial Reports Model correspondence rules:...................................18
3.6 Operations on views..19
3.2 Overview.. 19
3.3 Concerns and Stakeholders...19
3.3.1 Concerns.. 19
3.3.2 Typical Stakeholders.. 19
3.5 Reservation System Model:.. 20
3.5.1 Reservation System Model conventions:.. 20
3.5.2 Reservation System Model correspondence rules:...20
3.6 Operations on views..20
3.2 Overview.. 21
3.3 Concerns and Stakeholders...21
3.3.1 Concerns..21
3.3.2 Typical Stakeholders.. 21
3.4 Model kinds.. 21
3.5 Employee Roles Model:.. 21
3.5.1 Employee Roles Model conventions:.. 21
3.5.2 Employee Roles Model correspondence rules:...22
3.5 Permissions Model:...22

3.5.1 Permissions Model conventions:..22
3.5.2 Permissions Model correspondence rules:.. 22
3.6 Operations on views...22
3.2 Overview.. 23
3.3 Concerns and Stakeholders...23
3.3.1 Concerns..23
3.3.2 Typical Stakeholders.. 23
3.4 Model kinds.. 24
3.5 Scalability Assessment Model:...24
3.5.1 Scalability Assessment Model conventions:...24
3.5.2 Scalability Assessment Model correspondence rules:.................................... 24
3.5 User Interface Model:...25
3.5.1 User Interface Model conventions:...25
3.5.2 User Interface Model correspondence rules:.. 25
3.6 Operations on views...25
3.2 Overview.. 25
3.3 Concerns and Stakeholders...26
3.3.1 Concerns..26
3.3.2 Typical Stakeholders.. 26
3.4 Model kinds.. 26
3.5 Technical Feasibility Model:..26
3.5.1 Technical Feasibility Model conventions:.. 26
3.5.2 Technical Feasibility Model correspondence rules:..26
3.5 Operational Feasibility Model:..26
3.5.1 Operational Feasibility Model conventions:...26
3.5.2 Operational Feasibility Model correspondence rules:...................................... 26
3.6 Operations on views...26
3.2 Overview.. 27
3.3 Concerns and Stakeholders...27
3.3.1 Concerns..27
3.3.2 Typical Stakeholders.. 27
3.4 Model kinds.. 27
3.5 Risk Identification Model:... 27
3.5.1 Risk Identification Model conventions:... 27
3.5.2 Risk Identification Model correspondence rules:...28
3.5 Risk Mitigation Model:.. 28
3.5.1 Risk Mitigation Model conventions:..28
3.5.2 Risk Mitigation Model correspondence rules:..28
3.6 Operations on views...28
3.2 Overview.. 29
3.3 Concerns and Stakeholders...29
3.3.1 Concerns..29
3.3.2 Typical Stakeholders.. 29
3.4 Model kinds.. 29

3.5 Maintenance Plan Model:..........29
3.5.1 Maintenance Plan Model conventions:..........29
3.5.2 Maintenance Plan Model correspondence rules:..........29
3.5 Evolution Plan Model:..........29
3.5.1 Evolution Plan Model conventions:..........29
3.5.2 Evolution Plan Model correspondence rules:..........30
3.5 User Feedback Model:..........30
3.5.1 User Feedback Model conventions:..........30
3.5.2 User Feedback Model correspondence rules:..........30
3.5 Technological Advancements Model:..........30
3.5.1 Technological Advancements Model conventions:..........30
3.5.2 Technological Advancements Model correspondence rules:..........30
3.5 Feature Evolution Model:..........30
3.5.1 Feature Evolution Model conventions:..........30
3.5.2 Feature Evolution Model correspondence rules:..........31
3.6 Operations on views..........31
3.7 Correspondence rules..........31

1 Introduction

This chapter describes introductory information items of the AD, including identifying and supplementary information.

1.1 Identifying information

Architecture Name: Layered Architecture (Three-Tier Architecture)

System of Interest: Condo Management System

1.2 Supplementary information

- **Date of Issue and Status:** This document is a secondary version (v2.0) drafted on March 20th 2024 .
- **Authors:** Aly Hussein -
- **Project Scope:** This document outlines the high-level architecture for the core functionalities of the condo management system. Future versions will detail specific technologies, security measures, and database schema design.
- **Version Control Information:** This document is stored in a version control system to track changes and facilitate collaboration. Link of git
- **Glossary:** As the project progresses, a glossary of key terms used in the architecture description is going to be appended to the end (e.g., API Gateway, Single Sign-On).

1.3 Other information

1.3.1**. System Overview**

This document outlines the initial high-level architecture for the condo management system, encompassing a mobile app (Android & iOS) and its companion website.

The system caters to three main user groups:

- **Public Users:** Can create profiles and view public information. Require a registration key to become condo owners or rental users.
- **Condo Owners:** Can view property dashboards, access condo documents, submit requests, manage reservations, and potentially access additional features like a forum and event management (if implemented).
- **Condo Management Companies:** Manage properties, define condo fees, access financial reports, manage reservations, assign user roles, and handle resident requests.

The core functionalities include:

- User Management: Profile creation, registration key system for condo owners/rental users, employee account management with role-based access.

- Property Management: Property profile creation, unit/parking/locker management, document storage for condo owners.
- Financial Management: Condo fee definition, cost recording, automatic condo fee calculation, annual financial report generation.
- Reservation System: Management of reservable common facilities, calendar-based reservation interface, and real-time availability tracking.
- Request Management: Ability for condo owners to submit requests categorized by type, routing requests to corresponding employees, and notification system for request updates.

1.3.2. **Reader's Guide**

This document is intended for software developers and other stakeholders involved in the condo management system project. It provides a foundational understanding of the overall architecture.

- Section 3: System Components - Explains the system's breakdown into presentation layer (mobile app & web app), business logic layer (API Gateway & services), and data layer (database).
- Section 4: User Roles and Access Control - Defines user roles and their corresponding access levels within the system.
- Section 5: System Functionality Breakdown - Details the core functionalities and optional features of the condo management system.
- Section 6: Bonus Features (if implemented) - Discusses potential enhancements like cross-platform mobile app accessibility, multi-language support, and Single Sign-On.

1.3.3. **Rationale for Key Decisions**

- **Layered Architecture:** This approach promotes separation of concerns, allowing independent development and maintenance of the presentation layer, business logic, and data storage.
- **Relational Database:** A relational database like MySQL or PostgreSQL offers efficient data storage and retrieval for user profiles, property details, financial information, reservations, and request history.
- **Secure Authentication:** JWT tokens are a secure mechanism for user authentication, ensuring only authorised users can access system functionalities.

1.3.4. **Next Steps**

This document provides a preliminary architecture overview. Subsequent versions will delve deeper into specific details:

- Technology Selection: Specific frameworks and technologies will be chosen for the mobile app development, web application development, and API development.

- Security Measures: A comprehensive security plan will be established to address data security, user authentication, and authorization.
- Database Schema Design: The detailed structure of the database tables and their relationships will be defined.

This architecture description serves as a blueprint for the condo management system development. By outlining the core functionalities, user roles, and architectural approach, it facilitates clear communication and collaboration among stakeholders as the project progresses.

2 Stakeholders and concerns

This chapter contains information items for stakeholders of the architecture, the stakeholders' concerns for that architecture, and the traceability of concerns to stakeholders.

2.1 Stakeholders

The condo management system architecture caters to a variety of stakeholders with distinct interests and concerns:

- **Primary Users:**
    - **Condo Owners:** Their primary concern is user-friendliness and accessibility of features like property dashboards, document access, reservation management, and request submission. They also have an interest in data security and privacy regarding their personal information.
    - **Rental Users:** Their concerns are similar to condo owners, with a potential focus on features relevant to their specific needs within the condo community.
    - **Condo Management Company Staff:** Their concerns revolve around efficient management functionalities, including property management tools, financial reporting, user role assignment, and request handling. They also have a vested interest in system security and scalability to accommodate potential growth in users and data.
- **Secondary Users:**
    - **Public Users:** Their concerns are limited as they can only view public information. However, a user-friendly registration process might be of interest if they intend to become condo owners or rental users.
- **System Developers:** Their concerns involve the technical feasibility and maintainability of the chosen architecture. They are interested in selecting appropriate technologies (frameworks, databases) that ensure smooth development and future system updates.
- **Project Managers and Stakeholders:** Their concerns are broad, encompassing project success, budget adherence, and system scalability to

meet evolving needs. They value a clear architecture that facilitates efficient development and long-term system maintenance.

- **IT Operations (if applicable):** If the system is deployed within a dedicated IT infrastructure, IT operations staff would have a stake in system performance, reliability, and disaster recovery capabilities.

Understanding the needs and concerns of each stakeholder group is crucial for creating an architecture that effectively addresses their requirements. Future iterations of this architecture description should consider incorporating specific user interface (UI) design considerations for different user groups to optimise their experience within the system.

2.2 Concerns

Purpose(s) of the System-of-Interest:

- Enable public users (condo owners and rental users) to manage their profiles efficiently.
- Provide condo owners with a comprehensive view of their properties, financial status, and submitted requests.
- Allow condo management companies to create and manage property profiles, upload condo files, and enter detailed information for each unit, parking spot, and locker.
- Facilitate a simplified financial system, including the calculation and presentation of condo fees, recording of operational budget and costs, and generation of annual financial reports.
- Incorporate a reservation system for common facilities, allowing users to book facilities with ease.
- Support different roles for employees within condo management companies, such as managers responsible for daily operations or finance.

Suitability of the Architecture:

This section outlines the key concerns that have been considered in shaping the architecture of the condo management system:

- **System Purpose and Functionality:**
  - The primary purpose of the system is to provide a user-friendly platform for condo management, encompassing functionalities for residents, condo management companies, and potentially public users.
  - The chosen layered architecture with separate presentation, business logic, and data layers ensures modularity and facilitates the implementation of the core functionalities, including user management, property management, financial management, reservation systems, and request management.
- **Suitability and Feasibility:**
  - The layered architecture promotes independent development and maintenance of each layer, enhancing the system's maintainability and scalability in the long run.

    - The selection of a relational database like MySQL or PostgreSQL offers a well-established and robust solution for storing and managing the diverse data types within the system (user profiles, property details, financial records, reservations, requests).
    - Utilizing secure authentication mechanisms like JWT tokens safeguards sensitive user data and ensures authorized access to system functionalities.
- **Risks and Impacts:**
    - **Security Risks:** Data breaches, unauthorized access to user information, or manipulation of financial data are potential security concerns. The architecture prioritizes secure authentication and data encryption to mitigate these risks.
    - **Scalability Challenges:** As the condo community grows, the system might need to accommodate an increasing number of users and data. The chosen architecture allows for horizontal scaling (adding more servers) to address potential scalability issues.
    - **Usability Concerns:** A complex or unintuitive user interface could hinder user adoption. The architecture promotes a modular approach that facilitates the development of user-friendly interfaces for different user groups.
- **Maintainability and Evolution:**
    - The layered architecture promotes easier maintenance by allowing modifications to specific layers without impacting the entire system.
    - The use of well-established technologies and frameworks simplifies future system updates and integration of new features.

By addressing these fundamental concerns, the condo management system architecture strives to achieve a balance between functionality, maintainability, security, and scalability to meet the evolving needs of the condo community.

2.3 Concern–Stakeholder Traceability

Table 2.1: Table Showing the association of stakeholders to concerns in an AD

Stakeholder	Concerns
Condo Owners, Rental Users	Usability Concerns
Condo Owners	Data security and privacy regarding their personal information
Condo Management Company Staff	System security and scalability to accommodate potential growth in users and data
Condo Management Company Staff, System Developers	Feasibility to construct and deploy the system-of-interest

Project Managers and Stakeholders	System scalability to meet evolving needs, Budget adherence
IT Operations (if applicable)	System performance, reliability, and disaster recovery capabilities

3 Viewpoints

3.1 Functional Viewpoint

Synonyms/Other Common Names: Behavioural Viewpoint

3.2 Overview

Key Features

- **User Focus:** This viewpoint prioritizes the user experience by examining system functionalities from the perspective of condo owners, rental users, and potentially public users.
- **Functionality Decomposition:** It breaks down the system's functionalities into smaller, manageable components, facilitating an understanding of how individual functionalities interact to achieve the overall system goals.
- **Usability Analysis:** By analyzing user interactions and data flows, this viewpoint helps identify potential usability issues and ensures the system caters to the needs of different user groups.
- **Alignment with Architecture:** It verifies that the chosen architecture can effectively support the functionalities identified for the condo management system.

This viewpoint utilizes data flow diagrams (DFDs) and use cases as the primary modeling tools to represent user interactions and system responses.

3.3 Concerns and Stakeholders

The functional viewpoint in the condo management system architecture description (SAD) is specifically intended to address the needs of the following stakeholders and their corresponding concerns:

3.3.1 Concerns

- **Usability Concerns (Condo Owners, Rental Users):** By analyzing user interactions and data flows, we can identify potential usability issues and ensure the system is user-friendly for different user groups.

- **System Purpose and Functionality (Project Managers and Stakeholders):** This viewpoint helps confirm that the architecture can deliver the essential functionalities required for the condo management system, such as user management, property management, financial management, reservation systems, and request management.

3.3.2 Typical Stakeholders**:**

- **Condo Owners:** Their primary concern is usability. The functional viewpoint helps ensure the system offers a user-friendly experience for functionalities like property dashboard access, document management, reservation management, and request submission.
- **Rental Users:** Similar to condo owners, usability is a key concern for rental users. This viewpoint ensures they can easily navigate features relevant to their needs within the condo community.
- **Project Managers and Stakeholders:** Their broader concern is system purpose and functionality. The functional viewpoint helps verify that the chosen architecture effectively supports the core functionalities identified for the condo management system, aligning with the overall project goals.

By focusing on these stakeholders and their concerns, the functional viewpoint plays a vital role in shaping a user-centric and functionally sound architecture for the condo management system.

3.4 Model kinds

For Viewpoint 1: Functional Viewpoint in the condo management system architecture description (SAD), we will utilise two key model kinds:

1. **Data Flow Diagram (DFD)**
2. **Use Case**

We will delve into the specifics of each model kind in separate sections, following the guidelines set forth in ISO/IEC/IEEE 42010.

3.5 **Data Flow Diagram (DFD)**

3.5.1 User Profile Model conventions:

- The User Profile Model utilizes the Unified Modeling Language (UML) notation to represent the various components of a user profile.
- Conventions also involve the use of stereotypes within the UML notation to distinguish between different types of users and their respective profile characteristics.

- Analytical methods may include evaluating the user profile model for completeness, consistency, and conformity to predefined standards.

3.5.2 User Profile Model correspondence rules:

- Correspondence rules dictate that each user profile instance must be associated with a unique username and password combination.
- The correspondence between the User Profile Model and other related models, such as the Employee Roles Model, should be defined to establish the permissions and access levels associated with each user profile.
- Changes in user roles or permissions should be reflected in the User Profile Model, ensuring synchronization with the Employee Roles Model.

3.6 Operations on views

Construction Methods:

- Construction methods involve providing process guidance for efficiently managing profiles for public users, including condo owners and rental users.
- Templates for constructing user profiles, considering different user types, and capturing necessary information.

Interpretation Methods:

- Interpretation methods guide readers in understanding the constructed user profiles, their attributes, and the distinctions between various user types.
- Heuristic guidance for interpreting patterns or styles used in synthesizing user profiles.

Analysis Methods:

- Analysis methods are applied to check the completeness, consistency, and conformity of user profiles.
- Model correspondence rules are checked to ensure accurate associations between user profiles and related entities.

Implementation Methods:

- Implementation methods focus on the design and integration of the user profile model into the broader system.
- Guidance on incorporating user profiles into the overall system design and functionality.

Property and Financial Overview Viewpoint:

3.2 Overview

This viewpoint provides a comprehensive view of properties, financial status, and submitted requests for condo owners.

3.3 Concerns and Stakeholders

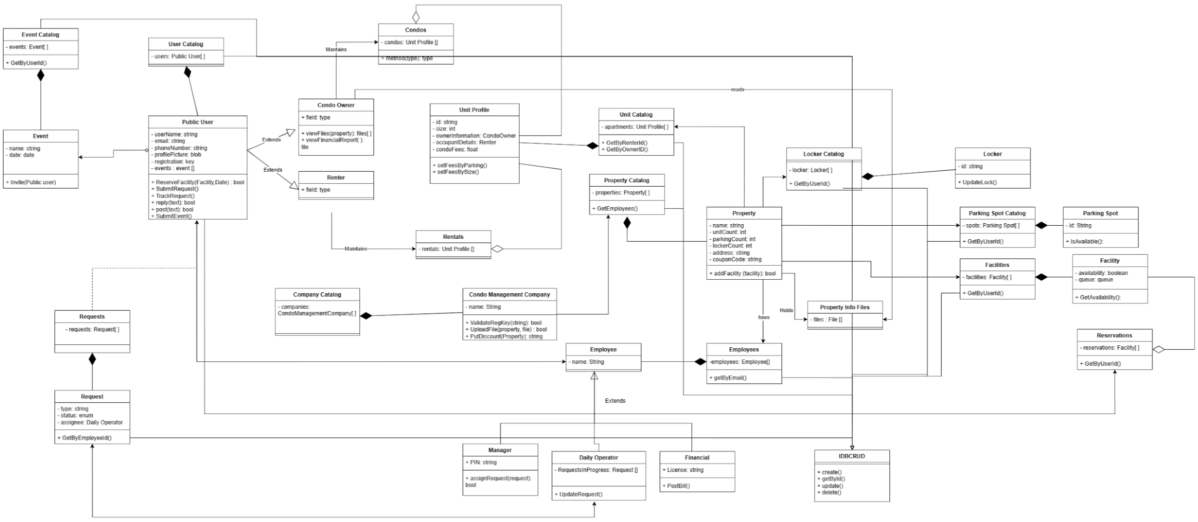
3.3.1 Concerns

Comprehensive property and financial overview.

3.3.2 Typical Stakeholders

Condo owners

3.4 Model kinds



3.5 Property Model:

3.5.1 Property Model conventions:

- The Property Model employs UML class diagrams to represent the structure of properties, financial status, and submitted requests.
- Conventions include defining classes for properties, relationships for financial status, and associations for submitted requests.

3.5.2 Property Model correspondence rules:

- Correspondence rules ensure that each property instance is associated with accurate financial status information.
- Changes in property details or financial status should be reflected in the Property Model to maintain data integrity.

3.5 Financial Overview Model:

3.5.1 Financial Overview Model conventions:

- The Financial Overview Model utilizes UML notation, focusing on class diagrams to represent financial components.
- Conventions involve defining classes for condo fees, operational budget, costs, and annual financial reports.

3.5.2 Financial Overview Model correspondence rules:

- Correspondence rules ensure accurate links between financial components and their representation in the model.
- Changes in financial calculations or reporting should be reflected in the Financial Overview Model.

3.6 Operations on views

Construction Methods:

- Construction methods involve process guidance for creating a comprehensive view of properties, financial status, and submitted requests for condo owners.
- Templates for representing property and financial information.

Interpretation Methods:

- Interpretation methods guide readers in understanding the relationships between properties, financial status, and submitted requests.
- Heuristic guidance for interpreting patterns or styles used in synthesizing property and financial overviews.

Analysis Methods:

- Analysis methods are applied to check the consistency of property and financial information.
- Model correspondence rules are checked to ensure accurate associations between properties and financial components.

Implementation Methods:

- Implementation methods focus on designing and implementing the property and financial overview model into the broader system.
- Guidance on incorporating property and financial information into the overall system design and functionality.

Property Management Viewpoint:

3.2 Overview

This viewpoint focuses on creating and managing property profiles, uploading condo files, and entering detailed information for each unit, parking spot, and locker.

3.3 Concerns and Stakeholders

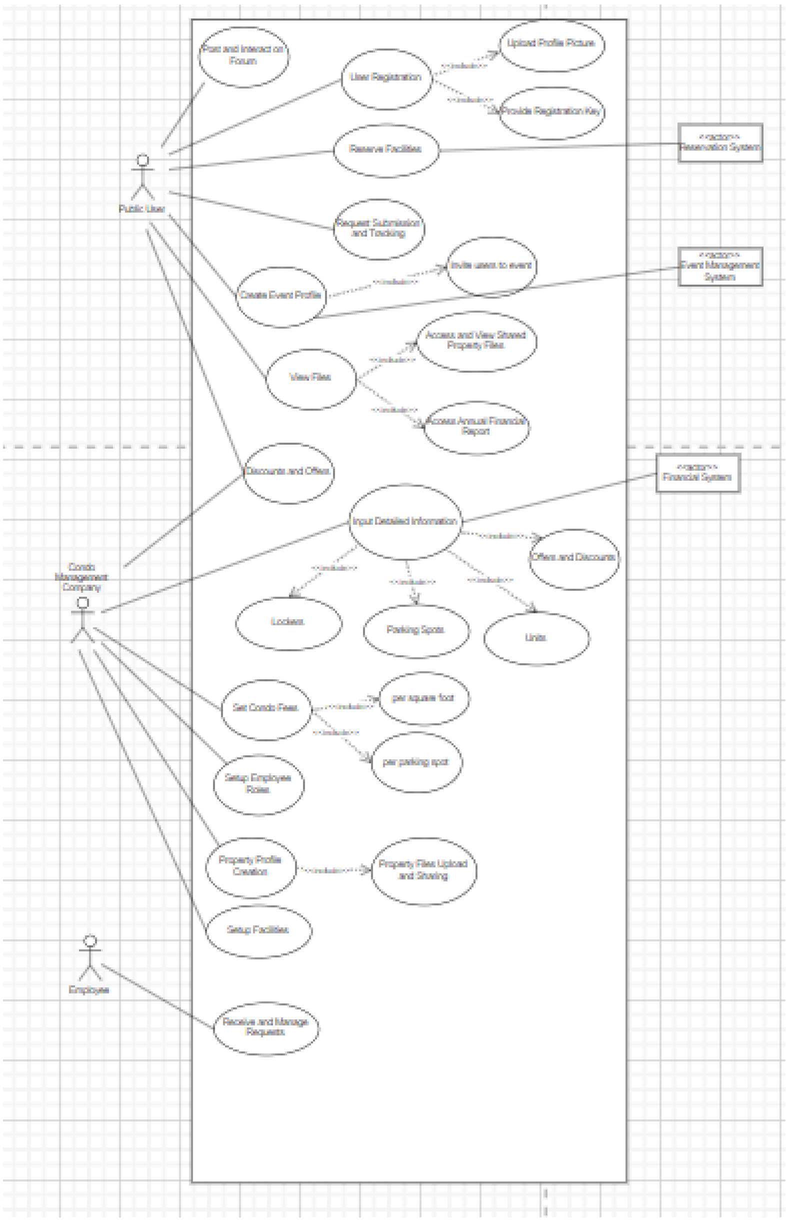
3.3.1 Concerns

Efficient property management.

3.3.2 Typical Stakeholders

Condo management companies

3.4 Model kinds



3.5 Property Profile Model:

3.5.1 Property Profile Model conventions:

- The Property Profile Model uses UML class diagrams to depict the structure of property profiles, condo files, unit information, and related details.
- Conventions include defining classes for property profiles, associations for condo files, and attributes for unit information.

3.5.2 Property Profile Model correspondence rules:

- Correspondence rules ensure that each property profile is associated with accurate condo files and unit information.
- Changes in property details, condo files, or unit information should be reflected in the Property Profile Model.

3.5 Condo Files Model:

3.5.1 Condo Files Model conventions:

- The Condo Files Model employs UML class diagrams to represent the structure of files associated with condos.
- Conventions include defining classes for file details, associations for related condos, and attributes for file information.

3.5.2 Condo Files Model correspondence rules:

- Correspondence rules ensure accurate associations between condo files and related condos.
- Changes in file details or associations should be reflected in the Condo Files Model.

3.5 Unit Information Model:

3.5.1 Unit Information Model conventions:

- The Unit Information Model uses UML class diagrams to represent the structure of detailed information for each unit.
- Conventions include defining classes for unit details, attributes for specifications, and associations for related properties.

3.5.2 Unit Information Model correspondence rules:

- Correspondence rules ensure that each unit instance is associated with accurate details and related property information.
- Changes in unit specifications or associations should be reflected in the Unit Information Model.

3.6 Operations on views

Construction Methods:

- Construction methods involve process guidance for creating and managing property profiles, uploading condo files, and entering detailed information for each unit, parking spot, and locker.
- Templates for representing property profiles, condo files, and unit information.

Interpretation Methods:

- Interpretation methods guide readers in understanding the relationships between property profiles, condo files, and unit information.
- Heuristic guidance for interpreting patterns or styles used in synthesizing property management information.

Analysis Methods:

- Analysis methods are applied to check the efficiency of property management processes.
- Model correspondence rules are checked to ensure accurate associations between property profiles, condo files, and unit information.

Implementation Methods:

- Implementation methods focus on designing and implementing the property management model into the broader system.
- Guidance on incorporating property management information into the overall system design and functionality.

Financial System Viewpoint:

3.2 Overview

This viewpoint facilitates a simplified financial system, including the calculation and presentation of condo fees, recording of operational budget and costs, and generation of annual financial reports.

3.3 Concerns and Stakeholders

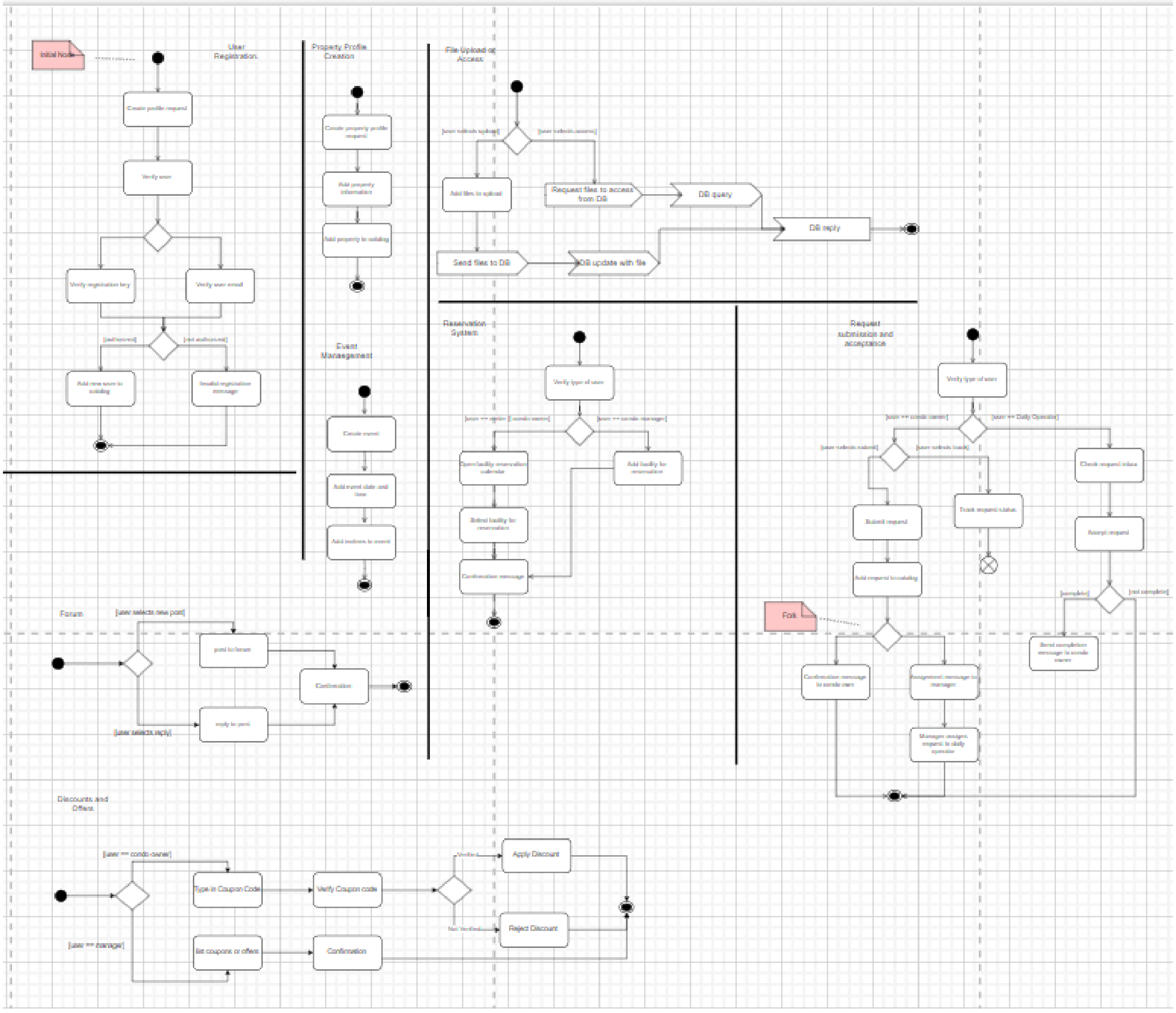
3.3.1 Concerns

Simplified financial system

3.3.2 Typical Stakeholders

Condo management companies

3.4 Model kinds



3.5 Condo Fees Calculation Model:

3.5.1 Condo Fees Calculation Model conventions:

- The Condo Fees Calculation Model utilizes UML class diagrams to depict the structure of calculations related to condo fees.
- Conventions involve defining classes for fee components, associations for related properties, and attributes for calculation details.

3.5.2 Condo Fees Calculation Model correspondence rules:

- Correspondence rules ensure accurate links between fee components and their representation in the model.
- Changes in fee calculations or related property details should be reflected in the Condo Fees Calculation Model.

3.5 Operational Budget Model:

3.5.1 Operational Budget Model conventions:

- The Operational Budget Model employs UML class diagrams to represent the structure of operational budget components.
- Conventions include defining classes for budget categories, associations for related properties, and attributes for budget details.

3.5.2 Operational Budget Model correspondence rules:

- Correspondence rules ensure accurate associations between budget components and related properties.
- Changes in budget categories or property associations should be reflected in the Operational Budget Model.

3.5 Costs Recording Model:

3.5.1 Costs Recording Model conventions:

- The Costs Recording Model uses UML class diagrams to represent the structure of recording costs associated with condo management.
- Conventions involve defining classes for cost entries, associations for related properties, and attributes for cost details.

3.5.2 Costs Recording Model correspondence rules:

- Correspondence rules ensure accurate links between cost entries and related properties.
- Changes in cost details or property associations should be reflected in the Costs Recording Model.

3.5 Annual Financial Reports Model:

3.5.1 Annual Financial Reports Model conventions:

- The Annual Financial Reports Model utilizes UML class diagrams to represent the structure of generating annual financial reports.
- Conventions include defining classes for report components, associations for related properties, and attributes for report details.

3.5.2 Annual Financial Reports Model correspondence rules:

- Correspondence rules ensure accurate associations between report components and related properties.
- Changes in report details or property associations should be reflected in the Annual Financial Reports Model.

3.6 Operations on views

Construction Methods:

- Construction methods involve process guidance for creating a simplified financial system, including the calculation and presentation of condo fees, recording of operational budget and costs, and generation of annual financial reports.
- Templates for representing condo fees, operational budget, costs, and financial reports.

Interpretation Methods:

- Interpretation methods guide readers in understanding the relationships between financial components and the overall financial system.
- Heuristic guidance for interpreting patterns or styles used in synthesizing financial information.

Analysis Methods:

- Analysis methods are applied to check the accuracy and simplicity of financial calculations and reporting.
- Model correspondence rules are checked to ensure accurate associations between financial components.

Implementation Methods:

- Implementation methods focus on designing and implementing the financial system model into the broader system.
- Guidance on incorporating financial information into the overall system design and functionality.

Reservation System Viewpoint:

3.2 Overview

This viewpoint incorporates a reservation system for common facilities, allowing users to book facilities with ease.

3.3 Concerns and Stakeholders

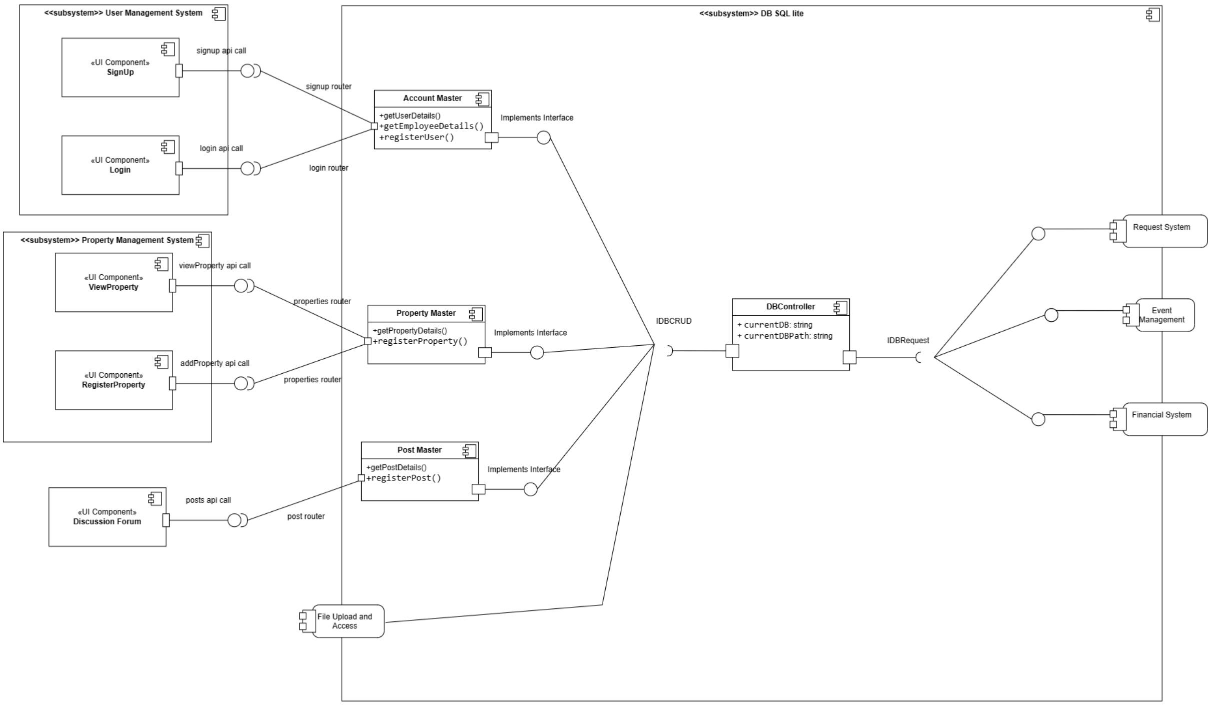
3.3.1 Concerns

Efficient reservation system

3.3.2 Typical Stakeholders

Condo owners
Rental users

3.4 Model kinds



3.5 Reservation System Model:

3.5.1 Reservation System Model conventions:

- The Reservation System Model employs UML class diagrams to represent the structure of reservations for common facilities.
- Conventions include defining classes for reservation details, associations for related properties, and attributes for reservation information.

3.5.2 Reservation System Model correspondence rules:

- Correspondence rules ensure accurate links between reservations and related properties.
- Changes in reservation details or property associations should be reflected in the Reservation System Model.

3.6 Operations on views

Construction Methods:

- Construction methods involve process guidance for incorporating a reservation system for common facilities, allowing users to book facilities with ease.
- Templates for representing reservation details and associated properties.

Interpretation Methods:

- Interpretation methods guide readers in understanding the relationships between reservations and properties.
- Heuristic guidance for interpreting patterns or styles used in synthesizing reservation information.

Analysis Methods:

- Analysis methods are applied to check the efficiency of the reservation system and its integration with the overall system.
- Model correspondence rules are checked to ensure accurate associations between reservations and properties.

Implementation Methods:

- Implementation methods focus on designing and implementing the reservation system model into the broader system.
- Guidance on incorporating reservation information into the overall system design and functionality.

Employee Roles and Permissions Viewpoint:

3.2 Overview

This viewpoint supports different roles for employees within condo management companies, such as managers responsible for daily operations or finance.

3.3 Concerns and Stakeholders

3.3.1 Concerns

Effective employee roles and permissions.

3.3.2 Typical Stakeholders

Employees within condo management companies

3.4 Model kinds

3.5 Employee Roles Model:

3.5.1 Employee Roles Model conventions:

- The Employee Roles Model uses UML class diagrams to depict the structure of different roles for employees within condo management companies.
- Conventions involve defining classes for role specifications, associations for related employees, and attributes for role details.

3.5.2 Employee Roles Model correspondence rules:

- Correspondence rules ensure accurate associations between roles and related employees.

- Changes in role specifications or employee associations should be reflected in the Employee Roles Model.

3.5 Permissions Model:

3.5.1 Permissions Model conventions:

- The Permissions Model employs UML class diagrams to represent the structure of permissions associated with different roles.
- Conventions include defining classes for permission details, associations for related roles, and attributes for permission specifications.

3.5.2 Permissions Model correspondence rules:

- Correspondence rules ensure accurate links between permissions and related roles.
- Changes in permission specifications or role associations should be reflected in the Permissions Model.

3.6 Operations on views

Construction Methods:

- Construction methods involve process guidance for supporting different roles for employees within condo management companies.
- Templates for representing employee roles and permissions.

Interpretation Methods:

- Interpretation methods guide readers in understanding the relationships between employee roles, permissions, and related entities.
- Heuristic guidance for interpreting patterns or styles used in synthesizing employee-related information.

Analysis Methods:

- Analysis methods are applied to check the effectiveness of employee roles and permissions.
- Model correspondence rules are checked to ensure accurate associations between roles, permissions, and related entities.

Implementation Methods:

- Implementation methods focus on designing and implementing the employee roles and permissions model into the broader system.
- Guidance on incorporating employee-related information into the overall system design and functionality.

Scalability and Accessibility Viewpoint:

3.2 Overview

This viewpoint ensures scalability to handle a growing number of users, properties, and associated data. It also focuses on providing a user-friendly interface for various stakeholders.

3.3 Concerns and Stakeholders

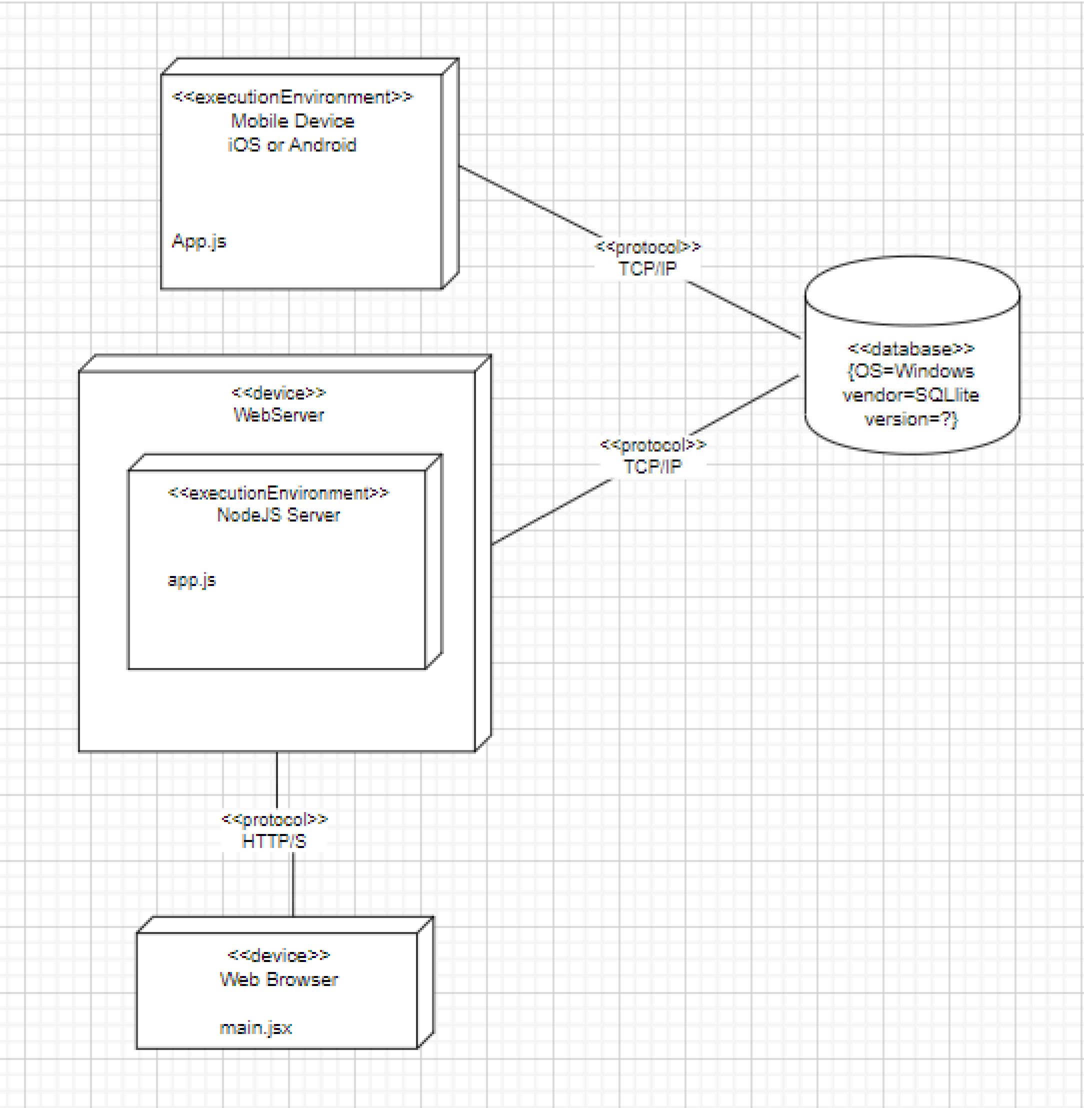
3.3.1 Concerns

System scalability and user-friendly interface.

3.3.2 Typical Stakeholders

All users and stakeholders

3.4 Model kinds



3.5 Scalability Assessment Model:

3.5.1 Scalability Assessment Model conventions:

- The Scalability Assessment Model uses UML class diagrams to represent the scalability considerations for handling a growing number of users and properties.
- Conventions involve defining classes for scalability components, associations for related systems, and attributes for scalability details.

3.5.2 Scalability Assessment Model correspondence rules:

- Correspondence rules ensure accurate links between scalability components and related systems.
- Changes in scalability considerations or system associations should be reflected in the Scalability Assessment Model.

3.5 User Interface Model:

3.5.1 User Interface Model conventions:

- The User Interface Model employs UML class diagrams or wireframes to represent the design and structure of a user-friendly interface.
- Conventions include defining classes for interface elements, associations for user interactions, and attributes for interface specifications.

3.5.2 User Interface Model correspondence rules:

- Correspondence rules ensure accurate links between interface elements and user interactions.
- Changes in interface specifications or user interactions should be reflected in the User Interface Model.

3.6 Operations on views

Construction Methods:

- Construction methods involve process guidance for ensuring scalability to handle a growing number of users, properties, and associated data.
- Templates for representing scalability components and user-friendly interfaces.

Interpretation Methods:

- Interpretation methods guide readers in understanding the relationships between scalability components, user interfaces, and overall system accessibility.
- Heuristic guidance for interpreting patterns or styles used in synthesizing scalability and accessibility information.

Analysis Methods:

- Analysis methods are applied to check the scalability and user-friendliness of the system.
- Model correspondence rules are checked to ensure accurate associations between scalability components and user interfaces.

Implementation Methods:

- Implementation methods focus on designing and implementing the scalability and accessibility model into the broader system.
- Guidance on incorporating scalability and accessibility information into the overall system design and functionality.

Feasibility Assessment Viewpoint:

3.2 Overview

This viewpoint assesses the feasibility of constructing and deploying the system within defined project constraints.

3.3 Concerns and Stakeholders

3.3.1 Concerns

Technical and operational feasibility.

3.3.2 Typical Stakeholders

Project stakeholders

3.4 Model kinds

3.5 Technical Feasibility Model:

3.5.1 Technical Feasibility Model conventions:

- The Technical Feasibility Model uses UML diagrams to represent technical aspects, ensuring that chosen technologies can support required functionalities.
- Conventions involve defining classes for technical components, associations for relevant technologies, and attributes for technical details.

3.5.2 Technical Feasibility Model correspondence rules:

- Correspondence rules ensure accurate links between technical components and relevant technologies.
- Changes in technical details or technology associations should be reflected in the Technical Feasibility Model.

3.5 Operational Feasibility Model:

3.5.1 Operational Feasibility Model conventions:

- The Operational Feasibility Model uses UML diagrams to assess the alignment of the system with organizational processes.
- Conventions involve defining classes for operational components, associations for organizational processes, and attributes for operational details.

3.5.2 Operational Feasibility Model correspondence rules:

- Correspondence rules ensure accurate links between operational components and organizational processes.
- Changes in operational details or process associations should be reflected in the Operational Feasibility Model.

3.6 Operations on views

Construction Methods:

- Construction methods involve process guidance for assessing the feasibility of constructing and deploying the system within defined project constraints.
- Templates for representing technical and operational feasibility components.

Interpretation Methods:

- Interpretation methods guide readers in understanding the relationships between feasibility components and project constraints.
- Heuristic guidance for interpreting patterns or styles used in synthesizing feasibility information.

Analysis Methods:

- Analysis methods are applied to check the technical and operational feasibility of the system.
- Model correspondence rules are checked to ensure accurate associations between feasibility components and project constraints.

Implementation Methods:

- Implementation methods focus on designing and implementing the feasibility assessment model into the broader system.
- Guidance on incorporating feasibility information into the overall system design and functionality.

Risk Management Viewpoint:

3.2 Overview

This viewpoint identifies and addresses potential risks and impacts related to data security, user adoption, technical challenges, legal and compliance issues.

3.3 Concerns and Stakeholders

3.3.1 Concerns

Risk identification and mitigation.

3.3.2 Typical Stakeholders

All stakeholders

3.4 Model kinds

3.5 Risk Identification Model:

3.5.1 Risk Identification Model conventions:

- The Risk Identification Model uses UML diagrams to identify potential risks related to data security, user adoption, technical challenges, and legal issues.
- Conventions involve defining classes for risk components, associations for risk factors, and attributes for risk details.

3.5.2 Risk Identification Model correspondence rules:

- Correspondence rules ensure accurate links between risk components and relevant risk factors.
- Changes in risk details or factors should be reflected in the Risk Identification Model.

3.5 Risk Mitigation Model:

3.5.1 Risk Mitigation Model conventions:

- The Risk Mitigation Model uses UML diagrams to outline strategies and actions for mitigating identified risks.
- Conventions involve defining classes for mitigation components, associations for risk factors, and attributes for mitigation details.

3.5.2 Risk Mitigation Model correspondence rules:

- Correspondence rules ensure accurate links between mitigation components and relevant risk factors.
- Changes in mitigation details or risk factors should be reflected in the Risk Mitigation Model.

3.6 Operations on views

Construction Methods:

- Construction methods involve process guidance for identifying and addressing potential risks and impacts related to data security, user adoption, technical challenges, legal, and compliance issues.
- Templates for representing risk identification and mitigation components.

Interpretation Methods:

- Interpretation methods guide readers in understanding the relationships between risk components and potential impacts.
- Heuristic guidance for interpreting patterns or styles used in synthesizing risk-related information.

Analysis Methods:

- Analysis methods are applied to check the effectiveness of risk identification and mitigation strategies.
- Model correspondence rules are checked to ensure accurate associations between risk components and potential impacts.

Implementation Methods:

- Implementation methods focus on designing and implementing the risk management model into the broader system.
- Guidance on incorporating risk-related information into the overall system design and functionality.

Maintenance and Evolution Viewpoint:

3.2 Overview

This viewpoint establishes a plan for the ongoing maintenance and evolution of the system, including regular updates, monitoring, adaptation to technological advancements, and feature evolution.

3.3 Concerns and Stakeholders

3.3.1 Concerns

Ongoing maintenance and system evolution.

3.3.2 Typical Stakeholders

System administrators, developers, and users

3.4 Model kinds

3.5 Maintenance Plan Model:

3.5.1 Maintenance Plan Model conventions:

- The Maintenance Plan Model uses UML diagrams to outline plans for regular updates, patches, and system improvements.
- Conventions involve defining classes for maintenance components, associations for update plans, and attributes for maintenance details.

3.5.2 Maintenance Plan Model correspondence rules:

- Correspondence rules ensure accurate links between maintenance components and update plans.
- Changes in maintenance details or update plans should be reflected in the Maintenance Plan Model.

3.5 Evolution Plan Model:

3.5.1 Evolution Plan Model conventions:

- The Evolution Plan Model uses UML diagrams to outline plans for adapting to technological advancements and ensuring long-term sustainability.
- Conventions involve defining classes for evolution components, associations for adaptation plans, and attributes for evolution details.

3.5.2 Evolution Plan Model correspondence rules:

- Correspondence rules ensure accurate links between evolution components and adaptation plans.
- Changes in evolution details or adaptation plans should be reflected in the Evolution Plan Model.

3.5 User Feedback Model:

3.5.1 User Feedback Model conventions:

- The User Feedback Model uses UML diagrams to capture feedback from users for continuous improvement.
- Conventions involve defining classes for feedback components, associations for user interactions, and attributes for feedback details.

3.5.2 User Feedback Model correspondence rules:

- Correspondence rules ensure accurate links between feedback components and user interactions.
- Changes in feedback details or user interactions should be reflected in the User Feedback Model.

3.5 Technological Advancements Model:

3.5.1 Technological Advancements Model conventions:

- The Technological Advancements Model uses UML diagrams to assess and incorporate advancements in technology.
- Conventions involve defining classes for technological components, associations for relevant advancements, and attributes for technological details.

3.5.2 Technological Advancements Model correspondence rules:

- Correspondence rules ensure accurate links between technological components and relevant advancements.
- Changes in technological details or advancements should be reflected in the Technological Advancements Model.

3.5 Feature Evolution Model:

3.5.1 Feature Evolution Model conventions:

- The Feature Evolution Model uses UML diagrams to outline plans for evolving features to meet changing user needs and industry standards.
- Conventions involve defining classes for feature components, associations for evolving plans, and attributes for feature evolution details.

3.5.2 Feature Evolution Model correspondence rules:

- Correspondence rules ensure accurate links between feature components and evolving plans.
- Changes in feature details or evolving plans should be reflected in the Feature Evolution Model.

3.6 Operations on views

Construction Methods:

- Construction methods involve process guidance for establishing a plan for the ongoing maintenance and evolution of the system.
- Templates for representing maintenance and evolution plans, user feedback, technological advancements, and feature evolution.

Interpretation Methods:

- Interpretation methods guide readers in understanding the relationships between maintenance and evolution components, user feedback, technological advancements, and feature evolution.
- Heuristic guidance for interpreting patterns or styles used in synthesizing maintenance and evolution-related information.

Analysis Methods:

- Analysis methods are applied to check the effectiveness of maintenance and evolution plans, user feedback, and adaptation to technological advancements.
- Model correspondence rules are checked to ensure accurate associations between maintenance and evolution components and relevant entities.

Implementation Methods:

- Implementation methods focus on designing and implementing the maintenance and evolution model into the broader system.
- Guidance on incorporating maintenance and evolution-related information into the overall system design and functionality.

3.7 Correspondence rules

User Profile Model

Association with Roles and Permissions:

- Each user profile must be associated with the appropriate roles and permissions.
- Correspondence between user profiles and roles/permissions must be maintained for access control.

Linkage to Property and Financial Information:

- Condo owners' profiles should be linked to their respective property and financial information.
- Correspondence rules to ensure accurate representation of the relationship between user profiles and property/financial data.

Property Model

Connection with Property Profiles:

- Each property entity within the property model should correspond to a specific property profile.
- Ensuring that the information in the property model aligns with the details in the property profiles.

Association with Financial Overview:

- Correspondence between properties and their financial overview, ensuring consistency in financial data representation.

Property Profile Model

Linkage with Unit Information:

- Property profiles should accurately link to unit information, parking spots, and locker details.
- Ensuring that the property profile model correctly represents the property's attributes.

Connection with Condo Files:

- Correspondence rules to ensure that condo files are appropriately associated with the relevant property profiles.

Condo Fees Calculation Model

Association with Property Profiles:

- Correspondence between condo fees calculation and property profiles to determine the fees for each property accurately.

Linkage with Operational Budget:

- Ensuring that condo fees calculation corresponds appropriately with the operational budget model.

Reservation System Model

Connection with Property Details:

- Each reservation should correspond to specific property details.
- Ensuring accurate representation of reservations in conjunction with the associated properties.

Employee Roles Model

Association with Permissions:

- Each employee role should correspond to the appropriate permissions.
- Ensuring that employees with specific roles have the corresponding permissions.

Linkage with Operational Processes:

- Correspondence rules between employee roles and the operational processes they are responsible for.

Scalability Assessment Model

Connection with User Interface:

- Correspondence rules between the scalability assessment model and the user interface model.
- Ensuring that the system's scalability aligns with the user-friendly interface.

Technical Feasibility Model

Correspondence with Chosen Technologies:

- Ensuring that the technical feasibility model corresponds accurately with the chosen technologies for system construction.

Risk Identification Model

Association with Mitigation Strategies:

- Correspondence between identified risks and the mitigation strategies outlined in the risk mitigation model.

Maintenance Plan Model

Linkage with User Feedback:

- Correspondence rules between the maintenance plan model and user feedback model.
- Incorporating user feedback into the maintenance plan for continuous improvement.