| **SOEN 390:**  **SOFTWARE ENGINEERING TEAM DESIGN PROJECT** |
| --- |
|  |
|  |
|  |
| **Condo Management System**  **Updated Software Architecture Document** |
| **(Sprint 4)** |
|  |
|  |
| **Team 14** |
| **Instructor: Dr. Jinqiu Yang  Date: April 14th, 2024** |
| **Winter 2024** |

# Contents

[**Contents 2**](#)

[**1 Introduction 4**](#)

[**1.1 Identifying information 4**](#)

[**1.2 Supplementary information 4**](#)

[**1.3 Other information 4**](#)

[**1.3.1 Architecture evaluations 4**](#)

[**1.3.2 Rationale for key decisions 4**](#)

[**2 Stakeholders and concerns 5**](#)

[**2.1 Stakeholders 5**](#)

[**2.2 Concerns 6**](#)

[2.3 Concern–Stakeholder Traceability 7](#_44sinio)

[**Table 3: Concern and Stakeholder relations 8**](#_bb8rr7g8tqw0)

[**3 Viewpoints 8**](#_2jxsxqh)

[3.A. Logical Viewpoint 8](#_2c2ol1f6zol2)

[1. Overview 8](#_16m73ddjrbt7)

[2. Concerns and stakeholders 8](#_4n971kf2r1be)

[Concern 1: Will the requirements change? 8](#_ths1i5ei0qd8)

[Concern 2: Is the system adaptable to a new environment? 9](#_7ukrp8cczzu9)

[Description: This viewpoint examines the system's compatibility and readiness for deployment in different environments, such as cloud, on-premises, or hybrid setups. It also evaluates the adaptability of system components and configurations to transition and operate in diverse environments while maintaining performance, reliability, and security. 9](#_suyzp7vng4if)

[- Project Manager: Responsible for Identifying target deployment environments early in the project lifecycle, and collaborating with architecture and development teams to ensure compatibility with target environments 9](#_mq9ktylsq7jg)

[- System Administrators: Responsible for providing input on target deployment environments and configuration requirements, assisting with environment setup and testing, and collaborating with development teams to resolve environment-specific issues. 9](#_x5oupitz02hi)

[Concern 3: Am I able to edit information about a property? 9](#_o82sjfva3t22)

[**3. Model kinds 10**](#)

[Figure 1: Domain Model for AnaCondo Management System 11](https://docs.google.com/document/d/1M57X8FDaL8sf92zBhMGARQByTzr7ULM1/edit#heading=h.8otfm0l94mhr)

[3.B. Development Viewpoint 11](https://docs.google.com/document/d/1M57X8FDaL8sf92zBhMGARQByTzr7ULM1/edit#heading=h.q3p1on8q9wj2)

[1. Overview 11](#_yxt2hbj5xu4c)

[2. Concerns and stakeholders 11](#_1gq3n1babz55)

[Concern 3: Is the system reusable and maintainable? 12](#_e901imrhk8yr)

[3. Model kinds 13](#_g8liqrvebkqk)

[3.C. Process Viewpoint 14](#_sj53d5xtrthm)

[1. Overview 14](#_adepysqix73m)

[3. Model kinds 15](#_31hxgemhkmzl)

[3.D. Physical Viewpoint 17](#_io4261t583rv)

[1. Overview 17](#_q4ymr8pedx4q)

[3.E. Scenarios Viewpoint 19](#_ydm5p6xdfrj2)

[1. Overview 19](#_ebgvxtpc1p1g)

[**5 Consistency and correspondences 25**](#)

[**5.1 Known inconsistencies 25**](#)

[**5.2 Correspondences in the AD 26**](#)

# 1 Introduction

## 1.1 Identifying information

The AnaCondo platform is a comprehensive condominium management system tailored for both web and mobile usage. AnaCondo offers accessibility across various web services on different operating systems like Android, IOS, Linux, MacOS & Windows. This versatile platform allows users to seamlessly manage condominium-related tasks and interactions, providing a unified experience across multiple devices and operating systems.

## 1.2 Supplementary information

Our platform deviates from conventional norms by eschewing the customary categorization of individuals seeking condominiums based on ownership preferences. Rather than segregating those looking to rent or own, our approach unifies these groups, fostering a symbiotic environment. This convergence facilitates seamless collaboration between owners and renters, particularly in financial matters and the coordination of various responsibilities such as cleaning requests. Departing from the standard paradigm, our platform endeavors to redefine the condominium living experience, emphasizing a more integrated and cooperative community dynamic for enhanced residential engagement.

## 1.3 Other information

### 1.3.1 Architecture evaluations

No evaluations were done so far in the sprint.

### 1.3.2 Rationale for key decisions

AnaCondo pursues a bifurcated approach in its front-end development, employing separate teams dedicated to ReactJS for web application design and React Native for mobile application development. The rationale behind this choice lies in React's comprehensive framework, offering accessibility and robust development capacities across both web and mobile platforms. The strategic use of React aligns with AnaCondo's commitment to delivering a coherent user experience irrespective of the chosen device. Supporting the front-end, Firebase emerges as the selected backend database, chosen for its simplicity and efficiency in facilitating a seamless amalgamation of design and functionality. This architectural decision underscores AnaCondo's dedication to optimal user experiences. The goal is to establish AnaCondo as a standard-bearer for user-friendly interfaces in both web and mobile environments.

# 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. See also: ISO/IEC/IEEE 42010, 5.3

## 2.1 Stakeholders

Anaconda’s stakeholders have varying profiles and may be grouped into subgroups with concording concerns, goals and interests. These stakeholders may be identified by consulting the following list.

| **Stakeholder** | **Description** | **Role** |
| --- | --- | --- |
| 1. Project manager | Manages the project | Responsible for the planning and execution of the  project. Must ensure the project is organized, delivered on time and stays on budget. |
| 1. Rental Users | Public users who rent properties | Temporary occupants who use the platform for rental user features. Utilize the platform to manage their profile, submit applications, and communicate with management staff. |
| 1. Condo Owners | Public users who own properties. | Using condo owner features, managing requests for renters such as access and overseeing their condo units, manage property-related documents. |
| 1. Condo Management Companies | Users who manage condos | Managing their employees, creating employee profiles, managing properties. |
| 1. Condo Management Employees | Employees of Condo Management Companies | Managing their tasks, marking tasks status, communicating with customers. |
| 1. Condo Management Company Employee\_Property Manager | Property Managers of Condo Management Companies | Managing property, managing requests. |
| 1. Condo Management Company Employee\_Property Financial Personnel | Financial Personnel of Condo Management Companies | Managing the financial aspect of the property. |
| 1. System Administrator | User who manages the system | Administration of the system. |
| 1. Architect | Software Architecture Designer | Designing and enforcing software architecture |
| 1. Developer | User who develops the system | Ensure Requirements are met by implementing the required features. |

***Table 1:*** *AnaCondo Management System Stakeholder List*

## 2.2 Concerns

| Concern ID | Description |
| --- | --- |
| 1 | Will the budget suffice? |
| 2 | Are the estimated completion time and task timelines reasonable? |
| 3 | Will the system be delivered on time? |
| 4 | Are there enough resources available to complete the project? |
| 5 | Is the system reusable and maintainable? |
| 6 | Am I able to create a profile? |
| 7 | Is the system easy to navigate through and understand? |
| 8 | Is the system compatible with my preferred platform? |
| 9 | Am I able to use all my profile features? |
| 10 | Am I able to submit a request and trust it will be assigned to the correct personnel? |
| 11 | Am I able to generate the appropriate keys for the corresponding profile types? |
| 12 | Am I able to edit information about a property? |
| 13 | Are the financial calculations accurate? |
| 14 | Am I able to see my assigned tasks? |
| 15 | Will there be unplanned Application Downtime? |
| 16 | Is the system safe to use and will any confidential information be compromised? |
| 17 | Is the system performance adequate? |
| 18 | Is the data recoverable if lost? |
| 19 | Is the architecture extendable? |
| 20 | Is the system adaptable to a new environment? |
| 21 | Are the requirements understandable? |
| 22 | Will the requirements change? |

***Table 2:*** *AnaCondo Management System Concerns and Description List*

## 2.3 Concern–Stakeholder Traceability



| **StakeHolder**  **Concern** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | X |  |  |  |  |  |  |  |  |  |
| **2** | X |  |  |  |  |  |  | X | X | X |
| **3** | X | X | X | X | X | X | X |  |  |  |
| **4** | X |  |  |  |  |  |  |  |  |  |
| **5** | X |  |  |  |  |  |  | X | X | X |
| **6** |  | X | X | X |  |  |  |  |  |  |
| **7** |  | X | X | X | X | X | X |  |  |  |
| **8** |  | X | X | X | X | X | X |  |  |  |
| **9** |  | X | X | X | X | X | X |  |  |  |
| **10** |  |  | X |  |  |  |  |  |  |  |
| **11** |  |  |  | X |  |  |  |  |  |  |
| **12** |  |  | X | X |  | X |  | X |  |  |
| **13** |  |  |  | X |  |  | X |  |  |  |
| **14** |  |  |  |  | X |  |  |  |  |  |
| **15** | X |  |  |  |  |  |  | X |  | X |
| **16** | X | X | X | X |  | X | X | X |  |  |
| **17** | X |  |  |  |  |  |  | X |  |  |
| **18** | X |  |  |  |  |  |  | X |  |  |
| **19** | X |  |  |  |  |  |  |  | X | X |
| **20** | X |  |  |  |  |  |  |  | X | X |
| **21** | X |  |  |  |  |  |  |  |  | X |
| **22** | X |  |  |  |  |  |  |  | X | X |

# ***Table 3:*** *Concern and Stakeholder relations*

# 3 Viewpoints

## **3.A. Logical Viewpoint**

Synonym: Functional view/Functional perspective

## Overview

The Functional Viewpoint provides a detailed perspective on the system's functionalities and how they are organized to meet user needs and business requirements. It focuses on describing the system's features, capabilities, and interactions from a functional perspective.

## Concerns and stakeholders

### **Concern 1:** Will the requirements change?

**Description:** This viewpoint assesses the likelihood and impact of potential changes to system requirements throughout the development lifecycle. In addition, it evaluates the system's flexibility to accommodate evolving business needs and stakeholder preferences.

**Stakeholders and Approach:**

* **Architect:** Responsible for designing the system with flexibility and adaptability in mind, adopting agile development practices to facilitate iterative requirement refinement and change management, and implementing modular architectures that can accommodate changing requirements.
* **Project Manager:** Responsible for anticipating potential changes to requirements based on evolving stakeholder needs and market conditions, implementing change control processes to assess and manage requirement changes, and communicating changes effectively to the development team.
* **Developer:** Responsible for designing and implementing code with flexibility and modularity by following coding best practices.

### **Concern 2:** Is the system adaptable to a new environment?

### **Description:** This viewpoint examines the system's compatibility and readiness for deployment in different environments, such as cloud, on-premises, or hybrid setups. It also evaluates the adaptability of system components and configurations to transition and operate in diverse environments while maintaining performance, reliability, and security.

**Stakeholders and Approach:**

### **Project Manager:** Responsible for Identifying target deployment environments early in the project lifecycle, and collaborating with architecture and development teams to ensure compatibility with target environments

### **System Administrators:** Responsible for providing input on target deployment environments and configuration requirements, assisting with environment setup and testing, and collaborating with development teams to resolve environment-specific issues.

* **Architect:** Responsible for guiding the evaluation of system adaptability to new environments.

### **Concern 3:** Am I able to edit information about a property?

**Description:** This viewpoint reviews the system's functionality and user interfaces for managing and updating property information. It also ensures that users can easily access, modify, and maintain property details, such as unit specifications, amenities, and occupancy status.

**Stakeholders and Approach:**

* **Condo Management Companies:** Responsible for defining user permissions and access levels for property editing functions, providing training and documentation on property management features, and implementing auditing mechanisms to track changes.
* **Condo Management Company Employee\_Property Manager:** Responsible for utilizing property management tools to update property information, verify accuracy of data changes, and communicate property updates to relevant stakeholders.
* **Administrator:** Responsible for overseeing property management processes, reviewing property information edits, and addressing any discrepancies or inconsistencies in property data reported by users.
* **Condo Owner:** Responsible for providing input on the user interface and functionality required to edit property information. Providing feedback on the usability and effectiveness ofthe editing features during user testing and validation phases.

**Concern 4:** Am I able to generate the appropriate keys for the corresponding profile types?

**Description:** This viewpoint examines the system's capabilities for generating and managing access keys or permissions associated with different user profiles. It enables administrators to assign and revoke keys accurately based on user roles and permissions, facilitating secure access control and data privacy while preventing unauthorized access to sensitive information.

**Stakeholders and Approach:**

* **Condo Management Companies:** Responsible for defining registration key generation processes and rules within the system, specifying key generation criteria based on user roles and permissions, and ensuring that generated keys are securely transmitted to users.

## Model kinds

A domain model diagram was used to demonstrate the abstract classes. It provides a conceptual framework that captures the key entities, their attributes, relationships, and the fundamental operations that can be performed on them within a specific problem domain. It serves as a blueprint for understanding the system's structure and behavior from a high-level perspective, making it easier for stakeholders to identify requirements, constraints, and opportunities for abstraction

**Conventions and Correspondence Rules**

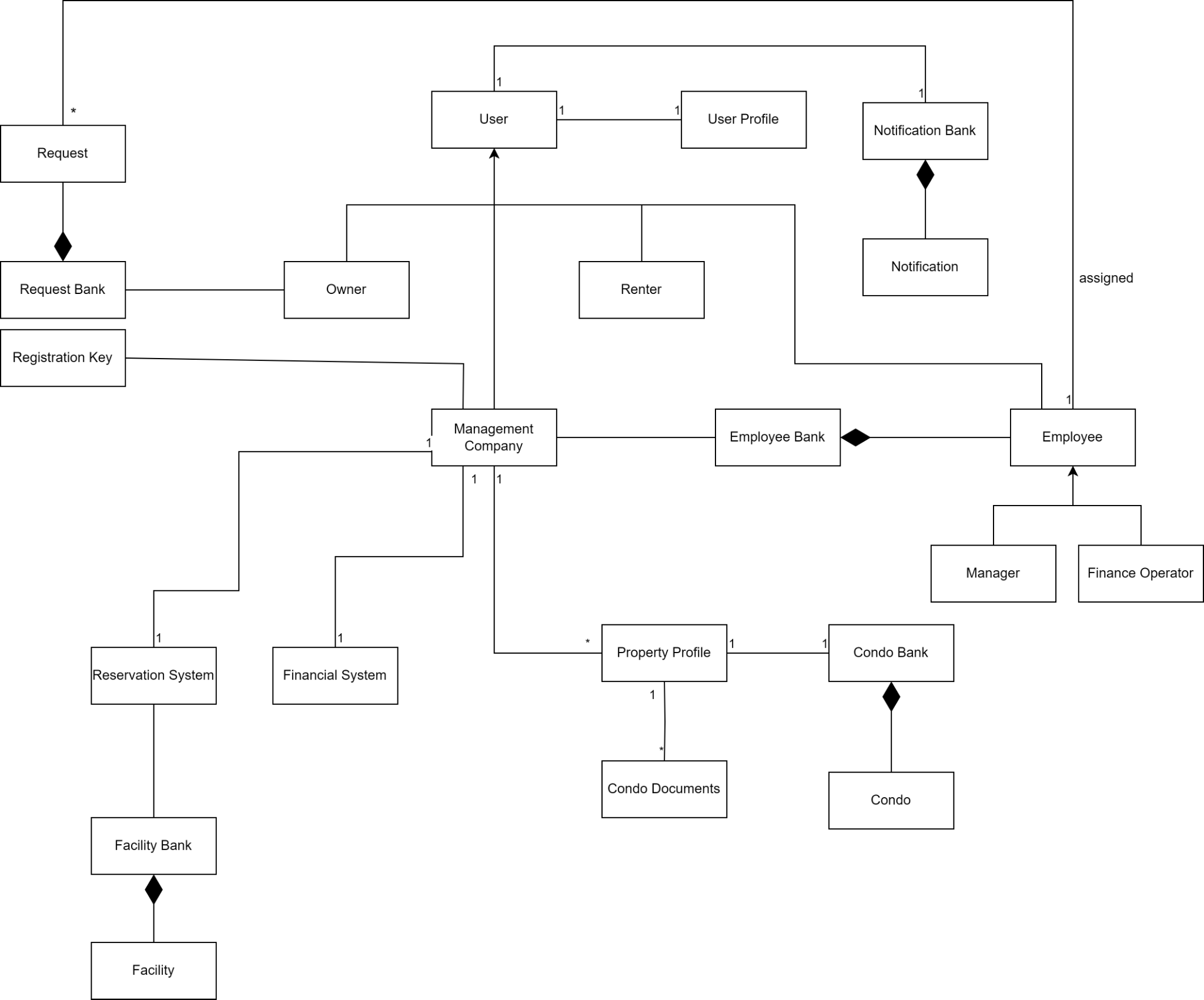
* Entities are represented as nouns within circles or rectangles.
* Attributes of entities are depicted as text within or adjacent to the entity representation.
* Relationships between entities are depicted as lines interconnecting with one another.
* Entities may have associated attributes that describe their properties or characteristics.
* Entities may be related to each other through various types of relationships such as association, aggregation, or composition.

Unified Modeling Language (UML) offers a detailed representation of the classes within a system, their attributes, methods, and the relationships among them. They help ensure consistency and coherence in the implementation phase, and support the principles of object-oriented design by emphasizing modularity, encapsulation, and inheritance, if present.

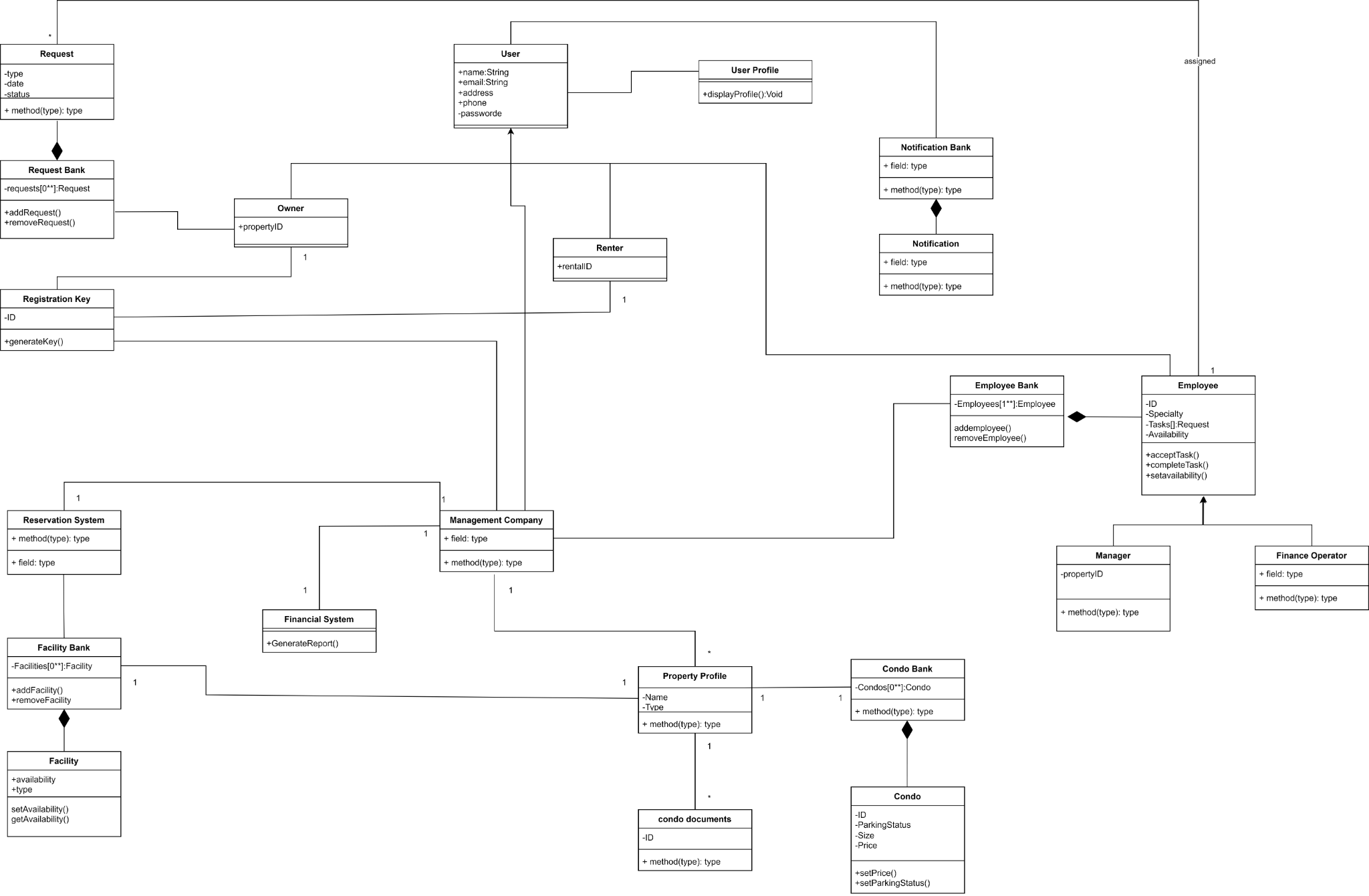
**Conventions and Correspondence Rules**

* Class names are typically singular nouns and are capitalized; represent the concept or entity in the system.
* Operations or methods describe the behavior or functionality of a class. They are also written with visibility, followed by the name, parameter list, and return type.
* Visibility of attributes and operations is indicated by symbols
* Relationships between classes are depicted using lines and include association, aggregation, composition, and inheritance.
* Relationships between classes are represented by cardinality correspondents. Can be expressed in terms of one to one, one to many, or many to many.

1. Views



***Figure 1****: Domain Model for AnaCondo Management System*



***Figure 11****: Class Diagram for AnaCondo Management System*

## **3.B. Development Viewpoint**

Synonym: Implementation Viewpoint

## Overview

The Development Viewpoint provides insights into the software development and implementation aspects of the system. It focuses on the tools, technologies, practices, and environments used in the development process.

## Concerns and stakeholders

**Concern 1:** Is the architecture extendable?

**Description:** This viewpoint examines the architectural design's adaptability to accommodate future business requirements, technological advancements, and scalability needs. Assessments include modularity, flexibility, and architectural patterns to ensure seamless integration of new features and functionalities.

**Stakeholders and Approach:**

* **Project Manager:** Responsible for allocating resources and time for architectural planning and design, and monitoring architectural evolution over time.
* **Developer:** Responsible for implementing code following design principles such as loose coupling and high cohesion, and designing interfaces and APIs that support extension points and customization.
* **Architect:** Responsible for assessing the architectural design to support future development efforts, and collaborating with development teams to implement modular, scalable, and flexible architectural solutions that allows the integration of new features.

**Concern 2:** Are the requirements understandable?

**Description:** This viewpoint reviews the clarity, completeness, and consistency of documented system requirements. It also ensures that the stakeholder’s needs and expectations are effectively communicated to the development team. This facilitates accurate implementation and reduces misunderstandings.

**Stakeholders and Approach:**

* **Project Manager:** Responsible for facilitating communication between stakeholders and the architecture team, and ensuring that requirements are documented.
* **Developer:** Responsible forreviewing requirements documentation to ensure clarity and consistency, and participating in requirement validation activities.

### **Concern 3:** Is the system reusable and maintainable?

**Description:** This viewpoint evaluates the software's structural integrity, code organization, and documentation quality to promote ease of reuse and maintenance. It also focuses on modularity, encapsulation, and adherence to coding standards to enhance code comprehensibility and maintainability.

**Stakeholders and Approach:**

* **Architect:** Responsible for designing the system with modularity and reusability in mind, and documenting design decisions to facilitate future maintenance.
* **Developer:** Responsible for implementing clean, modular code that follows best practices for maintainability, adhering to code standards and guidelines, and participating in code reviews to ensure code quality.
* **System Administrators:** Responsible for maintaining system documentation, implementing version control and configuration management practices, and establishing regular maintenance routines to keep the system in optimal condition.

## Model kinds

A component diagram was used to represent development viewpoint

**Conventions and Correspondence Rules**

* Components are represented as rectangles with rounded corners, labeled with their name.
* Interfaces are either depicted using filled circles attached to the component rectangle (provided) or as empty circles attached to the component rectangle (required).
* Provided interfaces represent the services or functionality offered by a component, while required interfaces represent the dependencies of a component on external functionality.
* Dependencies between components are indicated by lines connecting them with arrowheads pointing from the dependent component to the component it depends on, usually.

1. Component Diagram

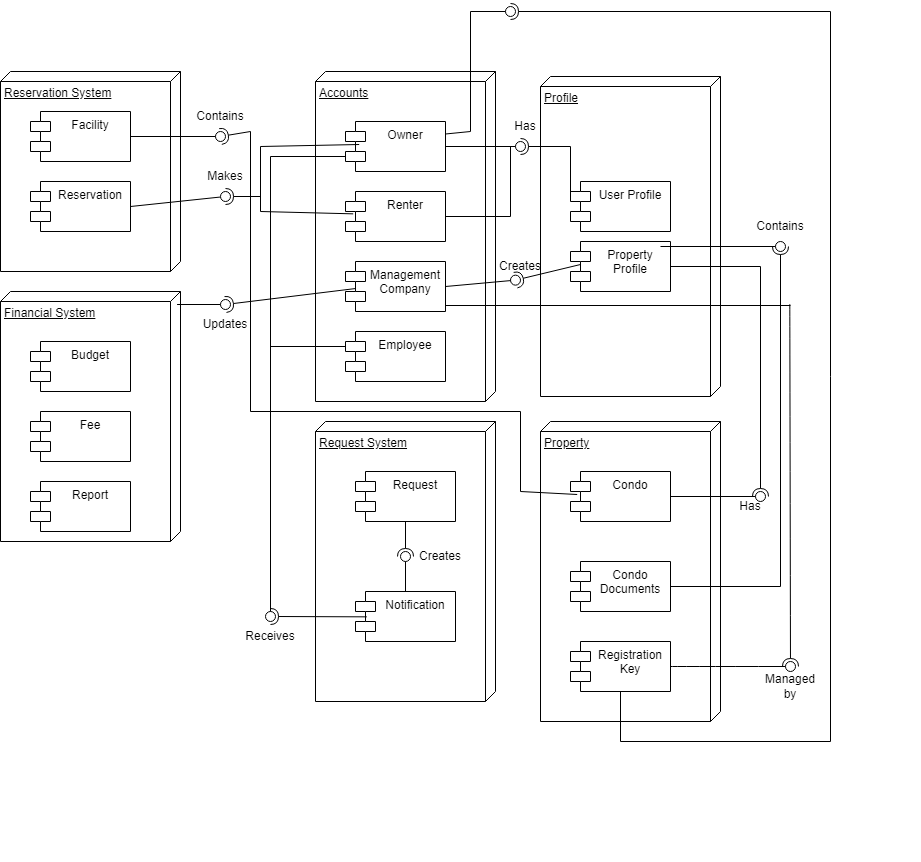


Figure 2: Component Diagram

## **3.C. Process Viewpoint**

## Overview

The Process Viewpoint provides insights into the system's runtime behavior and processes. It focuses on how the system operates, executes tasks, and interacts with its environment during runtime.

1. Concerns and Stakeholders

**Concern 1:** Will there be unplanned Application Downtime?

**Description:** The Process Viewpoint is utilized to analyze the operational processes and procedures related to system availability and reliability. It examines the strategies for preventing, detecting, and mitigating unplanned application downtime events. It allows stakeholders to assess the deployment processes, and monitoring systems to ensure continuous system availability and minimize disruptions to user activities.

**Stakeholders and Approach:**

* **System Administrators:** Responsible for implementing monitoring tools to detect potential downtime incidents, developing and maintaining disaster recovery plans to minimize downtime impact.
* **Project Manager:** Responsible for allocating resources and time for proactive maintenance and monitoring activities, and collaborating with system administrators to identify and address potential downtime risks, and communicating downtime impact to project stakeholders.
* **Developer:** Responsible for writing robust and resilient code that can handle unexpected errors and failures gracefully, and participating in load testing and stress testing to identify potential performance bottlenecks.

**Concern 2:** Are there enough resources available to complete the project?

**Description:** The Process Viewpoint is employed to evaluate the resource allocation and management processes essential for project completion. It encompasses aspects such as workforce planning, task allocation, and resource utilization.

**Stakeholders and Approach:**

* **Project Manager:** Responsible for conducting resource planning and allocation, and identifying staffing requirements and skill gaps.

## Model kinds

A sequence diagram was used to represent the process viewpoint

**Conventions and Correspondence Rules**

* Participants correspond to roles or entities in the system architecture.
* Each participant represents a distinct entity or component involved in the interaction, such as users, subsystems, or external systems.
* The flow of messages in the sequence diagram mirrors the sequence of actions or operations executed during the interaction.
* Messages exchanged between participants represent method calls, events, or communications that trigger specific behaviors or responses.

1. Sequence Diagram

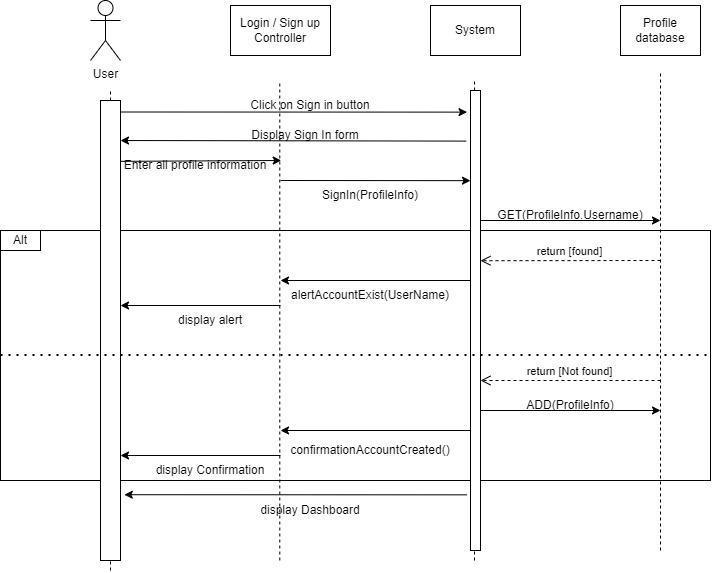


Figure 3: Sequence System Diagram for Registration of new Accounts

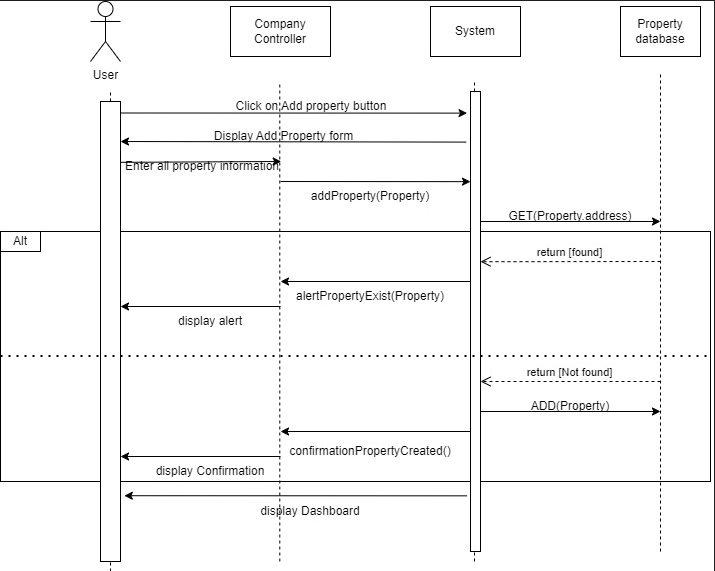


Figure 4: Sequence System Diagram for Adding New Property

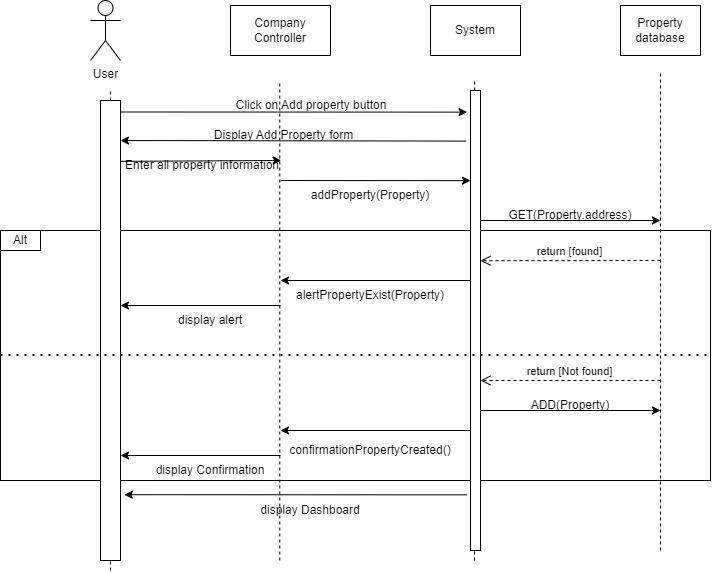


Figure 5: Chat Communication Sequence Diagram

## Activity Diagram

Here, we have an activity diagram representing the flow of Creating an Account.

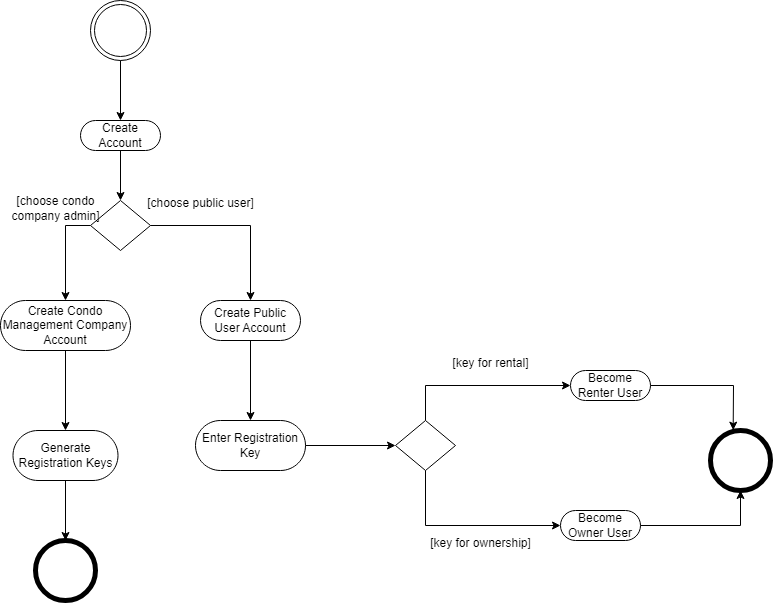


Figure 6: Create an Account Activity Diagram

## 

## **3.D. Physical Viewpoint**

## Overview

The Physical Viewpoint provides insights into the physical deployment of the system components onto hardware resources. It focuses on how the system is deployed, hosted, and managed in its operational environment.

1. Concerns and Stakeholders

**Concern 1:** Will the budget suffice?

**Description:** The Physical Viewpoint is utilized to assess the finances by using the system's physical infrastructure. It examines the costs associated with hardware procurement, installation, maintenance, and upgrades.

**Stakeholders and Approach:**

* **Project Manager:** Conducts budget planning and allocation for physical infrastructure requirements, identifies cost-saving opportunities, and monitors expenditure to ensure alignment with budget constraints.

**Concern 2:** Is the system safe to use and will any confidential information be compromised?

**Description:** The Physical Viewpoint is employed to evaluate the security measures implemented within the system's physical environment. It encompasses aspects such as data center security, access controls, disaster recovery plans, and compliance with regulatory standards.

**Stakeholders and Approach:**

* **Project Manager:** Responsible for allocating resources and time for security assessments and testing, prioritizing security requirements in project planning, and communicating security policies and procedures to project stakeholders.
* **Condo Owners and Users:** Responsible for providing input on security requirements and expectations regarding the safety of their personal information within the system.
* **Condo Management Companies:** Responsible for overseeing the security of the system and ensuring compliance with industry standards and regulations governing data protection and privacy. Collaborating with system architects to implement robust security controls and measures.
* **Condo Management Company Employee\_Property Manager:** Responsible for enforcing security policies and procedures related to property management activities. Ensuring that access controls and authentication mechanisms are implemented to prevent unauthorized access to sensitive property information.
* **Condo Management Company Employee\_Property Financial Personnel:** Responsible for handling financial data and sensitive information related to property finances. Collaborating with system administrators and security experts to implement means to protect financial data and prevent unauthorized disclosure or tampering.
* **System Administrators:** Responsible for implementing security measures such as firewalls, encryption. Conducts regular security audits and assessments.

**Concern 3:** Is the system performance adequate?

**Description:** The Physical Viewpoint allows to analyze the performance characteristics of the system's physical components. It evaluates factors such as processing speed, network bandwidth, storage capacity, and scalability.

**Stakeholders and Approach:**

* **System Administrators:** Responsible for monitoring system performance metrics such as CPU usage, memory utilization, and network traffic, identifying performance bottlenecks and optimizing system configurations.
* **Project Manager:** Responsible for collaborating with system architects, developers, and performance engineers to define performance requirements, establish performance metrics, and conduct performance testing. Monitoring performance metrics and analyzing test results to identify performance issues.

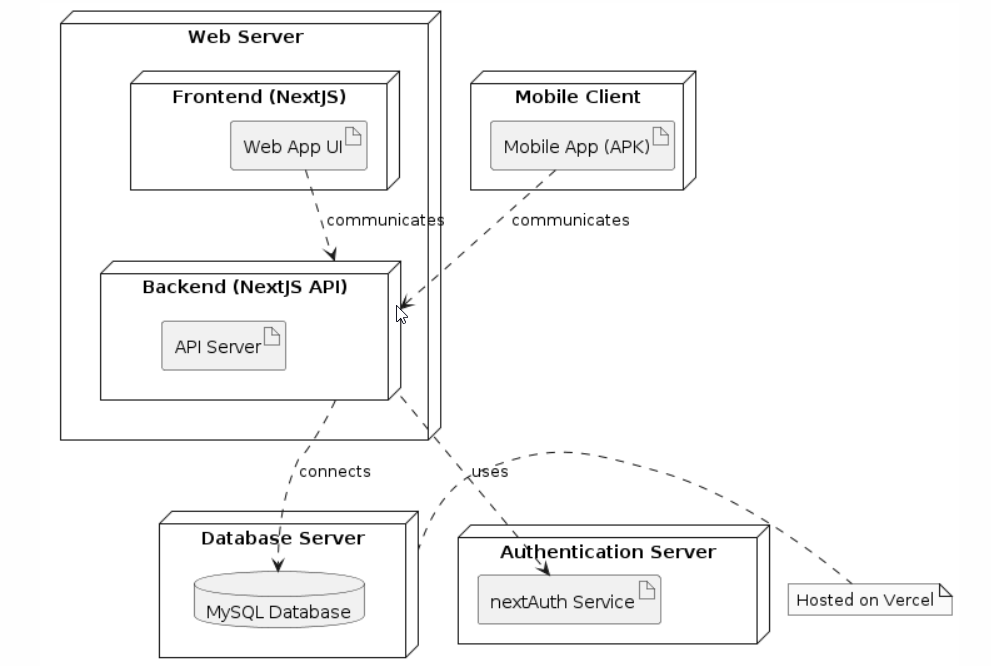
1. Model Kinds

A deployment diagram was chosen to represent the physical viewpoint

**Conventions and Rules**

* A deployment diagram contains nodes and artifacts
* Nodes , represented by boxes, showcase software or hardware components of the system architecture.
* Artifacts, represented by rectangles, depict software components deployed on nodes.
* Labels, represented by a paper-like rectangle, can be used to provide additional information on a node or artifact.
* Dotted lines with arrows represent relationships between components as well as their directionality.

1. Deployment Diagram



## Figure 3.1 : Deployment Diagram

## **3.E. Scenarios Viewpoint**

Synonym: Use Case Viewpoint

## Overview

The Scenarios Viewpoint focuses on describing the various interactions and behaviors of the system through concrete usage scenarios or use cases. This viewpoint provides a structured approach to capturing user requirements and system functionality from an end-user perspective.

1. Concerns and Stakeholders

**Concern 1:** Am I able to create a profile?

**Description:** This viewpoint evaluates the user registration and profile creation process within the system. Focuses on the ease of creating a new user profile, including data entry, validation, and user authentication mechanisms. It also verifies if the users can seamlessly register and set up their profiles to access system features and functionalities.

**Stakeholders and Approach:**

* **Rental Users and Condo Owners:** Responsible for providing input on user requirements and expectations for profile creation, ensuring that their needs are addressed in the use cases.
* **Condo Management Companies:** Responsible for creating profiles within the system to manage properties and interact with condo owners and tenants. Collaborating with system developers and designers to define user requirements and expectations for profile creation.

**Concern 2:** Is the system compatible with my preferred platform?

**Description:** This viewpoint examines the system's compatibility with various platforms, including web browsers, operating systems, and mobile devices. It also assesses the responsiveness, usability, and performance of the system across different platforms to ensure a consistent and optimal user experience.

**Stakeholders and Approach:**

* **Rental Users and Condo Owners:** Responsible for providing feedback on platform preferences and expectations for system compatibility, and ensuring that their needs are addressed in the use cases.
* **Condo Management Companies:** Responsible for ensuring that the system is compatible with the preferred platforms used by their employees for accessing and managing property-related information.
* **Condo Management Employees:** Responsible for providing input on preferred platforms and devices used for accessing the system. Assisting with testing and validation efforts to ensure that the system performs effectively on preferred platforms.
* **Condo Management Company Employee\_Property Manager:** Responsible for overseeing property management activities and utilizing the system to access relevant information and perform tasks, and specifying preferred platforms and devices for accessing the system based on needs and requirements.

**Concern 3:** Am I able to submit a request and trust it will be assigned to the correct personnel?

**Description:** This viewpoint reviews the process for submitting requests or inquiries within the system and the associated workflow for assignment to relevant personnel. It also evaluates the request submission interface, notification mechanisms, and assignment algorithms to ensure prompt and accurate routing of requests to the appropriate recipients.

**Stakeholders and Approach:**

* **Condo Owners:** Responsible for submitting requests within the system, providing feedback on the request submission process and monitoring the status of their requests.

1. Model Kinds

For the scenarios viewpoint, use case diagrams were used for AnaCondo

**Conventions and Rules**

* Use Cases represented as ovals or ellipse, where each use case is labeled with a descriptive name.
* Box or rectangle in the middle and Actors represented as stickmen placed on opposite ends.
* Name of the actor is written underneath the stick men figures.
  + “Extend” relationships depicted by a dashed line with an empty arrowhead.
  + An “include” relationship is depicted by a dashed arrow with an open arrowhead.

1. Use Case Diagrams

**AnaCondo System Use Case Diagram**

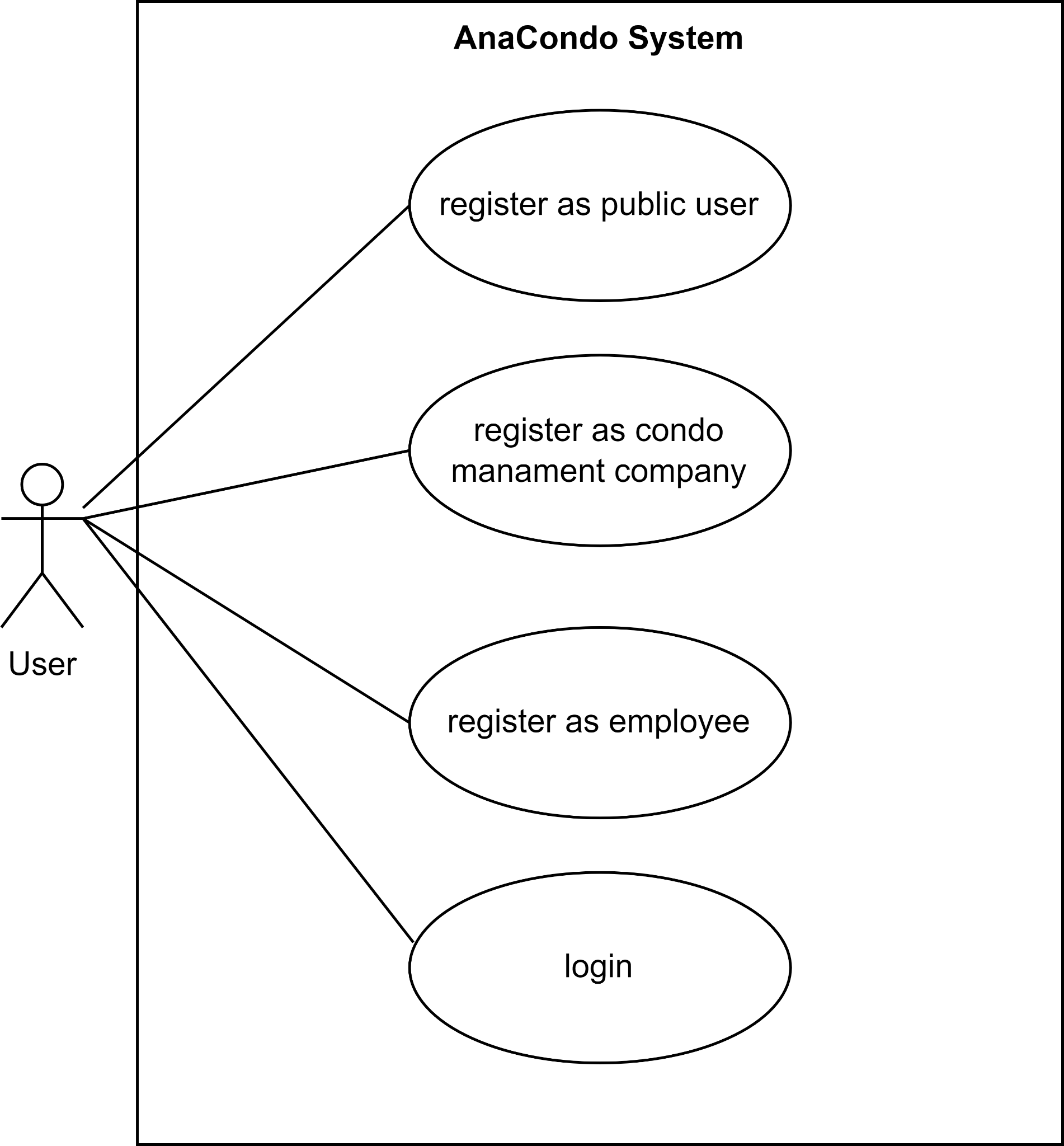


Figure 7: Use Case Public User System diagram

**UC1** Register as Public User: User can sign up as a public user using their google account or by filling a form

**UC2** Register as Condo Management Company: User can sign up as a Condo Management Company user using their google account or by filling a form

**UC3** Register as Employee: User can sign up as an Employee user using their google account or by filling a form

**UC4** Login: User can login with their google account or with an email and password already registered in the AnaCondo system

**Condo Management Company Use Case Diagram**

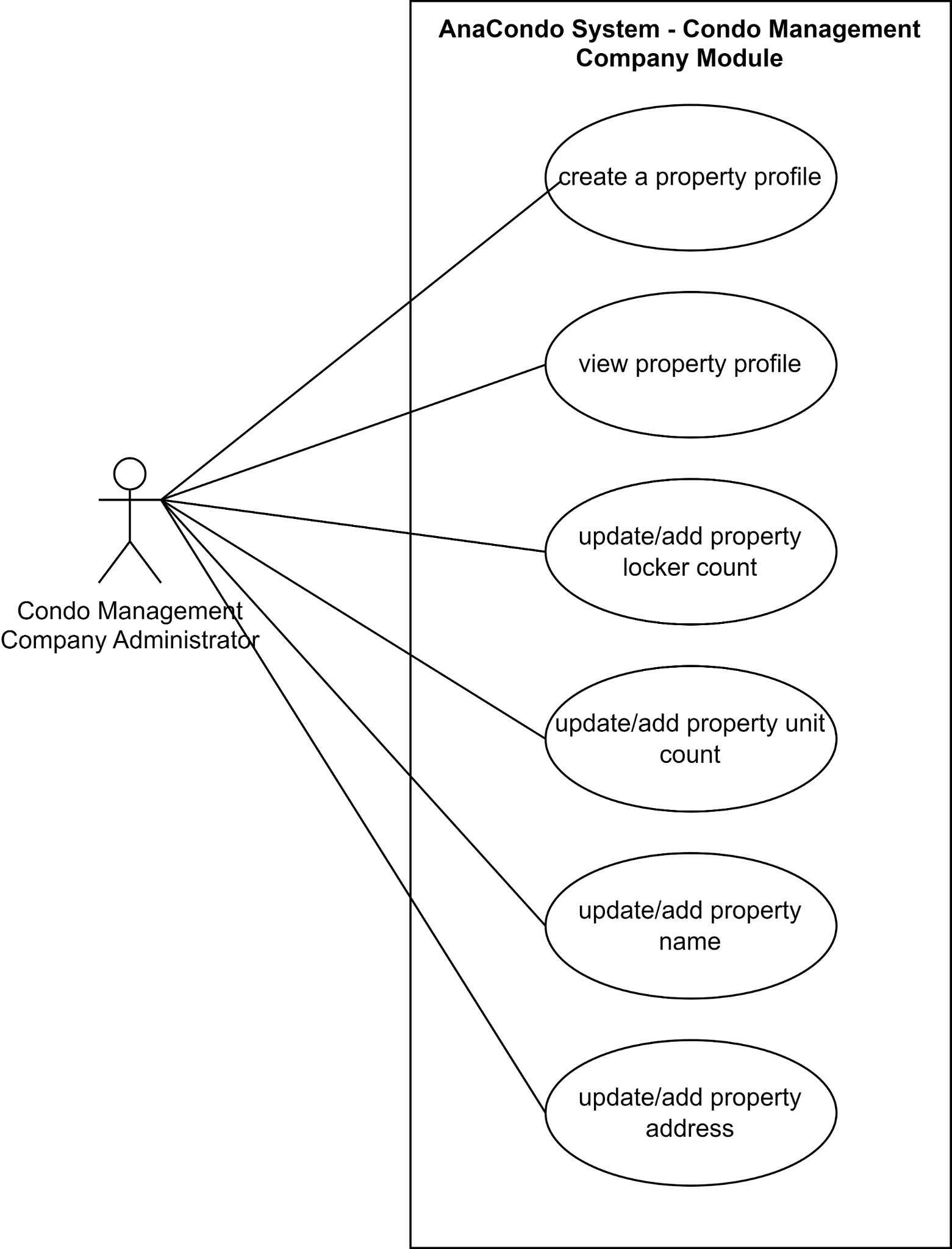
****

Figure 8: Management Company Module Use Case diagram

**UC1** Create a Property Profile: Condo Management Company Administrator can make a profile for a new property and add information about it

**UC2** View Property Profile: Condo Management Company Administrator can view the profile of a property already registered

**UC3** Update/Add Property Locker Count: Condo Management Company Administrator can add or modify the locker count of a property

**UC4** Update/Add Property Unit Count: Condo Management Company Administrator can add or modify the property unit count of a property

**UC5** Update/Add Property Name: Condo Management Company Administrator can modify the name of a property

**UC6** Update/Add Property Address: Condo Management Company Administrator can modify the address of a property

**Property File Use Case Diagram**

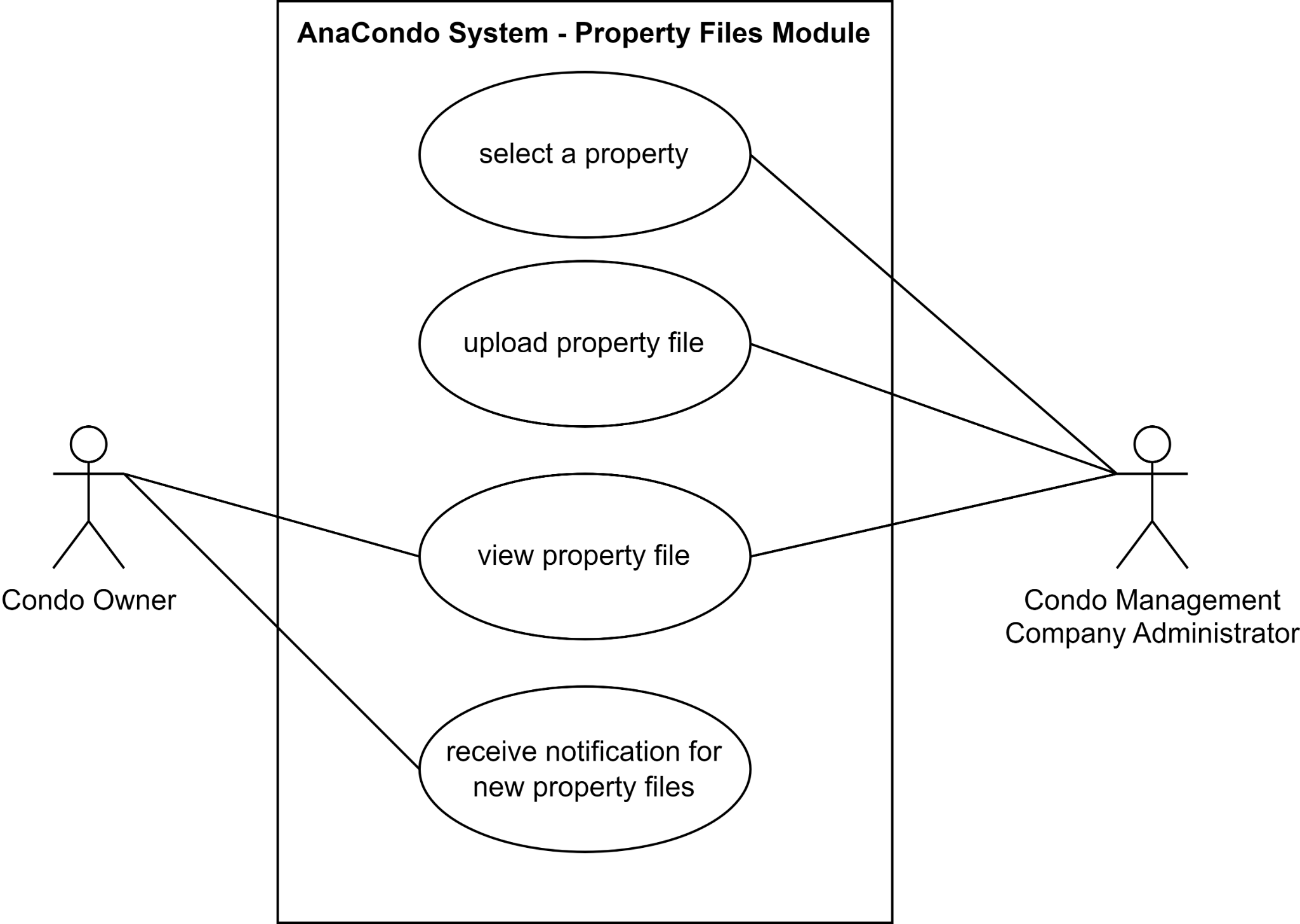


Figure 9: Property Files Module Use Case diagram

**UC1-UC2** Upload Property File: Condo Management Company Administrator can upload files relevant to the selected property

**UC3** View Property File: Condo Management Company Administrator and Condo Owner can view the files for the selected property

**UC4** Receive Notification for New Property Files: Condo Owner is notified when new files are uploaded for their property

**Registration Keys Use Case Diagram**

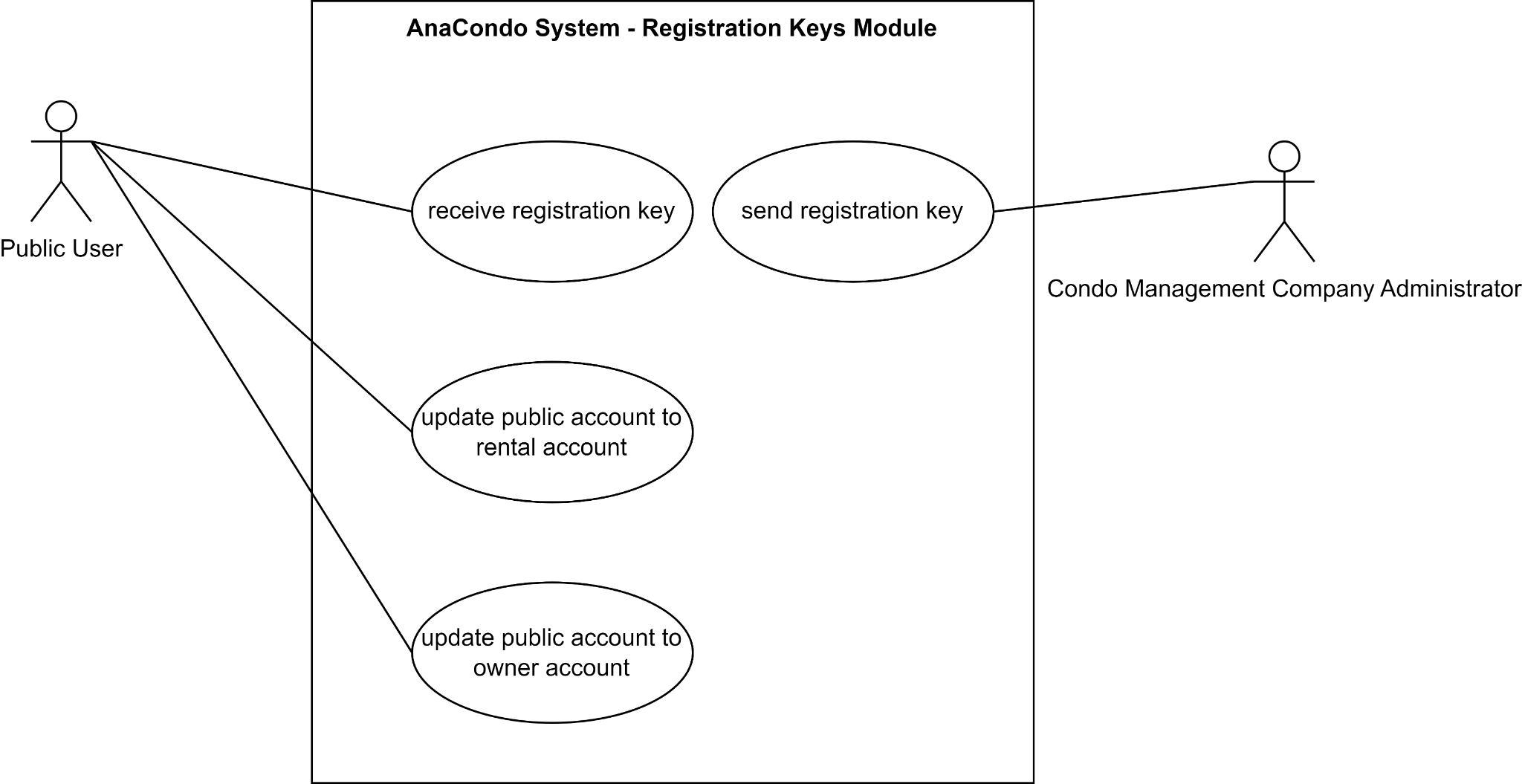


Figure 10: Registration Key Use Case diagrams

**UC1** Send Registration Key: Condo Management Company Administrator can send registration keys to specific users

**UC2** Receive Registration Keys: Public User can receive registration keys from Condo Management Company Administrator

**UC3** Update Public Account to Rental Account: Public user can update their account to a rental account using their registration key

**UC4** Update Public Account to Owner Account: Public user can update their account to a owner account using their registration key

**Reservations Use Case Diagram**

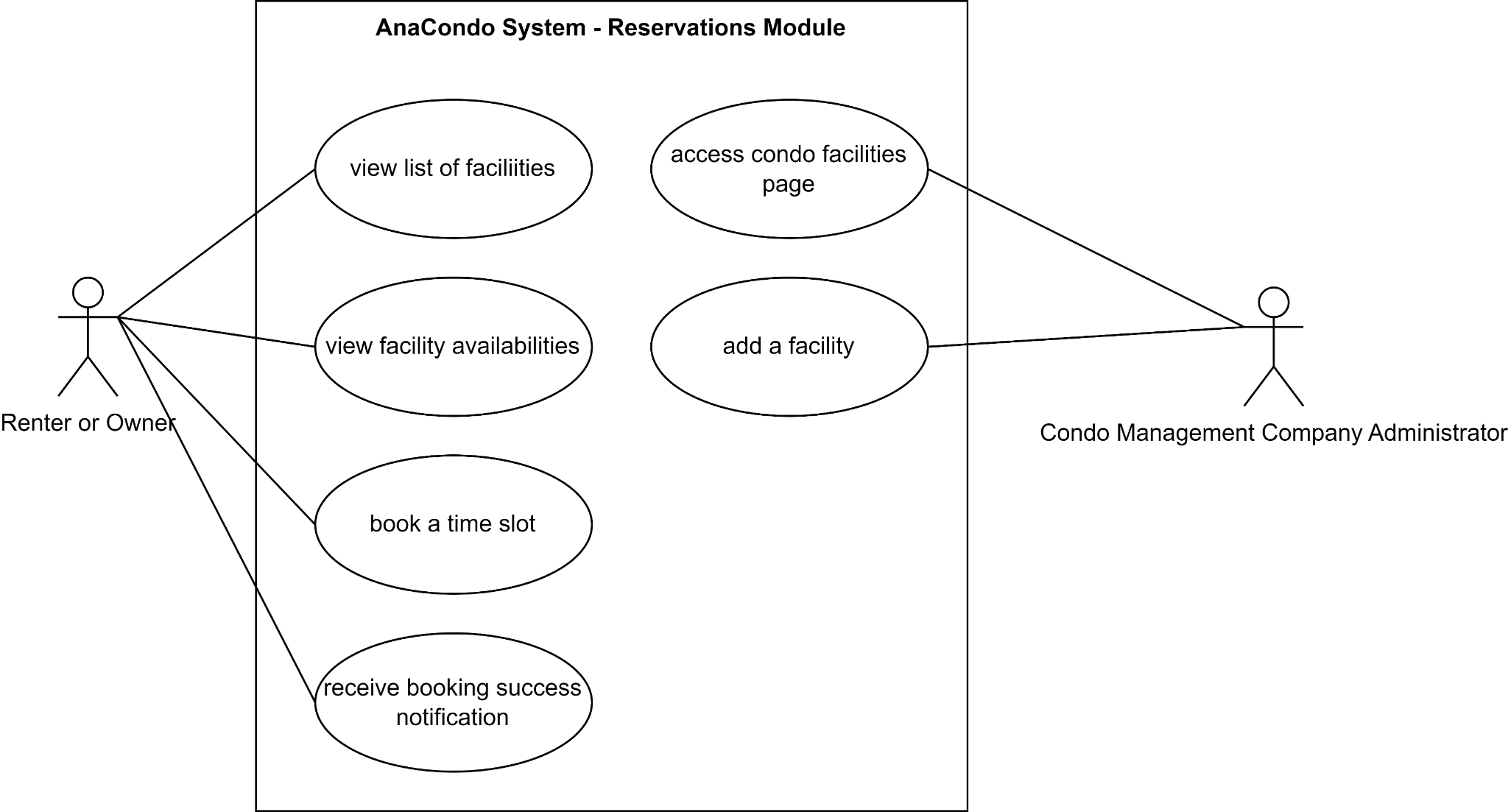


Figure 11: Reservation Use Case diagrams

**UC1 & UC2** Manage Facilities: Condo Management Company Administrator can add facilities or modify them

**UC3 & UC4** View Facility List and Availabilities: Renter and Owner can view a list of facilities and their availabilities

**UC5** Book a Time Slot: Renter and Owner can reserve a time slot for a specific facility

**UC6** Notification on Successful Booking: Renter and Owner receive a notification after successfully booking a time slot

# 5 Consistency and correspondences

## 5.1 Known inconsistencies

| Inconsistency | Description |
| --- | --- |
| 1. Incomplete Physical Viewpoint Diagram | The physical viewpoint can have a deployment diagram to represent how the components in the backend work together. We did not represent this with a deployment diagram because we do not intend to host our application using our own hardware but instead with the use of online services. |
| 1. Incomplete Sequence Diagram | The sequence diagrams presented in this document do not represent every single use-case since doing so would be redundant. The sequence diagrams shown represent some of the key features that will be used frequently. |
| 1. Incomplete Use Case Diagrams | The Use Case Diagrams do not represent every single use case of the application, the ones represented were key features that our application is based around. |
| 1. Chat Communication System | This feature is mentioned in ***Figure 5*** , but it was not included in any other model. This feature will be properly implemented once decisions about it are finalized. |

## 5.2 Correspondences in the AD

| Correspondence | Description | Concerned Figures |
| --- | --- | --- |
| User | In the Use Case Diagrams the User initiates the actions in the same way they do in the sequence diagram. | Figures 1-11 |
| Login And Registration | The Sequence diagram for the registration of public users in our system explicitly shows all the actions and steps taken by the user & system components to register a new account in our User database. The Use Case diagrams show the interaction and the possible action that a public user has inside our system. | Figure 3, 6 |
| Manage User Profile | The use case diagrams in figure 9 are all the possible ways a public user can update his type of account from public user to either Rental users, Condo Owners or Employee by adding a Key that the condo company will provide. | Figure 9 |
| Manage Reservations | In the figure 10, the use case diagrams we can see that the users (Owners/Tenants) have options to book a slot form the facilities for when they are free and view the different facilities connected to their Condo. In the diagrams, the Management Administrator is able to add a facility and also manage the existing ones. | Figure 10 |
| Manage Properties | In figure 4, the sequence diagram demonstrates how the User interacts with the system to add a new property: the user can input all information for a new property, then the system inputs the new information into the database. The database then stores the information and sends it back to the system. The system then displays to the user that the info has been registered. Figure 7 and 8 show use cases on how the condo management company administrator and condo owner may use the system | Figure 4, 7, 8 |

**Bibliography**

[1] Kruchten, P. (1995) Architectural Blueprints—The “4+1” View Model of Software Architecture. Available at: https://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf (Accessed: 28 January 2024).

[2]IBM. “Deployment Diagram. Available at: <https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=diagrams-deployment> (Accessed: 02 February 2024).