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

# **Contents**

[**Contents 2**](#)

[**1 Product Vision 6**](#_qbx3ae60swwx)

[1.1 Introduction 6](#_v2jmw19t0xx5)

[1.2 Positioning 6](#_w65m2ylwrqsw)

[1.2.1 Problem Statement 6](#_fvyoicyw2p7p)

[1.2.2 Product Position Statement 6](#_r99nfbnx9nwa)

[1.3 Stakeholder and User Descriptions 7](#_edh0fgxj6ufd)

[1.3.1 Stakeholder Summary 7](#_kdom5899id6r)

[1.3.2 User Summary 9](#_y22h3sdg0mu4)

[1.3.3 User Environment 10](#_qy8xsr9ikm8y)

[1.3.4 Key Stakeholder or User Needs 11](#_hrpcvka5u0g)

[1.3.5 Alternatives and Competition 15](#_xjjh62raftfp)

[1.4 Product Overview 15](#_ffvcda75exfh)

[1.4.1 Product Perspective 15](#_vxbcnqj3ya7t)

[1.4.2 Assumptions and Dependencies 15](#_qinryy6igycd)

[1.5 Product Features 16](#_x0a6hac9iobx)

[1.6 Other Product Requirements 16](#_pytq7pxit0fj)

[**2 Requirements and User Backlog 18**](#_u4le2eio648h)

[2.1 Legend 18](#_gb7rnci7t5rn)

[2.2 Epics and Requirements 19](#_g1vwc9n85rnf)

[2.2.1 Requirements 19](#_3o6nut206on4)

[2.2.2 Epics 22](#_eboze8gbf618)

[2.3 User Stories with Acceptance Criteria 24](#_r1w5uboay0r4)

[2.3.1 User Account Types 24](#_sxymf2t20cfu)

[2.3.2 Condo Management Companies 27](#_aqbo5ruv6wid)

[2.3.3 Reservation System 29](#_vb12r5v7hfgq)

[2.3.4 Communication and Events 31](#_uwseqknky0ua)

[2.3.5 Employee Roles 34](#_yc1e4jx1f6jr)

[2.3.6 Finance 36](#_iqck9nj967qb)

[2.3.7 Additional Features 38](#_ar9karhl5uy8)

[**3 Software Architecture Description 39**](#_i8frgkb5f61g)

[3.1 Introduction 39](#_v9cm8eroz454)

[3.1.1 Identifying information 39](#_bdew7v7tjitm)

[3.1.2 Supplementary information 39](#_ftaee1kgbgl2)

[3.1.3 Other information 39](#_ozmbdjmq9fgp)

[3.1.3.1 Architecture evaluations 39](#_2sxbuqao94xl)

[1.3.2 Rationale for key decisions 39](#_tf29qlfr8mwq)

[3.2 Stakeholders and concerns 40](#_j133vxtbuse8)

[3.2.1 Stakeholders 40](#_x8ojfb8ni3u3)

[3.2.2 Concerns 41](#_2u53lw4p4cyk)

[3.2.3 Concern–Stakeholder Traceability 42](#_44sinio)

[3.3 Viewpoints 44](#_2jxsxqh)

[3.3.A Logical Viewpoint 44](#_2c2ol1f6zol2)

[1. Overview 44](#_khp4za136qz3)

[2. Concerns and stakeholders 44](#_2usq0w8cx42u)

[3. Model kinds 46](#_w8hupk13jg7y)

[4. Views 47](#_y5fg49l6cj17)

[3.B. Development Viewpoint 48](#_crijciweguyl)

[1. Overview 48](#_rjlrkxfgy3l)

[2. Concerns and stakeholders 48](#_3gu08drpt1th)

[3. Model kinds 50](#_r3hfmkwlvmx0)

[4. Component Diagram 51](#_jfmspwn4sp37)

[3.C. Process Viewpoint 51](#_ofubx647d7h9)

[1. Overview 51](#_q7csia8dkoug)

[2. Concerns and Stakeholders 51](#_3d2biwb7xqxi)

[3. Model kinds 52](#_6noqw7nvccqh)

[4. Sequence Diagram 53](#_okr81es5as02)

[5. Activity Diagram 54](#_v5ttcrlmqs82)

[3.D. Physical Viewpoint 55](#_wzt1p2u2kand)

[1. Overview 55](#_en5dzwpmtdpw)

[2. Concerns and Stakeholders 55](#_5jvirdqcq9cn)

[3. Model Kinds 57](#_2x1osgcyf43s)

[3.E. Scenarios Viewpoint 57](#_91c1u1s8195s)

[1. Overview 57](#_vfvz1k3g2qmr)

[2. Concerns and Stakeholders 57](#_m664v9hqg6ft)

[3. Model Kinds 59](#_y9g56ox7mrdr)

[4. Use Case Diagrams 59](#_x20ibb878rtk)

[3.4 Consistency and correspondences 63](#_6kmreteyxryc)

[3.4.1 Known inconsistencies 63](#_9buo5e79ljg)

[3.4.2 Correspondences in the AD 64](#_woh2z4om7d3y)

[3.5 Bibliography 66](#_1k3asakx3v7a)

[**4 UI Prototypes 67**](#_vsm1fy85s29t)

[4.1 Prototypes for Website 67](#_bowo27e0zetd)

[4.2 Prototypes for App 70](#_tb5nkxaqc513)

[**5 Risk Assessment and Management Plan 73**](#_4dgywohfypo2)

[5.1 Purpose of the Risk Management plan 73](#_lkkgnfar2u26)

[5.2 Identification of risks and assessment 73](#_p595ee5j1yo)

[5.3 Criteria for Defining Impact and Probability Levels 74](#_sfezue638kok)

[5.4 Decision-Making Criteria 75](#_v6ygxypfwyaf)

[**6 Testing Report and Testing Plan for Sprint 5 83**](#_je9za2ch604i)

[6.1 Unit Testing 83](#_x5nrwj4p4tbh)

[6.2 Code Coverage 84](#_n1n45wxuwgvr)

[6.3 System Testing 86](#_aov4k7hbufg6)

[**7 Sprint 4 Retrospective 87**](#_ljqfq8g7bqry)

[7.1 Introduction 87](#_30j0zll)

[7.2 What went right 87](#_1fob9te)

[1 - Open Communication 87](#_3znysh7)

[2 - Team Management 87](#_2et92p0)

[3 - Coordination and Organisation 88](#_tyjcwt)

[4 - Adaptability and Problem Solving 88](#_3dy6vkm)

[5 - Collaboration Between Members 88](#_un85tod4mfcy)

[7.3 What went wrong 88](#_4d34og8)

[1 - Time Management 88](#_2s8eyo1)

[2 - Lack of Cohesion 89](#_17dp8vu)

[3 - Code Management 89](#_3rdcrjn)

[4 - Completeness 89](#_26in1rg)

[5 - Conclusion 90](#_lnxbz9)

[**8 Sprint 4 Release Plan 91**](#_d3yt1qfa9vw5)

[8.1 Completed Tasks 91](#_rk4cjyyi5bcc)

[**9 Code Management 93**](#_vsoajmh9y4d2)

[9.1 Software Implementation of Planned User Stories 93](#_unjkmrrgderk)

[9.2 Bug Reports and Fixing 95](#_ljiojwwgtsaa)

[9.3 Defect Tracking Tool 97](#_f11rytqkss5h)

[9.4 Code Review Tools 98](#_7tp1orp8rxrp)

[9.5 Usage of Design Patterns 99](#_8c62edxlx9m1)

[9.6 Code Coverage 100](#_o6v203cwhx0u)

[9.7 Design Quality as Measured 100](#_799dl2lvv1yy)

[9.8 Quality of Source Code Documentation 101](#_l3sh9m5b649z)

[9.9 Refactoring Activity Documented In Commit Messages 102](#_3s9w2dnqva9o)

[9.10 Quality and Detail of Commit Messages 102](#_ndoc1z3gfcri)

[9.11 Use of Feature Branches 103](#_zcx8ck2a9omn)

[9.12 Atomic Commits 103](#_7oncne3sm18c)

[9.12.1 Pull request message format: 104](#_uebwmbs02qz3)

[9.12.2 Commit message format: 104](#_zige2uwbmagq)

[9.13 Linking of Commits to Bug Reports/Features 104](#_29vpb9oegzfi)

# 

# **1 Product Vision**

## **1.1** **Introduction**

This document outlines the project vision for "AnaCondo," a feature-rich condo management system created to improve and simplify the administration of condominium buildings. "AnaCondo" is a platform that enables effective communication, financial management, and general property administration, responding to the interests of high-level stakeholders such as condo owners, renters, and condo management businesses.

## **1.2** **Positioning**

### **1.2.1 Problem Statement**

| The problem of | fragmented communication, manual property administration, lack of a transparent financial system, absence of a centralized platform |
| --- | --- |
| affects | Condo owners, Rental users, Condo Companies, Property Managers & Employees |
| the impact of which is | Limited understanding of financial aspects; Inefficiencies in property management; Communication gaps among stakeholders |
| a successful solution would be | * providing a complete system for all condo companies and clients * Financial transparency / log history of financial aspects * All help & documentation in arm’s length * User-Friendly system for all ages |

### **1.2.2** **Product Position Statement**

| For | Condo owners / Rental Users / Condo Management Companies / Staff |
| --- | --- |
| Who | need an application that will help with managing any services or real estate needs. |
| The “AnaCondo” | is a online system for Condo Management |
| That | has:   * very intuitive interface * really reliable way to communicate to a specific person / company * that will group more than just one condo company / owners * secured and transparents financials |
| Unlike | other systems where its only for one condo company (making it hard whenever there is a relocation); doesn’t have a good communication system or not a complete one; hard to use for certain type of users; the interface not being appealing |
| Our product | has a very good looking design while still being easy to use for everyone; has a complete communication system to connect you to the right person / company; thinking of security and transparency of financials aspects to users |

## **1.3** **Stakeholder and User Descriptions**

For this new system, the stakeholders that will be part of the Requirement Modeling Process are The condo owners, which are the people that owns a condo of a condo company, condo rental users, which are clients that are only willing to rent a condo from the company, the properties managers, which are the people that manages clients problems or organizes the management companies, the management companies, which are the companies that takes cares of maintaining the condo buildings, and the IT support & maintenance teams / development team, which needs to have a system that is easy to understand and not complex in the coding aspect with detailed documentation. All these stakeholders have 2 big requirements which is the need to have a system that has a very good communication implementation and intuitive interface to make it appealing and user-friendly. For the condo owners and rental users they will have a system that will make them have a communication system that is reliable and intuitive to use to communicate with their condo company managers and also manage their use of the condo facilities. The condo companies have a dashboard with all information about the facilities, finances and other aspects very clear and easy to use.

### **1.3.1 Stakeholder Summary**

| **Name** | **Description** | **Responsibilities** |
| --- | --- | --- |
| Support & Maintenance | These stakeholders are the people that will be helping to improve and fix bugs in a continuous way to prevent end-users not being able to access the information they need or to make actions that are urgent. | * Ensure system stability * Continuous Functionality * Data Backup & recovery * Scalability to users need * Documentation * Technical Support |
| Project Manager | They will be responsible of managing the entire project | * Execute the project according to the plan they have developed * Follow the plan and ensure project is organized, timely and budgeted |
| Condo owners | Public individuals who own individual condo unit | * Main users of the functionality in our app * Manage their tenants demands on the app * Store any important documentation in the app such as finances or important property demands |
| Rental User | Public individuals who will rent individual condo unit from the condo owners | * Use the features of the app while the duration of their lease * Create an account, manage their profile * Send requests through platforms * Contribute to conversations with other tenants * Communicate with management for specific services |
| Condo Management Company | Users who manage the condos and represents the company | * Create profiles on the app * Manage any financial aspects related to the company * Manages employees and distributes them tasks |
| Condo Management Employee | People who work for the condo management company | * Ensuring to complete their tasks * Communicating and helping the customers to the best of their abilities * Updating their superiors on the status of their tasks |
| App developer | People who will develop the entire system | * Develop all the required features * Ensure that the app will deliver to the user’s expectation |

### **1.3.2** **User Summary**

| **Name** | **Description** | **Responsibilities** | **Stakeholder** |
| --- | --- | --- | --- |
| Condo Owner | Individual condo unit owner | * Manages personal profile * Views property details * Handles financial interactions * Submits requests | Condo owners |
| Finance Manager | Manages all financial workers | * Manages financial transactions * invoicing specific documents to users * Making budgeting plans for the building * Creating payment plans for user | Condo Management company |
| Maintenance Manager | Manages all maintenance workers | * Oversees maintenance requests from users * schedules any repairs or specific maintenance requests | Condo Management company |
| Property Manager | Manages the property | * Manages property details * Updates and improves any condo units | Condo Management company |
| Rental User | Individual/s renting a unit | * Manages rental profile * Handles financial interactions * Submit requests | Rental Users |
| Customer Service Representative | Answers to condo renter/owners | * Handles the main inquiries and complaints | Condo Management company |
| Accounting Worker | Assists any financial needs | * Makes sure all finances on profiles and kept clean and updated * help process payments related to each units | Condo Management company |
| staff management | Employee responsible for daily operations | * Performs day-to-day operational tasks * Manages resident requests * Implements company policies | Condo Management company |
| Front desk Clerk | Greets people and handles on site demands | * Takes care of reception * Accepts packages and mails * Provides general assistance | Condo Management company |
| Security personnel | Ensure building is secure | * Provides a safe environment in the building * monitors access points * responds to immediate emergencies | Condo Management company |
| Condo Management Company | Represents management company | * Creates property profiles * Manages financial aspects * Handles unit and facility details * Distributes registration keys * Manages employee roles | Condo Management company |

### **1.3.3 User Environment**

* Support and maintenance will continuously improve our system and ensure that it runs as smoothly as possible while fixing bugs. This team will consist of around 10 to 15 people and can vary depending on the workload. The team will most likely be working from an office using a preferred computer to work on their tasks. They will require strong wifi network connection.
* Condo owners/rental users will create profiles and manage them on our system. They will also refer to financial documents regarding their condo on a specific page on their account. They will submit requests for maintenance or any kind of service needed and track the progress. They will need a compatible browser to use our application or a mobile phone.
* Staff management will use the system to manage the requests sent in by the residents and will ensure that they are being tended to by updating their status. They will also use the system to store the company policy documentation. Anacondo will also help manage the operational tasks to be done and track what is left to do.
* Condo Management Company will manage financial obligations by uploading documents on pages to specific profiles. They will also create profiles to showcase properties. They will also use the communication pages to divulge important messages about the condominium management. They will also manage their employees by using the platform to assign roles and to schedule the employees.

### **1.3.4 Key Stakeholder or User Needs**

Lack of communication within condominium:

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Broadcast messages on messaging channels | high | Lack of communication within condominium | Send email to potentially the right person | | Enable users to have a page where they can communicate directly within the condominium |
|  |  |  |  |  |  |

Lack of maintenance and priority on resolving issues:

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Page to request maintenance services | high | Lack of maintenance and priority on resolving issues | Send email for assistance but with the lack of details in email process takes longer | | Maintenance page on our application where you can request service and also track the status of that specific request and get direct and quick feedback |

Poor financial management:

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Page to manage financial obligations | high | Mismanagement of financial aspects where irregularities in accounting occur | Paper based system where documents are not organized and financial matters are dealt non efficiently. | | Users can use our finance page dedicated to keeping track of all financial obligations. Make sure that there are no discrepancies or missing sums in transactions. The page may also store any financial documents such as invoices and receipts. |

Profile Management

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Users need to be able to create, edit and add a profile to use the application | high | Without the profiles, the company won’t know the request and services belong to who | Send email for assistance but with the lack of details in email process takes longer | | Each user will have a profile associated with them and will be able to personalize it. |

Creating new property

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Condo management company need to create new properties on the website | high | Without the profiles, the companies won’t be able to save new information about newer property and won’t be able to add it to the system | Send email or manually ask register the new property and update it with new information every time | | Condo owners will be able to create property profiles for the condo management companies to utilize them. |

Updating property information

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Condo management company need access to edit properties and hold records to proper information | high | Without the profiles, the companies won’t be able to save new information added about the properties | Send email or manually ask register the new property and update it with new information every time | | Condo owners will be able to update property profiles for the condo management companies to utilize them. |

Buy properties

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Potential condo owners will need to be able to browse through different properties up for sale | medium | Without the properties up for sale, sellers will struggle to advertise and buyers will have less luck finding what they like | browse online and try to find a perfect condo to buy | | Our properties to sell will have all the needed information and will be all displayed for the user’s best interest |

Sell properties

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Potential condo buyers will need to be able to browse through different properties up for sale | medium | most sellers struggle to sell their properties because of poor advertisement | try to post the properties on websites to advertise but will not be efficient | | Our page will be able to showcase the properties for sale with all the right information and will not give the owner a hard time |

Reserving a Facility

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Many users would want to reserve facility in their building for personal reasons such as the tennis court, office room, etc | high | Most users don’t know how to proceed to reserve common facilities | No reservation and overcrowded common rooms | | Users can request to reserve common rooms to complete specific needs. The BBQ area for an afternoon to enjoy a good meal with friends. The Office area to have a personal working area form 2 pm to 5 pm. |

Set up a new Facility

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Condo management would want to be able to add a new facility in the building | high | Need to document the changes in the building if new facility is to be integrated | No documentation or way to show users that there is a new facility | | Adding the new facility will allow the users to then reserve it and be able to use it like the other common facilities |

Request Submission

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Many users would want to be able to send requests to management about issues such as requesting new fobs, reporting a violation,... | high | Most users need to communicate their needs to management but its is not effective because there are no designated areas for these concerns | Send email for assistance but with the lack of details in email process takes longer | | Users can go on our website and fill in a request for which ever need they need and will get a quick feedback from management |

Calculating Condo fees

| **Need** | **Priority** | **Concerns** | **Current Solution** | **Proposed Solutions** | |
| --- | --- | --- | --- | --- | --- |
| Many condo owners buy condos and don’t know that there is a condo fee every month. This will give them information about condo fees and will calculate the prices for them. | high | Condo owners will not know how much they need to pay each month for their condo fees | Send finance through paper mail and shock owners with the amount of money they owe for condo fees | | This tool will allow owners to know how much condo fees they owe |

### **1.3.5 Alternatives and Competition**

Stakeholders will use alternatives available to them to solve their issues. Instead of using our application, stakeholders can stick to traditional methods and manage with a paper based system or use external email platforms. When needed to file a request they will either fill in a form and send it to the management team in their office or they can send the request from their personal mail. The strengths for this option is that you can have paper proof of the requests and have heavy documentation. However, you can also lose or misplace a document leading to problems where the person who sent the request will get feedback. As for the emails, sometimes they may end up in junk mail leading to the same problem. They can also use another third-party platform that will most likely be our competitor. The strength for this option is that it will most likely fill the stakeholders needs, however our option is better because our app can be customized to their liking. For example, if they want to add a page for their new service they can.

## **1.4 Product Overview**

### **1.4.1 Product Perspective**

AnaCondo an independent product and is not related to external products or environments. It is self-contained. All of the application’s systems and layers will be built from within our team.

### **1.4.2 Assumptions and Dependencies**

In the rare cases where the user does not have access to the internet to be able to use our application, then any features present will not be at his disposal. Moreover, if the stakeholder and users decide to change a requirement, it will affect this Vision document. Another factor would be design, if the stakeholder does not approve of the design interface then we will also need to make alterations.

## **1.5** **Product Features**

Some key features Anaconda will have are:

Maintenance management: Users will be able to send in requests for service and this feature will also allow them to track the status of their submission and receive notifications. Maintenance team can use the same platform to create a schedule for the staff and to assign them specific service requests.

Financial management: Users can use our financial page to pay for any assessments such as a resident paying monthly condo fees or rent. The page will give them access to the invoice and receipt once they have paid to keep it as history.

Communication portal: Users can use this page as a means to talk to other residents. They can create channels and message whoever they need. There is also an announcement tab where news and important messages can be broadcasted to the residents. Residents can receive notifications from the community with updates about certain matters or emergency alerts.

Common area booking page: Most condominiums have shared common areas, however you can rent them out for specific private events. Therefore, users can go on the booking page and reserve specific dates for their events. The calendar will show available time slots.

Securitized portal: all the information about the condo community can be sensitive and private to the residents, therefore there will be a secure login to have access to specific pages such as important personal documents.

Visitor access: Online service to register guests before they come to visit. This allows the management team to have a log of all visitors for security purposes.

## **1.6** **Other Product Requirements**

High security to handle data privacy and to protect against cyber threats.

The application must be used on all web browsers such as google, safari, edge…It should have a design that will please the user’s experience but also be responsive. It also needs to be scalable for future modifications for the user’s needs.

It should be high performing and fast loading, it should have efficient database management to be able to manage all the user’s requests, needs, documents… High performing enough to have real time notifications and updates.

Cloud-based deployment to ensure that we are reducing the need and cost on hardware.

High performance: good response time and throughput to handle as many users as possible.

Robustness: our app will be able to handle errors and give feedback to users instead of crashing.

Fault tolerance: if ever a failure occurs, AnaCondo has a backup system that will temporarily be up and running for the users in the meantime.

Usability: the UI will be very aesthetic but also straightforward and will allow users to find what they need without wasting their time.

Security: users will need to authenticate themselves when entering the system to ensure that only authorized people are accessing specific things.

Reliability: our system will have tools to monitor its performance and ensure that it is at its best for the users. This tool will show the system’s health and performance metrics.

For design constraints, our system will be flexible and allow for future changes when users want to add a page or other features. Therefore, we need to keep in mind the space and aesthetic to add those options into our system.

For external constraints, our system needs to ensure that it will still be responsive on different types of devices such that it can go from a web browser on a mobile phone to a laptop.

For other dependencies, some people are not as used to technology therefore, we need to set some time aside for the people that will need a tutorial on how our product works.

A user manual may help with the older generation that will use our product. The user manual will explain to them in detail what our product is capable of doing and how they can utilize it. AnaCondo also needs to anticipate people who will refer to online help and how the team will cope with that aspect.

The priority of the other requirements are medium level, the main concerns are the main features that will satisfy the users. The other requirements are beneficial for the users however it can be risky for AnaCondo team members because it creates more pressure to deliver on time and to respect other aspects such as budget and resources.

# **2 Requirements and User Backlog**

## **2.1 Legend**

| **ID** | Unique number identifying user story in product backlog. Typically assigned by electronic tool. |
| --- | --- |
| **Title** | Simplified title of user story. |
| **Description** | Description of user story in user story format. |
| **Epic name** | The path to user story from user story map in case of complex requirements. |
| **MoSCoW** | How much the customer needs this story.  **M** – must  **S** – should  **C** – could  **W** – won’t |
| **Business Value** | How important this user story is from our perspective based on our business case.  **S** – small  **M** – medium  **L** – large  **XL** – extra large  **XLL** – larger than extra large |
| **Risk** | The risk level connected to development of this user story.  **L** – low  **M** – medium  **H** – high |
| **Effort** | How complex this user story is. Evaluated in story points from Fibonacci scale. |

## 

## **2.2 Epics and Requirements**

### **2.2.1 Requirements**

*Requirements associated with Epic 01*

* Public users can create their own (unique) profile. The minimum information needed in a profile is below.
* A profile should at least include a profile picture, user name, contact email, phone number.
* Public users need to provide a registration key obtained from their condo management company to become condo owners in the system.
* Public users need to provide a registration key obtained from their condo management company to become rental users in the system.
* Condo owners can have a good view (dashboard) of their properties, including general information, e.g., personal profile, condo information, financial status (i.e., remaining balance in terms of monthly condo fee payments), status of the submitted request, etc.

*Requirements associated with Epic 02*

* Condo management companies can create a profile for a property under their management.
  + The property profile should have at least property name, unit count, parking count, locker count, address.
  + Condo management companies can upload condo files for each property. The condo files are accessible to all condo owners of that property. Examples of such condo files (in pdf) include condo declarations, annual budgets, board meeting minutes etc.
  + Condo management companies can enter detailed information for each condo unit, each parking spot, and each locker in a building. Basic information of a condo unit includes unit id, size, unit owner, occupant information (i.e., may be occupied by a rental user instead of the owner), as well condo fee associated with the unit. Basic information of a parking spot (or a locker) includes parking spot id, spot owner, occupant information, and condo fee associated with the parking spot.
  + Condo management companies can send registration keys to unit owners or rental users for their dedicated units. Such registration keys will be used by unit owners or rental users to link a condo unit with their profiles.
  + Condo management companies can enter condo fee per square foot, per parking spot.
  + Discounts, offers. Condo management companies can choose to list coupons/offers that are visible to all unit owners or rental users of one property.

*Requirements associated with Epic 03*

* Condo management companies can set up different roles for different employees, who are responsible for the same property. Such roles include manager, or someone who is responsible for daily operations, or someone who is responsible for finance.

*Requirements associated with Epic 04*

* The condo management system contains a simplified financial system.
  + Condo fee of each unit will be calculated and presented to unit owners.
  + The financial system records operational budget (i.e., the collected condo fee) and cost. Condo management companies can enter the cost for each operation.
  + The financial system can generate an annual report. For example, all the condo fee collected for a given year.

*Requirements associated with Epic 05*

* The condo management system contains a simplified reservation system.
  + Condo management companies set up the common facilities, which require reservations. Examples include a sky lounge, a spa fitness center, etc.
  + The reservation system allows condo owners and rental users to reserve common facilities in a calendar-like interface.
  + The reservation system should show availability of common facilities.
  + The reservation is first-come-first-serve. Once a facility is booked, it will become unavailable for the reserved time.

*Requirements associated with Epic 06*

* A forum where users can post and reply.
* Events. Users can organize events and invite other occupants to attend.
* All the users have a notification page, where they can see the latest activities in submitted or assigned requests.

*Requirements associated with Epic 07*

* Condo owners can submit requests.
  + Examples of such requests include moving in/out (date for reserving elevators), intercom changes, requesting access (fobs, keys), reporting a violation, reporting deficiency found in common areas, or asking a question.
  + Each request will be assigned to a corresponding employee based on the type of the request.

### **2.2.2 Epics**

| **01** | **Create User Profile** | | |
| --- | --- | --- | --- |
| As a user, I can create a profile in which my personal information will be stored, so that I may access all the functionalities of the system. | | | |
| M | XL | L | 3 |

| **02** | **Create and Update Condos** | | |
| --- | --- | --- | --- |
| As a condo management administrator, I want to add and modify the properties on my dashboard. Further, I may modify the finer details of each property and must be provided the fields to do so. | | | |
| M | XL | H | 8 |

| **03** | **Communication with Occupants** | | |
| --- | --- | --- | --- |
| As a condo management administrator, I want an interface in which I may communicate with the occupants and employees of any condominium, so that they can be updated of any changes, requests, discounts, or registration keys. | | | |
| M | XL | M | 5 |

| **04** | **Financial Profile** | | |
| --- | --- | --- | --- |
| As a condo management administrator, I want a financial profile to be collected on each condominium in my dashboard, so I may supervise company finances. | | | |
| M | XL | M | 5 |

| **05** | **Reservations** | | |
| --- | --- | --- | --- |
| As a user, I want access to a reservation database of common facilities so I may either reserve time slots for common facilities or set up new common facilities. | | | |
| M | XL | M | 8 |

| **06** | **Communication with Management** | | |
| --- | --- | --- | --- |
| As an occupant, I want an interface in which I may communicate with management and other occupants, so that I may plan events and submit requests. | | | |
| M | XL | H | 5 |

| **07** | **Requests Management** | | |
| --- | --- | --- | --- |
| As a condo owner, I want to be able to submit a request for condo facilities and issues and expect a timely resolution, so that I may provide adequate management of my property. | | | |
| M | XL | H | 8 |

## **2.3 User Stories with Acceptance Criteria**

### **2.3.1 User Account Types**

| **~~0101~~** | **~~Public User Account~~** | | |
| --- | --- | --- | --- |
| ~~As a public user,~~  ~~I want to be able to create my own unique profile with my personal profile picture, user name, contact email and phone number.~~  ~~So that I may properly access the functionalities offered and manage my condo(s).~~ | | | |
| ~~Create User Profile\Public User Account\~~ | | | |
| ~~M~~ | ~~M~~ | ~~L~~ | ~~2~~ |

**~~0101~~** ~~Acceptance Criteria~~

1. ~~When entering the app for the first time, the user must be prompted to create an account (or login).~~
2. ~~A public account user must be able to enter all of his information on one screen.~~
3. ~~A public account user must be able to modify their personal profile picture, username, contact email, and phone number once their account is created (under the account section).~~
4. ~~A public account user must be able to enter a registration key.~~
5. ~~A public account user who does not enter a registration key must not have access to condo owner or rental user account functionalities.~~



| **~~0102~~** | **~~Logging Into Pre-Existing Account~~** | | |
| --- | --- | --- | --- |
| ~~As a user, I want to be able to log out and into my account, so I can access the contents of my account at any time.~~ | | | |
| ~~Create User Profile\Logging Into Pre-Existing Account\~~ | | | |
| ~~M~~ | ~~M~~ | ~~M~~ | ~~2~~ |

**~~0102~~** ~~Acceptance Criteria~~

1. ~~The system must recognize the username and open the appropriate account.~~
2. ~~The username of a condo owner should be granted access to an administrator’s or management’s account.~~
3. ~~Login credentials must be stored in a database.~~
4. ~~If the account does not exist or the credentials are wrong, an error message should pop up.~~



| **0129** | **Logging Out of User Account** | | |
| --- | --- | --- | --- |
| As a public user, I want to be able to log out of my account, so that I may end my session and close the application. | | | |
| Create User Profile\Logging out of User Account\ | | | |
| M | M | L | 2 |

**~~0129~~** ~~Acceptance Criteria~~

1. ~~The user must be able to successfully log out.~~
2. ~~After logging out, the user must be taken to the home page.~~
3. ~~The user must be logged in to logout.~~



| **~~0103~~** | **~~Condo Owner Account~~** | | |
| --- | --- | --- | --- |
| ~~As a condo owner, I want to be able to create a condo owner account with a registration key, so that I may access the condo owner functionalities such as managing one of my condo(s).~~ | | | |
| ~~Create User Profile\Condo Owner Account\~~ | | | |
| ~~M~~ | ~~M~~ | ~~L~~ | ~~2~~ |

**~~0103~~** ~~Acceptance Criteria~~

1. ~~A public user account must be updated to a condo owner account when linked with the correct registration key.~~
2. ~~A condo owner account user must only be able to access the condo owner functionalities within the system.~~



| **~~0104~~** | **~~Rental User Account~~** | | |
| --- | --- | --- | --- |
| ~~As a rental user, I want to be able to create a rental user account with a registration key, so that I may access the rental user functionalities.~~ | | | |
| ~~Create User Profile\Rental User Account\~~ | | | |
| ~~M~~ | ~~M~~ | ~~L~~ | ~~2~~ |

**~~0104~~** ~~Acceptance Criteria~~

1. ~~A public user account must be updated to a rental user account when linked with the correct registration key.~~
2. ~~A rental user account must only be able to access the rental user functionalities within the system.~~



| **~~0105~~** | **~~Employee User Account~~** | | |
| --- | --- | --- | --- |
| ~~As an employee of one of the condominiums, I want to create an account where I may have access to an employee dashboard listing my employee role, so that I may access the current requests submitted by the condominium manager and/or the condo company administration.~~ | | | |
| ~~Create User Profile\Employee User Account\~~ | | | |
| ~~M~~ | ~~M~~ | ~~L~~ | ~~2~~ |

**~~0105~~** ~~Acceptance Criteria~~

1. ~~Employee accounts must be stored in a database with the employee type.~~



| **~~0106~~** | **~~Dashboard of Properties~~** | | |
| --- | --- | --- | --- |
| ~~As a condo owner, I want to view all my properties in an easy-to-navigate and minimalist interface, so that I may know the remaining balance on my monthly condo fee payments, status of submitted requests and general condo information on my properties.~~ | | | |
| ~~Create User Profile\Dashboard of Properties\~~ | | | |
| ~~M~~ | ~~L~~ | ~~M~~ | ~~8~~ |

**~~0106~~** ~~Acceptance Criteria~~

1. ~~The properties dashboard must only be available to a condo owner account.~~
2. ~~The dashboard must have a user-friendly design with a rich user experience.~~

### 

### **2.3.2 Condo Management Companies**

| **~~0107~~** | **~~Condo Management Company Account~~** | | |
| --- | --- | --- | --- |
| ~~As a condo management company administrator, I want to create a profile for our properties, so that we may manage our property profiles.~~ | | | |
| ~~Create and Update Condos\Condo Management Company Account\~~ | | | |
| ~~M~~ | ~~M~~ | ~~L~~ | ~~3~~ |

**~~0107~~** ~~Acceptance Criteria~~

1. ~~The administrator can successfully add a property to their dashboard through a button on their admin account page when logged in.~~
2. ~~The form to create a property profile should include fields for the property name, the unit count, the locker count and address.~~
3. ~~Once the form has been submitted, the new created profile must be listed and viewable from the admin account page dashboard.~~
4. ~~The system successfully registers the new property into the appropriate database.~~
5. ~~The property must be accessible only through the administrators account credentials.~~

| **~~0208~~** | **~~Upload Files For Properties~~** | | |
| --- | --- | --- | --- |
| ~~As a condo management company administrator, I want to be able to upload files for each of my properties so that they may be accessible to all condo owners of that property.~~ | | | |
| ~~Create and Update Condos\Upload Files for Properties\~~ | | | |
| ~~C~~ | ~~M~~ | ~~L~~ | ~~3~~ |

**~~0208~~** ~~Acceptance Criteria~~

1. ~~Condo files uploaded by the administrator must be accessible to the condo owner of that property.~~
2. ~~Only an administrator logged in with the company administration credentials may upload files.~~
3. ~~Administrators must be able to interact with the properties listed on their dashboard so they can upload files.~~
4. ~~The system must search for all condo owners of that property and notify them that a file has been shared with them concerning their property.~~

| **~~0209~~** | **~~Property Information~~** | | |
| --- | --- | --- | --- |
| ~~As a condo management company, I want to be able to enter and modify information concerning units, parking spots and lockers in a building, so that I can properly manage my properties.~~ | | | |
| ~~Create and Update Condos\Property Information\~~ | | | |
| ~~C~~ | ~~L~~ | ~~M~~ | ~~5~~ |

**~~0209~~** ~~Acceptance Criteria~~

1. ~~Only condo management company users can enter and modify property details.~~
2. ~~Property information must be available to view to all public users.~~

| **0310** | **Send Registration Keys to Condo Owners and Rental Users** | | |
| --- | --- | --- | --- |
| As a condo management company administrator, I want to digitally send registration keys in batch or individually to Condo Owners or Rental Users so that their units can be linked to their profile. | | | |
| Communication With Condo Owner/Rental User\Send Registration Keys\ | | | |
| M | L | M | 5 |

**0310** Acceptance Criteria

1. Only condo management company users can send registration keys.
2. The system must notify a unit owner and rental user when a registration key is sent to them.
3. Unit owners must be able to enter their registration key by clicking a button which prompts them to enter the registration key they received.

### 

### **2.3.3 Reservation System**

| **~~0511~~** | **~~Reserving A Common Facility~~** | | |
| --- | --- | --- | --- |
| ~~As an occupant, I want to receive the availability of and reserve a common facility in a calendar like interface, so that I may enjoy the amenities offered by the condo company.~~ | | | |
| ~~Reservations\Reserving a Common Facility\~~ | | | |
| ~~S~~ | ~~L~~ | ~~H~~ | ~~8~~ |

**~~0511~~** ~~Acceptance Criteria~~

1. ~~Occupants of their condominium must be able to navigate to a facilities page, in which all facilities offered by the condominium will be listed.~~
2. ~~Each facility listed must be interactable, sending the occupant or user to a calendar interface in which they can view the facility’s availability.~~
3. ~~Upon selecting an available time slot, the system should display a button which allows the user to confirm their booking.~~
4. ~~Once a valid booking is confirmed, the system should block off that time slot so that other users cannot book the facility for that day and at that time.~~
5. ~~The system should include validation checks, disallowing users to book a time slot which has already been logged in the system.~~
6. ~~The reservation system should update the visual interface of the calendar when the booking is made.~~
7. ~~A log of all reservations should be kept by the system.~~
8. ~~A notification should be sent to the occupants notification page upon successful booking.~~

| **~~0512~~** | **~~Setting Up Common Facility~~** | | |
| --- | --- | --- | --- |
| ~~As a condo management company administrator, I want to set up and create new common facility interfaces whereby condo owners and rental users may access the interface to reserve a slot for the facilities offered so that we may continue to grow a loyal customer base and stellar reputation.~~ | | | |
| ~~Reservations\Setting Up Common Facility\~~ | | | |
| ~~M~~ | ~~L~~ | ~~M~~ | ~~5~~ |

**~~0512~~** ~~Acceptance Criteria~~

1. ~~Administrators must be able to navigate towards their facilities page for a condominium, accessed solely through an administrators account.~~
2. ~~The facilities page should have a form that allows the administrator to add a new common facility to the condominium.~~
3. ~~The form should contain the necessary fields to detail the purpose of the facility.~~
4. ~~The created facility must be added to a facilities database.~~
5. ~~The created facility must be intractable by the occupants of the building, so they may reserve time slots.~~

| **~~0713~~** | **~~Public Users Notifications~~** | | |
| --- | --- | --- | --- |
| ~~As a public user, I want to be notified of any activities in my submitted or assigned requests in a common interface or page so that I may easily access and monitor the latest activities in a timely manner.~~ | | | |
| ~~Communication with Management\Public Users notifications\~~ | | | |
| ~~S~~ | ~~L~~ | ~~H~~ | ~~3~~ |

**~~0713~~** ~~Acceptance Criteria~~

1. ~~Users must be able to navigate to a notifications page in a menu.~~
2. ~~The notifications page must be visible to all users, such as condo company administrators, condo owners and renters.~~
3. ~~The notification page should contain a list of all current and past notifications.~~
4. ~~Notifications should include if a user has booked a common facility, if a company administrator has uploaded a file to their condominium, events, submitted and assigned requests.~~
5. ~~Each notification must display the notification message and time it was sent.~~
6. ~~Notifications should be saved in a database.~~

### **2.3.4 Communication and Events**

| **~~0614~~** | **~~Message Board~~** | | |
| --- | --- | --- | --- |
| ~~As a public user, I want to have a page where I am able to post messages to a forum, so that I may communicate with other members in the condo building and address issues.~~ | | | |
| ~~Communication with Management\Message Board\~~ | | | |
| ~~C~~ | ~~M~~ | ~~H~~ | ~~2~~ |

**~~0614~~** ~~Acceptance Criteria~~

1. ~~The forum page must be navigable from the menu.~~
2. ~~Users should include administrators, condo owners and renters.~~
3. ~~The page should allow the user to submit a message from a text box.~~
4. ~~Messages should be displayed on the forum with the username of the account that had posted it.~~
5. ~~Messages should be posted in chronological order.~~

| **~~0615~~** | **~~Organize Events Through Portal~~** | | |
| --- | --- | --- | --- |
| ~~As a public user, I want a dedicated page where I can organize events through the portal and invite other occupants of the condo building to attend using my user profile so that I may facilitate community engagement.~~ | | | |
| ~~Communication with Management\Organize Events\~~ | | | |
| ~~S~~ | ~~M~~ | ~~M~~ | ~~3~~ |

**~~0615~~** ~~Acceptance Criteria~~

1. ~~The events page must be navigable from the menu.~~
2. ~~The page should contain a form field that the current user may use to create an event.~~
3. ~~The form should have the fields for the title of the event and a dropdown list of all members in their condominium that they may invite.~~
4. ~~Invited members should receive notifications on their notification page.~~
5. ~~The dropdown list must be taken from a database of occupants from the user’s condominium.~~

| **~~0316~~** | **~~Discount Offers~~** | | |
| --- | --- | --- | --- |
| ~~As a condo management company administrator, I want an announcements page with a dedicated section for coupons, so that we may offer discounts to the occupants of our condo building(s).~~ | | | |
| ~~Communication with Occupants\Discount Offers\~~ | | | |
| ~~S~~ | ~~M~~ | ~~L~~ | ~~2~~ |

**~~0316~~** ~~Acceptance Criteria~~

1. ~~Only a condo management company profile can list coupons/offers.~~
2. ~~Coupons/offers must be visible to all unit owners and rental users of a property.~~
3. ~~Users should be notified when a new coupon/offer is listed.~~



| **~~0717~~** | **~~Request Submission~~** | | |
| --- | --- | --- | --- |
| ~~As a condo owner, I want to be able to submit requests concerning moving in/out, elevator reservation, intercom changes, access requests, reports, or questions in order to achieve seamless management of my property.~~ | | | |
| ~~Requests Management\Request Submission\~~ | | | |
| ~~C~~ | ~~M~~ | ~~M~~ | ~~5~~ |

**~~0717~~** ~~Acceptance Criteria~~

1. ~~Condo owner is able to open a request.~~
2. ~~Condo owner is able to choose the request type.~~
3. ~~Condo owners are able to submit concerned units and condo units.~~
4. ~~Condo owner is able to input current date, occupant information and if applicable, date the request is intended for.~~
5. ~~Condo owner has communication with management.~~

| **~~0318~~** | **~~Request Assignment~~** | | |
| --- | --- | --- | --- |
| ~~As a Manager, I want to be able to assign requests to employees so that all requests are addressed and able to be fulfilled.~~ | | | |
| ~~Request Management\Request Assignment\~~ | | | |
| ~~M~~ | ~~M~~ | ~~M~~ | ~~5~~ |

**~~0318~~** ~~Acceptance Criteria:~~

1. ~~Manager is able to consult requests.~~
2. ~~Manager is able to browse Employees based on request type.~~
3. ~~Manager is able to view Employee availabilities.~~
4. ~~Manager is able to assign a request to an Employee.~~
5. ~~Manager is able to add notes to an assignment.~~
6. ~~Manager is able to communicate with the Condo Owner and Employee.~~

### **2.3.5 Employee Roles**

| **~~0319~~** | **~~Employee Roles~~** | | |
| --- | --- | --- | --- |
| ~~As a condo management administrator, I want a page where I am capable of setting the roles and managing my employees assigned to the same property that I may access through my dashboard, so that I can properly delegate the responsibilities of the property thereby allowing coherent property management and upkeep.~~ | | | |
| ~~Communication with Occupants\Employee Roles\~~ | | | |
| ~~M~~ | ~~M~~ | ~~S~~ | ~~2~~ |

**~~0319~~** ~~Acceptance Criteria~~

1. ~~The properties dashboard must be navigable from the menu.~~
2. ~~Must be accessible only through the company administrator’s account.~~
3. ~~Each property must be interactable, and open a new page when clicked.~~
4. ~~The page for each individual property, accessed through the dashboard, should have a section for employee roles.~~
5. ~~A list of employees should be displayed.~~
6. ~~Employees can be toggled to change their employee roles.~~
7. ~~Changes in roles should be reflected in an employee database.~~

## 

| **~~0620~~** | **~~Employee Notifications~~** | | |
| --- | --- | --- | --- |
| ~~As an employee of one of the condominiums, I want an interface in which I may interact and view requests received by the building management or the company administrator, so that I may know what duties and responsibilities I must fulfill as an employee.~~ | | | |
| ~~Communication with Occupants\Employee Notifications\~~ | | | |
| ~~S~~ | ~~M~~ | ~~M~~ | ~~5~~ |

## 

**~~0620~~** ~~Acceptance Criteria~~

1. ~~Requests must be stored in a database.~~
2. ~~Only employees can access.~~

| **~~0621~~** | **~~Employee Request Completion~~** | | |
| --- | --- | --- | --- |
| ~~As an employee of one of the condominiums, I want to mark requests displayed on the employees notification page as completed, so that I may communicate that I have fulfilled the request.~~ | | | |
| ~~Communication with Occupants\Employee Request Completion\~~ | | | |
| ~~S~~ | ~~M~~ | ~~L~~ | ~~2~~ |

**~~0621~~** ~~Acceptance Criteria~~

1. ~~Requests must be stored in a database.~~
2. ~~Only employees can access.~~
3. ~~Marked requests are deleted from the database.~~

### **2.3.6 Finance**

| **~~0422~~** | **~~Generate Annual Finance Report~~** | | |
| --- | --- | --- | --- |
| ~~As a condo management administrator, I want to generate an annual report of all condo fees collected in a given year through interacting with a digital element on the financial page, so that I may pay the necessary cost for the upkeep of shared areas, services and management.~~ | | | |
| ~~Financial Profile\Generate Annual Finance Report\~~ | | | |
| ~~M~~ | ~~XL~~ | ~~H~~ | ~~8~~ |

**~~0422~~** ~~Acceptance Criteria~~

1. ~~Only a condo management company user can generate an annual report.~~
2. ~~The report should show the collected fee of each property as well as the total collected fees of all properties.~~

| **~~0423~~** | **~~Setting Condo Unit Costs~~** | | |
| --- | --- | --- | --- |
| ~~As a condo management administrator, I want to enter the cost of a condominium unit being sold per square foot. Furthermore, I want to enter an added fee per parking spot requested in the purchase of a unit, so that I may place the unit on the market, determine property insurance or factor into the mortgage.~~ | | | |
| ~~Financial Profile\Setting Condo Unit Costs\~~ | | | |
| ~~M~~ | ~~XL~~ | ~~M~~ | ~~8~~ |

**~~0423~~** ~~Acceptance Criteria~~

1. ~~Only a condo management company user can enter these values.~~
2. ~~The condo fee per square foot and per parking spot must be clearly visible to all public users.~~

| **~~0424~~** | **~~Calculate Condo Fees~~** | | |
| --- | --- | --- | --- |
| ~~As a condo owner, I want the condo fee of each of my units to be calculated and presented in a figure on my properties dashboard, so that I may observe my housing costs and property transactions.~~ | | | |
| ~~Financial Profile\Calculate Condo Fees\~~ | | | |
| ~~M~~ | ~~XL~~ | ~~M~~ | ~~5~~ |

**~~0424~~** ~~Acceptance Criteria~~

1. ~~The calculated condo fees must be accessible to unit owners at all times on their properties dashboard.~~

| **~~0425~~** | **~~Calculate Operation Costs~~** | | |
| --- | --- | --- | --- |
| ~~As a condo management company user,~~  ~~I want to be able to view my condo building’s operational budget and cost,~~  ~~So that I can enter the cost for each operation (condo fees) and display them to unit owners.~~ | | | |
| ~~Financial Profile\Calculate Operation Costs\~~ | | | |
| ~~M~~ | ~~XL~~ | ~~H~~ | ~~5~~ |

**~~0425~~** ~~Acceptance Criteria~~

1. ~~Only condo management company users can view the condo’s operational budget and cost, as well as enter the cost for each operation.~~
2. ~~Condo management company users must be able to provide a description of the operation costs.~~
3. ~~Unit owners should be notified when their property is flagged with an operation cost and be able to view the fees and their description.~~

### **2.3.7 Additional Features**

| **0026** | **App On Multiple Digital Platforms** | | |
| --- | --- | --- | --- |
| As a user, I want the condo management system to be accessible on multiple digital platforms, including Android, iOS, Linux, MacOS, and Windows, so that it may be available on any device of choice. | | | |
| Additional Features\App On Multiple Digital Platform\ | | | |
| C | XL | H | 8 |

**0026** Acceptance Criteria

1. The application can be opened and run successfully on Android, iOS, Linux, MacOS, and Windows.

| **0027** | **Language Diversity** | | |
| --- | --- | --- | --- |
| As a public user, I want the application to be available in not only English but French as well, so that it may be more accessible to me. | | | |
| Additional Features\Language Diversity\ | | | |
| C | L | L | 3 |

**0027** Acceptance Criteria

1. The entire system should be translated into French properly.

| **0028** | **Single Sign-On** | | |
| --- | --- | --- | --- |
| As a user, I want to login using my Gmail account, so that I can easily create an account without additional login credentials. | | | |
| Additional Features\Single Sign-on\ | | | |
| C | XL | L | 3 |

**0028** Acceptance Criteria

1. Single Sign On with Gmail, Outlook or other services must successfully log the user into their correct account.

# **3 Software Architecture Description**

## 3.1 Introduction

### 3.1.1 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.

### 3.1.2 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.

### 3.1.3 Other information

#### 3.1.3.1 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.

## 3.2 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

### 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*

### 3.2.2 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*

### 3.2.3 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*

## 3.3 Viewpoints

### 3.3.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**

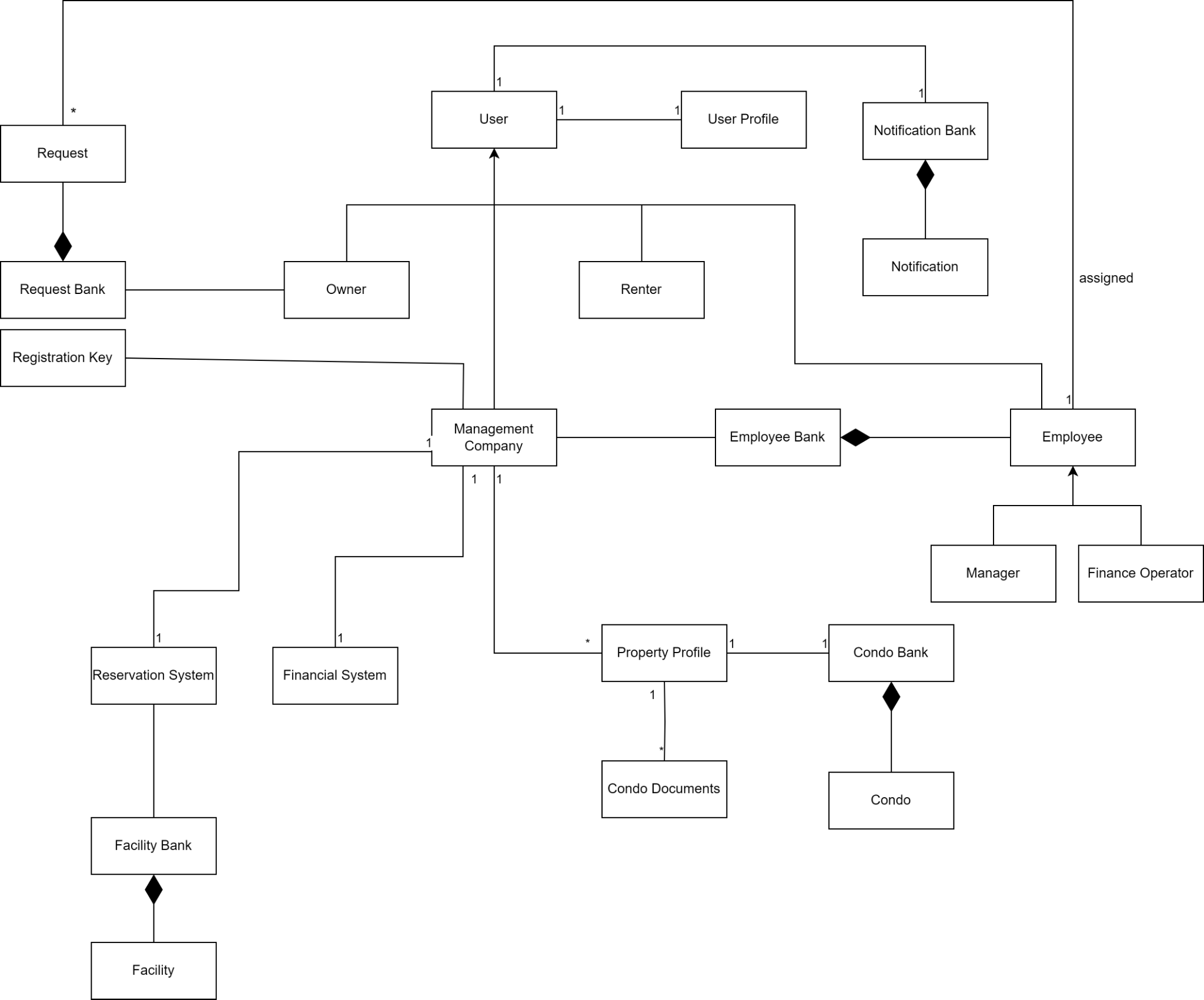
* Entities are represented as nouns within circles or rectangles.
* Attributes of entities are depicted as text within or adjacent to the entity representation.
* Relationships between entities are depicted as lines interconnecting with one another.
* Entities may have associated attributes that describe 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.

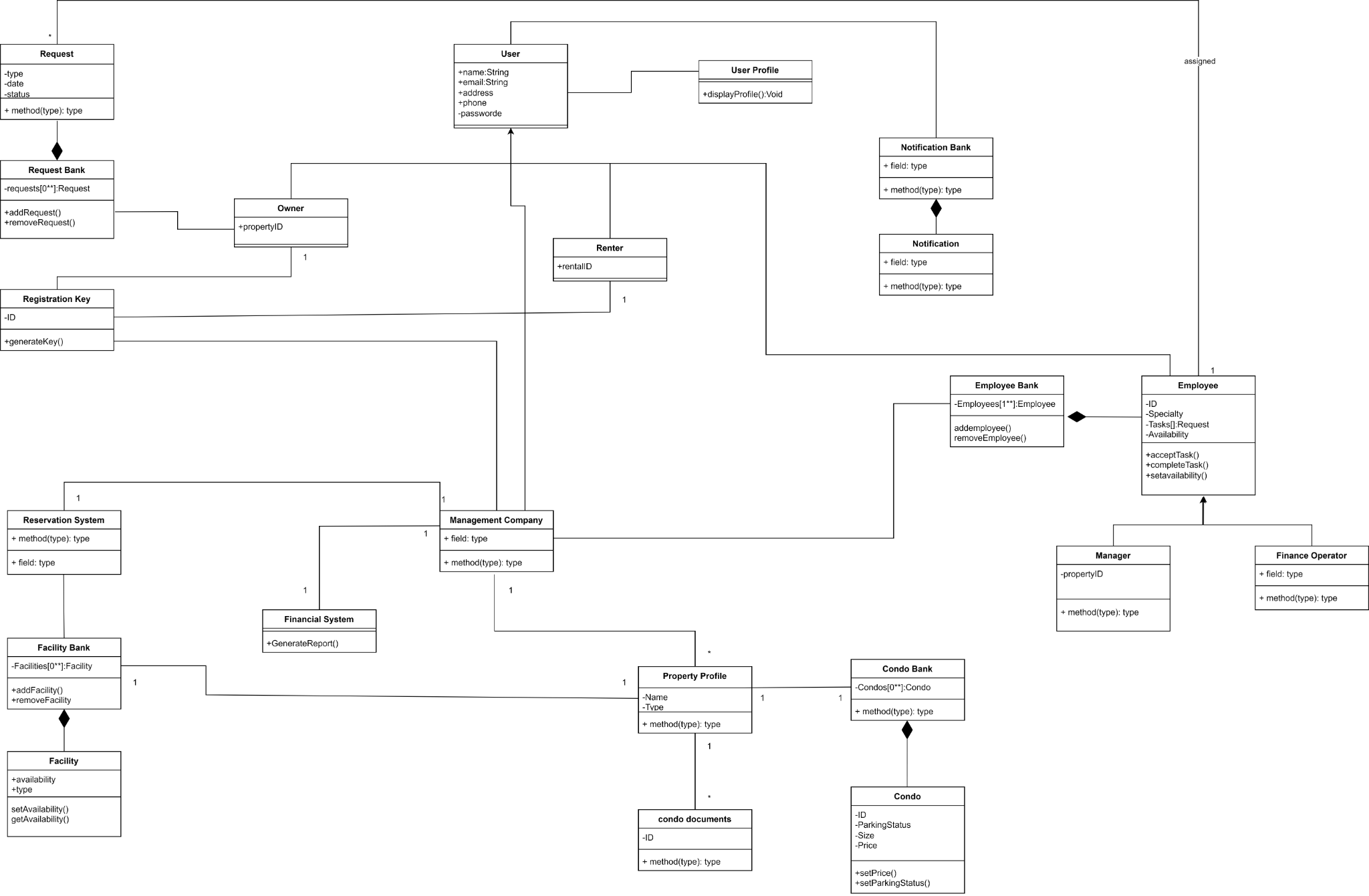
**Conventions and Correspondence Rules**

* 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.

#### 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.

#### Component Diagram

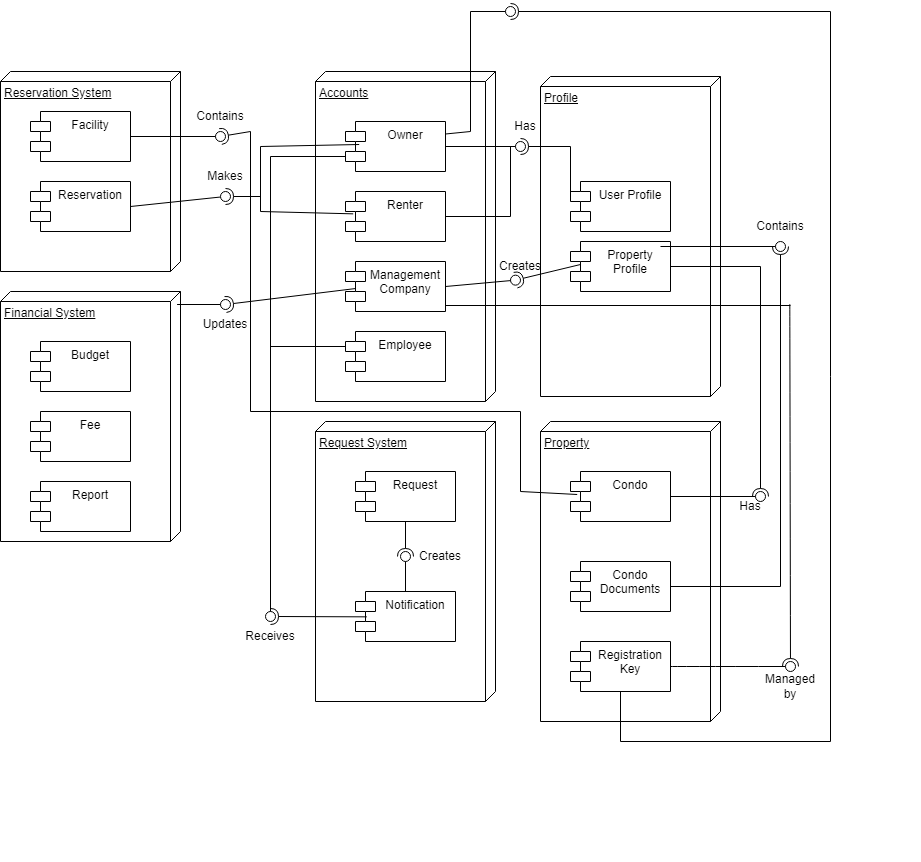


Figure 2: Component Diagram

### 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.

#### 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.

#### Sequence Diagram

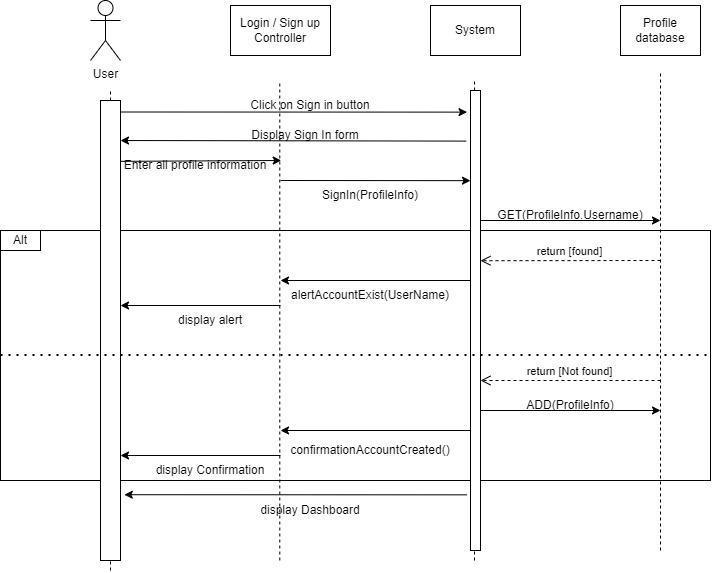


Figure 3: Sequence System Diagram for Registration of new Accounts

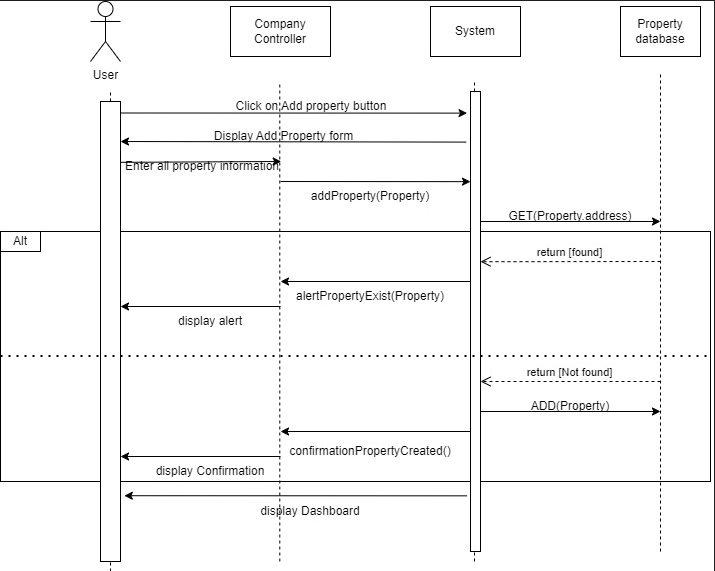


Figure 4: Sequence System Diagram for Adding New Property

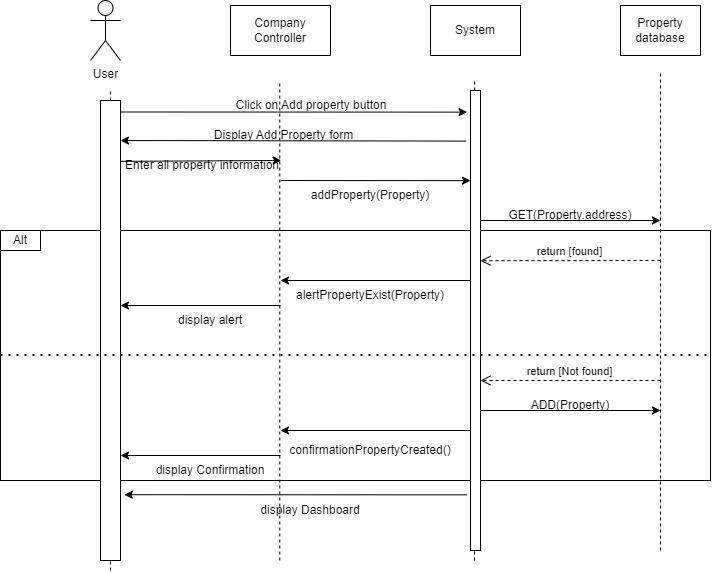


Figure 5: Chat Communication Sequence Diagram

#### Activity Diagram

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

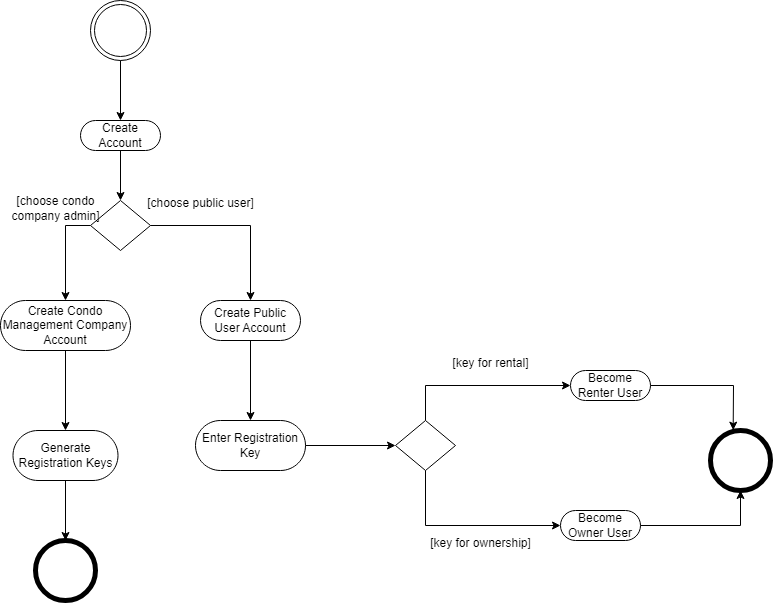


Figure 6: Create an Account Activity Diagram

### 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.

#### 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 us 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.

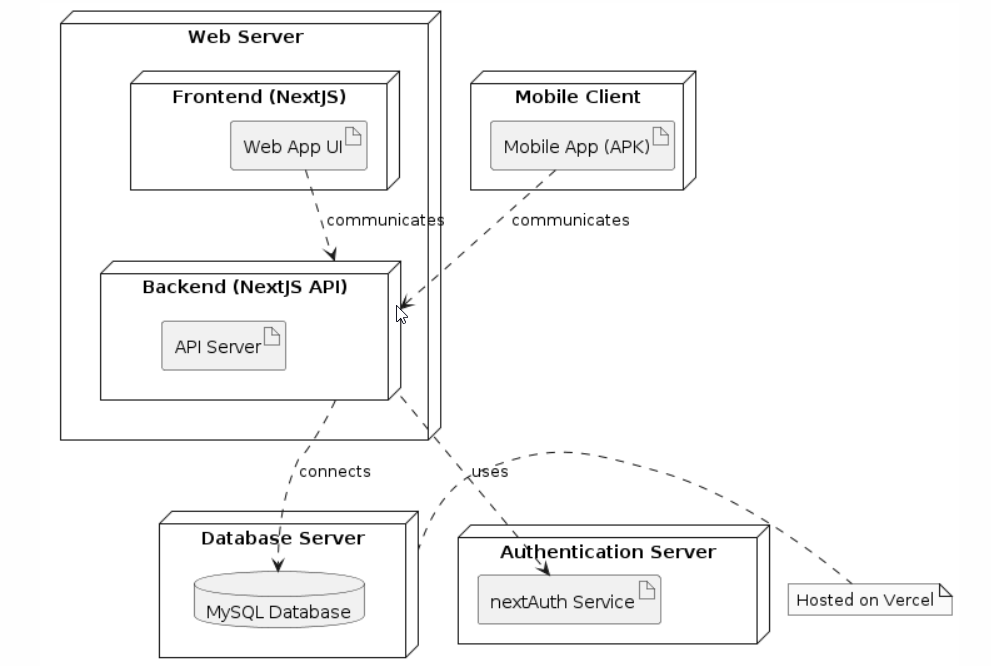
#### Model Kinds

A deployment diagram was chosen to represent the physical viewpoint

**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.

#### 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.

#### 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.

#### 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.

#### Use Case Diagrams

**AnaCondo System Use Case Diagram**

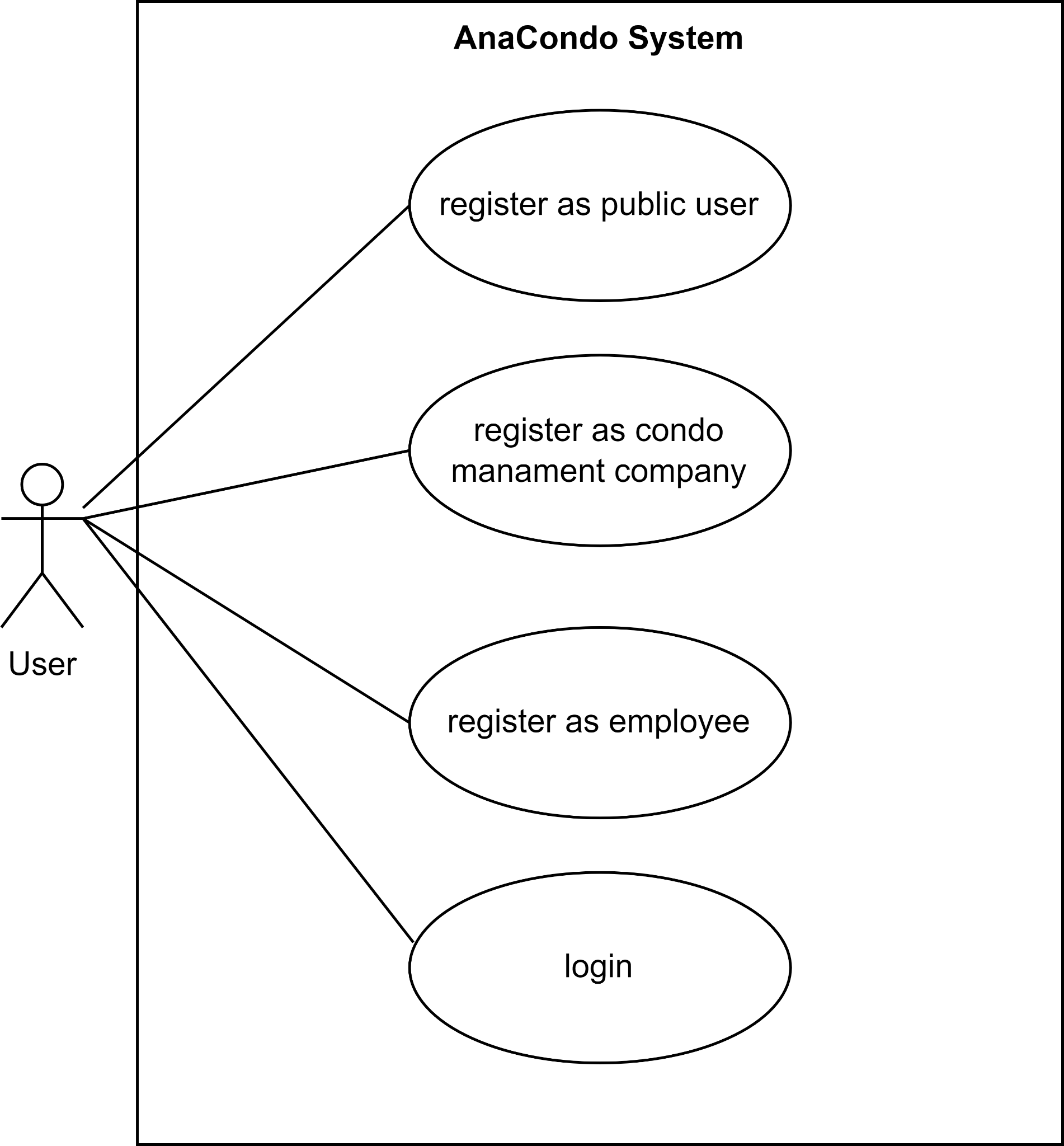


Figure 7: Use Case Public User System diagram

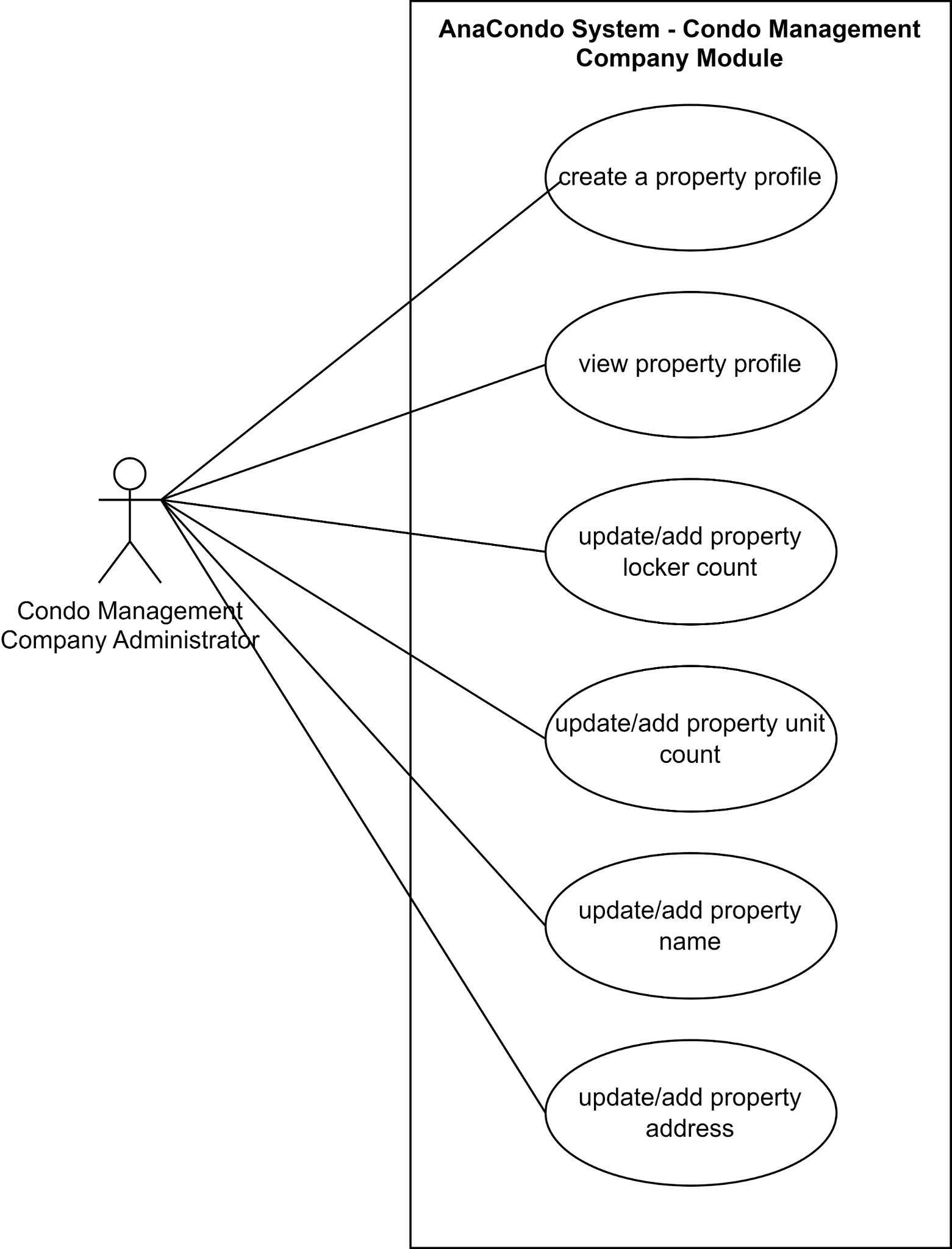
**UC1** Register as Public User: User can sign up as a public user using their google account or by filling a form

**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

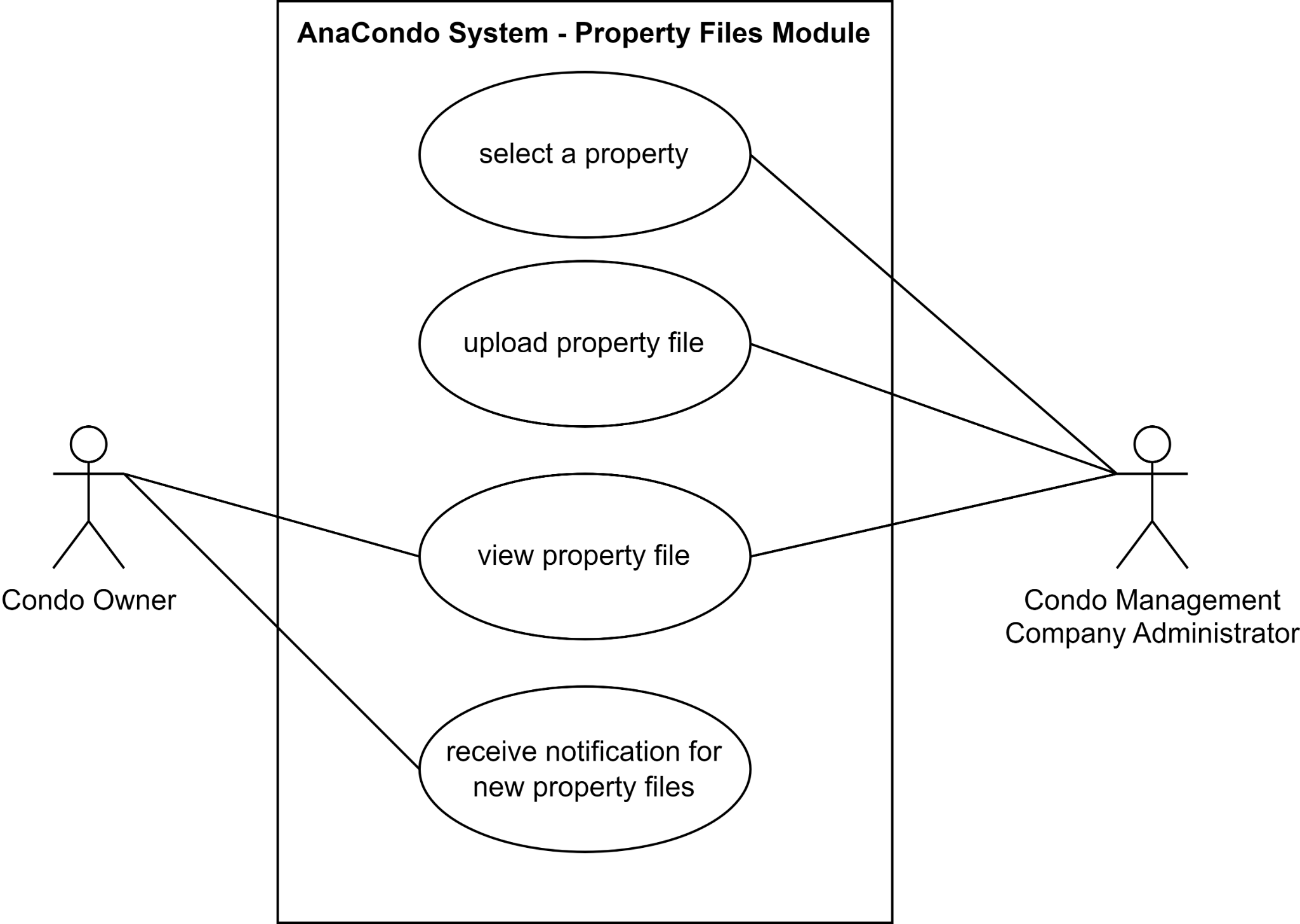
**UC3** Update/Add Property Locker Count: Condo Management Company Administrator can add or modify the locker count of a property

**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**



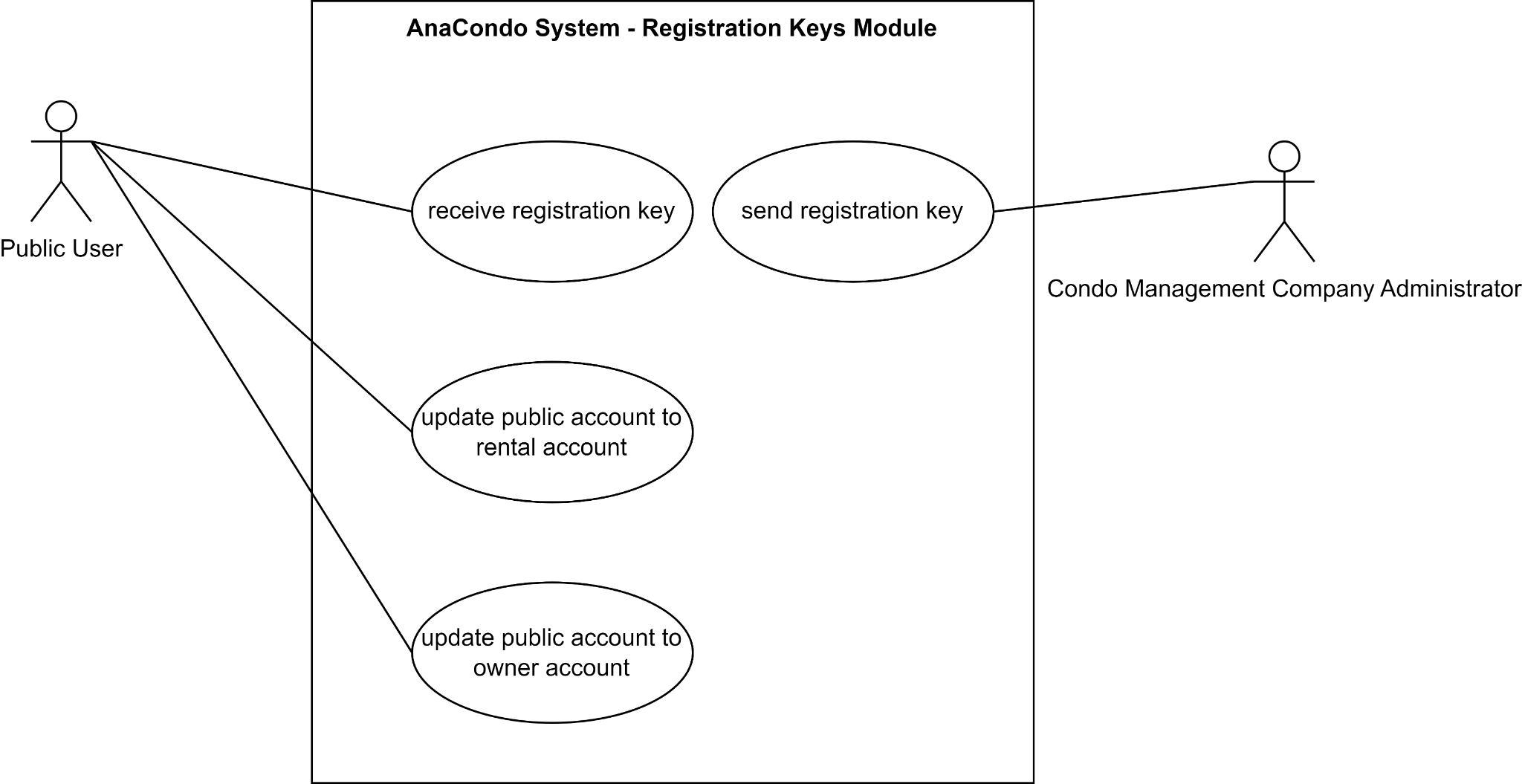
***Figure 9****: Property Files Module Use Case diagram*

**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**

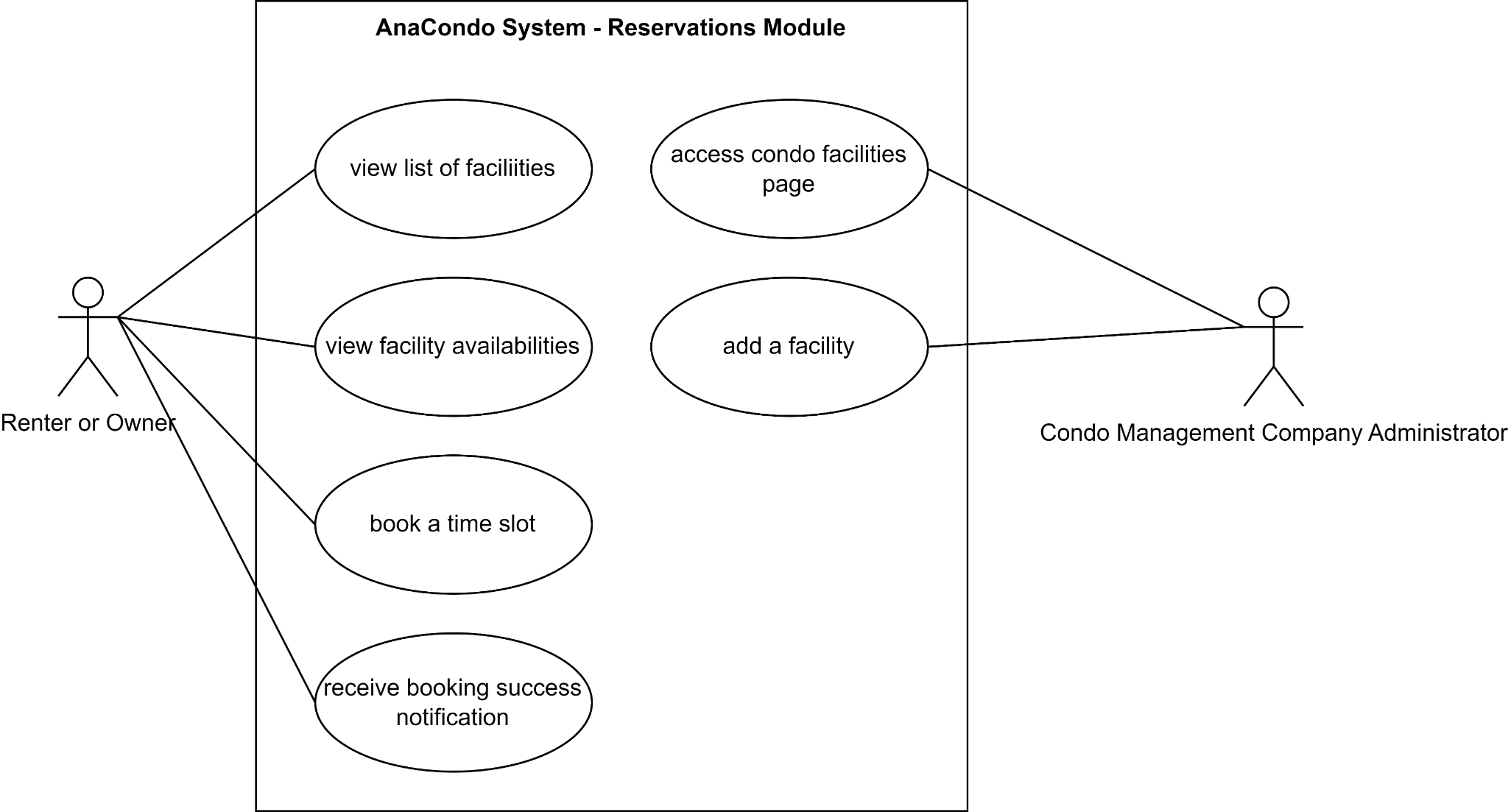


Figure 11: Reservation Use Case diagrams

**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

## 3.4 Consistency and correspondences

### 3.4.1 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. |

### 3.4.2 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 |

## 3.5 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).

# **4 UI Prototypes**

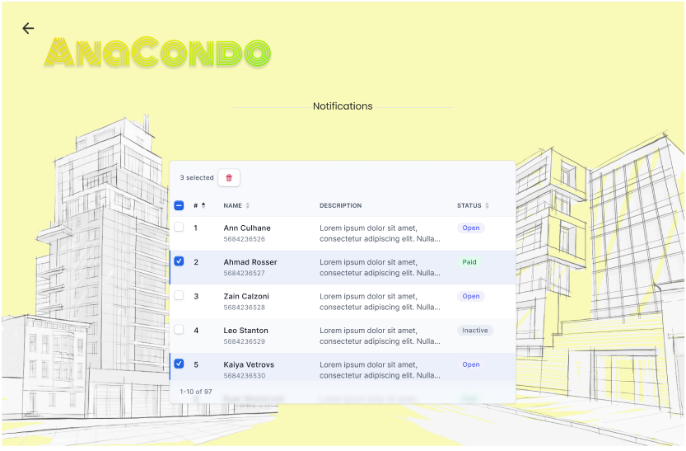
## 4.1 Prototypes for Website

The following section details and illustrates the planned user interface for Sprint 4’s user stories.

**[US-0621]**: [Employee Request Completion]

As an employee of one of the condominiums, I want to mark requests displayed on the employee’s notification page as completed, so that I may communicate that I have fulfilled the request.

A screenshot of a computer screen

Description automatically generated****

Employee User Page Employee Notifications Page

Steps:

1. The Employee user selects “Notifications”
2. The Employee user changes the status of their requests
3. The user removes the completed requests

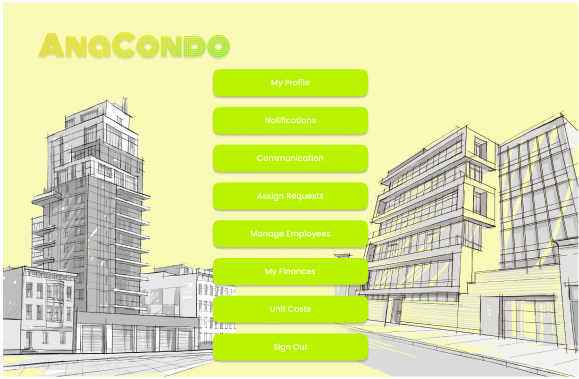
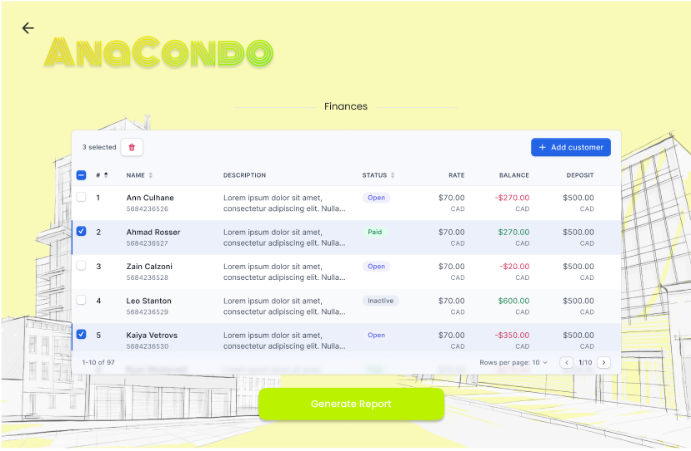
**[US-0422 && US-0425]**: [Generate Annual Finance Report && Calculate Operation Costs]

As a condo management administrator, I want to generate an annual report of all condo fees collected in a given year through interacting with a digital element on the financial page, so that I may pay the necessary cost for the upkeep of shared areas, services and management.

As a condo management company user,

I want to be able to view my condo building’s operational budget and cost,

So that I can enter the cost for each operation (condo fees) and display them to unit owners.

****

Condo Management Administrator Page Finances Page

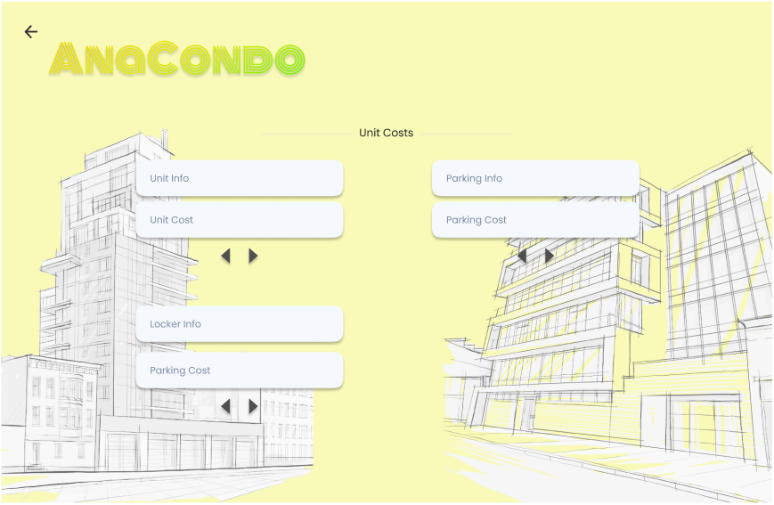
Steps:

1. The Condo Management Administrator user selects “My Finances”
2. The Condo Management Administrator user enters the cost for each operations
3. The Condo Management Administrator user selects “Generate Report” which creates a report with all operations displayed on the Finances page

**[US-0423]**: [Setting Condo Unit Costs]

As a condo management administrator, I want to enter the cost of a condominium unit being sold per square foot. Furthermore, I want to enter an added fee per parking spot requested in the purchase of a unit, so that I may place the unit on the market, determine property insurance or factor into the mortgage.

A screenshot of a computer screen

Description automatically generated

Condo Management Administrator User Page Unit Costs Page

Steps:

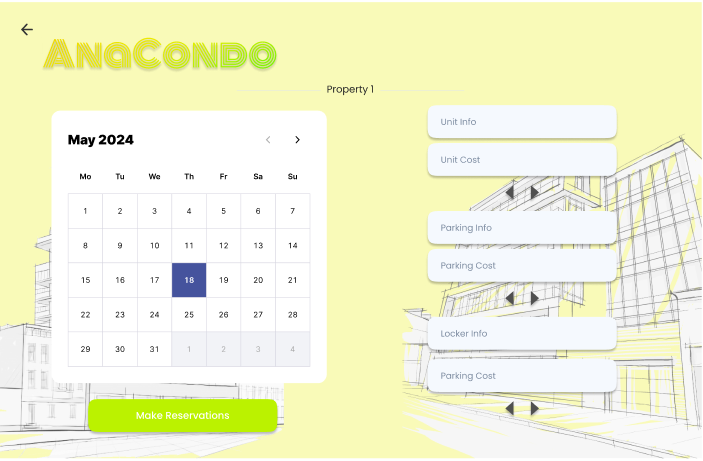
1. The Condo Management Administrator user selects “Unit Costs”
2. The Condo Management Administrator enters fees for specifics utilities of the condo

**[US-0424]**: [Calculate Condo Fees]

As a resident, I want the condo fee of each of my units to be calculated and presented in a figure on my properties dashboard, so that I may observe my housing costs and property transactions. A screenshot of a website

Description automatically generated

Resident User Page Property List Page



Property Page

Steps:

1. The Resident user selects “Reservations”
2. The Resident user selects the property they wish to view information for
3. The Resident user views the fees of the different units they reserved

## 4.2 Prototypes for App

These are the UI prototypes for the phone, most of these will be reused for many pages

Log In Tables to Grid Conversion

A screenshot of a login screen

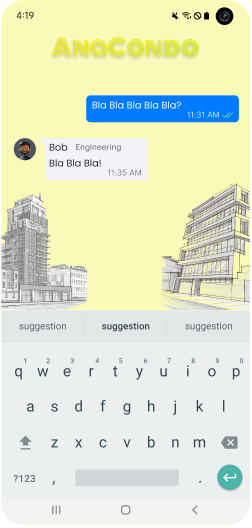
Description automatically generatedA screenshot of a phone

Description automatically generated

Unit Selection Calendar ReservationA screenshot of a phone

Description automatically generated

Message Board



# 5 Risk Assessment and Management Plan

## 5.1 Purpose of the Risk Management plan

The objective of the Risk Management plan is to outline a strategy for identifying, assessing, and prioritizing potential risks that could affect a software development project. This document aims to assist project managers and team members in foreseeing, preventing, or mitigating any adverse impacts these risks may have on the project's schedule, budget, and overall success. Having a RMP enhances project planning and decision-making. In addition, it provides project visibility and tracking of issues. Finally, it decreases project costs through proactive risk mitigation

## 5.2 Identification of risks and assessment

Implementation during Sprint 1 involved the team effectively identifying and assessing potential risks through a combination of brainstorming sessions and a checklist-based approach. In these collaborative sessions, team members discussed and identified potential risks based on their individual knowledge and past experiences. These methods enabled the team to comprehensively identify and document potential risks that could impact the project during Sprint 1.

**Risk charts**

| Impact  Probability | Low | Medium | High |
| --- | --- | --- | --- |
| Low | 6, 21 | 3, 15, 20 | 12, 13, 19 |
| Medium | 25, 27, 28 | 1, 4, 5, 9, 14, 16 | 7, 8, 10, 13, 18 |
| High |  | 11, 17, 22 | 2, 10, 24, 26 |

**Table 1:** Risk management chart

## 5.3 Criteria for Defining Impact and Probability Levels

It is essential to establish precise criteria for the effect and likelihood levels of hazards that have been discovered in order to guarantee a consistent and impartial approach to risk assessment. With the help of these criteria, the project team may precisely assess and classify each risk, enabling the use of efficient risk management techniques.

**Impact Levels**

* High Impact:

Risks that, if they came to pass, would seriously disrupt the project and might result in large delays, large cost overruns, or a serious compromise of the project's goals. These hazards may need a thorough review of the project's timelines, money, and scope.

* Medium Impact:

Hazards that could cause a minor amount of disruption, which could lead to delays or extra expenses that are controllable within the project's backup plans. Although these risks might need alterations to the project plan, they do not pose a threat to the project's overall success.

* Low Impact:

Risks whose effects would be easily mitigated by the project's current contingencies and whose influence would be negligible. These risks don't have a major impact on the project's overall goals or deliverables, but they could lead to tiny delays or slight cost increases.

**Probability Levels**

* High Probability:

Hazards that, given the state of the project, past performance, or professional opinion, are highly likely to materialize. In order to reduce the possible effects of these hazards, timely attention and planning are needed.

* Medium Probability:

Risks with a reasonable chance of occurring. While not certain, these risks are plausible enough to warrant consideration and preparation in the project's risk management strategy.

* Low Probability:

Risks that are unlikely to occur, given the current understanding of the project environment and external factors. These risks are monitored but are considered low priority for immediate action.

## 5.4 Decision-Making Criteria

The determination of impact and probability levels for each identified risk involves a combination of quantitative analysis and qualitative judgment. The project team employs a systematic approach, incorporating the following:

* Historical Data: Analysis of similar projects or past phases within the current project to identify trends and outcomes related to specific risks.
* Expert Judgment: Input from team members, stakeholders, and external experts, drawing on their experience and understanding of the project's context.
* Project Specifics: Consideration of the unique aspects of the current project, including scope, resources, and external factors, that might influence the likelihood and impact of risks.

| **Risk ID** | **Risk Name** | **Risk Type & Description** | **Risk Score** | **Resolved in Sprint** | **Strategy & Effectiveness** |
| --- | --- | --- | --- | --- | --- |
| 1 | Change of Requirements | Requirements-related change may occur during the project lifecycle, impacting deliverables and timelines. | Medium | Sprint 1, 2, 3, 4 | Accept |
| 2 | Unclear Requirements | Lack of clarity in project requirements leading to misunderstanding and rework. | High | Sprint 1, 2, 3, 4 | Avoid |
| 3 | Mobile & Web compatibility | Compatibility issues between mobile and web platforms leading to usability and functionality issues. | Low | Sprint 2, 3, 4, 5 | Mitigate |
| 4 | Resource Restrictions | Insufficient resources (e.g., budget, team members) causing delays and quality compromises. | Medium | Sprint 1, 2, 3, 4 | Accept |
| 5 | Rapid Technology Changes | Rapid changes in technology trends impacting project delivery | Medium | Sprint 1, 2, 3, 4 | Accept |
| 6 | Change in the Leadership Role | Change in project leadership leading to disruption in decision-making and project direction. | Low | Sprint 1, 2, 3 | Accept |
| 7 | Lack of Proper Planning | Inadequate planning leading to missed deadlines, scope creep, and budget overruns. | High | Sprint 1 | Mitigate |
| 8 | Poor Documentation | Incomplete or inaccurate documentation leading to misunderstandings and rework. | High | Sprint 1 | Mitigate |
| 9 | Insufficient Training | Lack of training for team members leading to inefficiencies and errors. | Medium | Sprint 1 | Mitigate |
| 10 | Lack of Proper Testing | Inadequate testing leading to undetected defects and poor software quality. | High | Sprint 2, 3, 4 | Mitigate |
| 11 | Miscommunication & No Communication | Poor communication leading to misunderstandings, conflicts, and delays. | High | Sprint 1, 2, 3, 4, 5 | Mitigate |
| 12 | Lack of Familiarity with Technology | Limited knowledge and experience with technology stack leading to inefficiencies and errors. | Medium | Sprint 1 | Mitigate |
| 13 | Poor Risk Management | Ineffective identification, assessment, and mitigation of risks leading to project failures. | High | Sprint 1 | Mitigate |
| 14 | System Performance | System performance issues leading to user dissatisfaction and decreased productivity. | Low | Sprint 1, 2, 3, 4, 5 | Mitigate |
| 15 | Vulnerable Security | Vulnerabilities in system security leading to data breaches and loss of trust. | Medium | Sprint 4 | Transfer |
| 16 | External Dependencies | Dependencies on external factors leading to delays and project bottlenecks. | Medium | Sprint 1, 2, 3, 4, 5 | Accept |
| 17 | Escalation of Complexity | Project complexity leading to difficulties in understanding, implementation, and maintenance. | High | Sprint 1, 2, 3, 4, 5 | Avoid |
| 18 | Code Scalability | Inability of the system to handle increased workload leading to performance degradation. | High | Sprint 2, 3 | Transfer |
| 19 | Regulatory Law Compliance & Actions | Failure to comply with regulatory laws leading to legal actions and penalties. | Medium | Sprint 4 | Avoid |
| 20 | Poor Data from User Feedback | Receiving unreliable user feedback leading to unnecessary or useless feature adoptions. | Low | Sprint 2, 3, 4 | Mitigate |
| 21 | External and Unpredictable events | External events such as outages, internet service provider issues that can delay the progress of the project | Low | Sprint 2, 3, 4, 5 | Accept |
| 22 | Inadequate encryption key management | Failure to implement robust key management practices can lead to risks by using weak keys, failing to rotate keys regularly, storing keys insecurely, and inadequate protection measures for keys in memory. | High | Sprint 4 | Mitigate |
| 23 | Cyber-attacks | Cyber attacks, such as malware, ransomware, can lead to unauthorized access to the system, data breaches, loss of sensitive information, and significant disruption to operations. | High | Sprint 2, 3, 4, 5 | Mitigate |
| 24 | Website Deployment Errors | Publishing a website on a server or hosting platform runs into issues. | High | Sprint 5 | Mitigate |
| 25 | Responsiveness | Making sure that UI adapts to various devices and screen sizes, which adds a level of complexity. | Low | Sprint 5 | Mitigate |
| 26 | Back-end integration | Although many of the components were completed in the previous sprint, many of them were not yet integrated with the backend. | High | Sprint 5 | Mitigate |
| 27 | Multiple Logins | Presence of multiple types of login users: administrator, condo owner, condo company, and employee. This adds a layer of complexity | Medium | Sprint 5 | Accept |
| 28 | Code Management | Presence of multiple pages for the UI of the website | Medium | Sprint 5 | Accept |

**Table 2:** Risk management chart

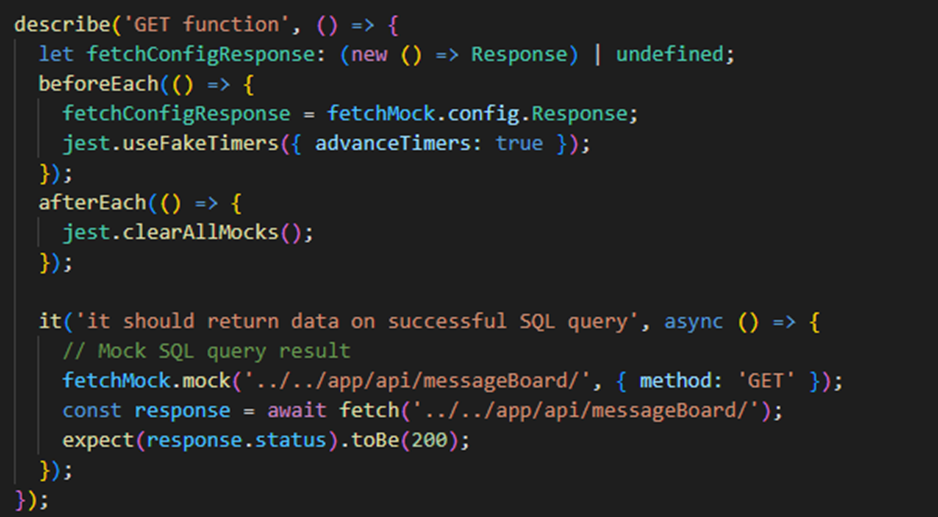
# 6 Testing Report and Testing Plan for Sprint 5

## 6.1 Unit Testing

An unanticipated issue hindered our progress in the third sprint: the unit testing files were corrupted, necessitating a thorough reimplementation. This difficulty led us to implement Node.js, which aligns with the core goal of unit testing: independently confirming that distinct code segments function as intended. Unit testing is focused on testing individual functions or components once they are created, focusing on the smallest units of an application that can be examined. A unit test typically processes inputs to provide a single, meaningful result.

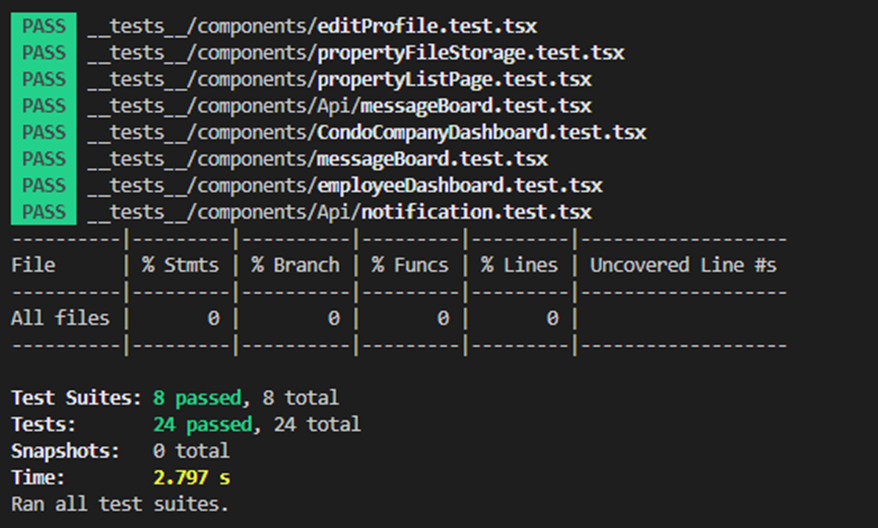
We kept using Jest as our testing library to keep our testing methodology consistent and to make use of well-known technologies. In order to ensure a full and comprehensive testing process, we improved our techniques to test the API rigorously and used several methodologies to evaluate the rendered elements of the sites. Jest's strength is its wide range of compatibility with various web development tools and its unique focus on making testing complex online applications easier. This framework is our first pick because it works well with Angular, Babel, TypeScript, Node.js, React, Vue.js, Svelte, etc.

Due to several obstacles, such as a great deal of faults that occurred during the deployment and testing stages, we have opted to postpone the integration of Continuous Integration (CI) procedures until Sprint 5. The reason for this calculated delay is the significant time constraints we encountered, which required us to reallocate our resources in order to prioritize and take care of the urgent testing issues. This timetable modification demonstrates our dedication to precision and high standards in our deployment procedure.

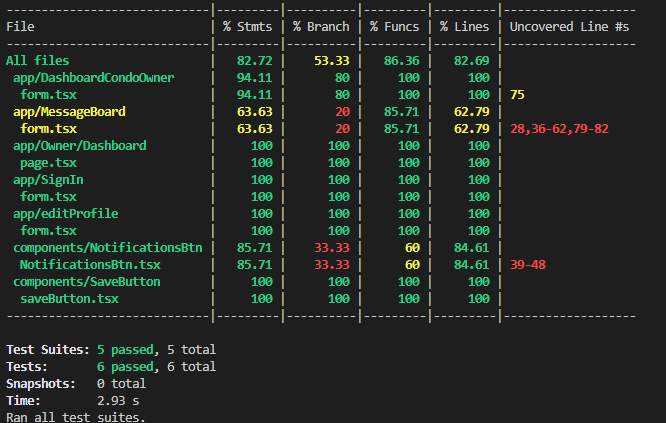


***Figure 1****: Example of our Unit Testing using Jest to test one of our API call*

## 6.2 Code Coverage



***Sprint 3****: test coverage report*



***Sprint 4****: test coverage report*

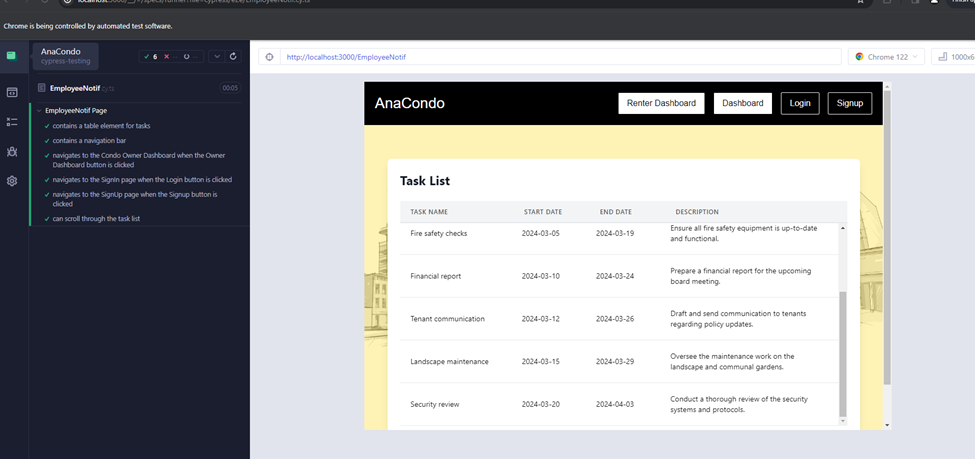
A strong and comprehensive testing strategy is shown by the report's overall high percentage of coverage, with many files reaching 100% across a range of parameters. Interestingly, though, some parts of several files—like components/NotificationBtn and app/messageBoard—show less coverage in some measures. This implies that there may be room in those files to add more test cases for certain scenarios, function calls, or lines. There is space for improvement in the components/SaveBtn file, particularly in branch coverage, which suggests that not all conditional branches have undergone thorough testing.

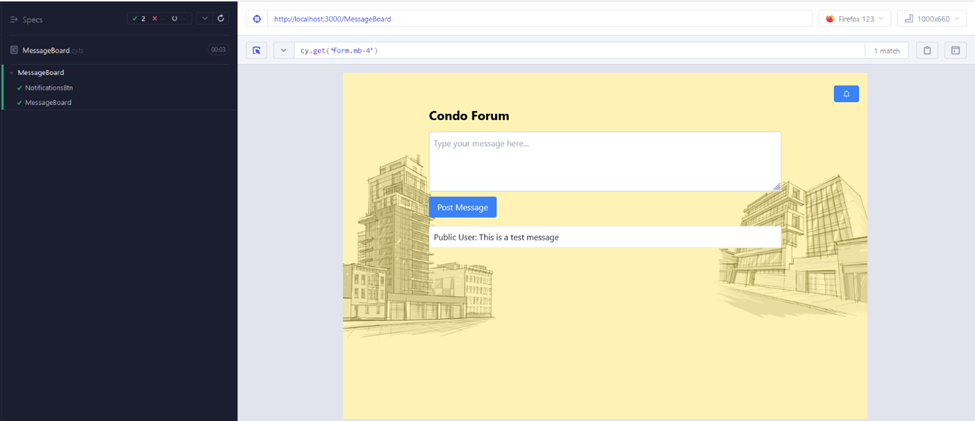
The summary line indicates that all six test suites—that is, all six test runs—have succeeded, indicating a flawless test run devoid of any errors. All of the tests took only 2.93 seconds to complete, which shows that developers may get feedback rather quickly.

In conclusion, the report demonstrates a strict testing methodology, but it also identifies certain areas where the test suite might be improved to obtain even more thorough coverage. Regressions can be prevented by new changes, and good code quality is maintained with the help of such comprehensive coverage measurements.

## 6.3 System Testing

For the system testing, we still implemented Cypress. The goal is to provide a realistic experience that follows the steps of a real-world user scenario. To fully validate the system, testing should be done on both the system that is being tested and any related subsystems. System testing was delayed for sprint 5 due to the pages not being linked together, and the only way to access each page was through the URL.





# **7 Sprint** 4 **Retrospective**

## **7.1 Introduction**

In Sprint 4, the team focused on implementing the feedback from our TA during the demo of Sprint 3. We mainly put emphasis on the coding aspect of the application; we had to complete or achieve close to completion all of our user stories from the release plan. This indicates that almost every feature from the website has been completed and every feature is intended to be the release version of the website. In addition, a lot more tests in the backend as well as the frontend were completed. Currently, we have 6 people working on the backend and 4 people working on the frontend and everyone tests on their end. However, we have yet to work on the mobile development side of the application.

## 7.2 What went right

### **1 - Open Communication**

Our team was open to talk the whole sprint, despite our lack of accuracy in forecasting the completion dates of many criteria. Members may use our team messaging channel to send messages for help or clarification, and they could expect a response, helpful or not, to their needs. This is a crucial component of a large group project because it gives students the chance to boost their confidence and sense of self, which are essential for efficiency and performance.

### **2 - Team Management**

Our group successfully divided up our deliverables in a just and equitable way. We were able to engage and communicate as needed because of this. Effective team management is necessary for high performance, enhanced output and efficiency, and great flexibility. Improved communication among team members may facilitate future sprint planning and meeting organization.

### **3 - Coordination and Organisation**

We were able to define the deliverables' deadlines thanks to the Agile technique that we employed. It did not take long for us to realize that setting internal deadlines was essential to finishing our sprint on time. In fact, we noticed that decisions had to be made quickly because certain deliveries depended on others. Furthermore, our team did a fantastic job assigning responsibilities and structuring the information. It appears that a single interface included all of the data needed for us to do our duties. Our productivity increased as we were clear about where to get the necessary paperwork for particular tasks. These aspects couldn't have been better because they closely resembled the planning and execution of real-world efforts.

### **4 - Adaptability and Problem Solving**

The group showed an amazing capacity to quickly come up with answers to unforeseen issues and adjust to new circumstances. Because of its flexibility, the project managed to stay on course even with the changing nature of the development process. The team's resilience and collaborative spirit were demonstrated by their proactive approach to problem-solving, which reduced potential delays and maximized output.

### **5 - Collaboration Between Members**

Pair programming turned out to be a very helpful practice. By having two team members collaborate on difficult code, we were able to enhance the software's quality and promote knowledge sharing amongst more and less experienced developers. Fewer defects, a common grasp of the software, and quicker learning for more recent hires were the results of this cooperation. The group's cooperation was excellent. Collaboratively, members from disparate functional domains tackled the intricacies involved in merging the front and back ends. The project's momentum was maintained by this synergy, which not only promoted a strong sense of team togetherness but also made sure that cross-functional difficulties were handled quickly and successfully.

## 7.3 What went wrong

### **1 - Time Management**

Ideally we would have had a meeting and split the work up for the sprint at the very beginning of the sprint, but it took about a week after the sprint start date before we distributed the work. This caused us to have less time to do the actual work. We made a coding deadline for ourselves, however at least half of the implementation which had been intended to be completed was still not done by the deadline we had made for ourselves.

### **2 - Lack of Cohesion**

Since we did not design and implement the database tables all together, our team ended up making the tables progressively, and each person created tables that were necessary for their parts. This was alright in the first few sprints, however by sprint 4 we started encountering problems because of this approach. For example, there was a table in the database called Assignments which was made for employee requests, however, this table had foreign keys of type VARCHAR that actually needed to be of type INTEGER, and this caused there to be foreign constraint violations anytime someone tried to insert into the table. Therefore another table had to be made called Request for the employee requests which fixed the foreign key data types. Then it turned out that there was also another table which had been made called submittedRequests which was also intended to be for the employee requests. We had to pick exactly which table we were using but by then implementation had already been done based on the structure of the different tables so some changes had to be made to accommodate which table we decided on and it was a very unideal situation which could have been avoided if we had planned and created the database in the first or second sprint.

### **3 - Code Management**

Actually with code management we did a lot better than in the previous sprints. Our commit messages and Pull requests messages were standardized well. However we still encountered issues with using feature branches. This is because most of the team was using forks, and making the switch to feature branches would have added extra work and slowed down the team. The time to work on implementation was already so limited so using up more time to switch from forks to branches was not ideal. Also, some members of the team had already started implementing some of their parts by the time we knew we needed to switch to branches, and in addition to that, some of the setup was difficult to redo, and in order to avoid these issues some members of the team stuck with using their fork instead of feature branches.

### **4 - Completeness**

Some areas of our website lack a sense of completeness. For example our nav bar is still missing functionality. Not all our pages have frontend designs which complement one another, some pages have a vastly different aesthetic than other pages. Overall the appearance of our frontend could be improved, but is at a minimum sufficient.

### **5 - Conclusion**

The challenges we are facing will enable us to have more experience and a better idea of how to succeed in future projects. For example, designing and making all the database tables at once in a cohesive manner, prior to any extensive implementation. In the future we would also use feature branches and other code management strategies from the very beginning of a project. We are definitely learning a lot from this experience and growing our skills as a team, as well as individually.

# **8 Sprint 4 Release Plan**

## **8.1 Completed Tasks**

**Completed Tasks**

Tasks concerning documentation that are expected to be completed and updated include the software architecture, risk assessment and risk management plan, product vision statement, requirements and user stories, testing plan for sprint 4, sprint 3 retrospective, release plan, UI prototypes for sprint 4 documents and code management.

What has been completed thus far from sprint 4 is the frontend for all user stories and the backend for employee roles, requesting assignments, organizing events through the portal, employee request completion, discount offers, generating an annual finance report, setting condo unit costs, calculating condo fees and calculating operating costs. Progress has begun on implementing the app on multiple platforms. Unit testing, UI testing and API testing have also progressed using atomic commits on Github.

Specific tasks to be completed during Sprint 5 include the following user stories,

0026: Implementing app on multiple platforms,

0310: Send registration keys to Condo Owners and Rental Users,

with a general focus on finalizing the overall system, all user stories, ensuring the interconnected components of the system and system testing. Further, deployment of the final product on more than one device, creating the traceability matrix, overall test coverage, complete retrospective, compiling the final report and source code, self-assessment and team assessment and presentation and demo.

The table below illustrates the features backlog for Sprint 5.

| ***User Story Backlog Sprint (5)*** | | | |
| --- | --- | --- | --- |
| **User Story ID** | **User Story Points**  **(USP)** | **Priority** | **Status** |
| 0310 | 5 | High | IN PROGRESS |
| 0026 | 8 | High | IN PROGRESS |
| **Total USP** | 13 |  |  |

**Sideways Bar Graph**



[Updated Release Plan of Project 4.xlsx](https://docs.google.com/spreadsheets/d/1SyAPCv006PvXugymmrOF00eEBFwx6usH/edit?usp=sharing&ouid=102107386882227747444&rtpof=true&sd=true)

# **9 Code Management**

## 9.1 Software Implementation of Planned User Stories

To effectively tackle our planned user stories, we broke down their implementation into three phases. The first phase involved conceptualizing and documenting the user stories. We used Github issues as well as Word documents to detail our user stories, including their difficulty, their descriptions, and their acceptance criteria. Then, we developed the user stories and split the work into frontend and backend. The work was assigned to team members who then took on the task of developing their part of the assigned user story. Finally, we ran unit tests on the implemented code to ensure it met the acceptance criteria and to ensure the code could handle different input.

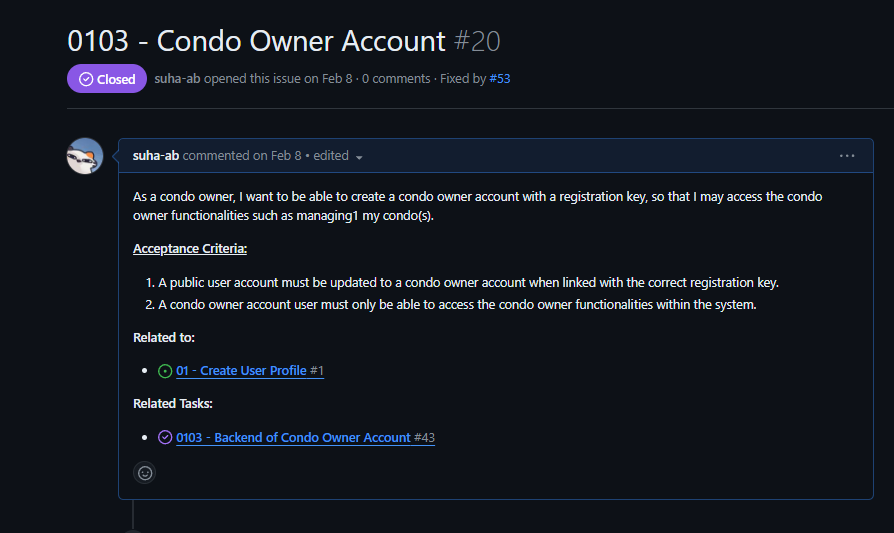


Image 1: Phase 1 of Implementing User Stories



Image 2: Phase 2 of Implementing User Stories

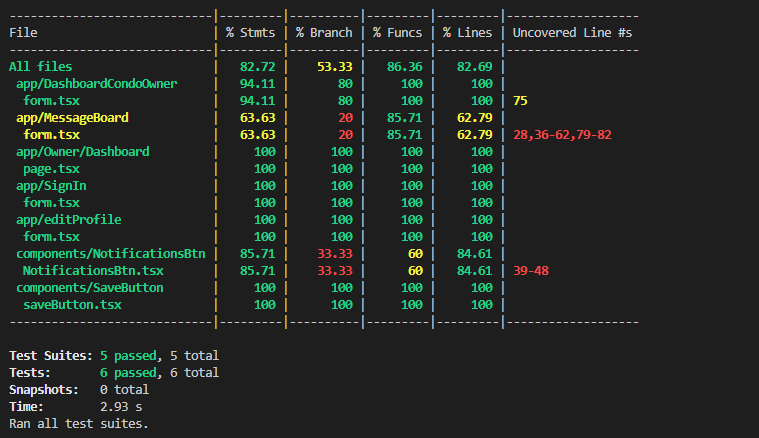


Image 3: Phase 3 of Implementing User Stories

## 9.2 Bug Reports and Fixing

We are using Github’s issues to track, report, and resolve bugs. When a bug is found during testing or development, a Github issue is opened with the label “bug”.

Bug severity is assessed according to the impact of the bug on the system and its functionality. A blocker or blocking bug is a bug that forces the system or functionality to be temporarily unavailable. Critical bugs are bugs that severely impact the functionality of the system, but do not block it completely. Major bugs are bugs that cause noticeable problems or inconveniences to users, but are not as severe as critical or blocking bugs. Normal severity bugs are those that have a moderate impact on a software’s functionality, are noticeable, but do not impact the core functionality significantly. Normal severity bugs are the most common during development and testing. Minor bugs are bugs with minimal impact on software functionality, but do not affect critical functionalities. Trivial bugs are those that have very little impact on software functionality and are usually not fixed immediately. Finally, enhancements are suggestions for improving the software’s functionality.

| **Bug ID** | 03 ([Github Issue #65](https://github.com/Ryan30012/AnaCondo/issues/65)) | | |
| --- | --- | --- | --- |
| **Originator** | Suha Abubakr | Email: [r\_sefri@live.concordia.ca](mailto:s_bubak@live.concordia.ca) | Signature :Github |
| **Submit Date** | April 10, 2024 | | |
| **Summary** | The condo companies' coupon page for adding new coupons can be accessed by anyone regardless of login whereas it should only be accessed by condo company accounts. | | |
| **Severity** | Critical (the software is still usable, but the bug is urgent and should be addressed immediately) | | |
| **Product** | Website (Anacondo) | | |
| **Component** | Condo Company Coupon Page | | |
| **Version** | 1.0 | | |
| **Platform** | PC | | |
| **OS** | Windows | | |
| **Browser** | Google Chrome | | |
| **URL** | localhost:3000/CondoCompany/Coupons/Add (website is running locally) | | |

Table 1: Bug Report 03

| **Bug ID** | 04 ([Github Issue #71](https://github.com/Ryan30012/AnaCondo/issues/71)) | | |
| --- | --- | --- | --- |
| **Originator** | Ryan Sefrioui | Email: [s\_bubak@live.concordia.ca](mailto:s_bubak@live.concordia.ca) | Signature :Github |
| **Submit Date** | April 14, 2024 | | |
| **Summary** | Deployment error caused by module ‘pg’ | | |
| **Severity** | Blocking (the bug must be addressed immediately since it prevents the coupons from being accessed) | | |
| **Product** | Website (Anacondo) | | |
| **Component** | Edit Profile Page, Registration Key Field/Form | | |
| **Version** | 1.0 | | |
| **Platform** | PC | | |
| **OS** | Windows | | |
| **Browser** | Google Chrome | | |
| **URL** | localhost:3000/CondoCompany/Coupons/page.tsx/ (website is running locally) | | |

Table 2: Bug Report 04

## 9.3 Defect Tracking Tool

For defect tracking, we opted to not use an external tool, because we felt it would over complicate things to have to track the bugs in an external platform. Instead we felt it would be easier to manage bugs if we tracked them directly in GitHub.

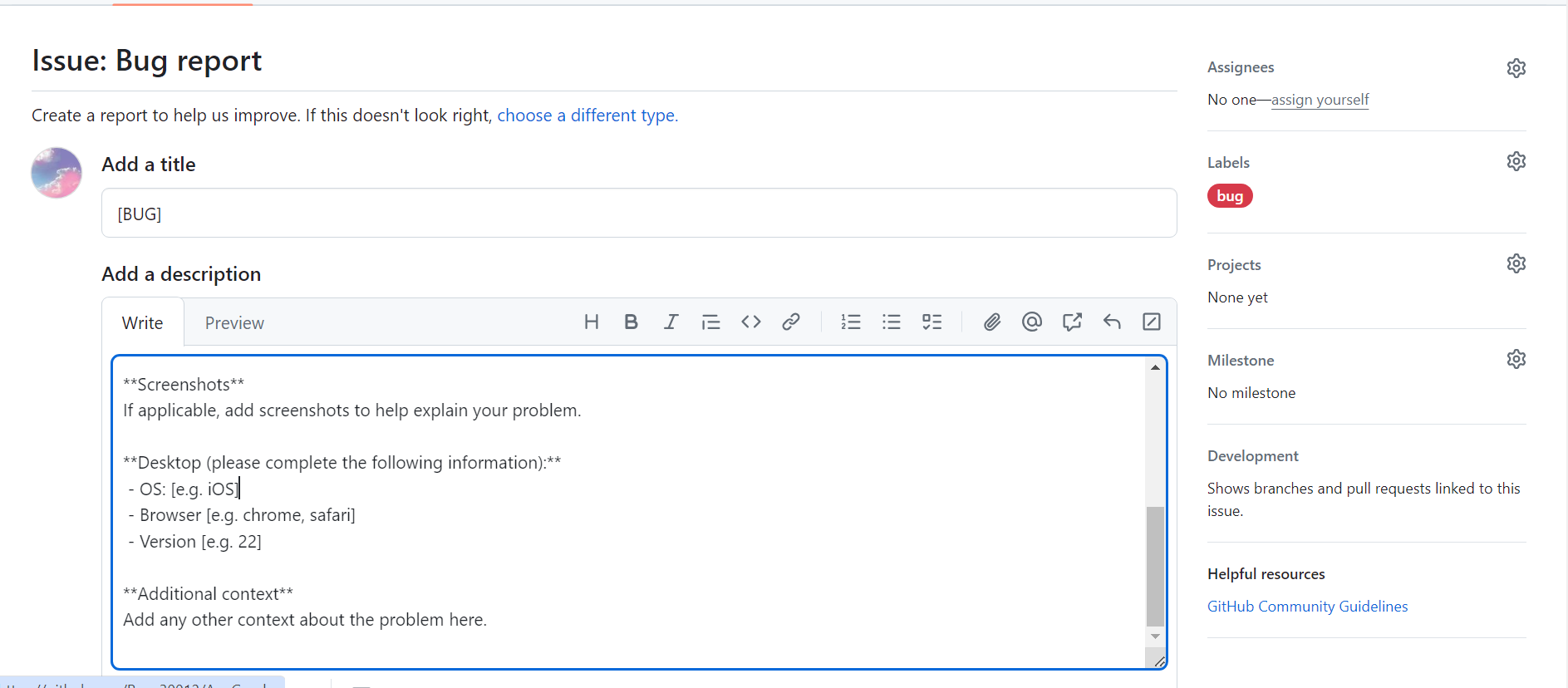
To streamline the bug tracking process in GitHub we have used a bug issue template. A screenshot of the template is below:  


Image 4: Bug Report Template

A proper example of how this template can be used is in the bug: ([Github Issue #46](https://github.com/Ryan30012/AnaCondo/issues/46)), which is shown in the following screenshot:

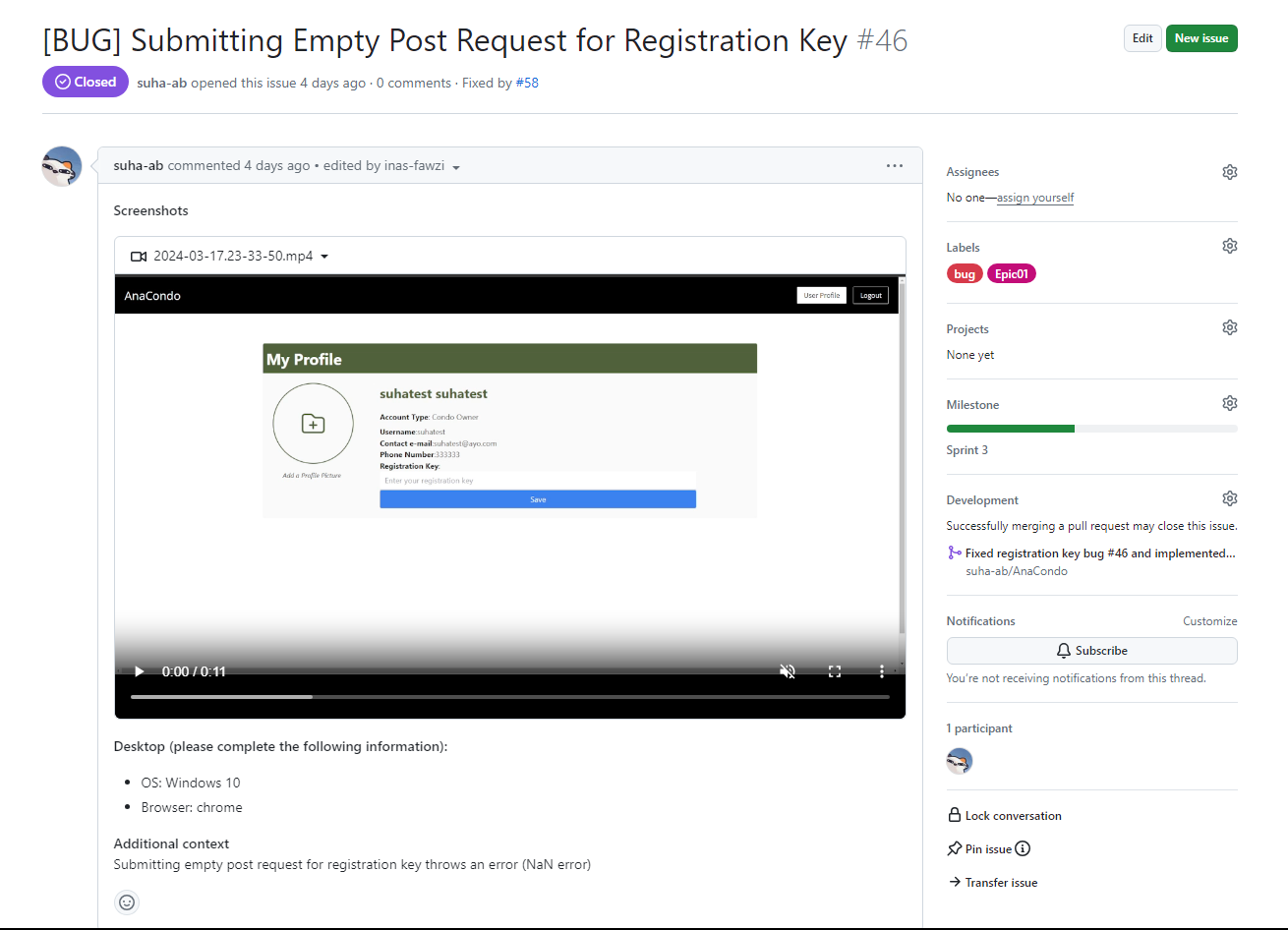


Image 5: Bug Report Example

## 9.4 Code Review Tools

Our main approaches to code management are pull request reviews and an automated service that evaluates our code, called Codacy.

For every pull request made a reviewer is added and the pull request is not merged until the reviewer has approved the pull request and added any comments they may have. If the reviewer disapproves the pull request the author must amend their code, or further explain their code to the reviewer, in an effort to get their pull request approved.

Aside from pull request reviews, we also occasionally do in person peer to peer code reviews, mainly between individuals working on related pieces of code. During code reviews the person explains their code and makes sure it can be understood by the reviewer. This serves to improve our code but also make sure that developers working on related parts can have a deeper understanding of each others’ work.

Codacy is used to automatically measure complexity and duplication. Codacy will also open issues to annotate pull requests based on problems found in the code.

Below is our code quality report generated by Codacy:

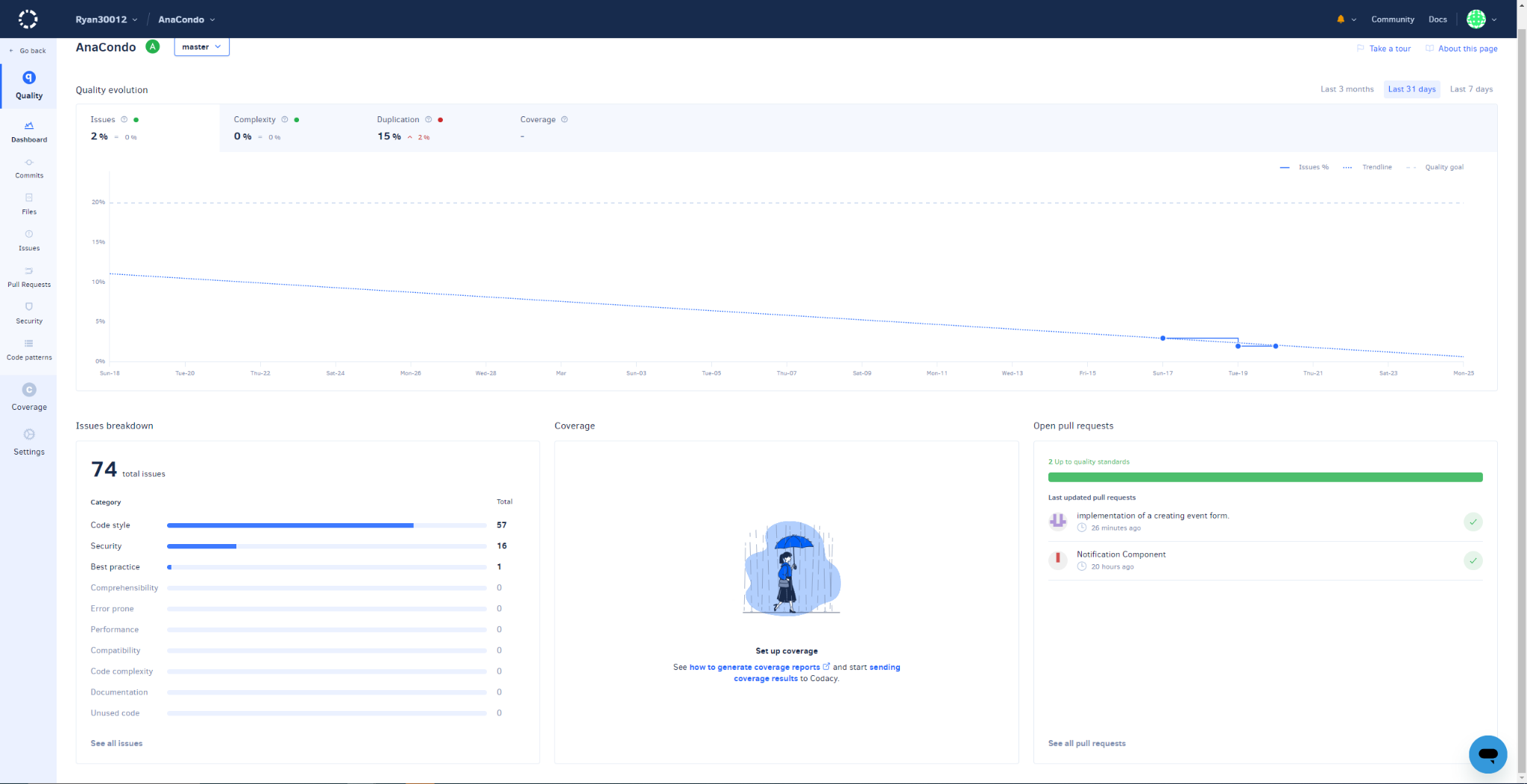


Image 6: Codacy Code Quality Report



Image 7: Codacy Code Quality Report

## 9.5 Usage of Design Patterns

In terms of design patterns, we have used one when implementing the condo owner dashboard. When creating this view, it requires the incorporation of several components to ensure that this user can view all the necessary components. For instance, it incorporates the components of RentalPropertyCard, RentalFinancialStatus, and RentalSubmittedRequests. It adapts these components into its own structure and adapts them to the user interface seen by the condo owner upon login. We had used the Adapter design pattern to create this structure when designing the condo owner dashboard. Additionally, the prototype pattern was used many times during our project. In fact, there were many cases where we have cloned the prototype due to needing reuse of code. Next, we have implemented the composite pattern for our users. It was necessary to define our user types in a hierarchical way. Finally, we used the strategy pattern to only allow certain features for certain types of users.

## 9.6 Code Coverage

As of sprint 4, we had aimed to achieve 70% code coverage as to be able to properly manage our upcoming sprints. To achieve this feat, we resorted to user story selection and separation amongst teammates. In fact, we managed a spreadsheet to indicate which team member would be given the responsibility for which user story completion, and have each worked on the completion of that user story. As for test cases, each team member was responsible for the test case related to their own user story in order to increase clarity of separation and team morale. Finally, concerning the database, we have achieved 90% coverage as it is necessary for the implementation of features.

## 9.7 Design Quality as Measured

Design quality can be further measured by metrics such as lines of code, number of packages, coupling between objects, and cyclomatic complexity. To retrieve our codebase’s lines of code, we used Codacy. Our repository currently has 27,628 lines of code. For our software, we did not have objects. Rather, we had typescript files that represented website routes. The coupling between the files is low since very little of them import each other. The files that require importing of other typescript files are placed in the same directory. If one file is a component, however, it is placed in the components directory and imported. For cyclomatic complexity, that can be measured using tools such as Codacy. Cyclomatic complexity measures the complexity of code by providing a quantitative measure of the number of linearly independent paths through the program's source code.

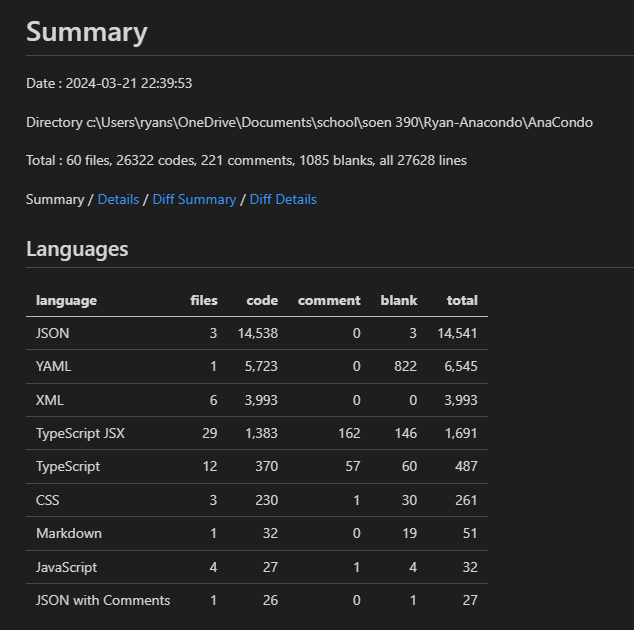


Image 8: Codacy Metrics Report

## 9.8 Quality of Source Code Documentation

To document our source code, we mainly resorted to inline comments where necessary. This was particularly useful when we needed to ask each other for help in debugging or implementing certain functionalities. An example of inline comments used in our code is provided below. JSDoc comments would be very useful for understanding functions as a whole, however, we currently do not have any JSDoc comments in our source code. Moving forward, we can gradually add more JSDoc and inline comments to improve our source code documentation.



Image 9: Inline Comments in Source Code

## 9.9 Refactoring Activity Documented In Commit Messages

As we are still in the midst of developing all the features for our product, refactoring has not been at the forefront of our sprint plans. However, there are a few instances of refactoring occurring within the code source storing our project. In this [first instance](https://github.com/Ryan30012/AnaCondo/commit/1bbcb991786b14f476ea689ca2b925074f2f72e5), the biggest trend was the removal of redundant code as well as the addition of shortened segments of code. While original functionality is preserved and improved, these smaller modifications allow for a quicker understanding of the file’s main feature. In this [second feature](https://github.com/Ryan30012/AnaCondo/commit/d95a0866fea3b37b142ec8aa89df048096f90fc8), code no longer deemed useful or relevant to the implemented function was removed from the code and some replacements were made to ensure that the current function works as intended. In this [third feature](https://github.com/Ryan30012/AnaCondo/commit/7cf9e91d118f52029220ae1015ae84790de92286), a focus on providing more readability through comments was made. Parts of the functionality that were not functioning were modified so that they would build. To keep record of the database usage for this feature, the sql commands were commented out for the sake of the future developers that will take over this portion of the code.

## 9.10 Quality and Detail of Commit Messages

A good commit message should indicate what change was made. Our team has agreed that commit messages should always mention exactly what was changed, and to avoid having commits that contain massive amounts of changes so that the commit message can effectively convey what the changes were. Unfortunately, our team is not consistent yet in making good commit messages. Some of them are better and others are worse. For example, a good commit message would say something like “fixed edit button in user dashboard” rather than “small fixes to user dashboard”. Compared to Sprint 3, our team’s commits have improved drastically. We adopted atomic commits and have agreed on a commit format.

## 9.11 Use of Feature Branches

In Sprint 4, we have fully transitioned to using feature branches. Now, every user story corresponds to a feature branch. Atomic commits are done on such a feature branch until the user story is completed. Once completed, a pull request is made which is reviewed by at least one team member. Once the pull request is reviewed and approved, it is merged into the main branch. For example, a feature branch was made for the user story 0424. Backend and frontend development for this user story was done on the feature branch before a pull request was made and the user story was merged into the master branch.

## 

Image 10: Feature Branch Pull Request

## 9.12 Atomic Commits

The following formats are to be used when contributing to the repository through pull requests and commit messages to ensure the code quality as well as a means to keep up the code coverage.

### 9.12.1 Pull request message format:

## Describe your changes

## Issue number

## Checklist before requesting a review

- [ ] I have performed a self-review of my code

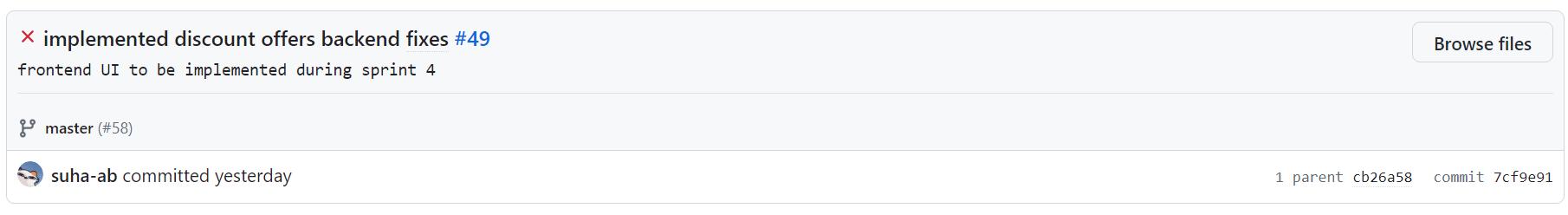
- [ ] There are no code conflicts with my current PR

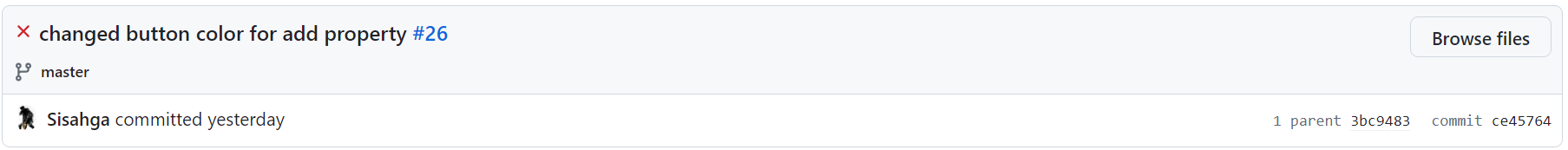
- [ ] There has been at least one unit test added for this feature being implemented/fixed

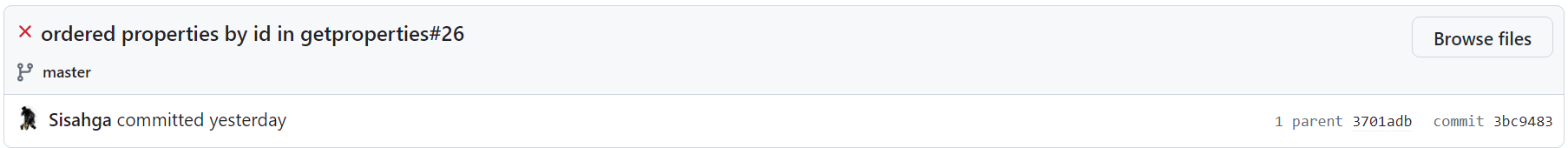
- [ ] I have gotten a peer-review of my code

### 9.12.2 Commit message format:

**[verb** eg: implementing/ed, updating/ed, adding/ed, fixing/ed**] [specific work item** eg: edit button, table rows, profile picture, login api…**](***optional:* **in [overarching area of work** eg: user dashboard, profile page…**]) [#issue number]**







Images 10-12: Good Commit Messages

## 9.13 Linking of Commits to Bug Reports/Features

Ideally, all commits and pull requests should be associated with an issue in GitHub for tracking purposes. This is a part of the process our team has agreed upon for doing commits and pull requests. However, this is another area where we are lacking consistency. Many commits and pull requests are properly linked to an issue, but there are many that are not linked as well. Going forward we will try to become more consistent. The format we are aiming to fully implement is for the commit message to be a description of the changes made to the files followed by the issue number. An example is given below.



Image 13: Linking Commits to Features