SOEN 390 - Sprint 1 Documentation

Wednesday 28th, 2024

Department of Computer Science & Software Engineering

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 [product category]
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	• Develop architectural plans and designs for the condominium
Tech companies	Professionals involved in the creation and maintenance of the software of the system	• Software development • Database management

and programmers		• Updates and maintenance
Financial institutions	Institutions or banks providing loans or financial services	• Assess risks related to lending and investments • Credit evaluation
Legal advisors	Professionals ensuring any operations regarding the condos complies with the law, and handle any legal issues	• Privacy and Data Protection

3.2. User Summary

Name	Description	Responsibilities	Stakeholder
Condo owners	Individuals who own condominium units	• manages unit details • submit requests (moving in/out, violation reports, etc.) • access/update their property dashboards • reserve common facilities • engaging in financial transactions	Condo management companies
Tenants	Individuals occupying condominium units through rental agreements	• occupies and maintains the rented units • may engage in some financial transactions related to rental payments	Condo management companies
Condo management companies	Companies responsible for managing condominium properties	• Create and manage property profiles • Upload condo files • Manage financial aspects • Set up reservation system for common facilities • Assign roles to employees • Handle requests from condo owners/tenants	N/A

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

Table 1: Key Stakeholder's and User's Needs

Description	Centralized messaging system
Type	System enhancement
Responsibilities	Develop system for sending/receiving messages and notification
Success Criteria	Easily accessible messages for stakeholders
Involvement	Product manager, development team
Deliverables Comments/Issues	

Table 2: Better Communication

Description	User-friendly financial module
Type	System Enhancement
Responsibilities	Design financial module with clear fee breakdowns
Success Criteria	Understandable financial information
Involvement	Product manager, financial analyst and development team
Deliverables Comments/Issues	

Table 3: Clearer Financial Information

Description	Intuitive booking system
Type	New feature
Responsibilities	Develop user-friendly booking system with personnalized recommendations
Success Criteria	Simplified, error-free booking process
Involvement	Product manager, UX/UI designer

Description	Real-time request tracking
Type	System enhancement
Responsibilities	Implement system for real-time tracking and updates on requests
Success Criteria	Easy request tracking and timely updates
Involvement	Product manager and development team

Deliverables Comments/Issues	

Table 4: Easy Booking

Deliverables Comments/Issues	

Table 5: Quicker Resolution

Description	Centralizes messaging platform
Type	New feature
Responsibilities	Develop platform for broadcasting announcements to all users
Success Criteria	Effective communication of announcements
Involvement	Product Manger, Communication specialist and development team
Deliverables Comments/Issues	

Table 6: Broadcast Messages

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

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 protools 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 palace 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 acros 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 associates 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, so that residents can 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, so that I can 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

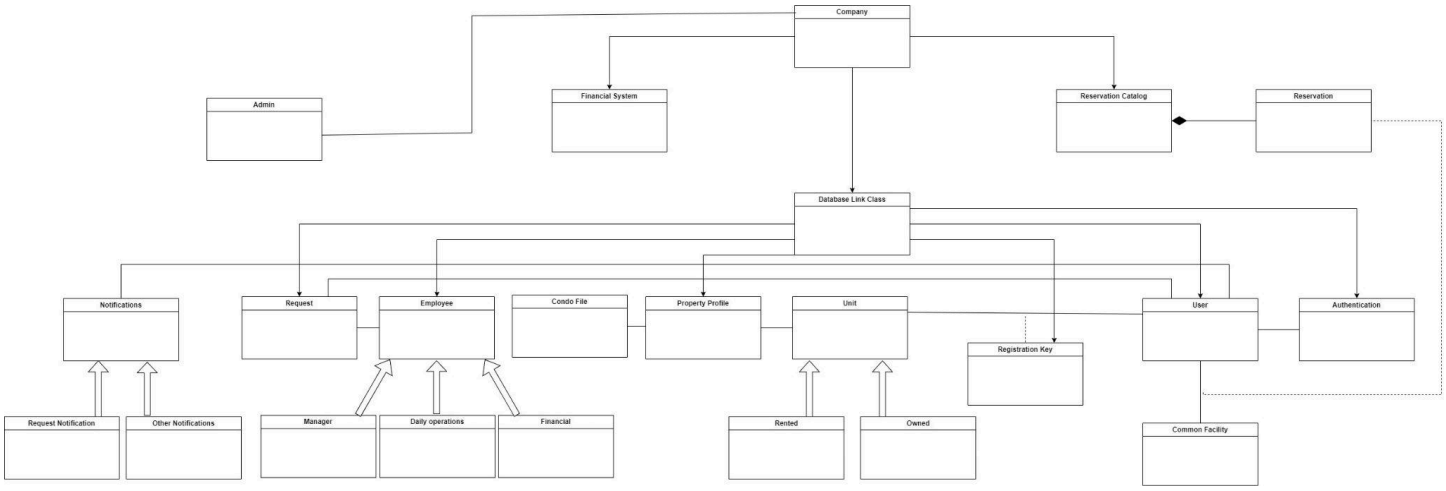
As a public user, I want the app to be available in both French and English.

User Story #26

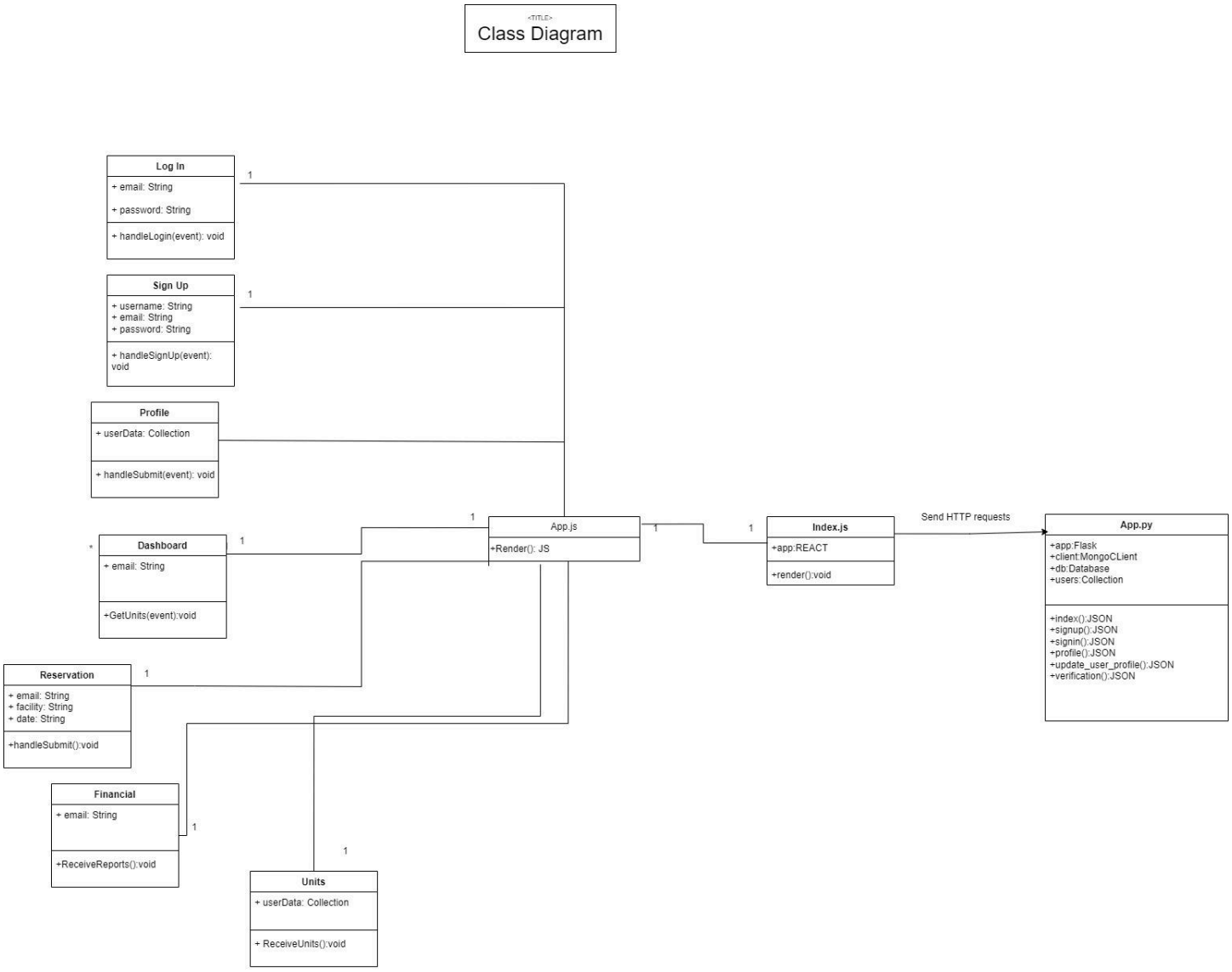
As a public user, I want to be able to sign in using my Gmail account or Apple account.

SAD

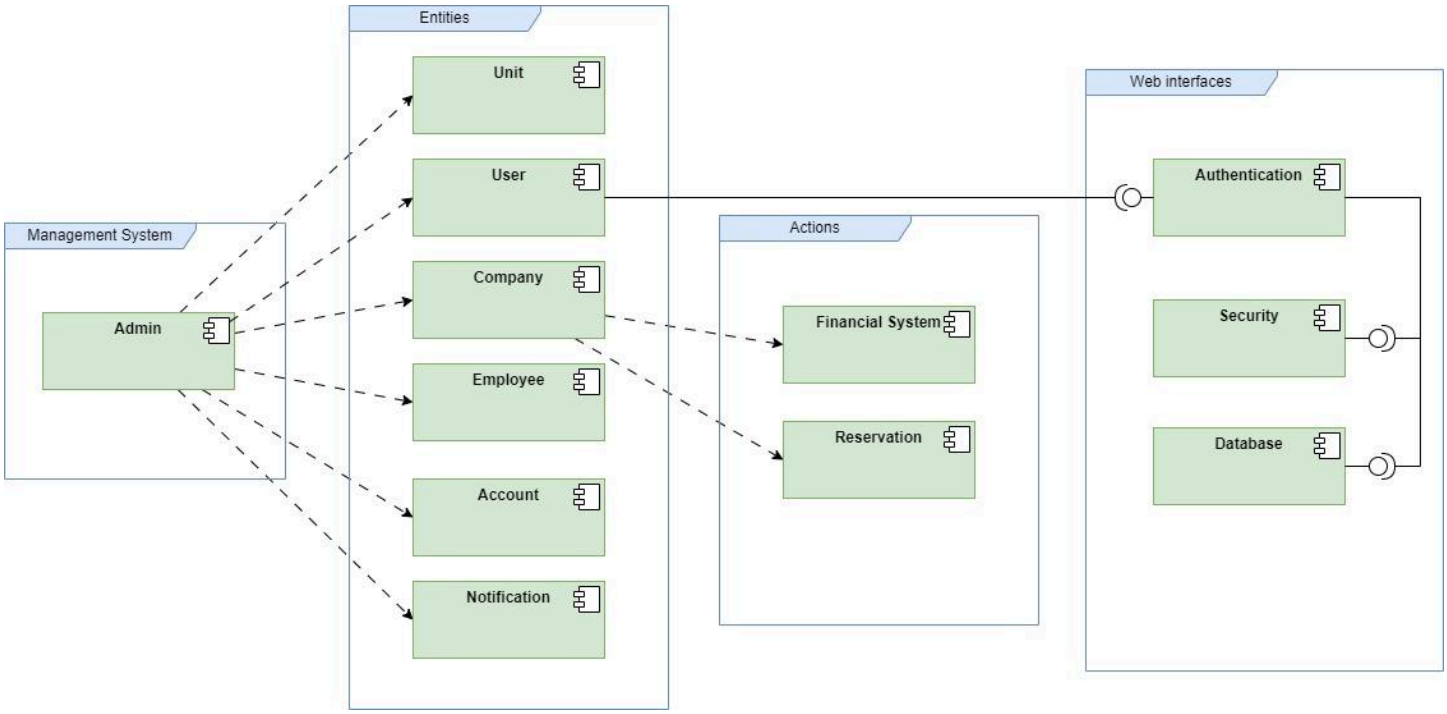
DOMAIN MODEL



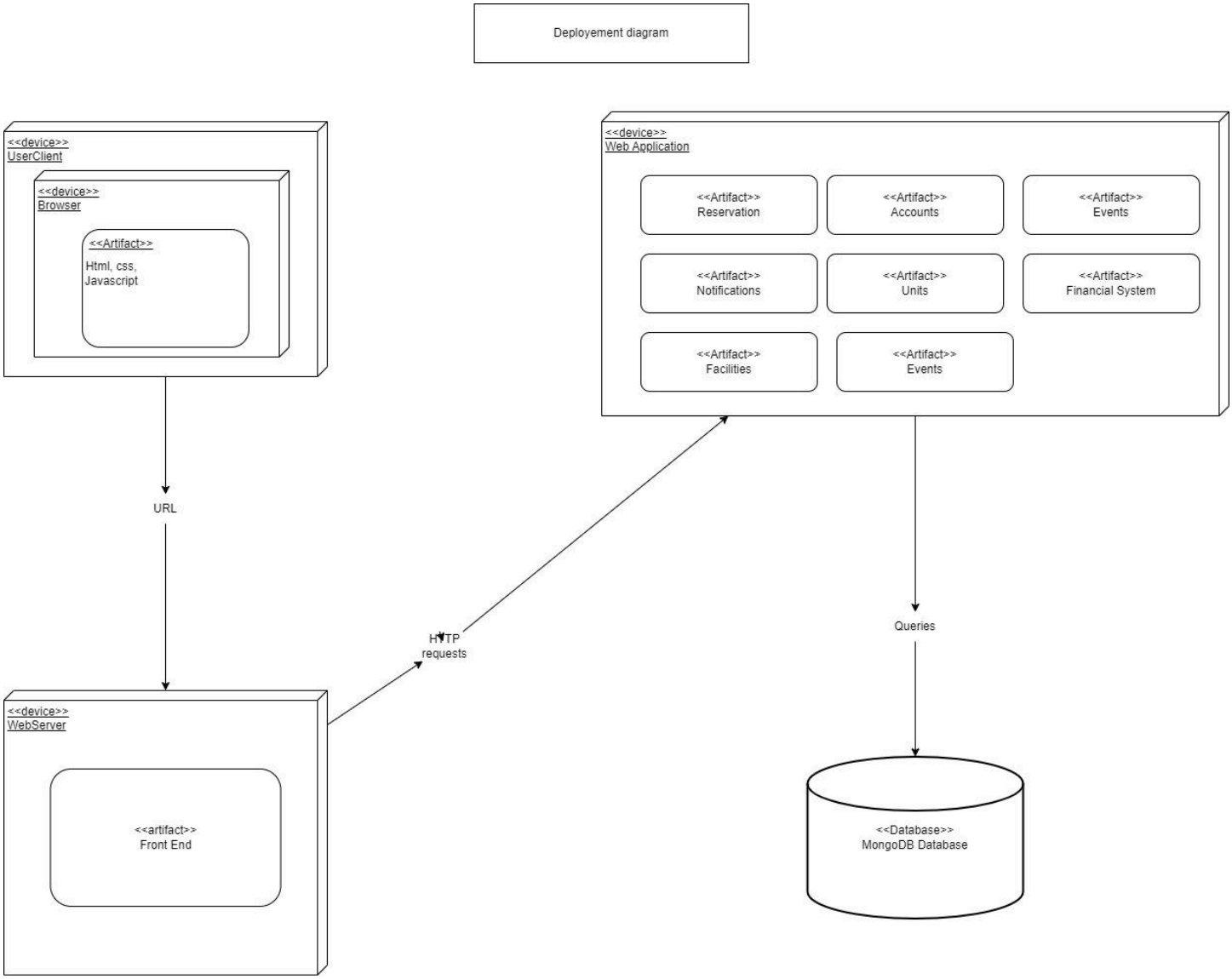
CLASS DIAGRAM



Component Diagram

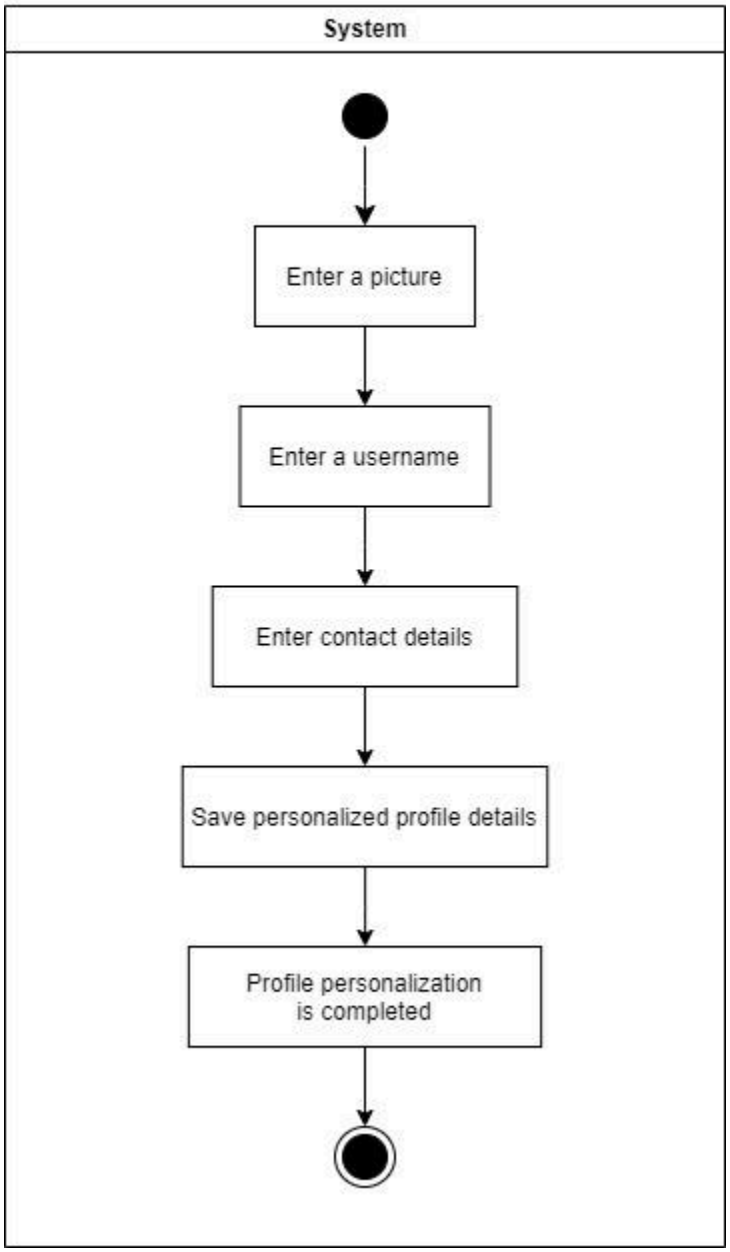


Deployment Diagram

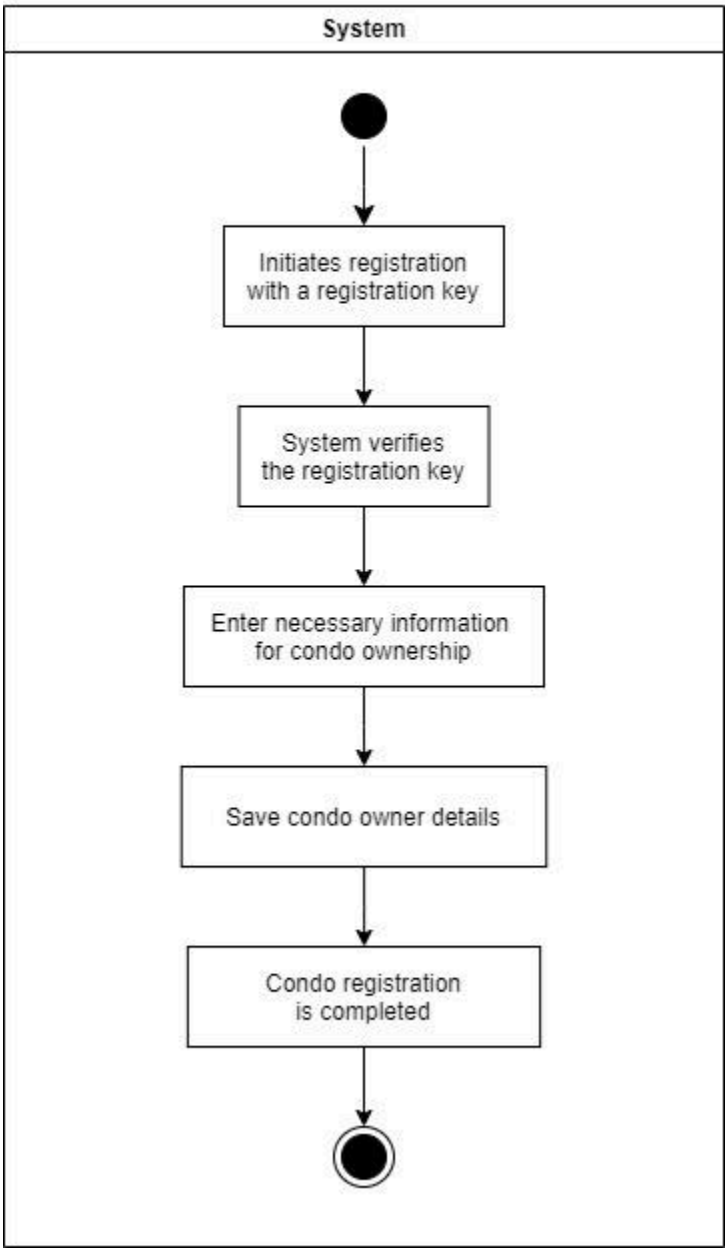


Activity Diagrams

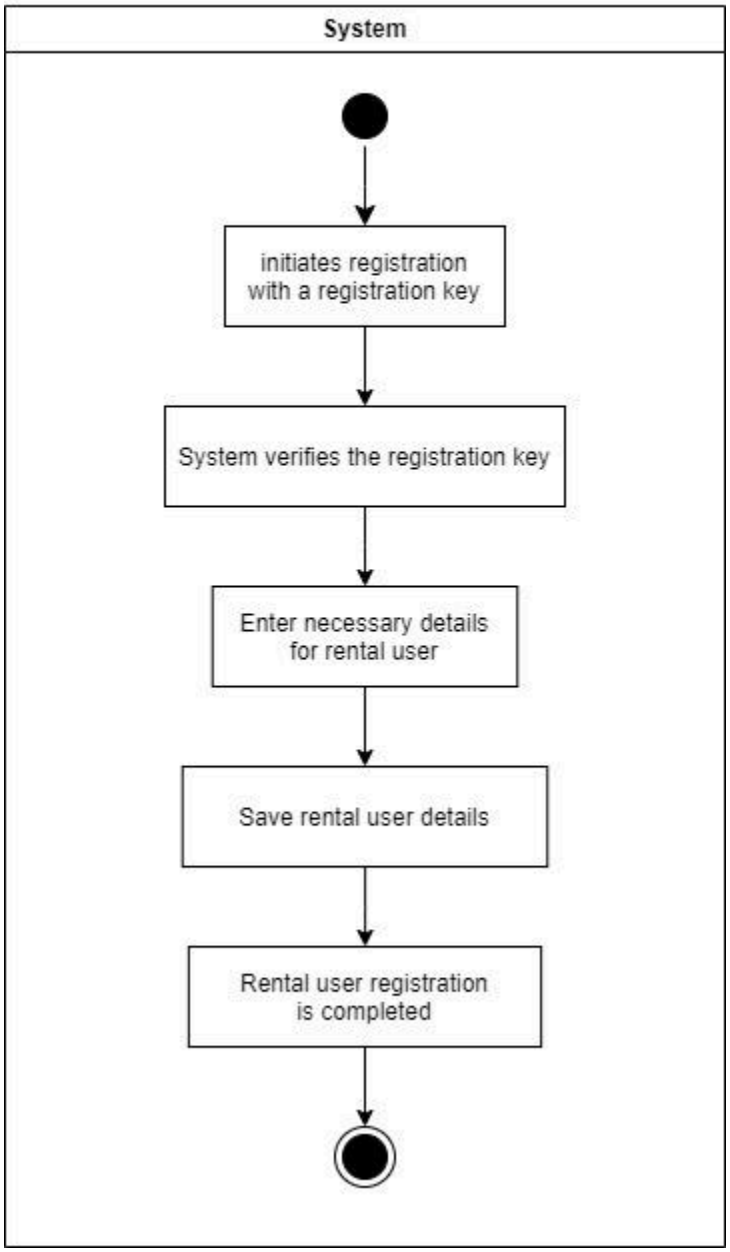
Title: Public User Profile Creation



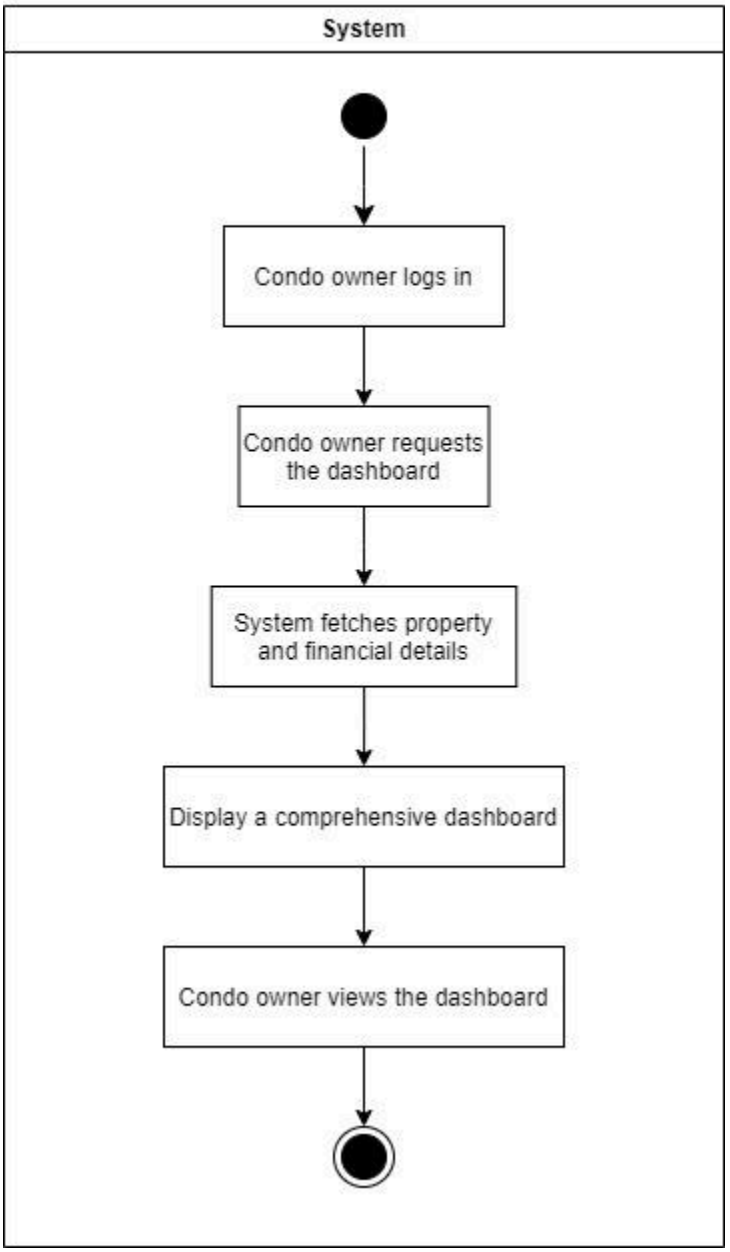
Title: Becoming a Condo Owner

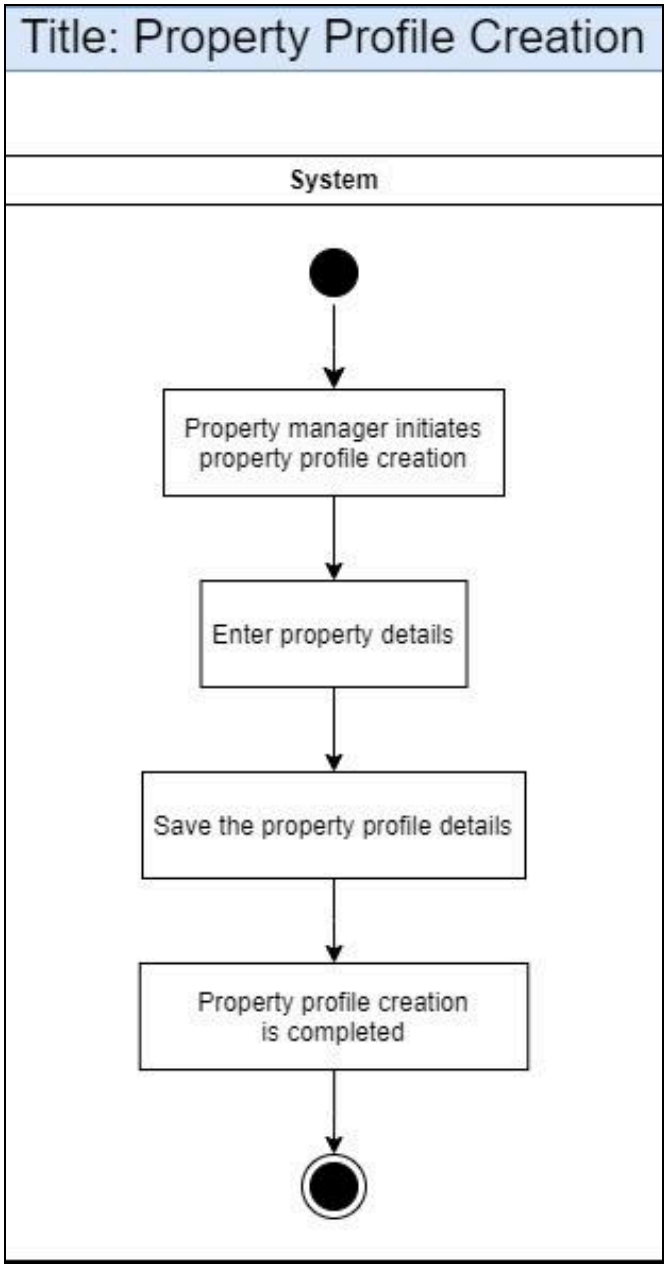


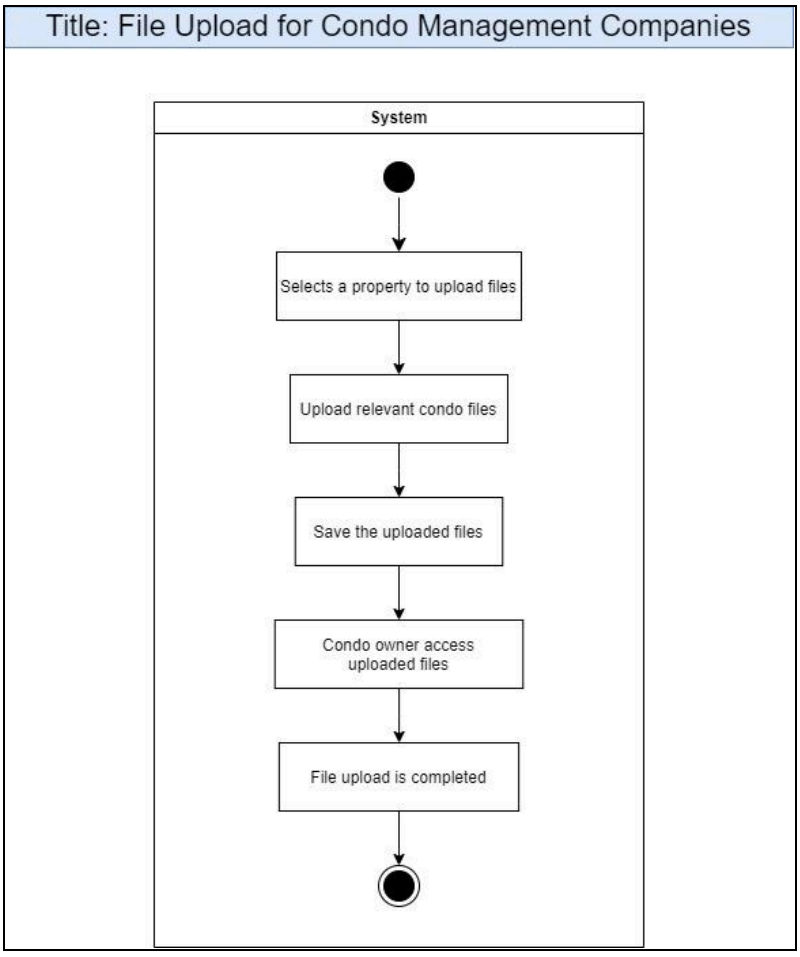
Title: Becoming a Rental User

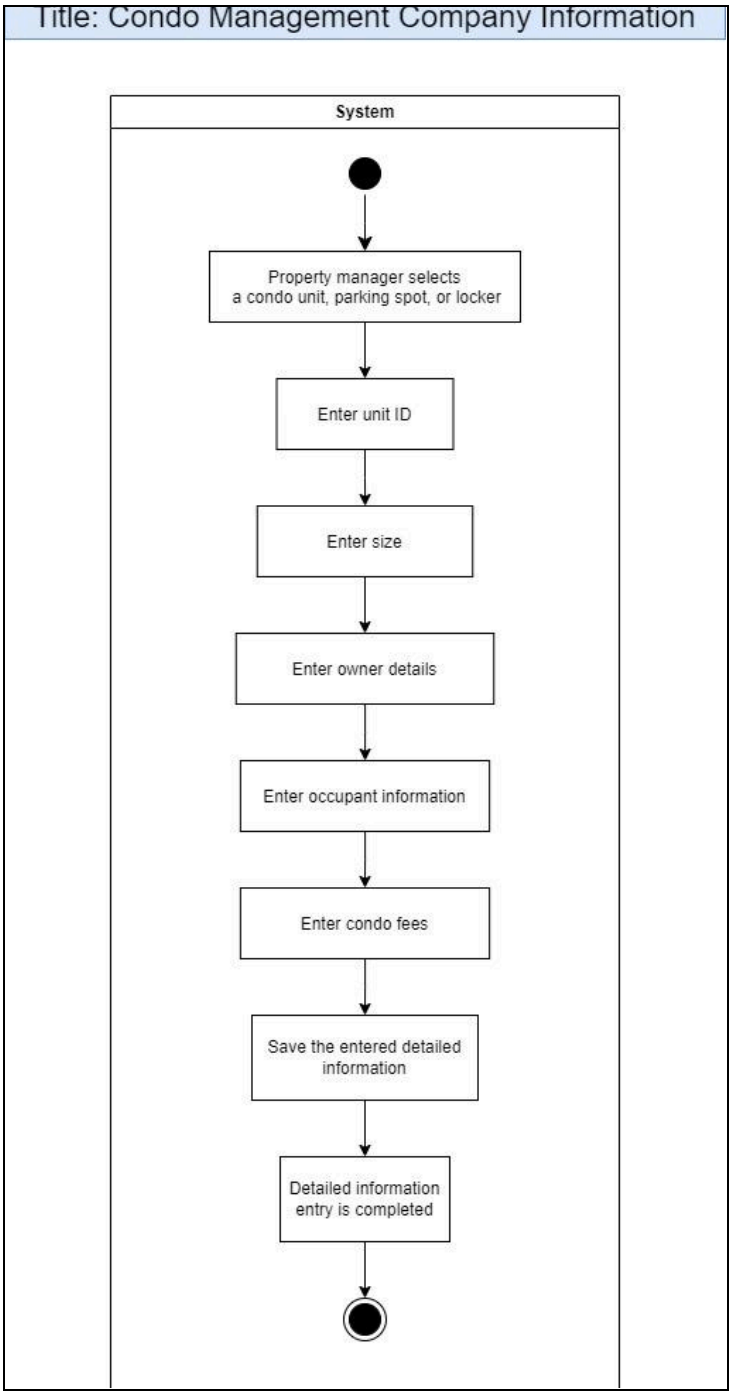


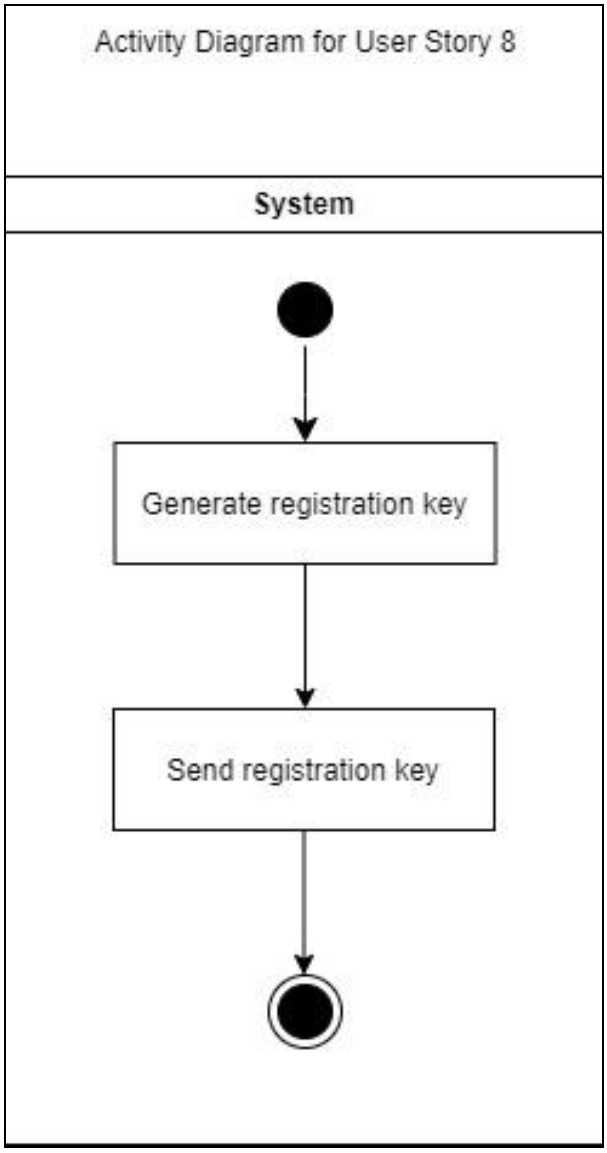
Title: Condo Owner Profile



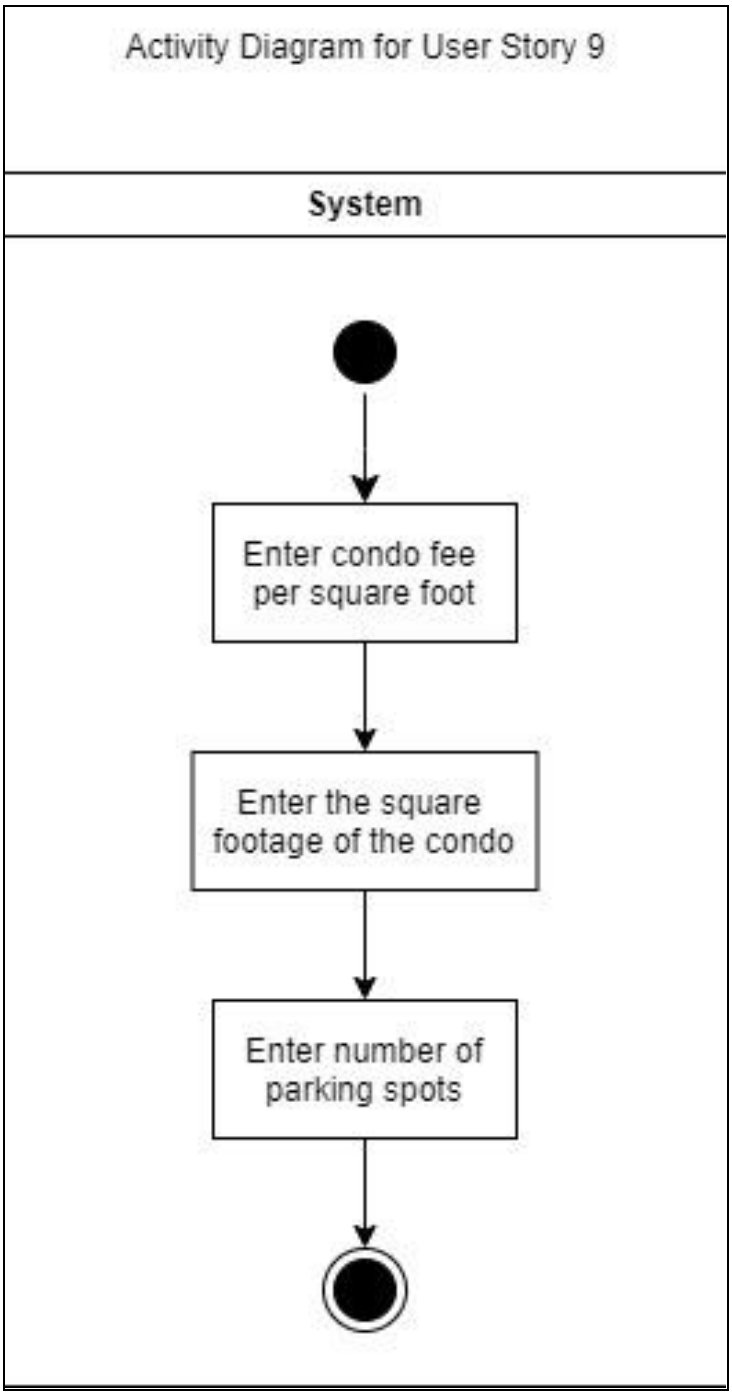

Title: Property Profile Creation
System
Property manager initiates property profile creation
Enter property details
Save the property profile details
Property profile creation is completed


Title: File Upload for Condo Management Companies
System
Selects a property to upload files
Upload relevant condo files
Save the uploaded files
Condo owner access uploaded files
File upload is completed

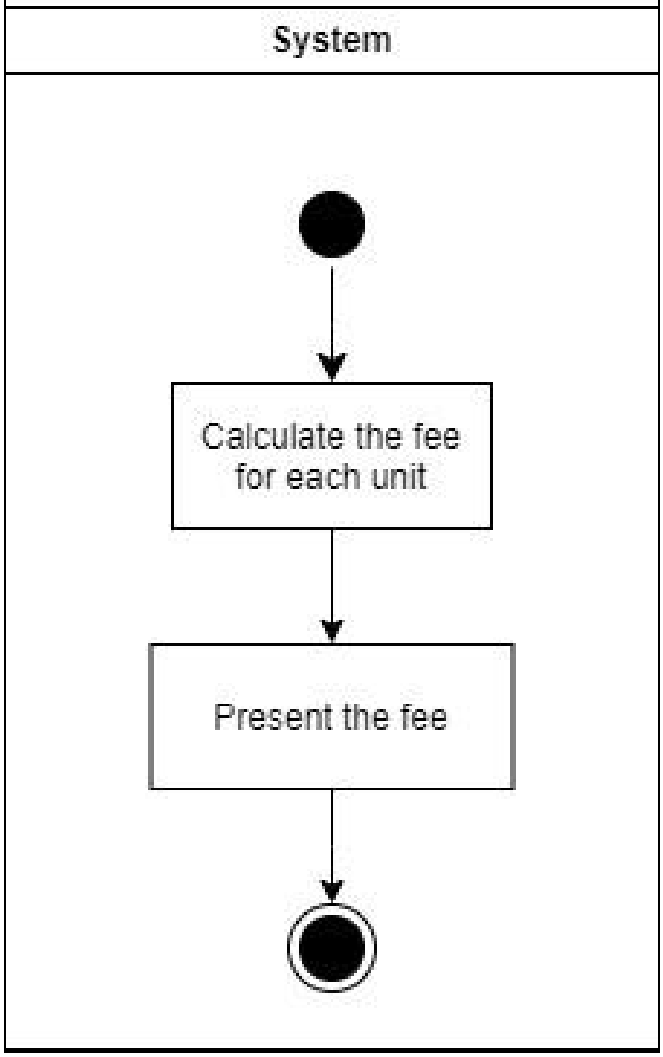

Title: Condo Management Company Information
System
Property manager selects a condo unit, parking spot, or locker
Enter unit ID
Enter size
Enter owner details
Enter occupant information
Enter condo fees
Save the entered detailed information
Detailed information entry is completed


Activity Diagram for User Story 8
System
Generate registration key
Send registration key

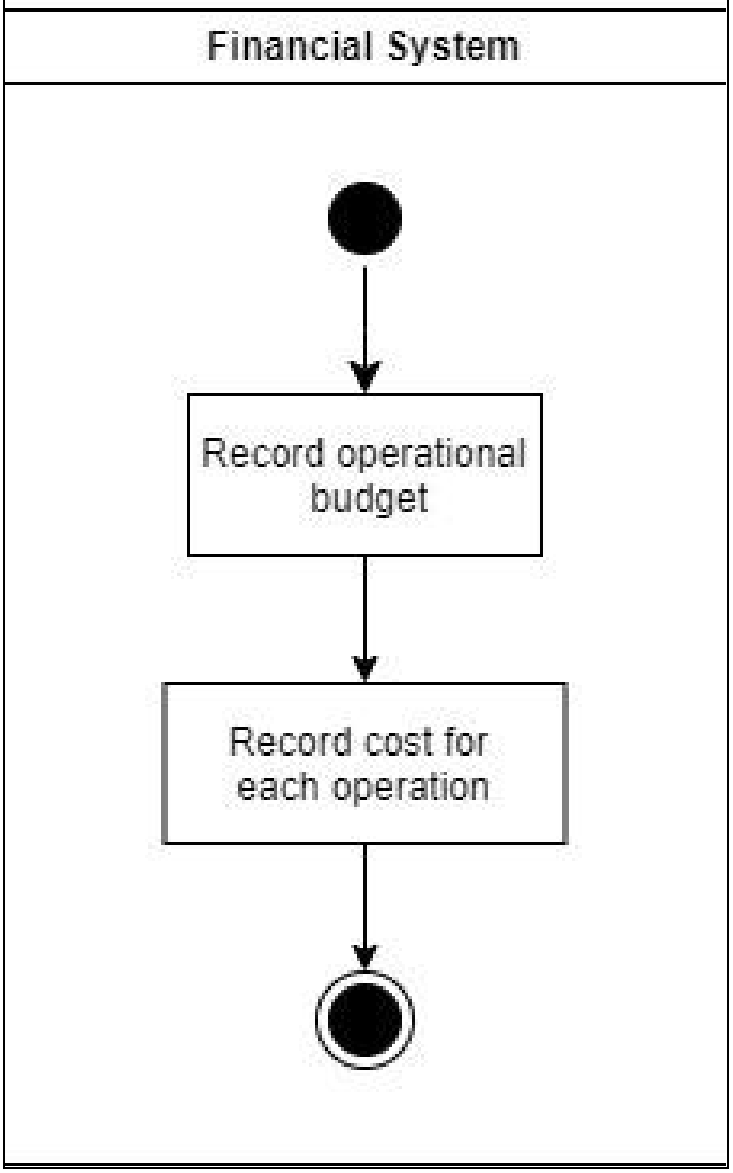
Activity Diagram for User Story 9



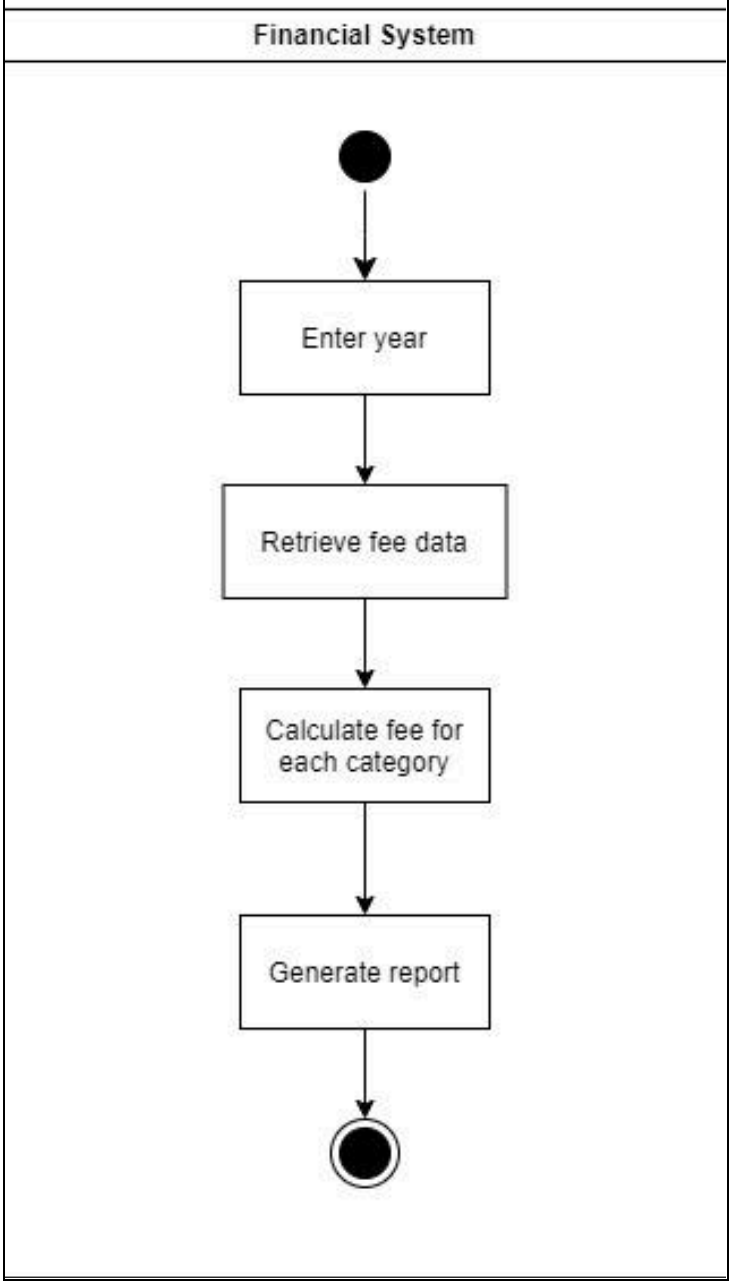
Activity Diagram for User Story 10



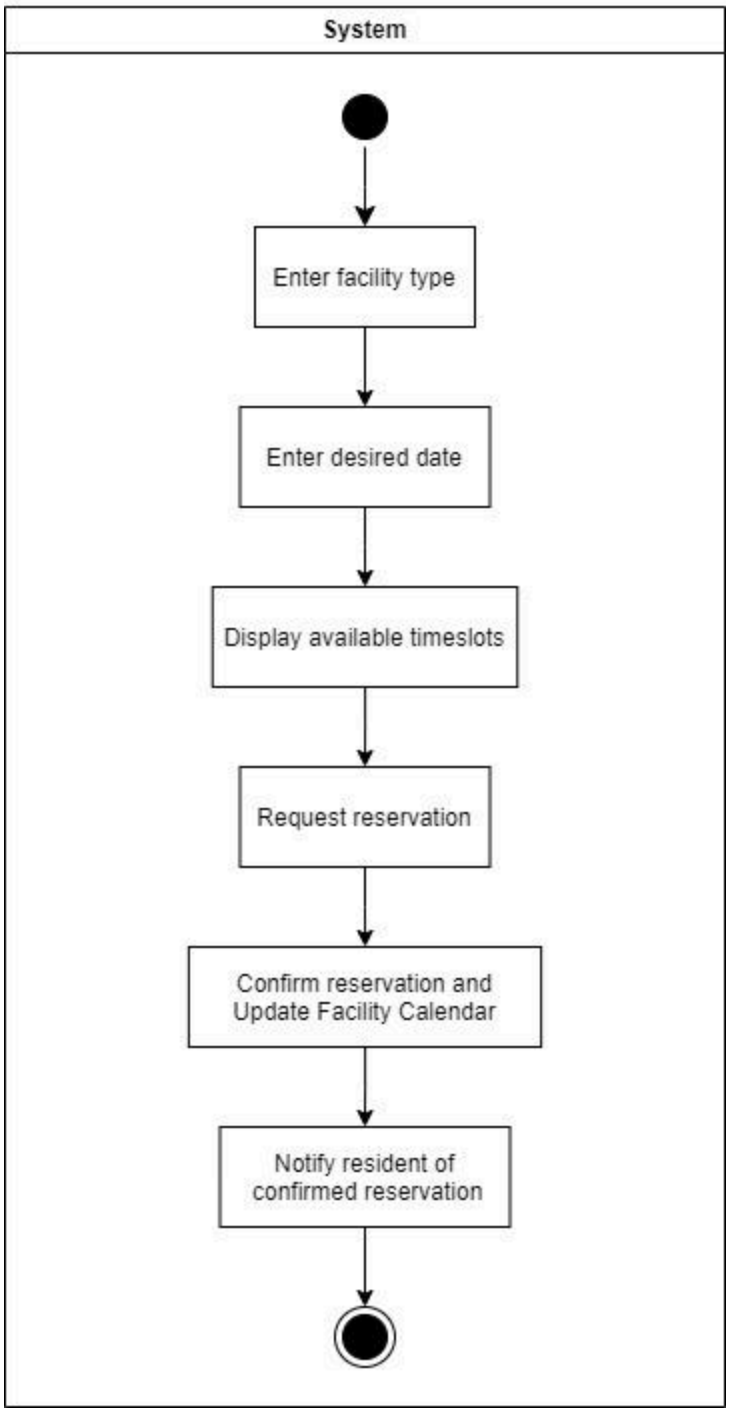
Activity Diagram for User Story 11



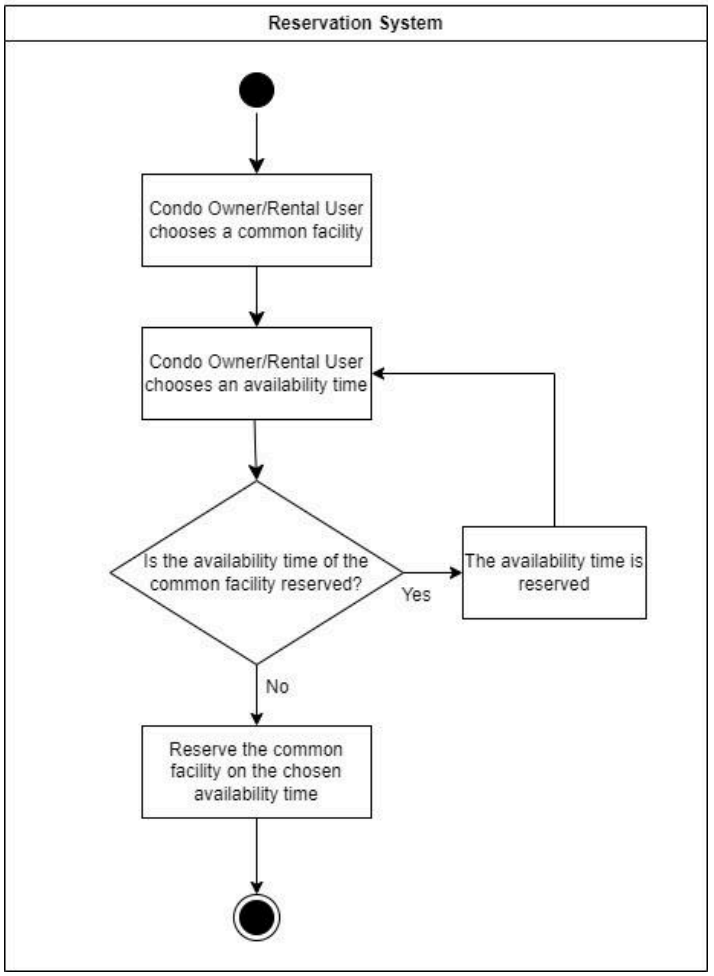
Activity Diagram for User Story 12



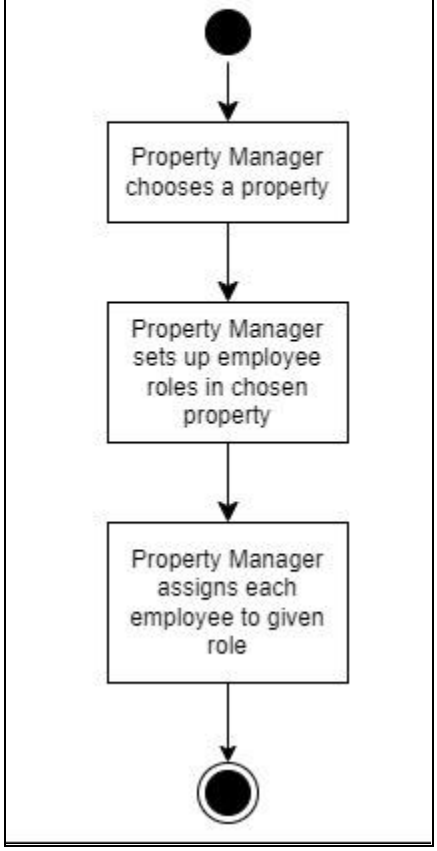
Activity Diagram for User Stories 13 and 14



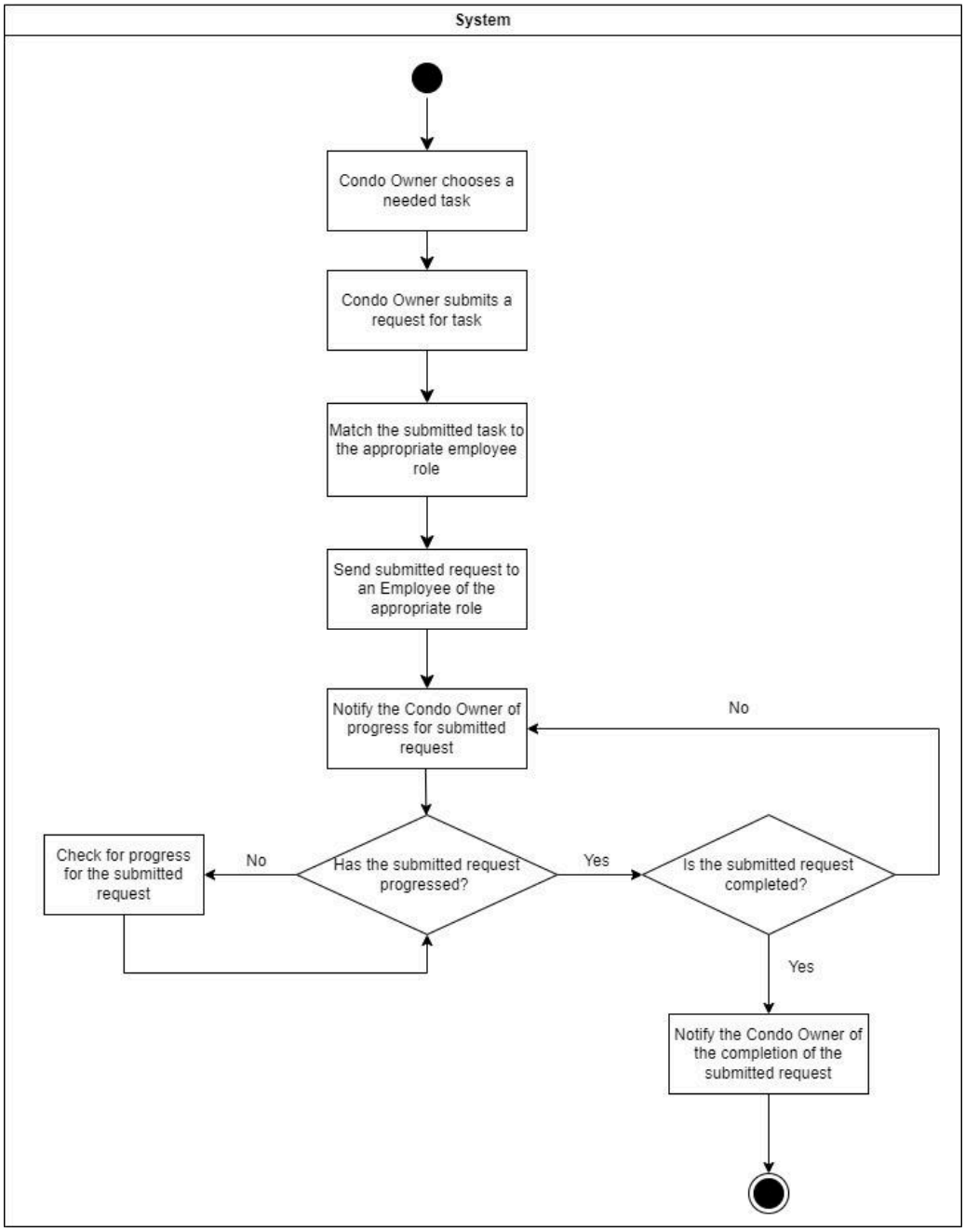
User Story #15-16



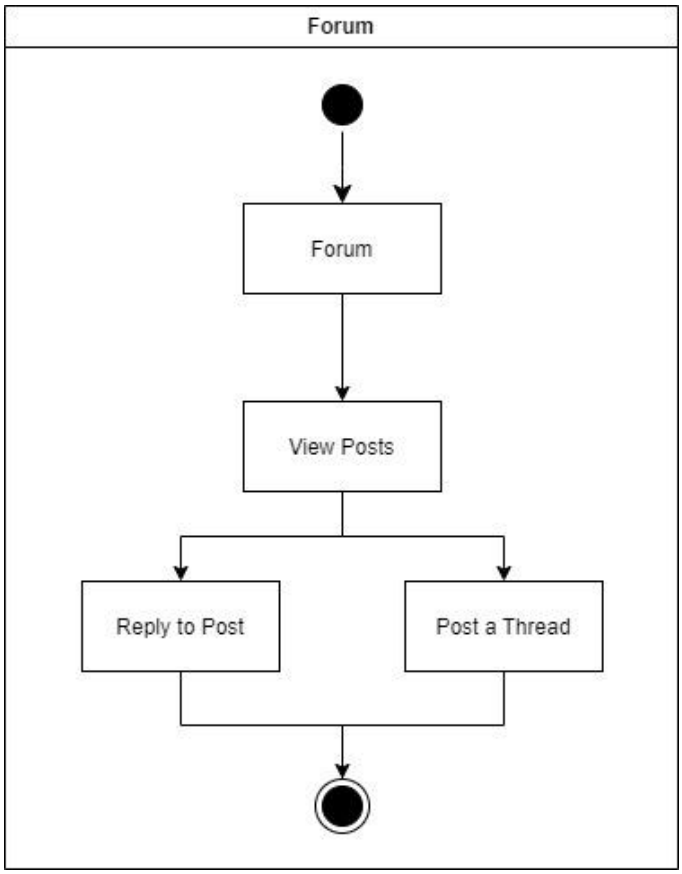
User Story #17



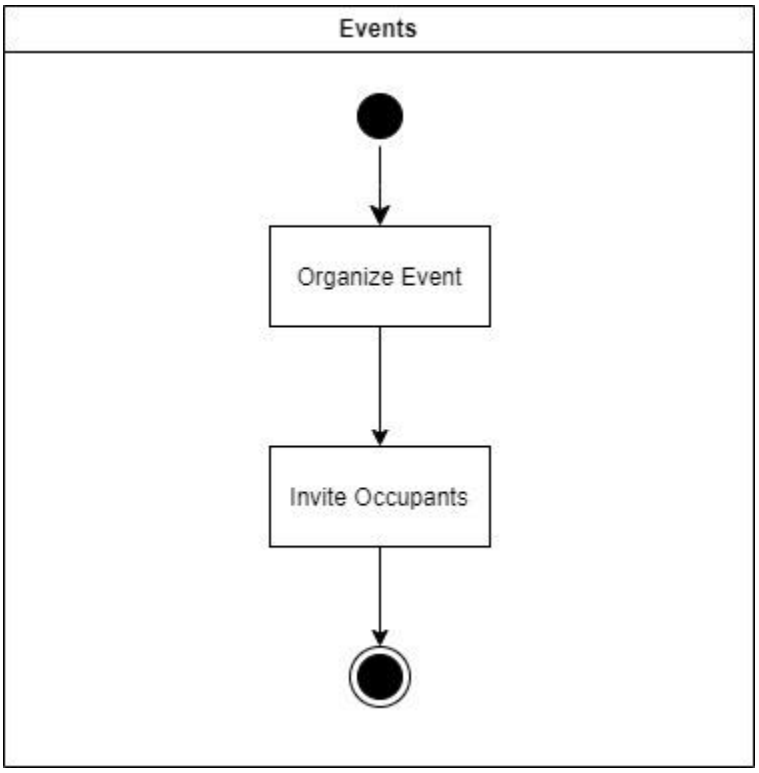
User Story #18-19-20



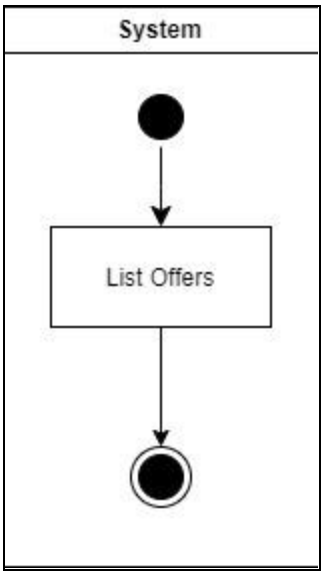
User Story # 21



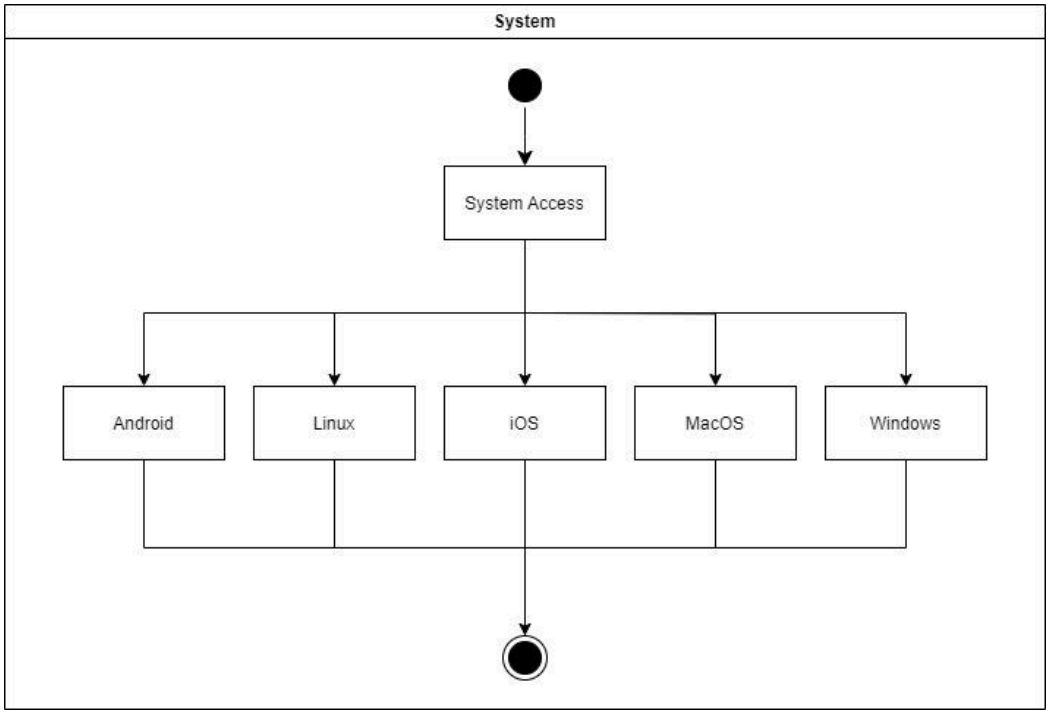
User Story # 22



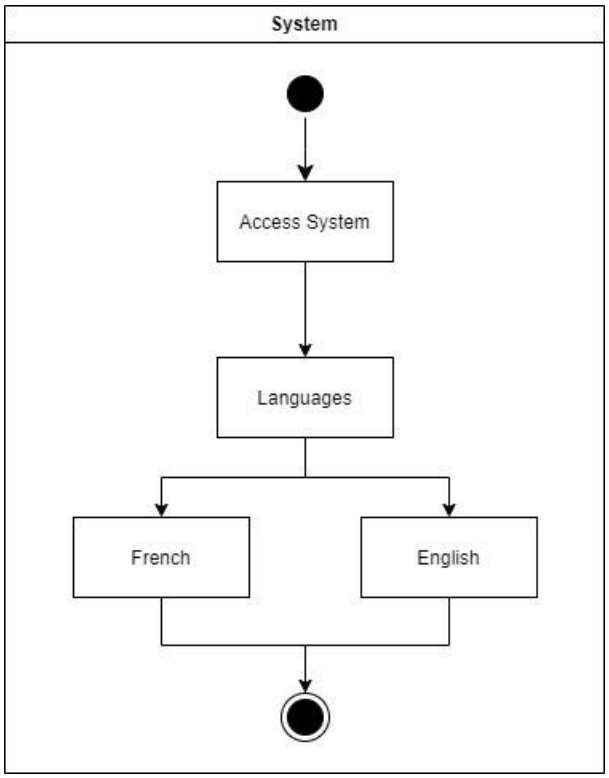
User Story # 23



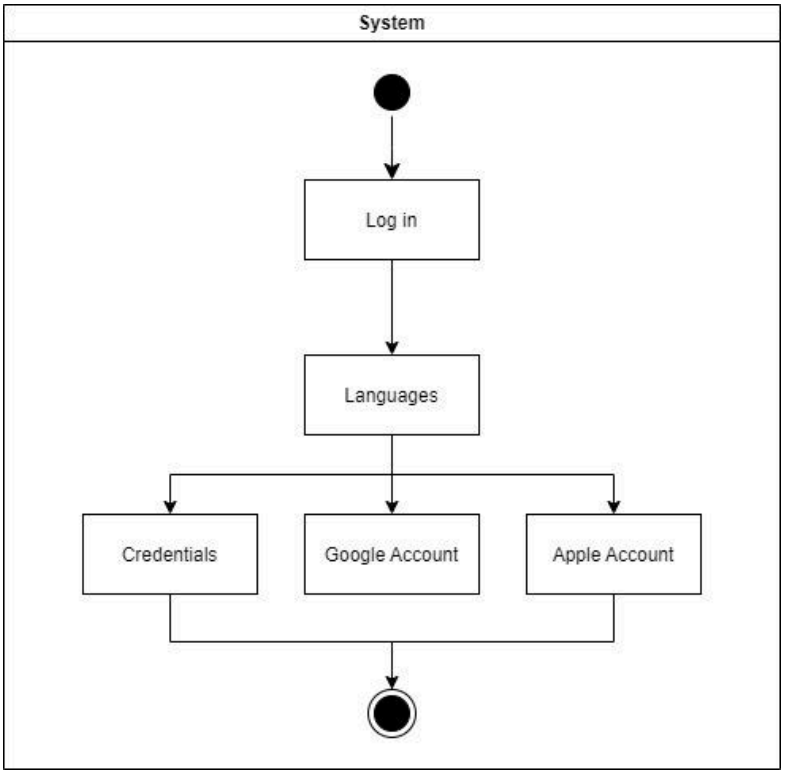
User Story # 24



User Story # 25

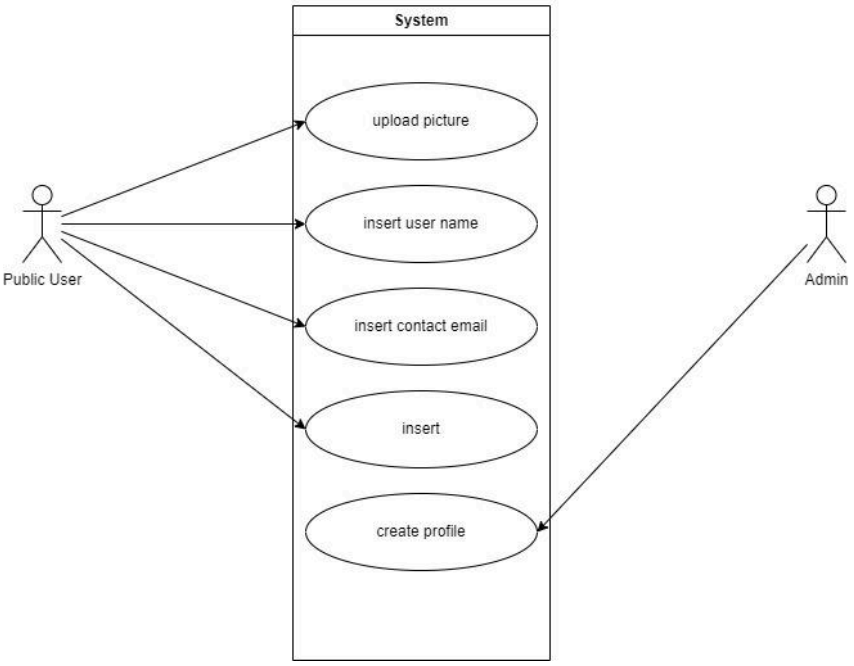


User Story # 26

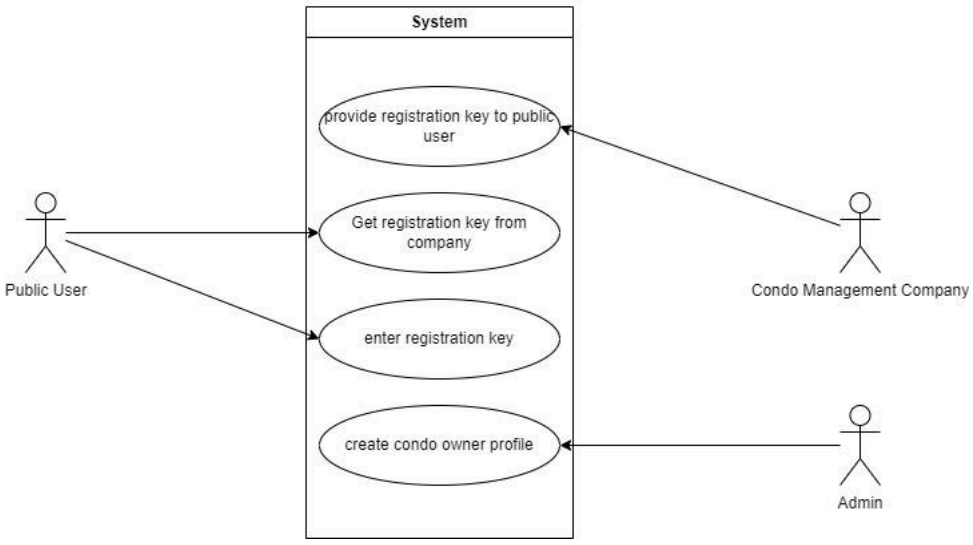


Use Case Diagrams

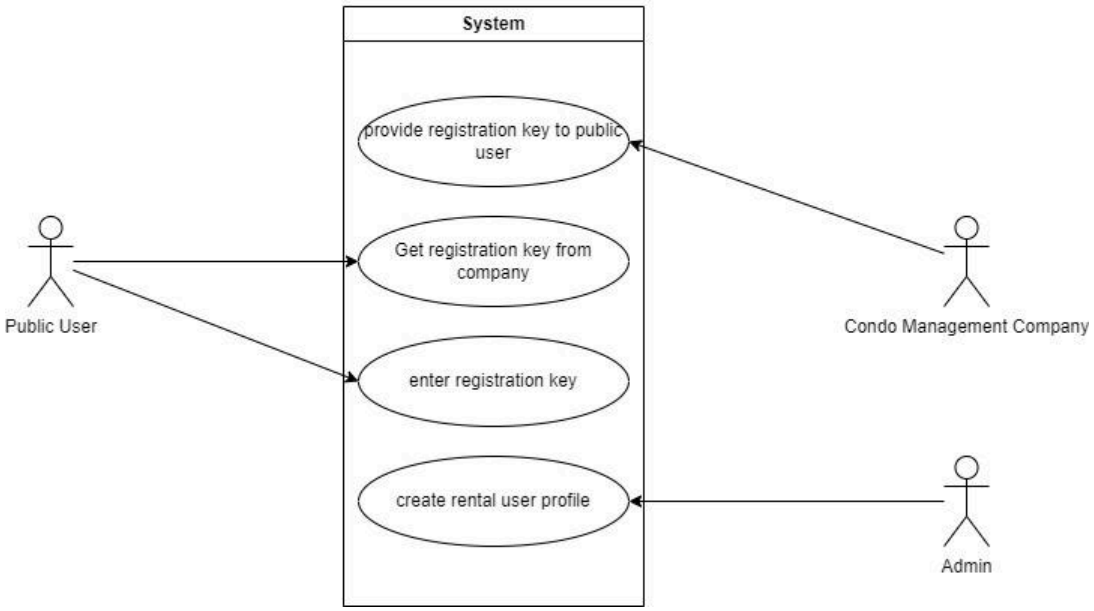
Title: Public User Profile Creation



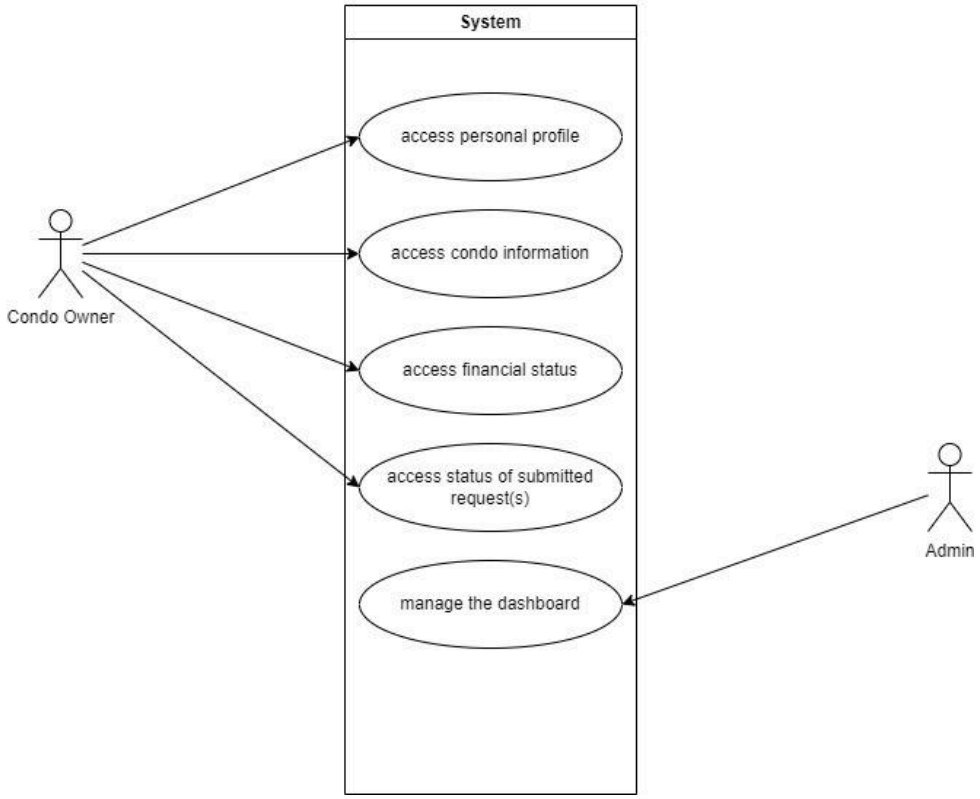
Title: Becoming a Condo Owner



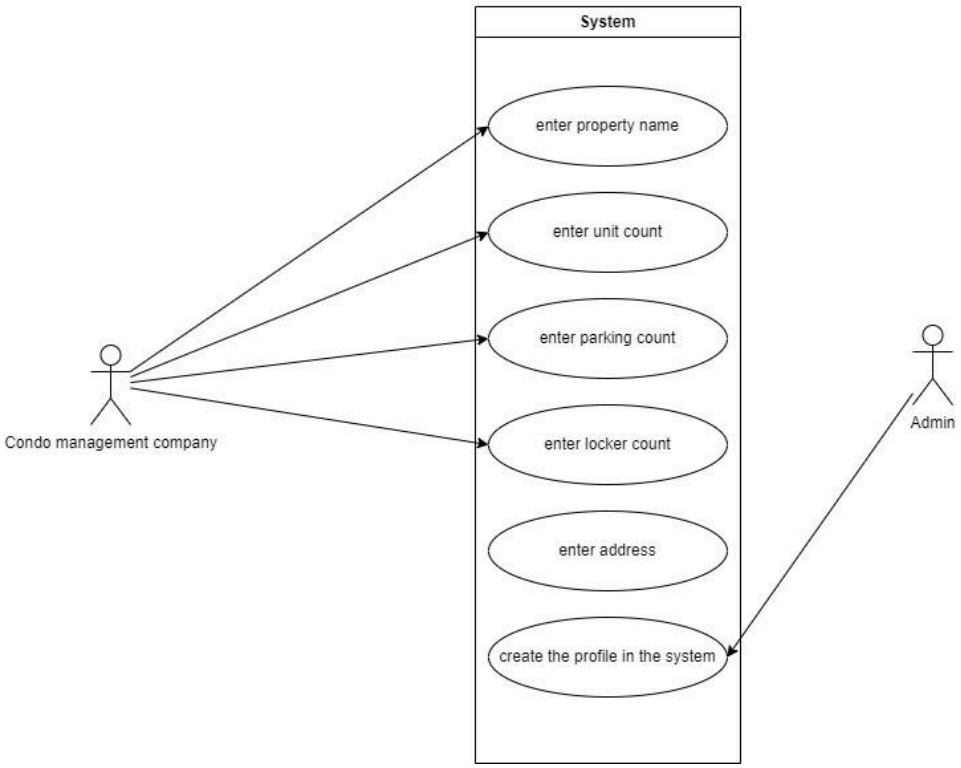
Title: Becoming a Rental User



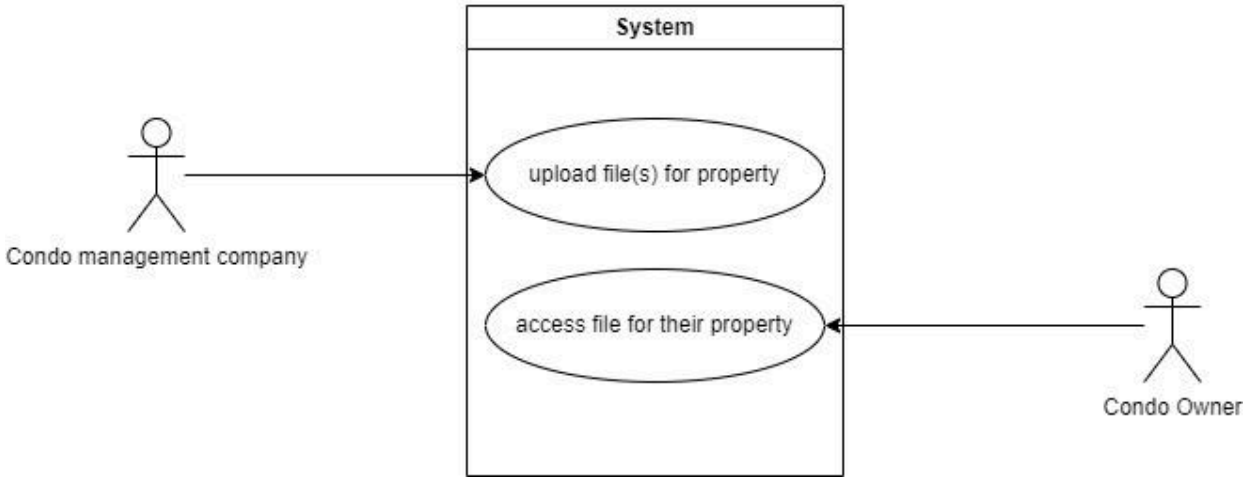
Title: Condo Owner Profile



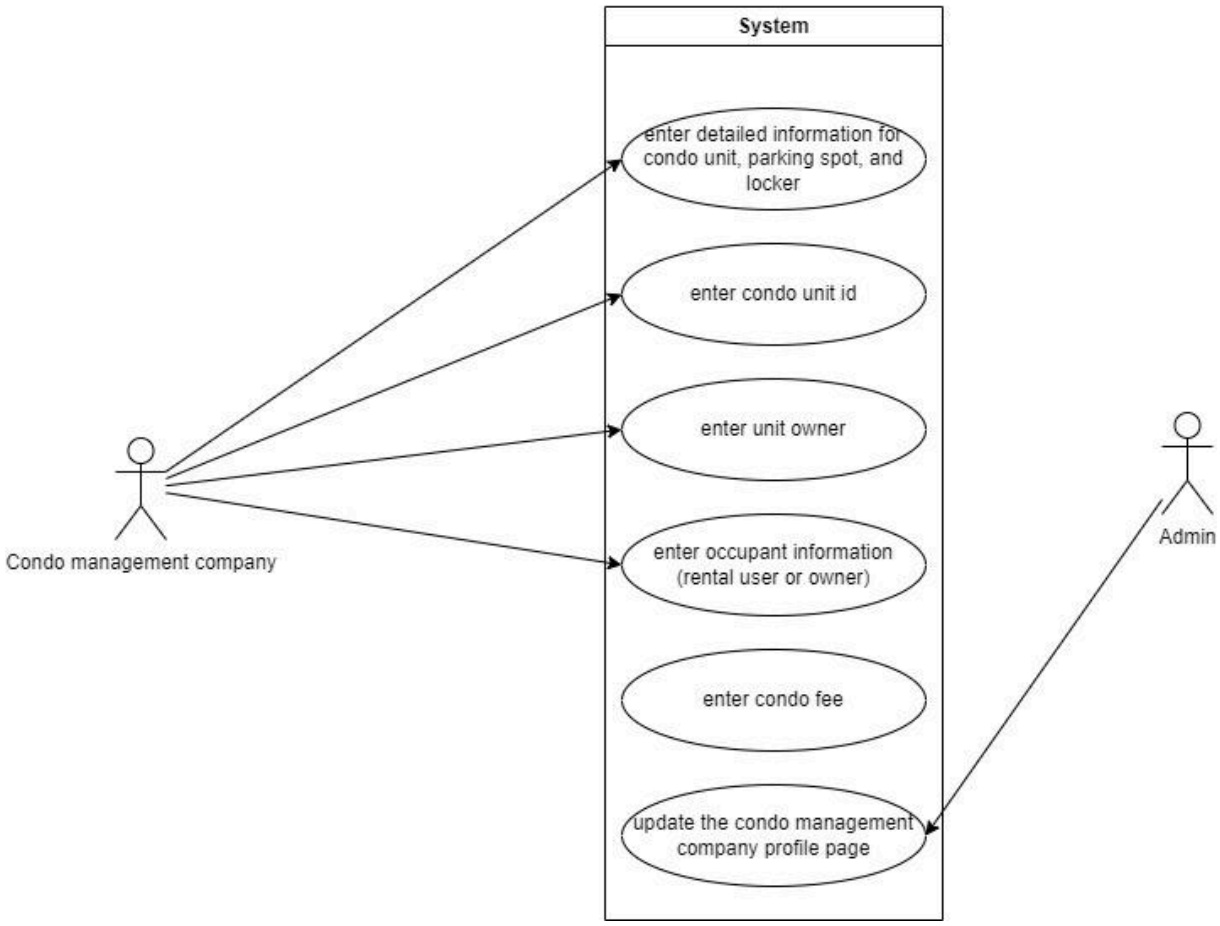
Title: Property Profile Creation



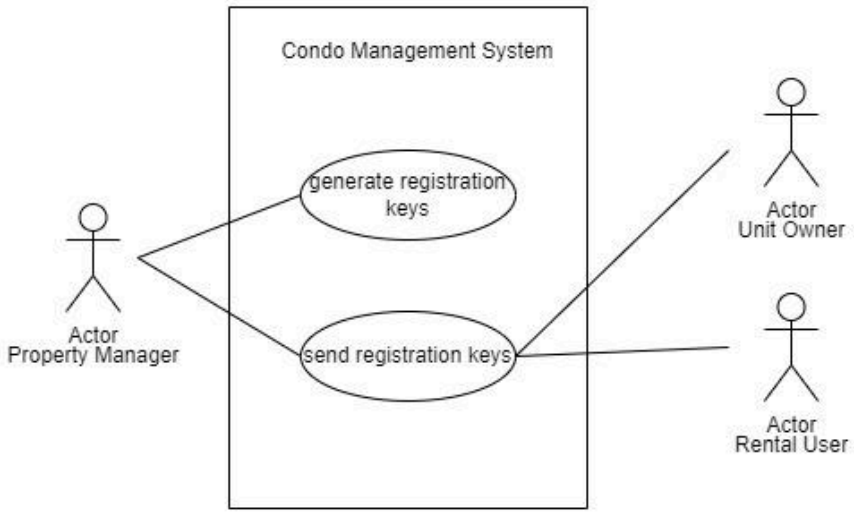
Title: File Upload for Condo Management Companies



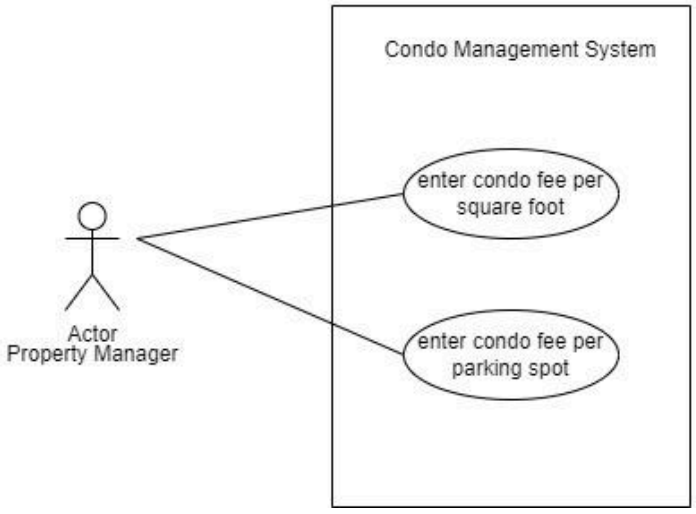
Title: Condo Management Company Information



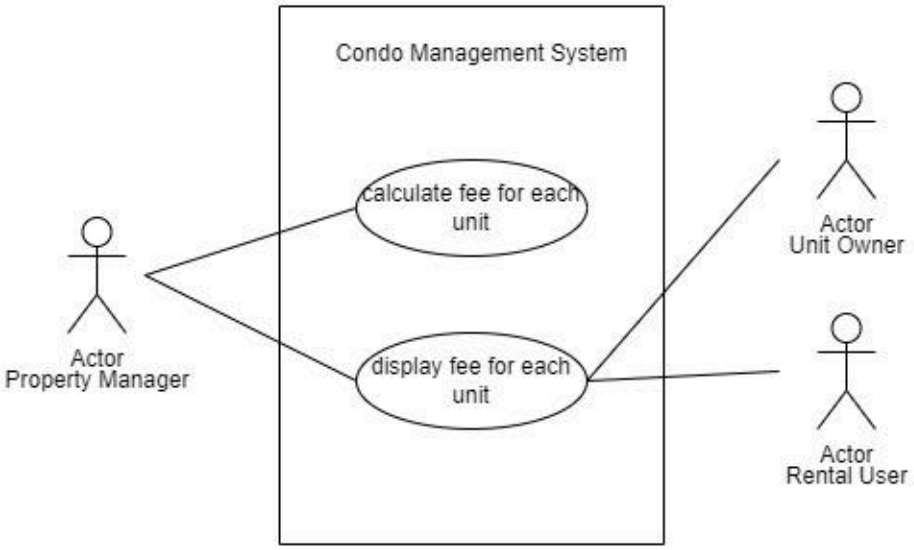
User Story #8



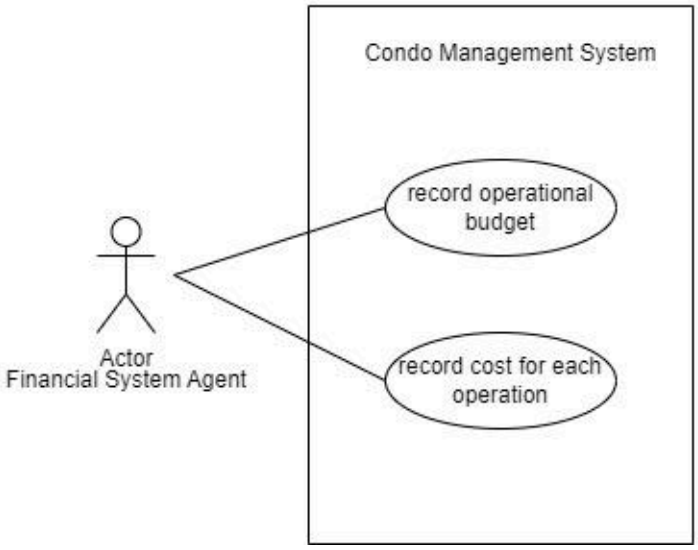
User Story #9



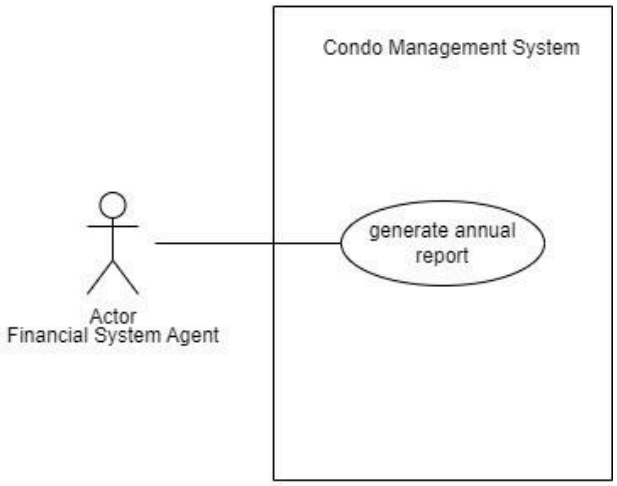
User Story #10



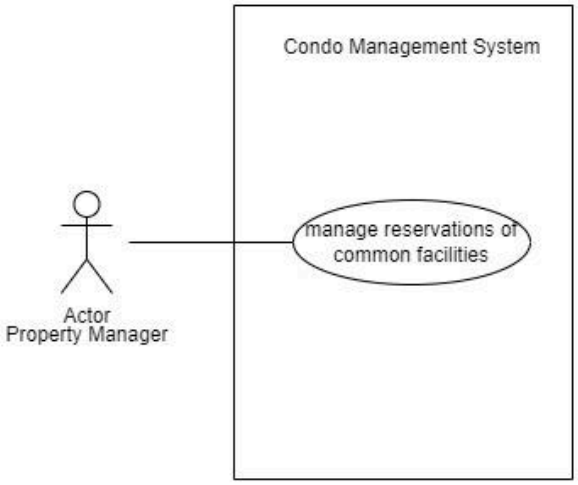
User Story #11



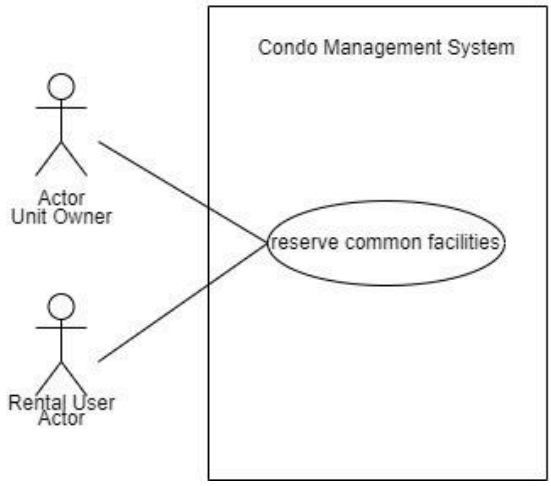
User Story #12



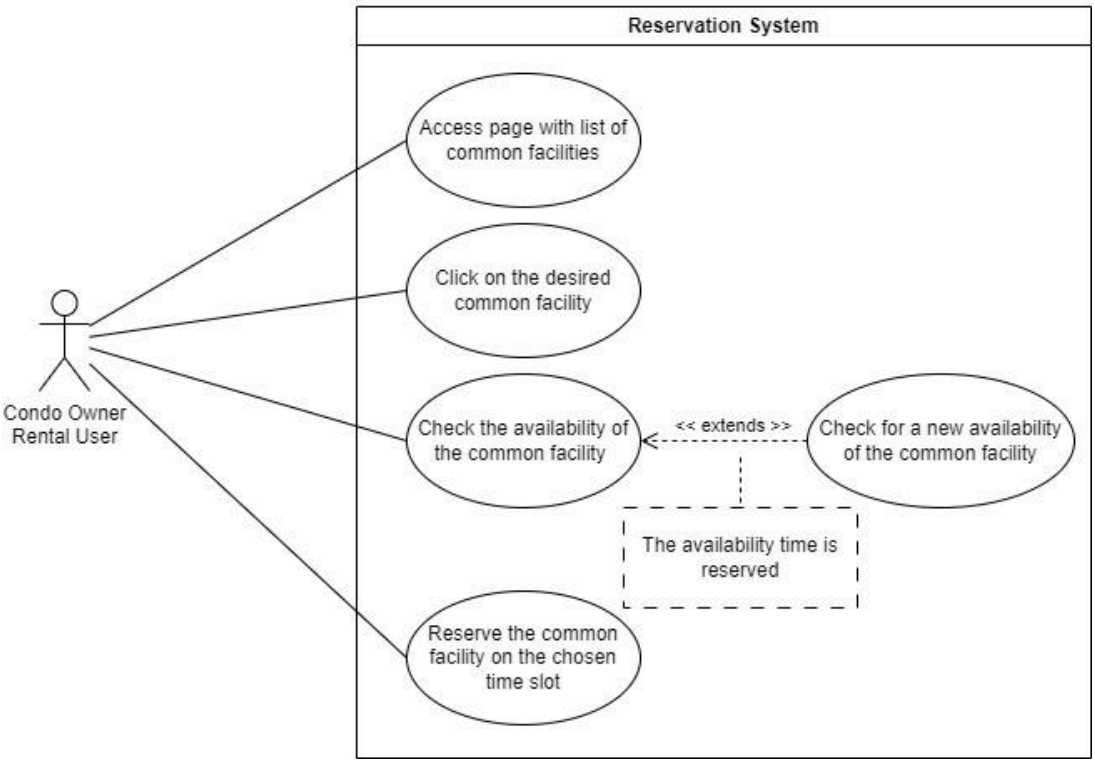
User Story #13



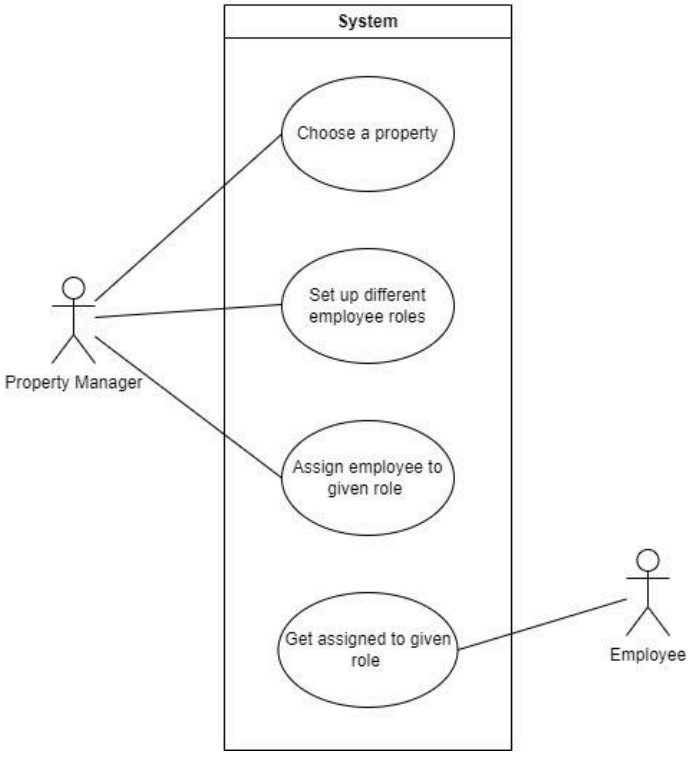
User Story #14



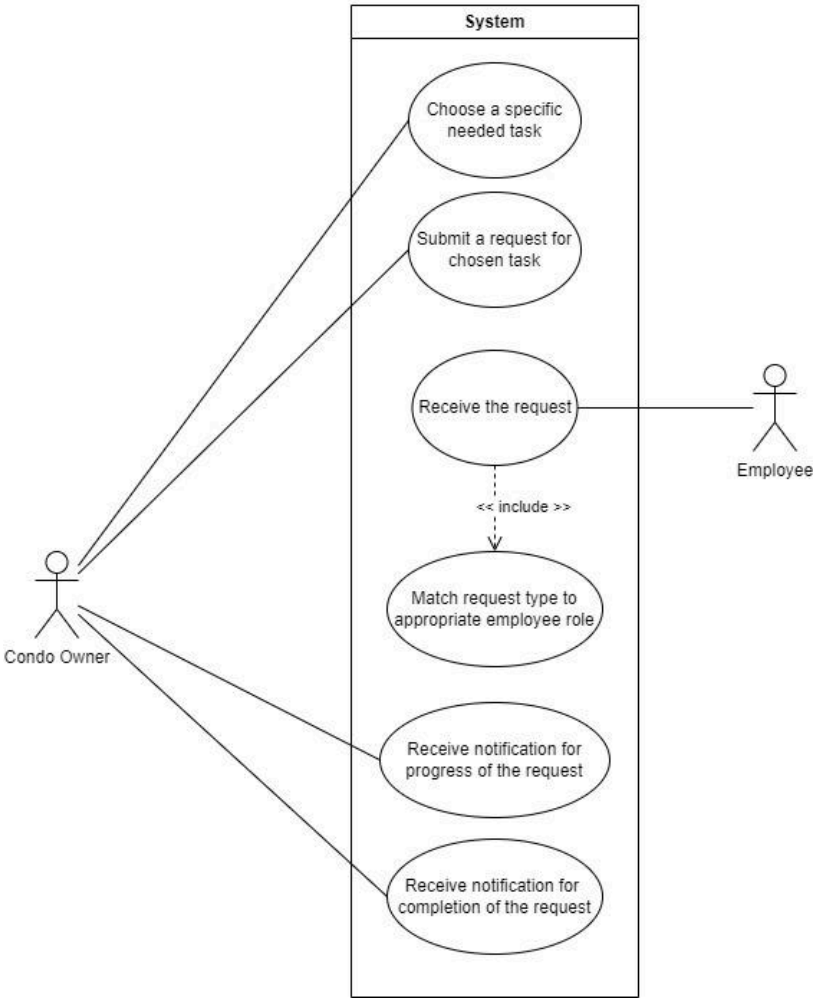
User Story #15-16



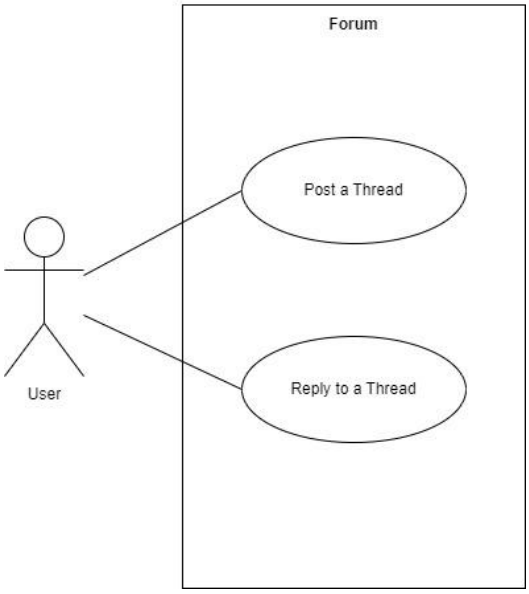
User Story #17



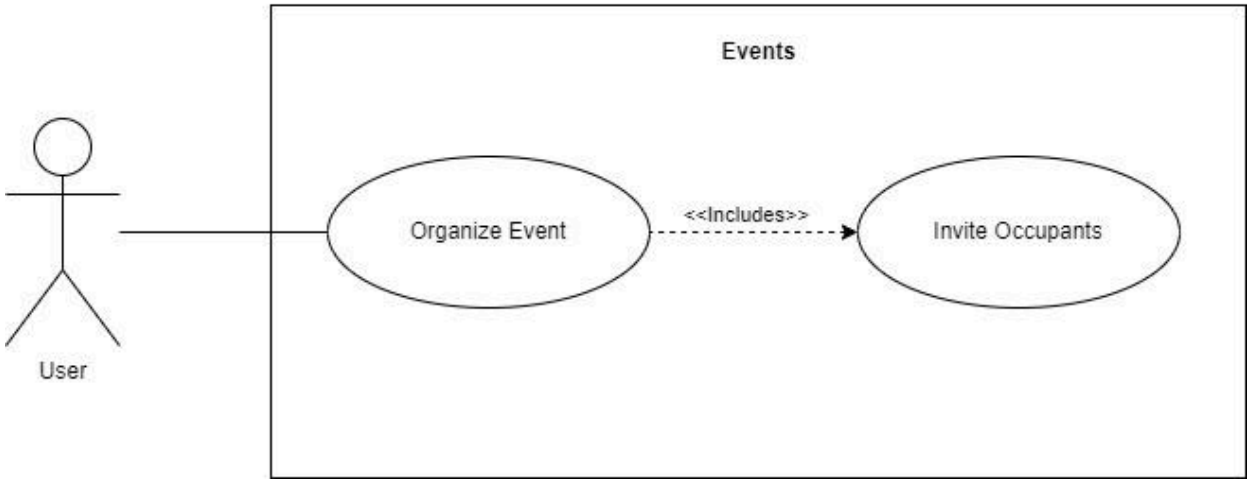
User Story #18-19-20



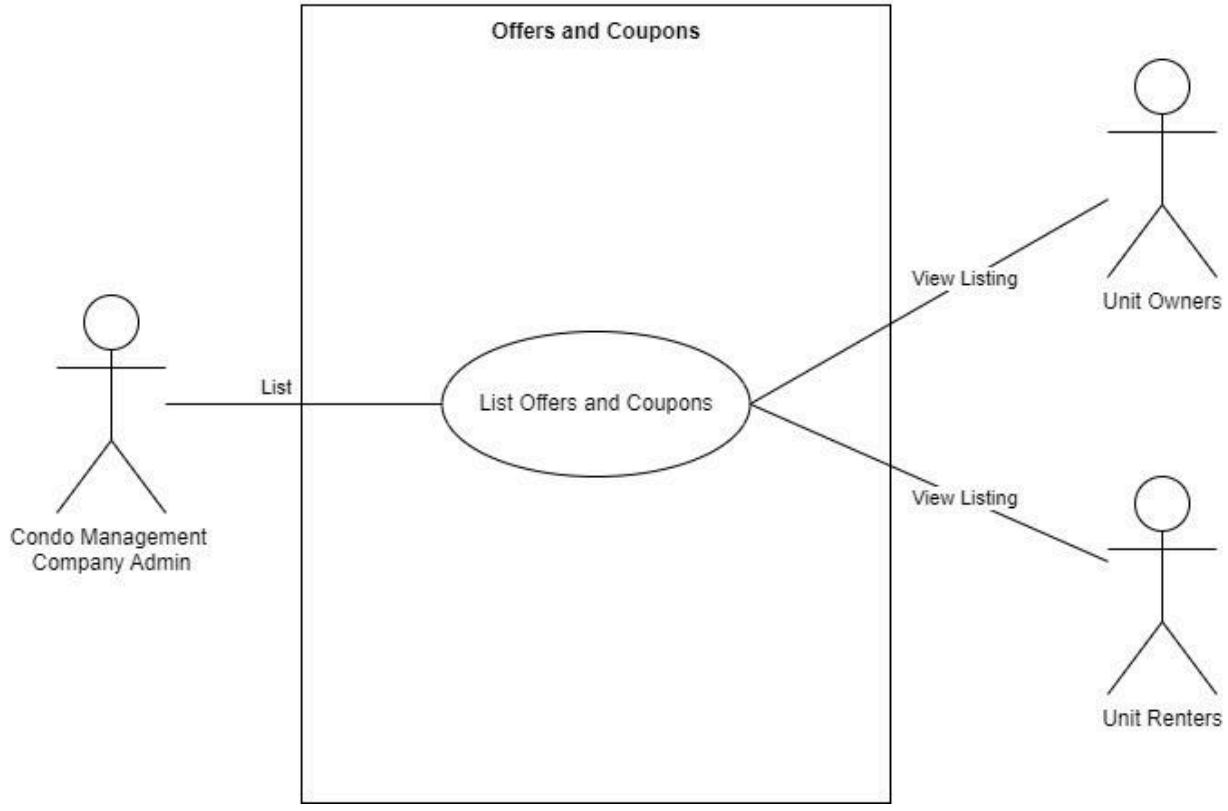
User Story # 21



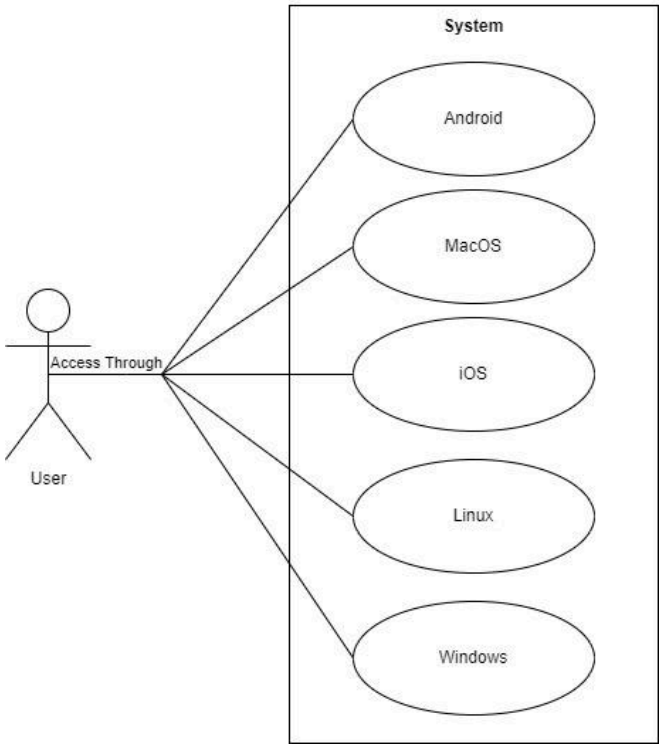
User Story # 22



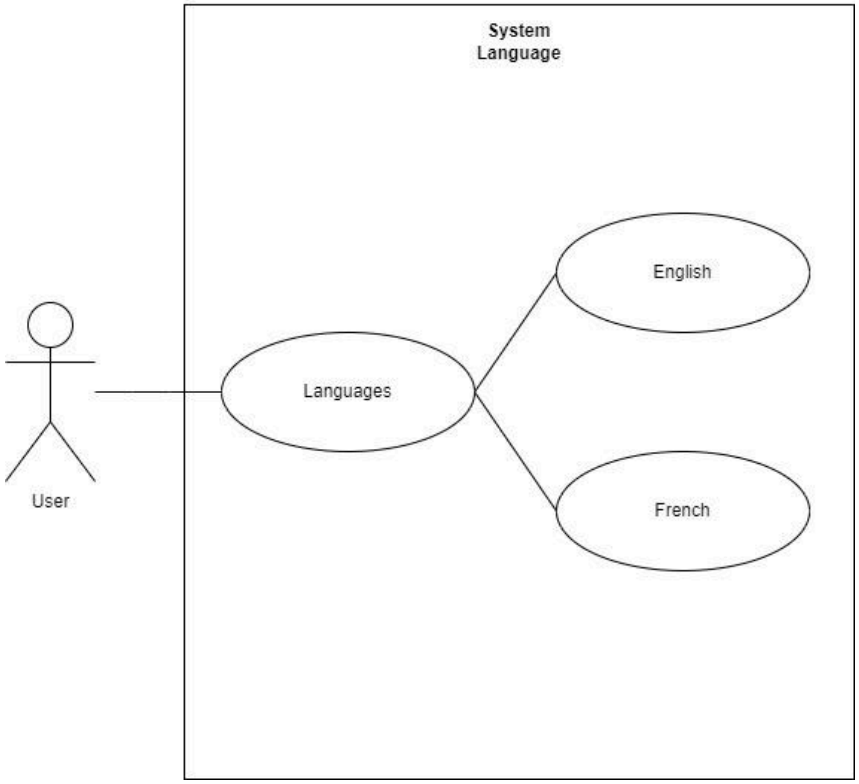
User Story # 23



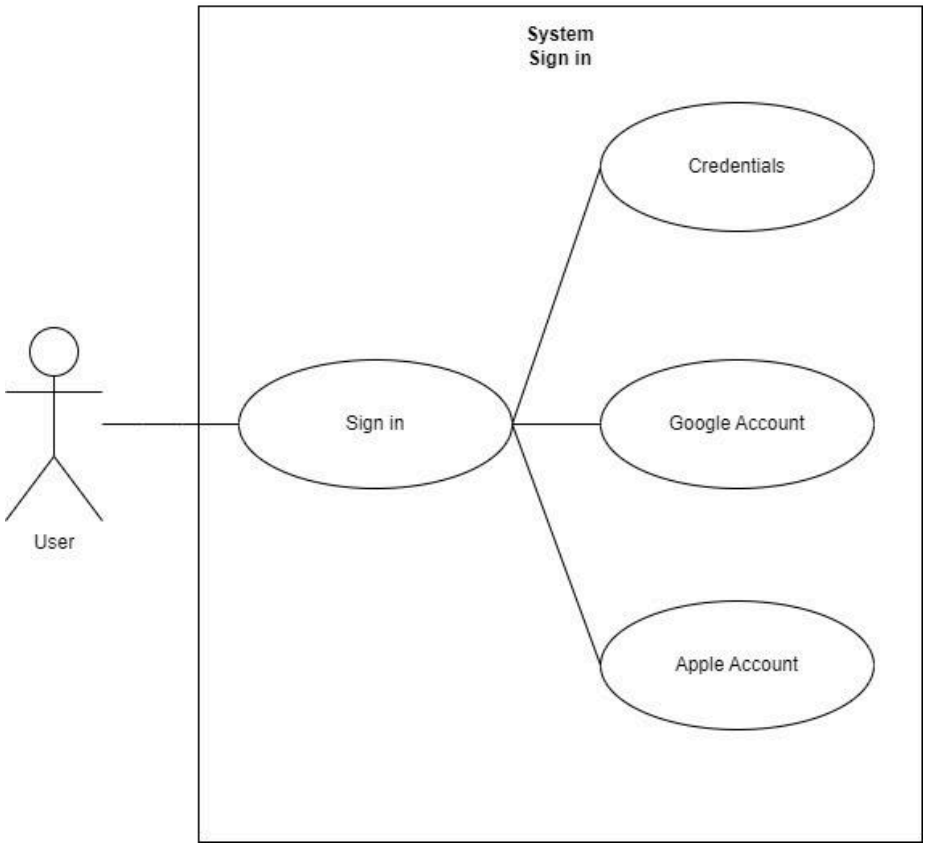
User Story # 24



User Story # 25



User Story # 26



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 / Probability	Low	Medium	High
Low			
Medium			
High			

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	Risk	Type	Probability	Impact	Mitigation
1	Technical complexity in integrating ReactJS, Python, and MongoDB	Technical	High	High	Regular code reviews and pair programming to ensure compatibility and identify issues early.
2	Inadequate expertise in ReactJS, Python or MongoDB among team members	Management	Medium	High	Allocate time for self-study and skill enhancement.
3	Unforeseen changes in project requirements	External	Medium	Medium	Implement Agile methodologies for flexible adaptation to changing requirements. Regular communication with stakeholders (TA) to manage expectations.
4	Security vulnerabilities	Technical	Medium	High	Regular security audits, encryption protocols, and adherence to secure coding best practices.
5	Data loss or corruption	Technical	Medium	High	Implementing regular backups, redundancy measures, and robust error-handling mechanisms for the MongoDB database.
6	User acceptance and adoption	Management	Medium	High	Conduct user testing and feedback sessions, incorporate user-centric design principles, and provide comprehensive user training and support.

Table [1]: List of identified risks

Risk Analysis UrbanKey

Risk analysis UrbanKey.xlsx

Short Sprint #1 Retrospective

Crafting Excellence

Key Takeaways from Our Project Postmortem

Introduction

As we wrap up our first sprint, it's imperative to reflect on our progress, achievements, and areas for improvement. This postmortem offers a comprehensive evaluation of our accomplishments, covering key aspects like product vision, software architecture, risk management, prototype development, testing, retrospective analysis, and planning for the upcoming sprint. Each component plays a crucial role in shaping our project's success, and through this review, we seek valuable insights to refine our processes and deliver an exceptional product.

We began by formulating user stories tailored to the product owner's requirements, laying the groundwork for our development process. These stories guided the creation of activity and use-case diagrams, providing a structured blueprint for our work. With a clear understanding of the project's scope, we proceeded to build essential architectural elements, including the domain model, class diagram, component diagram, and deployment diagram, ensuring a robust framework for our webpage. Concurrently, the development team focused on implementing the initial functionalities of our product, beginning with startup pages such as Login, Sign Up, and User Profile. Through collaborative efforts and meticulous attention to detail, we laid a solid foundation for the subsequent stages of development.

Additionally, amidst our sprint activities, we recognized the importance of integrating efficient project management tools, such as an Enterprise Resource Planning (ERP) system. This system will streamline our processes, enhance communication, and facilitate resource allocation, ultimately contributing to the success of our project. As we reflect on our progress and plan for the upcoming sprint, the implementation of an ERP system will further strengthen our project management framework, ensuring seamless coordination and optimal utilization of resources.

Going into the first sprint, there was a degree of uncertainty regarding our ability to cover all requirements and complete all deliverables within the designated timeframe. Despite this initial uncertainty, we are pleased with the progress achieved thus far. Through diligent teamwork and focused efforts, we have made significant strides in laying the foundation for our project. While challenges may have arisen along the way, our determination and commitment have enabled us to overcome obstacles and move forward with confidence.

What went wrong

1 - Building the Domain Model

During the initial stages of constructing the domain model, our team encountered uncertainties stemming from the diverse and sometimes overlapping requirements laid out for the project. These complexities presented a challenge as we grappled with how best to structure the domain model to accurately reflect the intricacies of the project. The sheer variety of requirements from different stakeholders added to the complexity, making it difficult to establish a cohesive framework.

To address these challenges, we convened team meetings aimed at dissecting each requirement and organizing them systematically. Through collaborative discussions and brainstorming sessions, we were able to identify common themes and patterns, which facilitated the segmentation of requirements into manageable components. This approach allowed us to break down the complexities of the project into more digestible units, enabling us to create a domain model that aligned more closely with the project's objectives. By leveraging the collective expertise and insights of the team, we were able to navigate through the uncertainties and establish a solid foundation for the domain model.

2 - Limited Time

Another challenge we faced during the initial phase of the project was the limited time available to fully comprehend the problem and grasp the intricacies of the project description. With deadlines looming and a complex project to unravel, there was a sense of urgency to gain a thorough understanding of the requirements. The time constraints added pressure to the team, making it challenging to delve deeply into each aspect of the project and formulate a comprehensive strategy.

To mitigate this challenge, we adopted a proactive approach by scheduling numerous team meetings dedicated to discussing the project in detail. These meetings served as invaluable platforms for sharing insights, clarifying doubts, and collectively brainstorming solutions. Additionally, we made frequent use of available resources, such as reaching out to the Teaching Assistant (TA) for clarification on ambiguous points and seeking guidance whenever needed. By leveraging these resources and fostering open communication within the team, we were able to overcome the hurdle of limited time and gain a better understanding of the project requirements.

3 - Difficulty in envisioning the website's functionality

Another challenge we encountered was the difficulty in envisioning the website's functionality, particularly in the nascent stages of the project. With only preliminary requirements and a rudimentary understanding of the project scope, it was challenging to visualize the website's intricate functionalities and user experience. This lack of clarity posed a significant obstacle, as it hindered our ability to conceptualize and plan for the development of the website effectively.

To address this challenge, we organized brainstorming sessions to generate ideas and perspectives on desired features and user interactions. Additionally, we utilized wireframing and prototyping tools to create visual representations of the website's functionalities and interfaces. These activities enabled us to overcome the initial difficulty in envisioning the website's functionality and gain a clearer understanding of the project's direction.

Through iterative refinement and continuous feedback, we laid the groundwork for developing a cohesive and user-friendly website.

What went right

1 - Meetings and Communication

One aspect that worked exceptionally well within our team was the ease of setting up meetings and fostering effective communication. We found that scheduling meetings was straightforward, and the majority, if not all, of our teammates consistently attended. This ensured that important discussions and decision-making processes involved active participation from all team members, promoting collaboration and synergy.

Furthermore, our team excelled in maintaining open and transparent communication channels. Whether through regular meetings, or instant messaging platforms, we found it easy to share updates, discuss ideas, and address any concerns promptly. This seamless communication facilitated smooth coordination among team members, allowing us to stay aligned with project goals and make informed decisions collectively. Overall, the ease of setting up meetings and effective communication played a pivotal role in enhancing team cohesion and productivity throughout the project.

2 - Splitting the workload

An advantageous aspect of our teamwork was our commitment to evenly distributing the workload among team members. Recognizing the diverse commitments and responsibilities of each member, we made a concerted effort to allocate tasks in a balanced manner. This allowed every team member the flexibility to work on their assigned parts at their own pace and convenience, accommodating their schedules and other academic obligations effectively.

By ensuring an equitable distribution of tasks, we fostered an environment where each team member could contribute meaningfully while managing their time efficiently. This approach not only promoted autonomy and ownership over individual responsibilities but also encouraged a sense of accountability within the team. As a result, team members were able to focus on their assigned tasks without feeling overwhelmed, thus maximizing productivity and facilitating a smoother workflow throughout the project duration.

3 - Setting short deadlines

One commendable practice within our team was our strategic approach to setting deadlines for individual tasks. Recognizing the importance of timely progress, we implemented a system where shorter deadlines were assigned to tasks requiring early completion. By setting deadlines a week or so prior to the end of the sprint for critical tasks, we ensured ample time for review, iteration, and addressing any unforeseen challenges. This proactive approach not only fostered a sense of urgency but also allowed us to stay on track and maintain momentum throughout the sprint.

This method of setting short deadlines proved highly effective in prioritizing tasks and allocating resources efficiently. It enabled us to focus our efforts on completing essential components of the project early on, laying a solid foundation for subsequent tasks. Moreover, by breaking down the project into manageable milestones

with clearly defined deadlines, we maintained a steady pace of progress and minimized the risk of last-minute rush or delays. Overall, this approach contributed to our team's productivity and success in achieving key objectives within the designated time frame.

Conclusion

As we conclude this phase of the project, it's crucial to acknowledge the valuable lessons learned and insights gained along the way. From navigating uncertainties to coordinating team efforts, each challenge has been met with resilience and effective problem-solving. By embracing challenges, leveraging strengths in communication and teamwork, and continuously refining processes, we've made significant progress towards our goals. Moving forward, it's essential to capitalize on successes, address any remaining weaknesses, and maintain our commitment to excellence. With a solid foundation laid, we're poised to tackle future endeavors with confidence and deliver a product that exceeds expectations.

Sprint 2 Release Plan

In Sprint 2 of the Condo Management System, we are set to significantly enhance the platform's capabilities across several key areas. There are a total of 15 user stories to be implemented in Sprint 2 of our project. These are user stories ranging from User Story #2 to #16.

The sprint's primary objectives include the implementation of secure registration processes for condo owners and rental users, enabling them to join the system using a unique registration key with robust validation and secure authentication mechanisms. A comprehensive dashboard will be developed for condo owners, providing them with detailed insights into property and financial information to facilitate efficient property management. Additionally, the property management functionality will be expanded to allow for the creation of detailed property profiles, uploading essential documents, and managing specific details for units, parking spots, and lockers. A financial management system will also be established, incorporating features for entering condo fees, recording budgets and costs, and generating annual reports to ensure transparent financial operations.

Furthermore, a reservation system for common facilities will be introduced, offering residents a user-friendly interface for booking amenities, thereby ensuring equitable access based on a first-come-first-serve principle. These enhancements aim to improve user experience significantly, streamline property management tasks, and lay a solid foundation for the system's ongoing development and scalability.

Release Plan Legend (on the Excel sheet):

User Story ID	User Story Points (USP)	Priority	Status
User Stories are from # 2 to #16. 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.	• High • Medium • Low	• TODO • DONE • PUSHED TO SPRINT 2 • REMOVED

There are a total of 178 story points.

Planning for future deployment in the Cloud/on devices:

For the future deployment of the condo management system, both in the cloud, we would adopt a comprehensive, phased approach that ensures scalability, security, and seamless user experience across platforms. Initially, we'll select a cloud provider that offers robust services aligned with our needs, such as AWS, Google Cloud, or Azure, taking advantage of their scalability, managed services, and global infrastructure. The deployment process would be automated using CI/CD pipelines, facilitating continuous integration and deployment through a tool like Jenkins, Travis CI, or GitHub Actions. This automation will include stages for code compilation, testing, and deployment to staging environments, followed by production deployment after approval, ensuring that every update is thoroughly tested and vetted.

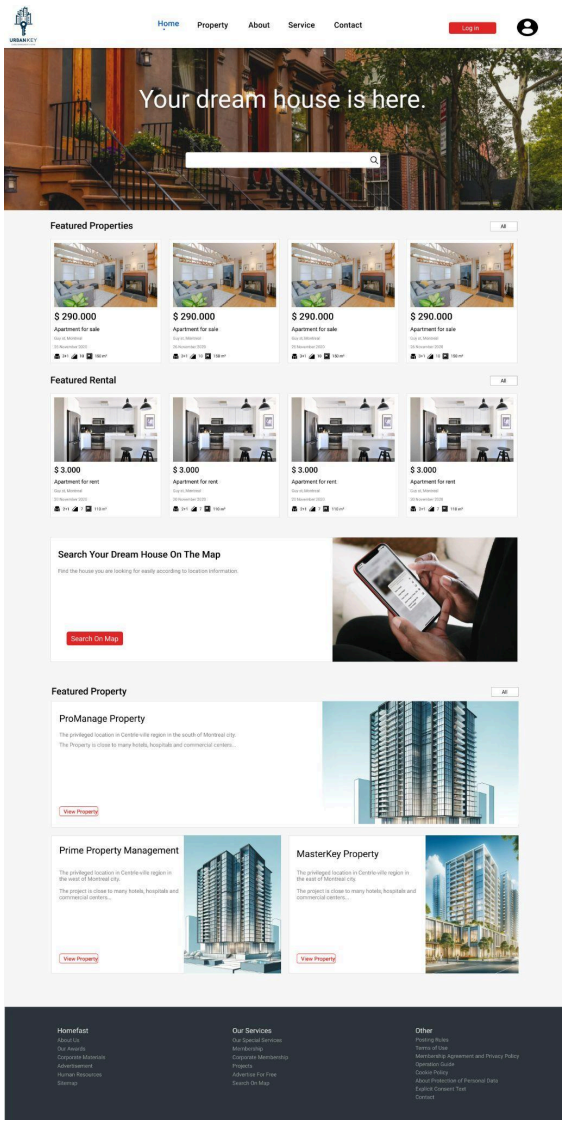
Additionally, we'll implement monitoring and logging solutions to track the system's performance and user interactions in real time, enabling us to quickly identify and resolve any issues post-deployment. The deployment strategy will also incorporate feedback mechanisms to gather user insights, further refining and optimizing the application to meet the evolving needs of our users.

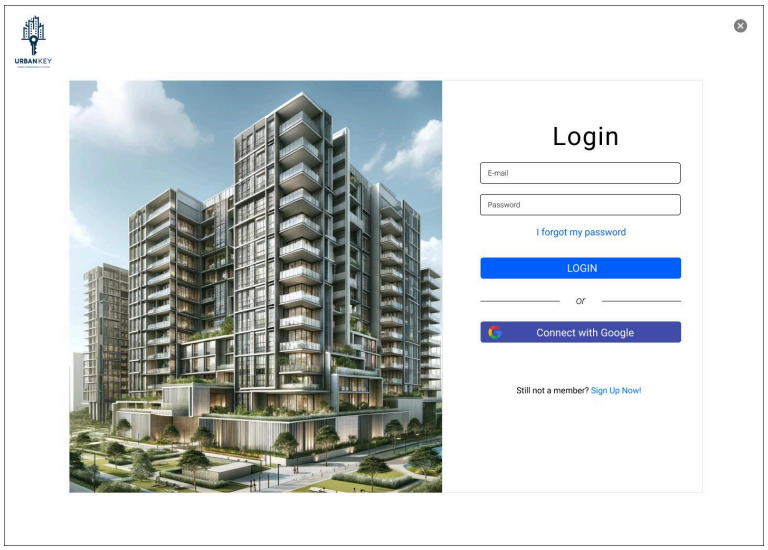
Release Plan Excel File

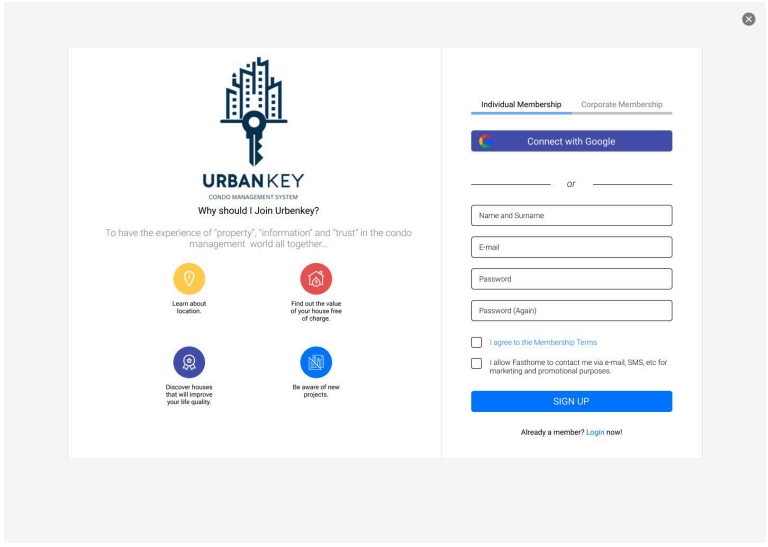
Release Plan.xlsx

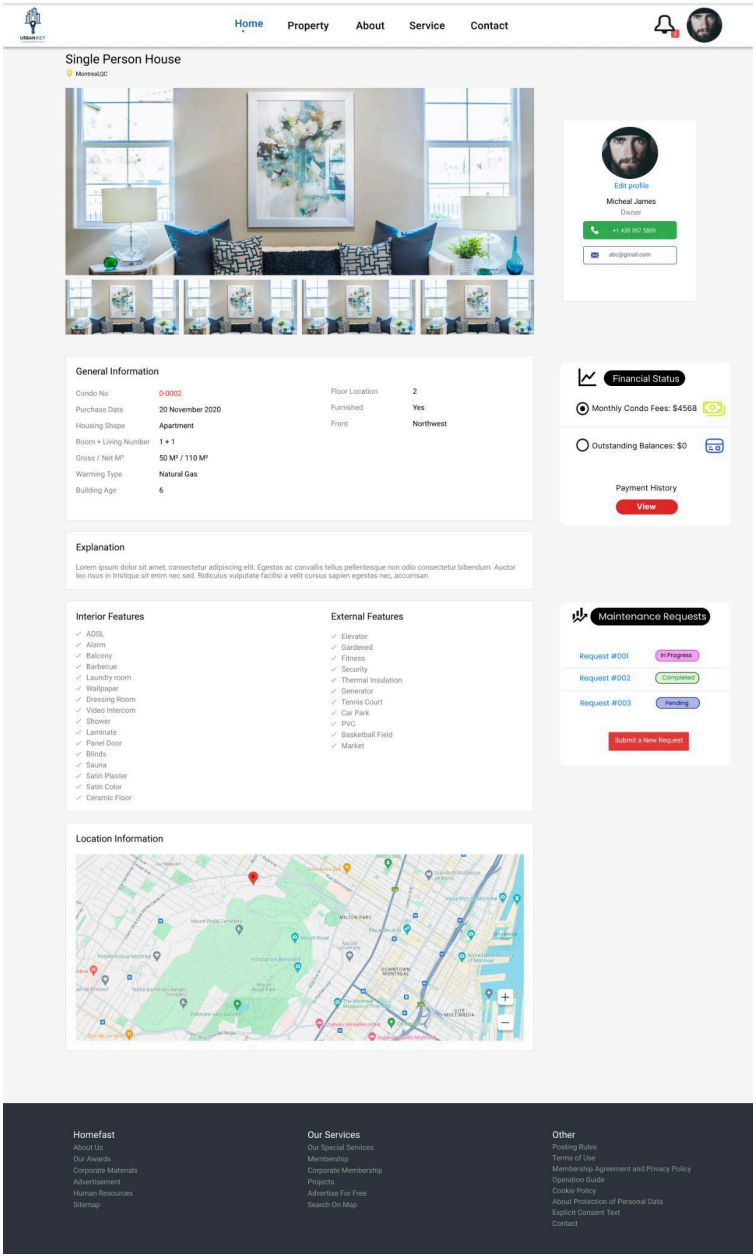
UI Prototypes

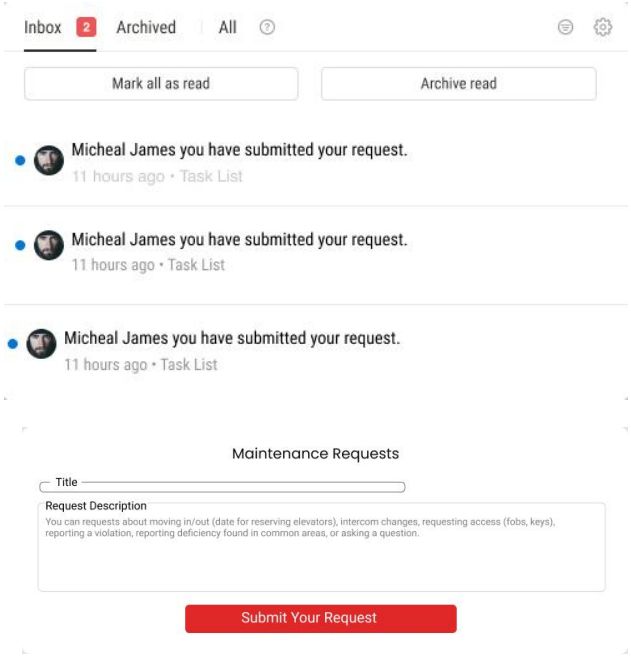
Figma Prototype: [Figma Link]









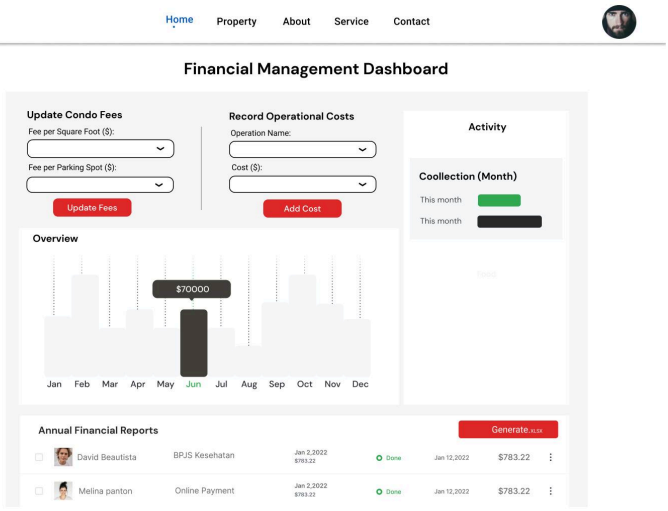


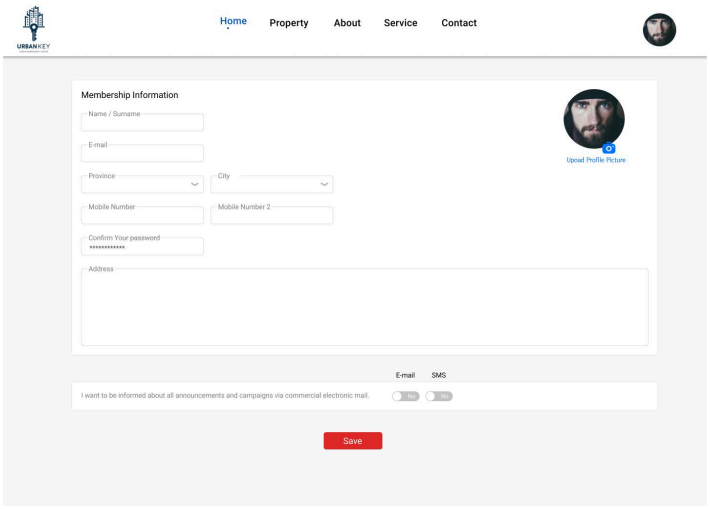
Payment History

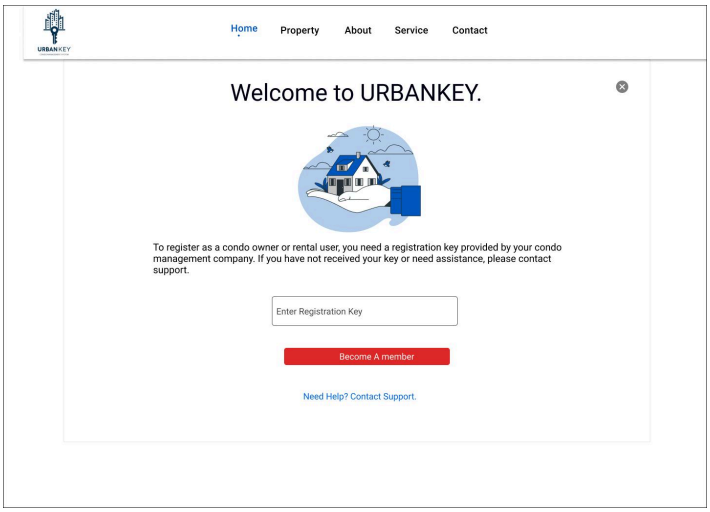
Payment Date	Amount	Payment Method	Reference Number	Status
September 2023	$4568.00	Bank Transfer	12345	Paid
October 2023	$4568.00	Credit Card	65078	Pending
November 2023	$4568.00	Debit Card	78965	Rejected

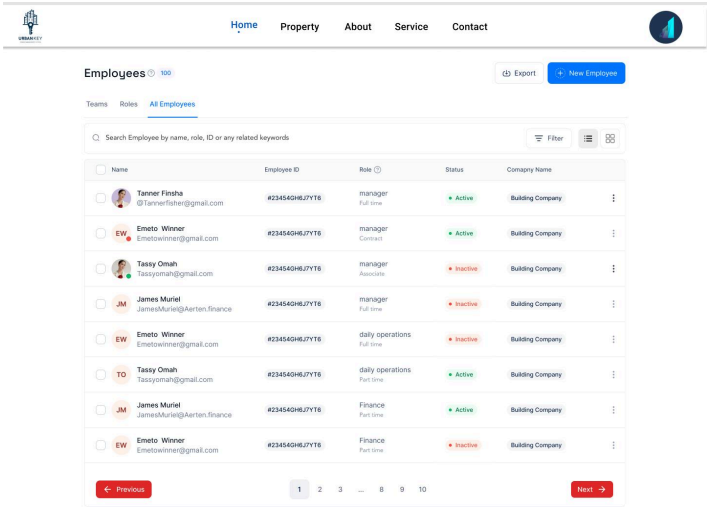
Go Back

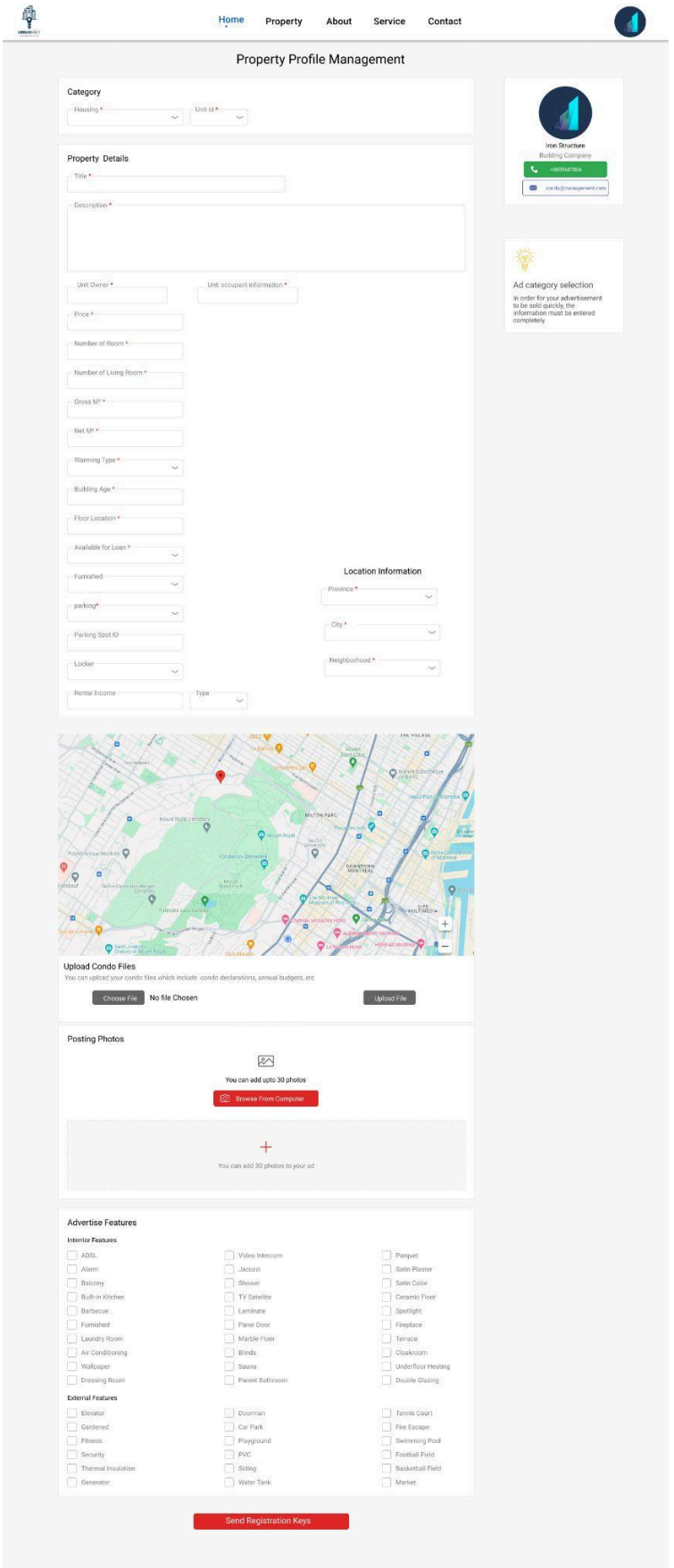
See Invoices

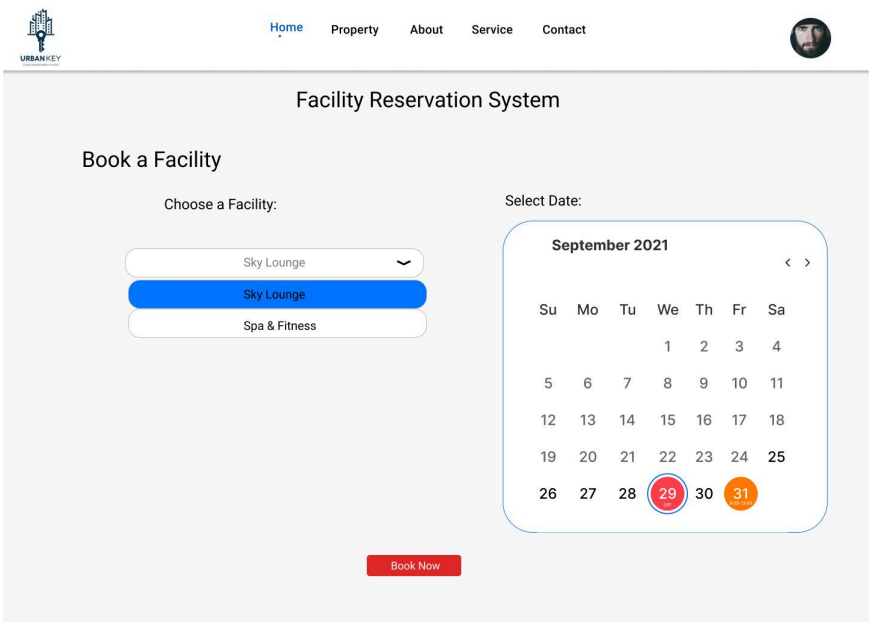


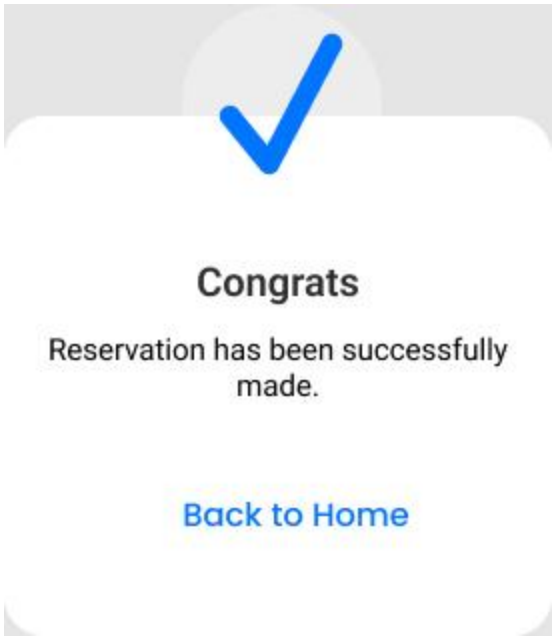

Home
Property
About
Service
Contact
Membership Information
Name / Surname
E-mail
Province
City
Mobile Number
Mobile Number 2
Confirm Your password
Address
Upload Profile Picture
E-mail
SMS
I want to be informed about all announcements and campaigns via commercial electronic mail.
Save


Home
Property
About
Service
Contact
Welcome to URBANKEY.
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.
Enter Registration Key
Become A member
Need Help? Contact Support.


Home
Property
About
Service
Contact
Employees
Export
New Employee
Teams
Roles
All Employees
Search Employee by name, role, ID or any related keywords
Filter
Name
Employee ID
Role
Status
Comapny Name
Previous
Next


Home
Property
About
Service
Contact
Property Profile Management
Category
Housing *
Unit Id *
Property Details
Title *
Description *
Unit Owner *
Unit occupant information *
Price *
Number of Room *
Number of Living Room *
Gross M² *
Net M² *
Warming Type *
Building Age *
Floor Location *
Available for Loan *
Furnished
parking*
Parking Spot ID
Locker
Rental Income
Type
Location Information
Province *
City *
Neighborhood *
Iron Structure
Building Company
Ad category selection
In order for your advertisement to be sold quickly, the information must be entered completely.
Upload Condo Files
You can upload your condo files which include condo declarations, annual budgets, etc
Choose File
No file Chosen
Upload File
Posting Photos
You can add upto 30 photos
Browse From Computer
You can add 30 photos to your ad
Advertise Features
Interior Features
ADSL
Alarm
Balcony
Built-in Kitchen
Barbecue
Furnished
Laundry Room
Air Conditioning
Wallpaper
Dressing Room
Video Intercom
Jacuzzi
Shower
TV Satellite
Laminate
Panel Door
Marble Floor
Blinds
Sauna
Parent Bathroom
Parquet
Satin Plaster
Satin Color
Ceramic Floor
Spotlight
Fireplace
Terrace
Cloakroom
Underfloor Heating
Double Glazing
External Features
Elevator
Gardened
Fitness
Security
Thermal Insulation
Generator
Doorman
Car Park
Playground
PVC
Siding
Water Tank
Tennis Court
Fire Escape
Swimming Pool
Football Field
Basketball Field
Market
Send Registration Keys


URBAN KEY
Home
Property
About
Service
Contact
Facility Reservation System
Book a Facility
Choose a Facility:
Sky Lounge
Sky Lounge
Spa & Fitness
Select Date:
September 2021
Su Mo Tu We Th Fr Sa
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
Book Now


Congrats
Reservation has been successfully made.
Back to Home

Sprint 2 Testing Plan

This project documentation refers to the outlining of a comprehensive strategy for verifying and validating the software developed during the second sprint of the Condo Management System project. In order to meet the specific 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.

Testing Tool Selection:

Firstly, it is essential to decide on the tools that will be used at each level of testing.

As the stack of our Condo Management Systems project, we selected to use React for the front-end of the website application, alongside with HTML/CSS/ JavaScript. Moreover, we will be using Python for the back-end and MongoDB as our NoSQL database. The API for the back-end is still to be determined.

Unit Testing: It is crucial to use a tool that supports isolation and mocking capabilities. It will allow for testing individual methods or components in isolation from the rest of the system.

For the front-end of the application, Jest is a popular testing framework that works well with React and vanilla JavaScript. Jest is widely used for its simplicity and support for React applications. It can handle testing of JavaScript code, including asynchronous code, and integrates well with Babel.

For the Python back-end, pytest supports fixtures, which can be used to manage test dependencies, and it's known for its ease of use and ability to handle asynchronous code. There is also unittest, which is a unit testing framework that comes with Python's standard library. It's inspired by JUnit and has a similar feature set, including test organization, test fixtures, and test runners.

Integration Testing: In order to effectively test interactions between the Python backend, MongoDB database, and any external APIs that will be used, integration tools will be employed.

- Postman is an ideal tool for testing API endpoints, which will ensure that the backend correctly handles requests and responses. It can be used to automate tests for the RESTful services and to verify the integration between the frontend and backend.

- PyTest with pytest-mongodb and requests will be a testing framework for Python. With the pytest-mongodb plugin, we can mock or spin up a MongoDB instance for testing purposes. The requests library or aiohttp (for asynchronous applications) can be used for testing API calls within Python tests, simulating integration between the backend and external services.

- MongoDB Memory Server is a Node.js package that spins up a real MongoDB server programmatically, in memory, for testing purposes. It is useful when for running tests that interact with the database

without affecting the production or development databases.

System Testing: For system testing, where the goal is to validate the integrated system's functionality, performance, and behavior under production-like circumstances, there are several tools that can be effective.

- Selenium is a widely used tool for automating web browsers. It can simulate user interactions with the React application, testing the system as a whole from the user's perspective. It can work well for testing complex user flows and interactions across the application.

- Cypress is an all-in-one testing framework that's gained popularity for its ease of use and support for modern web development frameworks, including React. Cypress can be used for writing end-to-end tests, simulating real user interactions with the application. It provides a modern and developer-friendly alternative to Selenium, with features such as automatic waiting, real-time reloads, and video recording of test runs.

Automation: For Continuous Integration (CI), a service like Jenkins, Travis CI, or GitHub Actions to automate the execution of tests upon every commit or merge request will be employed. This will ensure that tests are run frequently and consistently.

- Jenkins is an open-source automation server that provides hundreds of plugins to support building, deploying, and automating any project. It is highly customizable for complex workflows, supports a wide range of plugins for integration with various development tools, and can be used for both CI and Continuous Deployment (CD).

- GitHub Actions is an automation tool that enables you to automate your build, test, and deployment pipeline right within your GitHub repository. It can be directly integrated with GitHub repositories, supports CI/CD and much more, including issue labeling, releasing software, and automated workflows based on GitHub events.

When choosing tools, we are considering the team's familiarity with the technologies, the complexity of setup and maintenance

Testing Approach:

Afterwards, the testing approach should be identified.

The team will focus on writing a high volume of unit tests due to their speed and efficiency. Mock dependencies to test individual functions or classes in isolation. Ensure you cover positive cases, negative cases, and edge cases. Moreover, as part of the integration tests, we will write tests for critical interactions between components, such as database access layers (DAOs) and external services. We will ensure that these components work together as expected under different scenarios. Additionally, in terms of system testing, we will conduct tests on the entire application to verify that it meets the requirements specified. This will include testing the interaction between all components and external systems in a production-like environment. Lastly, we will supplement

automated tests with manual testing, especially for UI/UX aspects, accessibility, and other areas difficult to cover with automated tests. We might use UI automation tools like Selenium or tools for automated accessibility testing to reduce the effort.

Metrics and Coverage:

For Sprint 2, our team will set, once again, a target for code coverage, aiming for at least 80% with a focus on critical paths in the application. We will use coverage tools which are integrated with our testing frameworks to measure and report coverage.

Acceptance Tests:

For each user story in Sprint 2, an acceptance criteria should be developed, as well as automated and/or manual tests need to be created to verify that these criteria are met. Our acceptance tests will validate that the system performs as expected from an end-user perspective. For our website, acceptance tests will cover a wide range of scenarios, including user registration, property and financial management functionalities, and the use of common facilities, to simulate real-world use effectively. The emphasis will be placed on automating these tests to streamline the process, employing tools and frameworks that facilitate Behavior-Driven Development (BDD) for clearer, more understandable test scenarios. We might use BDD tools like Cucumber or SpecFlow to write acceptance tests in a language that is understandable.

Implementation Plan:

We will integrate testing into the development workflow by setting up a CI pipeline 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.