



## **SOEN 390 - Sprint 2 Documentation**

**March 4th, 2024**

**Department of Computer Science & Software Engineering**

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

**Concordia University**

**Winter 2024**

# 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

### 1.1. 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.

### 1.2. 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 repaires</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 Stakeholder's 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>	

**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>	

**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>	

**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>	

**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>	

**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>	

**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>	

*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>	

*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>	

*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>	

*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 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 Urbankey.

### 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 second to ensure a seamless experience. And system uptime must be maintained at 99.9% to minimize service disruptions, with downtime limited to schedules 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 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.

# 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.

## *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.

## *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 availabilities 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 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						
User Profile Feature						
Must	Bus. Value: M	Risk: M	Effort: 5			

## 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						
Registration Key Feature						
Must	Bus. Value: H	Risk: M	Effort: 6			

## User Story 3

003	Becoming a Rental User
<p><b>As</b> a public user</p>	
<p><b>I want</b> to use a registration key from the management company</p>	
<p><b>To</b> securely become a rental user in the system.</p>	
Registration Key Feature	
Must	Bus. Value: H
Risk: M	Effort: 6

## User Story 4

004	Condo Owner Profile
<p><b>As</b> a condo owner</p>	
<p><b>I want</b> a comprehensive dashboard displaying property and financial details</p>	
<p><b>Because</b> it allows for efficient property management.</p>	
Condo Owner Dashboard Feature	
Must	Bus. Value: H
Risk: M	Effort: 8

## User Story 5

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

## User Story 6

006	File Upload for Condo Management Companies					
<b>As a property manager</b>						
<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			

## User Story 7

007	Condo Management Company Information					
<b>As a property manager</b>						
<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			

## User Story 8

008	Generate and send registration code					
<b>As a property manager</b>						
<b>I want to</b> 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			

## User Story 9

009	Enter condo unit fee					
As a property manager						
I want to enter condo fees per square foot and per parking spot.						
To accurately calculate the fee for each condo unit.						
Must	Bus. Value: M	Risk: H	Effort: 3			

## User Story 10

010	Calculate condo unit fee					
As a property manager						
I want to calculate and present the fee for each unit.						
To present it to the unit owner.						
Condo Fee Calculation Feature	Must	Bus. Value: M	Risk: M			
			Effort: 8			

## User Story 11

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

## User Story 12

012	Generate annual report			
<b>As a financial system agent</b>				
<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.				
<b>Annual Report Feature</b>				
Should	Bus. Value: M	Risk: H	Effort: 13	

## User Story 13

013	Set up facility reservation system			
<b>As a property manager</b>				
<b>I want to</b> set up a system for managing reservations of common facilities				
<b>To</b> allow residents can schedule and access these amenities.				
<b>Facility Reservation Feature</b>				
Must	Bus. Value: M	Risk: M	Effort: 8	

## User Story 14

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

## 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			

## 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			

## 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			

## User Story 18

018	Submission of Requests		
As 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

## User Story 19

019	Request Assignment		
As 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

## User Story 20

020	Notification Page		
As 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

## User Story 21

021	Forum posts and replies					
<b>As a public user</b>						
<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						

## User Story 22

022	Organize Events					
<b>As a public user</b>						
<b>I want</b> to be able to organize events and invite other occupants						
<b>because</b> I want to interact with neighbors						

## User Story 23

023	Coupon Listing					
<b>As a property manager</b>						
<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						

## User Story 24

024	Multiple Platform Access					
<b>As a public user</b>						
<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						
Should	Bus. Value: M	Risk: M	Effort:6			

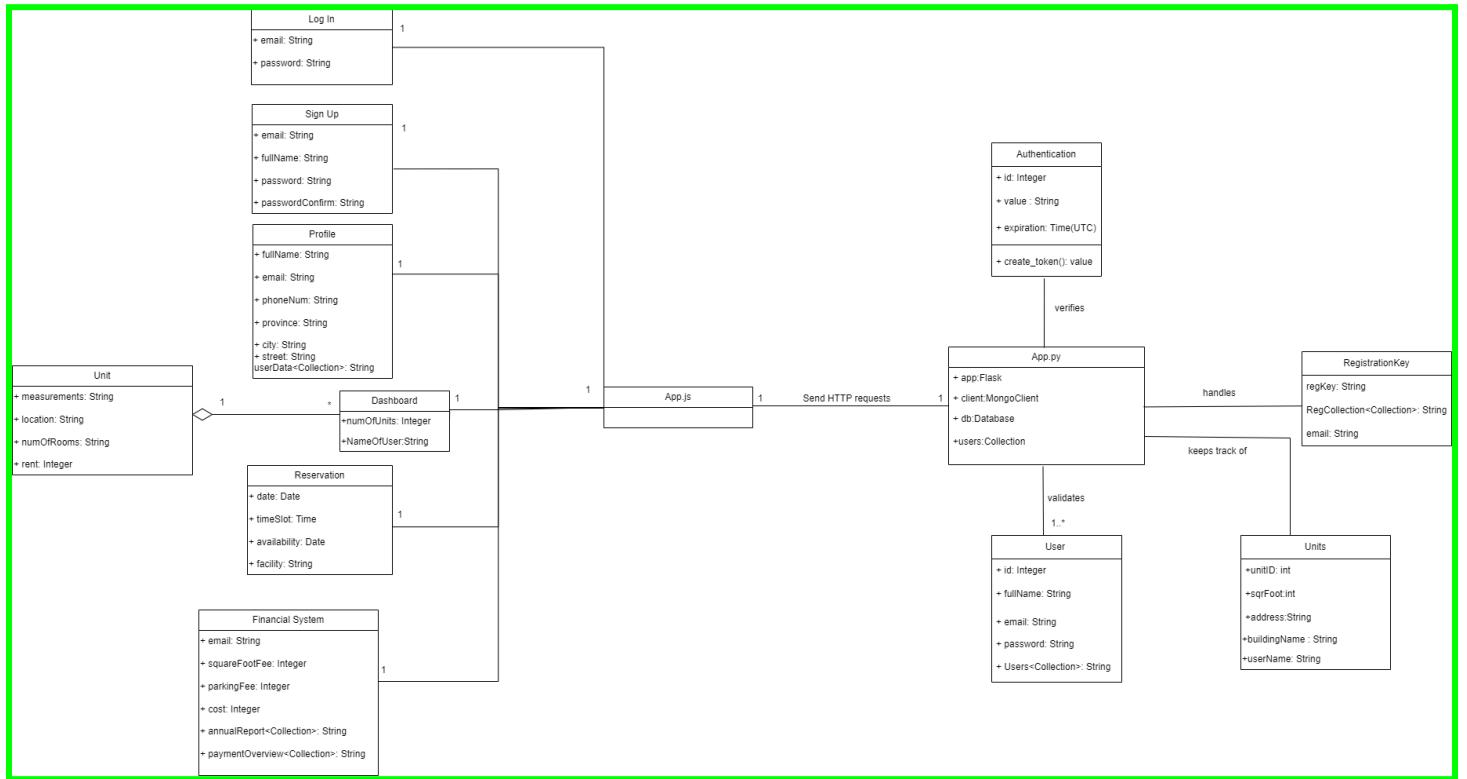
## User Story 25

025	Language Options					
<b>As a public user</b>						
<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		

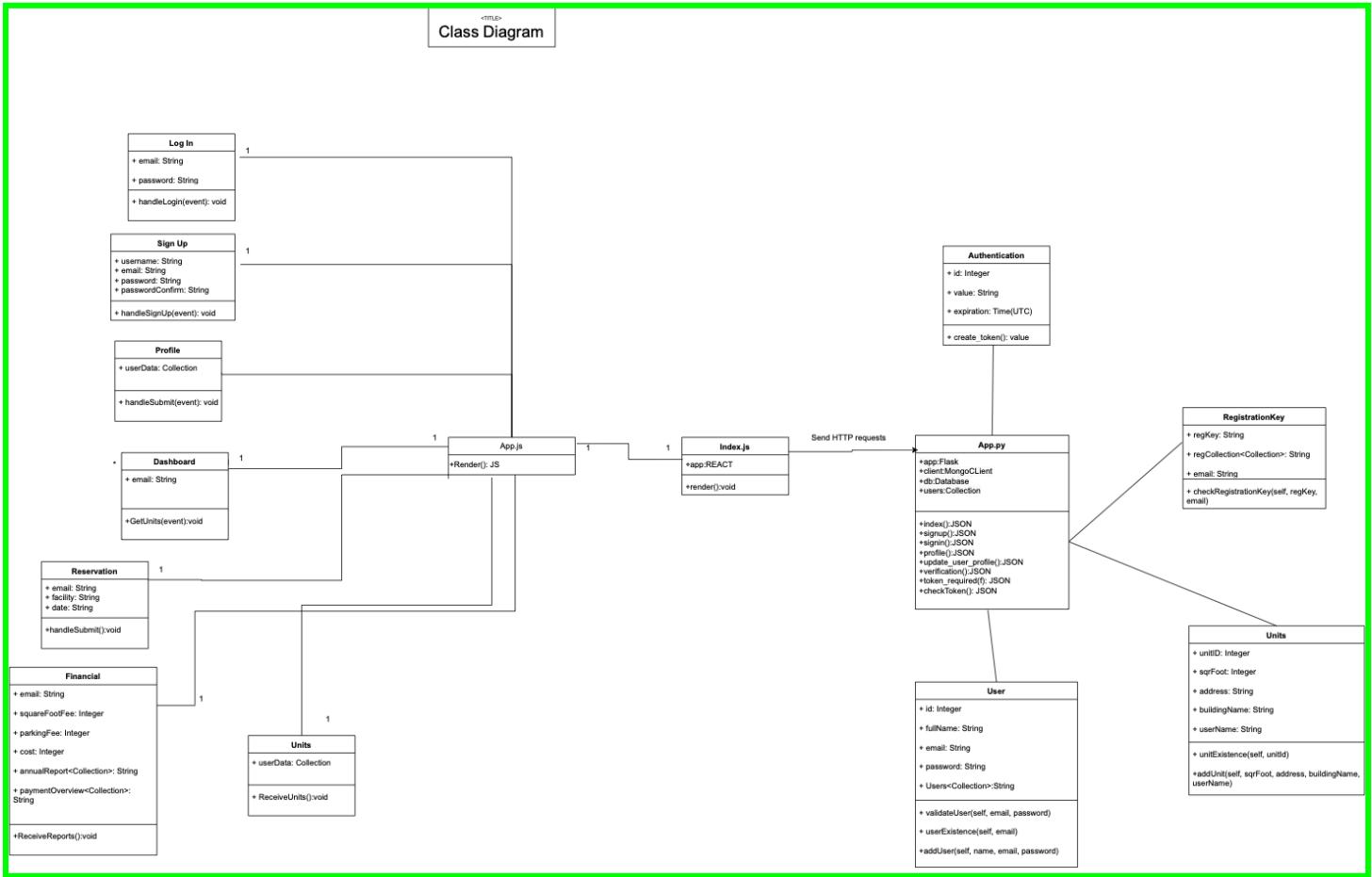
## User Story 26

026	Sign in Options					
<b>As a public user</b>						
<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		

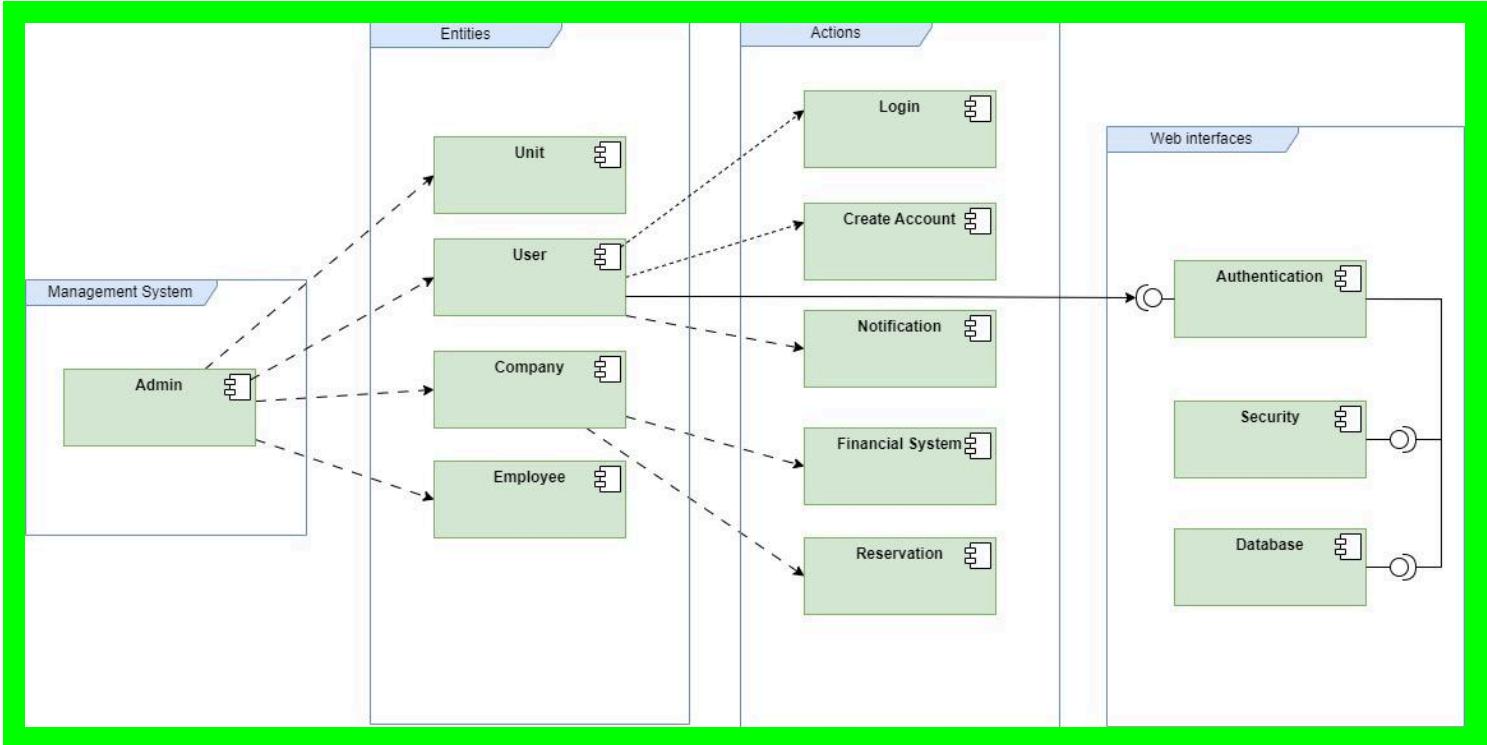
# SAD



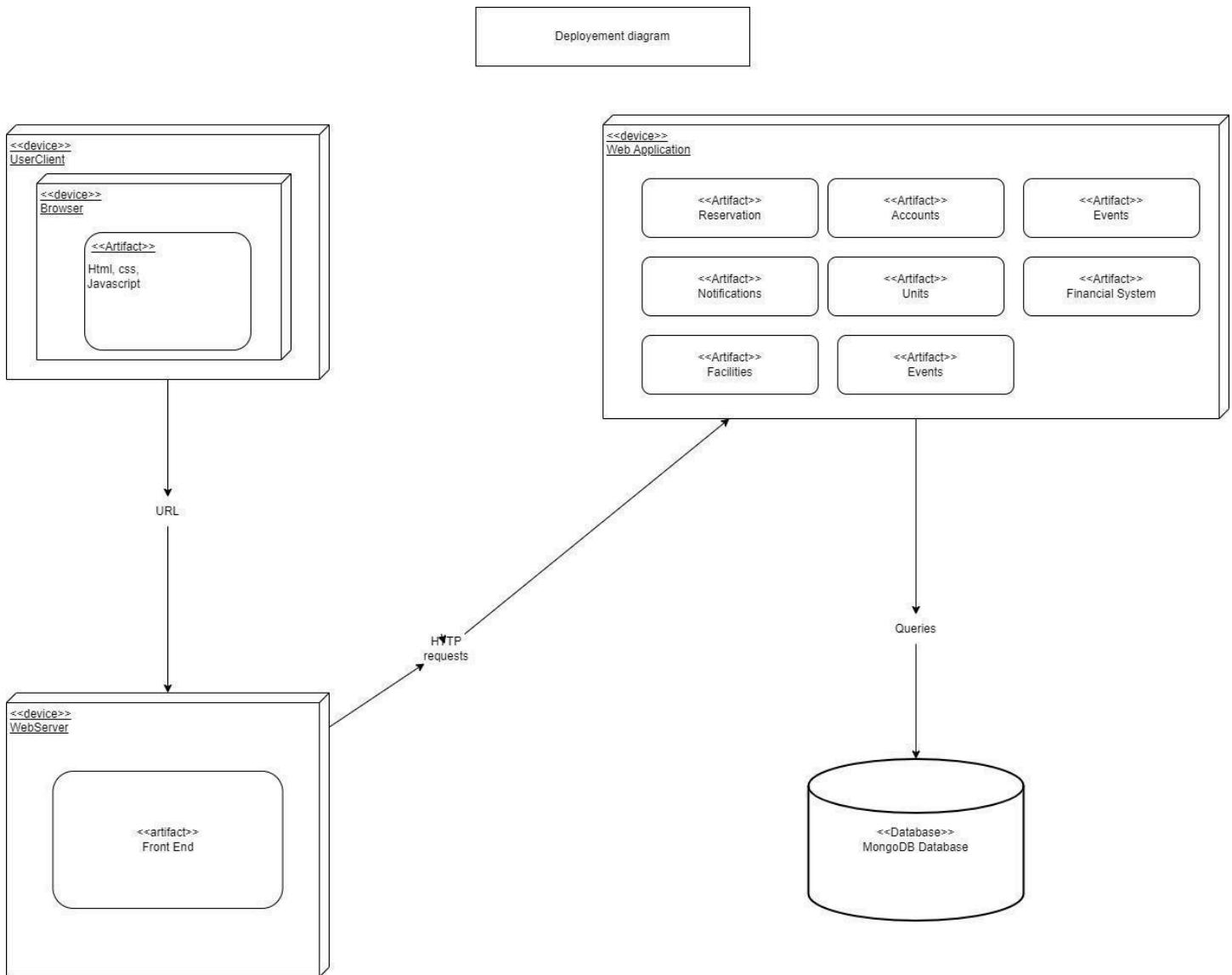
## Domain Model



## Class Diagram



## Component Diagram

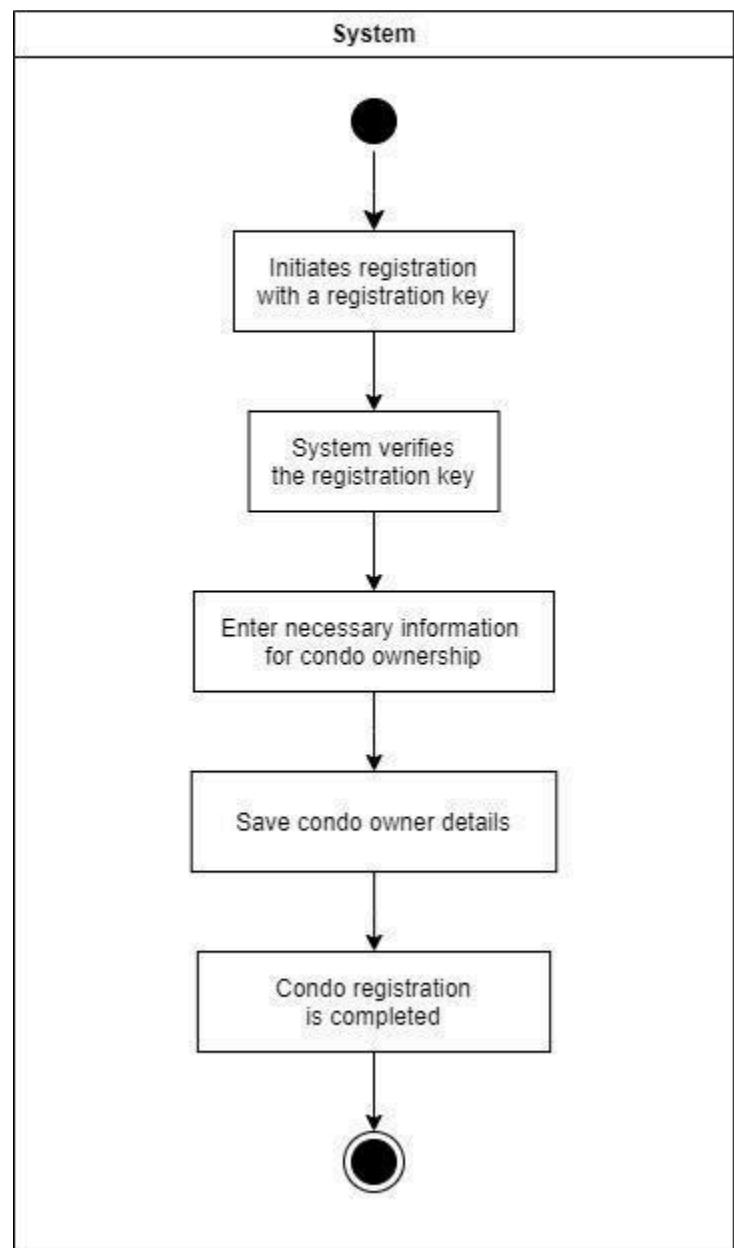
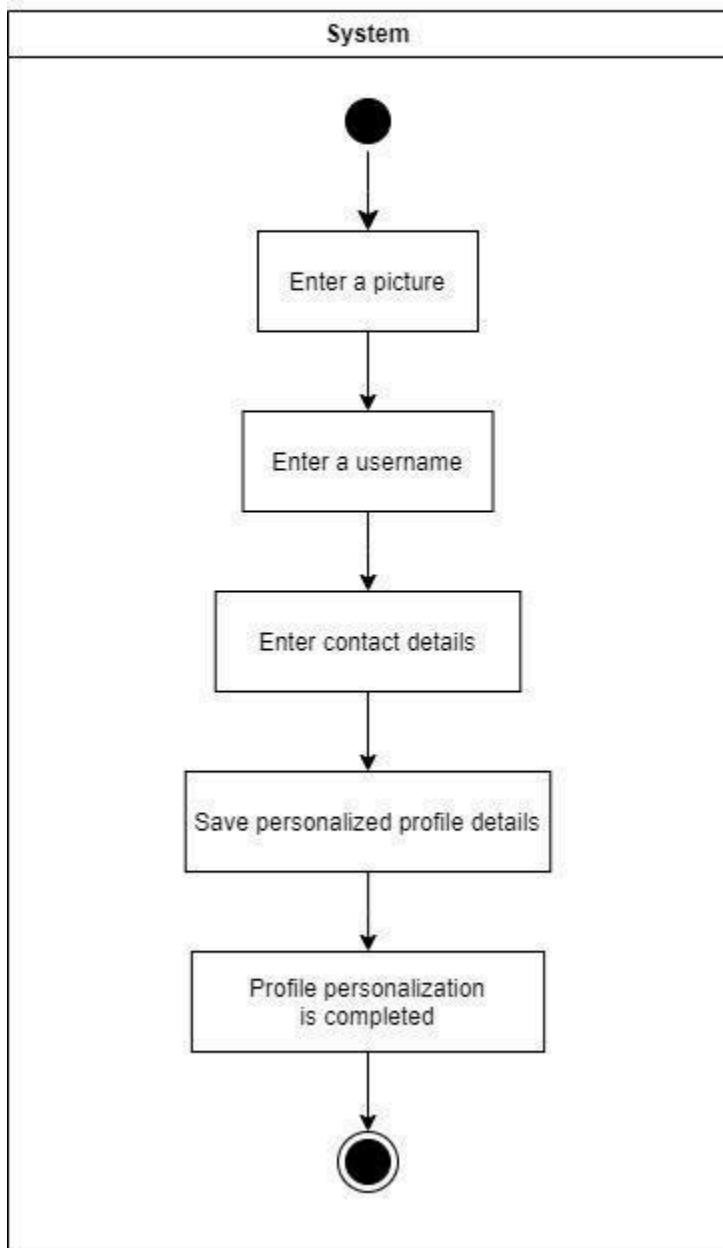


## Deployment Diagram

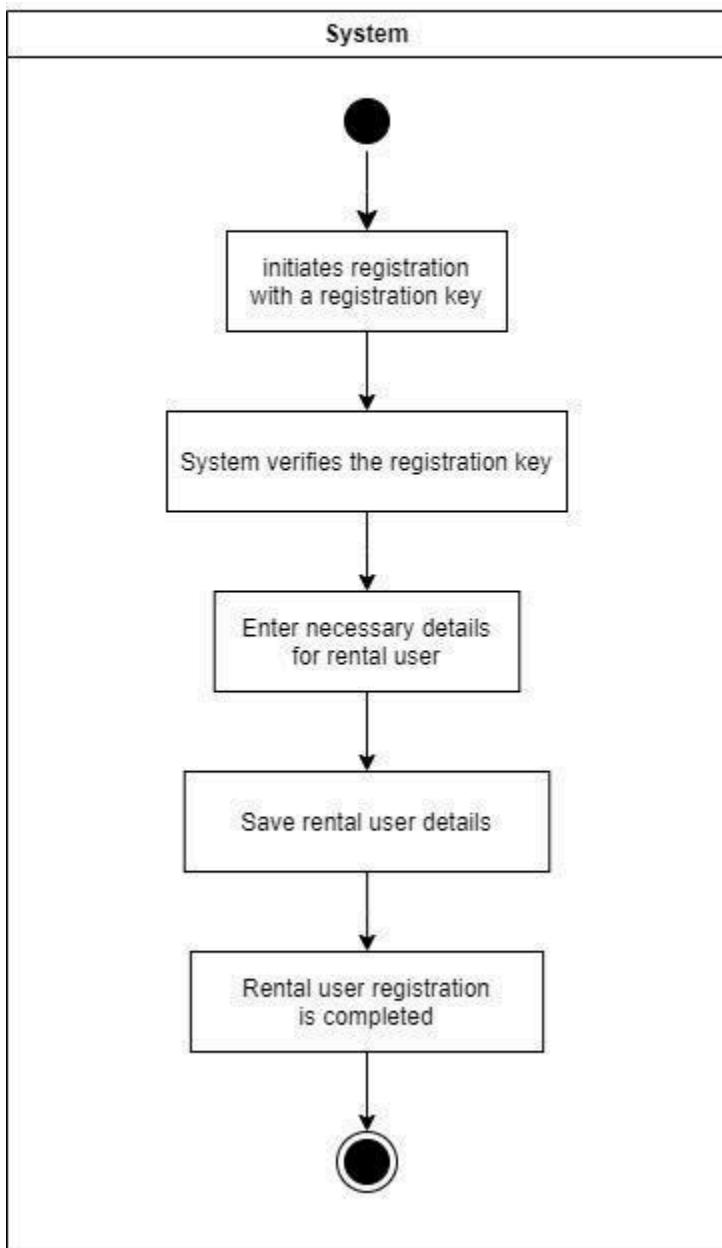
## Activity Diagrams

ID: 001

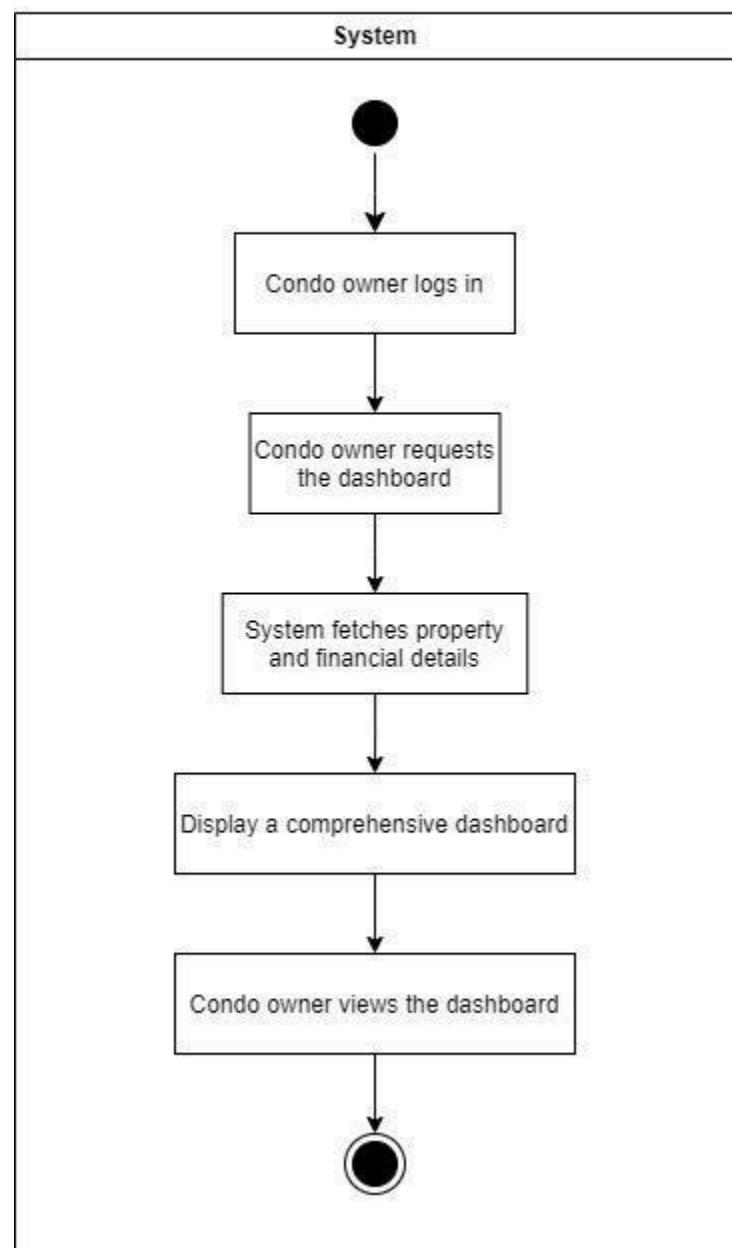
ID: 002



ID: 003

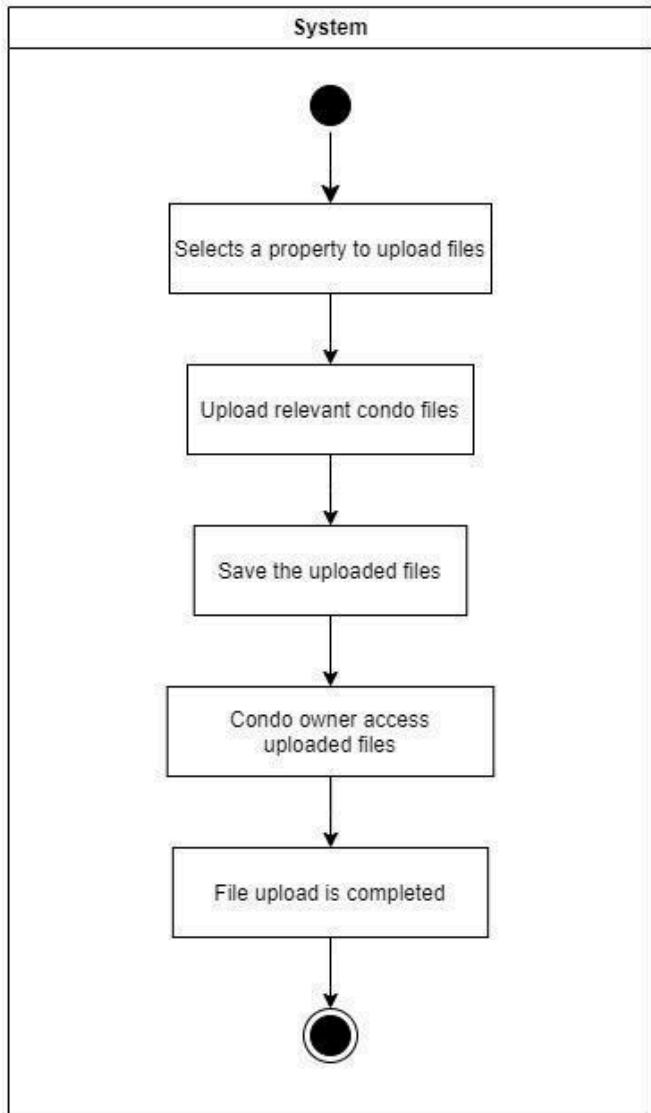
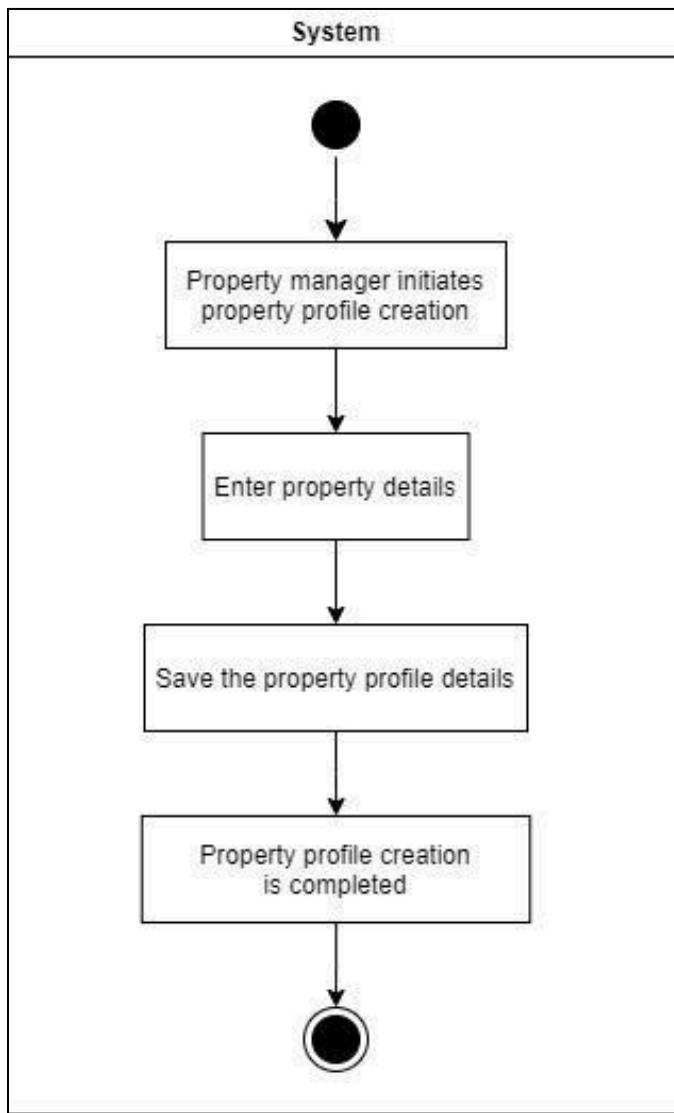


ID: 004

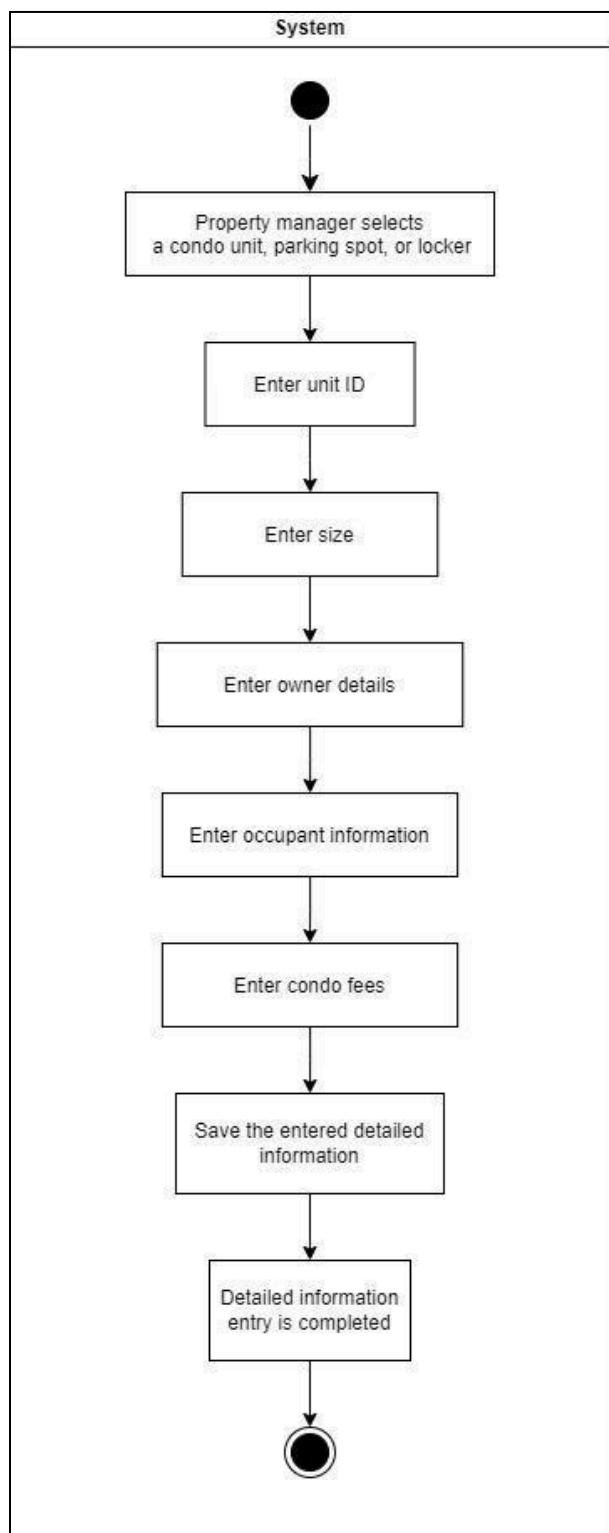


ID: 005

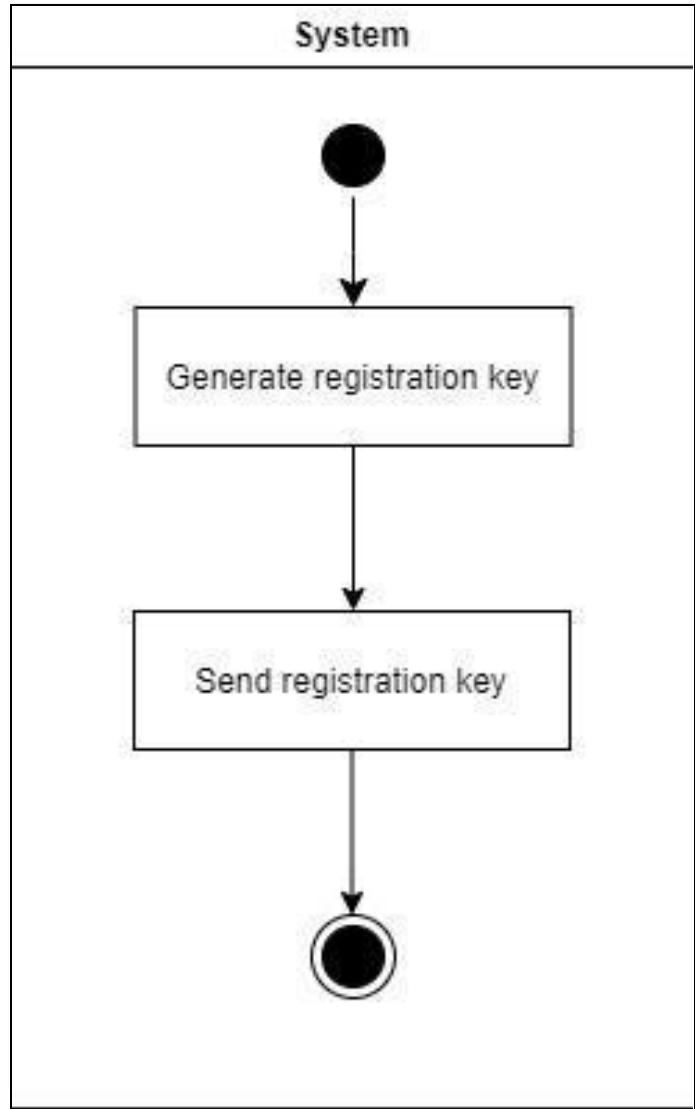
ID: 006



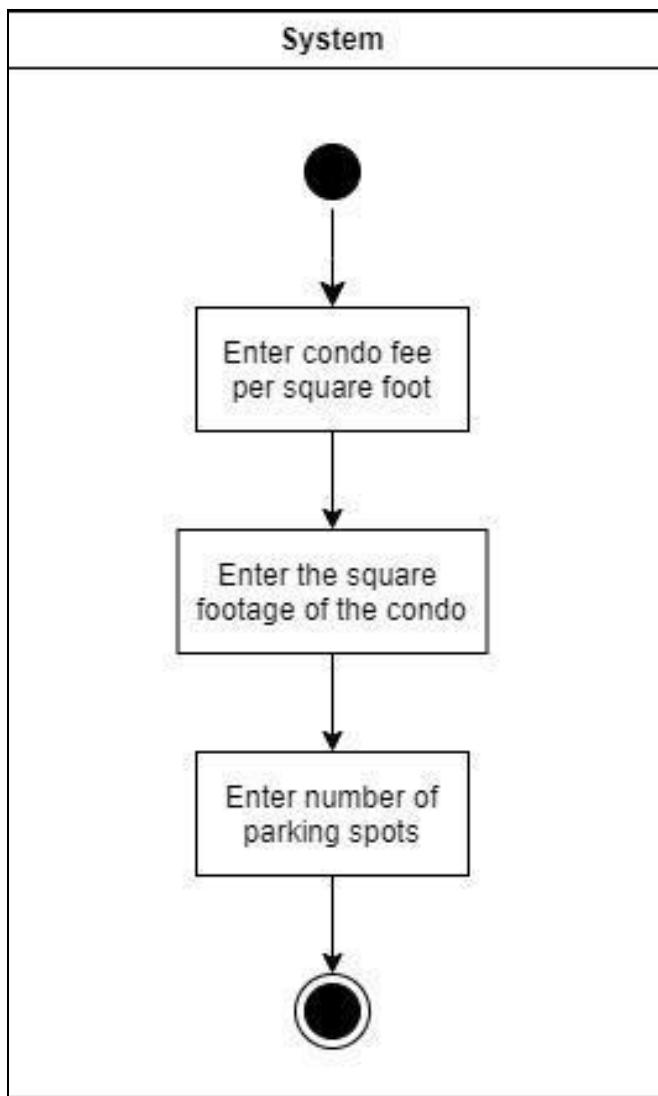
ID: 007



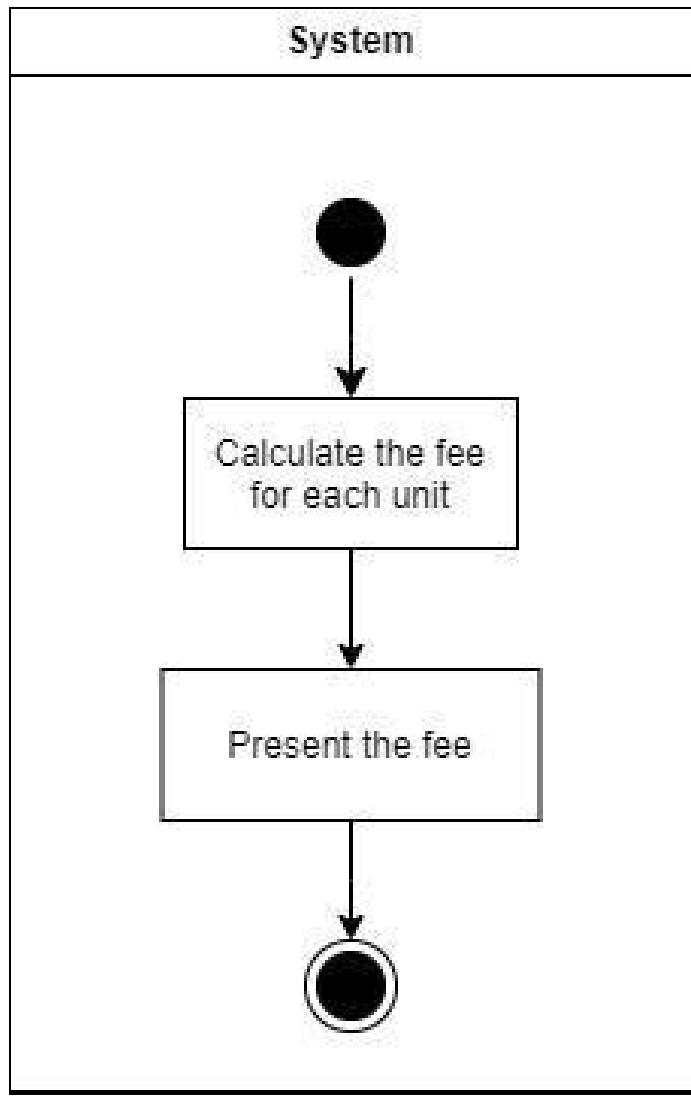
ID: 008



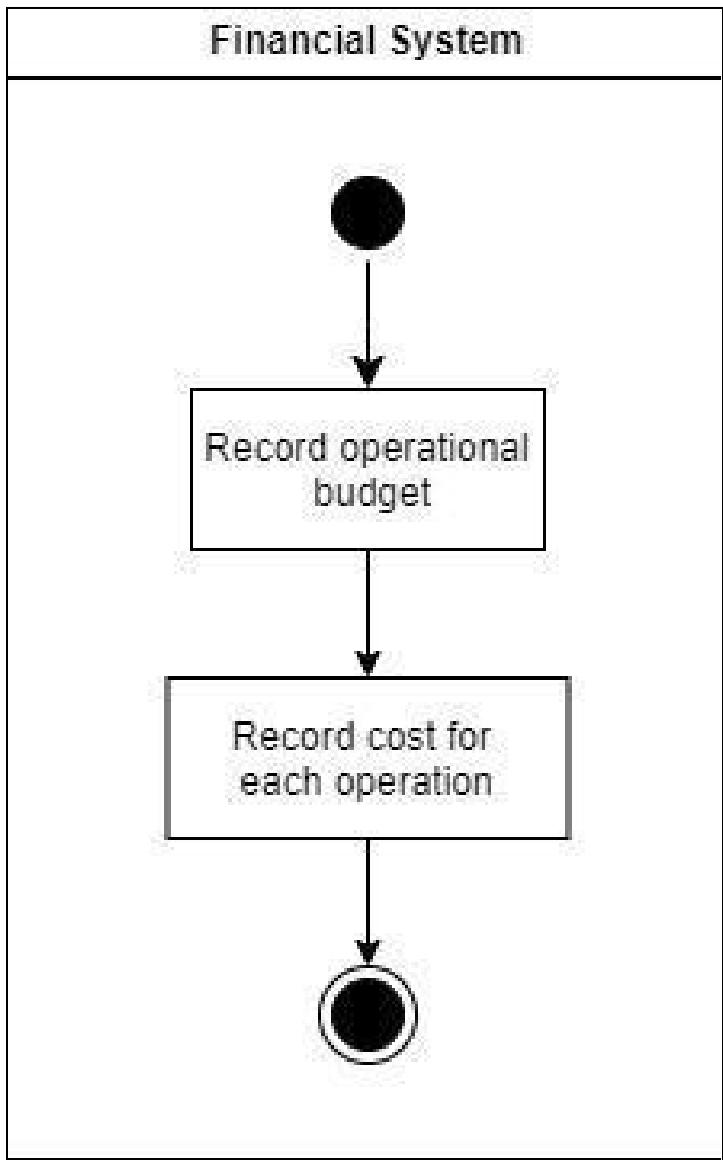
ID: 009



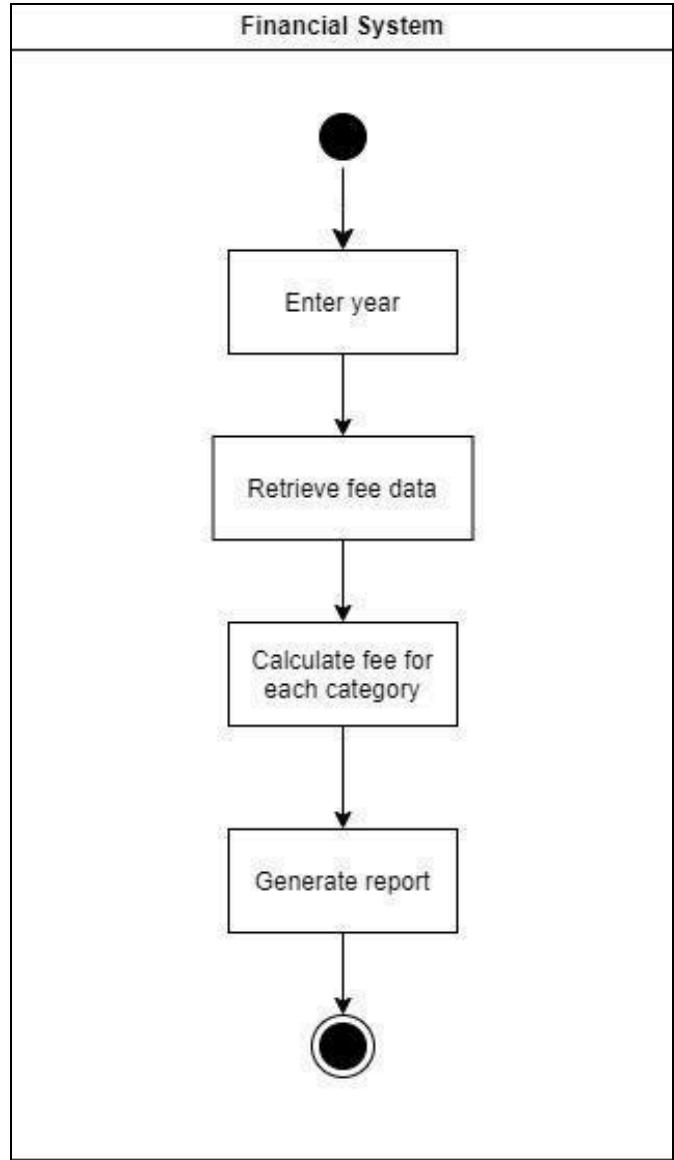
ID: 010



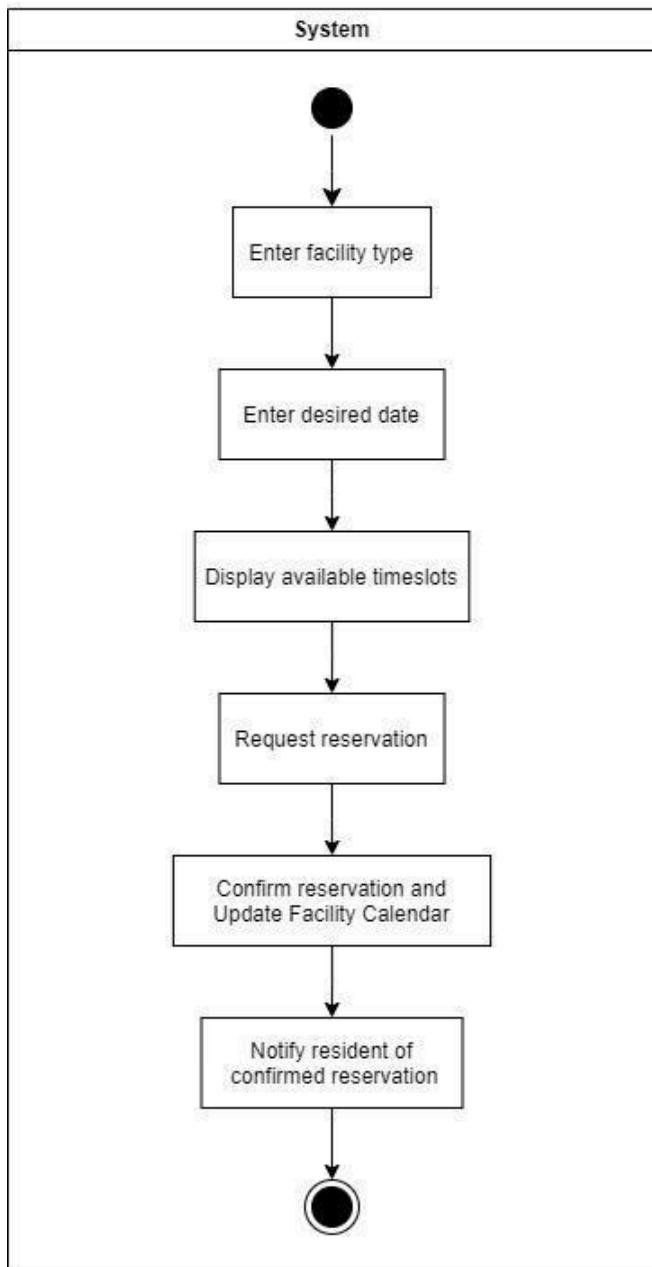
ID: 011



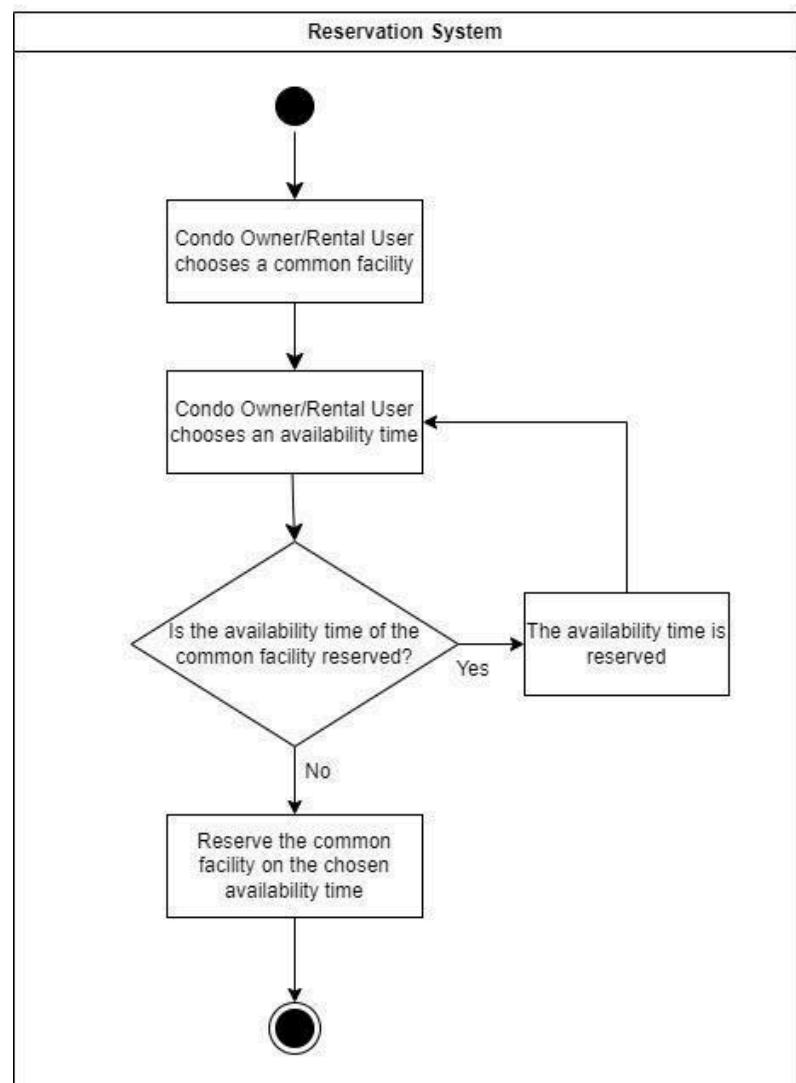
ID: 012



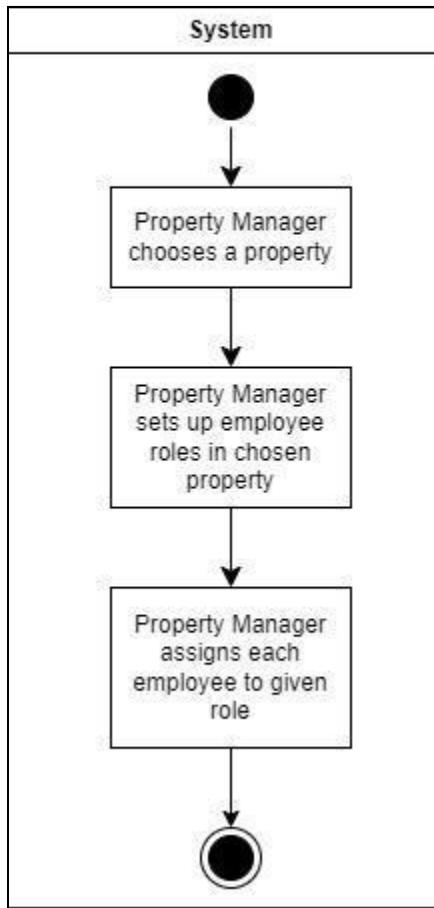
ID: 013-014



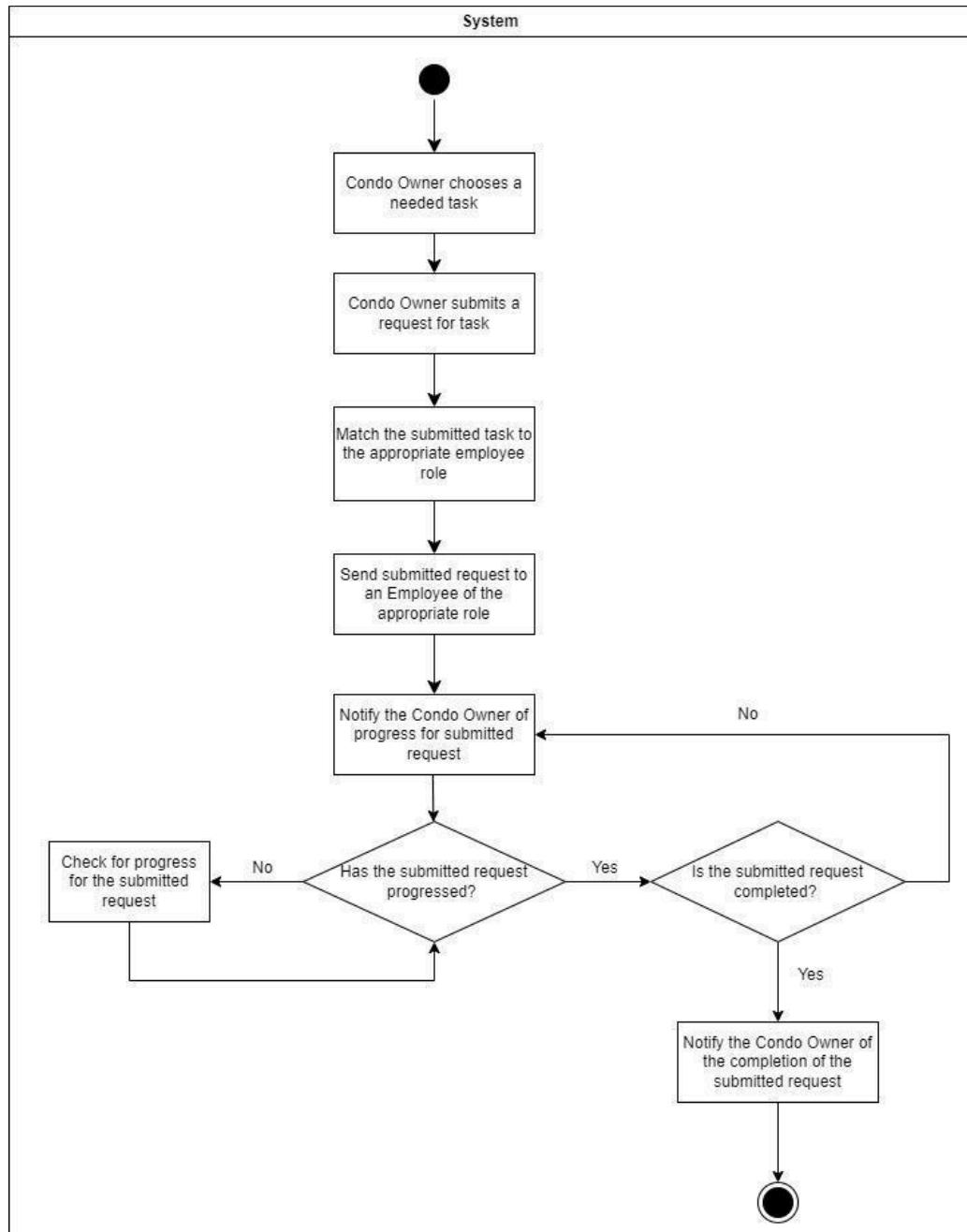
ID: 015-016



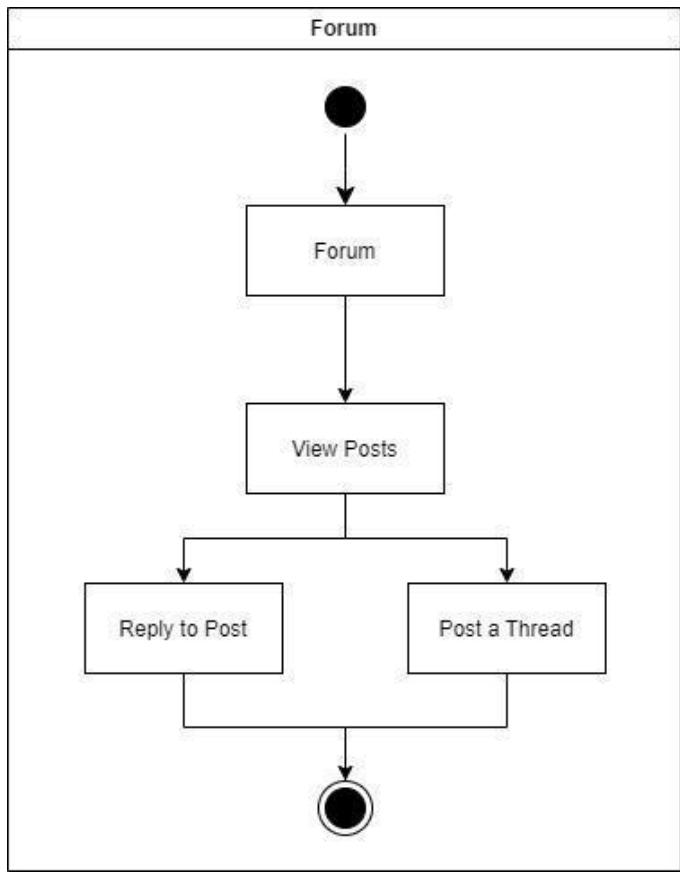
ID: 017



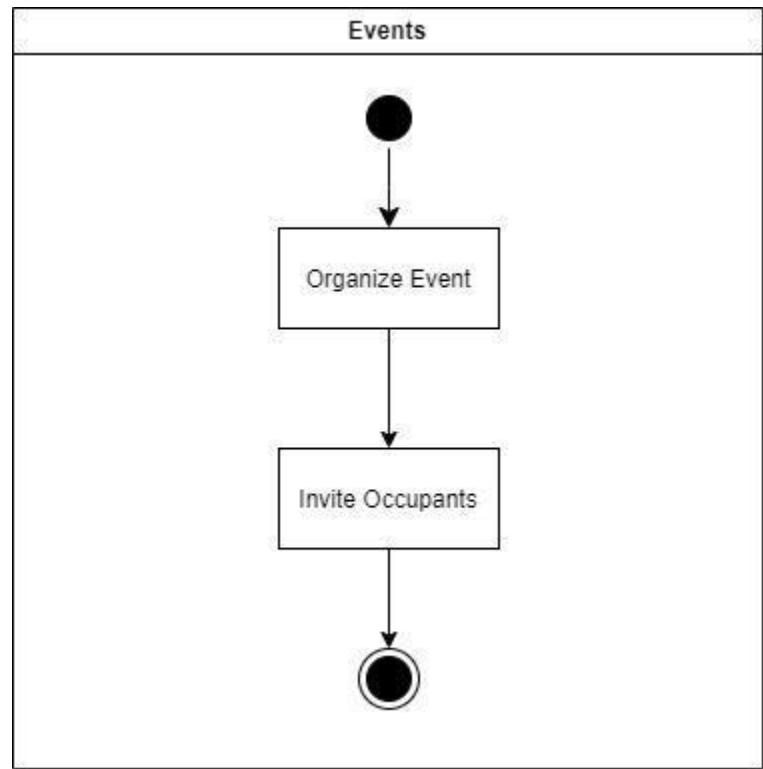
ID: 018 - 019 - 020



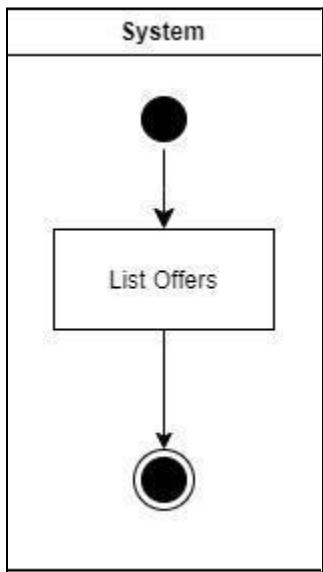
ID: 021



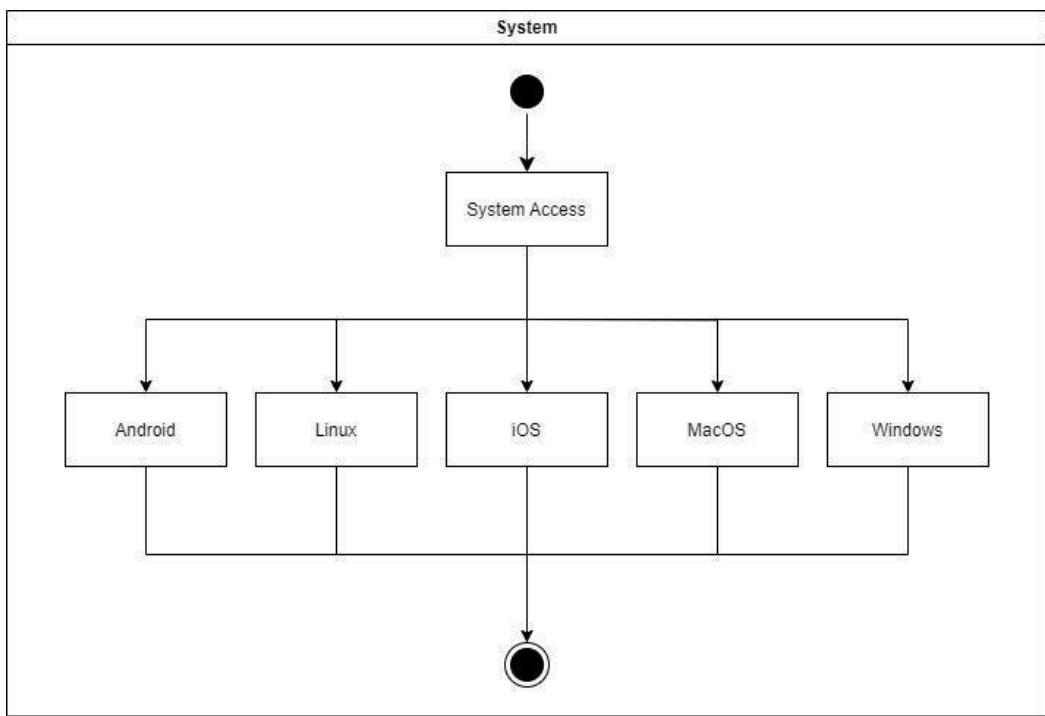
ID: 022



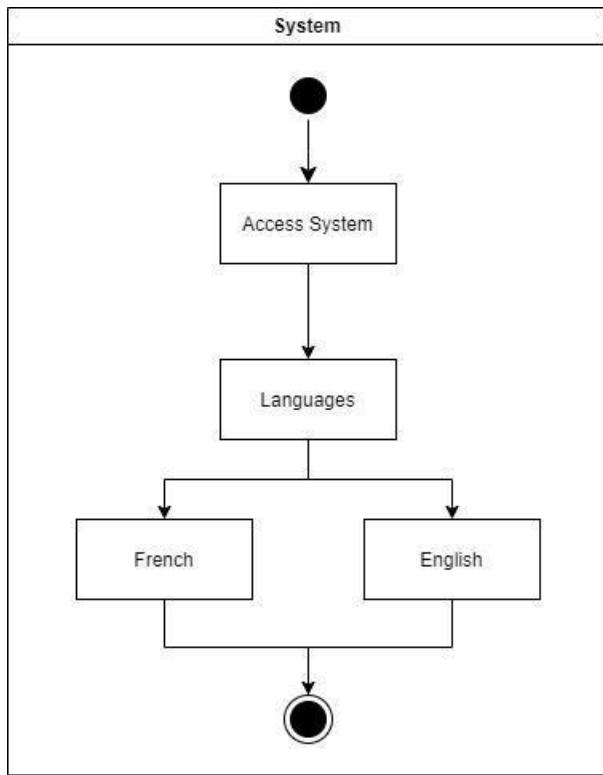
ID: 023



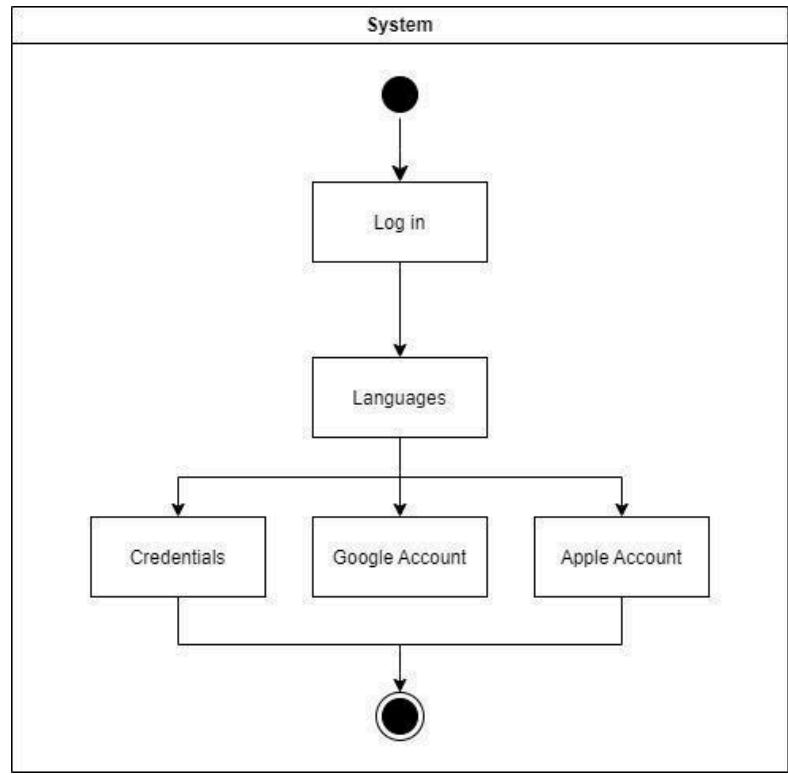
ID: 024



ID: 025

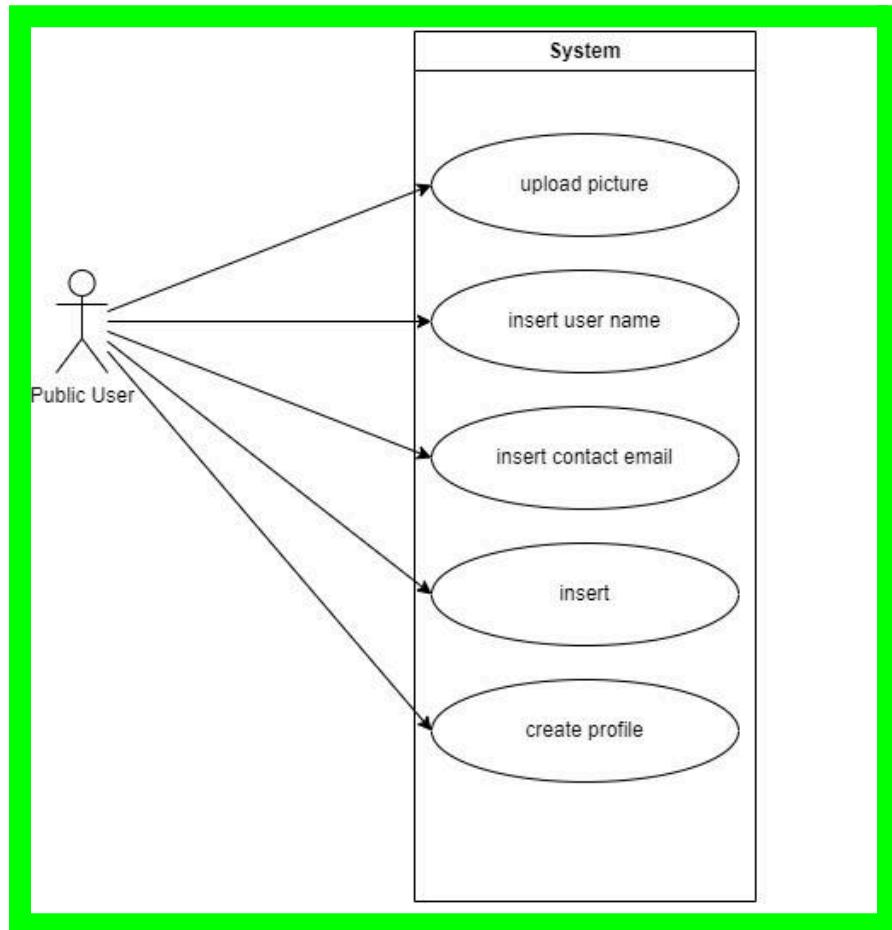


ID: 026

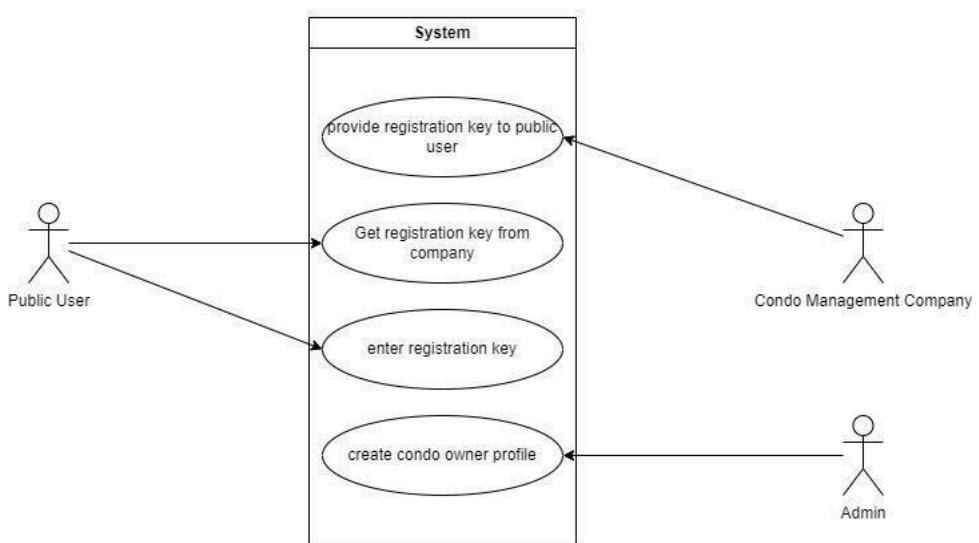


## Use Case Diagrams

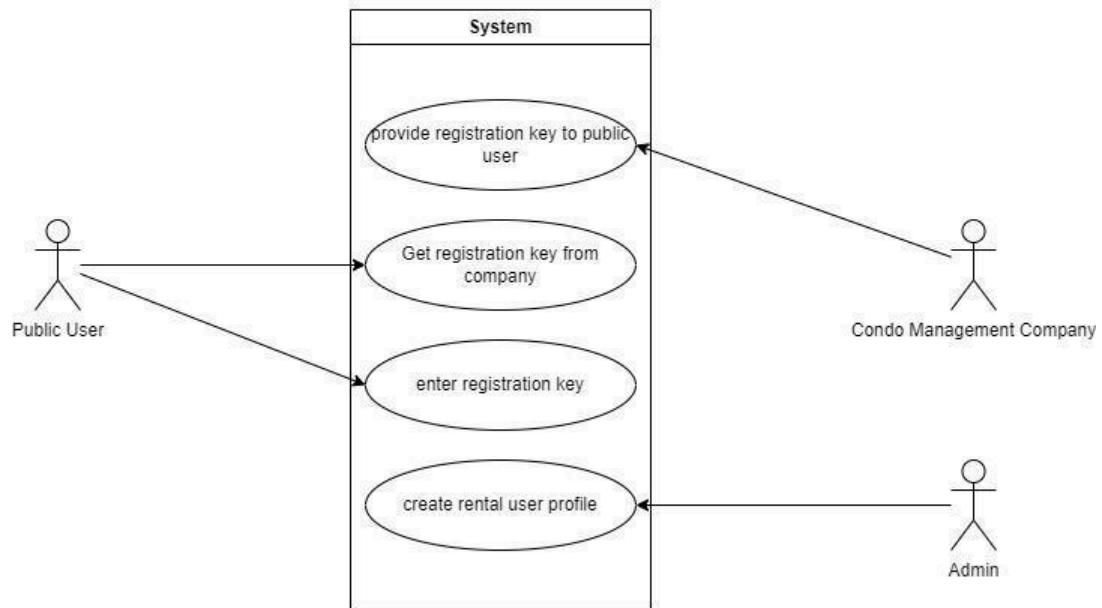
ID: 001



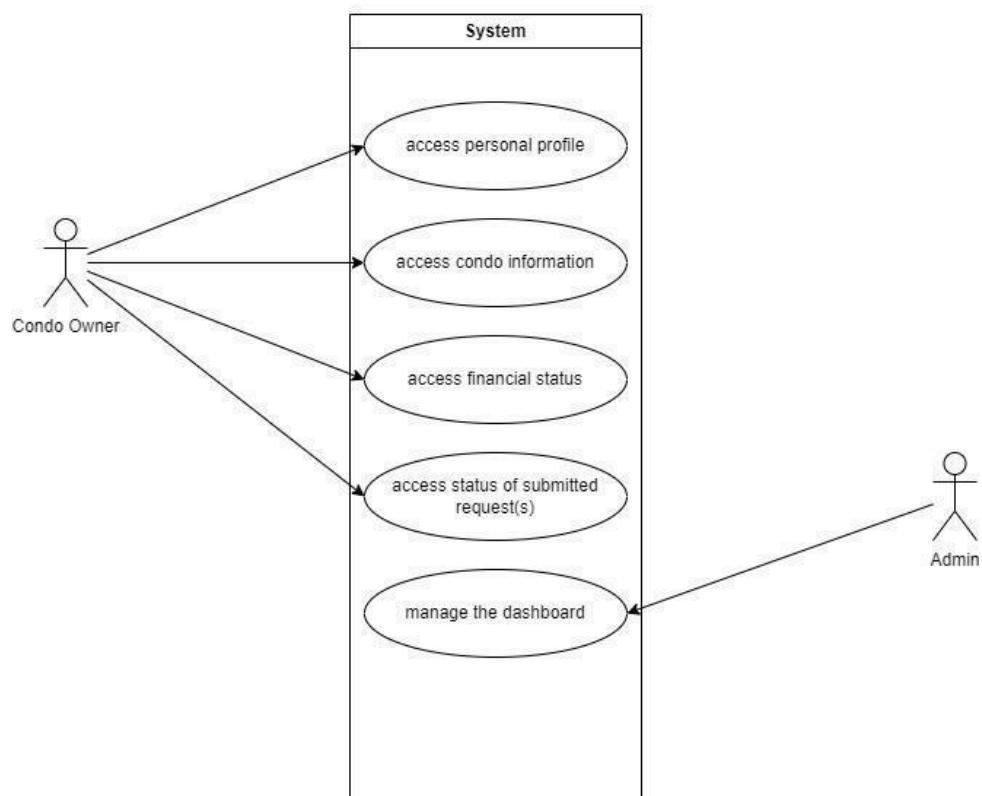
ID: 002



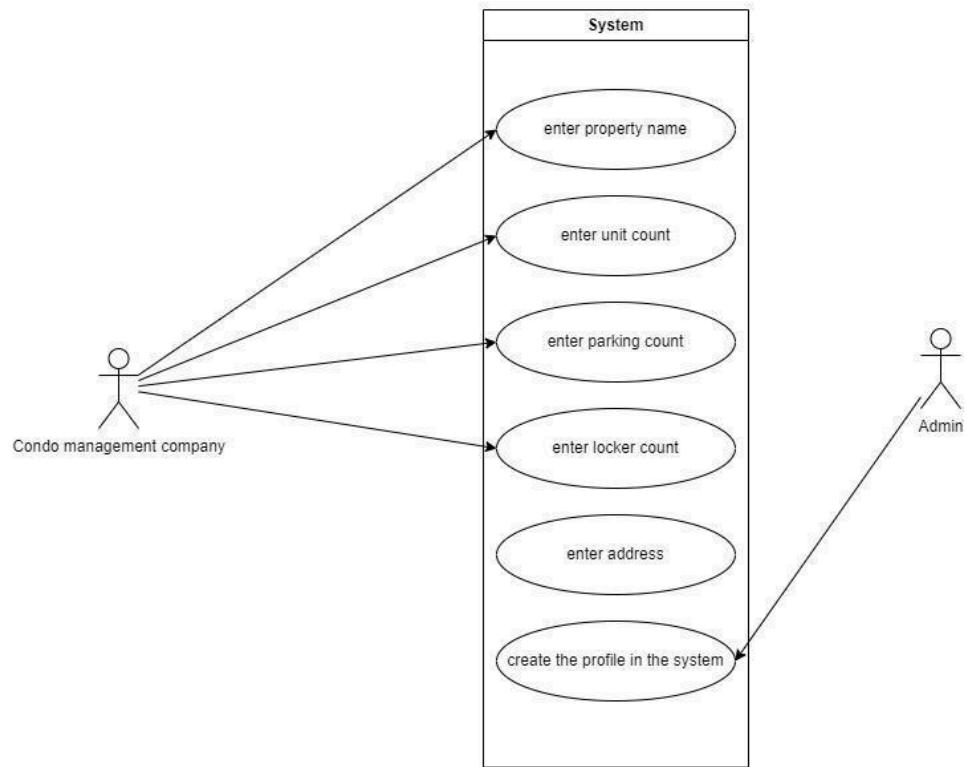
## ID: 003



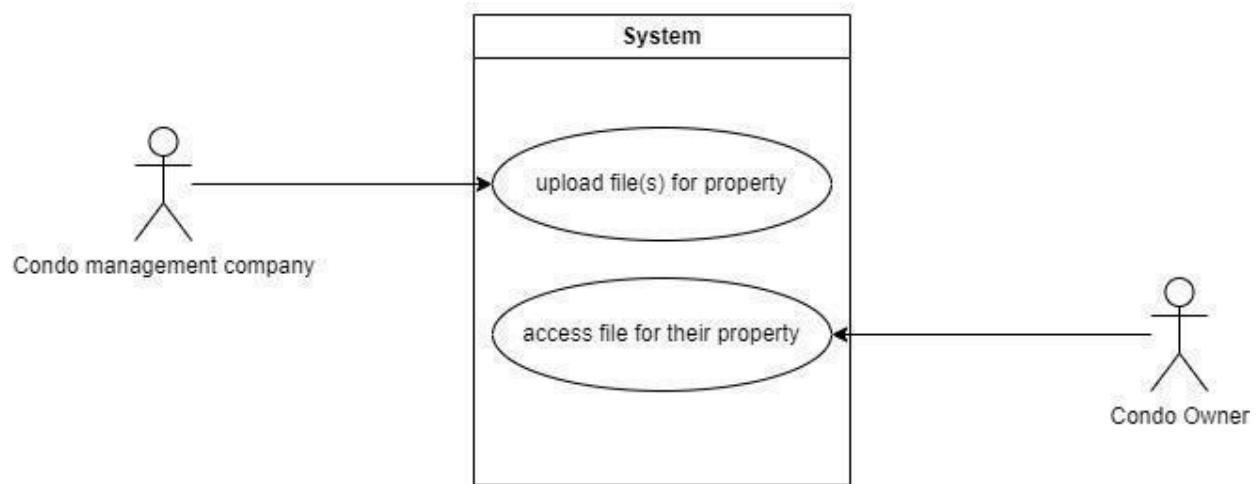
## ID: 004



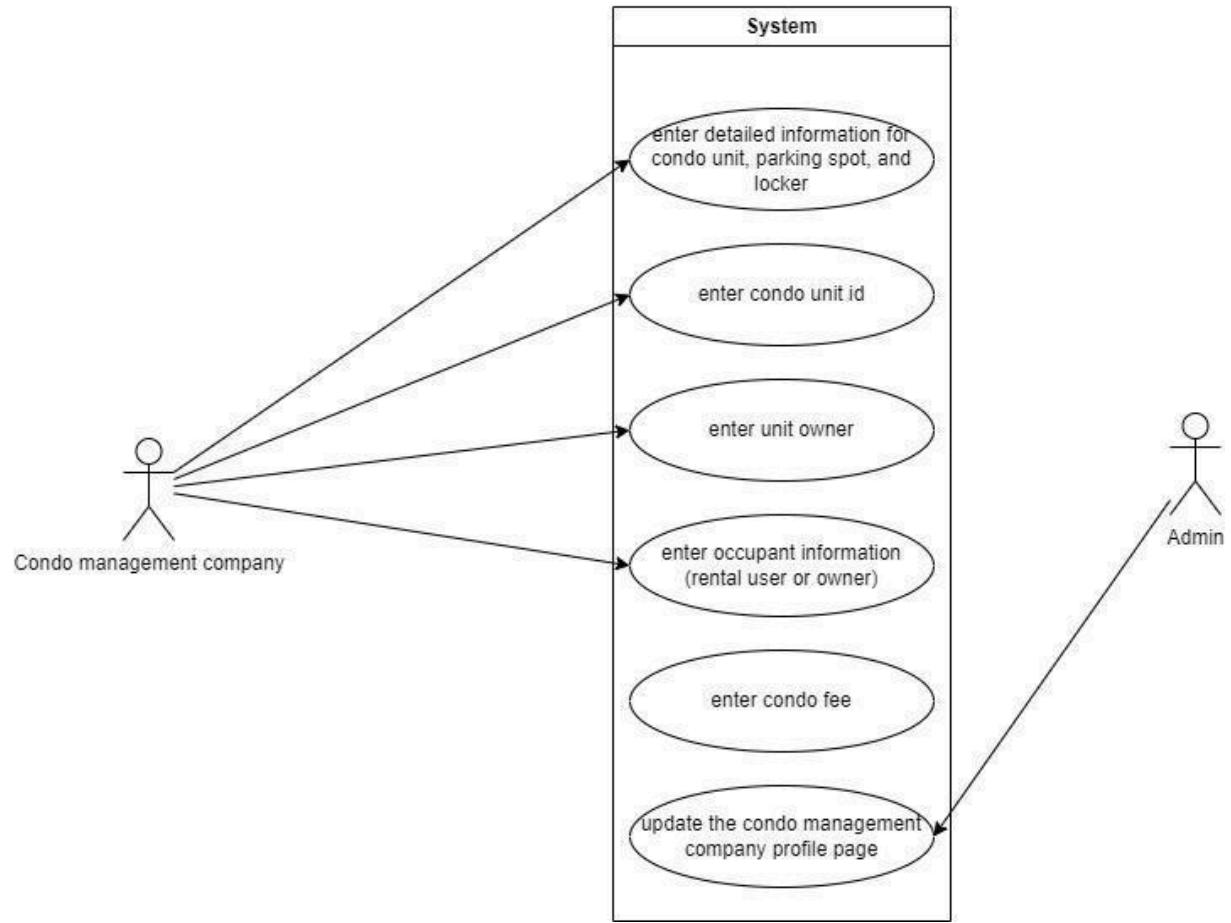
ID: 005



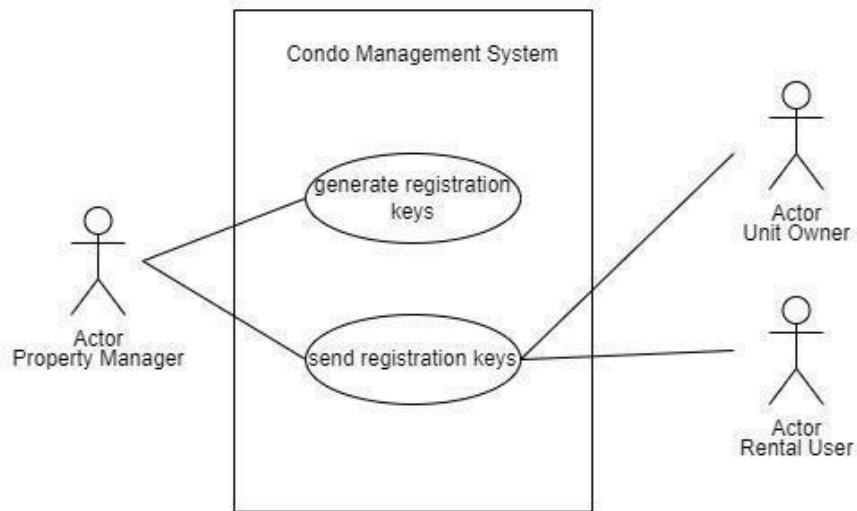
ID: 006



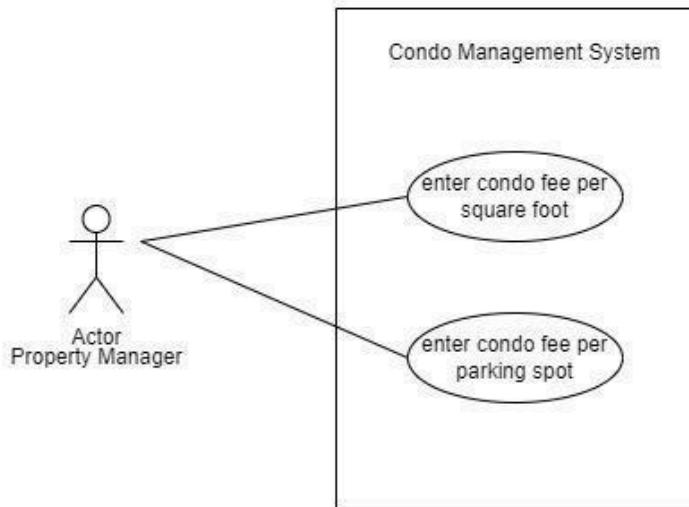
ID: 007



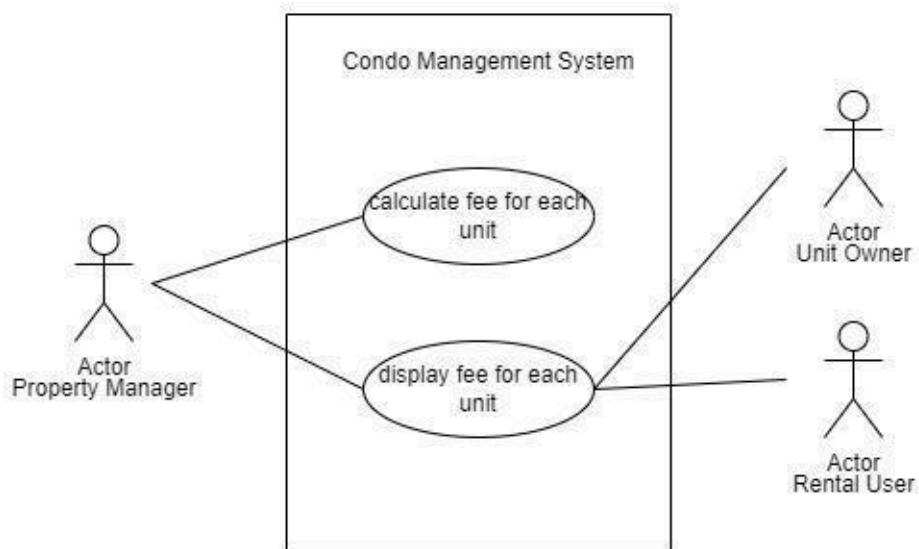
ID: 008



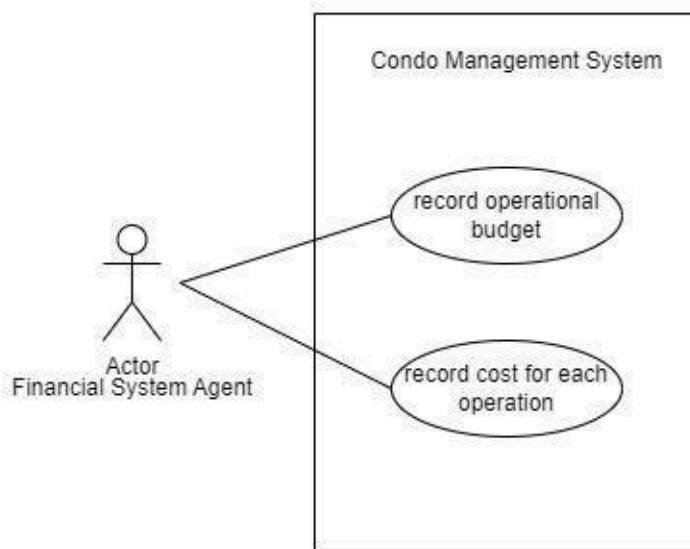
ID: 009



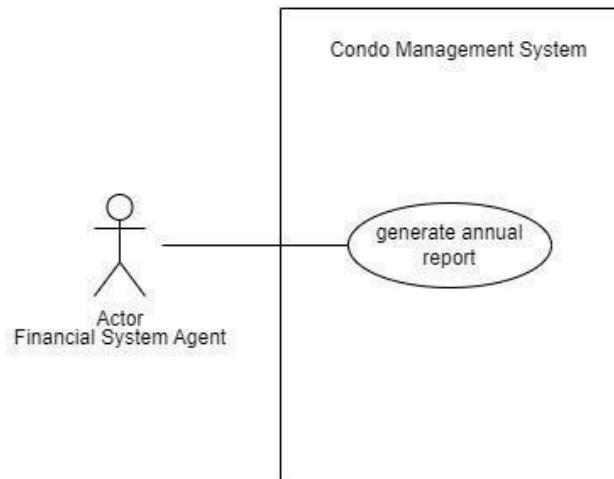
ID: 010



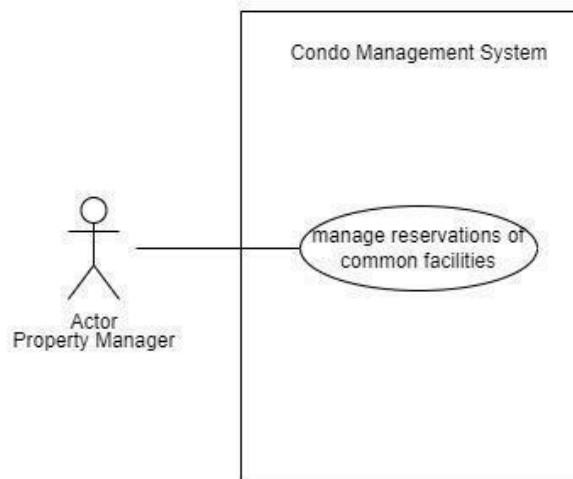
ID: 011



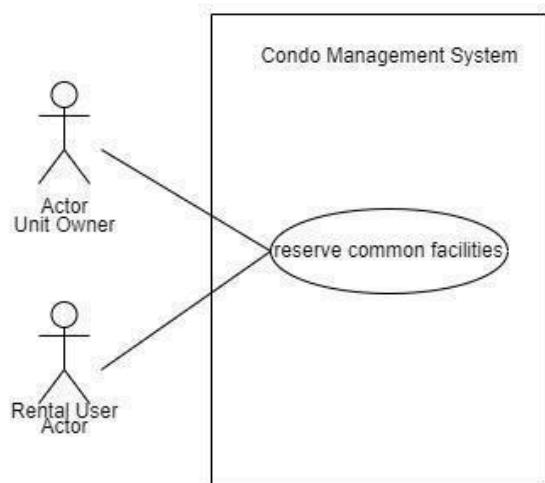
User Story #12



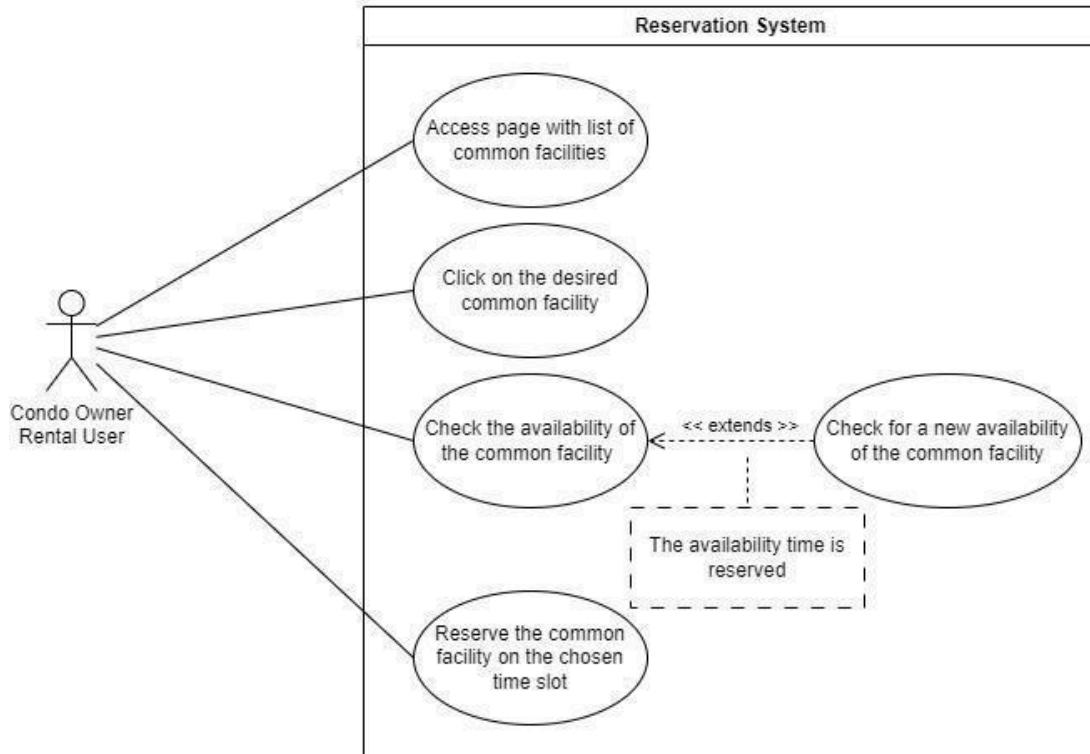
User Story #13



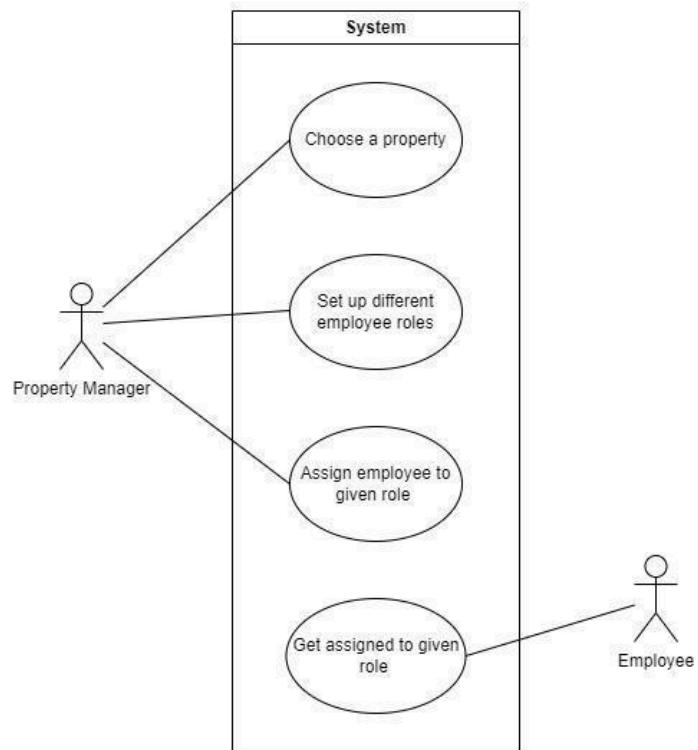
User Story #14



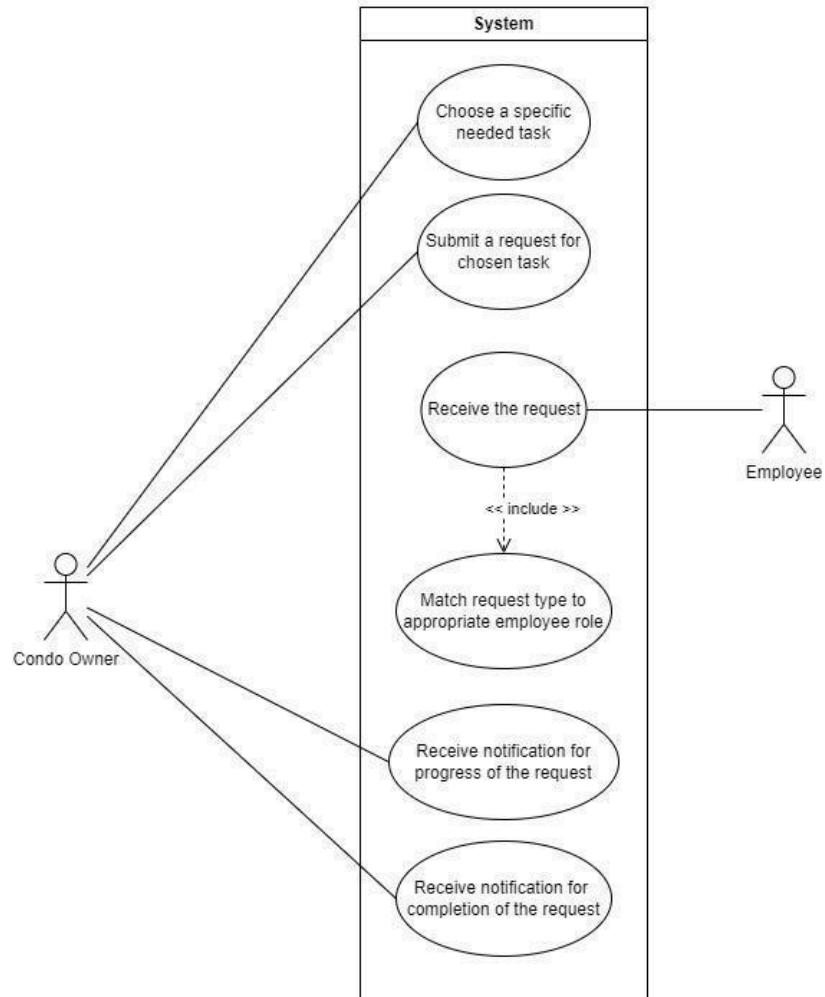
## User Story #15-16



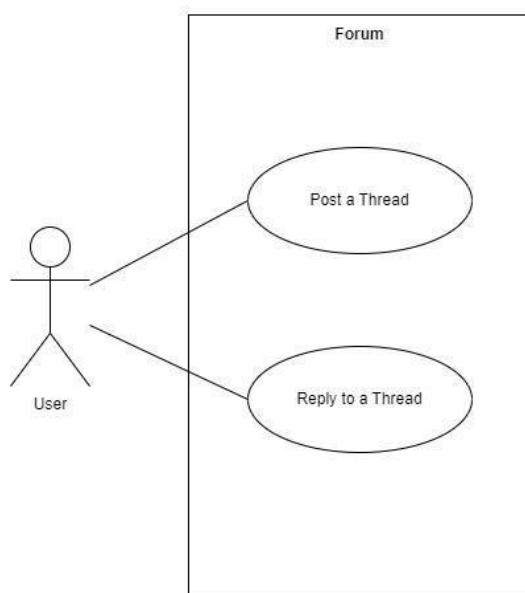
## User Story #17



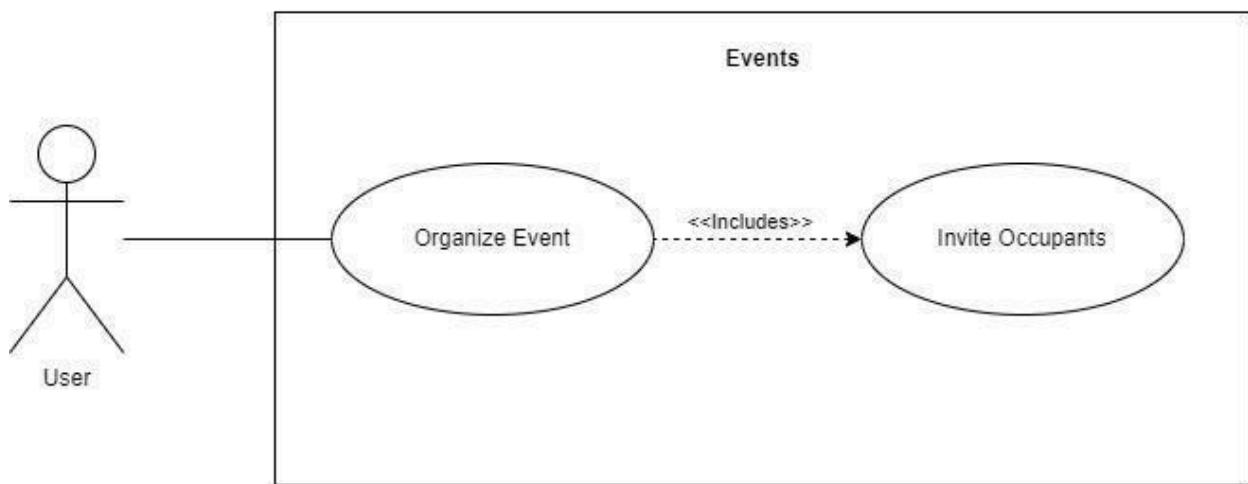
## User Story #18-19-20



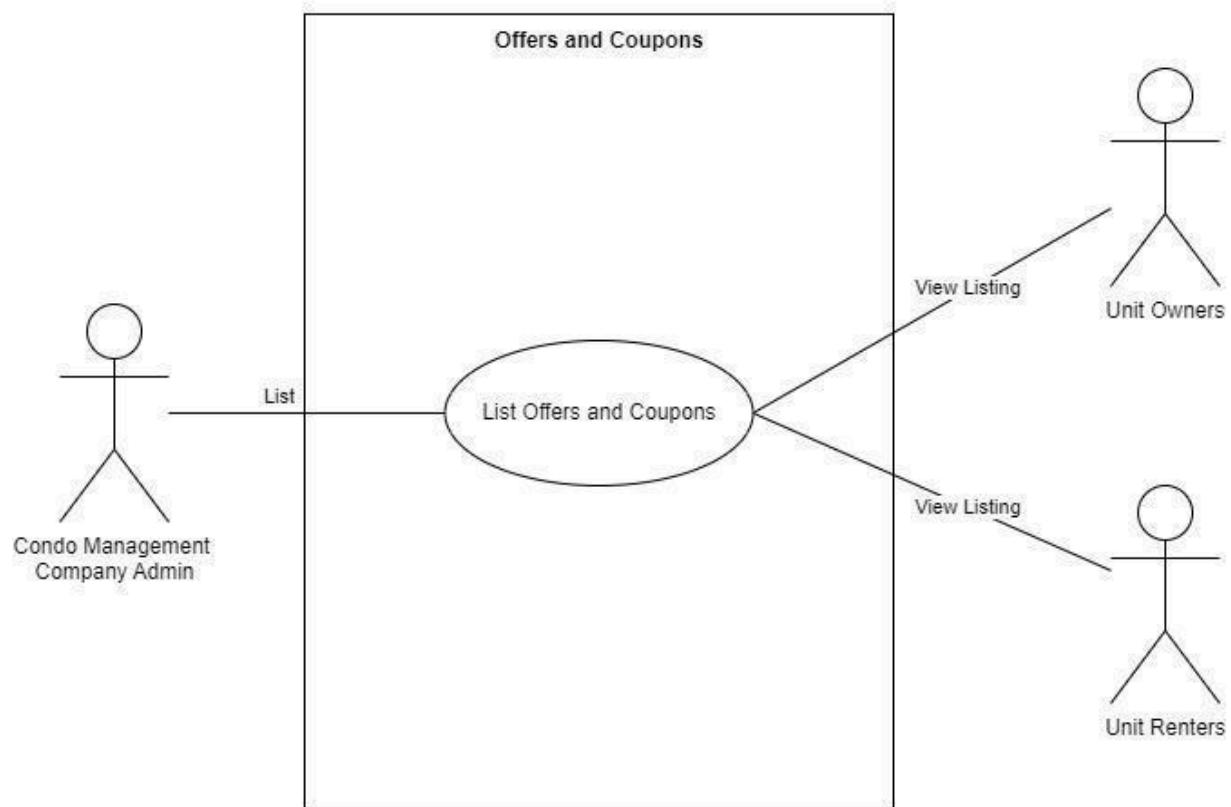
User Story # 21



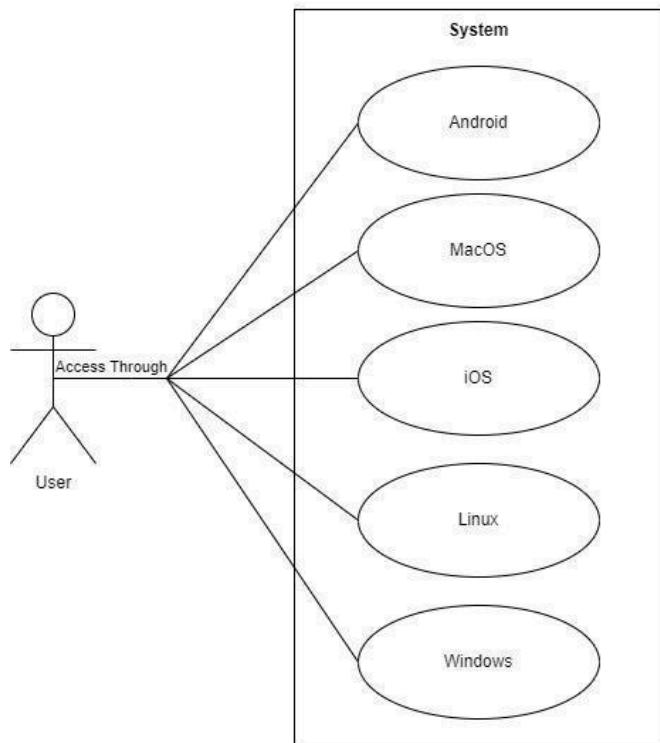
User Story # 22



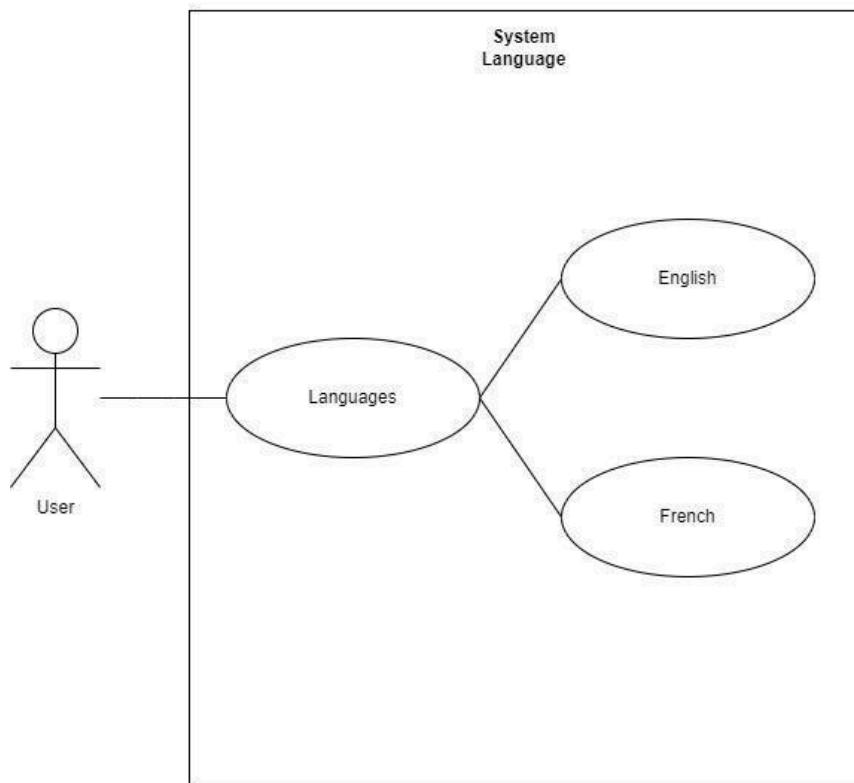
User Story # 23

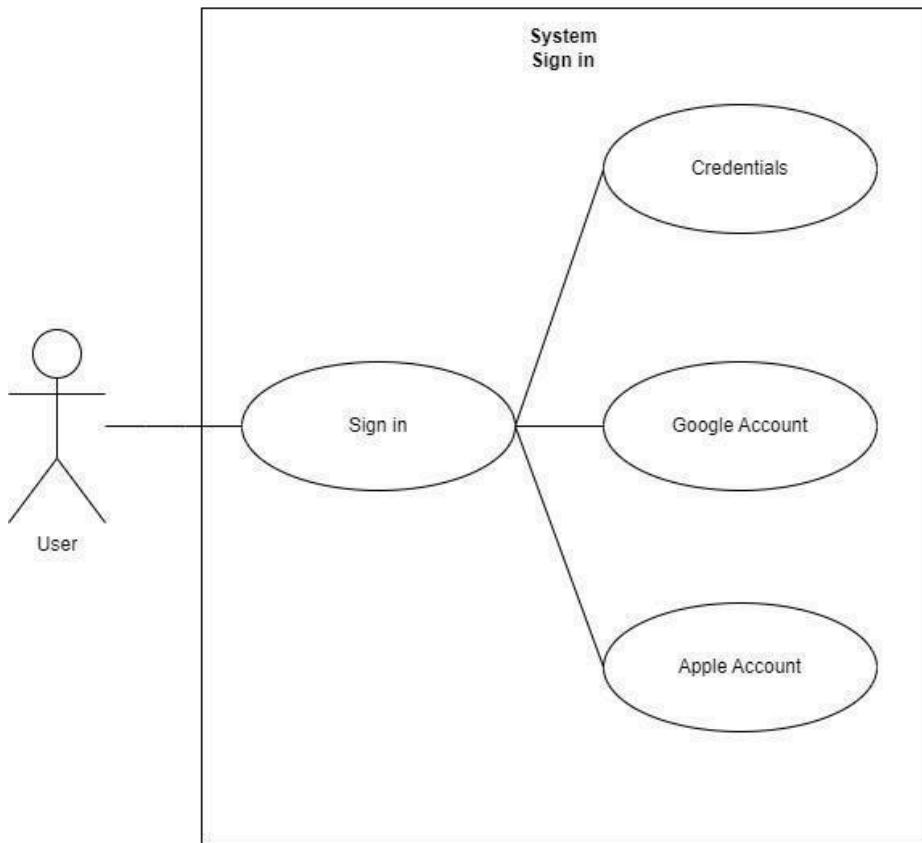


User Story # 24



User Story # 25





# 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.

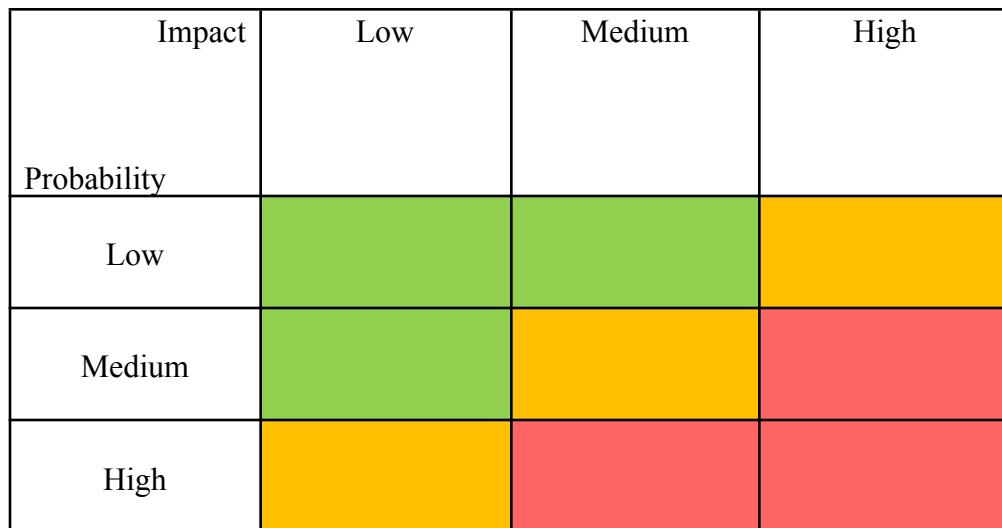


Figure [1]: Risk management chart

## 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.

Table [1]: List of identified risks

Link to the Risk Analysis UrbanKey:

[Risk analysis UrbanKey.xlsx](#)

# Short Sprint #2 Retrospective

Crafting Excellence

Key Takeaways from Our Project Postmortem

## **Introduction**

This postmortem reflects on the second sprint of our project, where our primary focus was on updating documentation and developing a functional webpage. Throughout this sprint, our objective was to work on both the front-end and back-end components of various pages within the project. Our initial expectation was to establish the basic structure of these elements, ensuring key functionalities were implemented without delving into intricate details by the sprint's end. However, the challenge arose as team members juggled commitments from other courses, which impacted our ability to fully realize this goal. Despite these constraints, we remained committed to delivering a functional webpage to the best of our abilities.

Despite facing time constraints due to other course commitments, our team dedicated their efforts to delivering a functioning webpage that we can take pride in. While we may not have achieved all the desired intricacies, we successfully implemented key elements across the front-end domain. Our collective dedication and hard work enabled us to overcome challenges and produce tangible results by the end of the sprint. Nonetheless, reflecting on this experience, we recognize the need to refine our time management strategies to optimize productivity in future sprints.

Moving forward, the lessons learned from this sprint will inform our approach to time management and project prioritization in the subsequent phases. We acknowledge the importance of effectively balancing commitments across various projects and courses to ensure consistent progress and quality deliverables. By leveraging these insights, we aim to enhance our team's efficiency and productivity in the upcoming sprint, ultimately driving us closer to achieving our project objectives.

## **What went wrong**

### *I - Managing the GitHub*

During this sprint, as more team members delved into coding tasks, maintaining a streamlined workflow posed a slight challenge. Ensuring that every member initiated a new branch in the GitHub repository before submitting a pull request became particularly crucial. The increased activity necessitated a heightened focus on adherence to this process to prevent conflicts and maintain code integrity.

Efforts were directed towards reinforcing the practice of creating dedicated branches for individual tasks among team members. Clear communication and reminders were employed to underscore the importance of this step in the development cycle. Emphasizing the significance of branch management facilitated smoother collaboration and minimized disruptions, fostering a more efficient and organized development environment.

## **2 - Understanding Website Page Interaction**

During this sprint, understanding website page interaction emerged as another notable challenge. Understanding the intricacies of how each page interconnected remained somewhat complex, impeding seamless navigation between different sections. Clarifying the pathways linking various pages and comprehending the navigation flow presented ongoing difficulties for the team.

Efforts were directed towards decoding the complexities of website page interaction. Collaborative discussions and exploratory exercises were employed to dissect the navigation structure and enhance comprehension. By delving into the intricacies of page linkage and navigation pathways, the team aimed to streamline user experience and mitigate any confusion surrounding website navigation during the sprint.

## **3 - Achieving Visual Consistency**

Maintaining visual consistency across the website posed a challenge as each team member tackled different front-end pages during the sprint. Adherence to a unified design language became important to ensure a cohesive user experience. To address this, the team implemented a strategy of closely adhering to the prototype established in the previous sprint. By precisely following the prototype's specifications, discrepancies in visual elements were minimized, fostering a more harmonious overall aesthetic.

Effective communication played a pivotal role in resolving any uncertainties that arose during the development process. Team members remained vigilant in seeking clarification whenever doubts surfaced, fostering a collaborative environment helpful to resolving issues promptly. Through this proactive approach and a dedicated commitment to the established design prototype, the team successfully mitigated challenges related to visual consistency across the website's front-end pages.

## **4 - Time Management**

Time management emerged as a significant issue during the sprint, with certain team members unable to complete front-end development tasks promptly. This delay resulted in a cascading effect, impacting the availability of sufficient time for back-end developers to integrate their components effectively. The lack of synchronization between front-end and back-end development phases hindered the seamless progression of the project, highlighting the importance of better time allocation and coordination among team members.

The delay in front-end development ultimately underscored the need for a more synchronized workflow and improved communication channels within the team. Going forward, strategies such as setting clearer deadlines, implementing regular progress check-ins, and fostering proactive collaboration will be essential to ensure a more streamlined development process and mitigate similar time management challenges in future sprints.

## **What went right**

### **1 - Documentation**

One major success during this sprint was the quick and easy completion of documentation tasks. The majority of the documentation was straightforward to finalize due to the groundwork laid in the previous sprint. With a significant portion of the documentation already in place, the focus primarily shifted to updating specific sections, which were clearly outlined and discussed with the Teaching Assistant beforehand. This proactive approach ensured that the documentation process proceeded smoothly, saving valuable time and effort.

Clear communication with the TA proved instrumental in efficiently updating the documentation. Discussing the areas requiring updates in advance helped prioritize tasks and allocate resources effectively. By leveraging the existing documentation framework and proactively addressing necessary revisions, the team successfully navigated the documentation phase of the sprint, setting a positive tone for subsequent tasks.

## ***2 - Front-End Development***

The front-end development phase proceeded smoothly as each team member was assigned a specific page of the UrbanKey website to design. By distributing tasks in this manner, the workload was evenly distributed, ensuring a balanced approach to development. Team members were able to focus their efforts on their designated pages, streamlining the design process and enhancing efficiency.

Following the previously established UI prototype facilitated cohesive and consistent development across all pages. With a clear guideline in place, team members could easily align their designs with the established aesthetic, minimizing deviations and promoting visual coherence throughout the website. This approach enabled seamless integration of individual page designs into the overall website framework, contributing to the success of the front-end development phase.

## ***Conclusion***

In conclusion, the sprint presented a series of challenges and successes, each contributing to the overall progress of the project. Challenges such as maintaining visual consistency and understanding website page interaction were effectively addressed through strategies such as adhering to prototypes and fostering clear communication. Meanwhile, successes such as streamlined documentation completion and efficient front-end development highlighted the team's ability to leverage pre-established frameworks and distribute tasks effectively. By navigating these hurdles and capitalizing on opportunities for collaboration and adherence to established processes, the team successfully advanced the UrbanKey project, setting a solid foundation for future sprints and project milestones.

# Sprint 2 Release Plan

In Sprint 3 of the Condo Management System, our focus will shift towards further refining and expanding the system's functionality, with an emphasis on user engagement, system accessibility, and operational efficiency. This sprint will tackle a set of 10 user stories, extending from User Story #17 to #26, each addressing critical areas of improvement identified in previous sprints.

The sprint's primary objectives are multifaceted, aiming to enhance the system's role management capabilities, streamline request handling processes, and create a more interactive and engaged user community. More specifically, we plan to introduce advanced role management tools that allow for the creation, assignment, and modification of roles within the system, which enables more organized and efficient task distribution among property management teams. A significant focus will also be placed on developing a more intuitive and user-friendly request submission and assignment system, ensuring that condo owners can easily submit various requests and that these requests are promptly assigned to the appropriate team members for resolution.

To enhance user engagement, we will implement a community forum and a notification system, which will provide users with platforms for interaction, information exchange, and timely updates on request statuses and other relevant activities. Additionally, the sprint will see the introduction of features supporting event organization and management, allowing users to create, publicize, and manage events within the condo community, which will foster a sense of community for users. Accessibility improvements are a focus, with plans to make the system available across multiple platforms, including mobile and desktop, and to offer multilingual support, focusing on English and French, to cater to a diverse user base. Furthermore, enhancing the system's security and user convenience through the integration of secure sign-in options via Gmail and Apple accounts is a priority.

These enhancements are designed to significantly improve the user experience, streamline property management tasks, and ensure the system's adaptability and scalability for future needs. By the end of Sprint 3, we anticipate having a more solid, user-friendly, and comprehensive condo management system that meets the requirements.

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

User Story ID	User Story Points (USP)	Priority	Status
User Stories are from # 17 to #26. The [redacted] 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/TODO</li><li>● DONE</li><li>● PUSHED TO SPRINT 4</li><li>● REMOVED</li></ul>

There are a total of 73 story points for Sprint 3, and approximately 68 hours of work.

**Link to the Release Plan Sprint 3 Excel file:**

[https://docs.google.com/spreadsheets/d/1fxH4Hrxxc8P\\_MPsNMSw-LHtfGgdFuyA/edit#gid=344660039](https://docs.google.com/spreadsheets/d/1fxH4Hrxxc8P_MPsNMSw-LHtfGgdFuyA/edit#gid=344660039)

# UI Prototypes

## Home Page

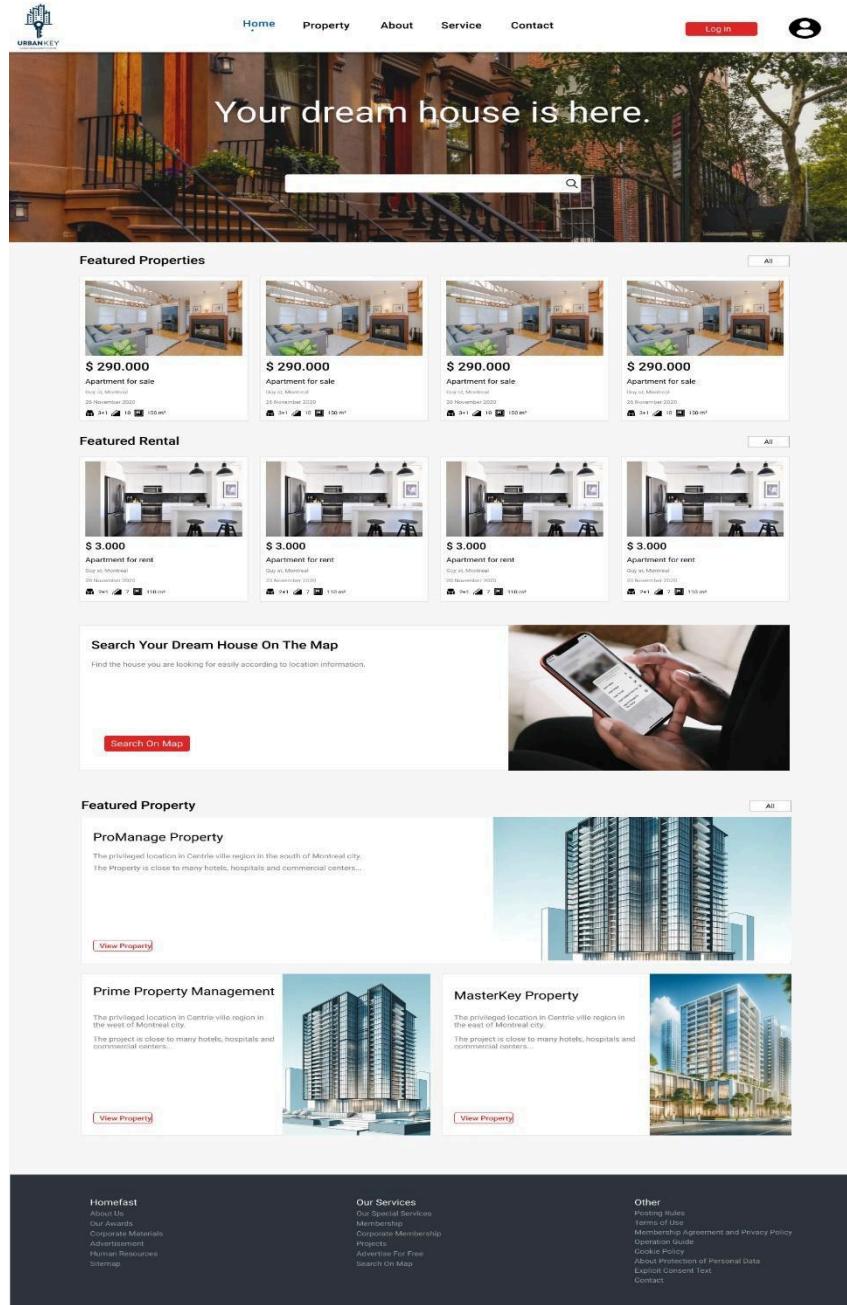
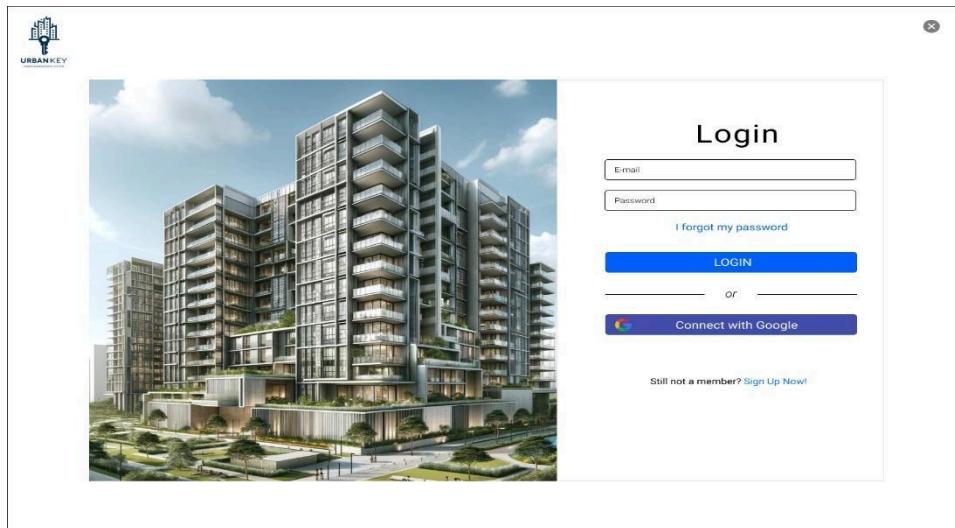


Figure 1

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.

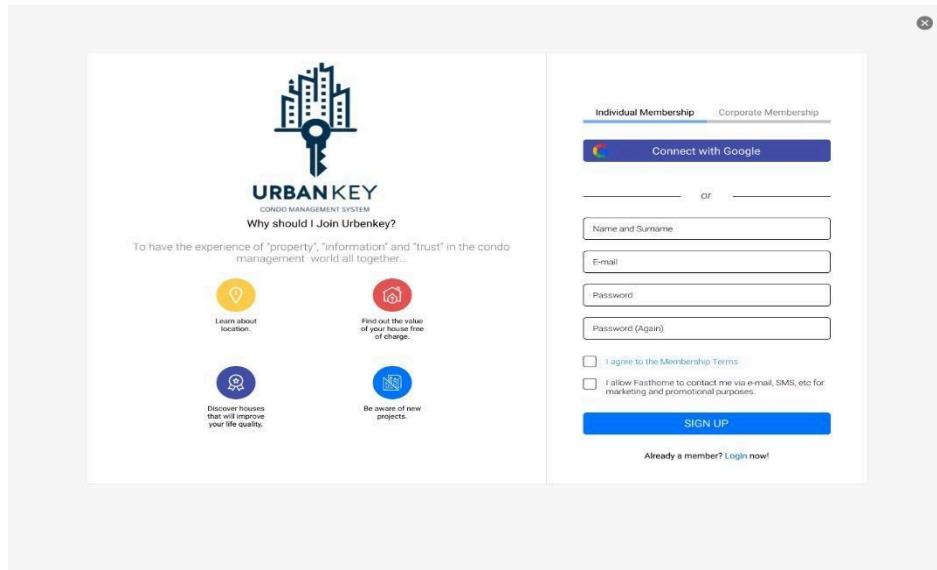
## ***Login***



**Figure 2**

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 3**

The form provides a choice between individual or corporate membership and the option to registration 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

**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		

**Financial Status**

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

**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
- ✓ Walk-in closet
- ✓ Dressing Room
- ✓ Video Intercom
- ✓ Shower
- ✓ Laminate
- ✓ Panel Door
- ✓ Blinds
- ✓ Sauna
- ✓ Satin Plaster
- ✓ Satin Color
- ✓ Ceramic Floor

**External Features**

- ✓ Elevator
- ✓ Gated
- ✓ Fitness
- ✓ Security
- ✓ Thermal Insulation
- ✓ Generator
- ✓ Tennis Court
- ✓ Car Park
- ✓ PVC
- ✓ Basketball Field
- ✓ Market

**Maintenance Requests**

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

**Location Information**

**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 4

## ***Notification***



**Figure 4.1**

## ***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 4.2**

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***

**Maintenance Requests**

Title \_\_\_\_\_

**Request Description**

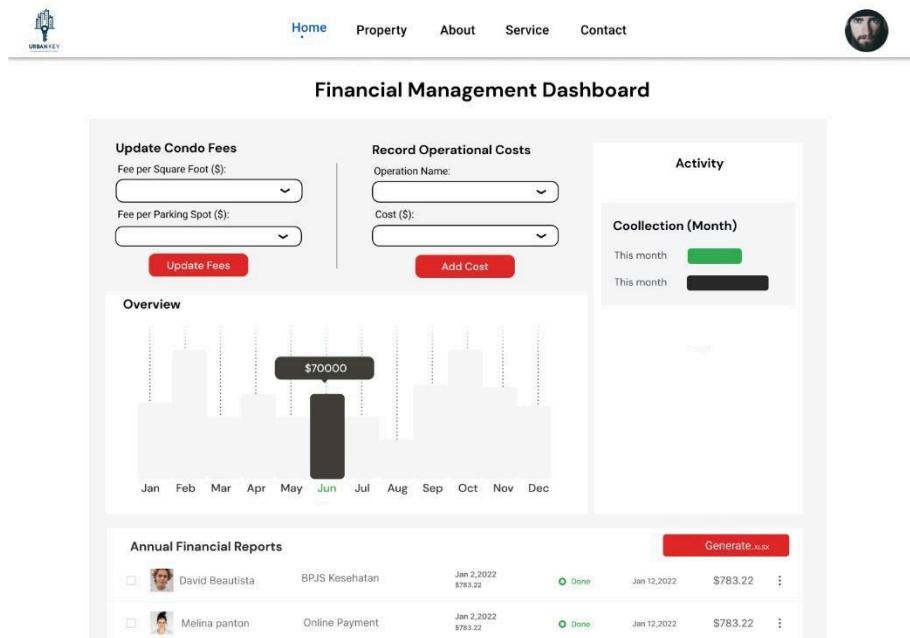
You can request about moving in/out (date for reserving elevators), intercom changes, requesting access (fobs, keys), reporting a violation, reporting deficiency found in common areas, or asking a question.

**Submit Your Request**

**Figure 4.3**

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

## ***Financial Management Dashboard***



**Figure 5**

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



[Home](#)   [Property](#)   [About](#)   [Service](#)   [Contact](#)



## Membership Information

Name / Surname

E-mail

Province  City

Mobile Number  Mobile Number 2

Confirm Your password  
\*\*\*\*\*

Address



Upload Profile Picture

[E-mail](#)   [SMS](#)

I want to be informed about all announcements and campaigns via commercial electronic mail.

No  Yes

[Save](#)

**Figure 6**

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. At the top, there's a navigation bar with links for Home, Property, About, Service, and Contact. The main content area is titled "Property Profile Management". It includes sections for "Category" (set to "Housing" and "Unit ID"), "Property Details" (Title, Description, Unit Owner, Unit occupant information, Price, Number of Rooms, Number of Living Rooms, Gross SF, Net SF, Building Type, Floor Location, Available for Loan, Furnished, Parking, Parking Spot ID, Locker, Rental Income, and Type), "Location Information" (Province, City, Neighborhood), and "Upload Condo Files" (with a "Choose File" button and a "No file Chosen" message). Below these is a "Posting Photos" section with a "Browse From Computer" button and a placeholder for adding photos. The final section is "Advertise Features" which lists numerous checkboxes for interior and exterior features like Air Conditioning, Laundry Room, and Swimming Pool. A red "Send Registration Keys" button is located at the bottom right of the form.

**Figure 7**

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 a facility booking system. At the top, there is a navigation bar with links for Home, Property, About, Service, and Contact. On the left, there is a logo for "URBANKY" featuring a stylized building icon. On the right, there is a user profile placeholder. Below the navigation bar, the page title "Facility Reservation System" is displayed. A section titled "Book a Facility" contains two dropdown menus: one for "Choose a Facility" (with options "Sky Lounge" and "Spa & Fitness", where "Sky Lounge" is selected) and another for "Select Date" (a calendar for September 2021 showing dates from 1 to 30, with the 29th and 30th highlighted in red circles). A red "Book Now" button is located at the bottom of this section.

**Figure 8**



**Congrats**

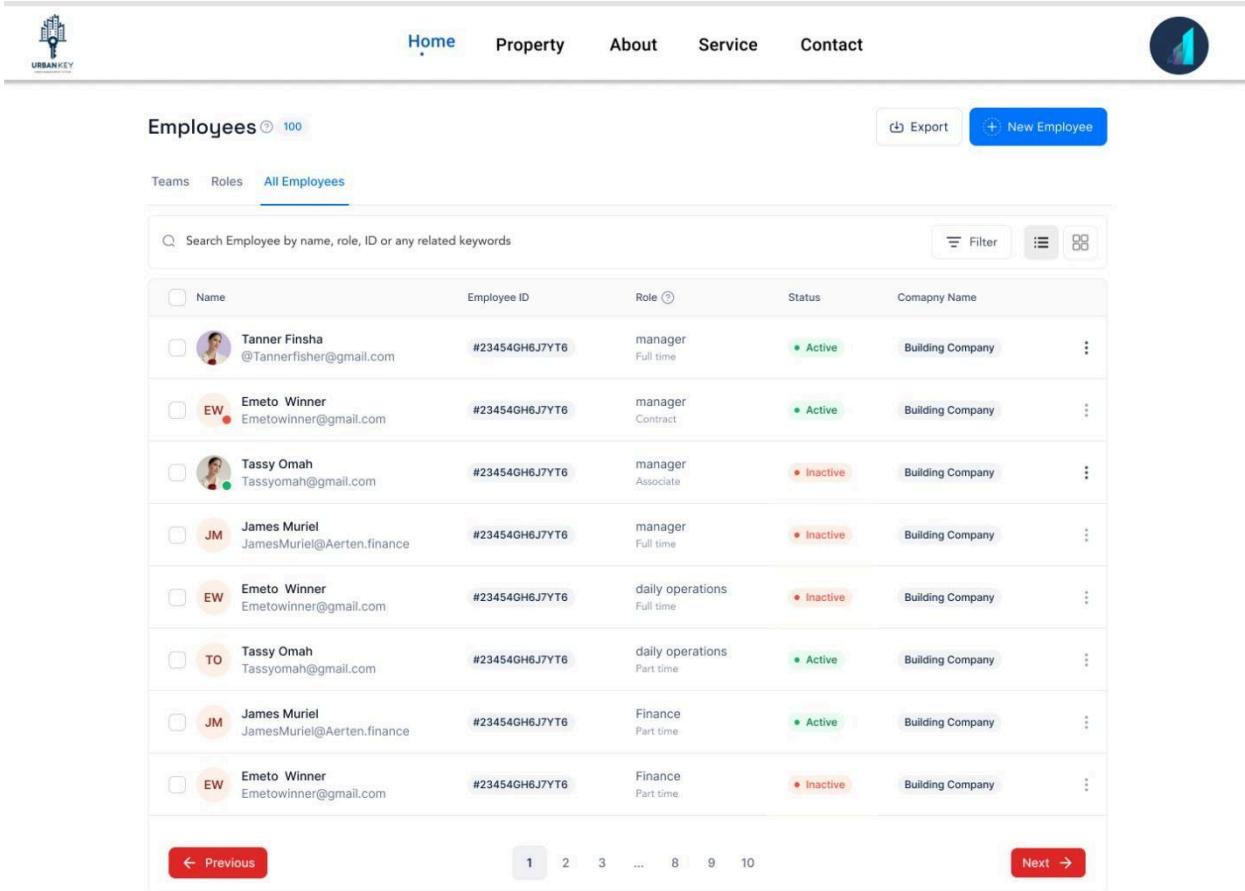
Reservation has been successfully  
made.

[Back to Home](#)

**Figure 8.1**

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



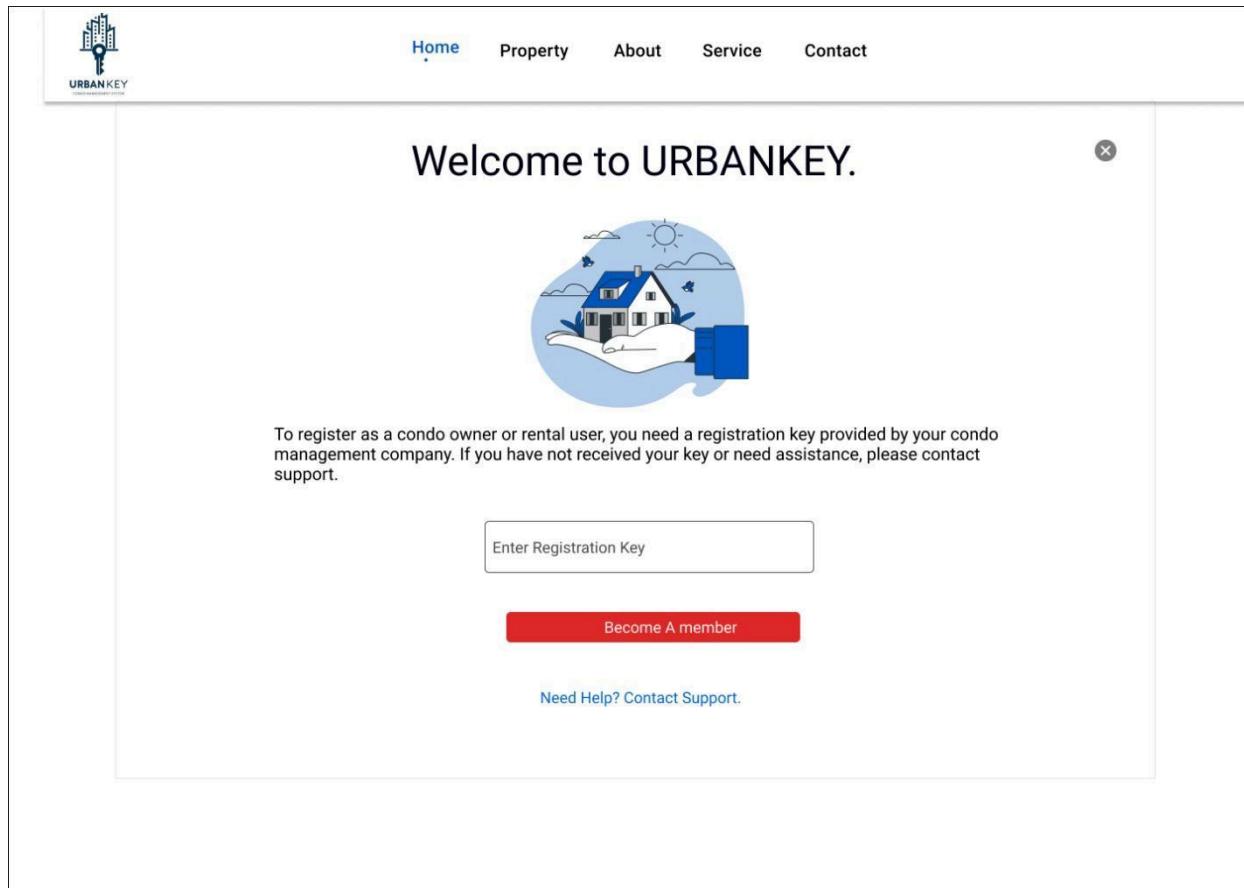
The screenshot shows a web-based employee management system. At the top, there is a navigation bar with links for Home, Property, About, Service, and Contact. On the far left is a logo for 'URBAN KEY' featuring a stylized building icon. On the far right is a circular profile picture placeholder. Below the navigation is a header bar with the title 'Employees' followed by a count of '100'. To the right of the count 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 Employee by name, role, ID or any related keywords'. To the right of the search bar are 'Filter' and 'View' buttons. The main content area is a table listing eight employee records. Each row includes a checkbox, a small profile picture, the employee's name, their ID, their role, their status (Active or Inactive), their company (Building Company), and a three-dot menu icon. The employees listed are Tanner Finsha, Emeto Winner, Tassy Omah, James Muriel, Emeto Winner, Tassy Omah, James Muriel, and Emeto Winner. The table has a header row with columns for Name, Employee ID, Role, Status, and Company Name. At the bottom of the table are navigation buttons for 'Previous' and 'Next'.

Name	Employee ID	Role	Status	Company Name
Tanner Finsha @Tannerfisher@gmail.com	#23454GH6J7YT6	manager Full time	Active	Building Company
EM Emeto Winner Emetowinner@gmail.com	#23454GH6J7YT6	manager Contract	Active	Building Company
TO 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 9

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 word "URBANKEY". At the top right is a navigation bar with links for Home, Property, About, Service, and Contact. Below the navigation bar, a large "Welcome to URBANKEY." message is displayed, accompanied by a small "X" icon in the top right corner. In the center is a circular illustration of a house on a hill under a sun and clouds. Below the illustration 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 this text is a white input field with a placeholder "Enter Registration Key". To the right of 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 10**

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

# Sprint 2 Testing Plan

This project documentation refers to a thorough testing strategy that was used to verify and validate the software developed during Sprint 2 of our Condo Management System project, titled UrbanKey. In order to meet the specific requirements of this iteration, it was crucial to incorporate a multi-level testing strategy that included a combination of various testing processes.

## **Testing Tool Selection:**

At this phase of the project, the sole, primary tool that was used for testing was Pytest.

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.

The 2 testing files currently present in our application are “app.py” and “test\_app.py”:

- The “app.py” file provides a structured backend setup with endpoints for basic user functionalities, employing Flask's simplicity and flexibility for web development, and MongoDB's schema-less nature for easy data management. The inclusion of error handling and debugging messages, such as printing exceptions and confirming database connection success, indicates the need for monitoring during development and testing phases.
- The `test\_app.py` file is structured to conduct automated tests using Pytest. It employs the Flask testing client to simulate requests to the application's endpoints, allowing for the verification of responses without running a live server. The tests are designed to cover various functionalities of the application, such as the base route accessibility, user signup process, profile information retrieval, and error handling in user registration. To isolate the tests from actual database interactions, the “unittest.mock.patch” decorator is utilized to mock the “find\_one” and “insert\_one” methods of the “users” collection. This approach enables testing of the application's logic under simulated conditions, such as database query failures or insertions, without requiring a connection to a real MongoDB database. For example, in testing the signup functionality, the tests simulate both a scenario where a user already exists (triggering an exception) and a successful user addition, checking the application's response in each case. Additionally, the tests verify the web app's response to invalid requests and simulate profile retrieval with predefined return values to ensure the correct user data is fetched and returned. By mocking database operations and simulating request scenarios, these tests effectively assess the robustness and correctness of the website's endpoint behaviors under controlled conditions, highlighting the UrbanKey's ability to handle both expected and unexpected interactions. This testing strategy provides valuable feedback on the application's functionality, helping in the identification and rectification of potential issues in a development environment.

## **Levels of Testing:**

***Unit Testing:*** In our files, the present tests are more focused on the behavior of routes and their interactions with mocked versions of database operations rather than testing individual functions within the Flask application in isolation. In future sprints, we will focus on implementing unit tests.

***Integration Testing:*** Inside the “test\_app.py” file, we have integration tests that use mocks to simulate database operations (**find\_one** and **insert\_one**). They are testing the Flask routes, considered as integration points, under different conditions by simulating interactions with the database layer. Since for now the database operations are mocked and not actually performed, these tests are more about testing the application logic under assumed conditions of how the database would respond, rather than actual integration tests with real database interactions.

***System Testing:*** Currently, there are no system tests implemented into our web app. System tests 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. The latter will be executed in future sprints.

***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 3, we plan on thoroughly researching and selecting the most suitable automation tool for our web application's needs.

## **Testing Approach:**

The testing approach employed via “app.py” and “test\_app.py” files centers on evaluating the functionality of our website application through integration testing, primarily by simulating interactions with its endpoints and using mocking techniques to isolate the application from its database dependencies. In “test\_app.py”, the Flask test client is utilized to send HTTP requests to various routes defined in “app.py”, such as those for user sign-up, login, profile retrieval, and updates, allowing for the testing of the application's responses under controlled conditions. Mock objects replace the actual database calls within these routes, simulating both successful operations and failure scenarios, such as database errors or conflicts (e.g., duplicate email sign-ups). This approach enables the verification of the application's logic and its ability to handle different data inputs and scenarios, ensuring that the application behaves as expected without the need for a live database connection. However, it does not extensively cover unit testing of individual functions or system testing of the application as a whole in a production-like environment, focusing instead on how well the application components integrate and respond within the scope of the provided functionalities. Future testing implementations will be added in subsequent sprints.

## **Metrics and Coverage:**

For Sprint 2, our team targeted a code coverage of least 80%, with a focus on critical paths in the application. We used coverage tools which are integrated with our testing frameworks to measure and report coverage.

## **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, 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 2 focused on developing 8 distinct pages 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 2, 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.

## 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 2, 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.

### *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. Each design pattern was carefully selected and implemented where appropriate, aligning with best practices in software architecture.

### *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.

### *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.

### *Source Code Documentation*

Comprehensive documentation accompanied our source code to provide insights into its functionality, usage, and implementation details. 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.

## *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.

## *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.

## *Use of Feature Branches*

Feature branches were employed to isolate development efforts and facilitate parallel work on different aspects of the project. Each team member worked on a dedicated feature branch when implementing new features or addressing specific issues. This approach enabled independent development, minimized conflicts, and allowed for more focused testing and review of code changes before integration into the main branch.

## *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.

## *Bug Reporting*

During Sprint 2, while we encountered issues and bugs, we didn't systematically report and track them using GitHub's issue tracking system. However, moving forward, we recognize the importance of this practice and plan to promptly report and document any encountered issues. By utilizing GitHub's issue tracking system, we aim to ensure timely resolution of bugs, improve software reliability, and enhance user satisfaction.

## *Use of Issue Labels*

In Sprint 2, we didn't utilize issue labels for tracking and filtering tasks. However, we acknowledge the value of this practice in streamlining task management and plan to implement it in future sprints. By categorizing tasks 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.

### ***Links Between Commits and Issues***

Although we didn't establish links between commits and corresponding issues during Sprint 2, we intend to adopt this practice in future sprints. 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 2 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.