



## **SOEN 390 - Sprint 3 Documentation**

**March 21th, 2024**

**Department of Computer Science & Software Engineering**

Team Members	
Tanya So Tin Yan	40208954
Fadoua Doghmane	40198495
Asmae Loulidi	40210936
Muiz Madadi	40226708
Shamma Markis	40211998
Ihana Fahmy	40209686
Sonali Patel	40176580
Mehdi Fouzail	40101308
Valeria Dolgaliova	40212218
Robayth Shahrin Dhrubo	40123503

**Concordia University**

**Winter 2024**

# Table of Contents

Software Product Vision.....	4
1. Introduction.....	4
2. Positioning.....	4
3. Stakeholder and User Descriptions.....	4
4. Product Overview.....	9
5. Product Features.....	10
6. Other Product Requirements.....	11
User Stories Backlog.....	13
User Story 1.....	15
User Story 2.....	15
User Story 3.....	16
User Story 4.....	16
User Story 5.....	17
User Story 6.....	17
User Story 7.....	18
User Story 8.....	18
User Story 9.....	19
User Story 10.....	19
User Story 11.....	20
User Story 12.....	20
User Story 13.....	21
User Story 14.....	21
User Story 15.....	22
User Story 16.....	22
User Story 17.....	23
User Story 18.....	23
User Story 19.....	24
User Story 20.....	24
User Story 21.....	25
User Story 22.....	25
User Story 23.....	26
User Story 24.....	26
User Story 25.....	27
User Story 26.....	27
User Story 27.....	27
Software Architecture Document.....	28
Domain Model.....	28
Class Diagram.....	29
Component Diagram.....	30

Deployment Diagram.....	31
Activity Diagrams.....	32
Use Case Diagrams.....	43
Risk Assessment & Management Plan.....	55
Sprint 3 Testing Report.....	58
Short Sprint #3 Retrospective.....	62
Sprint 4 Release Plan.....	65
UI Prototypes.....	66
Home Page.....	66
Login.....	67
Sign Up.....	67
Property Page.....	68
Notification.....	69
Payment.....	69
Maintenance Requests.....	70
Financial Management Dashboard.....	70
Membership Information.....	71
Property Profile Management.....	72
Reservation System.....	73
Employees.....	74
Registration Key Page.....	75
Sprint 4 Testing Plan.....	80
Code Management.....	83

# Software Product Vision

## 1. Introduction

The purpose of this document is to collect, analyze, and define high-level needs and features of the Urbankey. It focuses on the capabilities needed by the stakeholders, and the target users, and **why** these needs exist. The details of how the Urbankey fulfills these needs are detailed in the use-case and supplementary specifications.

## 2. Positioning

### 2.1 Problem Statement

The problem of	inefficient procedures for managing condominiums
affects	public users, condo owners, rental users, condo management companies
the impact of which is	issues of integration, accessibility and complexity for the users, as well as the lack of essential features.
a successful solution would be	better communication between the users and condo owners, simplified financial and reservation system which will ultimately lead to an increased satisfaction among all stakeholders.

### 2.2 Product Position Statement

For	condo management companies and tenants
Who	need a simple and efficient condominium management system
The Urbankey	is a Condominium Management System
That	enables effective property management, simplified financial tracking, simplified reservations system
Unlike	current alternatives which are missing user-centric features and design
Our product	guarantees a user-friendly experience, catering to the particular requirements of condo owners, tenants, and management companies.

## 3. Stakeholder and User Descriptions

### 3.1 Stakeholder Summary

Name	Description	Responsibilities
Architects and Engineers	Professionals involved in the design of the property	<ul style="list-style-type: none"><li>• Develop architectural plans and designs for the condominium</li></ul>
Tech companies and programmers	Professionals involved in the creation and maintenance of the software of the system	<ul style="list-style-type: none"><li>• Software development</li><li>• Database management</li><li>• Updates and maintenance</li></ul>
Financial institutions	Institutions or banks providing loans or financial services	<ul style="list-style-type: none"><li>• Assess risks related to lending and investments</li><li>• Credit evaluation</li></ul>
Legal advisors	Professionals ensuring any operations regarding the condos complies with the law, and handle any legal issues	<ul style="list-style-type: none"><li>• Privacy and Data Protection</li></ul>

## 3.2 User Summary

Name	Description	Responsibilities	Stakeholder
Condo owners	Individuals who own condominium units	<ul style="list-style-type: none"> <li>• manages unit details</li> <li>• submit requests (moving in/out, violation reports, etc.)</li> <li>• access/update their property dashboards</li> <li>• reserve common facilities</li> <li>• engaging in financial transactions</li> </ul>	Condo management companies
Tenants	Individuals occupying condominium units through rental agreements	<ul style="list-style-type: none"> <li>• occupies and maintains the rented units</li> <li>• may engage in some financial transactions related to rental payments</li> </ul>	Condo management companies
Condo management companies	Companies responsible for managing condominium properties	<ul style="list-style-type: none"> <li>• Create and manage property profiles</li> <li>• Upload condo files</li> <li>• Manage financial aspects</li> <li>• Set up reservation system for common facilities</li> <li>• Assign roles to employees</li> <li>• Handle requests from condo owners/tenants</li> </ul>	N/A
Managers	Individual responsible for overseeing daily operations of condominium properties	<ul style="list-style-type: none"> <li>• Supervises property management tasks</li> <li>• Ensures smooth functioning of common facilities</li> <li>• Manages employee roles and responsibilities</li> <li>• Addresses escalated issues from condo owners/tenants</li> </ul>	Condo management companies
Operations Managers	Individuals responsible for the day-to-day operations of condominium properties	<ul style="list-style-type: none"> <li>• Coordinates maintenance and repairs</li> <li>• Ensures compliance with regulations and standards</li> <li>• Oversees staff scheduling and training</li> <li>• Resolves operational issues and emergencies</li> </ul>	Condo management companies
Finance Officers	Individuals responsible for financial aspects of condominiums management	<ul style="list-style-type: none"> <li>• Manages condo fees and operational budget</li> <li>• Tracks financial transactions and expenses</li> <li>• Generates financial reports</li> <li>• Ensures financial compliance and accuracy</li> </ul>	Condo management companies

### 3.3 User Environment

- The number of people involved might vary, from condo owners or tenants taking care of their apartments to staff members of condo management companies handling several buildings. Depending on the participation of users or the property offers, the user base may grow or shrink.
- Task cycles change based on the type of activity of different users. For example, a condo owner can dedicate more time to financial transactions or managing reservations
- Users access the condo management system through devices such as laptops and smartphones, creating a mostly digital environment.
- Currently, the condo management system is accessible via web browsers. In the future, there are plans to develop dedicated Android and iOS applications, expanding the platform compatibility to include mobile devices.
- Users can use multiple applications, including messaging and communication platforms like email, and financial apps for transactions.

### 3.4 Key Stakeholder or User Needs

Need	Priority	Concerns	Current Solution	Proposed Solution
Better Communication	High	Slow responses and not clear	Emails, phone calls, in-person meeting	One place/page for all messages and notifications
Clearer Finances	High	Confusing fees and hard to understand	Using spreadsheets and done manually	Easy-to-read financial info and clear breakdowns
Easy Booking	High	Difficulty booking and prone to errors	Online booking with real-time updates	Simplified booking process with intuitive interface and personalized recommendations
Quicker Resolution	High	Delays in resolving issues and lack of visibility	Sending emails and making phone calls	Tracking status of requests and receiving updates in real-time
Broadcast Messages	Medium	Messages not reaching everyone and lack of central news hub	Notices and physical notes	Centralized platforms for important announcements and updates
User-friendly interface	High	Difficulty in navigating the system	Complex menus and unintuitive layout	Design an intuitive and responsive interface with clear navigation paths and consistent design elements
Data security	High	Concerns about privacy and data protection	Lack of encryption or data breaches	Implement robust data encryption protocols and regular security audits to ensure user data protection
Mobile accessibility	High	Limited access to features on mobile devices	Limited mobile app functionality or lack of compatibility across devices	Develop a responsive mobile application with feature parity and seamless user experience across different devices
Timely notifications	Medium	Missed updates or important events	Manual checking of emails or notifications	Enable push notifications and customizable alert settings to ensure timely updates and reminders
Streamlined workflows	Medium	Inefficient or redundant processes	Manual data entry and disjointed workflows	Automate repetitive tasks and streamline processes to improve efficiency and productivity

**Table 1: Key Stakeholders' and User's Needs**

<b>Description</b>	Centralized messaging system
<b>Type</b>	System enhancement
<b>Responsibilities</b>	Develop system for sending/receiving messages and notification
<b>Success Criteria</b>	Easily accessible messages for stakeholders
<b>Involvement</b>	Product manager, development team
<b>Deliverables</b> <b>Comments/Issues</b>	Ensure compatibility across various devices.

*Table 2: Better Communication*

<b>Description</b>	User-friendly financial module
<b>Type</b>	System Enhancement
<b>Responsibilities</b>	Design financial module with clear fee breakdowns
<b>Success Criteria</b>	Understandable financial information
<b>Involvement</b>	Product manager, financial analyst and development team
<b>Deliverables</b> <b>Comments/Issues</b>	Ensure scalability for future updates and additions.

*Table 3: Clearer Financial Information*

<b>Description</b>	Intuitive booking system
<b>Type</b>	New feature
<b>Responsibilities</b>	Develop user-friendly booking system with personalized recommendations
<b>Success Criteria</b>	Simplified, error-free booking process
<b>Involvement</b>	Product manager, UX/UI designer
<b>Deliverables</b> <b>Comments/Issues</b>	Ensure integration with existing reservation system.

*Table 4: Easy Booking*

<b>Description</b>	Real-time request tracking
<b>Type</b>	System enhancement
<b>Responsibilities</b>	Implement system for real-time tracking and updates on requests
<b>Success Criteria</b>	Easy request tracking and timely updates
<b>Involvement</b>	Product manager and development team
<b>Deliverables</b> <b>Comments/Issues</b>	Ensure compatibility with different request types.

*Table 5: Quicker Resolution*

<b>Description</b>	Centralizes messaging platform
<b>Type</b>	New feature
<b>Responsibilities</b>	Develop platform for broadcasting announcements to all users
<b>Success Criteria</b>	Effective communication of announcements
<b>Involvement</b>	Product Manager, Communication specialist and development team
<b>Deliverables</b> <b>Comments/Issues</b>	Ensure scalability for large user base.

*Table 6: Broadcast Message*

<b>Description</b>	Design an intuitive and responsive interface with clear navigation paths and consistent design elements
<b>Type</b>	UI/UX Design
<b>Responsibilities</b>	Designing user interface elements, navigation flows, and visual design
<b>Success Criteria</b>	User satisfaction with interface usability and navigation
<b>Involvement</b>	UI/UX Designers, Developers
<b>Deliverables</b> <b>Comments/Issues</b>	Ensure compliance with accessibility standards.

*Table 7: User-friendly interface*

<b>Description</b>	Implement robust data encryption protocols and regular security audits to ensure user data protection
<b>Type</b>	Security
<b>Responsibilities</b>	Implementing encryption algorithms, securing databases, and conducting security audits
<b>Success Criteria</b>	Absence of data breaches and compliance with privacy regulations
<b>Involvement</b>	Security Experts, Developers
<b>Deliverables</b> <b>Comments/Issues</b>	Ensure adherence to industry security standards.

*Table 8: Data Security*

<b>Description</b>	Develop a responsive mobile application with feature parity and seamless user experience across different devices
<b>Type</b>	Development
<b>Responsibilities</b>	Developing mobile application features and ensuring cross-device compatibility
<b>Success Criteria</b>	User engagement and satisfaction with mobile app functionality
<b>Involvement</b>	Developers, UI/UX Designers
<b>Deliverables</b> <b>Comments/Issues</b>	Ensure optimization for various screen sizes and resolutions.

*Table 9: Mobile accessibility*

<b>Description</b>	Enable push notifications and customizable alert settings to ensure timely updates and reminders
<b>Type</b>	Notification
<b>Responsibilities</b>	Implementing push notification functionality and user preference settings
<b>Success Criteria</b>	Users receiving timely notifications and reminders
<b>Involvement</b>	Developers, Product Managers
<b>Deliverables</b> <b>Comments/Issues</b>	Ensure compatibility across different notification channels.

*Table 10: Timely notifications*

<b>Description</b>	Automate repetitive tasks and streamline processes to improve efficiency and productivity
<b>Type</b>	Workflow Improvement
<b>Responsibilities</b>	Analyzing existing processes, identifying automation opportunities, and implementing workflow improvements
<b>Success Criteria</b>	Increased efficiency and productivity, reduced manual intervention
<b>Involvement</b>	Business Analysts, Developers
<b>Deliverables</b> <b>Comments/Issues</b>	Ensure user training and adoption for new workflow processes.

*Table 11: Streamlined workflows*

### 3.5 Alternatives and Competition

An alternative for the stakeholders would be to build their own management system. They could also buy an existing condo management system, add additional features and maintain it.

#### *Alternatives:*

- Build their own management system
- Buy an existing one, implement additional features and maintain it

These alternatives would allow the stakeholders to have full control on the system.

## ***Competition:***

- Existing condo management systems
- Real estate firms

Some companies that offer competition in the condo management systems market:

**Buildium:** Buildium offers a comprehensive property management solution designed for property managers, landlords, and homeowner associations. It includes features such as accounting, lease tracking, maintenance requests, and online payments.

**AppFolio Property Manager:** AppFolio is a cloud-based property management software that caters to residential, commercial, student housing, and community association property managers. It provides features like online rent payments, maintenance requests, and vacancy advertising.

**Condo Control Central:** Condo Control Central provides web-based property management software designed specifically for condominiums and HOAs. It offers features such as visitor management, amenity booking, maintenance tracking, and document management.

**TOPS Software:** TOPS Software offers a suite of community management solutions, including software for condominiums, homeowner associations, and co-ops. Their products include features like accounting, communication tools, architectural request management, and community websites.

**Yardi Voyager:** Yardi Voyager is a property management platform that serves various real estate sectors, including residential, commercial, and condominium management. It offers modules for accounting, leasing, maintenance, and reporting tailored to the specific needs of each property type.

To distinguish itself as a top condo management software from competitors such as Buildium, AppFolio Property Manager, Condo Control Central, TOPS Software, and Yardi Voyager, UrbanKey will focus on a smooth user experience with an intuitive UI and configurable functionality. UrbanKey would excel in financial management by providing full tools for fee tracking, budgeting, and transparent reporting. Mobile accessibility and strong data security safeguards would enable easy access and user data safety. UrbanKey will also innovate by offering unique features such as enhanced amenities booking and smart home integration, as well as superior customer service, to create user happiness and loyalty.

## **4. Product Overview**

This section presents both product perspective and assumptions and dependencies of the Urbankay.

### **4.1 Product Perspective**

The purpose of the condo management app and website is to simplify condo management procedures by activating as separate, self-contained platforms. The application enables users to interact with it, while the website provides further features that can be accessed by web browsers. User profiles, finance systems, property management tools, reservation capabilities, and request submission features are important parts. These platforms work together flawlessly to give management firms, condo owners, and renters all the resources they need to manage properties effectively. Widespread accessibility is ensured via external interfaces' interaction with several operating systems.

## **4.2 Assumptions and Dependencies**

The availability of registration keys from management organizations, correct property data input, and the smooth operation of the finance and reservation systems are some of the assumptions and dependencies for the condo management system. Any modifications to these elements may have an impact on the characteristics listed in the Vision report. Furthermore, assumptions on the app's compatibility with multiple platforms, language options, and login methods like Gmail or Single Sign-On are critical, since changes may need adjustments to the Vision document.

## **5. Product Features**

### **5.1 User Profile**

Public users can create their own unique profile. This profile should include a profile picture, user name, contact email, phone number. Moreover, public users are required to provide a registration key obtained by their condo management company to become a condo owner. To become rental users in the system, public users must input a registration key obtained from their condominium management company.

### **5.2 Condo Owner Dashboard**

Condo owner can view features of their properties such as general information, personal profile, condo information, financial status, status of the submitted request, etc in a dashboard

### **5.3 Condo Management Companies Profile**

The property profile in the system requires a property name, unit count, parking count, locker count, and address. Condo management companies can upload files for each property, and those files are accessible to all condo owners. They can include detailed information about condo units, parking spots, and lockers, encompassing unit size, owner details, occupant information, and associated condo fees. Furthermore, management companies can send registration keys to unit owners or rental users, allowing them to link their profiles with specific condo units.

### **5.4 Financial System**

Management companies input condo fees per square foot and parking spot. Condo fees for each unit are calculated and presented to owners, recorded in the financial system, along with operational budgets and costs. An annual report can be generated, summarizing condo fee collections for the year. The system also includes a reservation feature for common facilities, like a sky lounge or spa fitness. Condo owners and rental users can use a calendar-like interface to reserve facilities, with real-time availability display. Reservations operate on a first-come-first-serve basis, rendering a facility unavailable once booked.

### **5.6 Reservation System**

The condominium management system features a simple reservation system where condo management companies establish reservations for common facilities such as a sky lounge and a spa fitness center. This system is presented in a calendar-like interface, and allows both condo owners and renters to reserve these common facilities. Availability for these facilities is displayed, and reservations are processed on a first-come-first-serve basis. Once a facility is booked, it becomes temporarily unavailable for the reserved duration.

## **5.7 Reservation System**

Condo management firms have the ability to assign distinct roles to various employees overseeing the same property. These roles may include a manager, responsible for day-to-day operations, and an employee handling financial responsibilities.

## **5.8 Requests**

Condo owners can submit various requests, such as move-in/out dates for reserving elevators, intercom changes, access requests for fobs or keys, reporting violations, highlighting deficiencies in common areas, or seeking information. Each request is directed to the appropriate employee based on its type.

## **5.9 Notifications**

Every user has a notifications page where they can view the most recent activities related to their submitted or assigned requests.

# **6. Other Product Requirements**

## **Standards, Hardware, or Platform Requirements**

The software product must adhere to industry-standard security protocols to ensure the protection of user data. Additionally, it should be compatible with widely used operating systems including Android, iOS, Linux, MacOS, or Windows.

## **Performance Requirements**

Response time for critical user interactions, such as profile creation and reservation submissions, should be within 2 seconds to ensure a seamless experience. And system uptime must be maintained at 99.9% to minimize service disruptions, with downtime limited to scheduled maintenance windows.

## **Environmental Requirements**

The system should be built with optimal resource use to save CPU and battery consumption, and it should work well on both desktop and mobile devices. It is important to take into account how much bandwidth is used, particularly for customers who have spotty internet access, and strive for data transfer speeds that are suited for 3G and 4G networks.

## **Quality Ranges**

- **Performance:** average response time, peak response time, and throughput are important variables to consider when optimizing response times for effective user interactions.
- **Robustness:** metrics like error rates, crash frequencies, and mean time to failure (MTTF) should be used to gauge how well the system withstands unforeseen mistakes and handles heavy traffic loads.
- **Fault tolerance:** recovery time objective (RTO) and recovery point objective (RPO) metrics, together with error recovery and data backup mechanisms, should be in place to guarantee system dependability. The RTO provides the maximum permissible delay for system restoration following an incident, while RPO establishes the maximum allowable data loss in the case of a failure or interruption.

- **Usability:** tested by usability experts and user satisfaction questionnaires, the interface should be simple to use, easy to navigate, and provide insightful feedback.

## Design Constraints and Dependencies

Compatibility testing may be necessary to ensure smooth data interchange, and integration with current databases and condo management software may be necessary. Adherence to legal and regulatory obligations for privacy and data processing, with frequent audits carried out to confirm compliance with standards.

## Documentation Requirements

To guarantee widespread adoption, thorough user manuals and online help resources covering subjects like account creation, feature usage, and troubleshooting techniques should be made available. Accessibility metrics for the material should also be monitored. In order to maintain uniformity across platforms, labeling and packaging regulations may incorporate branding rules.

## Priority and Attributes

- **Stability:** maintaining user trust and satisfaction requires a sturdy and reliable system.
- **Benefit:** with key performance indicators set up to track the effects of features on user happiness and productivity, the system's features should offer users observable advantages like increased efficiency and communication
- **Effort:** ensure that the project deadlines are fulfilled, development efforts should be concentrated on integrating key features and guaranteeing peak performance. Metrics for allocating resources, such as development hours and job completion rates, should be tracked.
- **Risk:** risk assessment matrices are used to prioritize risk in order to mitigate the risks associated with data security breaches and system failures, which are crucial for safeguarding user information and preserving business continuity.
- **Communication:** Establishing clear and effective communication channels between stakeholders, development teams, and end-users is essential for ensuring alignment of goals and expectations throughout the project lifecycle. Regular status updates, meetings, and feedback sessions should be conducted to facilitate transparent communication and foster collaboration. Additionally, providing user-friendly interfaces for communication within the system can enhance user engagement and satisfaction.

# User Stories Backlog

## *User Story #1*

As a public user, I want to personalize my profile with a picture, username, and contact details for community engagement.

## *User Story #2*

As a public user, I want to use a registration key from the management company to securely become a condo owner in the system.

## *User Story #3*

As a public user, I want to use a registration key from the management company to securely become a rental user in the system.

## *User Story #4*

As a condo owner, I want a comprehensive dashboard displaying property and financial details, allowing efficient property management.

## [Issue #18](#)

## *User Story #5*

As a property manager, I want to create a property profile with essential details (property name, unit count, parking count, locker count, address) for accurate property management.

## [Issue #17](#)

## *User Story #6*

As a property manager, I want to upload condo files for each property so that all condo owners can access important information, enhancing transparency and communication.

## *User Story #7*

As a property manager, I want to enter detailed information for each condo unit, parking spot, and locker, including unit ID, size, owner details, occupant information, and condo fees, ensuring a comprehensive and accurate property database.

## *User Story #8*

As a property manager, I want to be able to generate and send registration keys to unit owners and rental users, so that they can link a condo unit to their profile.

## *User Story #9*

As a property manager, I want to enter condo fee per square foot and per parking spot, so that I can accurately calculate the fee for each condo unit.

## *User Story #10*

As a property manager, I want to calculate and present the fee for each unit, so that I can present it to the unit owner.

*User Story #11*

As a financial system agent, I want the system to record both the operational budget and the associated costs for each operation, so that we can maintain transparent financial management.

*User Story #12*

As a financial system agent, I want to generate an annual report summarizing all the fees collected for a given year.

*User Story #13*

As a property manager, I want to set up a system for managing reservations of common facilities, to allow residents to schedule and access these amenities.

*User Story #14*

As a condo owner/rental user, I want to be able to reserve common facilities within the complex through a calendar-like interface, to efficiently plan my activities.

*User Story #15*

As a condo owner/rental user, I want to know the availability of common facilities, so that I can easily plan and book the shared amenities for my personal or guest use.

*User Story #16*

As a property manager, I want the reservation system to operate on a first-come-first-serve basis so that once a facility is booked, it becomes unavailable for the reserved time and ensures fair access to common amenities.

*User Story #17*

As a property manager, I want to be able to set up different roles (e.g., manager, finance) for employees who are responsible for the same property so that the efficiency of assigning and managing tasks within the team becomes more organized.

*User Story #18*

As a condo owner, I want to submit requests for tasks like moving, intercom changes, access items, violation reports, common area issues, and questions so that addressing needs becomes easy and convenient.

*User Story #19*

As a property manager, I want each request to be assigned to the appropriate employee based on the type of request so that the resolution process for residents remains smooth and efficient.

*User Story #20*

As a public user, I want to have a notification page where I can view the latest activities related to my submitted or assigned requests so that I can stay informed on the progress and updates of the condo.

*User Story #21*

As a public user, I want to be able to post and reply to other users through a forum.

*User Story #22*

As a public user, I want to be able organize events and invite other occupants to attend.

*User Story #23*

As a property manager, I want to list coupons and offers for all the unit owners and tenants to see.

### *User Story #24*

As a public user, I want to access the app through a multitude of platforms like Android, iOS, Linux, MacOS, and Windows.

### *User Story #25*

As a public user, I want the app to be available in both French and English.

### *User Story #26*

As a public user, I want to be able to sign in using my Gmail account or Apple account.

### *User Story #27*

As a user, I want to be able to navigate the website efficiently so that I can access the information I need quickly.

## **User Story 1**

001	Public User Profile Creation					
<b>As a public user</b>						
<b>I want</b> to personalize my profile with a picture, username, and contact details						
<b>Because</b> I am looking for community engagement						
<b>User Profile Feature</b>						
Must	Bus. Value: M	Risk: M	Effort: 5			
<b>Status:</b>						

Front-end: Completed

Backend: User is able to personalize profile but **not add picture yet**

## **User Story 2**

002	Becoming a Condo Owner					
<b>As a public user</b>						
<b>I want</b> to use a registration key from the management company						
<b>To</b> securely become a condo owner in the system						
<b>Registration Key Feature</b>						
Must	Bus. Value: H	Risk: M	Effort: 6			
<b>Status:</b>						

Front-end: Completed

Backend: Completed

### User Story 3

003	Becoming a Rental User					
<b>As</b> a public user						
<b>I want</b> to use a registration key from the management company						
Must	Bus. Value: H	Risk: M	Effort: 6			

Status:

Front-end: Completed

Backend: Completed

### User Story 4

004	Condo Owner Profile					
<b>As</b> a condo owner						
<b>I want</b> a comprehensive dashboard displaying property and financial details						
Must	Bus. Value: H	Risk: M	Effort: 8			

Status:

Front-end: Completed

Backend: Completed

## User Story 5

005	Property Profile Creation					
<b>As</b> a property manager						
<b>I want</b> to create a property profile with essential details (property name, unit count, parking count, locker count, address)						
<b>For</b> accurate property management						
Condo Management Company Feature						
Must	Bus. Value: H	Risk: M	Effort: 8			

Status:

Front-end: Completed

Backend: Not Started

## User Story 6

006	File Upload for Condo Management Companies					
<b>As</b> a property manager						
<b>I want</b> to upload condo files for each property so that all condo owners can access important information						
<b>Because</b> I would like to enhance transparency and communication						
File Upload Feature						
Could	Bus. Value: M	Risk: M	Effort: 5			

Status:

Front-end: Completed

Backend: Not Started

## User Story 7

007	Condo Management Company Information					
<b>As</b> a property manager						
<b>I want</b> to enter detailed information for each condo unit, parking spot, and locker, including unit ID, size, owner details, occupant information, and condo fees						
<b>To</b> ensure a comprehensive and accurate property database						
Condo Management Company Feature						
Must	Bus. Value: H	Risk: M	Effort: 8			

Status:

Front-end: Completed

Backend: Not Started

## User Story 8

008	Generate and send registration code					
<b>As</b> a property manager						
<b>I want</b> to be able to generate and send registration keys to unit owners and rental users						
<b>To</b> link a condo unit to the user's profile.						
Registration Key Feature						
Must	Bus. Value: M	Risk: H	Effort: 5			

Status:

Front-end: In Progress

Backend: Not Started

## User Story 9

009	Enter condo unit fee					
<b>As</b> a property manager						
<b>I want to</b> enter condo fees per square foot and per parking spot						
Must	Bus. Value: M	Risk: H	Effort: 3			

Status:

Front-end: Completed

Backend: Not Started

## User Story 10

010	Calculate condo unit fee					
<b>As</b> a property manager						
<b>I want to</b> calculate and present the fee for each unit,						
Must	Bus. Value: M	Risk: M	Effort: 8			

Status:

Front-end: Completed

Backend: Not Started

## User Story 11

011	Record operational budget					
<b>As</b> a financial system agent						
<b>I want</b> the system to record both the operational budget and the associated costs for each operation						
<b>To</b> maintain transparent financial management.						
Operational Budget Feature						
Should	Bus. Value: M	Risk: H	Effort: 5			

Status:

Front-end: Completed

Backend: Not Started

## User Story 12

012	Generate annual report					
<b>As</b> a financial system agent						
<b>I want to</b> generate an annual report summarizing all the fees collected for a given year						
<b>To</b> organize the finances for each year.						
Annual Report Feature						
Should	Bus. Value: M	Risk: H	Effort: 13			

Status:

Front-end: Completed

Backend: Not Started

## User Story 13

013	Set up facility reservation system					
<b>As</b> a property manager						
<b>I want to</b> set up a system for managing reservations of common facilities						
Must	Bus. Value: M	Risk: M	Effort: 8			

Status:  
Front-end: In Progress  
Backend: Not Started

## User Story 14

014	Reserve common facilities					
<b>As</b> a condo owner/rental user						
<b>I want to</b> be able to reserve common facilities within the complex through a calendar-like interface						
Must	Bus. Value: M	Risk: M	Effort: 5			

Status:  
Front-end: Completed  
Backend: Not Started

## User Story 15

015	Knowing the Common Facilities' Availability					
<b>As</b> a condo owner/rental user						
<b>I want</b> to know the availability of common facilities						
<b>So</b> that I can easily plan and book the shared amenities for my personal or guest use						
Reservation System Feature						
M	Bus. Value: H	Risk: L	Effort: 4			

Status:

Front-end: Completed

Backend: Not Started

## User Story 16

016	First-Come-First-Serve System					
<b>As</b> a property manager						
<b>I want</b> the reservation system to operate on a first-come-first-serve basis						
<b>So</b> that once a facility is booked, it becomes unavailable for the reserved time and ensures fair access to common amenities						
Reservation System Feature						
M	Bus. Value: M	Risk: M	Effort: 6			

Status:

Front-end: Completed

Backend: Not Started

## User Story 17

017	Setting Up Different Roles					
<b>As</b> a property manager						
<b>I want</b> to set up different roles (e.g., manager, finance) for employees who are responsible for the same property						
<b>So</b> that the efficiency of assigning and managing tasks within the team becomes more organized						
Setting different roles for different employees Feature						
M	Bus. Value: H	Risk: H	Effort: 7			
Status: Front-end: Completed Backend: Not Started						

## User Story 18

018	Submission of Requests					
<b>As</b> a condo owner						
<b>I want</b> to submit requests for tasks like moving, intercom changes, access items, violation reports, common area issues, and questions						
<b>So</b> that addressing needs becomes easy and convenient						
Request Feature						
M	Bus. Value: M	Risk: M	Effort: 6			
Status: Front-end: Completed Backend: Not Started						

## User Story 19

019	Request Assignment					
<b>As</b> a property manager						
<b>I want</b> each request to be assigned to the appropriate employee based on the type of request						
<b>So</b> that the resolution process for residents remains smooth and efficient						
Request Feature						
M	Bus. Value: H	Risk: H	Effort: 8			

Status:

Front-end: Completed

Backend: Not Started

## User Story 20

020	Notification Page					
<b>As</b> a public user						
<b>I want</b> to have a notification page where I can view the latest activities related to my submitted or assigned requests						
<b>So</b> that I can stay informed on the progress and updates of the condo						
Notification Page Feature						
C	Bus. Value: M	Risk: M	Effort: 7			

Status:

Front-end: Completed

Backend: Not Started

## User Story 21

021	Forum posts and replies					
<b>As</b> a public user						
<b>I want</b> to be able to post and reply to other users through a forum						
<b>because</b> I want to interact with neighbors						
Forum feature						
Should	Bus. Value: M	Risk: L	Effort:5			

Status:

Front-end: Completed

Backend: Not Started

## User Story 22

022	Organize Events					
<b>As</b> a public user						
<b>I want</b> to be able to organize events and invite other occupants						
<b>because</b> I want to interact with neighbors						
Event Organization feature						
Could	Bus. Value: L	Risk: L	Effort:5			

Status:

Front-end: Completed

Backend: Not Started

## User Story 23

023	Coupon Listing					
<b>As</b> a property manager						
<b>I want</b> to list coupons and offers for all the unit owners and tenants to see						
<b>because</b> I want to offer discounts or deals						
Coupon listing feature						
Could	Bus. Value: L	Risk: L	Effort:5			

Status:

Front-end: Completed

Backend: Not Started

## User Story 24

024	Multiple Platform Access					
<b>As</b> a public user						
<b>I want</b> to access the app through a multitude of platforms						
<b>because</b> I want to be able to use Android, iOS, Linux, MacOS, and Windows						
Multiple platform access feature						
Should	Bus. Value: M	Risk: M	Effort:6			

Status:

Front-end: Completed

Backend: Not Started

## User Story 25

025	Language Options					
<b>As</b> a public user						
<b>I want</b> the app to be available in both French and English						
<b>because</b> I might only know either or, and we are in Quebec						
Language feature						
Must	Bus. Value: H	Risk: M	Effort:6			
Status: Front-end: Completed						

## User Story 26

026	Sign in Options					
<b>As</b> a public user						
<b>I want</b> to be able to sign in using my Gmail account or Apple account						
<b>because</b> I want everything synched						
Sign in options feature						
Should	Bus. Value: M	Risk: M	Effort:6			
Status: Front-end: Completed Backend: Not Started						

## User Story 27

027	Navigation Bars					
<b>As</b> a user						
<b>I want</b> to be able to navigate the website efficiently						
<b>because</b> I want to access the information I need quickly						
Navigation Bars Feature						
Should	Bus. Value: H	Risk: M	Effort:8			

Front-end: Completed  
Backend: In progress

# Software Architecture Document

## Domain Model

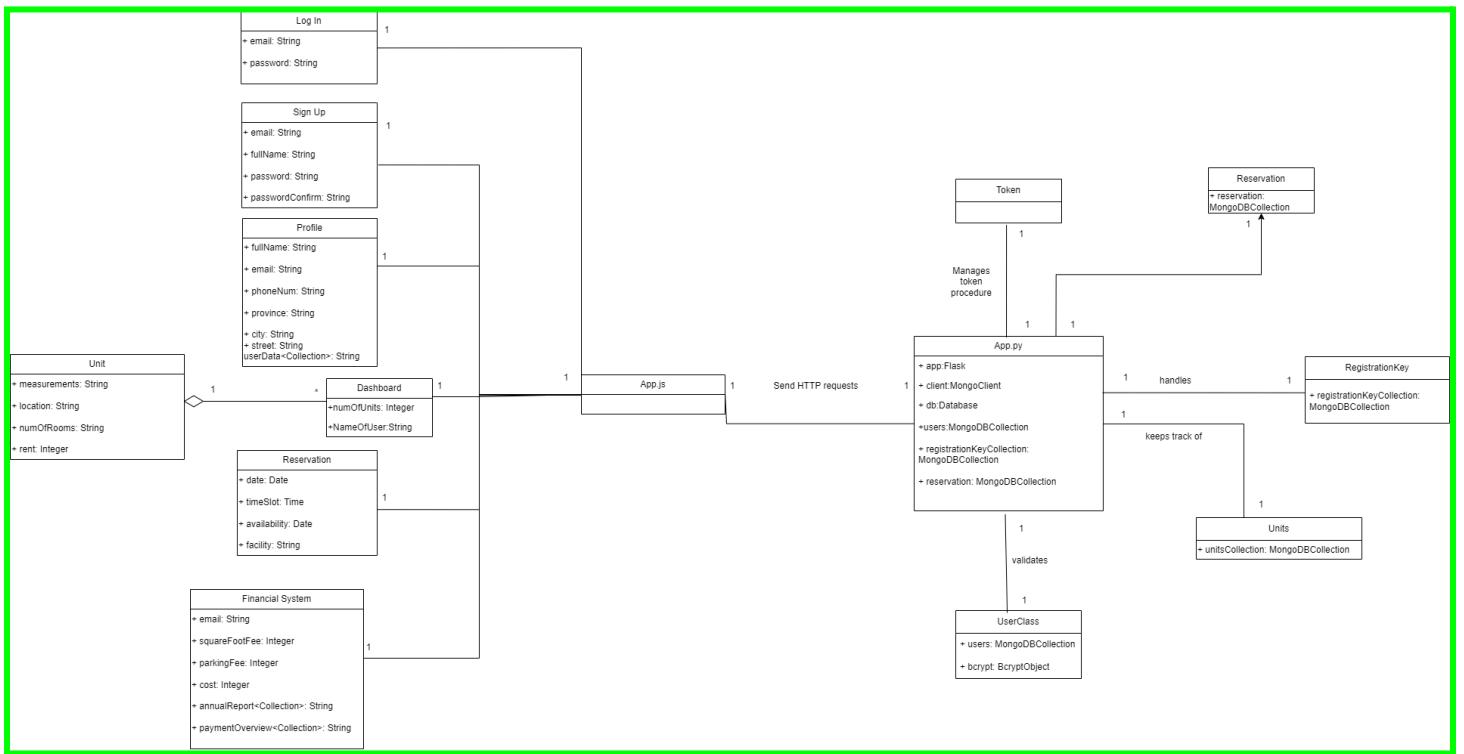
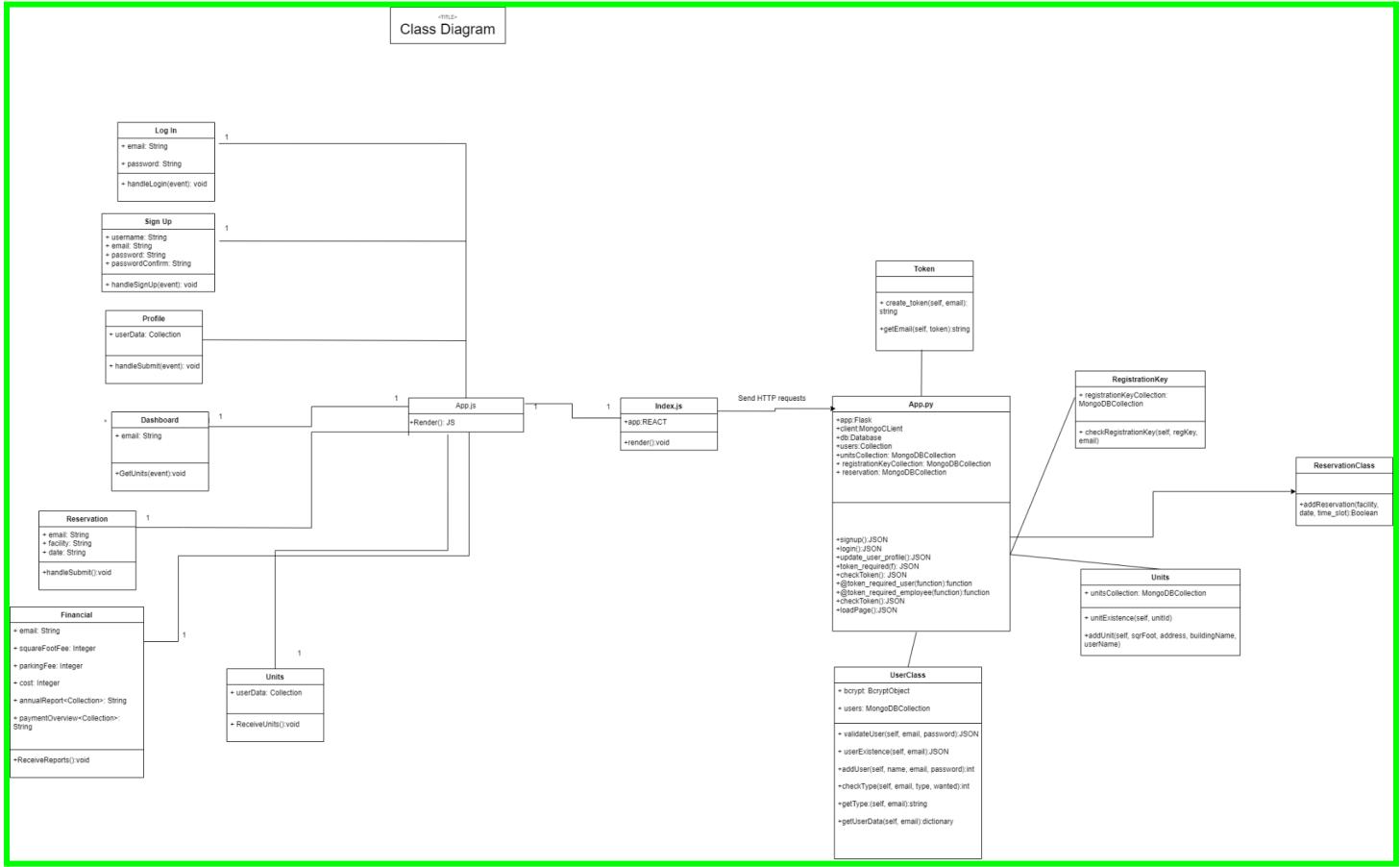


Figure 1: Domain Model

# Class Diagram



**Figure 2: Class Diagram**

# Component Diagram

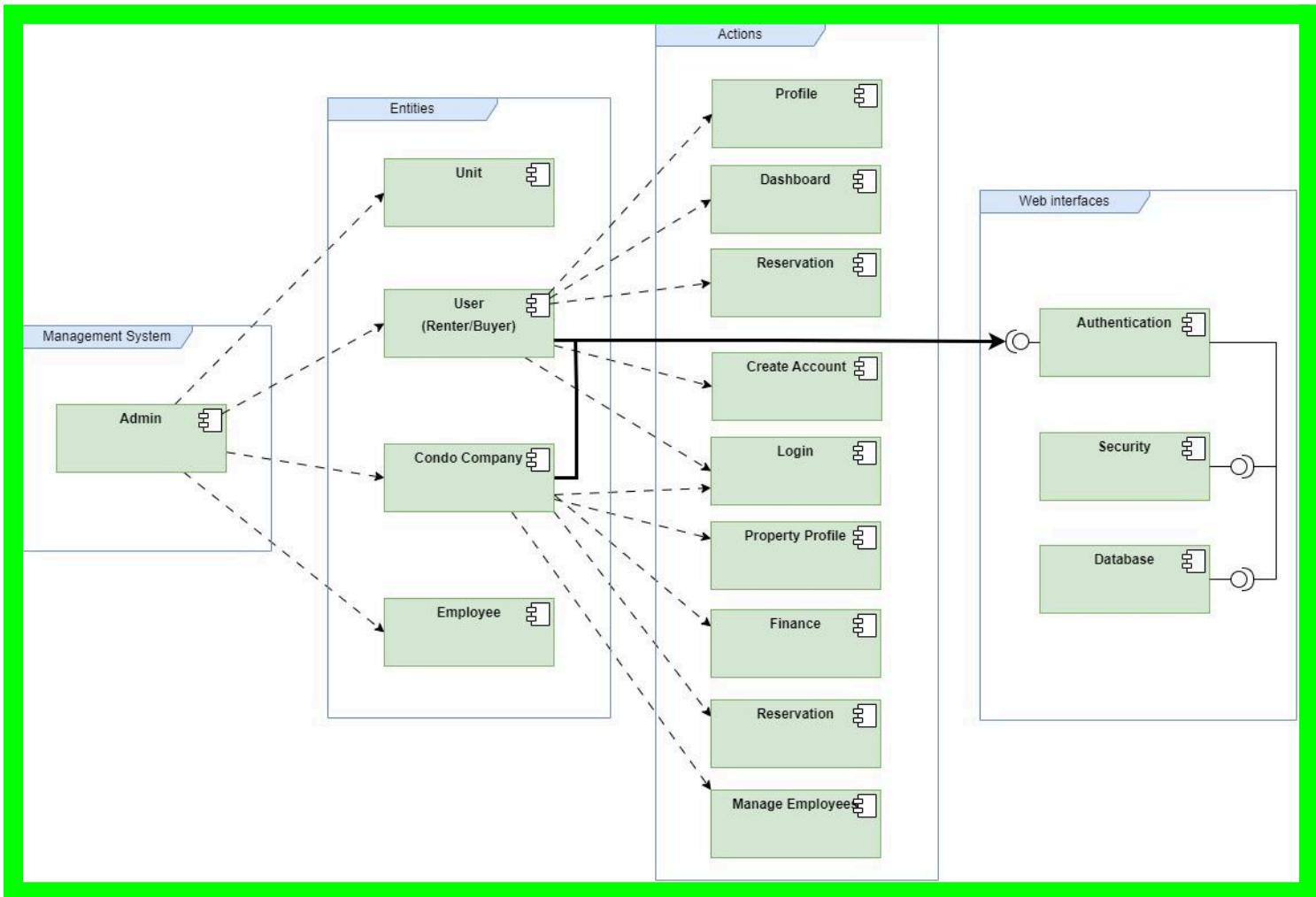
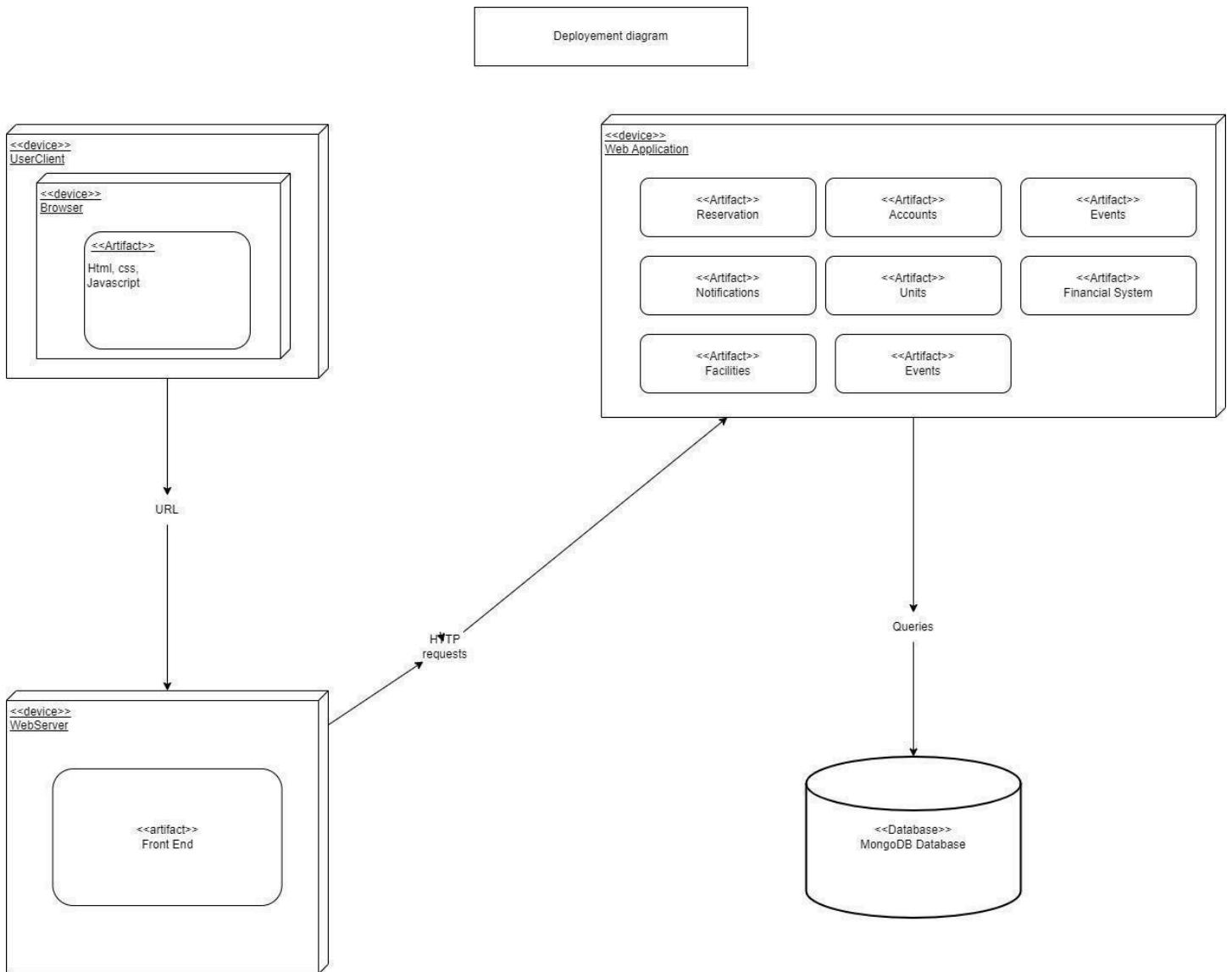


Figure 3: Component Diagram

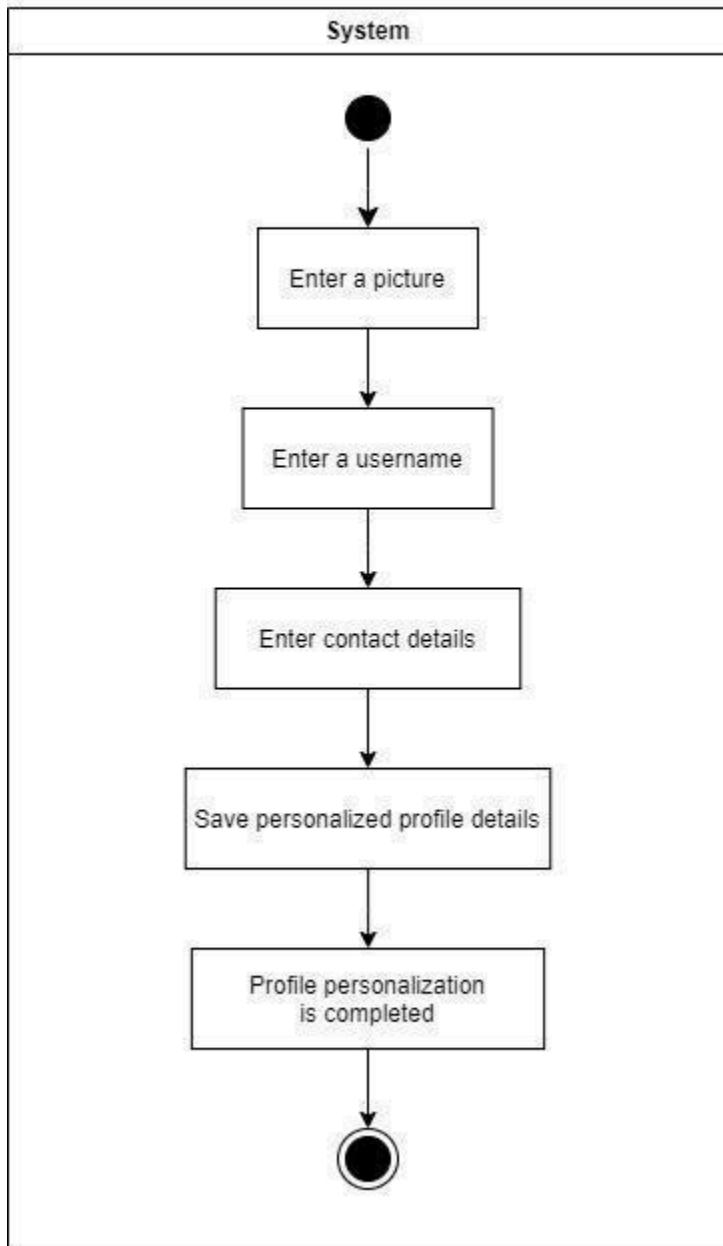
# Deployment Diagram



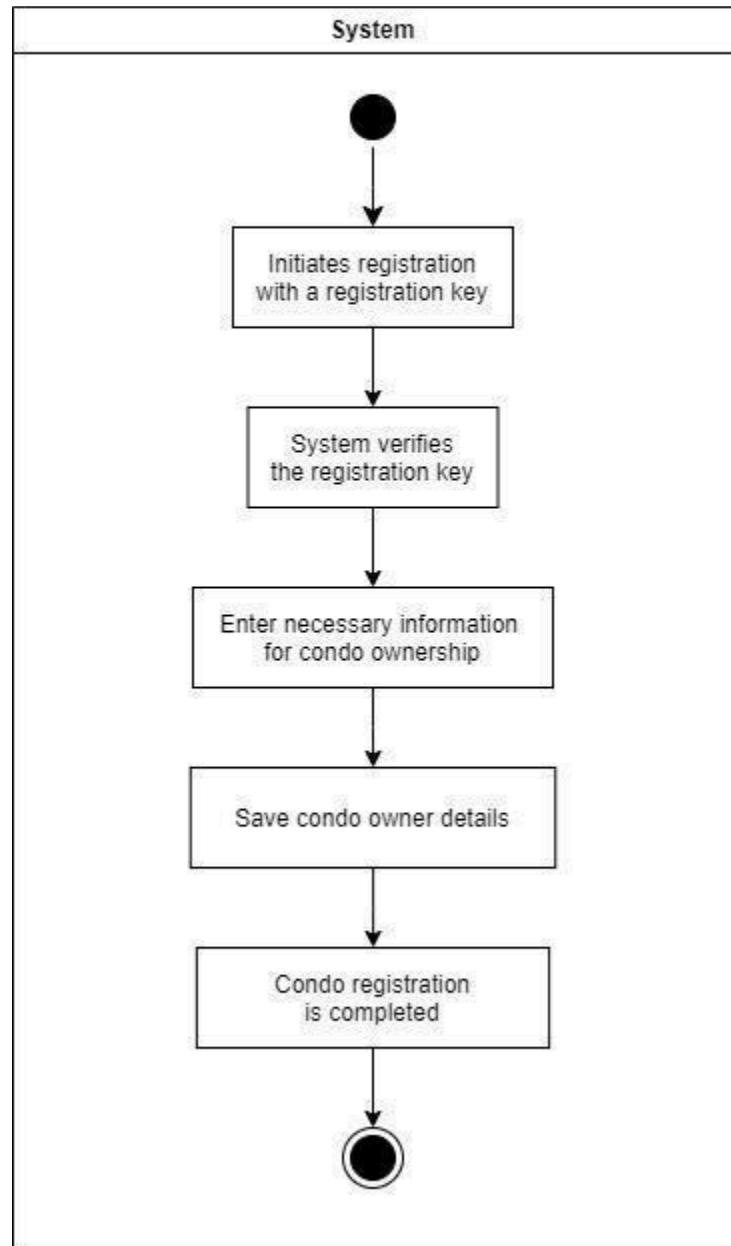
**Figure 4: Deployment Diagram**

## Activity Diagrams

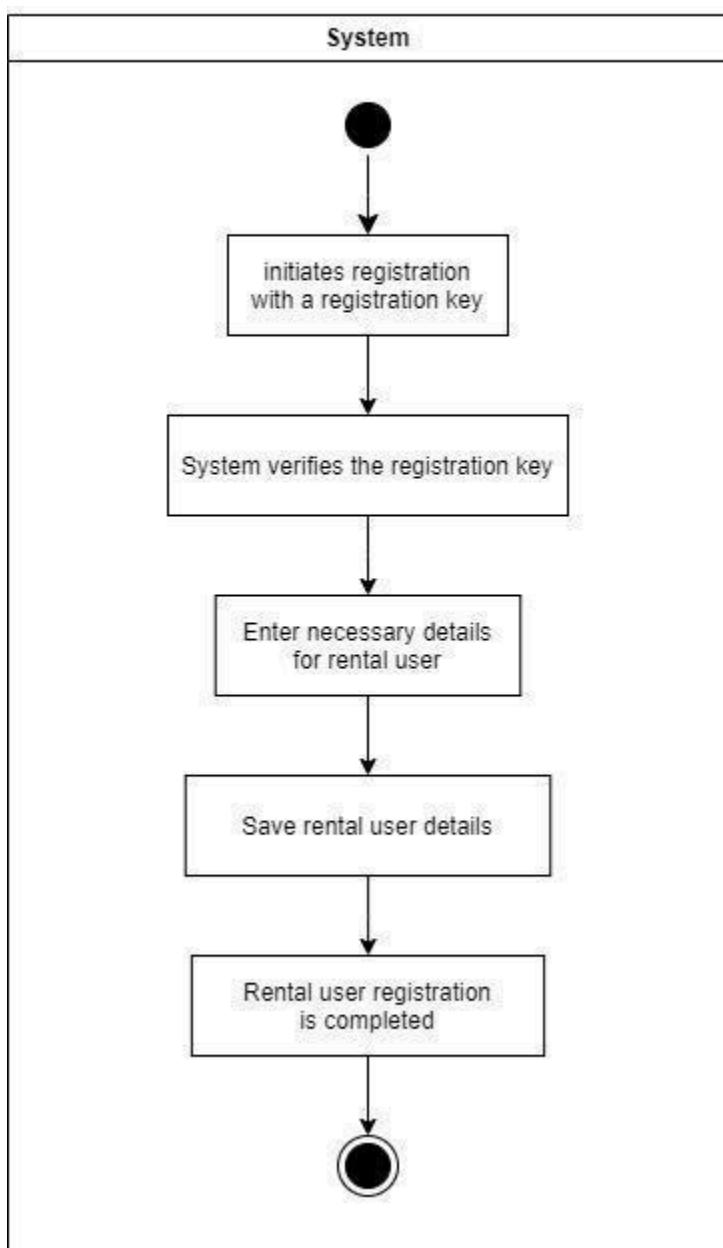
ID: 001



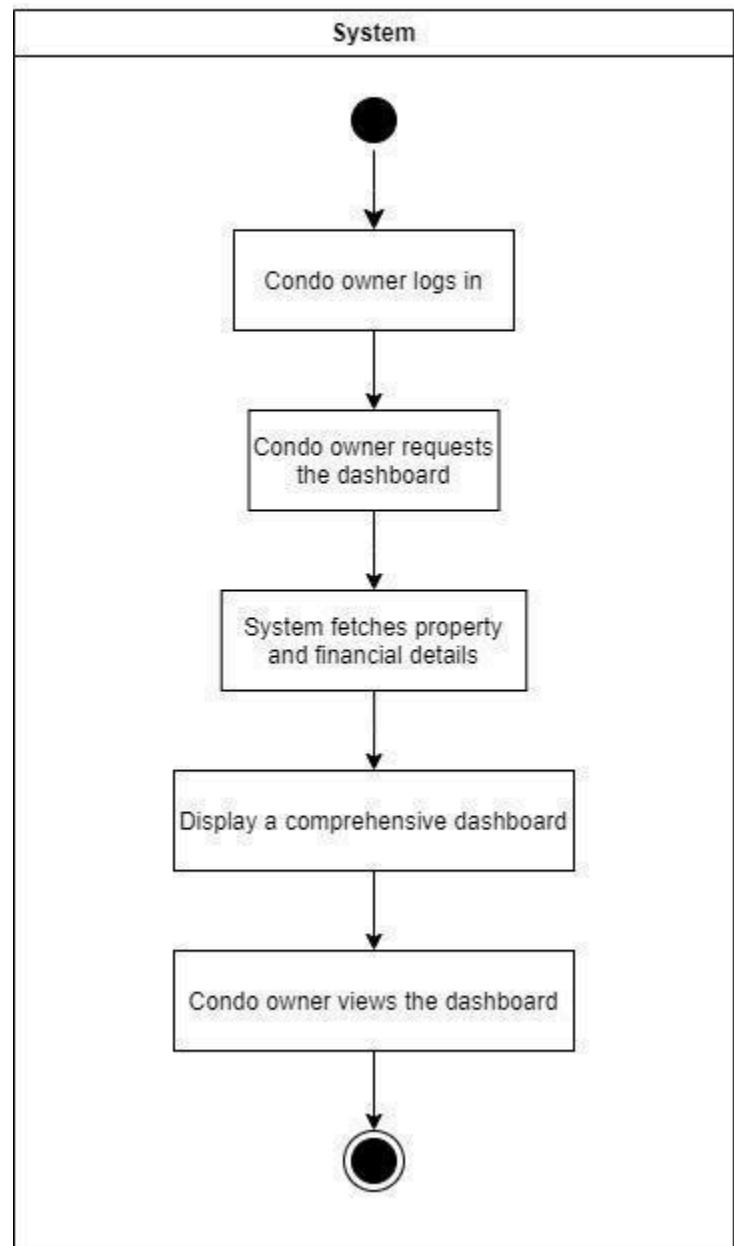
ID: 002



ID: 003

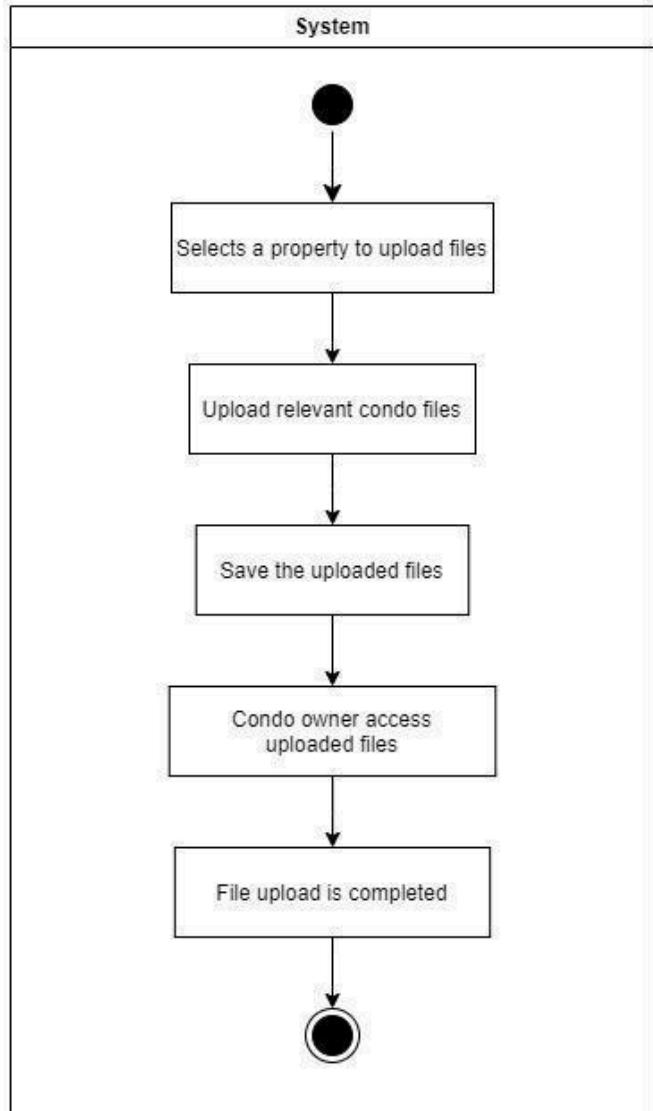
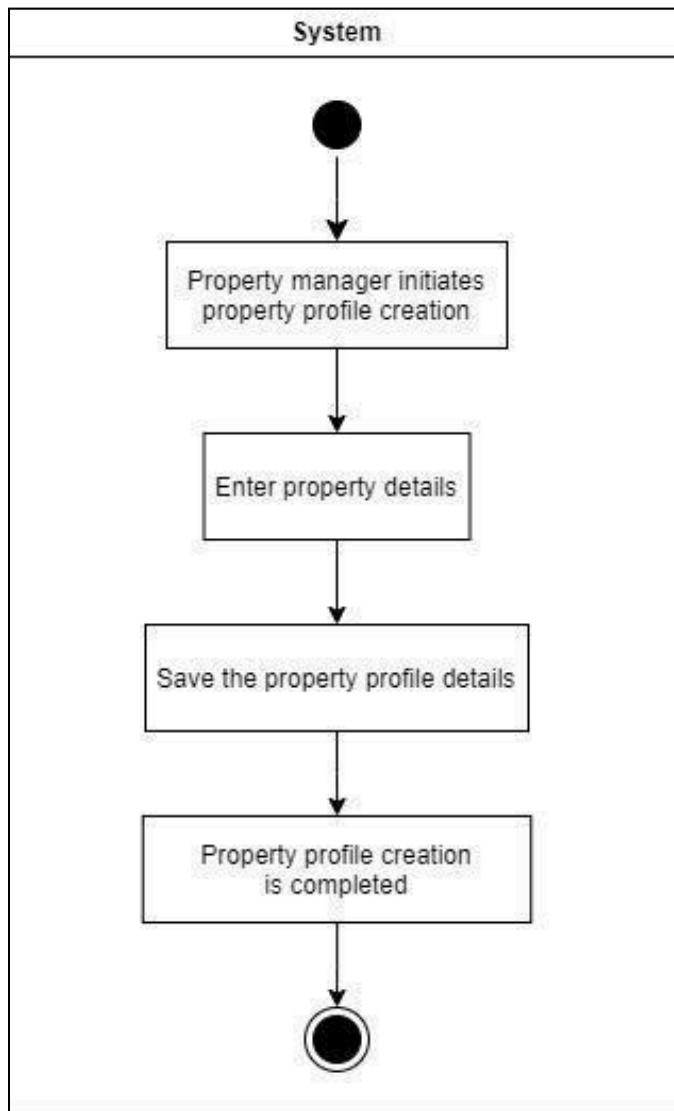


ID: 004



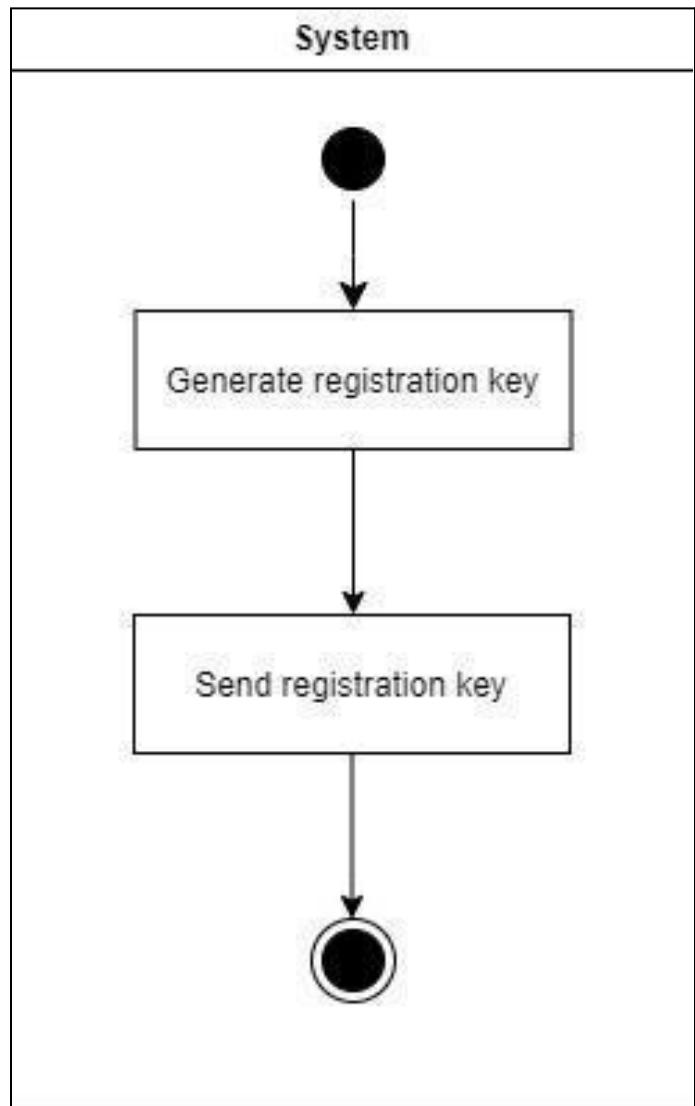
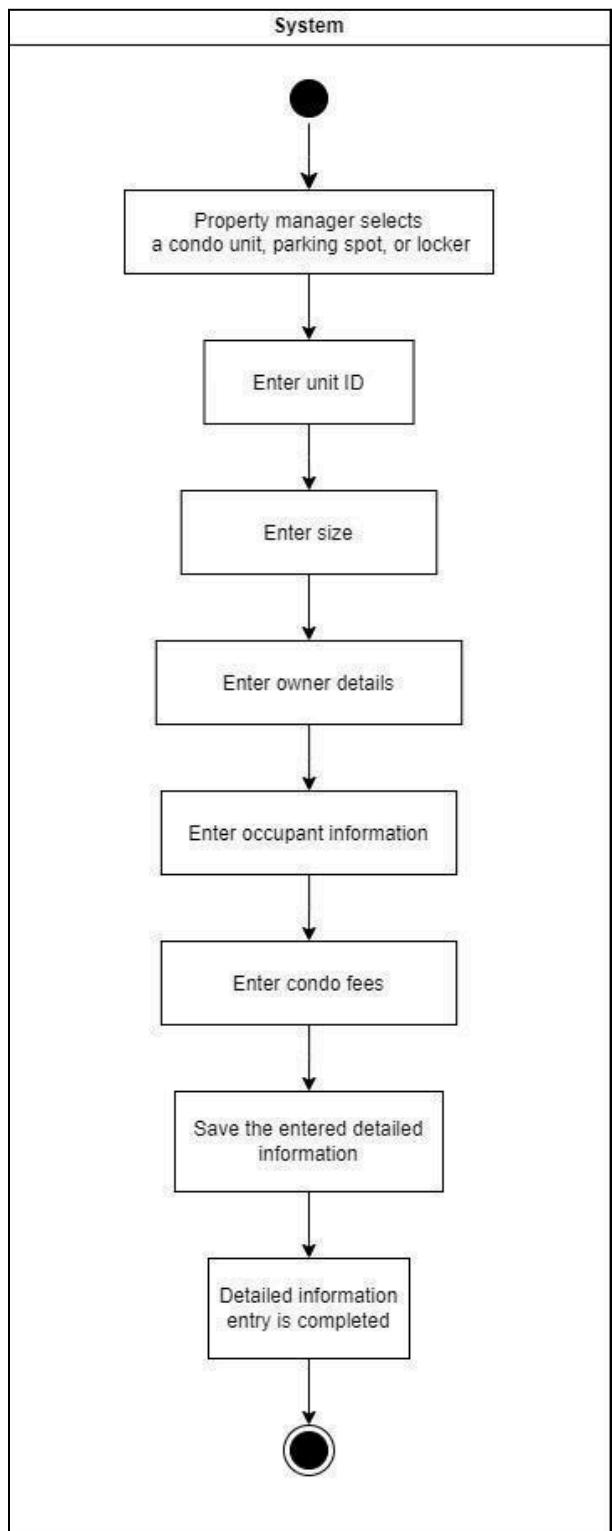
ID: 005

ID: 006

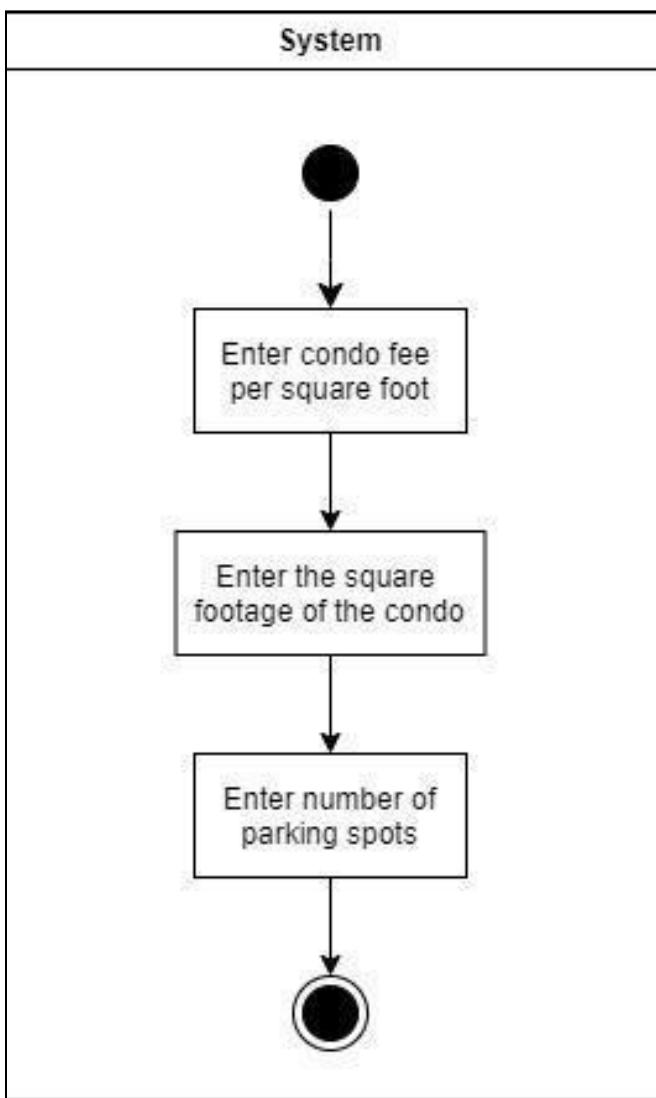


ID: 007

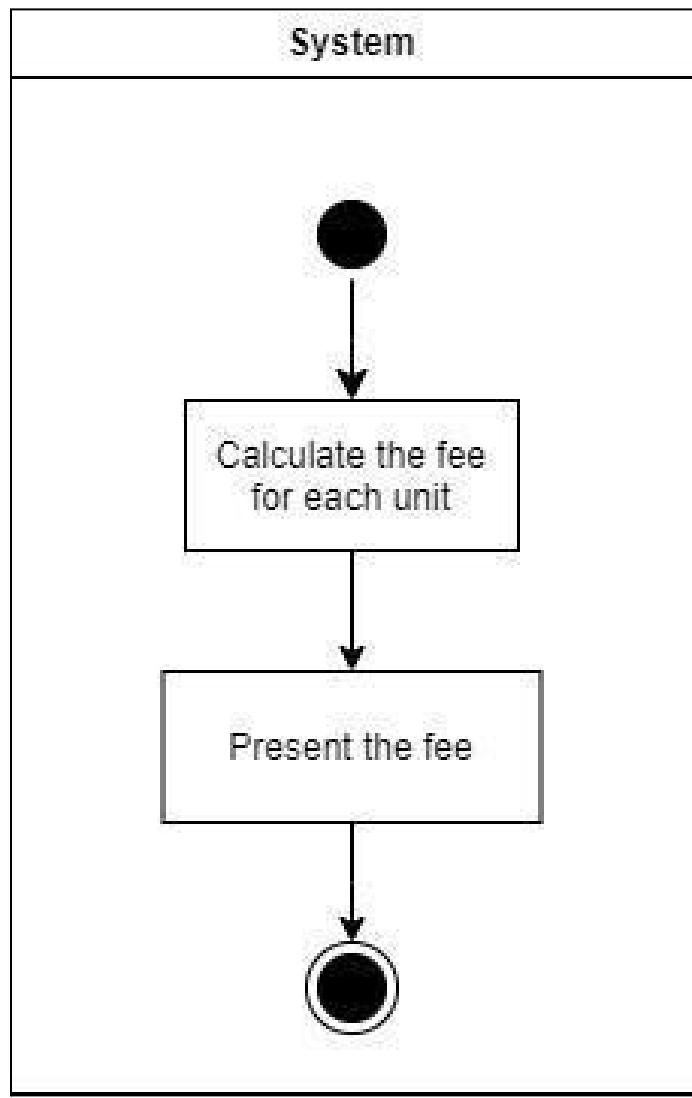
ID: 008



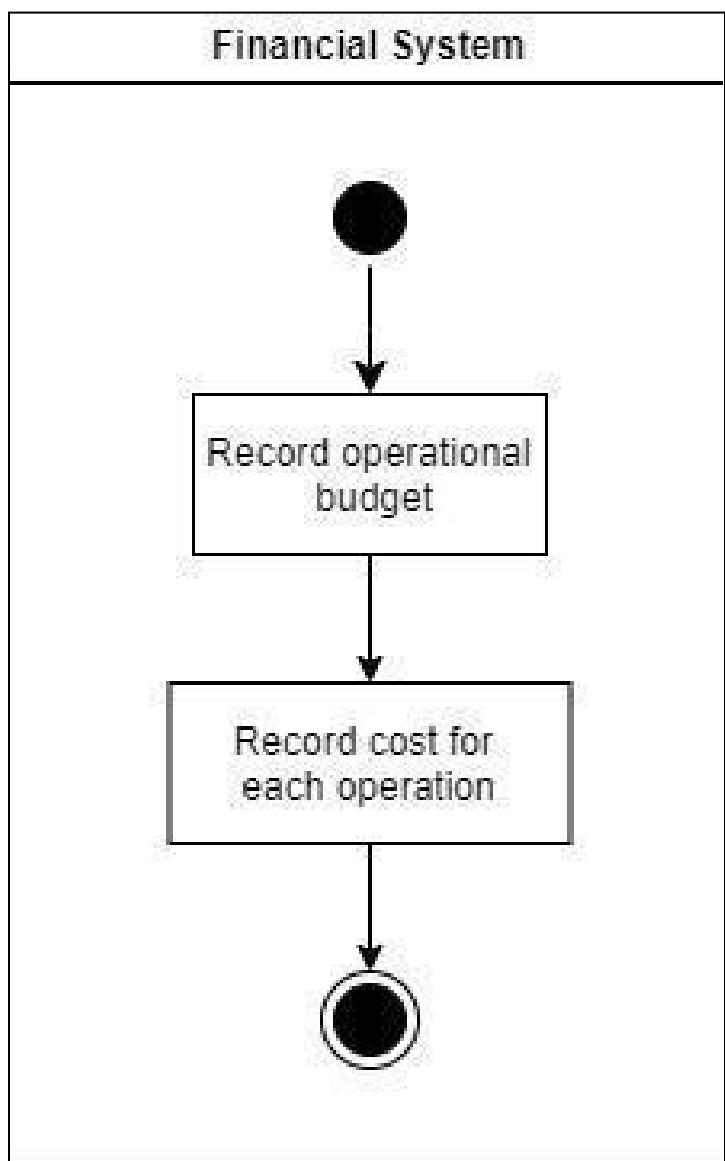
ID: 009



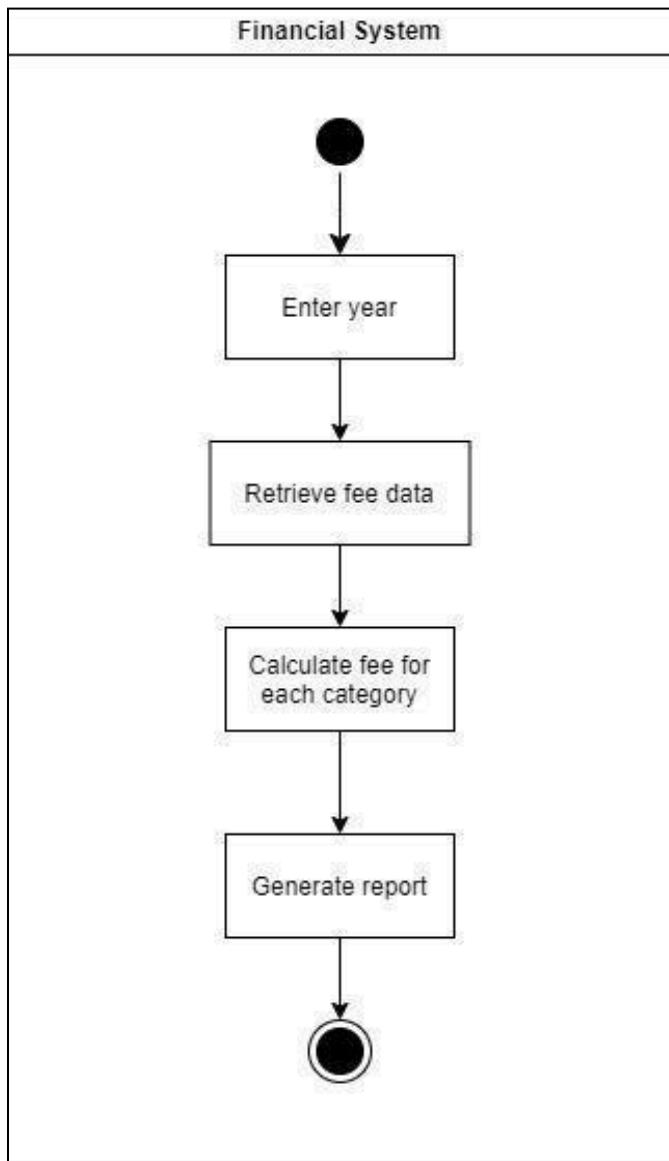
ID: 010



ID: 011

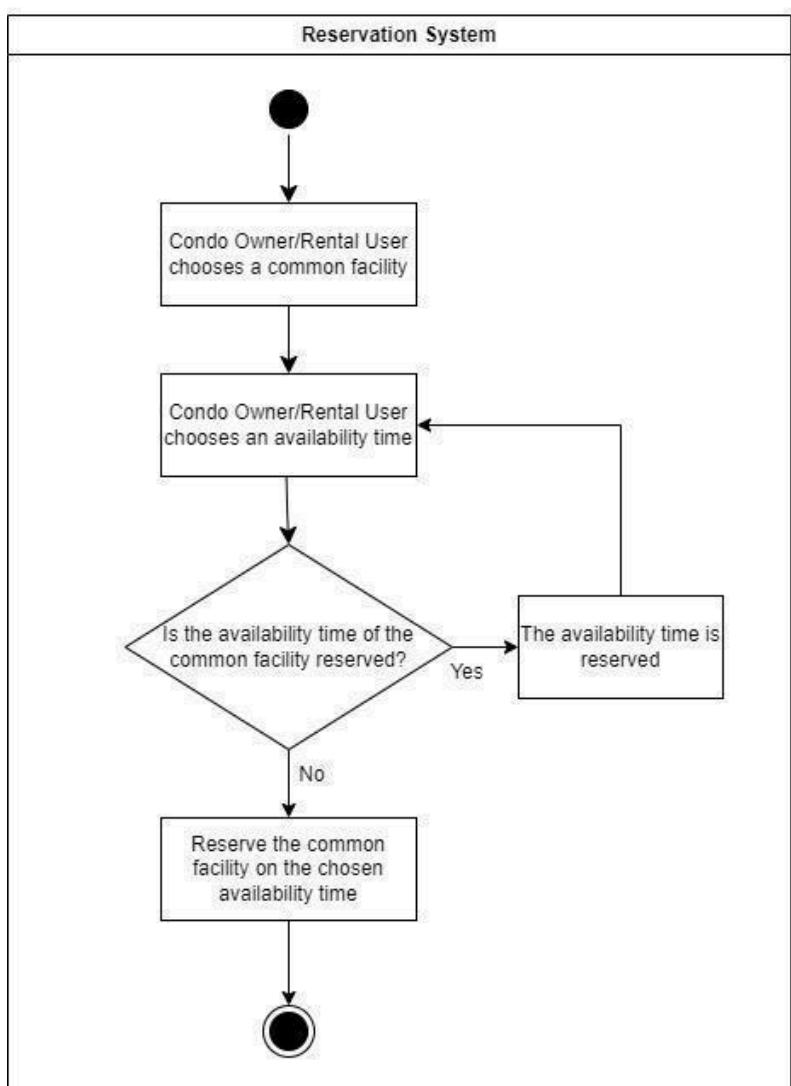
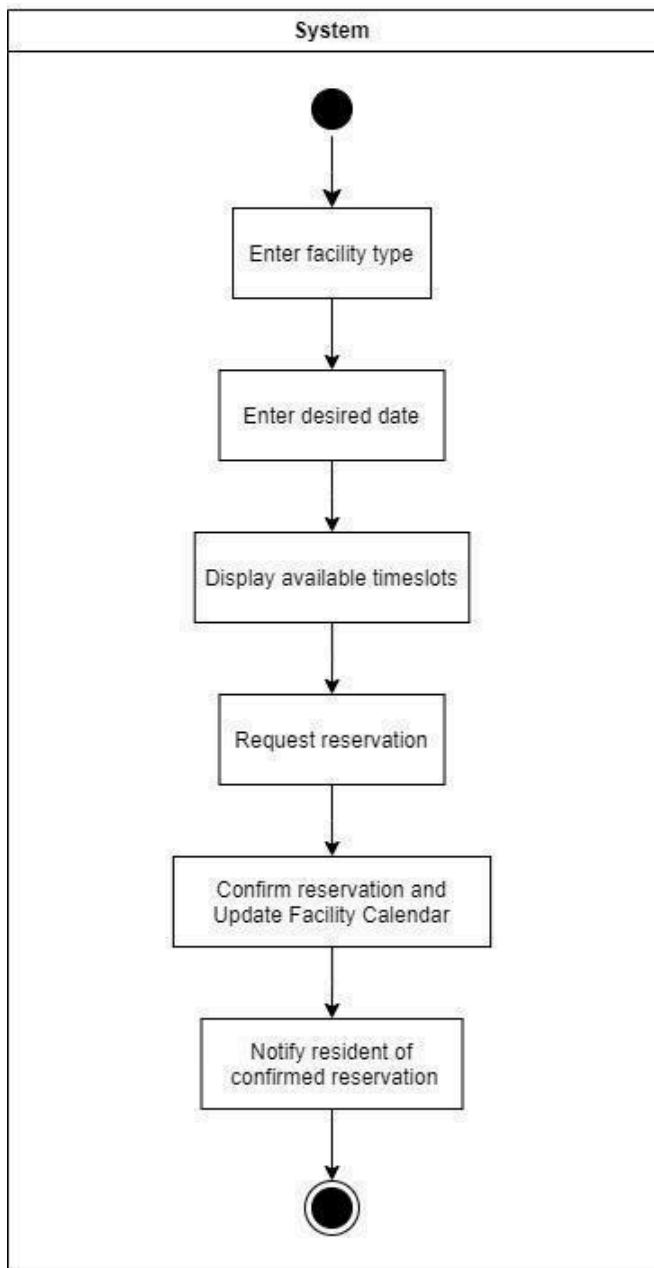


ID: 012

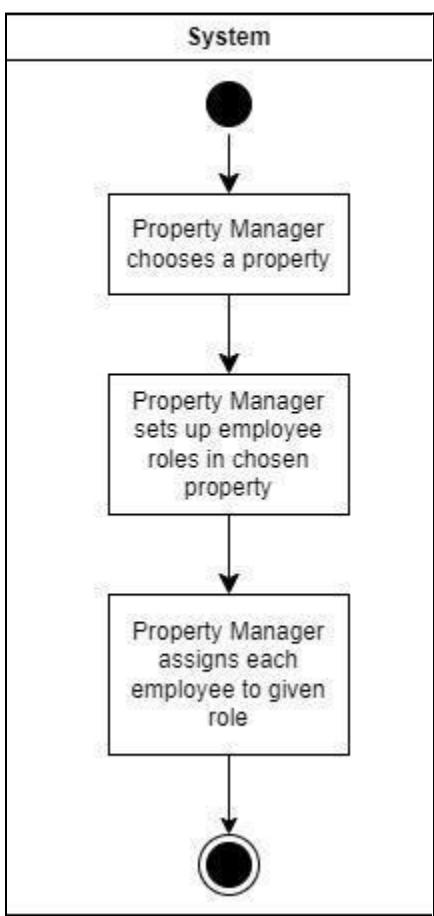


ID: 013-014

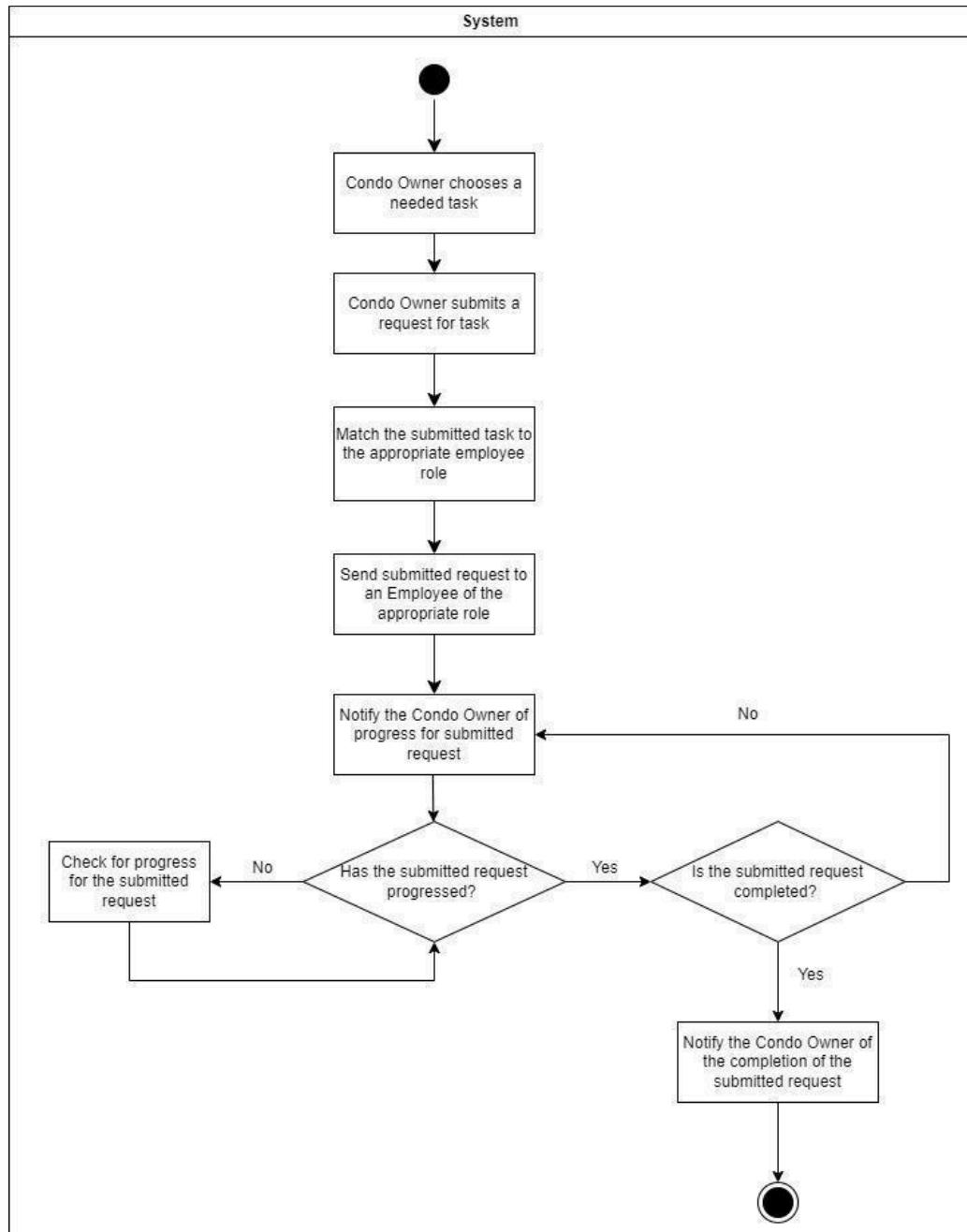
ID: 015-016



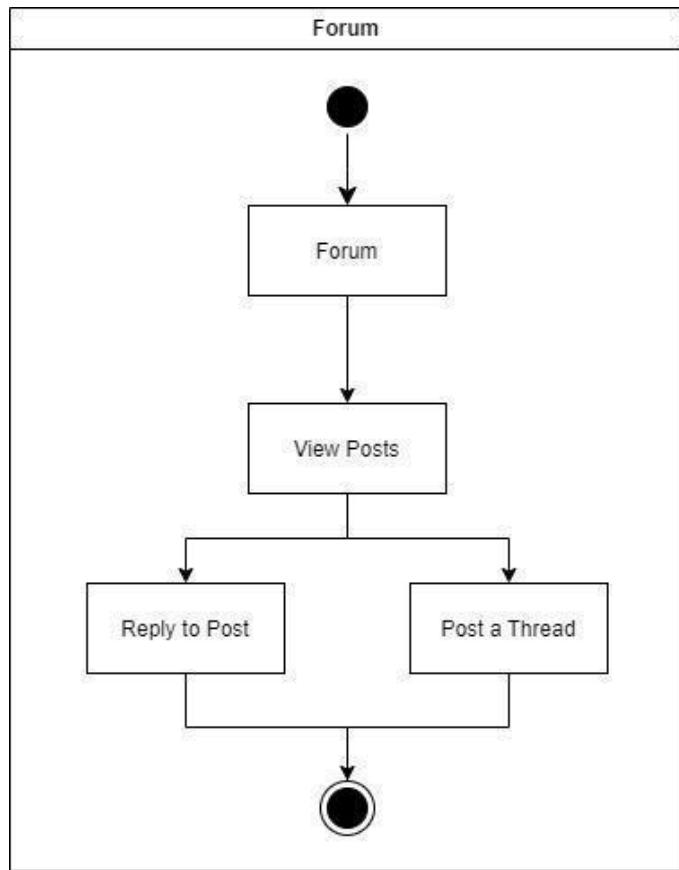
ID: 017



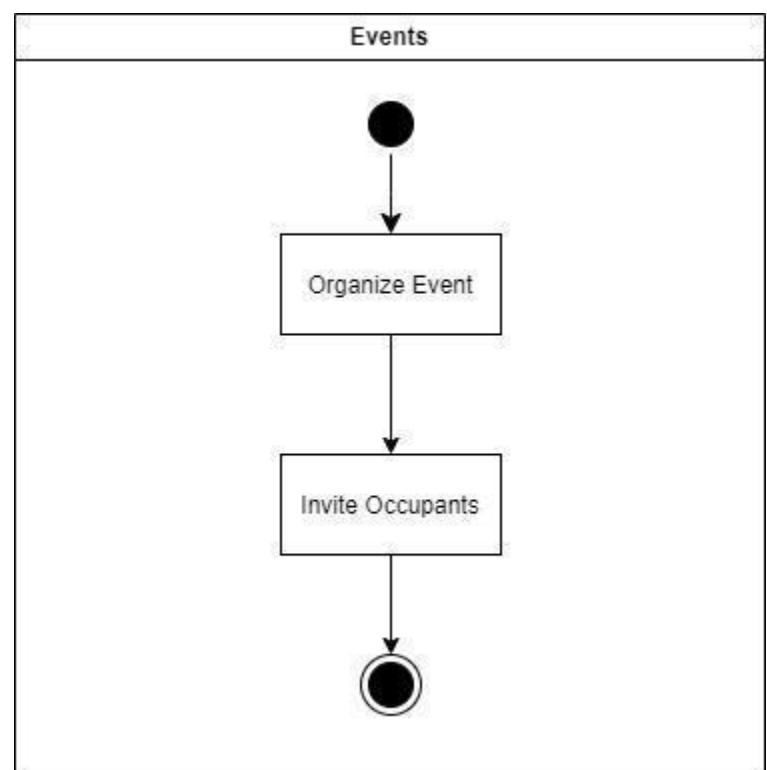
ID: 018 - 019 - 020



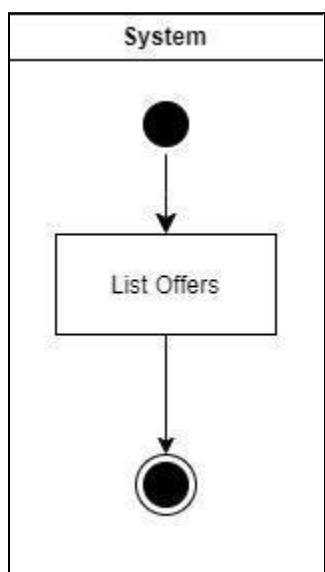
ID: 021



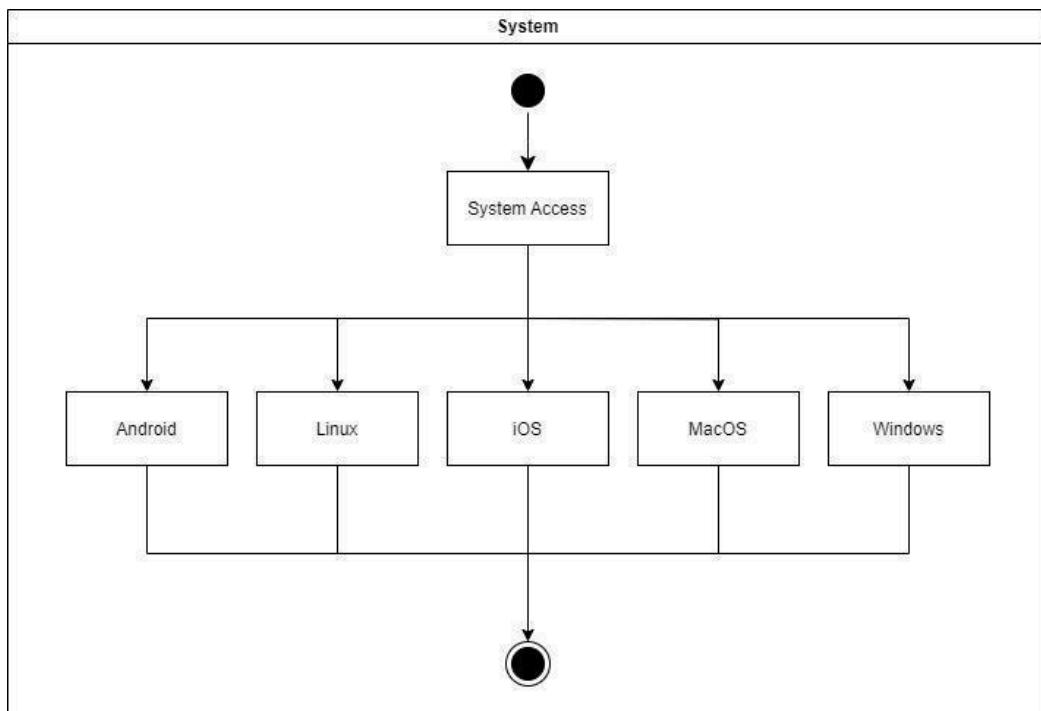
ID: 022



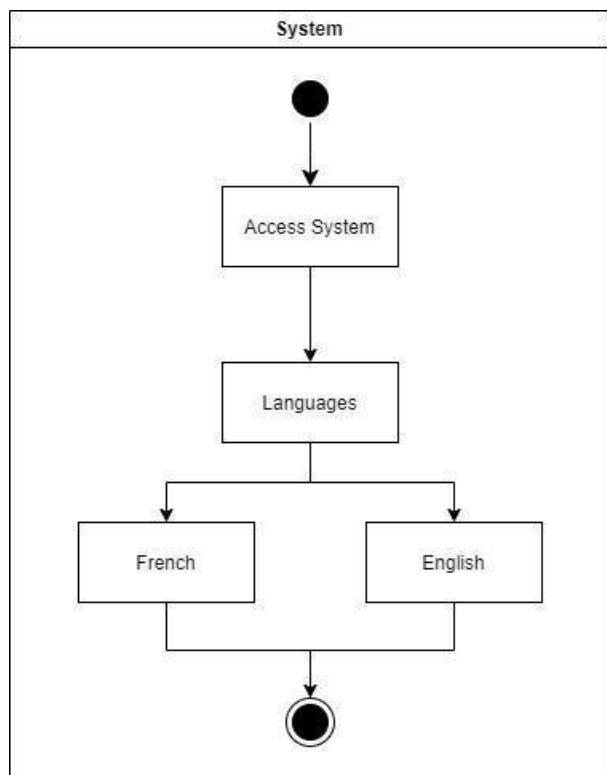
ID: 023



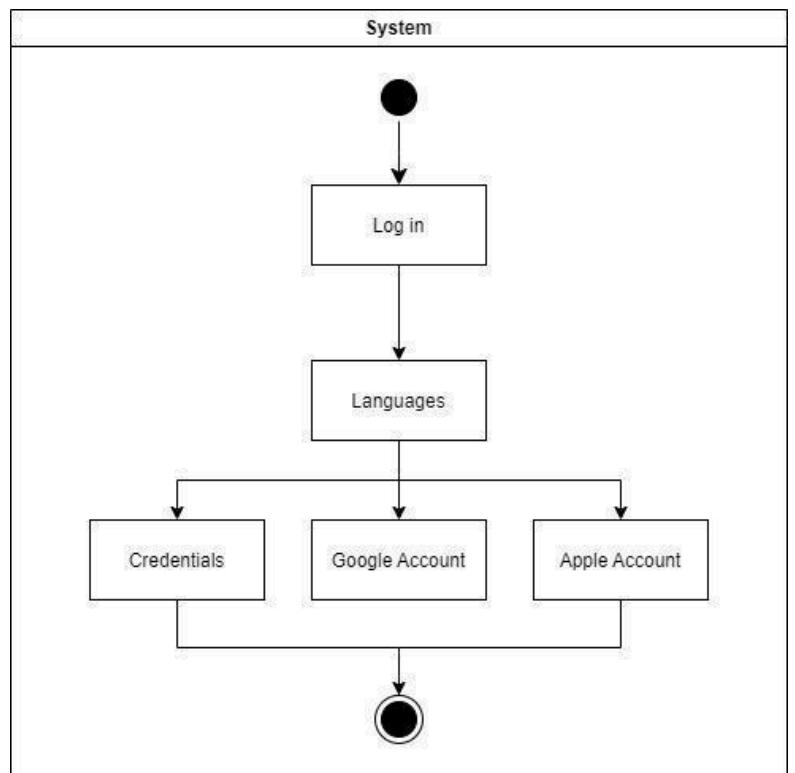
ID: 024



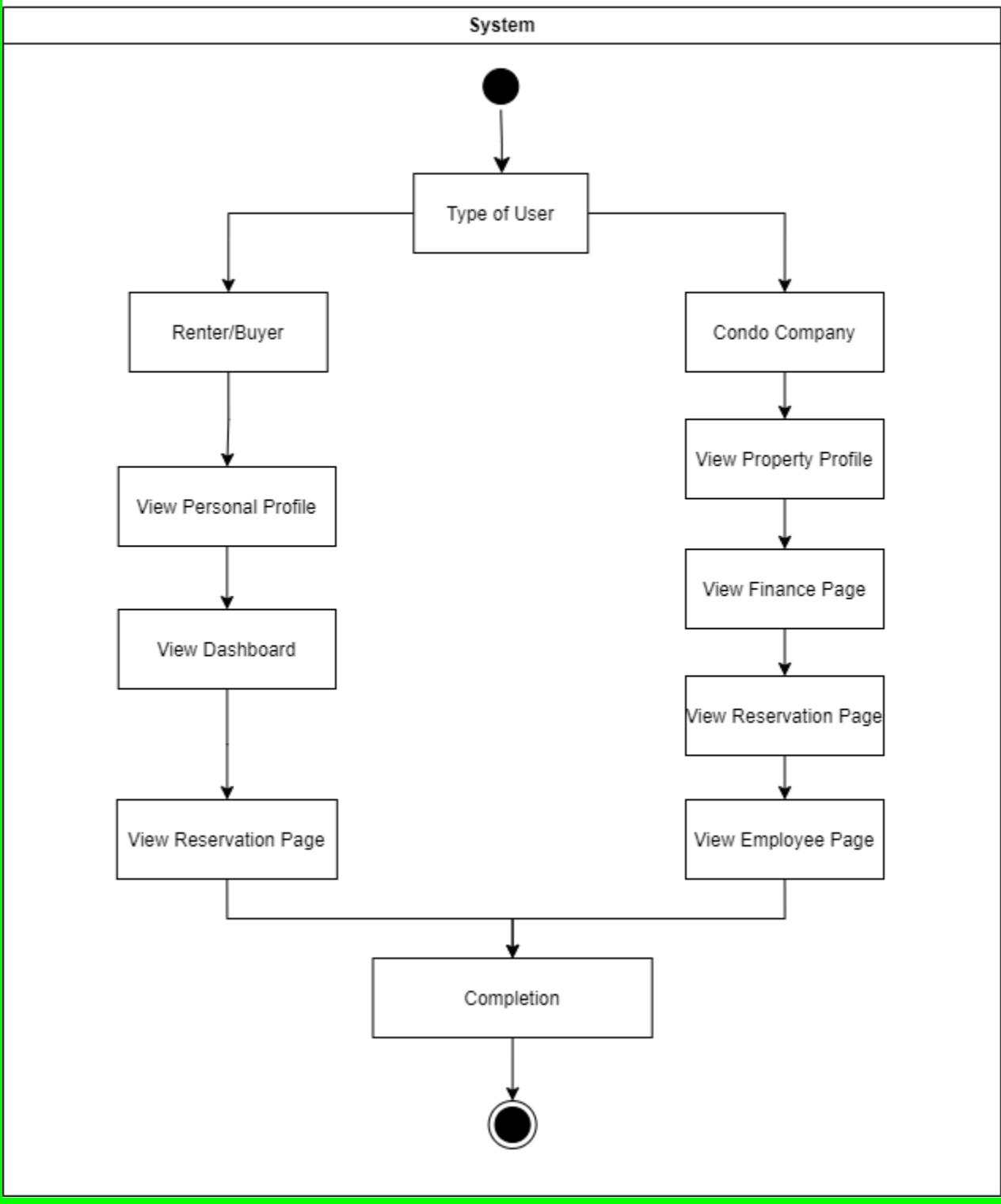
ID: 025



ID: 026

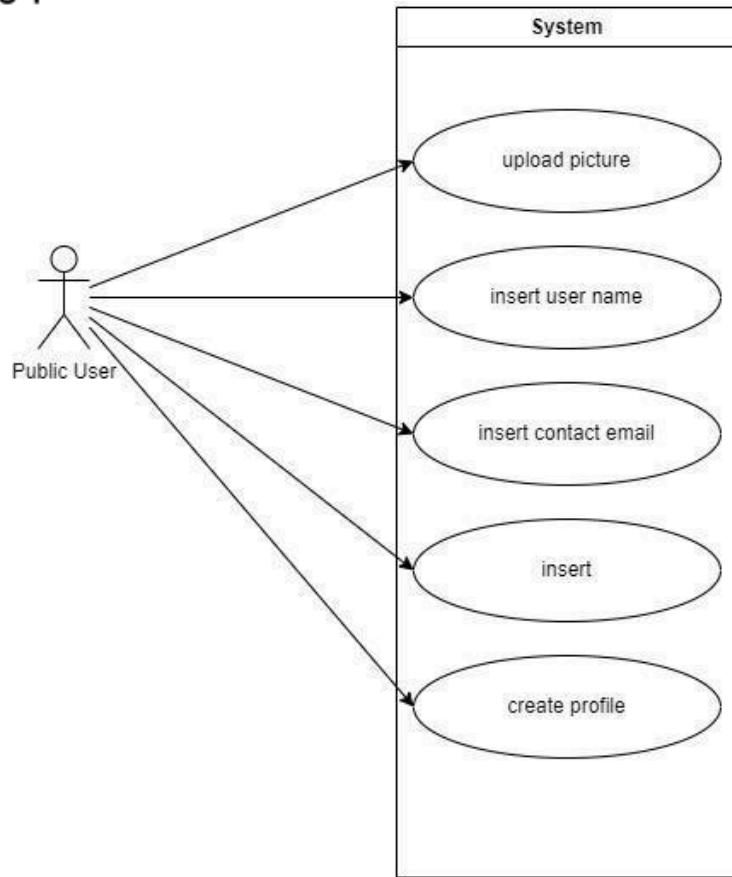


ID: 027

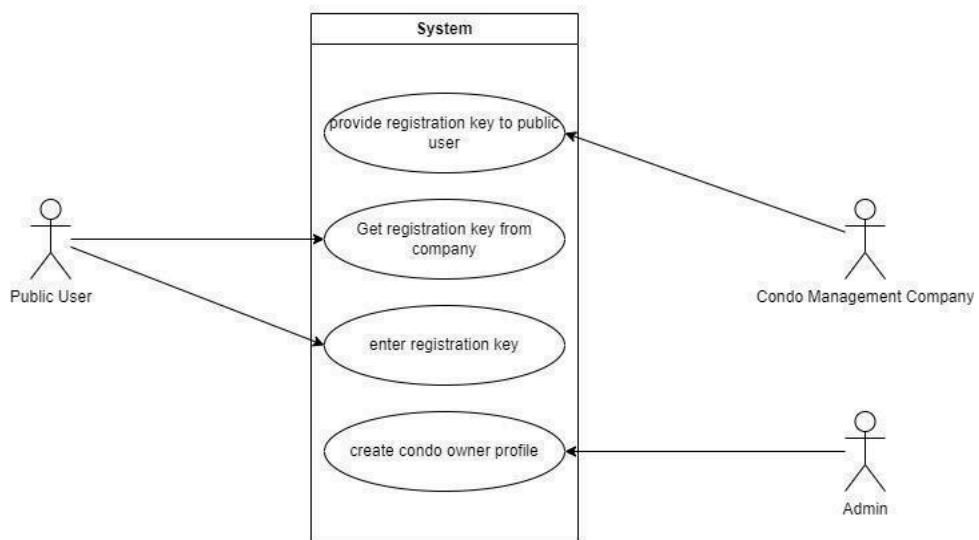


# Use Case Diagrams

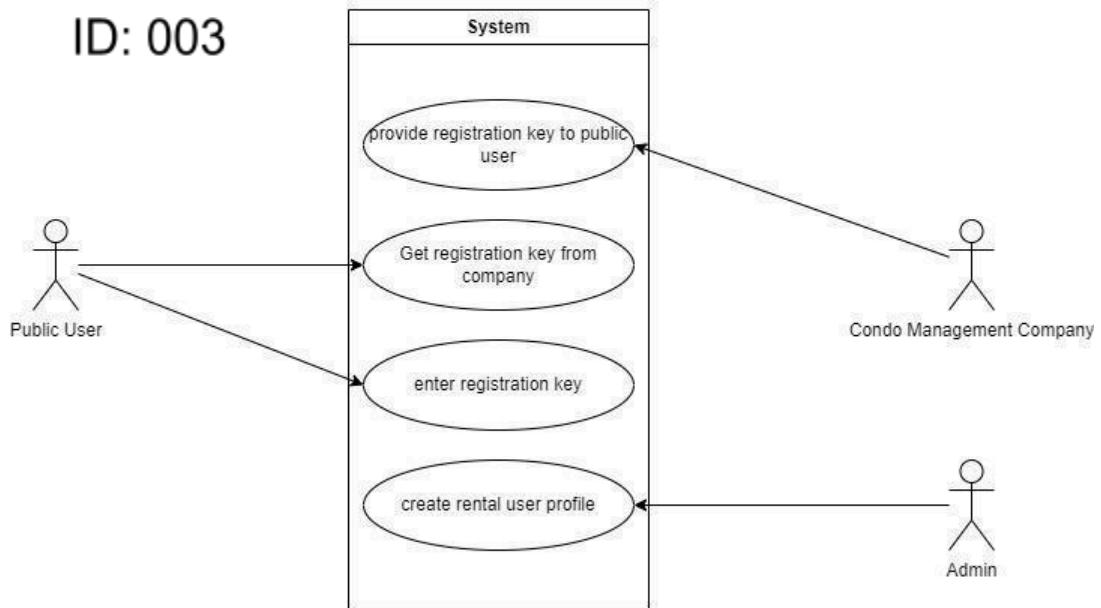
ID: 001



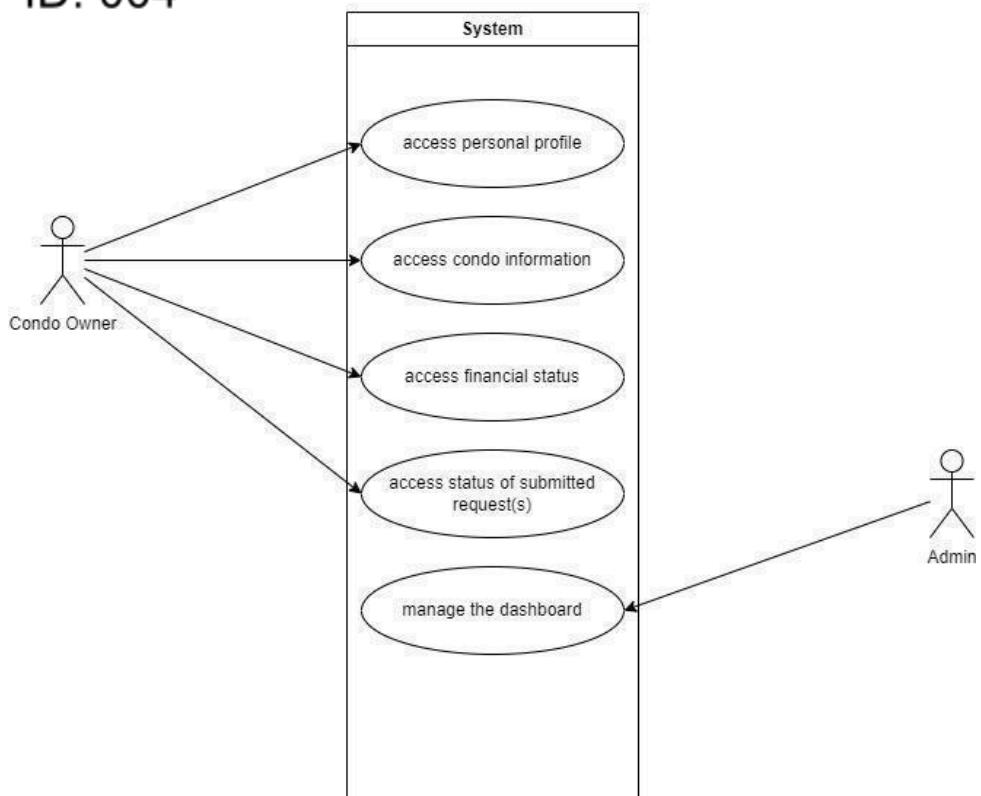
ID: 002



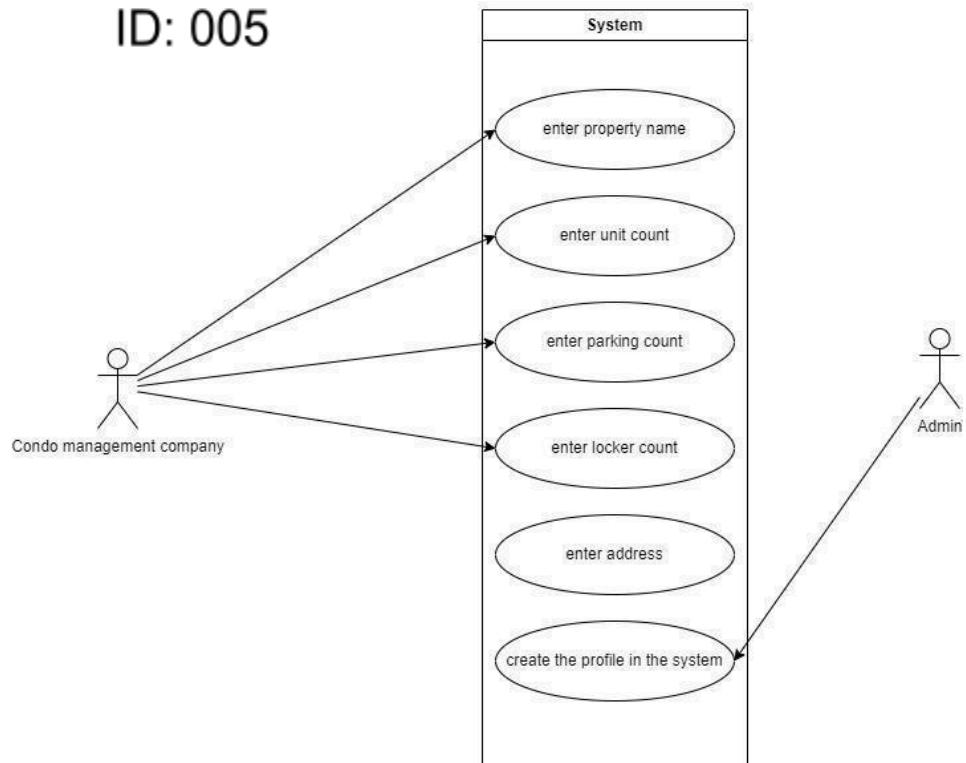
ID: 003



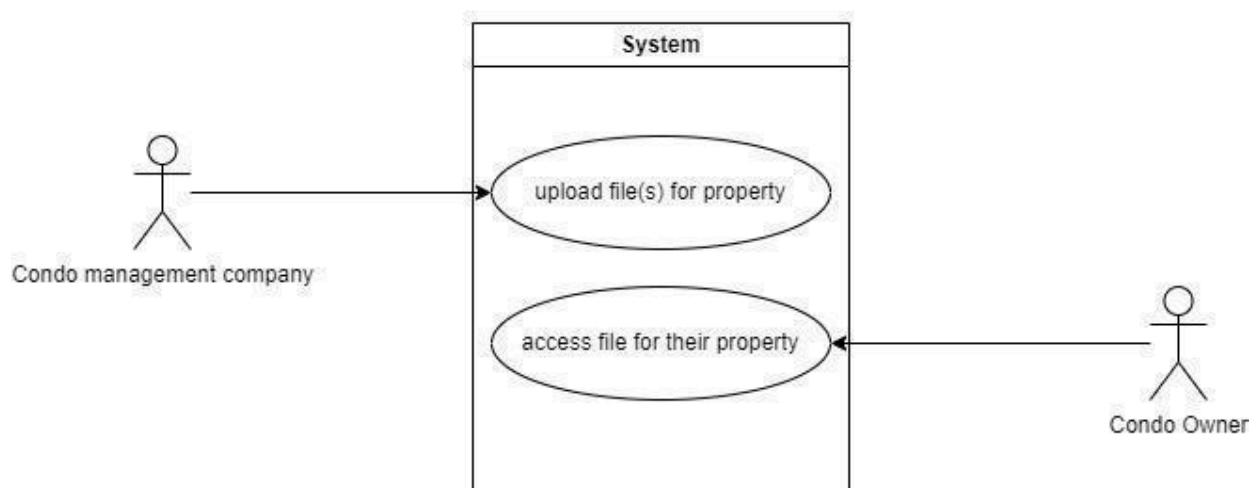
ID: 004



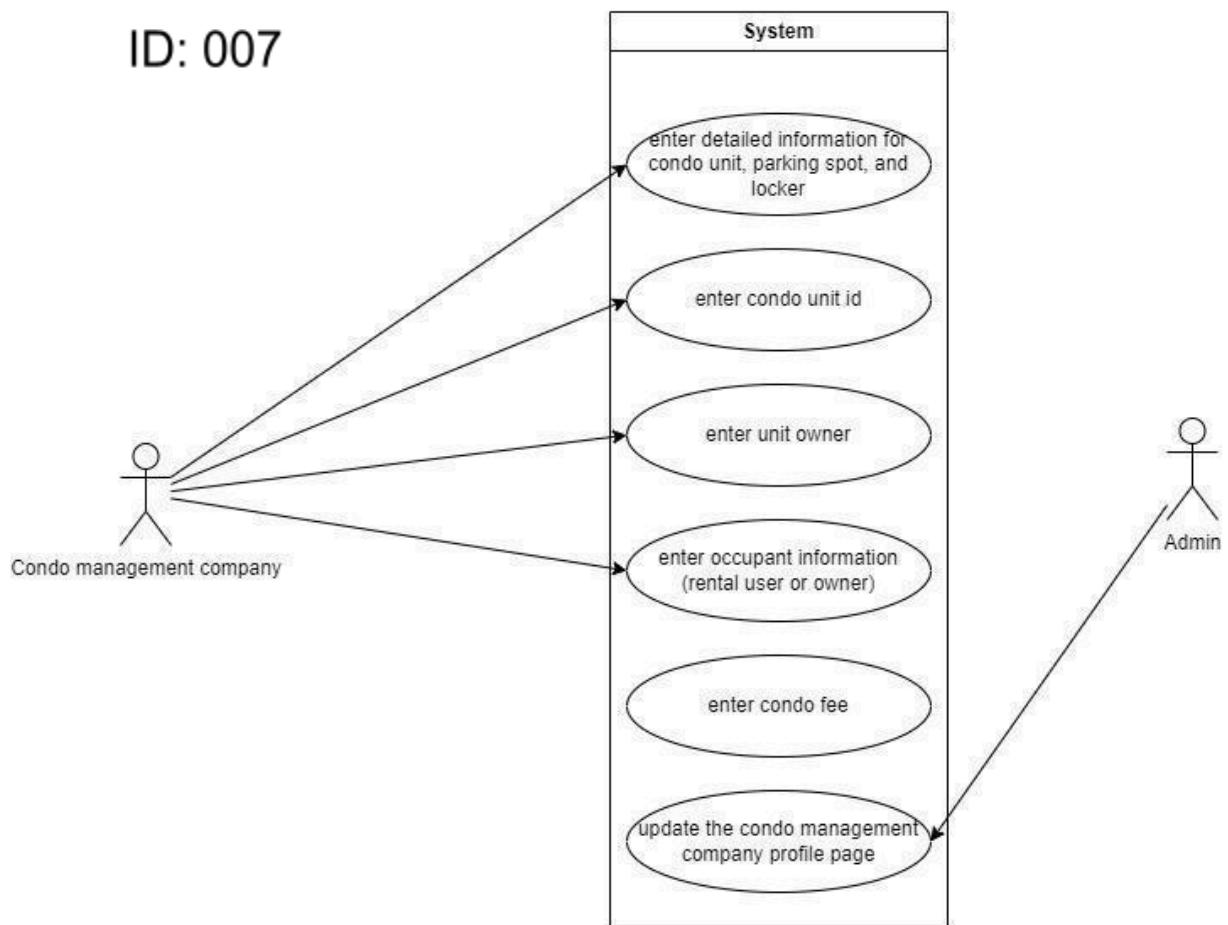
ID: 005



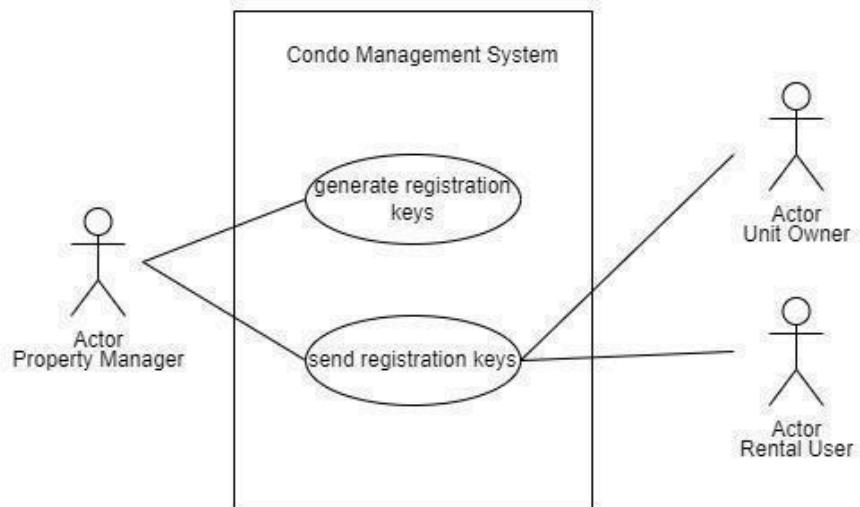
ID: 006



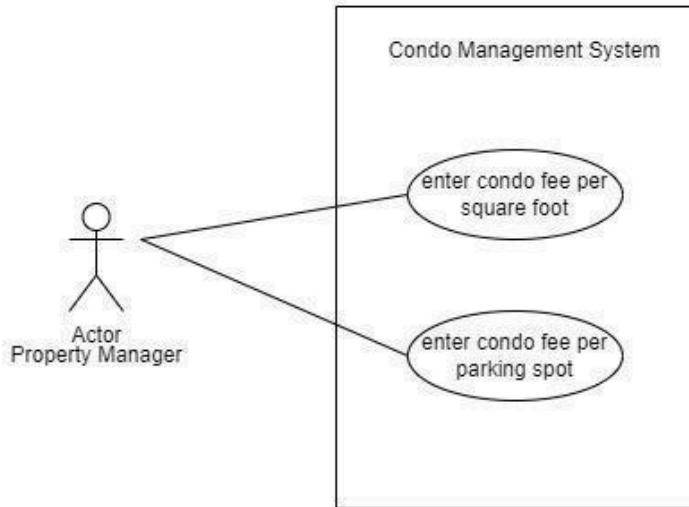
ID: 007



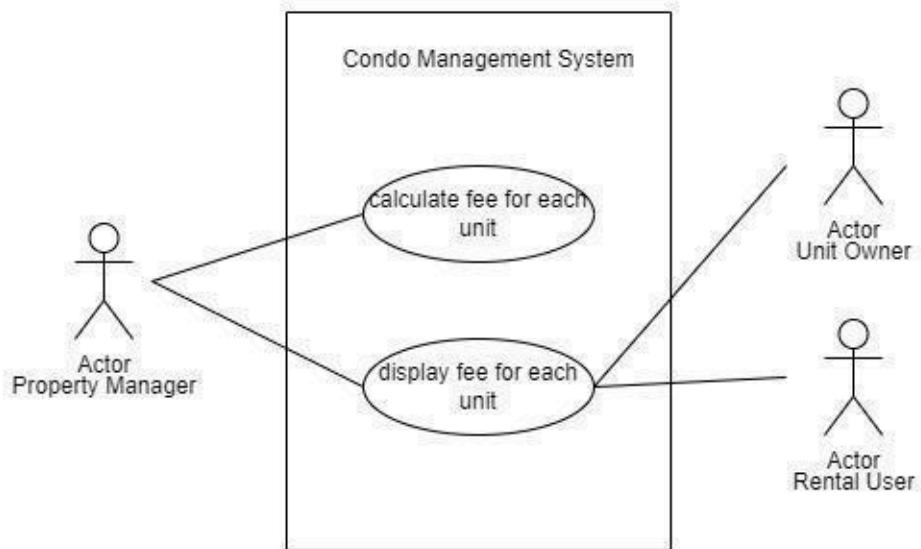
ID: 008



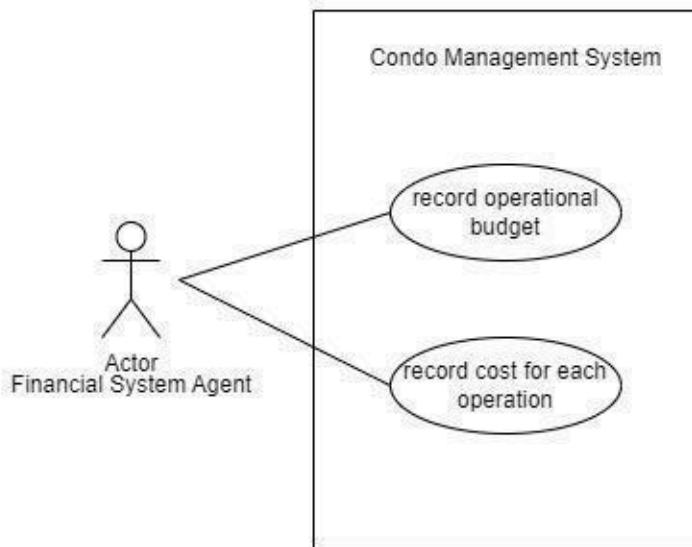
ID: 009



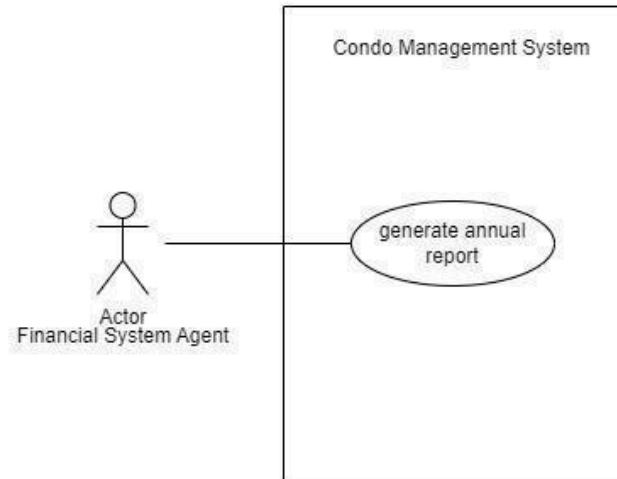
ID: 010



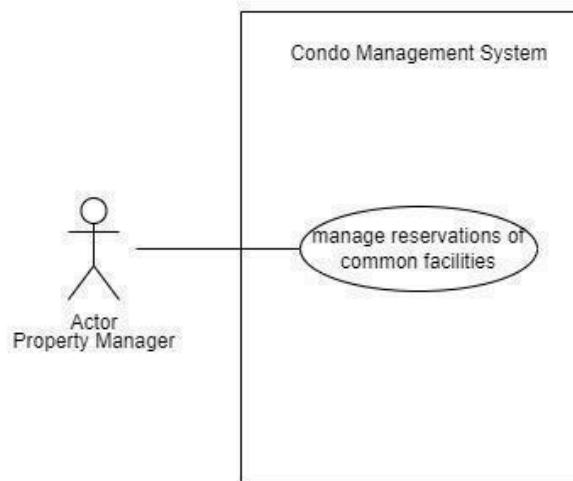
ID: 011



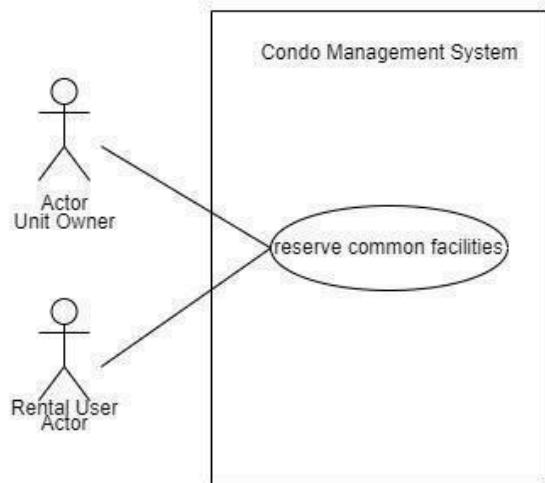
User Story #12



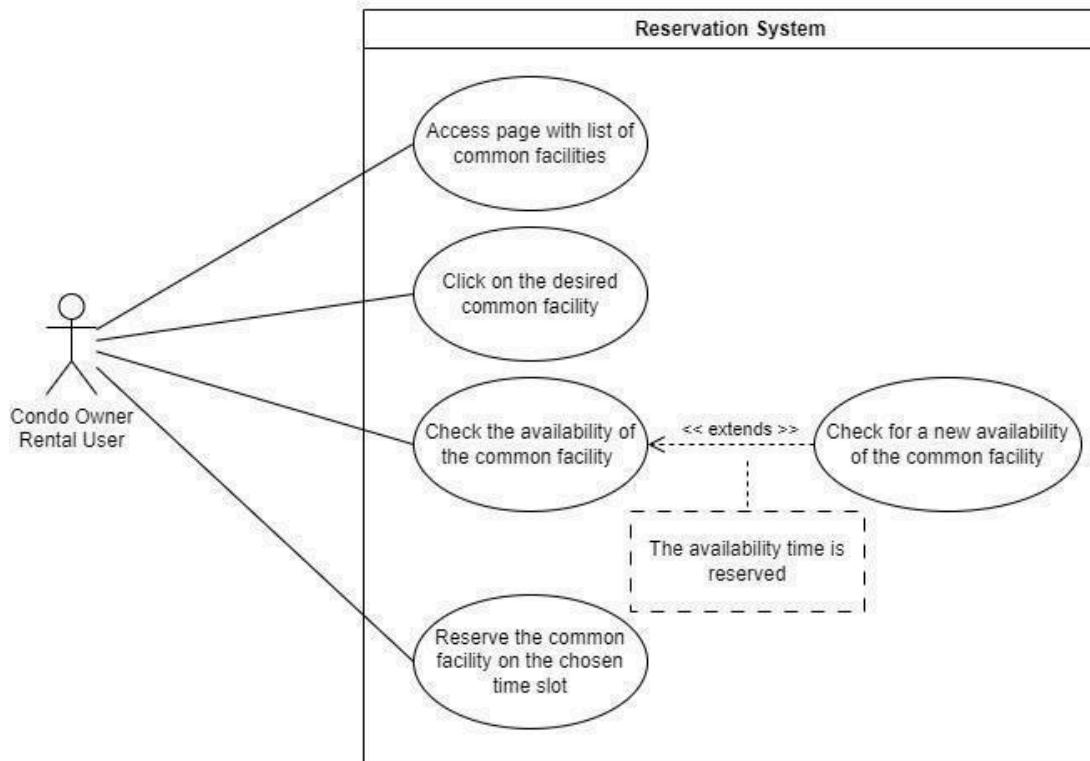
User Story #13



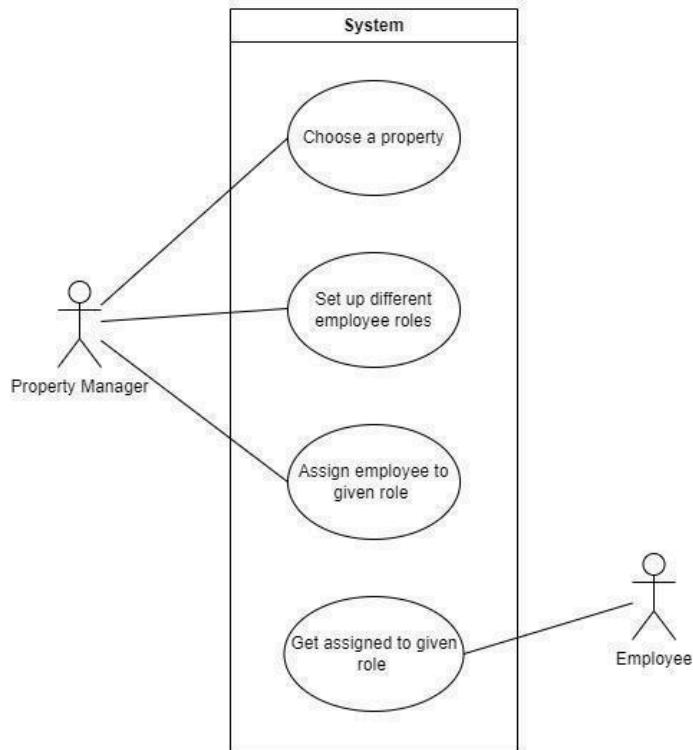
User Story #14



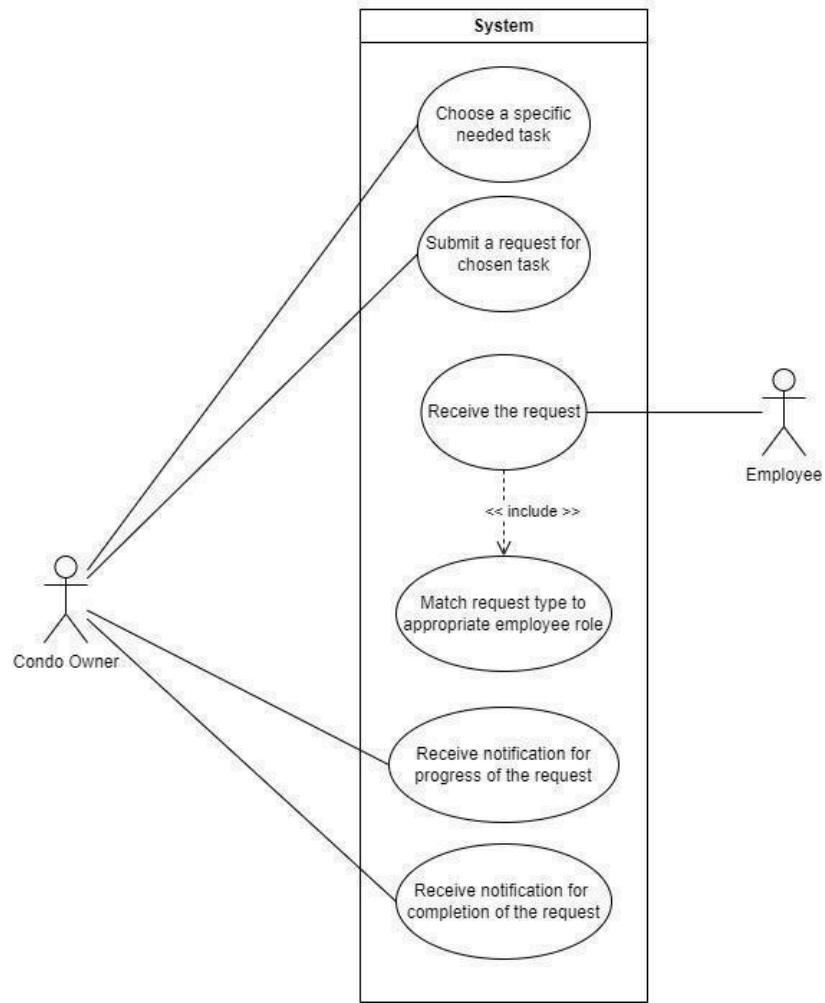
## User Story #15-16



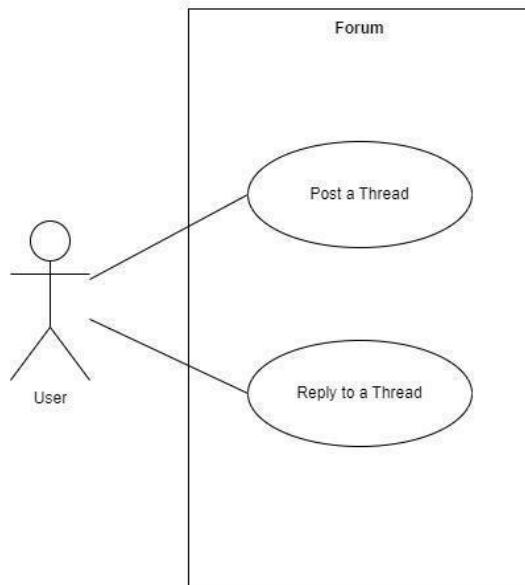
## User Story #17



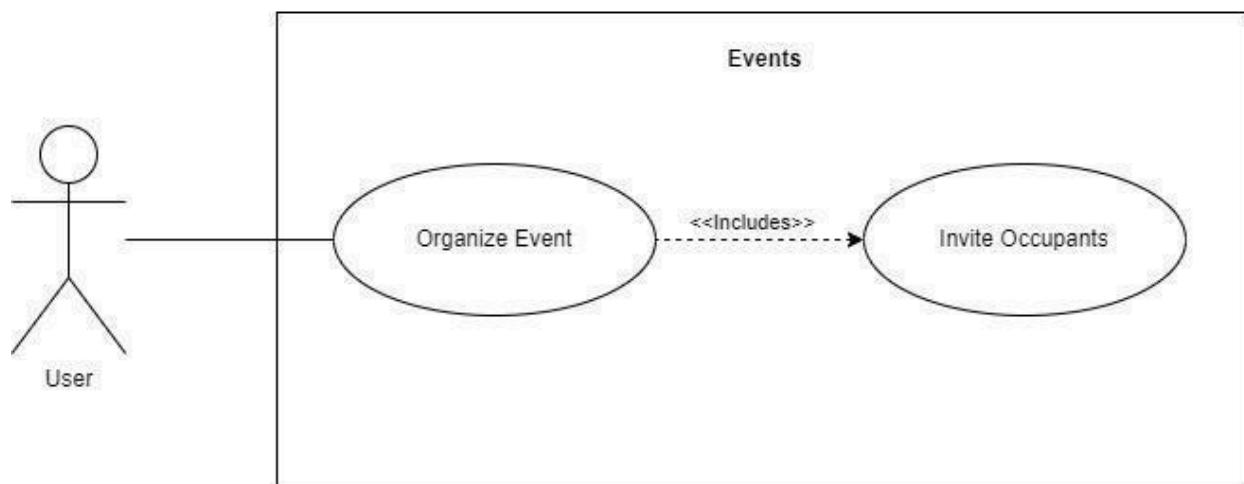
## User Story #18-19-20



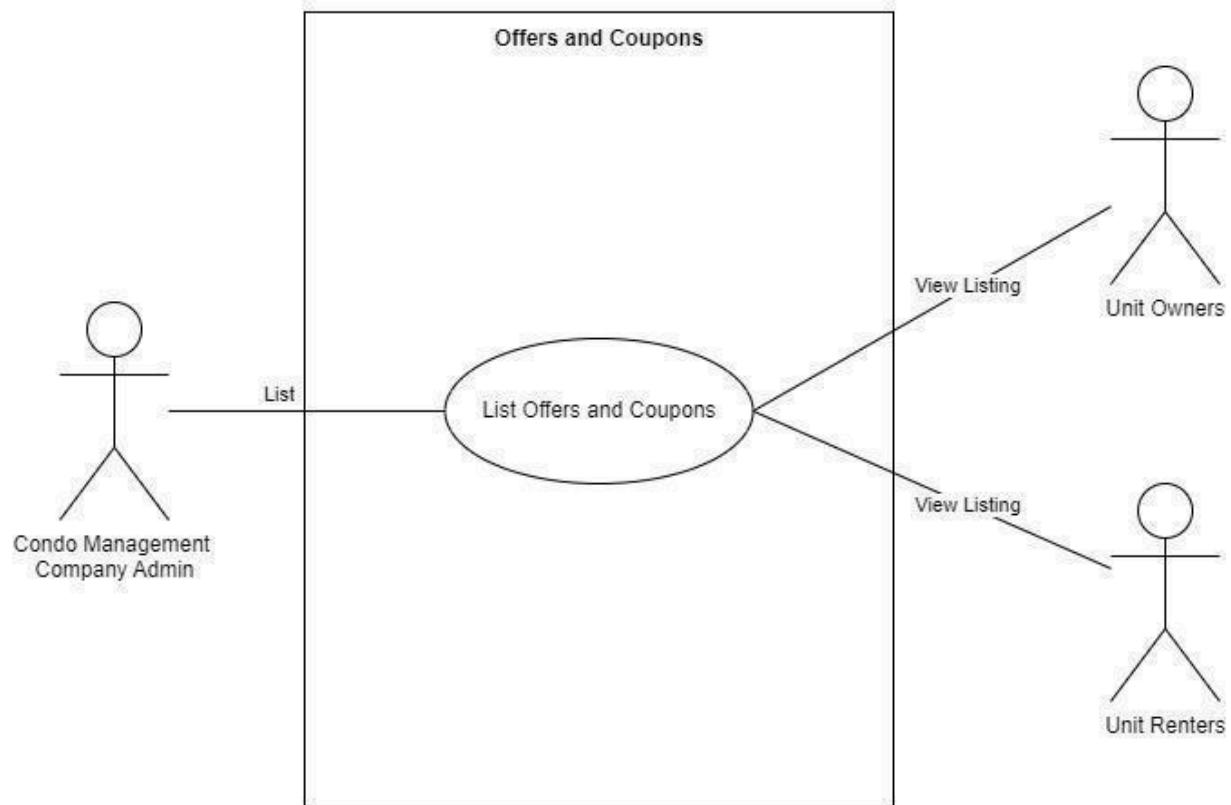
User Story # 21



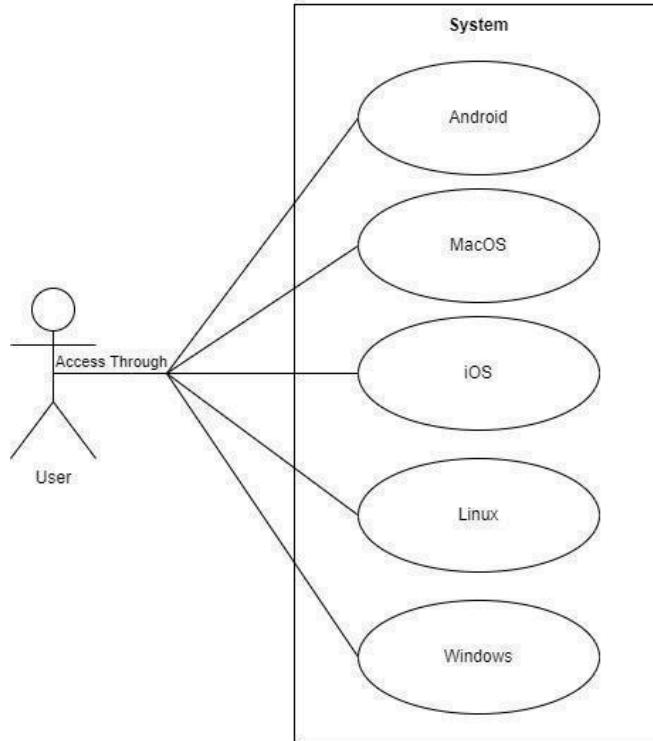
User Story # 22



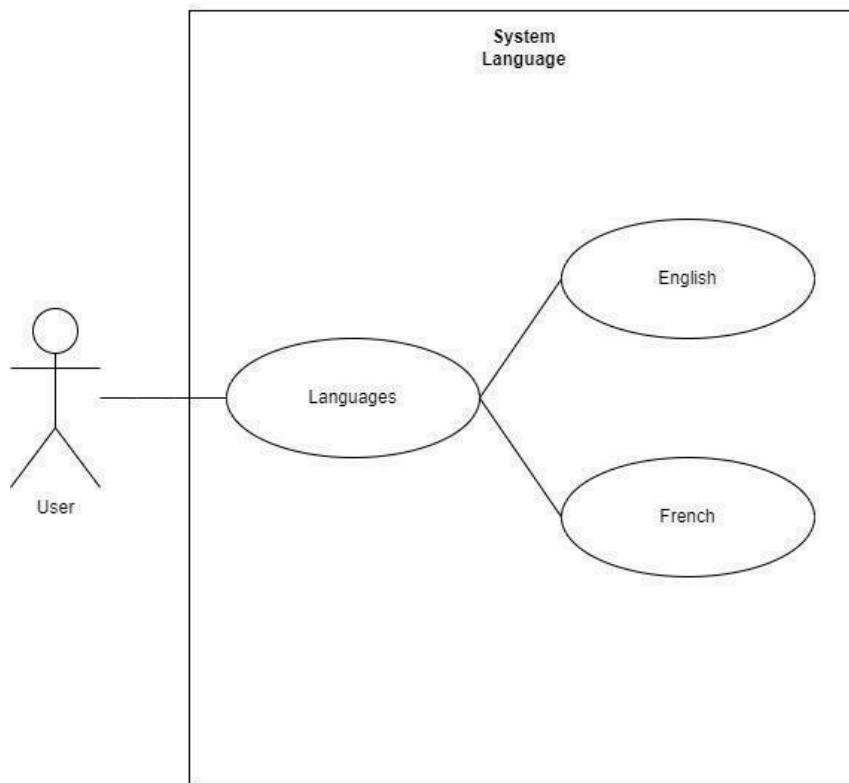
User Story # 23



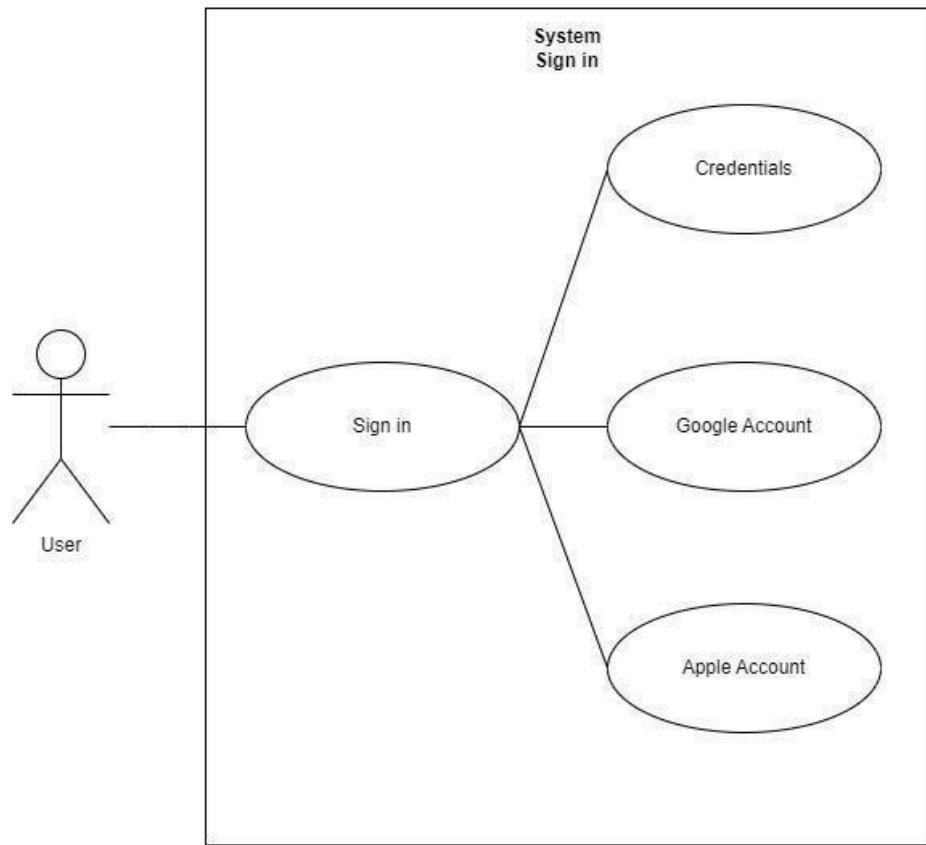
User Story # 24



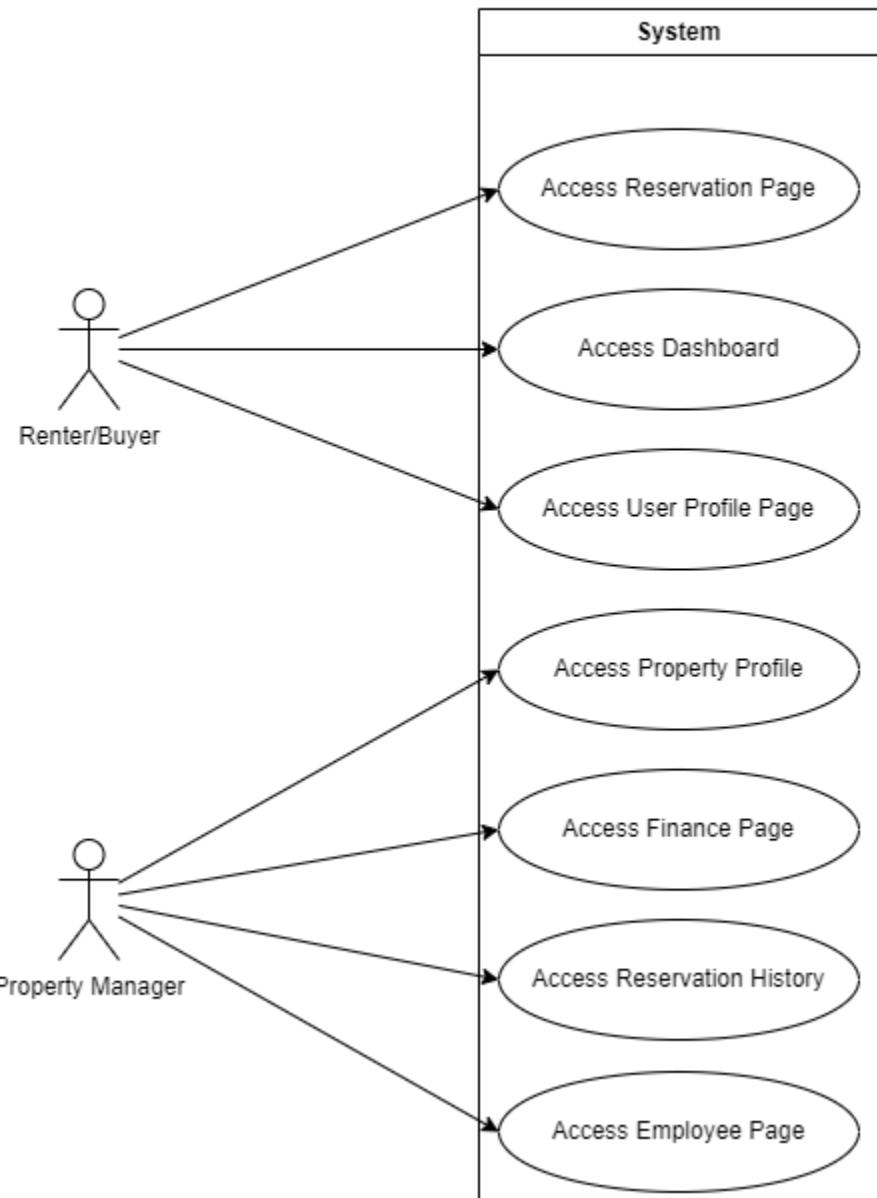
User Story # 25



User Story # 26



ID: 027



# Risk Assessment & Management Plan

## Purpose of the risk assessment and management plan along with its benefits

The Risk Assessment and Management Plan serves as a proactive approach to identify, assess, prioritize, and mitigate potential risks that may arise during the project lifecycle. Its purpose is to anticipate challenges, minimize negative impacts, and ensure the successful completion of the project within the defined constraints of time, budget, and quality. By systematically identifying and addressing risks, the plan helps to enhance project visibility, stakeholder confidence, and overall project resilience.

## How the risks and assessment were identified among our team

The identification and assessment of risks within our project were predominantly conducted through regular team meetings. During these meetings, team members engaged in open discussions and collaborative brainstorming sessions to identify potential risks across various aspects of the project, such as technical complexities, resource constraints, and communication challenges. Drawing upon the diverse perspectives and expertise of each team member, we collectively examined project documentation, shared insights from past experiences, and solicited input from stakeholders to comprehensively identify and assess risks. Through this iterative process of dialogue and reflection, we prioritized risks based on their likelihood and potential impact, laying the groundwork for effective risk management strategies and mitigation plans.

Impact	Low	Medium	High
Probability			
Low			#6, #16
Medium		#20	#1, #9, #11, #13
High	#25		#24

**Figure 5: Risk management chart**

## **Legend:**

High – Greater than <70%> probability of occurrence

Medium – Between <30%> and <70%> probability of occurrence

Low – Below <30%> probability of occurrence

ID	User Story ID	Risk	Type	Probability	Impact	Mitigation
1	1	Poor usability due to complex profile setup	User Experience	Medium	High	Conduct user testing and feedback sessions throughout development. Prioritize simplicity and clarity in design. Provide intuitive tooltips and guidance for profile setup.
2	6	Vulnerabilities in condo file upload leading to unauthorized access	Technical	Low	High	Implement input validation and file type restrictions. Use secure file storage mechanisms and regularly audit access controls.
3	9	Incorrect calculation of condo fees due to coding errors	Technical	Medium	High	Implement thorough unit testing and code reviews for fee calculation logic. Utilize standardized libraries or frameworks for financial calculations.
4	11	Inadequate logging of financial transactions leading to auditing challenges	Technical	Medium	High	Implement comprehensive logging of financial operations with proper data encryption. Regularly review and monitor logs for anomalies.
5	13	Performance issues in reservation system due to inefficient code	Technical	Medium	High	Optimize database queries and server-side processing. Implement caching mechanisms for frequently accessed data. Conduct load testing to identify and address performance bottlenecks.
6	16	Security vulnerabilities in reservation system leading to data breaches	Management	Low	High	Follow secure coding practices, such as input validation and parameterized queries, to prevent SQL injection and other attacks.
7	20	Lack of notification updates leading to user frustration	Management	Medium	Medium	Implement real-time notification system, provide manual refresh option. Optimize notification delivery, prioritize critical updates.
8	24	Cross-platform compatibility issues affecting user accessibility	Technical	High	High	Conduct thorough testing on various platforms, prioritize compatibility fixes. Implement responsive design, utilize cross-platform development frameworks.

9	25	Language barrier impacting user engagement	Management	High	Medium	Implement multilingual support, provide language selection options. Translate key content, offer language-specific forums or communities.
---	----	--	------------	------	--------	---

**Table 1: List of identified risks**

*Link to the Risk Analysis UrbanKey:*

[Risk analysis UrbanKey.xlsx](#)

# Sprint 3 Testing Report

## Front-End Testing

### **Introduction:**

Front-end testing is a crucial aspect of ensuring the reliability and functionality of web applications. In our Condo Management System Website project, we employed Jest, a widely used testing library compatible with React, to conduct front-end testing. This report outlines the tools utilized, the testing approach adopted, and presents the code coverage achieved through our testing efforts.

### **Tools Used:**

We employed Jest as our primary testing library due to its seamless integration with React. Jest offers a robust framework for creating and executing tests, making it an ideal choice for our front-end testing needs.

### **Testing Approach:**

To organize our testing efforts, we established a dedicated folder named "Front-end Testing" within the src directory of our project repository. Within this folder, we created individual testing files corresponding to each front-end page of our website. Following the naming convention recommended by Jest, each testing file was named after its respective page, suffixed with .test.js.

For example, if a page in our application is named Employee.js, we created a corresponding test file named Employee.test.js. Within these test files, we crafted unit tests focusing on the key features and functionalities of each page. These tests were designed to verify the expected behavior of the front-end components and ensure that they operate as intended.

Upon completing the creation of test files for each front-end page, we executed the tests using the command "npm test" in the terminal. Additionally, to assess the code coverage achieved by our testing suite, we utilized the command "npm test -- --coverage". This command generated detailed insights into the extent of code covered by our tests, allowing us to identify areas that require further testing or optimization.

### **Code Coverage:**

The code coverage analysis provided valuable insights into the effectiveness of our testing efforts. The following screenshots illustrate the code coverage metrics obtained from our front-end testing:

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	34.86	18.29	37.81	34.07	
Components/CondoOwnerDashboard	100	100	100	100	
OwnerDashboard.js	100	100	100	100	
Components/Employees	100	100	100	100	
Employee.js	100	100	100	100	
Components/FinanceDashboard	80	100	50	78.57	
Finance.js	80	100	50	78.57	15,21,69-102
Components/HomePage	100	100	100	100	
Home.js	100	100	100	100	
Components/LogIn	0	0	0	0	
Login.js	0	0	0	0	19-83
Components/NavBar	52.94	42.85	56.52	51.11	
NavBar.js	0	0	0	0	9-108
NavBar_Company.js	100	100	100	100	
NavBar_HomePage.js	100	100	100	100	
NavBar_User.js	91.66	100	83.33	90	20
Components/Popups	100	100	100	100	
MaintenanceRequest.js	100	100	100	100	
Notification.js	100	100	100	100	
PaymentHistory.js	100	100	100	100	
ReservationSuccess.js	100	100	100	100	
Components/ProfilePage	0	100	0	0	
Profile.js	0	100	0	0	14-180
Components/PropertyProfileManagement	60	0	50	66.66	
PropertyProfileManagement.js	60	0	50	66.66	17-21
Components/RegistrationKey	100	100	100	100	
RegistrationKey.js	100	100	100	100	
Components/Reservation	48.88	35	50	48.88	
Reservation.js	48.88	35	50	48.88	24-28,33,38,48-49,67-82,102
Components/SignUp	0	0	0	0	
SignUp.js	0	0	0	0	11-301
CustomeHooks	0	100	0	0	
useAuth.js	0	100	0	0	4-5

Test Suites: 8 failed, 10 passed, 18 total  
 Tests: 6 failed, 50 passed, 56 total  
 Snapshots: 0 total  
 Time: 22.631 s

Figure 6: Front-End Code Coverage

## **Back-End Testing**

### **Introduction:**

Backend testing is essential for ensuring the functionality, performance, and security of web applications. In our Condo Management System Website project, we employed Pytest, a powerful testing framework for Python, to conduct backend testing. This report outlines the tools utilized, the testing approach adopted, and presents the code coverage achieved through our testing efforts.

### **Tools Used:**

Pytest was selected as the primary testing framework for our backend testing needs. With its simplicity and flexibility, Pytest provides a comprehensive suite of features for writing and executing tests in Python applications, making it an ideal choice for our project.

### **Testing Approach:**

To organize and streamline our backend testing efforts, we established a dedicated folder named "backend" within the project repository. Within this folder, we created individual testing files corresponding to different backend functionalities and pages of our website.

For instance, if a backend module is responsible for handling employee-related operations, we created a test file named `test_employee.py` within the `backend` folder. Following the conventions of Pytest, we crafted test functions within these files to validate the behavior of backend components and functionalities.

Our testing approach focused on covering various aspects of backend functionality, including data validation, business logic, API endpoints, and database interactions. Each test function was designed to verify specific scenarios and edge cases, ensuring comprehensive test coverage.

Upon completing the creation of test files and test functions, we executed the tests using Pytest's command-line interface. Pytest automatically discovered and executed the test cases within the `backend` folder, providing detailed feedback on test results and identifying any failures or errors encountered during testing.

### **Code Coverage:**

The code coverage analysis provided valuable insights into the effectiveness of our backend testing efforts. By integrating code coverage measurement into our testing workflow, we gained visibility into the percentage of code executed by our test suite and identified areas that require additional testing or optimization.

## Coverage report: 90%

coverage.py v7.4.4, created at 2024-03-21 05:29 -0400

Module	statements	missing	excluded	coverage ↓
__init__.py	0	0	0	100%
config.py	6	0	0	100%
model\__init__.py	0	0	0	100%
model\handleRefresh.py	22	0	0	100%
routes\__init__.py	0	0	0	100%
services\__init__.py	0	0	0	100%
tests\__init__.py	0	0	0	100%
tests\test_auth.py	111	0	0	100%
tests\test_tokens.py	0	0	0	100%
tests\test_todo.py	135	1	0	99%
model\user.py	36	4	0	89%
routes\user_routes.py	27	3	0	89%
app.py	20	3	0	85%
services\token_service.py	27	6	0	78%
routes\auth_routes.py	18	5	0	72%
model\auth.py	81	26	0	68%
<b>Total</b>	<b>483</b>	<b>48</b>	<b>0</b>	<b>90%</b>

coverage.py v7.4.4, created at 2024-03-21 05:29 -0400

**Figure 7: Back-End Code Coverage**

# Short Sprint #3 Retrospective

## Crafting Excellence

### Key Takeaways from Our Project Postmortem

#### Introduction

In this sprint retrospective, our primary objective was to advance the development of our website by refining both front-end and back-end components. With the overarching goal of achieving an almost fully functioning website by the sprint's end, our efforts centered on seamlessly merging all front-end pages and integrating essential navigation bars. This sprint proved pivotal in bridging the gap between design and functionality as we diligently connected all elements cohesively, moving closer to our project milestones.

While we experienced minimal changes in diagrams and documentation, allowing for focused attention on pressing tasks, challenges emerged that required our attention. Inequitable task division, task interdependence, and difficulties understanding webpage interaction were notable hurdles. Task allocation disparities led to some team members bearing heavier workloads, exacerbating the challenges posed by interdependent tasks. Additionally, grappling with the intricacies of webpage interaction hindered the cohesion of our development efforts. These challenges highlighted the importance of addressing task distribution and deepening our understanding of project dynamics to enhance future performance.

#### What went wrong

##### *1 - Understanding Website Page Interaction*

During this sprint, our team encountered the challenge of comprehending the interactivity between each webpage. While we had previously worked on front-end components in the last sprint, this time, our focus shifted to merging all pages together.

To address this issue, we convened to dissect the purpose and functionality of each page, identifying their connections and interactions with one another. This collaborative approach helped us gain clarity on the navigation flow and streamline the integration process for a more cohesive website structure.

##### *2 - Task Interdependence*

Interdependence among tasks emerged as a significant challenge during this sprint, leading to unfair time constraints for certain team members. Some tasks were contingent upon the completion of others, leaving those with limited time to complete their work feeling overwhelmed. To address this issue in the next sprint, we plan to implement shorter deadlines that allow all team members sufficient time to fulfill their responsibilities. This proactive approach aims to alleviate time pressure and ensure equitable distribution of workload across the team.

Recognizing the impact of task interdependence on individual workload, we acknowledge the importance of setting realistic deadlines to accommodate the sequential nature of tasks. By establishing shorter, more manageable deadlines for each phase of the project, we can mitigate the risk of undue stress and promote a more

balanced workflow. Moving forward, fostering greater awareness of task dependencies and implementing timely deadlines will be essential to optimizing team productivity and fostering a collaborative work environment.

### ***3 - Task Division***

Addressing the challenge of task division proved essential during this sprint, as disparities arose in the workload distribution among team members. While some faced a multitude of complex tasks, others encountered fewer and simpler assignments.

To rectify this issue moving forward, we plan to convene another meeting to ensure a more equitable distribution of work for subsequent sprints. By fostering open communication and collectively strategizing task allocation, we aim to promote fairness and optimize productivity within the team.

### ***What went right***

#### ***1 - Documentation and Diagrams***

One notable success in this sprint was the minimal need for updates to diagrams and documentation. This allowed team members to redirect their focus towards more time-consuming tasks without the burden of extensive revisions. With these foundational elements largely unchanged, valuable time and resources were freed up, enabling team members to dedicate their efforts towards completing other critical tasks within the project.

This efficiency in documentation management served to optimize workflow and maximize productivity throughout the sprint, highlighting the benefits of thorough planning and execution in project management.

#### ***2 - Communication***

Another success in this sprint was the efficient communication among all team members whenever problems arose. Utilizing Discord as our primary communication platform, we promptly addressed issues and coordinated solutions. Whether through direct messaging or planning quick meetings, our team ensured that communication remained open and effective, fostering a collaborative environment conducive to problem-solving and progress.

This proactive approach to communication allowed us to swiftly address challenges and maintain momentum throughout the sprint. By promptly sharing updates and coordinating responses, we mitigated potential setbacks and kept our project on track. Moving forward, we recognize the importance of maintaining this level of communication to navigate future obstacles effectively and achieve our project objectives.

### ***Conclusion***

In this sprint retrospective, we encountered a blend of challenges and successes that provided valuable insights into our project's progress. While we made significant strides in advancing the development of our website, challenges such as understanding webpage interaction, inequitable task division, and task interdependence underscored the importance of refining our project management strategies. However, amidst these challenges, notable successes emerged, including minimal changes in documentation and diagrams, as well as efficient communication among team members.

Moving forward, we are committed to addressing the challenges identified during this sprint by implementing strategies to improve task distribution, deepen our understanding of project dynamics, and enhance communication channels. By leveraging the lessons learned and building upon our successes, we are confident in our ability to overcome obstacles and continue making meaningful progress towards achieving our project goals in future sprints.

# Sprint 4 Release Plan

In Sprint 4 of the Condo Management System, our trajectory is set to advance the platform's capabilities with a concerted emphasis on financial transparency, backend robustness, and user empowerment. This sprint will delve into a series of 28 sub user stories, spanning from Sub User Story #4.2 to #22.2, meticulously selected to address pivotal enhancements identified from prior development cycles. We had to push sub user stories a couple of user stories from Sprint 3 to Sprint 4 to complete, however our team is confident it will be able to do so.

Our goal for this sprint is to focus on delivering seamless financial integration within the owner's dashboard, fortifying backend infrastructure, and enhancing user autonomy within the system. We aim to introduce a real-time financial overview on the dashboards, allowing condo owners to monitor their financial status promptly and accurately. A considerable portion of our efforts will be dedicated to developing secure and efficient backend services for storing and managing property profiles and file data in MongoDB, directly tying detailed property specifics to their corresponding units for heightened management accuracy. We are set to implement comprehensive role management functionalities. These include crafting a system for the generation, distribution, and management of registration keys, thereby ensuring that property managers can oversee unit associations with greater ease. Further refining our financial modules, we will introduce automated calculations of condo fees based on unit and parking space dimensions, alongside deploying dashboard utilities for presenting these fees to owners, ensuring utmost clarity and transparency. To boost user engagement and foster community within the condo environment, we will establish interactive forums and versatile notification systems, aiming to provide a dynamic platform for communication, collaboration, and community building. Additionally, we will rollout support features for event organization, empowering users to host, manage, and attend community events, thereby nurturing community spirit. In tandem with these functional upgrades, we are committed to extending the system's reach through cross-platform compatibility and bilingual support, thereby accommodating a diverse user demographic. Reinforcing the system's security will also be a paramount goal, with the integration of reliable sign-in mechanisms through mainstream accounts.

As we forge ahead with these strategic initiatives, we expect to significantly elevate the user experience, streamline property management workflows, and lay a robust foundation for the system's future expansion. By the conclusion of Sprint 4, we envision a system that is not only technologically advanced but also aligns seamlessly with the evolving needs and expectations of our user community.

## Release Plan Legend (on the Excel sheet):

User Story ID	User Story Points (USP)	Priority	Status
User Stories are from # 17 to #26. The [ ] sub-user stories are in the format 2.1, 2.2, 3.1, etc.	The user story points are done based on the Fibonacci sequence.	<ul style="list-style-type: none"><li>● High</li><li>● Medium</li><li>● Low</li></ul>	<ul style="list-style-type: none"><li>● NOT STARTED</li><li>● TODO IN SPRINT</li><li>4</li><li>● DONE</li><li>● PUSHED TO SPRINT 4</li></ul>

There are a total of 132 story points for Sprint 4, and approximately 217 hours of work.

**Link to the Release Plan Sprint 4 Excel file:**

<https://docs.google.com/spreadsheets/d/1Ulh0fSHoO57ndRfikHvgb0A5O0gpeY3/edit?usp=sharing&ouid=112342122863028184816&rtpof=true&sd=true>

# UI Prototypes

## Home Page

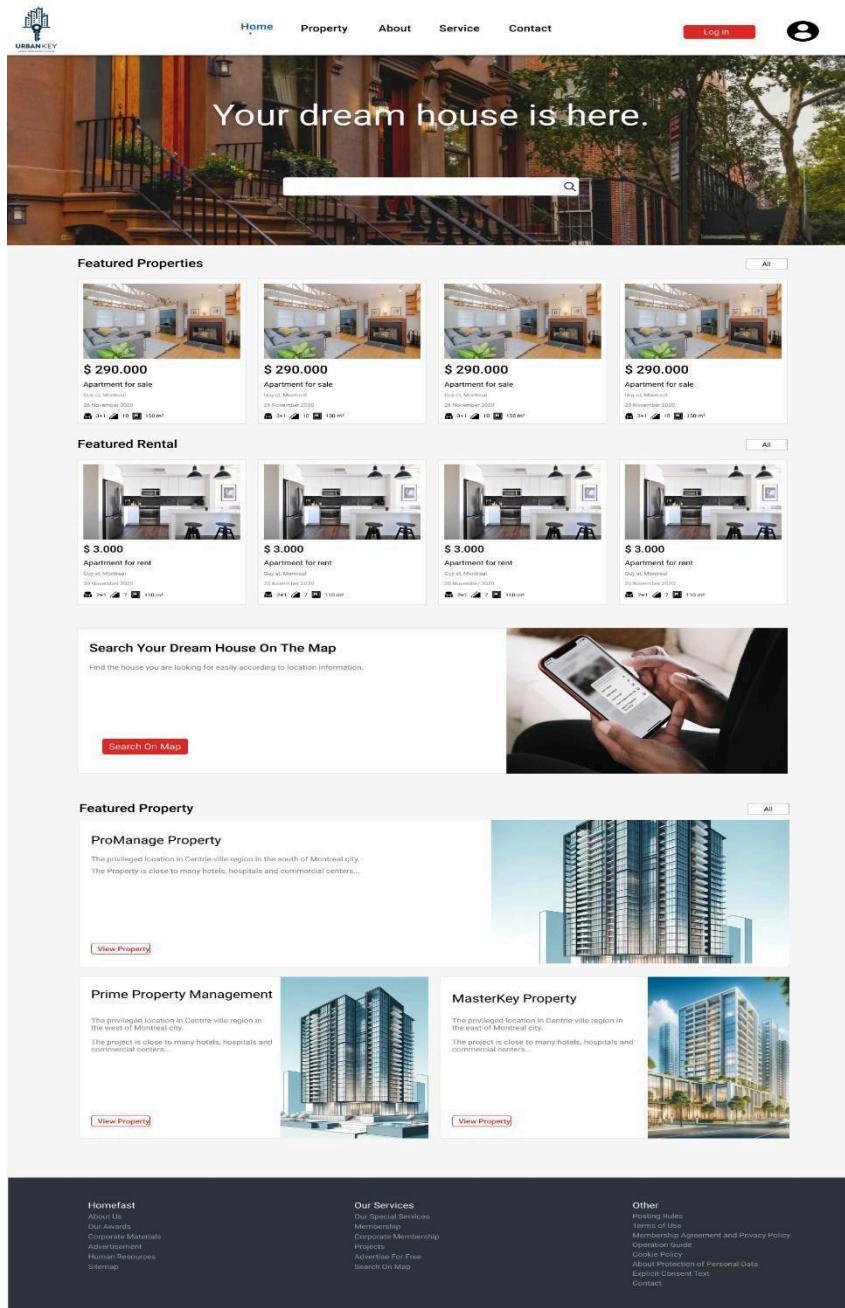


Figure 8

The homepage design for a real estate service features a sophisticated, user-centric interface. It includes an intuitive navigation bar, a striking banner for property searches, neatly organized listings of featured properties and rentals, and a map search functionality.

## Login

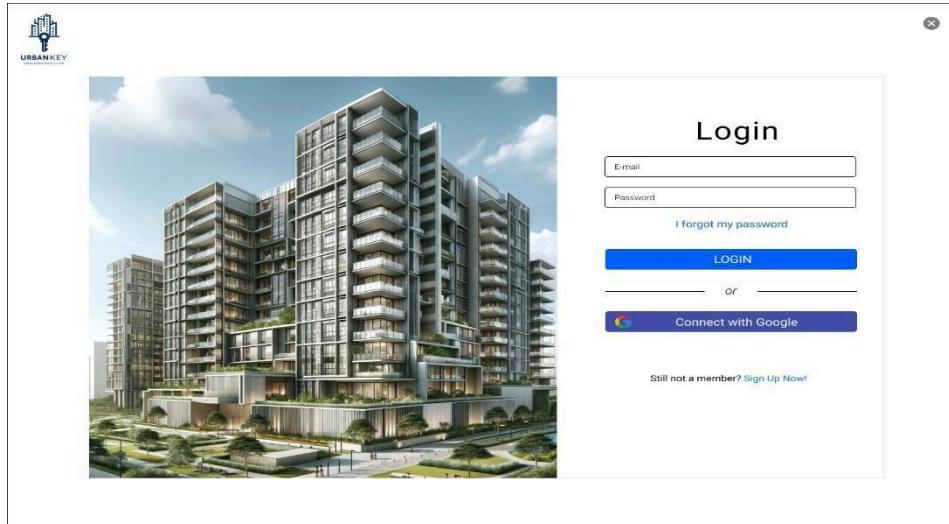


Figure 9

It offers users the choice to log in using their email and password or via Google. A link for those who have forgotten their password and a prompt to sign up for new members enhance the user experience with convenience and accessibility.

## Sign Up

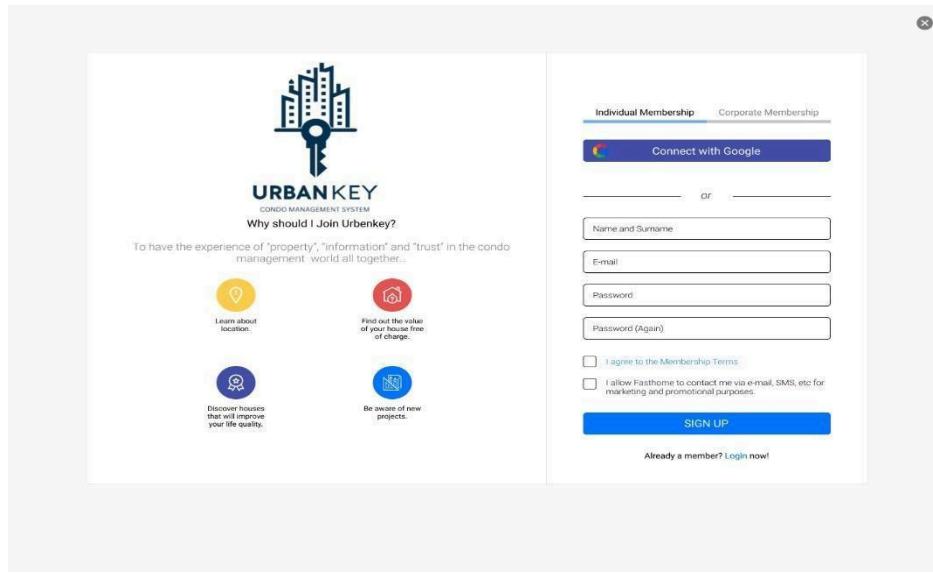


Figure 10

The form provides a choice between individual or corporate membership and the option to register with options for Google sign up, and clearly presents terms of service and promotional contact opt-ins, ensuring an informative yet streamlined user journey.

# Property Page

The screenshot displays a detailed property listing for a "Single Person House" located in Montreal, Quebec. The top navigation bar includes links for Home, Property, About, Service, and Contact, along with a user profile icon and a notification bell.

**Single Person House**  
Montreal, QC

**General Information**

Condo No.	0-0002	Floor Location	2
Purchase Date	20 November 2020	Furnished	Yes
Housing Shape	Apartment	Front	Northwest
Room + Living Number	1 + 1		
Gross / Net M <sup>2</sup>	50 M <sup>2</sup> / 110 M <sup>2</sup>		
Warming Type	Natural Gas		
Building Age	6		

**Explanation**  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Egestas ac convallis tellus pellentesque non odio consectetur bibendum. Auctor leo risus in tristique sit enim nec sed. Ridiculus vulputate facilisi a velit cursus sapien egestas nec, accumsan.

**Interior Features**

- ✓ ADSL
- ✓ Alarm
- ✓ Balcony
- ✓ Barbecue
- ✓ Laundry room
- ✓ Wallpaper
- ✓ Dressing Room
- ✓ Video Intercom
- ✓ Shower
- ✓ Laminate
- ✓ Panel Door
- ✓ Blinds
- ✓ Sauna
- ✓ Satin Plaster
- ✓ Satin Color
- ✓ Ceramic Floor

**External Features**

- ✓ Elevator
- ✓ Gardened
- ✓ Fitness
- ✓ Security
- ✓ Thermal Insulation
- ✓ Garbage
- ✓ Tennis Court
- ✓ Car Park
- ✓ PVC
- ✓ Basketball Field
- ✓ Market

**Financial Status**

- Monthly Condo Fees: \$4568
- Outstanding Balances: \$0

**Maintenance Requests**

Request #001	In Progress
Request #002	Completed
Request #003	Pending

**Location Information**

A map showing the location of the property in Montreal, Quebec, with various landmarks and streets labeled.

**Homefast**

- About Us
- Our Awards
- Corporate Materials
- Advertisement
- Human Resources
- Sitemap

**Our Services**

- Our Special Services
- Membership
- Corporate Membership
- Projects
- Advertise For Free
- Search On Map

**Other**

- Posting Rules
- Terms of Use
- Membership Agreement and Privacy Policy
- Operation Guide
- Cookie Policy
- About Protection of Personal Data
- Explicit Consent Text
- Contact

Figure 11

## Notification

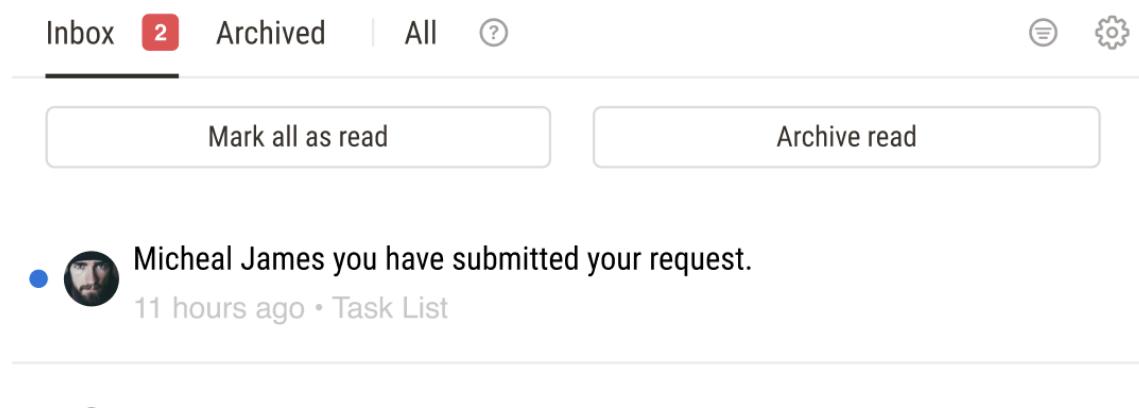


Figure 12

## Payment

Payment History				
Payment Date	Amount	Payment Method	Reference Number	Status
September 2023	\$4568.00	Bank Transfer	12345	Paid
October 2023	\$4568.00	Credit Card	65078	Pending
November 2023	\$4568.00	Debit Card	78965	Rejected

[Go Back](#) [See Invoices](#)

Figure 13

The property page layout is detailed, featuring a photo gallery, essential information about the property, and an interactive map. The design also includes sections for financial details for the user when they pay rent or mortgage and so on.

# Maintenance Requests

The screenshot shows a web form titled "Maintenance Requests". At the top is a text input field labeled "Title". Below it is a section titled "Request Description" with a text area containing placeholder text about reporting various property issues. At the bottom is a large red button with the text "Submit Your Request".

Figure 14

It encourages users to report issues or make requests related to property upkeep directly through the website.

## Financial Management Dashboard

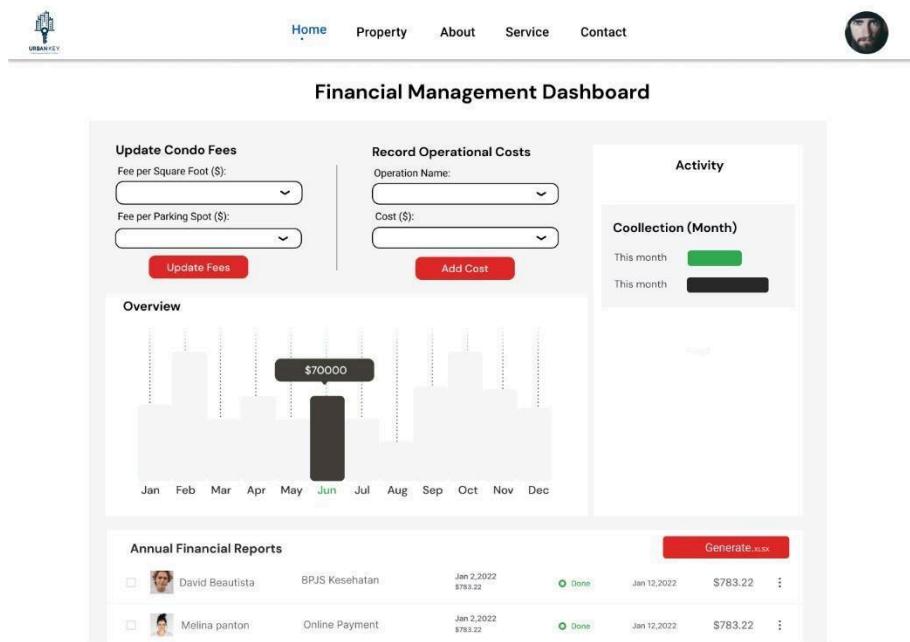


Figure 15

That allows updating of condo fees, recording operational costs, and provides an overview of financial activities, including a bar graph and a section to get annual financial reports.

# Membership Information

The screenshot shows a web page titled "Membership Information". At the top, there is a logo for "URBANKEY" featuring a stylized building and key icon, followed by navigation links for "Home", "Property", "About", "Service", and "Contact". A circular profile picture placeholder is located in the top right corner.

The main content area contains several input fields:

- Name / Surname: An input field with a placeholder.
- E-mail: An input field with a placeholder.
- Province: A dropdown menu with a placeholder.
- City: A dropdown menu with a placeholder.
- Mobile Number: An input field with a placeholder.
- Mobile Number 2: An input field with a placeholder.
- Confirm Your password: An input field containing "\*\*\*\*\*" as a placeholder.
- Address: A large text area for entering address details.

On the right side of the form, there is a circular placeholder for a profile picture with a camera icon and the text "Upload Profile Picture".

Below the form, there are two buttons: "E-mail" and "SMS". Underneath these buttons, there is a checkbox labeled "I want to be informed about all announcements and campaigns via commercial electronic mail." followed by two radio buttons labeled "No" and "Yes".

A red "Save" button is positioned at the bottom center of the form.

**Figure 16**

This allows users to manage their profile information, with fields for personal details, contact information, and preferences. It may also allow users to upload a profile picture and opt-in for notifications.

# Property Profile Management

The screenshot displays a web-based property listing form titled "Property Profile Management". At the top, there's a navigation bar with links for Home, Property, About, Service, and Contact. A logo for "PROPERTY" is on the left, and a user icon is on the right.

**Property Details:** This section contains various input fields for property information such as Category (Housing), Title, Description, Unit Owner, Price, Number of Rooms, Number of Living Rooms, Gross SF, Net SF, Warning Type, Building Age, Floor Location, Available for Loan, Published, Parking, Postings Spot ID, Locker, Rental Income, and Type.

**Location Information:** Fields for Province, City, and Neighborhood.

**Map:** A map showing the location of the property, with a red marker indicating the exact spot.

**Upload Condo Files:** A section for uploading condo files, with "Choose File" and "Upload File" buttons.

**Posting Photos:** A section for posting photos, with a note that up to 30 photos can be added. It includes a "Browse From Computer" button and a placeholder area for photos.

**Advertise Features:** Two columns of checkboxes for advertising features. The first column includes Interior Features like A/C, Air Conditioning, Balcony, Built-in Kitchen, Furnished, Laundry Room, Air Conditioning, Wallpaper, and Dressing Room. The second column includes Exterior Features like Elevator, Garden, Fence, Security, Thermal Insulation, and Generator. The third column includes Video Intercom, Lift, Shower, TV Satellite, Laminate, Panel Door, Marble Floor, Blinds, Sinks, Parent Bathroom, Doormat, Car Park, Playground, DCC, Billing, and Water Tank. The fourth column includes Pergola, Landscaping, Satin Cover, Ceramic Floor, Spotlight, Fireplace, Terrace, Gazebo, Underfloor Heating, Double Glazing, Tennis Court, Fire Escape, Emergency Exit, Football Field, Basketball Field, and Market.

**Buttons:** "Send Registration Keys" and "Send" buttons at the bottom.

Figure 17

This allows property owners to enter details about their property for listing purposes. It includes sections for property features, location mapping, file uploads for important documents, and a photo gallery.

# Reservation System

The screenshot shows the homepage of the Facility Reservation System. At the top, there is a navigation bar with links for Home, Property, About, Service, and Contact. To the right of the navigation is a user profile icon. Below the navigation, the title "Facility Reservation System" is centered. Underneath the title, the heading "Book a Facility" is displayed. On the left, a dropdown menu titled "Choose a Facility:" lists "Sky Lounge" (which is highlighted in blue), "Spa & Fitness", and "Gym". On the right, a calendar titled "Select Date:" shows the month of September 2021. The days of the week are labeled from Su to Sa. Specific dates are marked: 29th is circled in red with the text "Booked", 30th is circled in orange with the text "Available", and 31st is also circled in orange. A "Book Now" button is located at the bottom center of the form.

Figure 18



Congrats

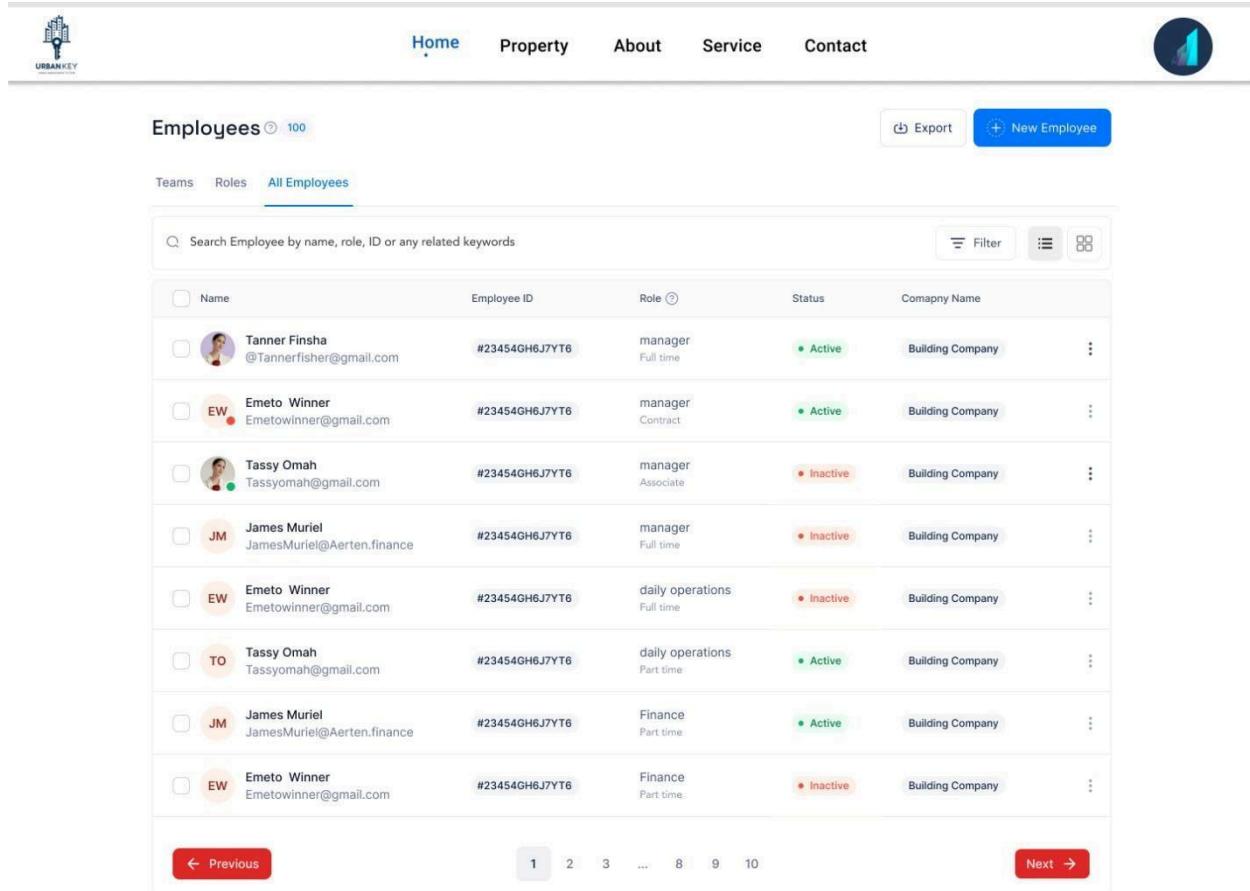
Reservation has been successfully made.

[Back to Home](#)

Figure 19

This is the facility booking system, where users can select from available facilities and dates, followed by a confirmation message indicating a successful reservation.

# Employees



The screenshot shows a web-based employee management system. At the top, there's a navigation bar with links for Home, Property, About, Service, and Contact. On the far left is a logo for 'URBANKY'. On the far right is a circular profile picture placeholder. Below the navigation is a header bar with the title 'Employees' followed by a count of '100'. To the right of the title are two buttons: 'Export' and 'New Employee'. Underneath the header are three tabs: 'Teams', 'Roles', and 'All Employees', with 'All Employees' being the active tab. A search bar allows users to search by name, role, ID, or keywords. To the right of the search bar are filter and view mode buttons. The main content area is a table listing eight employee records. Each record includes a checkbox, a profile picture, the employee's name, their email, their Employee ID (#23454GH6J7YT6), their role (e.g., manager, daily operations, Finance), their status (Active or Inactive), their company (Building Company), and a three-dot menu icon. At the bottom of the table are navigation buttons for 'Previous' and 'Next' pages, along with a page number indicator showing pages 1 through 10.

Name	Employee ID	Role	Status	Company Name
Tanner Finsha @Tannerfisher@gmail.com	#23454GH6J7YT6	manager Full time	Active	Building Company
Emeto Winner Emetowinner@gmail.com	#23454GH6J7YT6	manager Contract	Active	Building Company
Tassy Omah Tassyomah@gmail.com	#23454GH6J7YT6	manager Associate	Inactive	Building Company
JM James Muriel JamesMuriel@Aerten.finance	#23454GH6J7YT6	manager Full time	Inactive	Building Company
EW Emeto Winner Emetowinner@gmail.com	#23454GH6J7YT6	daily operations Full time	Inactive	Building Company
TO Tassy Omah Tassyomah@gmail.com	#23454GH6J7YT6	daily operations Part time	Active	Building Company
JM James Muriel JamesMuriel@Aerten.finance	#23454GH6J7YT6	Finance Part time	Active	Building Company
EW Emeto Winner Emetowinner@gmail.com	#23454GH6J7YT6	Finance Part time	Inactive	Building Company

**Figure 20**

This is an employee management dashboard, listing employee details and providing functionality for filtering, searching, and managing employee records within a real estate or property management company.

# Registration Key Page

The screenshot shows a web page for URBANKEY. At the top left is the URBANKEY logo, which includes a stylized building icon and the text "URBANKEY CONDOMINIUM MANAGEMENT". The top navigation bar has links for "Home", "Property", "About", "Service", and "Contact". Below the navigation, the main heading "Welcome to URBANKEY." is displayed in large black text. To the right of the heading is a small "X" icon. In the center is a circular graphic featuring a house, a sun, clouds, and a hand holding a key. Below this graphic is a text block: "To register as a condo owner or rental user, you need a registration key provided by your condo management company. If you have not received your key or need assistance, please contact support." Below the text is a white input field with a placeholder "Enter Registration Key". Underneath the input field is a red button labeled "Become A member". At the bottom of the page, there is a link "Need Help? Contact Support.".

**Figure 21**

This is to prompt users to enter a registration key to gain access or to contact support for assistance.

## Reservation Page for Condo Company

The screenshot shows a web-based appointment management system. At the top, there's a navigation bar with links for Home, Property, About, Service, Contact, and Appointments. A search bar at the top right allows users to search for appointments by name or keyword. Below the search bar is a toolbar with filters: 'Select Facility' (dropdown), 'Select username' (dropdown), 'Hide visited' (checkbox), 'Show empty' (checkbox), and date range pickers for '05.03.2022' to '05.12.2022' and 'June 1, 2022'. A blue button labeled '+ New Appointment' is located in the top right corner of the toolbar.

Time	User Name	Owners type	Facility	Contact Number	Email	Creator	Status	waiting time	Actions
7:00 - 7:10	Braha Marlam Roh	Condo	Sky Lounge	(912) 330-4635	caronni@optonline.net		Visited	00:00:00	
7:20 - 7:30	Jordyn Dokidis	Condo	Sky Lounge	(692) 656-8460	keijser@gmail.com		Visited	00:00:00	
7:40 - 7:50	Aspen Passaquinidici Arcand	Condo	Spa & Fitness	(807) 317-8350	krueger@mac.com		Visited	00:00:00	
8:00 - 8:10	Carter Botosh	Condo	Spa & Fitness	(465) 233-0398	kosact@comcast.net		Scheduled	00:00:00	
8:20 - 8:30	Carter Smith	Condo	Sky Lounge	(435) 918-6920	jbartha@outlook.com		Waiting	05:54	
8:40 - 8:50	Ashlynn Botosh	Condo	Sky Lounge	(319) 263-6761	whimzy@outlook.com		Scheduled	00:00:00	
9:00 - 9:10	Ashlynn Botosh	Condo	Spa & Fitness	(796) 225-6190	pierce@yahoo.ca		Scheduled	00:00:00	
9:20 - 9:30	Talan Sepdimus	Condo	Spa & Fitness	(999) 440-1134	pakaste@sbcglobal.net		Scheduled	00:00:00	
9:40 - 9:50									
10:00 - 10:10	Alfonco Franci	condo	Spa & Fitness	(999) 440-1035	duncand@sbcglobal.net		Scheduled	00:00:00	

At the bottom left, there's a page number indicator '10' with a dropdown arrow, followed by 'Show on page' and a 'Save' button. The bottom right has a 'Page' button and a navigation bar with arrows and page numbers (1, 2, 3, ..., 8, 9, 10).

Figure 22

This interface displays all successful reservations made by condo owners. It features a table with reservation details including the user's email, the booked facility, and the selected date.

## Manager Employee Page

The screenshot shows a web-based request management system. At the top, there's a navigation bar with links for Home, Property, About, Service, Contact, and a personalized greeting 'Hello Manager,'. A search bar at the top right allows users to search for employees by name, role, ID, or keywords. A blue button labeled '+ New Request' is located in the top right corner of the toolbar.

Condo Owner	Title of the request	Description	Contact Number	User Email	Assigned Employee	Roles	Status	ID	
Braha Marlam Roh	A leaky faucet	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....	(912) 330-4635	caronni@optonline.net	Tanner Finsha Tannerfisher@gmail.com	daily operations	Active	#234540H6J7YT6	
Jordyn Dokidis	A broken washing machine	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....	(692) 656-8460	keijser@gmail.com	Emeto Winner Emetowinner@gmail.com	daily operations	Active	#234540H6J7YT6	
Aspen Arcand	A sliding door that's come off of its track	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....	(807) 317-8350	krueger@mac.com	Tassy Omaha Tassymah@gmail.com	daily operations	Inactive	#234540H6J7YT6	
Carter Botosh	Kitchen appliance not working	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....	(465) 233-0398	kosact@comcast.net	James Muriel JamesMuriel@aerten.finance	daily operations	Inactive	#234540H6J7YT6	
Carter Smith	plumbing problems that cause flooding	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....	(435) 918-6920	jbartha@outlook.com	James Muriel JamesMuriel@aerten.finance	daily operations	Active	#234540H6J7YT6	
Ashlynn Botosh	A security threat	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....	(319) 263-6761	whimzy@outlook.com	Tassy Omaha Tassymah@gmail.com	daily operations	Inactive	#234540H6J7YT6	
Ashlynn Botosh	The presence of harmful gases	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....	(796) 225-6190	pierce@yahoo.ca	Emeto Winner Emetowinner@gmail.com	daily operations	Active	#234540H6J7YT6	
Talan Sepdimus	Parking keyfop	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....	(999) 440-1134	pakaste@sbcglobal.net	Emeto Winner Emetowinner@gmail.com	daily operations	Active	#234540H6J7YT6	
							Active	#234540H6J7YT6	
Alfonco Franci	Small holes in the wall	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....	(999) 440-1035	duncand@sbcglobal.net	Tanner Finsha Tannerfisher@gmail.com	daily operations	Active	#234540H6J7YT6	

At the bottom left, there's a 'Previous' button with a left arrow, followed by a page number indicator '1 2 3 ... 8 9 10' and a 'Next' button with a right arrow. The bottom right has a 'Page' button and a navigation bar with arrows and page numbers (1, 2, 3, ..., 8, 9, 10).

**Figure 23**

Allows a manager to assign daily operation tasks to employees. Shows owner names, request IDs, titles, descriptions, and request statuses.

## Daily Operation Employee Page

The screenshot shows a web application interface titled "Hello Daily operations," part of the URBANKEY system. The top navigation bar includes links for Home, Property, About, Service, Contact, and a user icon. Below the navigation is a search bar with placeholder text "Search Employee by name, role, ID or any related keywords" and a "New Request" button. A filter icon is also present. The main content area displays a table of assigned tasks:

	Condo Owner	Title of the request	Description	Contact Number	User Email	Condo no#	Assigned Employee	Roles	Status
<input type="checkbox"/>	Braha Marlarn Roh	A leaky faucet	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....)	(912) 330-4635	caronni@optonline.net	109	Tanner Finsha Tannerfisher@gmail.com	daily operations	<span>In progress</span>
<input type="checkbox"/>	Jordyn Dokidis	A broken washing machine	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....)	(692) 656-8460	keijser@gmail.com	202	Emeto Winner Emetowinner@gmail.com	daily operations	<span>In progress</span>
<input type="checkbox"/>	Aspen Arcand	A sliding door that's come off its track	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....)	(807) 317-8350	krueger@mac.com	505	Tassy Omaha Tassyomaha@gmail.com	daily operations	<span>Not yet</span>
<input type="checkbox"/>	Carter Botosh	Kitchen appliance not working	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....)	(466) 233-0398	kosact@comcast.net	1220	James Muriel JamesMuriel@Aerten.finance	daily operations	<span>Not yet</span>
<input checked="" type="checkbox"/>	Carter Smith	plumbing problems that cause flooding	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....)	(435) 918-6920	jbarla@outlook.com	1307	James Muriel JamesMuriel@Aerten.finance	daily operations	<span>Done</span>
<input type="checkbox"/>	Ashlynn Botosh	A security threat	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....)	(319) 263-6751	whimsy@outlook.com	1602	Tassy Omaha Tassyomaha@gmail.com	daily operations	<span>Not yet</span>
<input type="checkbox"/>	Ashlynn Botosh	The presence of harmful gases	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....)	(796) 226-6190	pierce@yahoo.ca	1808	Emeto Winner Emetowinner@gmail.com	daily operations	<span>Done</span>
<input type="checkbox"/>	Talan Sepdmus	Parking keyfop	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....)	(999) 440-1134	pakaste@sbcglobal.net	2202	Emeto Winner Emetowinner@gmail.com	daily operations	<span>In progress</span>
<input type="checkbox"/>	Alfonco Franci	Small holes in the wall	I am writing to request repairs to the (appliance, heating/air conditioning, plumbing issue.....)	(999) 440-1035	duncand@sbcglobal.net	1003	Tanner Finsha Tannerfisher@gmail.com	daily operations	<span>In progress</span>

At the bottom, there are navigation buttons for "Previous" and "Next" with page numbers 1 through 10.

**Figure 24**

For daily operation employees to view and manage their assigned tasks. They can update the status of tasks, which reflects on both the manager's page and the condo owner's dashboard.

# User Story Coverage Analysis

The Simplified Condo Management App is designed to facilitate the management of condos by providing a platform for public users, condo owners, rental users, and condo management companies to interact, manage properties, and access information. These functionalities are translated into user stories that describe the desired features from the perspective of the app's users. The aim is to ensure that the development efforts align with user needs and expectations, providing a clear and concise guide for the development team.

## • Feature 1: Public User Profile Creation

- User Story 1.1: Ensures public users can engage with the community, highlighting the app's social integration and personalization capabilities.

## Registration and Verification

## • Features 2 & 3: Registration Key for Ownership and Rental

- User Stories 2.1 & 3.1: Address the need for a secure and verified entry into the system as either condo owners or rental users, emphasizing the app's focus on security and proper authorization.

## Condo Owner Features

## • Feature 4: Condo Owner Dashboard

- User Story 4.1: Allows condo owners to have a comprehensive view of their properties, ensuring a seamless management experience.

## Condo Management Company Features

## • Features 5 to 7: Property Management

- User Stories 5.1, 6.1, & 7.1: Cater to the operational needs of condo management companies, facilitating document management, detailed property information entry, and profile management.

## Financial System Features

## • Features 9 to 12: Financial Management

- User Stories 9.1, 10.1, & 11.1: Emphasize the importance of a simplified yet detailed financial system within the app, supporting budget management and financial transparency.

## Reservation System Features

## • Features 13 to 16: Facility Reservation

- User Story 13.1: Ensures users can efficiently plan and book common facilities, enhancing user convenience and facility utilization.

## Role and Request Management Features

## • Features 17 to 19: Role Assignment and Request Submission

- User Stories 17.1 & 18.1: Focus on the administrative aspect of condo management, allowing for efficient role assignment and request handling.

## Notification and Additional Features

- Feature 20: Notifications

- User Story 20.1: Keeps users informed of activities related to their submissions or assignments, fostering an environment of transparency and responsiveness.

## Community Engagement and App Accessibility

- Features 21 to 26

- User Stories 21.1 & 24.1: Highlight the app's commitment to community building and accessibility, ensuring it caters to a wide audience with varying needs.

## Navigation and User Interface

- Feature 27: Navigation and Navbar

- User Story 27.1: Underlines the necessity of an intuitive and user-friendly interface, allowing for easy navigation throughout the app's features.

# Sprint 4 Testing Plan

This project documentation refers to the outlining of a comprehensive strategy for verifying and validating the software developed during the fourth sprint of the Condo Management System project. In order to meet the requirements of the project, it is crucial to incorporate a multi-level testing strategy that includes unit tests, integration tests, system tests, and a combination of automated and manual testing processes.

As a reminder, the stack we have selected and used for our Condo Management System is React for the front-end of the website app, as well as HTML/CSS/ JavaScript. For the backend, we have used Python alongside Flask, a micro web framework. Moreover, MongoDB was used as our NoSQL database. The APIs remain to be determined in future sprints.

## Testing Tool Selection:

As we've progressed in the development of our project, we have continued to employ Pytest, a testing framework for Python, as our tool for backend testing. Additionally, we have incorporated the usage of Jest, a JavaScript testing framework, alongside the *React Testing Library* in order to test our front-end React components. We will stick with these tools moving forward in our subsequent sprints.

## Front-end Testing:

The combination of Jest and the React Testing Library allows us to thoroughly test the behavior and functionality of our React components in isolation. Each React component is accompanied by a corresponding *.test.js* file, which contains the unit tests for that component. These tests can be found in the `FrontEndTesting` directory of our GitHub repository.

## Back-end Testing:

Using Pytest, our tests are designed to verify the functionality of our Flask-based REST API endpoints and the underlying business logic. The back-end tests can be found in the `backend/tests` directory of our repository. Specifically, the files “`test_auth.py`” and “`test_todo.py`” contain the tests for authentication and todo functionalities, respectively.

- The “`app.py`” file assesses the functionality and robustness of the application's user profile management and authentication token processes. It simulates critical user interaction scenarios, such as profile retrieval, profile updates, and handling of refresh tokens under various conditions. By using MagicMock and patch from the unittest.mock library, the tests avoid direct database interactions and cryptographic operations, instead simulating these processes to validate the application's behavior in controlled environments. For example, it tests the profile update functionality by simulating a user request to update their profile information, expecting successful completion or handling of exceptions gracefully. The refresh token handling is properly tested under different circumstances, including valid tokens, expired tokens, and invalid tokens, ensuring the application correctly validates and responds to each scenario. Additionally, the file incorporates tests for custom decorators used for route protection, verifying that routes are accessible only with valid authentication tokens and appropriately rejecting

unauthorized access attempts. This approach to testing not only guarantees the security and efficiency of the user management and authentication systems but also ensures that potential issues can be identified and resolved promptly, enhancing the overall integrity and reliability of our web app.

- The “test\_auth.py” file is structured to conduct automated tests for the Flask-based backend, specifically focusing on user authentication mechanisms, utilizing the Pytest framework. It effectively simulates various authentication-related scenarios including user sign-up, sign-in, token refresh, and logout processes, thereby ensuring the integrity and security of the user management system. By harnessing mocks for database interactions and bcrypt hashing, the tests are insulated from real database transactions, enabling a controlled testing environment that mirrors expected operational behaviors without actual data persistence. For instance, the tests examine the system's handling of duplicate email registrations, verify the correctness of generated JWT tokens upon user login, and assess the system's response to token refresh requests, amongst other functionalities. This approach not only safeguards the application against unauthorized access scenarios but also rigorously checks the system's response to both routine and exceptional user behaviors. Through strategic mocking of database responses and cryptographic operations, these tests offer a solid examination of the authentication workflows, highlighting any potential discrepancies in the authentication logic. This testing procedure is critical in preemptively identifying and rectifying any vulnerabilities or logic flaws, hence strengthening the application's reliability.

## **Levels of Testing**

**Unit Testing:** The backend testing, as seen in files like “test\_auth.py” and “test\_todo.py”, encompasses unit tests targeting individual modules or functionalities, such as the authentication processes and todo item management. These tests, by leveraging mocks for dependencies like database connections, aim to verify that specific backend functionalities perform as expected independently from the rest of the system. On the frontend side, the files ending in `.test.js` are geared towards unit testing of React components. These tests critically assess whether individual components behave correctly in isolation, including checks for correct rendering, responsive user input handling, and appropriate state management changes.

For Sprint 4, we plan on expanding test coverage to ensure all critical paths and edge cases are examined as paramount, alongside enhancing test isolation to guarantee that tests evaluate components in a vacuum, hence avoiding unintended dependencies. We also want to refactor tests for clarity and maintainability ensures that the test suite remains accessible and easy to update alongside application changes.

**Integration Testing:** On the backend, represented by files like “test\_auth.py” and “test\_todo.py”, integration tests go beyond individual unit tests by simulating the interaction between various modules, such as authentication processes and database operations. These tests validate the cohesive functionality of the system under test scenarios that mirror real-world usage, albeit with dependencies like databases being mocked. This ensures that different parts of the application work harmoniously to accomplish tasks like user registration and todo item management. In the frontend, the React component tests (`.test.js` files) subtly incorporate integration testing by examining how components interact with each other and with the overall application state. Through the React Testing Library, these tests assess not only the rendering and user interaction within individual components but also their ability to integrate and function within the larger application ecosystem, such as

handling API responses or user input events. This approach to testing underscores the importance of both isolated functionality and integrated performance, ensuring that all parts of the application operate cohesively to deliver a seamless user experience.

For Sprint 4, we plan to broaden the scope to encompass more comprehensive interactions between components, improving the simulation of production environments, and incorporating real database interactions can uncover more subtle integration issues.

**System Testing:** For Sprints 4, we plan on implementing system tests, which would involve testing the complete application, including real database interactions, through automated UI tests or API tests that hit the actual endpoints without mocking the database or internal logic. We plan on implementing End-to-End (E2E) testing tests in Sprint 4 to test the application as a whole, simulating real user scenarios from start to finish. This includes testing the integration between the frontend, backend, and any external services. We might use a tool such as Cypress, which is a powerful tool for E2E testing that can simulate real user interactions with our React application. Cypress tests can cover scenarios such as signing up, logging in, navigating through the app, and interacting with web elements.

**Automation:** Currently, our development process does not include the use of a Continuous Integration (CI) tool, such as Jenkins, Travis CI, or GitHub Actions, which would automate the execution of our tests upon every commit or merge request. During Sprint 4, we plan on selecting the most suitable automation tool for our web application's needs. This will facilitate early detection of flaws, and including performance benchmarks within tests.

## **Testing Approach:**

The testing strategy for this application is comprehensive, encompassing both backend and frontend to ensure robust coverage across all functionalities. On the backend, with files such as “test\_auth.py” and “test\_todo.py”, the approach leans heavily on unit testing to verify the correctness of individual modules, such as authentication and todo list management, in isolation. These tests are instrumental in validating the business logic and integration points within the application, employing mocks for external dependencies like databases to simulate real-world interactions without the overhead of actual database operations. This layer ensures that each component performs as expected under various scenarios, laying a solid foundation for application reliability.

Transitioning to the frontend, the testing regime utilizes the React Testing Library to focus on unit tests for individual React components, as demonstrated in the various `.test.js` files. These tests scrutinize components in isolation, checking for correct rendering, user interaction handling, and state management. Beyond unit testing, the frontend tests subtly weave in aspects of integration testing, examining how components interrelate and function within the larger application context, such as integrating with APIs and managing application-wide state changes.

Together, these testing approaches form a multi-layered defense, ensuring that both discrete functionalities and their interactions are thoroughly vetted. This strategy not only facilitates early detection of issues within isolated units but also safeguards against integration pitfalls, paving the way for a more reliable and maintainable application. Through diligent application of both unit and integration testing, the development team can

confidently evolve the application, knowing that each change is properly evaluated to maintain the high standards of quality and user experience.

### **Metrics and Coverage:**

For Sprint 3, we used coverage tools which are integrated with our testing frameworks to measure and report coverage. For instance, to assess our front end coverage using Jest, we incorporated a test coverage script, named "test:coverage", into our package.json file. This script generates a coverage report, including metrics like statement, branch, function, and line coverage.

All metrics and coverage for the front end and the back end testing can be found in the Sprint 3 Testing Report on page 58 of this documentation.

In Sprint 4, we aim to increase our coverage % as much as possible.

### **Acceptance Tests:**

At this point in the development process, no acceptance tests were developed yet in order to validate that the system performs as expected from an end-user perspective. In future sprints, acceptance tests will cover a wide range of scenarios for our website, including user registration, property and financial management functionalities, and the use of common facilities, to simulate real-world use effectively. We might use Behavior-Driven Development (BDD) tools like Cucumber or SpecFlow to write acceptance tests in a language that is understandable, however it has not been decided yet.

### **Implementation Plan:**

As our project progresses through its development phases, we will integrate testing into the development workflow by setting up a CI pipeline in Sprint 4, as mentioned previously, that runs tests automatically on every commit. This will ensure that the pipeline includes steps for running unit tests, integration tests, and system tests. Additionally, we will schedule regular review sessions to assess test coverage and effectiveness, adjust testing strategies as needed, and ensure that testing keeps pace with development. Lastly, we will attempt to encourage collaboration between the teammates in charge of both development and testing to ensure that testing reflects the user's needs and project requirements.

# Code Management

## Project Overview:

Sprint 3 focused on updating the front-end pages and working on the backend to enhance the functionality and user experience of the platform. These pages include:

- Registration key page
- Homepage
- Condo owners dashboard
- Condo management company profile for property under their management
- Financial system
- Reservation system
- Notifications
- Employee page

## Code Management Approach:

In Sprint 3, we adopted a systematic code management strategy to ensure efficient collaboration and maintain high code quality. Each team member was responsible for developing specific pages within the project. 7 team members were working on the front end of the pages and 3 team members were working on the backend of the website. The majority of the front-end pages were already done during Sprint 2, therefore we updated them for a better user experience if necessary.

## Workflow Overview:

### *Frontend Development*

Upon completion of frontend development for a page, the respective team member created a pull request on our GitHub repository.

### *Pull Request Management*

Two designated team members oversaw the pull request process, ensuring the integration of frontend pages into the website and verifying functionality.

### *Backend Development*

Following successful frontend integration, backend development commenced to implement the necessary functionality.

### *Issue Resolution*

In cases of encountered problems, team meetings were held to discuss and find appropriate solutions. Our team will also start using the 'Issues' functionality of Github for future problems.

## *Branch Management*

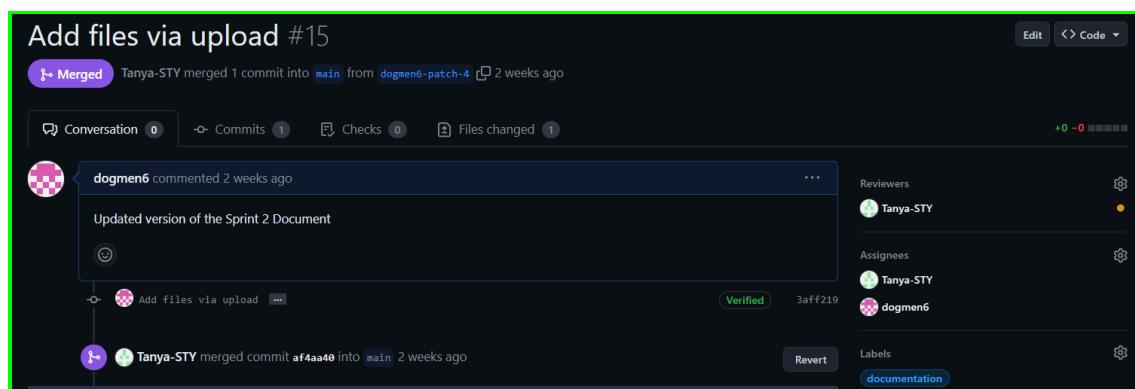
Our repository was organized with individual branches for each team member, facilitating clear tracking of contributions and merging branches for integration.

## **Code Quality and Management Metrics:**

### ***Quality of Source Code Reviews***

During Sprint 3, our team implemented a robust peer code review process. This involved team members thoroughly examining each other's code to identify potential issues, provide feedback, and ensure adherence to project standards. By conducting rigorous code reviews, we aimed to maintain high code quality, improve overall understanding of the codebase, and foster knowledge sharing among team members.

To ensure comprehensive reviews, we enforced the practice of assigning reviewers to each pull request and issues. Assigning reviewers helped distribute responsibility for code inspection and ensured that multiple perspectives were considered during the review process. Additionally, we designated assignees to issues to clarify ownership and accountability for resolving specific tasks or bugs. By linking reviewers and assignees to pull requests and issues, we fostered a collaborative environment where feedback could be exchanged effectively, leading to improved code quality and faster resolution of issues.



**Figure 25**

### ***Correct Use of Design Patterns***

Design patterns were strategically employed throughout the development process to enhance code modularity and scalability. By leveraging established design patterns such as MVC (Model-View-Controller), Singleton, Factory, and Observer, we aimed to promote code reuse, improve maintainability, and facilitate future enhancements.

Additionally, we used the Component-Based Design Pattern, given our use of React and the creation of modular components. Examples of our components include the Registration Key Page, Homepage, Condo Owners Dashboard, Condo Management Company Profile for Property Under Their Management, Financial System, Reservation System, Notifications, and Employee Page. This approach not only ensured the efficient development of our web application but also laid the foundation for a scalable and easily maintainable system.

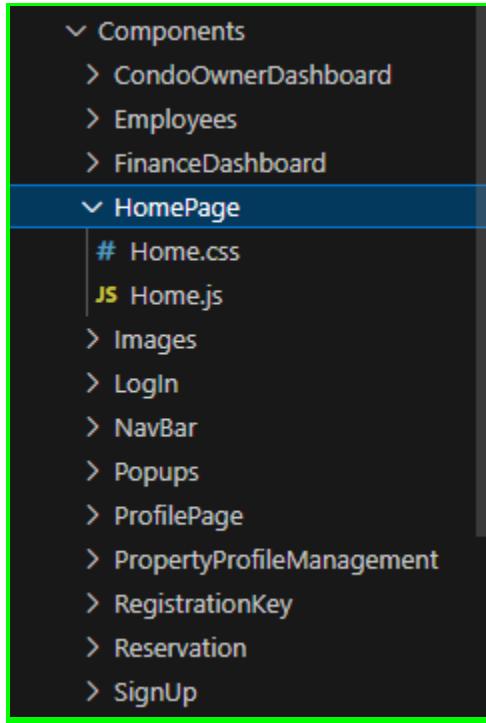


Figure 26

### Respect to Code Conventions

Adherence to established code conventions was a cornerstone of our development approach. We enforced consistent coding styles, naming conventions, and formatting guidelines across the project to ensure readability and maintainability. By maintaining uniformity in coding practices, we aimed to minimize confusion, streamline collaboration, and promote code comprehensibility for all team members.

For example, in our components, such as the RegistrationKey component, we consistently apply camelCase naming conventions for variables and functions, maintain a clear folder structure, and use descriptive class names and comments to improve code readability and maintainability. This adherence to conventions ensures that our codebase remains cohesive and easy to understand, promoting efficient development and scalability.

```
<h1 className="welcomeTitle">Welcome to URBANKEY.</h1>
<div className="houseImage">
  <img src= {houseImage} alt="House Image"/>
</div>

<div className="registrationText">
  To register as a condo owner or rental user, you need a registration key provided by your condo management company.
  <br/>
  If you have not received your key or need assistance, please contact support.
</div>
```

Figure 27

### Design Quality

Emphasis was placed on optimizing the design of our codebase to enhance maintainability and extensibility. We carefully evaluated factors such as the number of classes/packages, code size, coupling, and cohesion to ensure

a well-structured and scalable architecture. By maintaining a balance between granularity and cohesion, we aimed to facilitate easier navigation, debugging, and future enhancements of the codebase.

#### Code Metrics:

Code metrics offer quantitative assessments of different facets of code quality, aiding developers in gauging the efficacy and maintainability of their codebase. Metrics like cyclomatic complexity, essential complexity, lines of code, and code duplication serve as indicators, pinpointing areas within the codebase that necessitate enhancement.

Cyclomatic complexity quantifies the number of linearly independent paths through a program's source code. In simpler terms, it measures how many unique paths there are through a program's code, which can indicate its overall complexity and potential difficulty to understand, maintain, and test.

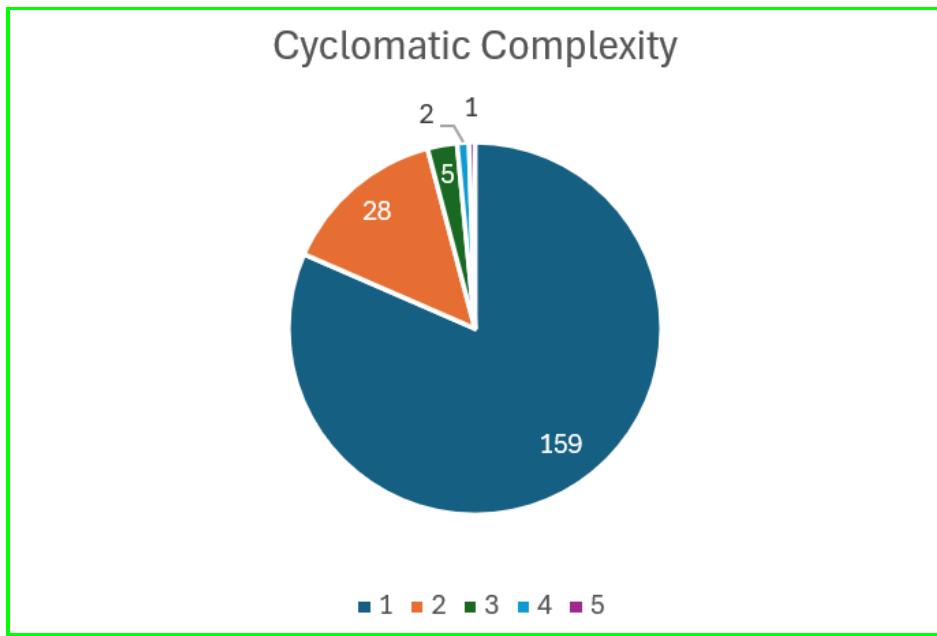
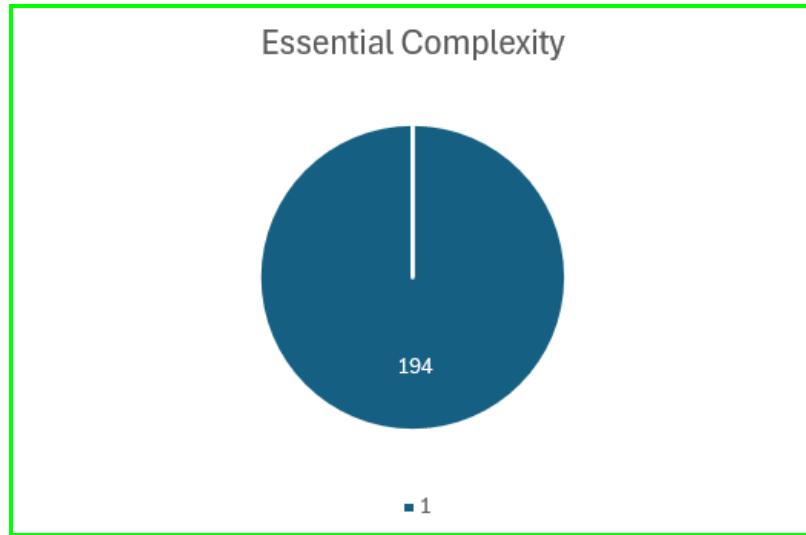


Figure 28

As we can see in the above graph, the majority of our code's cyclomatic complexity is of 1. This indicates that there are no decision points such as if statements, loops, or switch statements present in the code. It also means that it is easy to understand and maintain because there are no branching conditions or loops to complicate the logic.

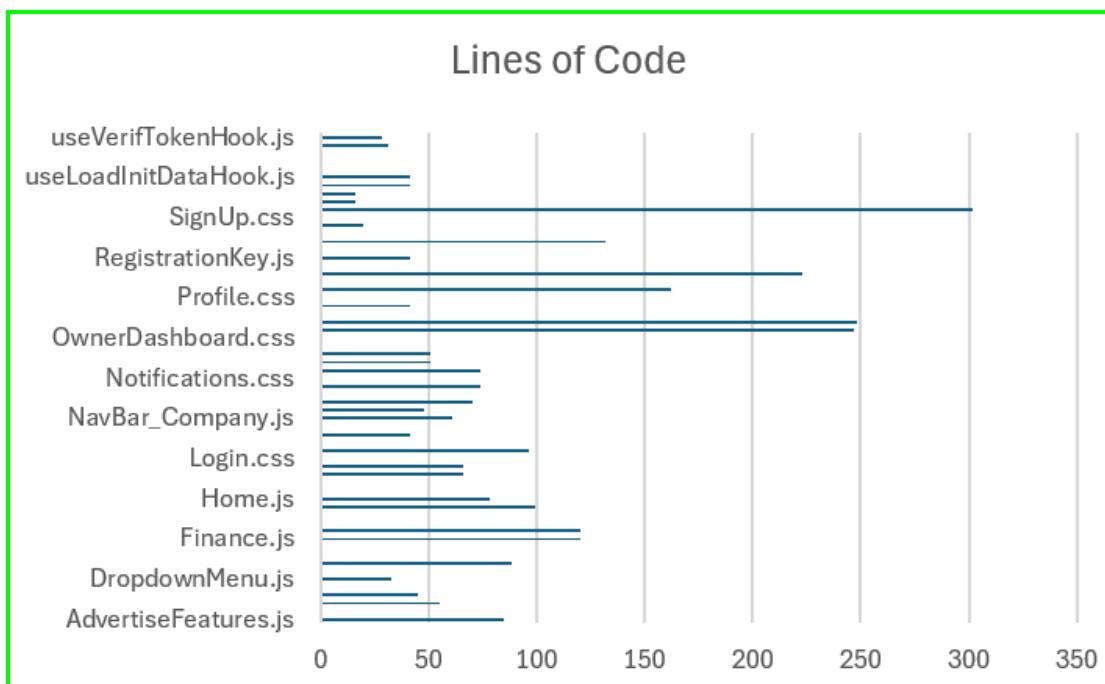
Essential complexity is a metric aimed at assessing the structural quality of a program. It evaluates the count of entry points, termination points, and non-deductible nodes. A value closer to 1 indicates a better structured program.



**Figure 29**

As we can see in the above graph, the essential complexity of the code is 1. This demonstrates that our project is overall well structured.

Lines of code is a simple and commonly used metric in software development that quantifies the size of a program by counting the number of lines in the source code. This metric includes all lines of code, including comments and blank lines, and provides a rough estimate of the overall complexity and scale of a software project.

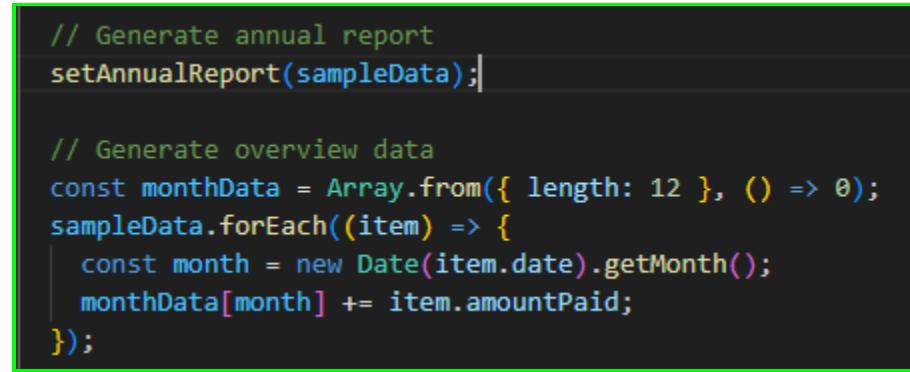


**Figure 30**

As we can see in the above graph, most of LOCs in our project files are below 100, meaning that the complexity is low. This also means that the project is generally easier to maintain and to review.

## Source Code Documentation

Comprehensive documentation accompanied our source code to provide insights into its functionality, usage, and implementation details. Most of the code documentation we used is in the form of inline comments next to our code. This documentation served as a valuable resource for understanding the codebase, facilitating future development efforts, and onboarding new team members. By documenting key aspects of our code, we aimed to promote clarity, reduce ambiguity, and enhance maintainability throughout the project lifecycle.



```
// Generate annual report
setAnnualReport(sampleData);

// Generate overview data
const monthData = Array.from({ length: 12 }, () => 0);
sampleData.forEach((item) => {
  const month = new Date(item.date).getMonth();
  monthData[month] += item.amountPaid;
});
```

Figure 31

## Refactoring Activity

Refactoring activities were systematically documented in commit messages to track code improvements and optimizations. Whenever code refactoring was performed to enhance readability, performance, or maintainability, detailed explanations were provided in commit messages. By documenting refactoring activities, we aimed to maintain transparency, communicate rationale behind code changes, and ensure continuous improvement of the codebase.



Figure 32

## Commit Message Quality

Commit messages were crafted to be detailed and informative, providing context and rationale for each code change. Clear and concise commit messages helped team members understand the purpose and impact of code modifications, facilitating smoother code review processes and enhancing overall collaboration. By adhering to

high standards of commit message quality, we aimed to improve traceability, code comprehension, and collaboration efficiency.

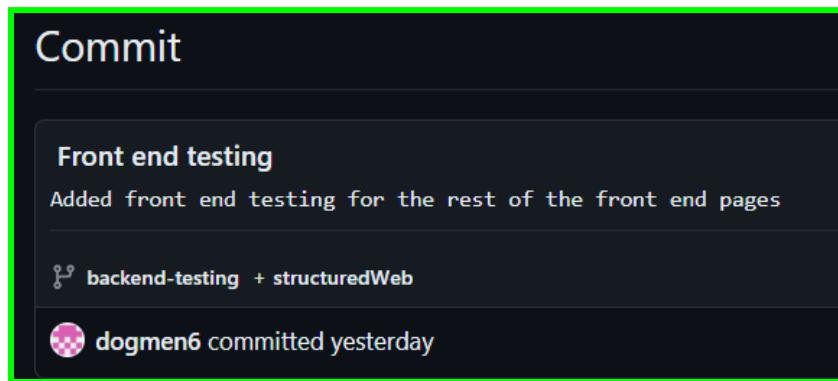


Figure 33

### Use of Feature Branches

The use of feature branches is a common practice in software development, and it plays a crucial role in our team's workflow. Each team member is assigned their own feature branch to work on a specific task or feature independently, allowing them to develop and test their code in isolation without affecting the main codebase. This approach promotes parallel development and minimizes conflicts when multiple developers are working on different parts of the project simultaneously. Additionally, we maintain a structured branch where we merge all completed features or pages to ensure compatibility and integration before merging them into the main branch. This structured branch serves as a staging area where we can review and test the combined functionality of different features. Furthermore, we utilize a separate backend branch to manage all backend code, keeping it separate from the frontend development. This separation of concerns helps streamline development and ensures that changes to the frontend and backend are managed independently. Overall, the use of feature branches in our workflow enhances collaboration, reduces conflicts, and promotes a systematic approach to development.

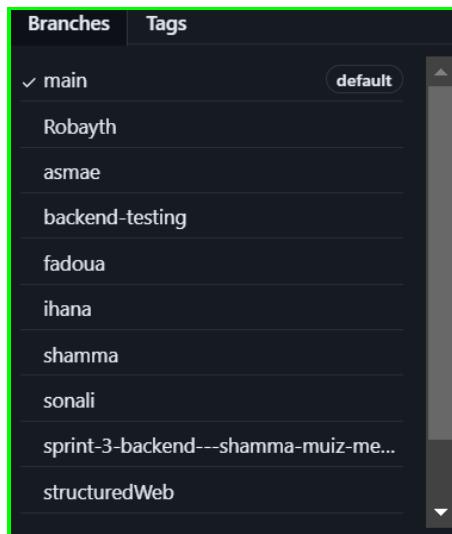
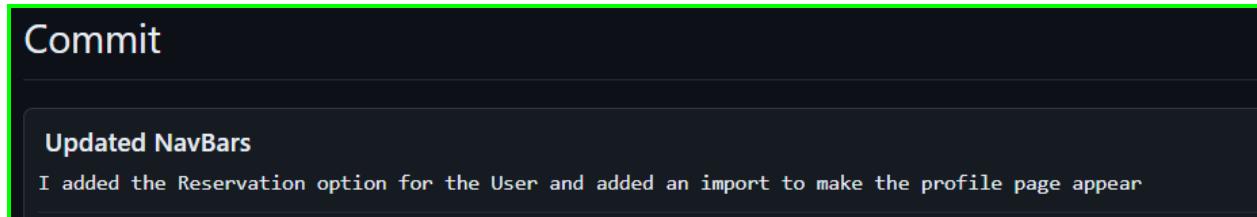


Figure 34

## *Atomic Commits*

Commits were kept atomic, focusing on a single logical change to enhance code review and facilitate rollback if necessary. By breaking down changes into smaller, self-contained units, we aimed to improve code traceability, reviewability, and maintainability. Atomic commits also facilitated easier identification and resolution of issues, contributing to overall code quality and stability.



**Figure 35**

## *Bug Reporting*

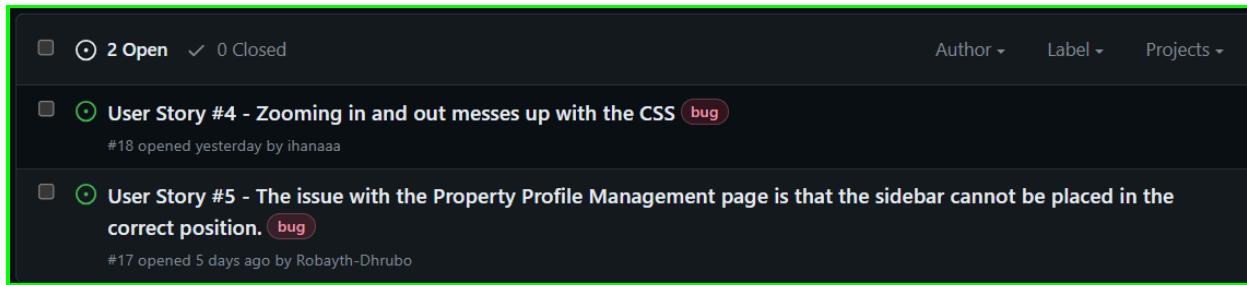
If we encounter any issues or bugs, we systematically reported and tracked them using GitHub's issue-tracking system. By utilizing GitHub's issue-tracking system, we aim to ensure timely resolution of bugs, improve software reliability, and enhance user satisfaction.

ID	User Story	Description	Link to Bug	Status
#17	#5	The issue with the Property Profile Management page is that the sidebar cannot be placed in the correct position	<a href="https://github.com/Tanya-STY/Urban-Key/issues/17">https://github.com/Tanya-STY/Urban-Key/issues/17</a>	Not Started
#18	#4	Zooming in and out messes up with the CSS	<a href="https://github.com/Tanya-STY/Urban-Key/issues/18">https://github.com/Tanya-STY/Urban-Key/issues/18</a>	Not Started

**Table 2: Bug Report**

## *Use of Labels for Issues/Pull Requests*

In Sprint 3, we utilized issue labels for tracking and filtering tasks. By categorizing tasks and pull requests using appropriate labels such as bug, enhancement, feature, or priority level, we aim to prioritize effectively, allocate resources efficiently, and provide clear visibility into the status of ongoing development efforts. It also makes our Github repository more organized for all team members.



**Figure 36**

### *Links Between Commits and Issues*

By systematically linking commits to relevant bug reports or features, we aim to enhance traceability, accountability, and collaboration within the development team. This approach will facilitate easier navigation, contextual understanding, and efficient management of code changes.

### **Conclusion:**

The code management process employed during Sprint 3 ensured effective collaboration, maintained code quality, and facilitated the timely delivery of project milestones. By adhering to established practices and leveraging collaborative tools, our team successfully navigated the complexities of development, paving the way for future iterations and enhancements of the condominium management website.