



SOEN390

-

## Compiled Report for EstateFlow

Team 12:

Sarah Abellard - 40184667  
Andy Sun - 40192040  
Barthan Thirunavukarasu - 40098158  
Abisan Uthamacumaran - 40078084  
Yousef Enein - 40115494  
Ziyi Wang - 40167122  
Fatoumata Binta Barry - 40213443  
Samuel Collette - 40175048  
Arshpreet Singh - 40172137  
Jeremie Garzon - 40062316

Github:

<https://github.com/Irisvella/SOEN-390-W2024>

Website:

<https://estate-flow.vercel.app/>

# Contents

<b>Contents</b>	<b>2</b>
<b>1. Product Vision</b>	<b>4</b>
1. Introduction	4
1.1. References	4
View appendix A	4
2. Positioning	4
1.1. Problem Statement	4
1.2. Product Position Statement	5
3. Stakeholder and User Descriptions	6
3.1. Stakeholder Summary	6
3.2. User Summary	8
3.3. User Environment	9
3.4. Key Stakeholder or User Needs	10
3.5. Alternatives and Competition	12
4. Product Overview	13
4.1. Product Perspective	13
4.2. Assumptions and Dependencies	13
5. Product Features	14
6. Other Product Requirements	14
<b>2. Software Architecture</b>	<b>16</b>
1. Domain Model	16
2. Component Diagram	17
3. Use Case Diagrams	18
4. Class Diagram	24
5. Deployment Diagram	25
<b>3. Requirements and User Story Backlog</b>	<b>26</b>
<b>4. Risk Assessment and Management</b>	<b>36</b>
<b>5. Web Hosting</b>	<b>42</b>
<b>6. Code Management</b>	<b>43</b>
7.1 Sprint 2 Release Plan	53
7.2 Sprint 3 Release Plan	56
7.3 Sprint 4 Release Plan	70
7.4 Sprint 5 Release Plan	72
<b>8 Testing Plans</b>	<b>75</b>
8.1 Sprint 2 Testing Plan	75
8.2 Sprint 3 Testing Plan	76
8.3 Sprint 4 Testing Plan	77
8.4 Sprint 5 Testing Plan	80
<b>9. Sprint Retrospectives</b>	<b>94</b>

9.1 Sprint 1	94
9.2 Sprint 2	95
9.3 Sprint 3	96
9.4 Sprint 4	98
9.5 Sprint 5	100
<b>10 UI Prototypes</b>	<b>102</b>
10.1 Sprint 2 UI Prototypes	102
10.2 Sprint 3 UI Prototypes	102
10.3 Sprint 4 UI Prototypes	102
10.4 Sprint 5 UI Prototypes	116
<b>APPENDIX A: Sources</b>	<b>124</b>

# 1. Product Vision

## 1. Introduction

This document serves to collect, analyze and define all the high level needs and features for our application EstateFlow. We are interested in exploring what our stakeholders and users need, as well as how we can best serve them. We will be focusing on how EstateFlow attempts to meet and exceed expectations, and why a user should choose it over competitors.

### 1.1. References

View [appendix A](#)

## 2. Positioning

### 1.1. Problem Statement

The problem of	Inefficient condominium management and lack of centralized access to property-related information
affects	Condo residents, owners, property managers, and service staff
the impact of which is	Time-consuming property management, difficulty in tracking costs and fees, challenges in booking amenities, and inadequate communication within the condo community
a successful solution would be	A complete, user-friendly software that provides a centralized dashboard for property overview, financial administration, reservation systems, and improved transparency, efficiency, and community interaction for all users.

## 1.2. Product Position Statement

For	Condo residents, owners, and management companies seeking an integrated property management solution
Who	need a efficient, and user-friendly system for managing their properties, its logistics and communication
The (EstateFlow)	is a property and client management system
That	facilitates the process of managing properties. It offers an easy profile creation process, as well as simplifies property and financial management for users.
Unlike	conventional, disjointed property management tools that lack comprehensive features and cross-platform compatibility.
Our product	stands out due to its intuitive approach and user-friendly experience in property management; setting it apart from other products in the market.

### 3. Stakeholder and User Descriptions

#### 3.1. Stakeholder Summary

Name	Description	Responsibilities
Software Developers	Professionals responsible for coding and programming	<ul style="list-style-type: none"><li>• Implementing the specified features and functionality accurately and efficiently.</li><li>• Collaborating with other team members to ensure seamless integration of software components.</li><li>• Ensuring code quality, maintainability, and adherence to coding standards.</li></ul>
Project Manager	Oversees and coordinates the project	<ul style="list-style-type: none"><li>• Developing project plans, timelines, and resource allocation to ensure successful project completion.</li><li>• Communicating with stakeholders, including developers and clients, to ensure alignment with project goals and expectations.</li><li>• Monitoring progress, tracking deliverables, and making adjustments as</li></ul>

		needed to meet project milestones.
Legal Team	Professionals responsible for legal aspects of the project	<ul style="list-style-type: none"> <li>Mitigating legal risks and providing guidance on legal matters throughout the development process.</li> </ul>
Investors/Shareholders	Individuals or entities providing financial support for the project	<ul style="list-style-type: none"> <li>Ensuring the project aligns with strategic goals and has the potential for a positive return on investment.</li> <li>Monitoring financial aspects, including budgeting and resource allocation.</li> <li>Reviewing and approving funding for different phases of the project.</li> </ul>
Client/users	Users of the application on a daily basis to meet their condo management needs.	<ul style="list-style-type: none"> <li>Interacting with the system to complete payments, reservation, etc...</li> <li>Adding condo information and pricing information as an owner.</li> <li>Submitting and resolving tickets.</li> </ul>

### 3.2. User Summary

Name	Description	Responsibilities
Condo Owners	Individuals who own a condo within a managed property.	<ul style="list-style-type: none"> <li>Access a dashboard with property information, personal profile, financial status, and request status.</li> <li>Link their condo unit with profiles using registration keys.</li> <li>Submit requests for various purposes.</li> </ul>
Renters	Individuals renting a condo within a managed property.	<ul style="list-style-type: none"> <li>Similar to condo owners in terms of profile creation, access to property information, and submission of requests.</li> </ul>
Condo Management Companies	Organizations responsible for managing and overseeing condo properties and rentals	<ul style="list-style-type: none"> <li>Create property profiles with details such as property name, unit count, parking count, locker count, and address.</li> <li>Upload and manage condo files for each property.</li> <li>Enter detailed information for condo units, parking spots, and lockers.</li> <li>Set up roles for employees (e.g., manager, finance) responsible for property operations.</li> <li>Set up and manage the financial system, including condo fees and operational costs.</li> <li>Set up the reservation system for common facilities.</li> <li>Manage employee roles and responsibilities.</li> </ul>
Employees (ex: Manager, Finance)	Individuals within condo management companies with specific roles.	<ul style="list-style-type: none"> <li>Responsible for daily operations or finance tasks within the managed property.</li> </ul>
Testers	Individuals responsible for testing the Condo Management System.	<ul style="list-style-type: none"> <li>Execute test cases to validate system functionality.</li> <li>Identify and report defects or issues.</li> </ul>

		<ul style="list-style-type: none"> <li>• Collaborate with developers and other stakeholders to ensure quality standards.</li> <li>• Verify that the system meets specified requirements.</li> <li>• Conduct various types of testing, including functional, performance, and usability testing.</li> </ul>
--	--	--

### 3.3. User Environment

#### **Number of People Involved:**

- Condo Owners: Each condo owner interacts individually with the system.
- Rental Users: Similar to condo owners, rental users interact individually.
- Condo Management Companies: Multiple employees within the management company, including roles such as managers and finance personnel.

#### **Task Cycle Duration:**

- Creating a User Profile: Relatively short task cycle: it should only take a few minutes.
- Accessing the Dashboard: Ongoing and frequent, depending on user activity.
- Submitting Requests: Varied, but typically short task cycles for specific activities.
- Managing Property Profiles (Condo Management Companies): Ongoing task cycle for property management.
- Financial System Management (Condo Management Companies): Continuous and ongoing.
- Reservation System Usage: Short task cycles when reserving common facilities.

#### **Time Spent in Each Activity:**

- Creating a User Profile: 5-10 minutes.
- Accessing the Dashboard: Varies based on user engagement.
- Submitting Requests: Varies by the type of request but typically a few minutes.
- Managing Property Profiles (Condo Management Companies): Ongoing and time-consuming, depending on updates and property changes.
- Financial System Management (Condo Management Companies): Ongoing and can be time-intensive during financial reporting periods.
- Reservation System Usage: A few minutes for each reservation activity.

### **Environmental Constraints:**

- Users may access the system from various locations, including home, work, or on-the-go, hence why developing the mobile version is relevant.
- Mobile access is essential for convenience.

### **System Platforms:**

- The app and website should be accessible on Android, iOS, Linux, MacOS, and Windows.

### **Integration with Other Applications:**

- The system may need to integrate with email services for notifications.
- Integration with financial tools or platforms for seamless financial management.
- Forum and event features may involve integration with communication or calendar applications.

### **3.4. Key Stakeholder or User Needs**

#	Needs	Priority	Concerns	Current Solution	Proposed Solutions
1	User Interface Design	Medium	Difficult to navigate and use. Underutilization of features.	Generic, non-customized interfaces.	Intuitive, user-friendly design. Customizable dashboard for different user roles. Modular features that can be enabled or disabled as needed.
2	Reservation System for Amenities	Medium	Inefficient booking process.	Manual booking through phone or email.	Online reservation system with a calendar interface.

			Lack of availability visibility.		Real-time availability updates. Automated reservation confirmations and reminders.
3	Broadcast messages	High	Inefficient communication. Low engagement. Lack of personalization.	Email blasts. Physical bulletin boards.	In-app real-time notification system. User preferences for personalized messages. - Integration with popular messaging platforms
4	Efficient Issue Resolution	High	Slow response to resident requests and issues. Miscommunication and misassignment of tasks.	Manual sorting and assignment of issues.	Automated ticketing system that routes requests to the appropriate staff. Tracking and status updates for issue resolution.
5	Mobile Accessibility	High	Limited or no mobile support.	Desktop-only access or poorly designed mobile sites.	Fully responsive design or dedicated mobile app. Cross-platform compatibility for Android and iOS.
6	Integration with Other Systems	High	Disparate systems for different tasks. Manual data entry leads to errors.	Standalone systems with manual data syncing.	Seamless integration with accounting, booking, and other relevant systems. Automated data transfers and updates between systems.

7	Community Engagement	Low	Lack of interaction between residents. Difficult to share information and events.	Physical notice boards. Informal social media groups.	Built-in forum for discussion and information sharing. Event organization and RSVP features. Virtual bulletin board for announcements and community news.
---	----------------------	-----	--	--	---

### 3.5. Alternatives and Competition

When building a new product, market research is a crucial step, as it enables stakeholders to evaluate different options and find a gap in the market to disrupt a given industry. In our case, we researched other property management systems and evaluated their strengths and weaknesses. If the value proposition of our product is strong enough, building a homegrown solution will be the most effective option.

First, we examined BuildingLink, a popular property management website..

**Strengths:**

- Offers many management features such as maintenance requests, package tracking, and resident communication tools.
- It's established in the market (around 2 million users), which suggests that it is reliable and supports many users.
- There is also an app, which improves convenience for users.

**Weaknesses:**

- The platform can be costly, making it less accessible for smaller communities and small businesses.
- Its comprehensive feature set, while valuable, may overwhelm users seeking simpler solutions.

Then we examined *Entrata*:

**Strengths:**

- High level of security for the user's data
- It offers solutions for a multitude of different users such as, multifamily, student and commercial

**Weaknesses:**

- While versatile, it may lack some advanced customization options found in larger systems.
- Some users might find its pricing model less favorable, depending on the community's size and needs.

### **Strategic Positioning for EstateFlow:**

We want to prioritize a balance between comprehensive functionality and user-friendly design, aiming to offer an intuitive system that addresses the potential complexity and usability issues identified in competitors.

## **4. Product Overview**

[This section provides a high level view of the product capabilities, interfaces to other applications, and systems configurations. This section usually consists of two subsections, as follows:

### **4.1. Product Perspective**

EstateFlow serves as a centralized point in connecting unit owners, rental users, and management companies.. The online and mobile apps communicate with external systems. It integrates with financial software for fee management , document storage solutions for easy access to condo-related files and alerts systems in order to deliver a more optimal management experience. Therefore, it is not a stand-alone solution, but rather a component of a bigger system that optimizes condo administration. However , while offering extensive interoperability with the systems just mentioned , it is still intended to function independently

### **4.2. Assumptions and Dependencies**

A number of important variables can impact our Condo Management System: the ease of use and design of user interfaces; the security and privacy of user data; the effectiveness of data management in managing property and financial information; and the precision of financial computations. Additionally, support for multiple platforms and integration with third-party services like Single Sign On are functionalities that can be beneficial to users, because it will provide them with a smoother user experience.

Assumptions that could necessitate revisions in the Vision document include changes in operating system support or requirements, adjustments in the availability or policies of third-party services used for Single Sign On and language support, updates in privacy

laws or data protection regulations impacting user data handling, shifts in technology or development frameworks, and changes in user expectations or market trends. These factors highlight the project's need for flexibility to adapt its development strategy and feature set in response to evolving technical, legal, and market environments.

## 5. Product Features

1. User Account Creation: Users will be able to identify themselves by creating a profile that includes personal information and a photo. Additionally , they will also receive a registration key from their management company to become condo owners or rental users in the system.
2. Property Dashboard: Our system should provide Condo owners with access to a dashboard that displays property data, financial condition, and change requests.
3. Property and Profile Management: Condo management companies will be able to quickly develop and manage property profiles, which will include documentation that owners may view.
4. Unit and Facility Information: Our system will be able to track management data such as condo units, parking spaces, and lockers. This covers ownership and fee information.
5. Financial System Integration: Financial administration of condo fees, budgeting, and yearly reporting will be simplified
6. Reservation System: Owners and renters can reserve shared amenities using a calendar interface on a first-come, first-served basis.
7. Employee Role-Based Access: Different roles can be allocated to employees that manage the same property, focus on daily operations, or handle finances.
8. Request and Notification System: Allows users to submit and monitor requests while also receiving updates about the current actions.
9. Optional features include community forums, event organization, and user-specific discounts or offers.

## 6. Other Product Requirements

In developing the application, the team will strive to comply with standard coding and security practices, as well as data protection regulations. We will be aiming to achieve full compatibility with different browsers, and across IOS and Android.

In terms of quality ranges, for performance we are aiming for response times of less than a second and the ability to concurrently serve hundreds of users. We will also be focusing on guaranteeing a robust program through extensive testing, good fault tolerance with error handling and preventative measures, and an intuitive user interface to enhance usability, as well as the overall user experience.

The most notable external constraints would be firstly laws regarding data security and privacy which must be respected and planned for. Secondly, complying with common condo association guidelines to be as attractive as possible to potential management companies.

Our quality ranges and constraints are paramount, and must be constantly revisited and kept track of to ensure a safe, successful application. Achieving platform coverage is a secondary requirement in that regard.

## 2. Software Architecture

### 1. Domain Model

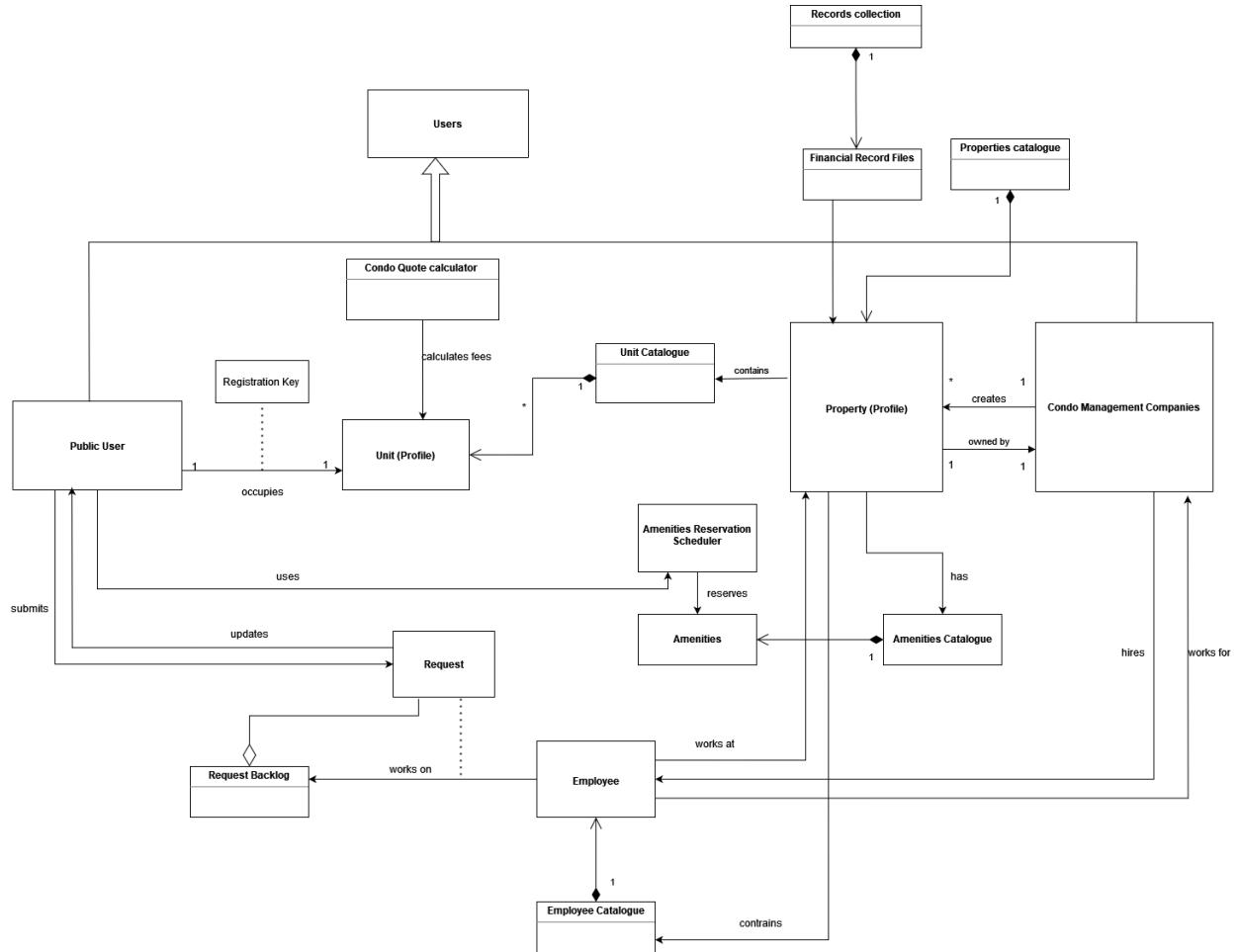


Figure 1: Domain Model of EstateFlow

This domain model represents our current system and its entities, as well as the flow of data between said entities. The domain model helps to provide a strong foundation upon which the team can develop the architecture envisioned. It helps clear up ambiguities and keeps the entire team on the same page regarding the structure of our system.

They help define relationships between entities. They also serve as the basis for discussion regarding design decisions.

Many of the requirements and/or features can be mapped in this model. For example, in the case of reserving a sauna room (reserving an amenity), a “Public User” will be interacting with the “Amenities Reservation Scheduler”, which in turn will reserve an

“Amenity” that can be found in the “Amenity Catalog”. All of our other user stories and required features can be mapped similarly on this diagram. In the case that a feature is not supported by this diagram, that will change in the near future during further sprints.

## 2. Component Diagram

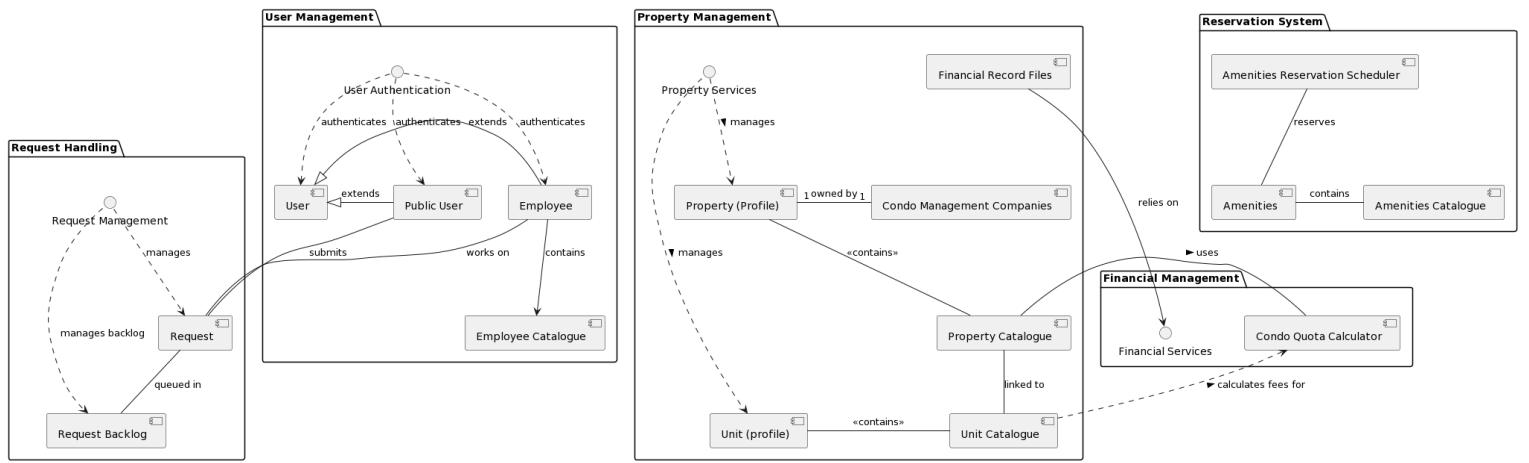


Figure 2: Component Diagram of EstateFlow

A component diagram represents a high level abstraction of the different components and working parts of our system. It is less specific in terms of components and how they are designed in comparison to the domain model, but it is far more interested in the interactions between components. Our system can be broken down into five major parts: “Request Handling”, “User Management”, “Property Management”, “Financial Management”, and “Reservation System”.

### 3. Use Case Diagrams

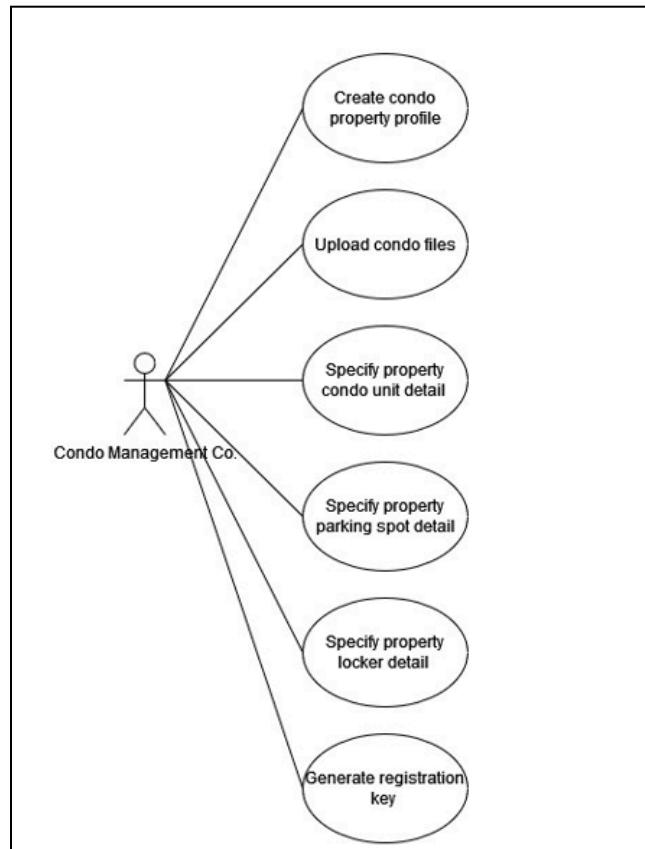


Figure 3: Use case diagram for a condo management company being able to create and manage profiles for their properties

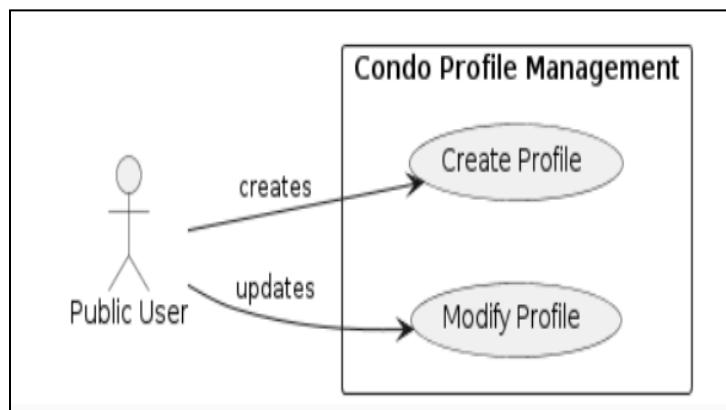


Figure 4: Use case diagram for a condo user being able to manage their profile

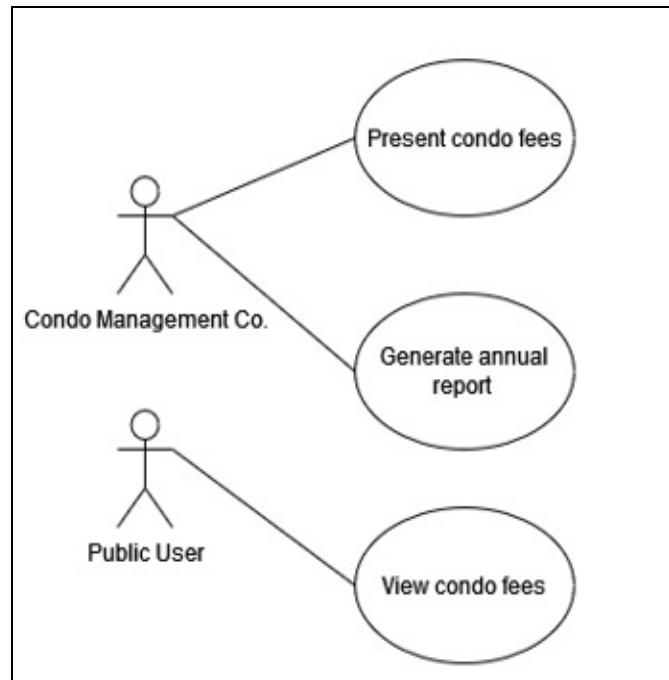


Figure 5: Use case diagram representing condo financial calculations

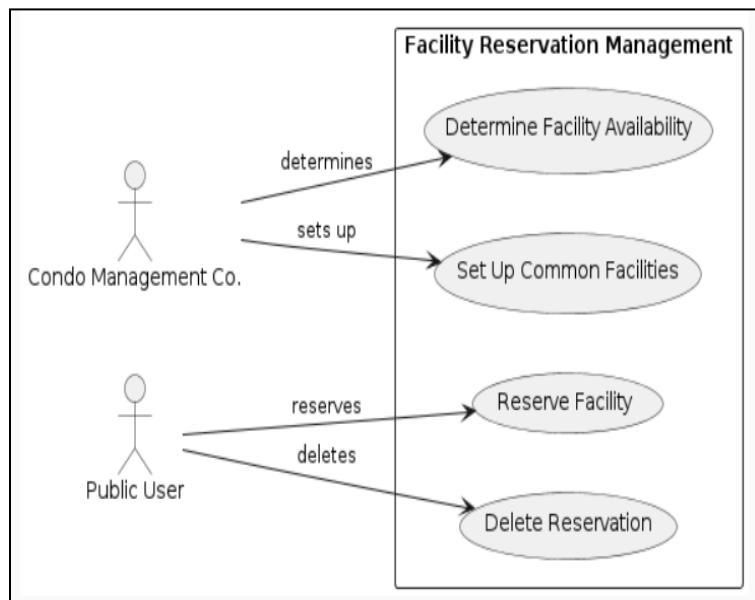


Figure 6: Use case diagram representing the common facilities reservation system

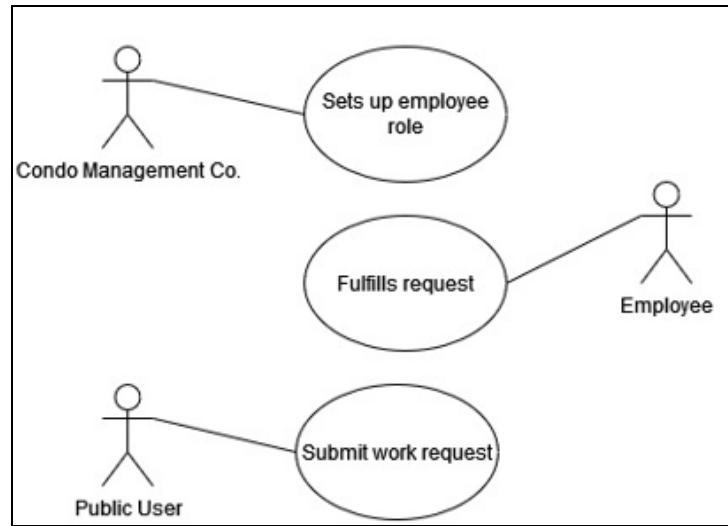


Figure 7: Use case diagram representing how work requests are treated

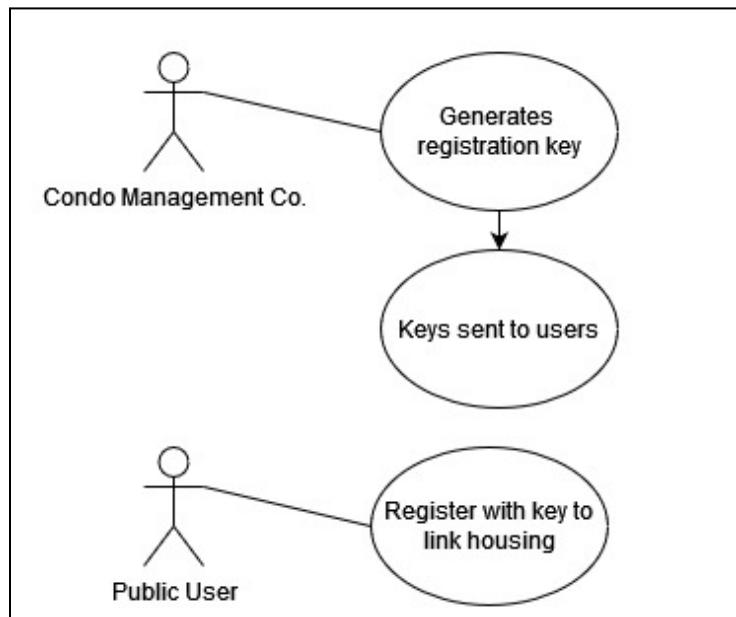


Figure 8: Use case diagram representing how registration keys can be generate and used

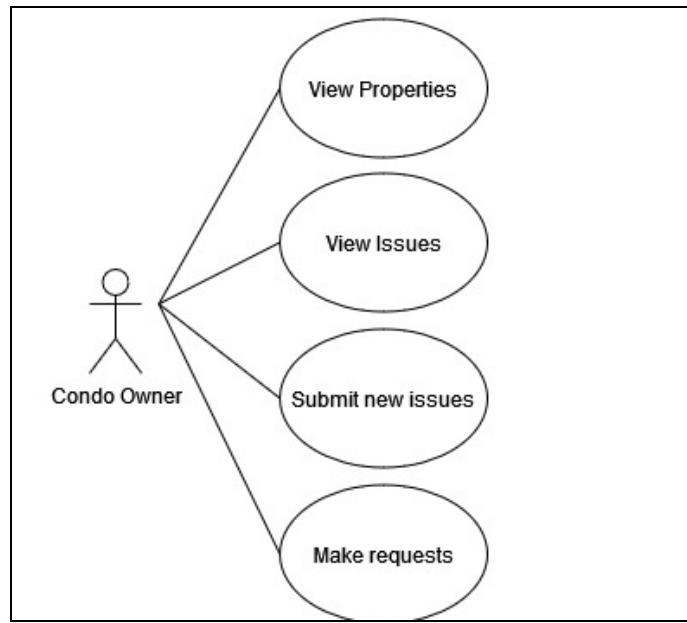


Figure 9: Use case diagram for a condo owner being able to access their features through their dashboard

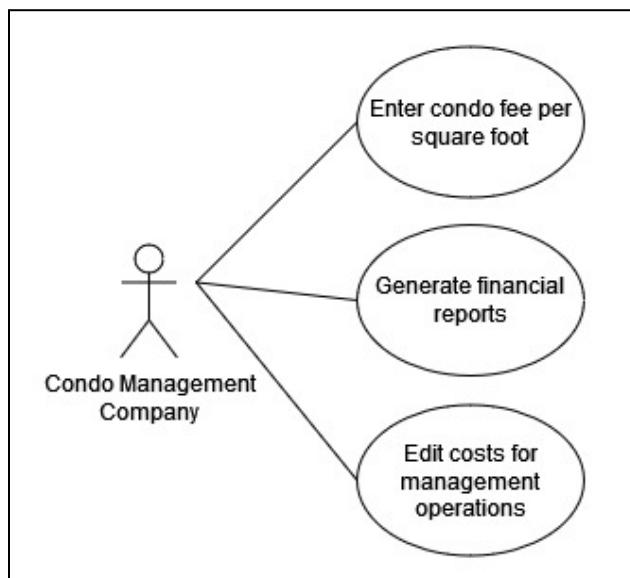


Figure 10: Use case diagram for a condo management company using their financial system features

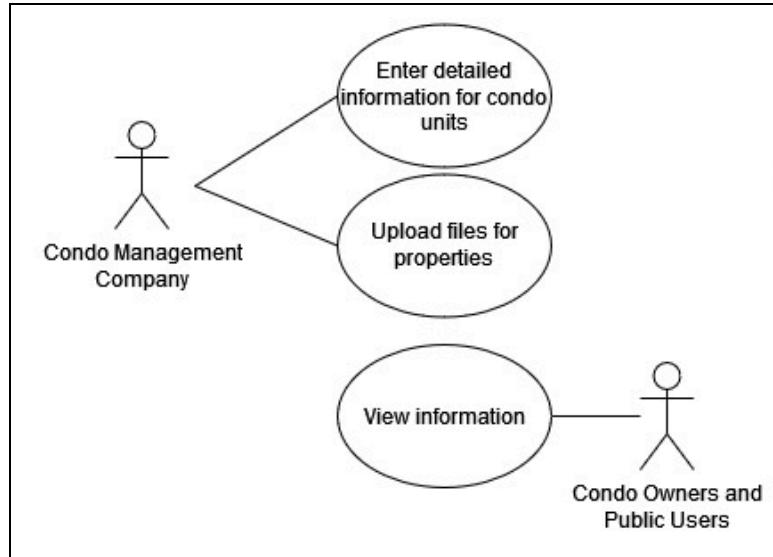


Figure 11: Use case diagram representing a condo management company uploading important information, to be viewed by other users

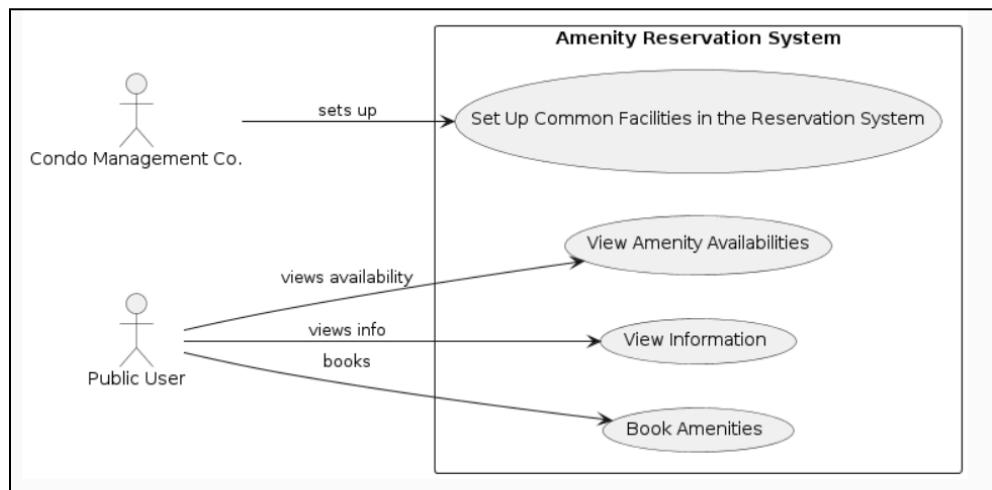


Figure 12: Use case diagram representing viewing, checking and booking amenities

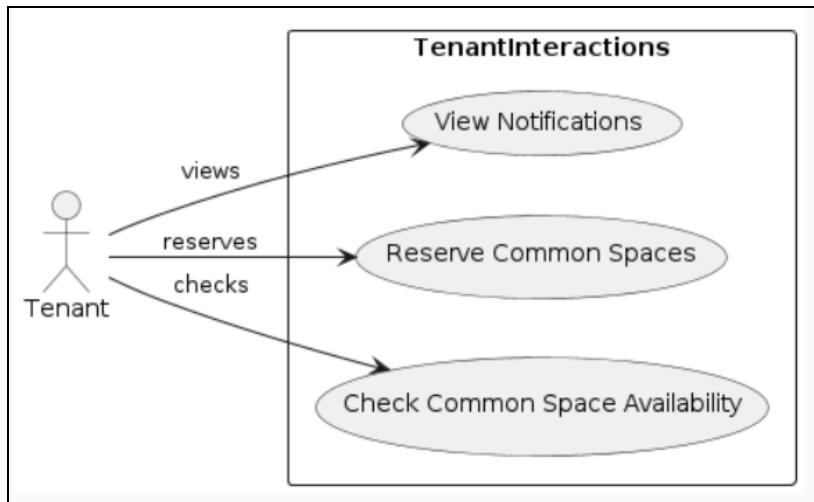


Figure 13: Use case for a tenant being able to see their relevant notifications

## 4. Class Diagram

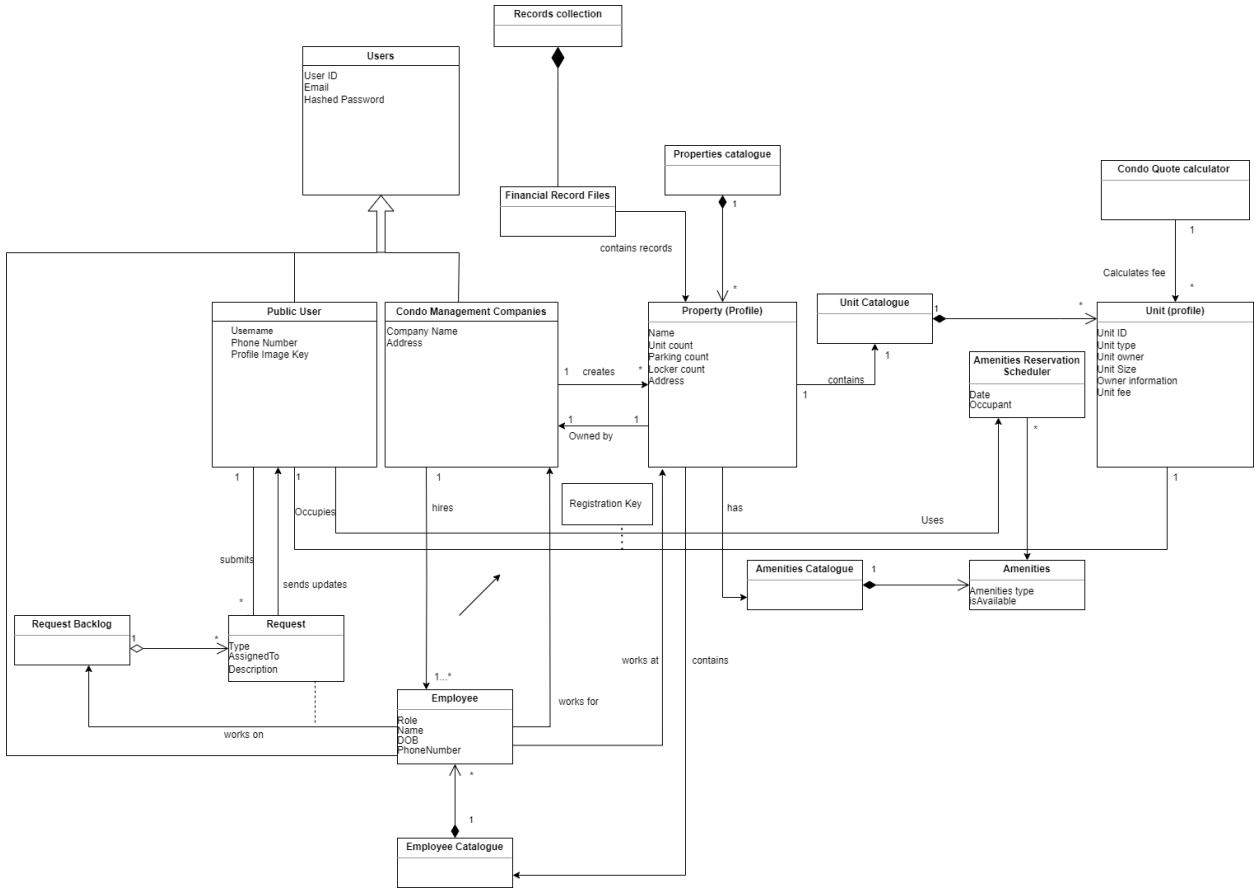


Figure 14: Class Diagram of EstateFlow

The class diagram breaks down our important classes along with their attributes and methods. This is also helpful for identifying associations and dependencies in our system.

Although it looks somewhat similar to the domain model, this is more focused on being a structural representation of our backend. It is a complex representation of all the different classes, along with their attributes and relevant methods. Due to the complexity and completeness of the model, it is quite useful as a structural blueprint for the implementation phase of the project, and thus must be built correctly early on in the planning phase.

## 5. Deployment Diagram

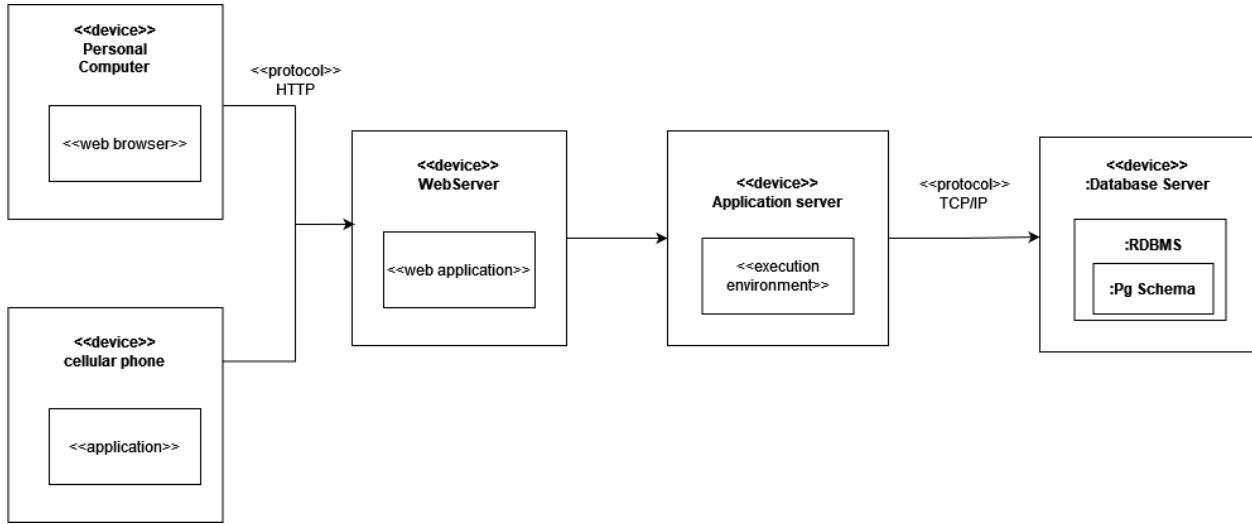


Figure 15: Deployment Diagram of EstateFlow

This deployment diagram represents the deployment structure of our system consisting of a web, application, and database server. Client devices are personal computers accessing the system through our web app, and cellular phones through our android app. The physical client devices communicate with the web server through HTTP requests. The web server acts as an entry point for the client, as well as handling tasks such as routing, and serving frontend elements to devices. The application server serves mainly as a connection to the database by receiving HTTP requests from the web server, interpreting them, and performing operations on the database. It also serves to execute some application frameworks like Node.js. Finally, the database server hosts our database and all the data found within it. Our Relational Database Management System (RDBMS) runs PostgreSQL.

### 3. Requirements and User Story Backlog

As part of any software project, shareholders and clients have specific requirements for their system. This particular project is centered around the development of a condo management system with a list of main features, as well as some additional features that may be explored in the future. As part of the process of dissecting the project requirements and splitting tasks into manageable and understandable goals, user stories are created. This document serves as a backlog of all user stories and tracks their progress throughout our sprints.

The user stories will take the form of cards using this template:

ID	Title	Open/Closed Status	
<b>As</b> <user> <b>I want</b> <what> <b>So that</b> <value>			
MoSCoW	Business Value	Risk	Effort

The Open/Closed status will also be highlighted with one of the three following colors with their associated meanings:

- Green: New user story
- Yellow: Updated ongoing user story
- Blue: Ongoing user story unchanged from last sprint
- Red: Closed/completed user story

Once acceptance tests pass for the user story, the card's title is crossed out.

Our user stories can primarily be found in the [issues tab of our github](#), but this document will always be updated as sprints progress, and will contain more detailed stories. User story IDs follow IDs automatically generated on github, and as such some of the numbering might be off due to other non user story issues also being part of the numbering system such as pull requests.

2	<a href="#"><u>Upload a profile picture</u></a>	Closed Status	
<b>As</b> any type of user <b>I want</b> to be able to upload a profile picture during account creation <b>So that</b> my profile can be personalized			
S	M	Low	3

3	<a href="#"><u>Registration keys</u></a>	Closed Status	
<b>As</b> a condo management company <b>I want</b> to be able to send out registration keys to users <b>So that</b> their accounts can be linked to their housing			
MoSCoW	XL	Medium	3

4	<a href="#"><u>Properties have unique keys identifying them</u></a>	Closed Status	
<b>As</b> a condo management company <b>I want</b> my properties to be linked to a unique property key <b>So that</b> condo renters and owners can use this as a registration key			
M	M	Medium	2

5	<a href="#"><u>Property dashboard</u></a>	Closed Status	
<b>As</b> a condo owner or renter <b>I want</b> to have a good view (dashboard) of my property units <b>So that</b> they can be managed and I can keep track of issues			
S	XL	Low	4

6	<a href="#"><u>Profile creation</u></a>	Closed Status	
<b>As</b> a public user <b>I want</b> to be able to create a personal profile <b>So that</b> I can access the condo platform			
M	XL	Low	3

7	<a href="#"><u>Create/edit property profiles</u></a>	Closed Status	
<b>As</b> a condo management company <b>I want</b> to be able to create/edit a profile for our properties <b>So that</b> I can manage data related to them			
M	M	Medium	4

8	<a href="#"><u>Property file upload</u></a>	Closed Status	
<b>As</b> a condo management company <b>I want</b> to be able to upload files for each property <b>So that</b> they can be viewed by condo owners and residents			
S	M	Medium	2

9	<a href="#"><u>Edit condo unit information</u></a>	Open Status	
<b>As</b> a condo management company <b>I want</b> to be able to enter detailed information for each condo unit <b>So that</b> condo users and renters can have access to their units' information			
S	L	Medium	4

10	<a href="#"><u>Editing condo fees</u></a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a condo management company to be able to enter a condo fee per square foot and per parking lot unit owners can view costs and estimates			
M	L	Medium	4	

11	<a href="#"><u>Entering costs for annual financial reports</u></a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a condo management company to be able to enter the cost of each operation for the financial system annual financial reports can be generated			
M	M	Low	5	

12	<a href="#"><u>Condo management sign in</u></a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a condo management company to be able to sign in to the condo management system I can access features related to management			
M	XL	Medium	4	

13	<a href="#"><u>View notifications</u></a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a tenant to be able to see any relevant notifications I can be aware of things happening			
S	M	Low	4	

14	<a href="#"><u>Submit employee requests</u></a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a condo management company to be able to assign requests to the relevant employees pressing issues can be resolved quickly			
M	L	Low	3	

15	<a href="#"><u>Common space reservations</u></a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a tenant to be able to reserve common spaces on my calendar I can have guaranteed access to facilities when the time comes			
M	M	Medium	2	

16	<a href="#"><u>Check common space availability</u></a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a tenant to be able to check the availability of a common facility I can plan when I will use that facility, and possibly reserve it			
C	S	Low	2	

17	<a href="#"><u>Rental user registration using key</u></a>			DELETED REDUNDANT
<b>As</b> <b>I want</b> <b>So that</b>	a public user to be able to enter a registration key provided by the condo management company I can gain access to relevant tenant functionalities			
M	XL	High	2	

18	<a href="#"><u>Rental user registration using key</u></a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a public user to be able to enter a registration key provided by the condo management company I can gain access to relevant tenant functionalities			
M	XL	High	2	

19	<a href="#"><u>Set up common facilities for reservation</u></a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	condo management company to be able to set up common facilities on a reservation system tenants can reserve spaces			
S	M	Low	3	

20	<a href="#"><u>Set up employee roles</u></a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a condo management company to set up different roles for employees responsible for the same property tasks can be efficiently distributed and managed			
C	M	Low	2	

21	<a href="#"><u>Elevator reservations</u></a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	condo owner planning to move in/out to be able to reserve an elevator to simplify moving in/out I can have unmitigated access to said elevator for a given time			
C	S	Medium	3	

22	<a href="#"><u>Property dashboard toggle</u></a>	Closed Status	
<b>As</b> condo management company <b>I want</b> to be able to toggle between dashboards for different properties my company manages <b>So that</b> I can easily keep track of status, requests, and financial information for each property			
S	L	Medium	5

24	<a href="#"><u>Condo owner registration with key</u></a>	Closed Status	
<b>As</b> public user <b>I want</b> to be able to enter a registration key from my condo management company <b>So that</b> I can register as a condo owner and access condo owner features/privileges			
M	XL	High	2

25	<a href="#"><u>Registration key generation</u></a>	Closed Status	
<b>As</b> a condo management company <b>I want</b> to be able to generate specialized registration keys <b>So that</b> I can send them to users for their authentication			
M	XLL	High	1

27	<a href="#"><u>Dashboard with functionalities displayed</u></a>	Closed Status	
<b>As</b> a condo management company <b>I want</b> to be able to have my property dashboard display all available functionalities to me <b>So that</b> I can easily navigate through them and have a general overview			
S	M	Medium	3

40	<a href="#">View requests as condo owner</a>			Closed Status
<b>As</b> a condo owner <b>I want</b> to be able to view my requests that are assigned, open, or completed <b>So that</b> I can keep track of issue resolution				
S	L	Low	3	

41	<a href="#">View requests as management company</a>			Closed Status
<b>As</b> a condo management company <b>I want</b> to be able to view all open requests <b>So that</b> they can be assigned to the relevant employees to be handled				
M	L	Low	3	

42	<a href="#">Review request</a>			Closed Status
<b>As</b> a condo owner <b>I want</b> to be able to review a request already submitted <b>So that</b> I can see what the company responded				
C	M	Low	3	

43	<a href="#">Pay rent/balance</a>			Closed Status
<b>As</b> tenant <b>I want</b> to be able to pay any outstanding balance relating to my condo <b>So that</b> I do not accrue any late fees or penalties				
M	L	High	2	

49	<a href="#">Navigate between profile and dashboard</a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a user to be able to quickly navigate between my profile and dashboard using the navbar I can efficiently navigate through the website			
C	S	Low	1	

54	<a href="#">Edit property information</a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a condo management company to be able to edit properties under our management I can view and manage data relating to them			
M	L	Medium	4	

68	<a href="#">Tenant landing page</a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a tenant a landing page displaying all my functionalities I can easily navigate through them and have a general overview			
C	M	Low	2	

81	<a href="#">View all units in a given property</a>			Closed Status
<b>As</b> <b>I want</b> <b>So that</b>	a condo management company to see all units associated to my property I can easily manage them			
M	L	Medium	3	

87	<a href="#">View all bills sent by condo management</a>	Closed Status	
<b>As</b> a public user <b>I want</b> to be able to see all the bills sent by a condo management company <b>So that</b> I can keep track of the status of my payments			
C	L	Medium	2

## 4. Risk Assessment and Management

The purpose of a risk assessment and risk management plan is to preemptively identify, analyze, and address the various potential risks inherent in our project. These risks may manifest in different types, such as technical or management risks. They also exhibit variations in risk scores, determined by comparing the impact and probability of each risk (refer to Figure 1). The Risk Management Plan (RMP) holds significant importance in any software project as it empowers teams to foresee challenges and explore potential solutions to issues that could arise, ultimately impacting the project's success.

Risks were identified during brainstorming meetings among as many teammates as possible. This was to get as many raw ideas as possible and to identify all possible risks with less chance of any slipping through the cracks, or personal bias limiting the scope of our assessment. Once we had all this raw data, a dedicated team of 3 students met to go through the risks and elaborate on them, as well as assign them impact and probability ratings from low to high. This led to the creation of the tables below.

When looking at Impact and Probability, we use these estimates to assess risks:

Impact	1-3 days	3-7 days	7+ days
Probability	0–5%	5–15%	15+%

These are rough estimates of course, but they help us categorize our risks and organize discussions regarding where to place each risk in the table below. Without these concrete numbers, two people might end up with too much variance in their assessments of a risk even if in reality, they were thinking of the same risk categories.

Impact	Low	Medium	High
Probability	1.4 1.6 1.8	1.9 1.12 1.14 1.17	1.3 1.5 1.13 1.15 1.16 1.19
Medium		1.7 1.10	1.11 1.18
High		1.1 1.2	

Table 1: Risk management chart

Risk ID	Risk Type and Description	Risk Score	Resolved in Sprint	Strategy and Effectiveness
US-1.1	<ul style="list-style-type: none"> <li>• Technical</li> <li>• Management</li> <li>• External</li> <li>• Budget</li> <li>• Schedule</li> <li>• Etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Low</li> <li>• Medium</li> <li>• High</li> </ul>		<ul style="list-style-type: none"> <li>• Mitigate</li> <li>• Accept</li> <li>• Avoid</li> <li>• Transfer</li> </ul>
T.1.1	<p><b>Technical</b></p> <p><i>Integration challenges between app and website:</i> The website front-end may not be completely responsive and the components developed in the website may not translate well in the app</p>	High probability Medium impact	Sprint 2	<p><b>Consequence</b> is a reduced user experience for mobile</p> <p><b>Avoid</b> through constant cross-platform and different screen size testing and using react/react native, a framework that facilitates code reuse</p>
T.1.2	<p><b>Technical</b></p> <p><i>Unfamiliarity with the tech stack for front-end and back-end integration:</i> The teams unfamiliarity with the tech stack used can lead to integration difficulties</p>	High probability Medium Impact	Sprint 2	<p><b>Consequence</b> is a higher proportion of bugs and issues in the finished product, as well as a slower development time</p> <p><b>Mitigation</b> through training sessions for team members on the elements that are lesser known technologies, utilize team members strength and previous experiences, frequent meetings so there is coherence between front and back team, so collaboration is seamless.</p>
T.1.3	<p><b>Technical</b></p> <p><i>Inconsistent data handling for different user types:</i> There are many different types of users with different requirements and views of the platform, this could result in</p>	High impact Low probability	Sprint 2	<p><b>Consequence</b> is data errors and thus bugs</p> <p><b>Avoidance</b> through making sure that the data handling protocols are clear and data validation checks are made through the development process</p>

	inconsistencies and potential data errors			
T.1.4	<b>Technical</b> <i>Dependence on external APIs:</i> Dependencies on external APIs for certain features may introduce risks, including changes in API specifications or unexpected downtime.	Low probability Low impact	Sprint 2	<b>Consequence</b> is security risks and dependency on API changes  <b>Accept</b> , we have no control on external software and have no contact with API providers
T.1.5	<b>Technical</b> <i>Insufficient documentation</i> for the system architecture can lead to data redundancy, lack of understanding and inconsistent updates when implementing the database	High impact Low probability	Sprint 1, Sprint 2	<b>Consequence</b> is lower overall code quality and poor understanding by new programmers on the project  <b>Avoid</b> by working on the domain model early on and ensuring collaboration between the development team and the team members working on the system architecture
T.1.6	<b>Management</b> <i>Disorganized teams</i> leading to double work. Multiple teams/members may work on the same features or systems and overwrite each other's contributions. As well as waste time in the process.	Low probability Low impact	Sprint 1, Sprint 2	<b>Consequence</b> is time lost and possible conflict to decide which version is “better”  <b>Mitigate</b> by having team leaders communicating frequently, a Github repository so all members are aware of current work, and task management software such as Trello to organize tasks.
T.1.7	<b>Management</b> <i>Not meeting deadlines</i> or meeting internal goals. Hard deadlines may be missed, disappointing the client and disrupting the overall development timeline	Medium probability Medium impact	Sprint 2	<b>Consequence</b> is disappointing the client, as well as breaching contractual obligations.  <b>Mitigate</b> by creating internal “soft” deadlines to be respected by the team, as well as encouraging communication and cooperation.

T.1.8	<b>Budget</b> <i>Limited budget for tooling can present challenges since this is for a course.</i>	Low probability Low impact	TBA	<b>Consequence</b> is having limited options for hosting and/or tools  <b>Mitigate</b> by using free trials and free versions of tools. Some companies offer free premium versions for students
T.1.9	<b>Management</b> <i>Unavailability of team members due to unforeseen circumstances can impact project progress</i>	Low probability Medium impact	Sprint 2	<b>Consequence</b> is slowing development due to dependency on certain members  <b>Transfer</b> Cross-training team members to be able to perform a variety of tasks and outsourcing tasks when necessary
T.1.10	<b>Technical</b> <i>Insufficient user testing may lead to undetected bugs or usability issues</i>	Medium Probability Medium Impact	Sprint 2	<b>Consequence</b> is bugs and usability issues are discovered by users, impacting their experience negatively  <b>Accept</b> It is an impractical goal to want to ensure with certainty that there are no bugs in the system. Our time and resources being limited, we will not be able to cover all bases. We still need to ensure an adequate amount of testing. We aim for 80% coverage.
T.1.11	<b>External</b> <i>Hacking into the mobile or web application, allowing for abuse of the registration, financial, or reservation systems.</i>	Medium Probability High Impact	Sprint 2	<b>Consequence</b> is data breaches and abuse by a third party  <b>Mitigate</b> by employing responsible coding practices and keep security in mind when designing the system.
T.1.12	<b>Schedule</b> <i>Team may face challenges in conducting regular scrum meetings</i>	Low Probability Medium Impact	Sprint 2	<b>Consequence</b> is a lack of team cohesion and confusion.  <b>Mitigation</b> by holding virtual meetings, utilizing when to meet, making meeting minutes

				so team members can stay updated and establishing clear meeting times to maximize attendance.
T.1.13	<b>Technical</b> <i>Neglecting core features or requirements.</i>	Low Probability High Impact	Sprint 1	<b>Consequence</b> is not meeting client/contractual expectations  <b>Avoid</b> by generating extensive user stories and thorough dissection of client needs.
T.1.14	<b>Management</b> <i>Resource Availability:</i> Potential shortfall in tools and programmers.	Low Probability Medium Impact	Sprint 2	<b>Consequence</b> is starvation for important tasks  <b>Mitigation</b> Resource planning, cross-training and contingency reserves.
T.1.15	<b>Management</b> <i>Lack of stakeholder engagement</i> in the project.	Low probability High Impact	Sprint 2	<b>Consequence</b> is a disconnect between stakeholder wants and developer assumptions  <b>Mitigation</b> Regular update meetings, stakeholder management strategies.
T.1.16	<b>Technical</b> <i>Non-adherence to UI/UX standards:</i> Failure to meet user interface design norms.	Low probability High impact	Sprint 2	<b>Consequence</b> is deprecating the user experience  <b>Mitigation</b> Following UI/UX best practices , user testing.
T.1.17	<b>Technical/Management</b> <i>Misaligned code management practices:</i> Inconsistent use of version control and coding workflows.	Low probability Medium Impact	Sprint 2	<b>Consequence</b> is lower overall code quality and poor understanding by new programmers on the project  <b>Mitigation</b> Mandatory code review policies.
T.1.18	<b>Technical</b> <i>Requirements and user stories backlog mismanagement:</i> Ineffective tracking and prioritization of project tasks.	Medium probability High impact	Sprint 2	<b>Consequence</b> is possibility of missing requirements, as well as a bad priority system of requirements  <b>Mitigation</b>

				Use backlog grooming sessions to keep the backlog updated and prioritized
T.1.19	<p><b>Technical</b>  <i>Design pattern misuse:</i>            Incorrect applications of the design patterns.</p>	Low probability High Impact	Sprint 2	<p><b>Consequence</b> is a worsened user experience</p> <p><b>Mitigation</b>            Conduct design review sessions.</p>

Table 2: List of identified risks

## **5. Web Hosting**

Our web app is being hosted at this link: <https://estate-flow.vercel.app/>

Choosing Vercel to host our website was a decision rooted in reliability, efficiency, and scalability. We saw good reviews and feelings towards this option online. It was also a big advantage that it is free, since we are students and on a limited budget.

# 6. Code Management

All code and documentation can be found on our github <https://github.com/Irisvella/SOEN-390-W2024>. Below, you will find descriptions of our approaches and some examples of the different types of code management techniques undertaken during this project. If any further information is needed, everything can be found in the github provided above.

## Quality of source code reviews

By working in teams, multiple people go over the same code to heighten the quality of the overall code. Our team is divided into teams which fluctuate in members throughout different sprints, as well as during sprints if people have finished their tasks, or reprioritization of features takes place. In light of this, subteam meetings happen a few times per sprint to go over internal goals and to make sure the code is of good quality, as well as clear of bugs. If bugs are found they can be logged and discussed.

## Correct use of design patterns

In designing our website, app, and database, we focused on several design principles:

- We followed a composite pattern when building our structure by splitting our users into types such as “employee” or “management company” and interconnecting them.
- The decorator pattern is utilized by giving different roles, types and other attributes to our entities, providing them with more depth and behaviors.
- There are some observer patterns used when looking at auto increments and decrements that are incorporated into our database structure. Their activators can be said to observe the system and trigger increments/decrements based on events.
- Using an SQL database like us inherently invokes the repository pattern since our data is stored in tables. Each table can store and manage specific types of data.
- During the creation of different types of users, we can say the factory pattern is in play since we are creating types of objects based on a larger common type “user”.

## Respect to code conventions

As we are using React, we use similar coding conventions to Javascript. As part of this, we respect camelCase for identifier names, leave spaces around operators to improve code readability, indent our code, and follow object rules. Comments are also added before methods.

```
// Route to handle the submission of a new listing
router.post(
  '/',
  verifyToken,
  async function (req: Request, res: Response, next: NextFunction) {
    try {
      jwt.verify(
        req.token as string,
        process.env.SECRET as jwt.Secret,
        async (err, decoded) => {
          if (err) {
            return res.status(401).json("Unauthorized");
          } else {
            console.log("decoded ---- ", decoded);
            const { id, role, email } = (any>decoded).data;

            if (role === "company") {
              const body = req.body; //constant

              async function createProperty(address: string) {
                const property = await prisma.property.create({
                  data: {
                    //address on the table to the left
                    //address on the body is to the right
                    address: address,
                    size: 0,
                    condo_fee: 0,
                    unit_id: 0,
                  },
                });
                await new Promise(f => setTimeout(f, 1000)); // wait so that the database has time to update the added property
              }
            }
          }
        }
      )
    }
  }
)
```

## Design quality

Design quality can be measured by many metrics and some of them will be seen below. The main metrics I want to highlight here are Readability, Modularity, Consistency, Scalability, Testability, Documentation and Version Control.

Clear naming conventions and consistent formatting were used to achieve readability by members within our team, but also by outside developers. This is especially important as our repository will remain active for others to reference and learn from.

Modularity was also heavily implemented by the use of components. By building pieces of our site in components, we can both reuse code efficiently (for example the navbar) and cleanly. This also allows for much more manageable code blocks, which can be maintained efficiently.

By having early meetings and laying down rules for our coding style and code conventions, we were able to keep a fairly consistent style across our code. It also helped that we chose our modular, component based coding design. Again, this modularity makes maintaining code and conventions easier.

Scalability was kept in mind when designing our database, our design concept, and our choice of languages. Our database was built with the idea that it could always be scaled up, our modularity allows for simple reuse of components to make expanding the site easy, and our architecture choices leave us some flexibility to grow.

As we are designing an application page by page, which are made of smaller components, testing each one is far simpler than having to test something much larger and more imposing. This ease of testing promotes reliability and reduces the likelihood of bugs passing unnoticed.

Documentation can be seen abundantly out of the code with heavy documentation covering all aspects of the project and offering insight to our intentions, as well as decisions. It can also be found inside the code with proper inline comments, an organized README file and good document title comments.

Finally, Version Control is achieved through the use of Git to constantly moderate our codebase. Not only does it offer the ability to track changes in our project, but it improves our collaboration immensely and protects the codebase from harmful changes being made before it can be checked. In case anything harmful does occur, Git allows for rolling back to previous versions, giving even more safety.

## Quality of source code documentation

We followed three types of documentation conventions: file level comments, component comments and function comments. File level comments help guide the reader and explain what to expect, who wrote this code and any dependencies. Component comments are similar, but are more precise in the role of this component in the larger scheme of things. Function comments are more atomic explanations of how the function operates. As seen in the examples below, these practices are respected.

```
// Filename: addEmployee.jsx
// Author: Sarah Abellard, Samuel Collette, Abisan
// Description: Component for adding new employees to the system.
// Dependencies: React, MUI (Material-UI)
import React, { useState } from 'react';
import Navbar from '../components/Navbar';
import { Box, Typography, TextField, Button, Select, MenuItem, FormControl, InputLabel } from '@mui/material';

const AddEmployee = () => [
  // State for employee data
  const [employeeData, setEmployeeData] = useState({
    email: '',
    role: '',
  });

  // Handle input change
  const handleChange = (e) => {
    const { name, value } = e.target;
    setEmployeeData((prevData) => ({
      ...prevData,
      [name]: value,
    }));
  };

  // Submit form data
  const handleSubmit = async () => {
    const token = localStorage.getItem('token');
    try {
      const response = await fetch('http://localhost:3000/add-employee', {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
          'Authorization': `Bearer ${token}`,
        },
        body: JSON.stringify(employeeData),
      });

      if (response.ok) {
        console.log('Employee data submitted successfully');
        // Optionally, reset the form
        setEmployeeData({
          email: '',
          employee_id: '',
          company_id: '',
        });
      }
    } catch (error) {
      console.error(error);
    }
  };
]
```

## Refactoring activity documented in commit messages

Refactoring was mostly present in the db-changes branch of our project where changes have been made to our database schema as the project has progressed. Many more of these refactors can be found among the commits we have made but they are across all branches. As any other commit, proper naming and descriptions for commits can help us find them among all the commits. Below is an example from the db-changes branch.

add db schemas for operating_fee bt919 committed 3 days ago	2a3fb0f		
Merge pull request #59 from Irisvella/db-changes Irisvella committed 3 days ago - ✓ 2 / 2	(Verified) a79a7a4		
add not null constraints to db bt919 committed 3 days ago ✓ 2 / 2	01d04fd		
fix createEditListing to fit new db bt919 committed 3 days ago ✘ 0 / 2	fce367a		
add db support for condo units bt919 committed 3 days ago ✘ 0 / 2	08a5dfa		
removed deprecated "username" attribute Irisvella committed 3 days ago - ✓ 2 / 2	bad6ae4		
Revert "Merge branch 'main' into db-changes" ... Irisvella committed 3 days ago - ✘ 1 / 2	21b8ab1		
Merge branch 'main' into db-changes Irisvella committed 3 days ago - ✘ 0 / 2	(Verified) 3ad614b		
Fixed Navbar and profile dash ... Irisvella committed 3 days ago	3687a86		

## Quality/detail of commit messages

Each commit is given a brief, specific message to better track commit history and explain the changes made during this specific commit. Along with the use of many, small commits, we can see the progression of any given feature/branch before it is merged into our main branch.

db changes and billing route #69

Merged sarahabellard merged 9 commits into `main` from `db-changes` 4 hours ago

Conversation 1 Commits 9 Checks 2 Files changed 4 +392 -68

Commits on Mar 17, 2024

- add db schemas for operating\_fee**  
bt919 committed 2 days ago

Commits on Mar 18, 2024

- fixed query for dashboard.ts**  
Vaqint committed yesterday
- add billing table to db**  
bt919 committed yesterday
- add billing router, and POST endpoint**  
bt919 committed yesterday
- tweak constraint on registration table**  
bt919 committed yesterday

Commits on Mar 19, 2024

- add GET /billing route**  
bt919 committed 7 hours ago
- add status field to billing table**  
bt919 committed 7 hours ago
- add PATCH /billing route**  
bt919 committed 6 hours ago
- merge main into db-changes**  
bt919 committed 6 hours ago ✓

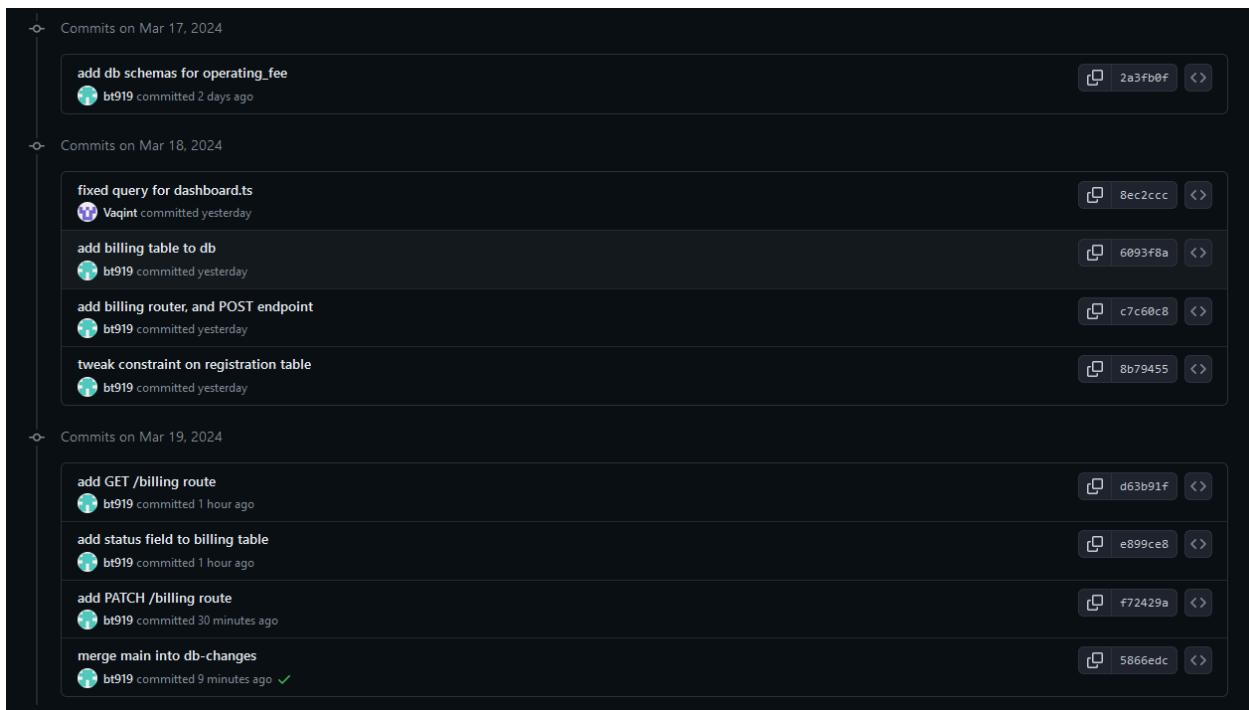
## Use of feature branches

User stories and new features are developed on separate branches from “main” in order to maintain code quality and integrity of the main branch. Upon finishing, a merge request is made and the branch is reviewed. If no problems are found, it can be merged.

<input type="checkbox"/>  <a href="#">Updated Navbar ✓</a>	 1
#51 by Irisvella was merged 2 weeks ago • Approved	
<input type="checkbox"/>  <a href="#">Landing Management On-clicks Update (Dashboard and Listing) ✓</a>	 1
#48 by Zeebruh326 was merged 3 weeks ago • Approved	
<input type="checkbox"/>  <a href="#">Nav column (haven't update on-clicks) ✓</a>	 1
#47 by Zeebruh326 was merged 3 weeks ago • Approved	
<input type="checkbox"/>  <a href="#">Enable testing and build for web-app portion + Jest set up ✘</a>	
#46 by Irisvella was closed 5 days ago • Review required	
<input type="checkbox"/>  <a href="#">Employees roles tables ✓</a> <small>sprint 2 user story</small>	
#45 by sarahabellard was closed 3 weeks ago • Review required	
<input type="checkbox"/>  <a href="#">Dashboard ✓</a>	 3
#44 by Ashx11 was merged 3 weeks ago • Approved	
<input type="checkbox"/>  <a href="#">Create edit listing ✓</a>	 1
#39 by Fatoumatabintabarry was merged 3 weeks ago • Approved	
<input type="checkbox"/>  <a href="#">Front-end , EditListingPage ✓</a>	 2
#38 by Fatoumatabintabarry was closed 3 weeks ago • Changes requested	
<input type="checkbox"/>  <a href="#">Add files via upload ✓</a>	 1
#37 by Ashx11 was merged 2 weeks ago • Approved	
<input type="checkbox"/>  <a href="#">Landing management</a>	 1
#36 by Zeebruh326 was merged 3 weeks ago • Approved	
<input type="checkbox"/>  <a href="#">Add files via upload ✘</a>	 1
#35 by Ashx11 was closed 3 weeks ago • Review required	
<input type="checkbox"/>  <a href="#">Integration testing for authentication function ✓</a>	 1
#34 by Irisvella was merged 3 weeks ago • Approved	

## Atomic commits

Team members are encouraged to commit often and to keep commits focused. This is to help other members stay aware of progress of their work. This mainly helps keep the team up to date and organized, but also helps avoid multiple people working on the same feature and only realizing much too late when they make a larger commit. On a broader level however, this practice makes code review, finding bugs, and reverting changes if need be extremely efficient.



The screenshot shows a GitHub repository interface with three sections of commits:

- Commits on Mar 17, 2024:**
  - add db schemas for operating\_fee  
by  bt919 committed 2 days ago
- Commits on Mar 18, 2024:**
  - fixed query for dashboard.ts  
by  Vaqint committed yesterday
  - add billing table to db  
by  bt919 committed yesterday
  - add billing router, and POST endpoint  
by  bt919 committed yesterday
  - tweak constraint on registration table  
by  bt919 committed yesterday
- Commits on Mar 19, 2024:**
  - add GET /billing route  
by  bt919 committed 1 hour ago
  - add status field to billing table  
by  bt919 committed 1 hour ago
  - add PATCH /billing route  
by  bt919 committed 30 minutes ago
  - merge main into db-changes  
by  bt919 committed 9 minutes ago ✓

## Bug reporting

In the issues tab of our github repository, bugs are reported and labeled to keep track. When they are addressed, the issue can be marked as closed. All necessary information about the bug can be found in the “bug” issue.

-  #63 Build Failure bug Request2Management (merge #62)   
#63 opened 2 days ago by Fatoumatabintabarry
  -  #57 Create (add) new property listing bug.   
#57 opened 2 weeks ago by Fatoumatabintabarry
  -  #54 As a condo management company, I want to be able to edit properties under our management to be able to view and manage the data related to them (4 USP)   
#54 opened 2 weeks ago by Fatoumatabintabarry
  -  Carousel Bug - displaying vertical stack   
#50 opened 2 weeks ago by Vajint
- 
-  No routes to escape login/signup/profile loop   
#31 by Irisvella was closed last month

## Use of issue labels for tracking and filtering

Labels such as “user story”, “minor bug”, “sprint 1”, etc... are used to identify and categorize issues.

		Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾	
<input type="checkbox"/>	<input checked="" type="checkbox"/> As a user, I want to be able to quickly navigate between my profile and dashboard using the navbar (2USP)		<a href="#">user story</a>					
		#49 by Irisvella						was closed 2 weeks ago
<input type="checkbox"/>	<input checked="" type="checkbox"/> No routes to escape login/signup/profile loop		<a href="#">minor bug</a>					 1
		#31 by Irisvella						was closed last month
<input type="checkbox"/>	<input checked="" type="checkbox"/> #27 As a condo management company, I would like to be able to have a dashboard that displays all the functionalities available to me , so that i can easily navigate through them and have a general overview .							 1
		(3 USP)	<a href="#">sprint 2</a>					
		#27 by Fatoumatabintabarry						was closed 2 weeks ago
<input type="checkbox"/>	<input checked="" type="checkbox"/> As a public user, I need to enter a registration key from my condo management company to register as a rental user, gaining access to relevant functionalities.		<a href="#">user story</a>					
		#17 by Fatoumatabintabarry						was closed on Feb 3
<input type="checkbox"/>	<input checked="" type="checkbox"/> #12 As a user of the condo management system, I would like to sign in to the condo management system to be able to use the features related to my user type (USP 4)		<a href="#">sprint 1</a>	<a href="#">user story</a>				 1
		#12 by sarahabellard						was closed 3 weeks ago
<input type="checkbox"/>	<input checked="" type="checkbox"/> #7 As a condo management company, I want to be able to create properties under our management to be able to view and manage the data related to them (4 USP)		<a href="#">sprint 2</a>	<a href="#">user story</a>				
		#7 by sarahabellard						was closed 2 weeks ago
<input type="checkbox"/>	<input checked="" type="checkbox"/> #2 As a public user, I want to upload a profile picture during account creation, so my profile feels personalized. (3 USP)		<a href="#">sprint 2</a>	<a href="#">user story</a>				 1
		#2 by sarahabellard						was closed 3 weeks ago

## Links between commits and bug reports/features



Fatoumatabintabarry mentioned this pull request 34 minutes ago

#54 As a condo management company, I want to be able to edit properties under our management to be able to view and manage the data related to them (4 USP) #54

Closed

## 7. Release Plans

### 7.1 Sprint 2 Release Plan

**Goal:** Enhance user experience and functionality for condo owners and management companies, focusing on request management, condo listing management, and user account features.

**Duration:** 3 weeks

#### Key Features:

##### Condo Owner Requests Management:

- Condo owners can view open requests.
- Condo owners can create new requests.
- Condo owners can send additional responses if needed.
- Seamless Interface for Request Creation:
- Develop a user-friendly interface for creating requests to ensure a smooth user experience.

##### Condo Management Company Request Handling:

- Condo management companies can view requests.
- Management companies can respond to requests.
- Management companies can assign specific jobs to employees related to the requests.

##### Efficient Condo Navigation for Owners:

- Implement features allowing condo owners to navigate swiftly through all their condos within their account.

### **Condo Listing Management for Management Companies:**

- Management companies can edit condo listings.
- They can add/remove pictures.
- They can add amenities and create descriptions for places and surroundings.
- They can specify the number of units available and parking spots

### **Employee Management for Management Companies:**

- Management companies can add roles to their employees.
- They can edit existing employee information.
- They can add or remove employees as necessary.

### **User Account Condo Management:**

- Users can add condos to their account.
- Added condos will be accessible for viewing and managing.

### **Tasks Breakdown:**

#### **UI/UX Design:**

- Design a seamless interface for request creation.
- Design an intuitive navigation system for condo owners.

#### **Frontend Development:**

- Implement frontend components for request management.
- Develop the user interface for condo navigation and request creation.

#### **Backend Development:**

- Set up database schemas for request management.
- Develop backend functionalities for request handling by management companies.
- Implement backend logic for condo listing management and user account features.

**Testing:**

- Conduct thorough testing of all features.
- Fix bugs and optimize performance.

**Documentation:**

- Document API endpoints and usage instructions.
- Update user manuals and guides.

**Timeline:**

Week 1:

UI/UX Design

Frontend Development (Request Management UI)

Backend Development (Request Management)

Week 2:

Frontend Development (Condo Navigation UI, Condo Listing Management UI)

Backend Development (Condo Listing Management, User Account Features)

Week 3:

Testing and Documentation

**Release:** At the end of Sprint 2, the release will be deployed to production, and users will have access to the new features and enhancements.

## 7.2 Sprint 3 Release Plan

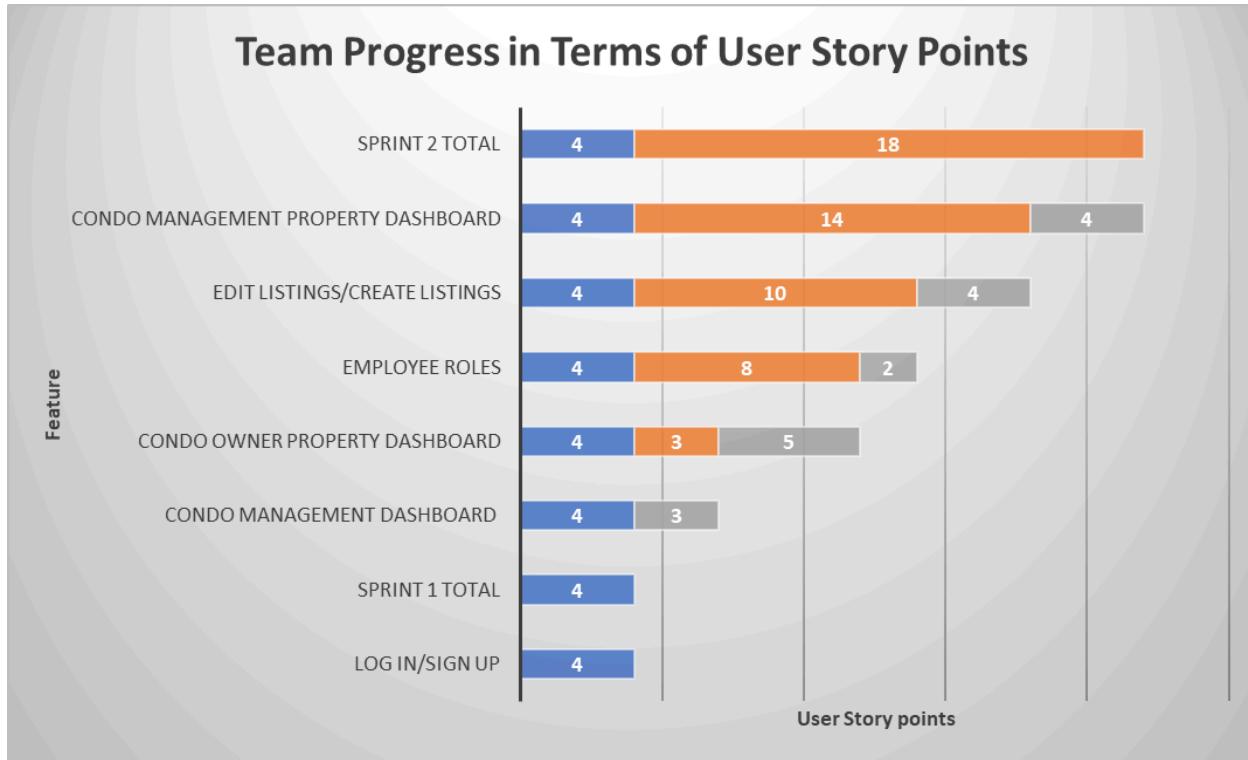


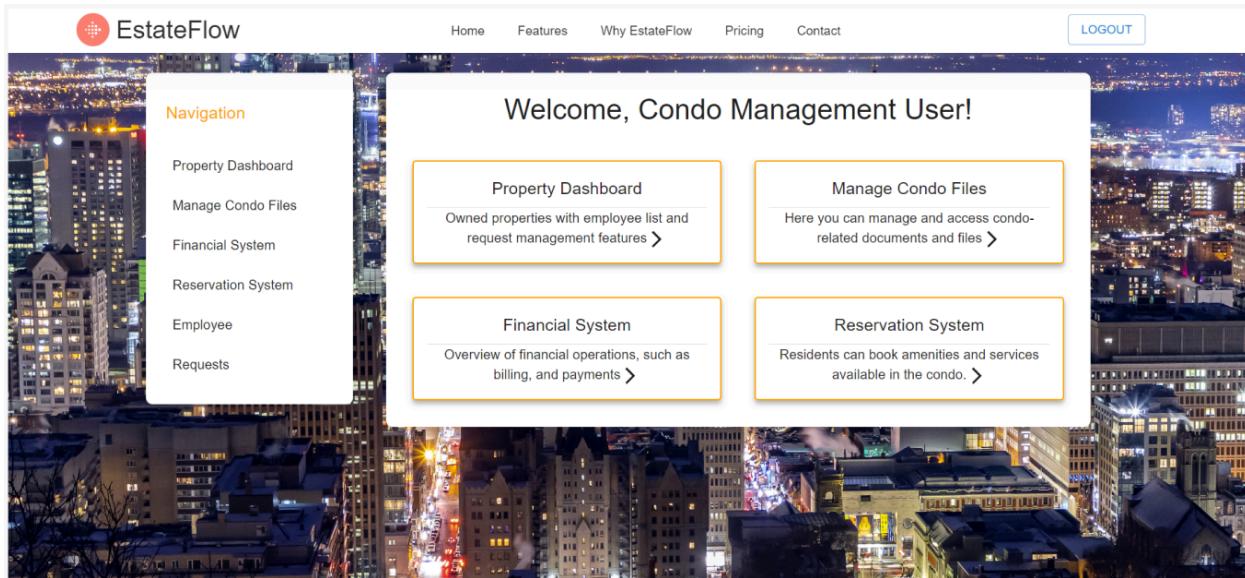
Chart 1: Team progress Sprint 2

Regarding our Sprint 2 Release plan. We first realized that it was quite ambitious to be able to do everything that was previously mentioned. We had a meeting to refocus our priorities regarding the deliverables. Rearrange our user story points in terms of difficulty and importance. Although our previous ambitions were quite high we were still available to deliver 18 user story points. The Condo Management company side of the app was a lot more focused on this sprint, with features such as editing and creating listings, and creating and managing employee roles.

In Sprint 3, we plan on releasing many features more focused towards the user side of the app. This will include the financial capabilities for the user and request creation.

User Story ID	User Story Points (USP)	Priority	Status
<a href="#">#27</a>	3	• High	• DONE
<a href="#">#22</a>	5	• High	• DONE

<a href="#"><u>#20</u></a>	2	● Low	● DONE
<a href="#"><u>#7</u></a>	4	● High	● DONE
<a href="#"><u>#5</u></a>	4	● Medium	● DONE
<a href="#"><u>#14</u></a>	3	● Medium	● PUSHED TO SPRINT 3
<a href="#"><u>#21</u></a>	3	● Medium	● PUSHED TO SPRINT 3
<a href="#"><u>#15</u></a>	2	● Medium	● TO DO
<a href="#"><u>#14</u></a>	3	● Medium	● PUSHED TO SPRINT 3
<a href="#"><u>#41</u></a>	3	● Medium	● PUSHED TO SPRINT 3
<a href="#"><u>#40</u></a>	3	● Medium	● PUSHED TO SPRINT 3
<a href="#"><u>#42</u></a>	3	● Medium	● PUSHED TO SPRINT 3
<a href="#"><u>#43</u></a>	2	● Medium	● PUSHED TO SPRINT 3
<a href="#"><u>#24</u></a>	2	● High	● PUSHED TO SPRINT 3
<b>Total USP</b>	42		



User Story ID	User Story Points (USP)	Priority	Status
<a href="#"><u>#27</u></a>	3	• High	• DONE

Properties

EstateFlow

Home Features Why EstateFlow Pricing Contact

Welcome, CoolCompany ▾





**36 Lee drive, H8B 3M6**

[View Employee List](#)

[Requests](#)



User Story ID	User Story Points (USP)	Priority	Status
<a href="#">#22</a>	5	• High	• DONE

[View employees page](#)

EstateFlow

[Home](#) [Features](#) [Why EstateFlow](#) [Pricing](#) [Contact](#)Welcome,  
CoolCompany

36 Lee drive, H8B 3M6

ID	First name	Last name	Role	Actions
1	John	Doe	Plumber	
2	Jane	Doeh	Lifeguard	
2	Jeremy	Robertson	Representative	

[Submit](#)

User Story ID	User Story Points (USP)	Priority	Status
<a href="#">#20</a>	2	● Low	● DONE

[Edit listing page](#)[Home](#) [Features](#) [Why EstateFlow](#) [Pricing](#) [Contact](#)Welcome,  
CoolCompany ▾

## Photo album



Address: 36 Lee drive

Postal Code: H8B 3M6

Total Units: 12

Parking spaces: 16

**Amenities:**

- Indoor pool
- Includes indoor parking
- Newly renovated (2020)

**Description:** Beautiful newly renovated condo, in a great neighborhood. Schools nearby, great for new families. Includes an indoor pool and underground parking. Feel free to contact us for any inquiries.  
 Phone number: 514-565-5859  
Email: [Inquiries@Coolcompany.com](mailto:Inquiries@Coolcompany.com)

[Submit](#)

User Story ID	User Story Points (USP)	Priority	Status
<a href="#">#7</a>	4	• High	• DONE



36 Lee drive, H8B 3M6

[View Requests](#)[Make request](#)[Financial Information](#)

User Story ID	User Story Points (USP)	Priority	Status
<a href="#">#5</a>	4	● Medium	● DONE

Create Request


**EstateFlow**

- [Home](#)
- [Features](#)
- [Why EstateFlow](#)
- [Pricing](#)
- [Contact](#)

Welcome, John



**36 Lee drive, H8B 3M6**

Request type: **Select type...**

Date:  

Time:  

Reason for request:

**Submit**

User Story ID	User Story Points (USP)	Priority	Status
<a href="#"><u>#21</u></a>	3	● Medium	<ul style="list-style-type: none"> <li>● PUSHED TO SPRINT 3</li> </ul>
<a href="#"><u>#15</u></a>	2	● Medium	<ul style="list-style-type: none"> <li>● TO DO</li> </ul>

As a user of our app, people need to be able to create certain requests like being to reserve certain amenities like the elevator or more luxurious amenities like a spa lounge. This is of medium importance because the amenities can be reserved through emails to start off and our website can add this feature later to give help to these management companies. We plan to implement more serious matters at first to be sure that elevator requests are managed for example.



Type: Deficiency in common areas

Status: In Progress

36 Lee drive, H8B  
3M6

Date Created: 05/02/2024

## Description:

No more snacks in the common area. We would like to have granola bars restocked and also the fresh fruit.

## Your Initial Response:

The fresh fruit will not be returning as it was too expensive to keep restocking. However, we have heard your request for the granola bar requests and will try and get those restocked as soon as possible.

## Assigned personnel:

Jeremy Robertson

[Update](#)[Mark as Resolved](#)[Reject Request](#)

User Story ID	User Story Points (USP)	Priority	Status
#14	3	• Medium	• PUSHED TO SPRINT 3

As a condo management company, they need to be able to assign personnel to the tasks and requests being created by users. This is of medium importance as it can be done internally through personal communications in the mean time. Now it will be one of the priorities to be able to release the management companies.

Open Requests - Management



# EstateFlow

Home Features Why EstateFlow Pricing Contact

Welcome, John ▾



**36 Lee drive, H8B 3M6**

- 1. Moving Out (Elevator Request) - 18/03/2024 - 12pm - 3pm - Awaiting Response
- 2. Deficiency in common areas - No more snacks in the common area, we... - Task assigned
- 3. Reporting a violation - Unit 4 has been practicing tap dancing during... - Resolved

User Story ID	User Story Points (USP)	Priority	Status
<a href="#">#41</a>	3	● Medium	● PUSHED TO SPRINT 3

As a company, they should be able to see all the opened requests and with a quick view be able to identify what is an important issue versus something they can push to the side. A medium priority is placed as this can again be done through an email system for the mean time, although not ideal. Which is why it is important for us to deliver this to our stakeholders for sprint 3.

Open Requests


**EstateFlow**

- [Home](#)
- [Features](#)
- [Why EstateFlow](#)
- [Pricing](#)
- [Contact](#)

Welcome, John



**36 Lee drive, H8B 3M6**

1. Moving Out (Elevator Request)	-	18/03/2024	-	12pm - 3pm	-	Pending...
2. Deficiency in common areas	-	No more snacks in the common area, we...	-	In Progress		
3. Reporting a violation	-	Unit 4 has been practicing tap dancing during...	-	Resolved		

User Story ID	User Story Points (USP)	Priority	Status
#40	3	● Medium	● PUSHED TO SPRINT 3

Similarly to above, the users should be able to have the same, or similar view of their requests. For the same reason as before, the medium priority is placed, and which is why these pages and tasks were pushed to sprint 3.

Request 2


**EstateFlow**

- Home
- Features
- Why EstateFlow
- Pricing
- Contact

Welcome, John



Type: Deficiency in common areas

Status: In Progress

**36 Lee drive, H8B 3M6**

Date Created: 05/02/2024

Your description of request:

No more snacks in the common area. We would like to have granola bars restocked and also the fresh fruit.

Initial Response:

The fresh fruit will not be returning as it was too expensive to keep restocking. However, we have heard your request for the granola bar requests and will try and get those restocked as soon as possible.

Still having issues? Please contact us through email if you have additional inquiries.

[Email us](#)
[Mark as Resolved](#)
[Cancel Request](#)

User Story ID	User Story Points (USP)	Priority	Status
<a href="#">#42</a>	3	● Medium	● PUSHED TO SPRINT 3

User should be able to have a more detailed view of their requests that have been assigned. Without this they wouldn't be able to know if their requests have been heard and accepted. This again is placed with a medium priority.

Make payment


**EstateFlow**

- [Home](#)
- [Features](#)
- [Why EstateFlow](#)
- [Pricing](#)
- [Contact](#)

Welcome,  
John



**Payment Due: 1256.76\$**

**Due Date: 05/03/2024**

**36 Lee drive, H8B 3M6**



**Card number:**

**Expiry date:**  

User Story ID	User Story Points (USP)	Priority	Status
<a href="#">#43</a>	2	• High	• PUSHED TO SPRINT 3

The importance of the users being able to make payments is very important in our opinion. Users should not have to rely on checks or other forms of physical payment. User should be able to do it straight from our website. This is a High priority for this sprint.

Add condos


**EstateFlow**

- [Home](#)
- [Features](#)
- [Why EstateFlow](#)
- [Pricing](#)
- [Contact](#)

Welcome,  
John

Please enter Condo Code

**Submit**

User Story ID	User Story Points (USP)	Priority	Status
<a href="#"><u>#24</u></a>	2	• High	• PUSHED TO SPRINT 3

This sprint will also allow users to add their condos themselves, which will be very good in terms of independence. Users will be able to easily add their condos whether they are a renter or a owner of the condo.

## 7.3 Sprint 4 Release Plan

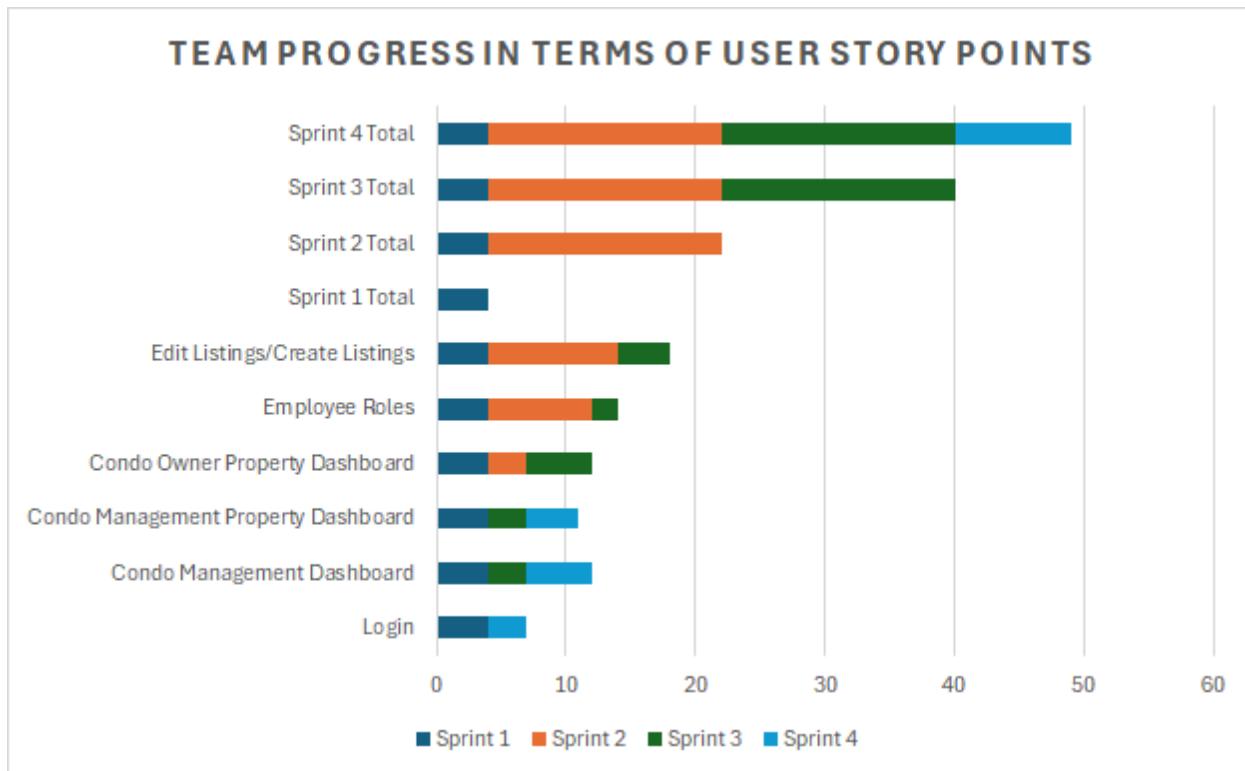


Chart 2: Team progress Sprint 4

As Sprint 2 was extended, we had less time to tackle our projected tasks for Sprint 3. Despite this challenge, we managed to complete numerous user stories, primarily focusing on front-end page development and ensuring database updates aligned with requirements. However, due to time constraints, some functionalities could not be fully realized. Nonetheless, notable advancements were achieved, particularly within critical areas such as the financial system and handling requests. Additionally, addressing lingering bugs from previous sprints consumed a portion of our resources. Consequently, the number of user stories slated for Sprint 4 will be influenced by these factors.

User Story ID	User Story Points (USP)	Priority	Status
	3	• Medium	• SPRINT 4

<u>[#3]</u>			
<u>#5</u>	4	● High	● SPRINT 4
<u>#9</u>	4	● Medium	● SPRINT 4
<u>#10</u>	4	● Medium	● SPRINT 4
<u>#11</u>	5	● Low	● SPRINT 4
<u>#40</u>	3	● Medium	● SPRINT 4
<u>6</u>	3		● DONE
<u>20</u>	2		● DONE
<u>21</u>	3		● DONE
<u>54</u>	4		● DONE
<u>68</u>	2		● DONE
<b>Total USP</b>	<b>343</b>		

## 7.4 Sprint 5 Release Plan

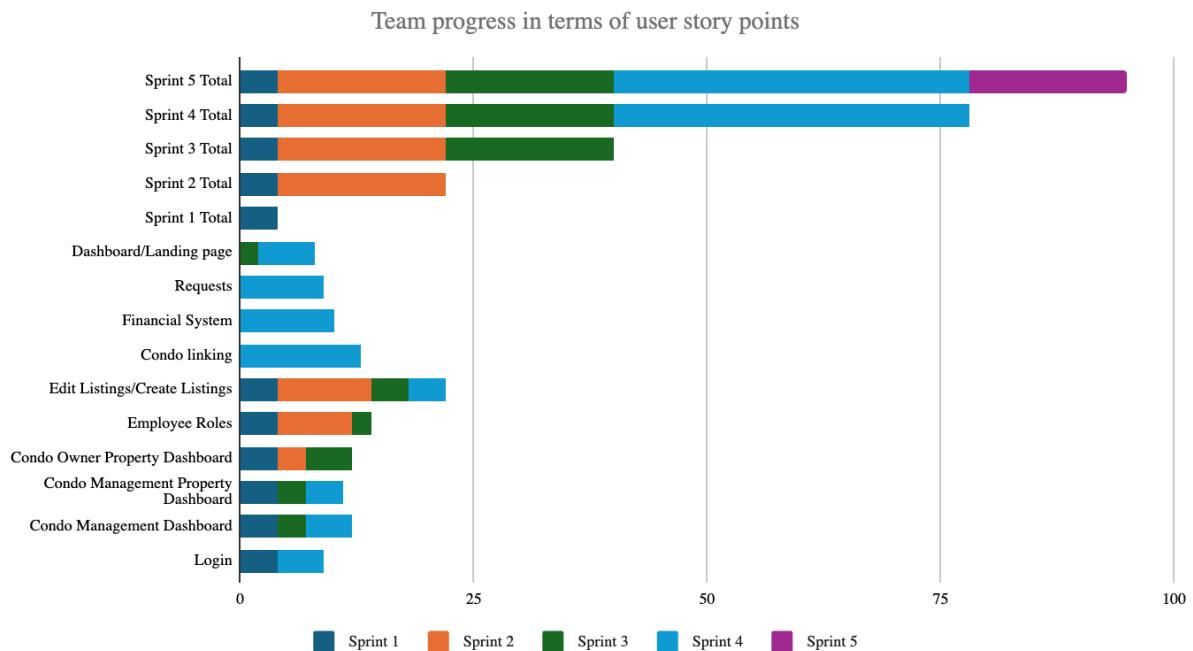


Chart 3: Team progress Sprint 5

As Sprint 4 was extended, our team capitalized on the additional time, achieving a record-breaking completion of 40 user story points. This milestone not only reflects our highest productivity yet but also demonstrates our adaptability and dedication to the project's goals. We successfully addressed a significant portion of the backlog, pushing forward critical features and enhancements.

Looking ahead to Sprint 5, while recognizing that there is still substantial work to be completed, we want to channel our efforts towards the reservation system and notifications. These features are crucial for the project's next phase and will be our top priority. Our goal for the upcoming sprint is not only to continue our momentum but to finalize all the planned functionalities, ensuring a robust and efficient delivery.

User Story ID	User Story Points (USP)	Priority	Status
<a href="#"><u>8</u></a>	2	LOW	SPRINT 5
<a href="#"><u>9</u></a>	4	HIGH	SPRINT 5
<a href="#"><u>13</u></a>	4	MEDIUM	SPRINT 5
<a href="#"><u>15</u></a>	2	MEDIUM	SPRINT 5
<a href="#"><u>16</u></a>	2	LOW	SPRINT 5
<a href="#"><u>19</u></a>	3	HIGH	SPRINT 5
<a href="#"><u>[#3]</u></a>	3	-	DONE
<a href="#"><u>#10</u></a>	4	-	DONE
<a href="#"><u>#11</u></a>	5	-	DONE
<a href="#"><u>#40</u></a>	3	-	DONE
<a href="#"><u>#24</u></a>	2	-	DONE
<a href="#"><u>#25</u></a>	1	-	DONE

<u>#41</u>	3	-	DONE
<u>#18</u>	2	-	DONE
<u>#4</u>	2	-	DONE
<u>#42</u>	3	-	DONE
<u>#14</u>	3	-	DONE
<u>#81</u>	3	-	DONE
<u>#87</u>	2	-	DONE
<u>#10</u>	4	-	DONE
<b>SPRINT 4 Total</b>	40		
<b>SPRINT 5 Total</b>	17		
<b>Total USP</b>	57		

# 8 Testing Plans

## 8.1 Sprint 2 Testing Plan

### Sprint 1 Overview

Because no functionalities were fully implemented during Sprint 1, no tests were written yet. Instead, focus was shifted to discovering and learning about testing tools and methods.

### Testing Plan for Sprint 2

Since the applications are in React/React Native, Jest will be used for unit testing. Testing will be automated through a CI/CD pipeline and will run on each push. We wish to achieve a recommended code coverage of 80%.

### Sprint 2 Unit tests

Unit tests will be written for features we started to implement during Sprint 1, and those we will implement in Sprint 2, which include:

- handleLogin(): A user logs in
- handleSignUp(): A user signs up
- updateProfile(): A user updates information on their profile
- addListing() : A condo management company adds a listing

### Sprint 2 Integration tests

Integration testing will be conducted to test interactions with our database. While unit tests will check if functions such as handleLogin() work using hardcoded data, integration testing will test if the handleLogin() component successfully communicates with the database. We will set up a testing state database for such purposes.

## 8.2 Sprint 3 Testing Plan

### Sprint 2 Overview

#### What was done

We began integration testing and unit testing for the backend functions and the frontend for the web application using Jest. The login and logout functions were tested using integration testing and the frontend elements were tested for correct rendering. Due to the unexpected length of development and testing, not all features implemented during Sprint 2 were tested.

Additionally, a CI/CD pipeline was established to run tests and check for build errors on each push.

#### Testing coverage

For the frontend tests, a coverage of 80% in most sections was achieved for elements tested.

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	80	100	66.66	80	
Navbar.jsx	80	100	66.66	80	21

For the backend integration testing, a coverage of 80% was achieved for elements tested.

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	94.18	85.71	100	94.04	
login.ts	97.14	88.88	100	97.14	81
signup.ts	92.15	80	100	91.83	79,136-140

Test Suites: 2 passed, 2 total  
Tests: 15 passed, 15 total

## **Acceptance Testing**

User acceptance testing was performed at the end of the sprint to ensure that all features implemented respect the requirements outlined in the project files. For the features implemented during Sprint 2, user acceptance testing passed.

## **Testing Plan for Sprint 3**

### **Sprint 3 Unit tests**

For sprint 3, we plan on testing all of the frontend elements implemented during Sprint 2, which include the user and company dashboard, the employee listing, and the condo listing.

### **Sprint 3 Integration tests**

For sprint 3, we plan on testing all backend-frontend communications for features implemented during Sprint 2, which include the dashboard, adding and editing a listing, editing a user profile, and adding employees. Furthermore, testing for features implemented during Sprint 3 is expected to be done, which will include features such as requesting a key, requesting a ticket, reserving amenities, etc.

## **8.3 Sprint 4 Testing Plan**

## **Sprint 3 Overview**

### **What was done**

We continued integration testing and unit testing for the backend functions and the frontend for the web application using Jest. This sprint, we focused on unit testing and integration testing of front-end elements. All authentication components were tested and a coverage of 80% was achieved.

Sprint 3 contained a database overhaul which led to many tests implemented during Sprint 2 to fail, especially those involving backend integration testing. Thus, a substantial amount of effort was dedicated to fixing the broken tests.

## Testing coverage

For the frontend tests, a coverage of 80% was achieved for elements tested.

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	83.82	85.15	81.66	83.45	
components	77.58	57.14	56.25	77.58	
Navbar.jsx	77.58	57.14	56.25	77.58	36-42,86-172,196
components/Authentication	94.96	94.11	100	94.77	
CompanyLoginForm.tsx	92	93.75	100	91.66	35-36
CompanySignupForm.tsx	95.23	90.9	100	95	56-57
Login.tsx	100	100	100	100	
SignUp.tsx	100	100	100	100	
UserLoginForm.tsx	92	93.75	100	91.66	35-36
UserSignupForm.tsx	95.23	96.96	100	95	53-54
components/EditProfile/src/components/Content	58.18	20	60	58.18	
AccountSettings.jsx	45.23	20	42.85	45.23	25-36,47-72,87-108
Actions.jsx	100	100	100	100	
Content.jsx	100	100	100	100	

For the backend integration testing, a coverage of 80% was achieved for elements tested.

File	%Stmts	%Branch	%Funcs	%Lines	Uncovered Line #s
All files	94.18	85.71	100	94.04	
login.ts	97.14	88.88	100	97.14	81
signup.ts	92.15	80	100	91.83	79,136-140

Test Suites: 2 passed, 2 total  
Tests: 15 passed, 15 total

## Acceptance Testing

User acceptance testing was performed at the end of the sprint to ensure that all features implemented respect the requirements outlined in the project files. Acceptance tests include: A user creating an account as a Public User, going through their dashboard, and being able to edit their profile. A user can also create an account as a Management Company, go through their properties dashboard, create a new property, edit their existing properties, and edit their profile.

For the features implemented during Sprint 3, user acceptance testing passed.

# **Testing Plan for Sprint 4**

## **Sprint 4 Unit tests**

For sprint 4, we plan on testing all of the frontend elements implemented during Sprint 3, which include the employee listing, adding units to a property, and the condo listing.

## **Sprint 4 Integration tests**

For sprint 4, we plan on testing all backend-frontend communications for features implemented during Sprint 3, which include adding and editing a listing, editing a user profile, and adding employees. Furthermore, testing for features implemented during Sprint 3 is expected to be done, which will include features such as requesting a key, requesting a ticket, reserving amenities, etc.

## 8.4 Sprint 5 Testing Plan

### Sprint 4 Overview

#### What was done

We continued integration testing and unit testing for the backend functions and the frontend for the web application using Jest. This sprint, we focused on unit testing and integration testing of front-end elements. All authentication components were tested and a coverage of 80% was achieved.

Sprint 4 contained many new features and focus was on performing acceptance tests and writing acceptance test cards for all previous and new user story features.

#### Testing coverage

For the frontend tests, a coverage of 80% was achieved for elements tested.

File	%Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	83.82	85.15	81.66	83.45	
components	77.58	57.14	56.25	77.58	
Navbar.jsx	77.58	57.14	56.25	77.58	36-42,86-172,196
components/Authentication	94.96	94.11	100	94.77	
CompanyLoginForm.tsx	92	93.75	100	91.66	35-36
CompanySignupForm.tsx	95.23	90.9	100	95	56-57
Login.tsx	100	100	100	100	
SignUp.tsx	100	100	100	100	
UserLoginForm.tsx	92	93.75	100	91.66	35-36
UserSignupForm.tsx	95.23	96.96	100	95	53-54
components/EditProfile/src/components/Content	58.18	20	60	58.18	
AccountSettings.jsx	45.23	20	42.85	45.23	25-36,47-72,87-108
Actions.jsx	100	100	100	100	
Content.jsx	100	100	100	100	

For the backend integration testing, a coverage of 80% was achieved for elements tested.

File	% Stmt	% Branch	% Func	% Lines	Uncovered Line #s
All files	94.18	85.71	100	94.04	
login.ts	97.14	88.88	100	97.14	81
signup.ts	92.15	80	100	91.83	79,136-140
Test Suites: 2 passed, 2 total					
Tests: 15 passed, 15 total					

## Acceptance Testing

User acceptance testing was performed at the end of the sprint to ensure that all features implemented respect the requirements outlined in the project files. Here are the acceptance tests:

### AT - 1

<b>Related User Story</b>	#2 As a public user, I want to upload a profile picture during account creation, so my profile feels personalized. (3 USP) sprint 2 user story
<b>Related System Tests</b>	Unit testing
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"><li>1. User clicks on their profile</li><li>2. User clicks “add profile image”</li><li>3. User adds profile image</li><li>4. Image is added as their profile picture</li></ol>
<b>Label</b>	Open
<b>Comments</b>	On mobile

### AT-2

<b>Related User Story</b>	#3 As a condo management company, I would like to be able to send out registration keys to users so their account can be linked to their housing (3 USP) user story
<b>Related System Tests</b>	Unit testing
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"><li>1. Condo company clicks on a unit</li><li>2. Company clicks on “generate key”</li><li>3. A registration key is generated</li></ol>
<b>Label</b>	Closed
<b>Comments</b>	On web-app

### AT 3

Related User Story	#4 As a condo management company, I would like my properties to be linked to a unique property key to be used by condo renters and condo owners as a registration key (2 USP) user story
Related System Tests	Unit testing
Acceptance Criteria	<ol style="list-style-type: none"><li>1. Condo company clicks on a unit</li><li>2. Company clicks on “generate key”</li><li>3. A unique registration key is generated</li></ol>
Label	Closed
Comments	

### AT-4

Related User Story	#5 As a condo owner, I would like to have a good view (dashboard) of my properties to be able to manage them and keep track of issues (4 USP) sprint 2 user story
Related System Tests	Unit testing
Acceptance Criteria	<ol style="list-style-type: none"><li>1. User clicks on “dashboard”</li><li>2. Use can see their properties on their dashboard</li><li>3. If the user doesn’t have any properties, nothing is displayed.</li></ol>
Label	Closed
Comments	

**AT-5**

<b>Related User Story</b>	#6 As a public user, I would like to be able to create a profile to access the condo management platform (USP 3) sprint 3 user story
<b>Related System Tests</b>	Integration testing, unit testing
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"><li>1. A user clicks on “Create an Account” on the home page</li><li>2. A user creates their accounts by entering their credentials</li><li>3. A user can log in to their profile</li></ol>
<b>Label</b>	Closed
<b>Comments</b>	

**AT-6**

<b>Related User Story</b>	#7 As a condo management company, I want to be able to create properties under our management to be able to view and manage the data related to them (4 USP) sprint 2 user story
<b>Related System Tests</b>	Unit testing
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"><li>1. A condo management company clicks on “view dashboard”</li><li>2. They click on “Add property”</li><li>3. They enter all the property details and click on “Add Property” again</li><li>4. The property is now on their dashboard</li></ol>
<b>Label</b>	Closed
<b>Comments</b>	

**AT-7**

<b>Related User Story</b>	#10 As a condo management company, I want to be able to enter a condo fee per square foot and per parking spot to be calculated and presented to unit owners. (4 USP) user story
<b>Related System Tests</b>	Unit testing
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"><li>1. A condo management company clicks on “view dashboard”</li><li>2. They click on “Add property”</li><li>3. They enter all the property details such as condo fee per square feet and click on “Add Property” again</li><li>4. The property is now on their dashboard</li></ol>
<b>Label</b>	Closed
<b>Comments</b>	

**AT-8**

<b>Related User Story</b>	#11 As a condo management company, I want to be able to enter the cost for each operation for the financial system to generate an annual financial report (5 USP) user story
<b>Related System Tests</b>	Unit testing
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"><li>1. A condo management company clicks on “dashboard”</li><li>2. They click on “Financial System”</li><li>3. They can enter costs for each operation</li><li>4. They click “Confirm”</li></ol>
<b>Label</b>	Open

<b>Comments</b>	
-----------------	--

#### AT-9

<b>Related User Story</b>	#12 As a user of the condo management system, I would like to sign in to the condo management system to be able to use the features related to my user type (USP 4) sprint 1 user story
<b>Related System Tests</b>	Integration testing, Unit testing
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"> <li>1. A user clicks on “Create an Account” on the home page</li> <li>2. A user creates their accounts by entering their credentials</li> <li>3. A user can log in to their profile</li> </ol>
<b>Label</b>	Closed
<b>Comments</b>	Same as AT-5

#### AT-10

<b>Related User Story</b>	#14 As a condo management company, I want to be able to submit requests to the relevant employees, so that pressing issues can be resolved quickly. (3 USP) user story
<b>Related System Tests</b>	-
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"> <li>1. As a condo management company, click on “dashboard” from the dropdown menu</li> <li>2. Click on “employee list” from dashboard</li> <li>3. Assign requests by employee ID</li> </ol>
<b>Label</b>	Open

<b>Comments</b>	
-----------------	--

### AT-11

<b>Related User Story</b>	#18 As a public user, I need to enter a registration key from my condo management company so that i can register as a rental user, gaining access to relevant functionalities. (2 USP) user story
<b>Related System Tests</b>	Unit test
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"> <li>1. User navigates to their profile through the dropdown in the navbar</li> <li>2. User clicks on “Link Condo”</li> <li>3. User enters code and click “Link”</li> <li>4. The condo unit is now linked to them</li> </ol>
<b>Label</b>	Closed
<b>Comments</b>	

### AT-12

<b>Related User Story</b>	#20 As a condo management company, I want to set up different roles for employees responsible for the same property so that tasks can be efficiently distributed and managed. (2 USP) sprint 3 user story
<b>Related System Tests</b>	
<b>Acceptance Criteria</b>	
<b>Label</b>	Open
<b>Comments</b>	

**AT-13**

<b>Related User Story</b>	<b>#21 As a condo owner planning to move out, I need to be able to schedule elevator time for moving through the app, ensuring that I can secure the date and time without having to physically visit the management office or make phone calls. (3 USP) sprint 3 user story</b>
<b>Related System Tests</b>	
<b>Acceptance Criteria</b>	
<b>Label</b>	<b>Open</b>
<b>Comments</b>	

**AT-14**

<b>Related User Story</b>	<b>#22 As a condo management company with multiple properties, I want the ability to toggle between the dashboards of each property within the app, so I can easily manage and keep track of the status, requests, and financial information of all my units without having to log in separately for each one. (5 USP)</b>
<b>Related System Tests</b>	
<b>Acceptance Criteria</b>	
<b>Label</b>	<b>Open</b>
<b>Comments</b>	<b>App only</b>

**AT-15**

<b>Related User Story</b>	<b>#24 As a public user, I need to enter a registration key from my condo management company so that I can register as a condo owner, ensuring verified access to owner-specific features. (2 USP) user story</b>
<b>Related System Tests</b>	<b>Unit test</b>
<b>Acceptance Criteria</b>	<b>1. User navigates to their profile through the dropdown in the navbar 2. User clicks on “Link Condo” 3. User enters code and click “Link” 4. The condo unit is now linked to them</b>
<b>Label</b>	<b>Closed</b>
<b>Comments</b>	

**AT-16**

<b>Related User Story</b>	<b>#25 As a condo management company, I would like to be able to generate registration keys so that i can send them to users for their authentication (1 USP) user story</b>
<b>Related System Tests</b>	<b>-</b>
<b>Acceptance Criteria</b>	<b>-</b>
<b>Label</b>	<b>Closed</b>
<b>Comments</b>	<b>Duplicate</b>

**AT-17**

<b>Related User Story</b>	<b>#27 As a condo management company, I would like to be able to have a dashboard that displays all the functionalities available to me , so that i can easily navigate through them and have a general overview . (3 USP) sprint 2</b>
<b>Related System Tests</b>	<b>Unit test</b>
<b>Acceptance Criteria</b>	<b>1. As a condo company, click on “Dashboard” from the navbar 2. It should bring you to your dashboard</b>
<b>Label</b>	<b>Closed</b>
<b>Comments</b>	

**AT-18**

<b>Related User Story</b>	<b>#40 As a condo owner, I would like to view my requests that are assigned, created and finished (USP 3) sprint 4 user story</b>
<b>Related System Tests</b>	<b>Unit test</b>
<b>Acceptance Criteria</b>	<b>1. As a user, click on “Dashboard” 2. Click on “View Requests” 3. Assigned, created, and finished requests will show as a list</b>
<b>Label</b>	<b>Closed</b>
<b>Comments</b>	

**AT-19**

<b>Related User Story</b>	#41 As a condo management company, I would like to view all open requests that my condo owners have created to properly attend to them. (USP 3) sprint 4 user story
<b>Related System Tests</b>	-
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"><li>1. As a user, click on “Dashboard”</li><li>2. Click on “View requests”</li><li>3. View all the requests</li></ol>
<b>Label</b>	
<b>Comments</b>	Condo Manager side of AT-18

**AT-21**

<b>Related User Story</b>	#43 As a public user i would like to be able to pay my balance for my condo. (2 USP) user story
<b>Related System Tests</b>	-
<b>Acceptance Criteria</b>	-
<b>Label</b>	Closed
<b>Comments</b>	Automatic Financial System Not planned

**AT-22**

<b>Related User Story</b>	As a user, I want to be able to quickly navigate between my profile and dashboard using the navbar (2USP) user story
<b>Related System Tests</b>	Unit test
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"><li>1. Click on profile Icon on the navbar available throughout the website</li><li>2. Click on the dropdown</li><li>3. Click on “Profile”</li><li>4. Redirects to profile</li></ol>
<b>Label</b>	Closed
<b>Comments</b>	

**AT-23**

<b>Related User Story</b>	#54 As a condo management company, I want to be able to edit properties under our management to be able to view and manage the data related to them (4 USP) sprint 3 user story
<b>Related System Tests</b>	Unit Test
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"><li>1. As a condo management company, click on “Properties” on the dashboard</li><li>2. Select your property from the properties dashboard</li><li>3. Click on “Edit”</li><li>4. Edit the relevant information from the property</li><li>5. Click “Save”</li></ol>

	<b>6. Changes are updated</b>
<b>Label</b>	<b>Closed</b>
<b>Comments</b>	

#### AT-24

<b>Related User Story</b>	#68 As a public user, I would like to be able to have a landing page that displays all the functionalities available to me , so that i can easily navigate through them and have a general overview . (2 USP)
<b>Related System Tests</b>	-
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"> <li>1. As a user, click on “Dashboard” from the dropdown menu on the navbar</li> <li>2. Navigate to dashboard</li> </ol>
<b>Label</b>	<b>Closed</b>
<b>Comments</b>	

#### AT-26

<b>Related User Story</b>	#81 As a condo management company, i would like to see all units associated to my property so that i can manage them (USP 3) sprint 4 user story
<b>Related System Tests</b>	-
<b>Acceptance Criteria</b>	<ol style="list-style-type: none"> <li>1. As a condo management company, click on “properties” from your dashboard</li> <li>2. All properties owned will display. Click “View” on the property to view its units</li> <li>3. All the units of the property will now display</li> </ol>

<b>Label</b>	<b>Closed</b>
<b>Comments</b>	

## Testing Plan for Sprint 5

### Sprint 5 Unit tests

For sprint 5, we plan on finishing the tests for features implemented during Sprint 4, such as registration key generation, condo unit generation, the financial system, and the editing system.

### Sprint 5 Integration tests

For sprint 5, we plan on testing all backend-frontend communications for features implemented during Sprint 4, which include adding and editing a listing and adding employees. Furthermore, testing for features implemented during Sprint 4 is expected to be done, which will include features such as requesting a key, requesting a ticket, reserving amenities, etc.

### Sprint 5 System testing

As the project wraps up, Sprint 5 will focus on system testing and finishing the user acceptance testing for all features. There will be functional testing of requirements, database testing, and API testing. Tools such as Postman will be used for API testing.

# 9. Sprint Retrospectives

## 9.1 Sprint 1

### What went well:

- Frequent meetings
- Flexible teammates
- Task division & efficiency
- Task completion on time
- Communication between all team members inside and outside of meetings

### What we could make better:

- Stay consistent on meeting minutes
- Better communication on changes

From the beginning, our team decided that a cornerstone of our group would be communication. We have a weekly meeting scheduled Mondays, and at least one additional meeting per week scheduled more flexibly. We assigned tasks to each member, and created sub-teams to better organize the workload and communication. However, as tasks were undergone we checked in frequently to see which tasks needed more help, and reassigned people quickly to help maintain our timeline. This dynamic led to an efficient execution of tasks within time constraints. We also encouraged constant and consistent communication between members inside and outside of their sub-teams regarding issues or concerns linked to their tasks. This led to better results through collaboration.

The sprint was however not perfect and we learned of a few things we need to improve to make our team better. Primarily, we need to keep meeting minutes more effectively, and make them easier to read. We had many meetings and not every member of the team could attend each one, thus having more comprehensive meeting reviews would have benefitted the team immensely. A second point to focus on was in regards to communication. More specifically communicating changes during this very volatile part

of development. Ideas and decisions changed frequently in the beginning as we brainstormed and learned more about the system we would be designing. A few moments of confusion occurred as members were brought up to speed with new changes which slowed us down a tad. Better communication of changes will be worked on in further sprints.

## 9.2 Sprint 2

### **What went well**

- More of the team got involved with developing the app than vs sprint 1
- Communication is more consistent and frequent among teammate
- Clear separation between front and backend for the same task .

### **What we can work on and how/why**

- Better response time for PR reviews (we should create a disc. Channel dedicated to open PRs)

During sprint 1, documentation was quite important and thus much of the team was dedicated to working on that part. During this sprint however, we were able to get more of the team involved in app development and collaborating. There was also a more clear separation of front and back end for each task, to help clear ambiguity and generally increase efficiency. We have also worked hard to solve our communication issues from last sprint. We were able to make communication more consistent in our discord server, as well as through more thorough meeting minutes to keep people who cannot participate in meetings on track.

As development has taken front stage, an issue we came upon was PR reviews. We need to work on better response times for these reviews to keep a more efficient workflow. Our current solution involves creating a discord channel dedicated to PRs in which teammates can be tagged directly to see when they are needed to consult on a PR. As in our case, response times on discord are much faster than on github, this should help us.

## 9.3 Sprint 3

### **What went well**

- Backend team - worked well with each other to change the DB schema
- Good feedback and engagement with prof/stakeholder on things that needed to be clarified
- Improved response time for PRs

### **What we can work on and how/why**

- More communication on what everyone is specifically working on at any time so we can better track progress
- More cumulative communication between teams (frontend - backend, backend - app dev)
- Better code management practices as many backend updates had breaking changes that took a while to go back and fix on front-end & app.
- Being more active in github issues (assigning ourselves to tasks, and having discussions there)

This sprint was shortened by the extension of the previous one but nonetheless, we were able to get quite a bit of work done. On the positive end, we were able to improve many things and notice some positives:

- As the backend team was expanded this sprint to further push development, all members worked well together and kept constant communication. This led to efficient work, especially in regards to schema changes that had to be done quickly.
- We tried to up our communication with the professor who represents an important stakeholder. This helped us to plan ahead more clearly and set long term expectations regarding the future of the project.
- Last sprint, we had trouble getting pull requests approved in a timely fashion but this has been improved as planned. This helped immensely in reducing downtime.

As teams were expanded however, and development work took the front seat, we saw a few issues we will work harder to address pop up:

- We only held two meetings this sprint and communication was down due to busy schedules and personal matters. Because of this, we were not able to track individual progress as effectively, and thus were less efficient. We will work to get back on a regular meeting schedule to remedy this.
- As I said, teams expanded in scope this sprint and although inner team communication was good, inter team communication could be improved. This led to some conflicts between front and back end for example. Once again, a regular meeting schedule would help, but an additional meeting between key members in each team would help resolve this issue.
- Code management has been a sore spot for us and although we are working to catch back up, we are not quite there yet. In light of this, some important backend changes broke parts of the front-end and it took longer than it could have to resolve them. With better managed code, this could have been achieved more efficiently. Thus, the solution is to take better care in following coding conventions to make our code as understandable, retraceable and structured as possible.
- An issue we came across was that people would work on user stories and github issues without necessarily assigning themselves issues in Github. It would improve code management quality and better organize the team.

## 9.4 Sprint 4

### **What went well**

- Coordination of mini tasks
- Cleaner code with better documentation
- More features were implemented
- More user stories were completed than others sprints

### **What we can work on and how/why it happened**

- Doing better preemptive exploration of tasks, namely focusing on prerequisites and prioritization.
- Prioritizing features across web/app. We are trying to implement as much as we can to cover all aspects of requirements but some things aren't specified in requirements although it may seem as a no brainer to have those features included.
- Reinforcing soft deadlines
- More frequent meetings

This was the last sprint before finals and all of our collective assignments and term projects were due around the same time. This is evidenced by the extension of this Sprint and although it was arduous, good progress was made. We were also able to pick up the pace in a meaningful way.

- As was mentioned last sprint, as a large team with few meetings, coordination of tasks and whose responsibility they were was an issue. This sprint however, we saw a markedly better performance in that respect. Meetings were more focused and all members were asked to participate more actively to get a more detailed, nuanced idea of progress. This also helped us coordinate because related tasks/features could be discussed more between the people responsible for them, and opinions or suggestions could be generated by the rest of the group for solutions.
- A simple, yet meaningful initiative to improve documentation after comments by the TA about our code management were discussed among the team.

Reinforcing its importance and presenting templates helped keep the team consistent, active in documentation, and respecting coding standards.

- Our team had been lacking for a while in terms of development progress and that has been a constant issue, but this sprint we have really hit our stride. We were able to pick up quite a bit of slack and progress immensely in terms of features. We hope to keep up this momentum and believe that we will be able to finish our features in time.
- This sprint has been our most successful in terms of user stories. To elaborate a little from the point above, this has been the sprint where the most user stories were closed and acceptance tests performed. This is again a large improvement, and we hope to keep it up to finish our last few.

Although we made meaningful and substantial progress, we are still behind the expected timeline. This has led to the resurgence of old issues we worked on in the past.

- As we have consistently received feedback that our prototype is behind schedule, we have been rushing to meet expectations. This has led to issues regarding our preemptive/exporative work for tasks ahead of time. This has led to tasks taking longer than expected, or not marked as prerequisites for other tasks, leading to a slower timeline. This happened because, again, we were lacking in our preemptive work on the tasks, namely looking at prerequisites and prioritization more in depth. The simple solution will be to be more diligent in these explorations during meetings before assigning the tasks, as well as the order to complete them.
- By rushing, and not doing enough research into the tasks as mentioned above, we are also running into the issue of unexpected work appearing. Many of the small tasks/additions seem very obvious in retrospect but were not accounted for when planning for time. This has also contributed to a slower turnaround time. Again, this happened because of a lack of preparation and the solution involves spending more time as a team on this.
- As was the issue in early sprints when the team was still lacking cohesion and organization, work is being done in sporadic and disorganized bursts. As a team, soft deadlines are implemented to better breakdown our sprints and lead to simpler progress milestones to reach. Although we've been setting them, we have been too lax and these soft deadlines are not being respected as much as we would like. Thus, we need to start reinforcing them, and reminding members of their importance to smooth out the development process.
- Finally, our last issue is a lack of meetings. In the beginning of our sprints, we were only doing 1 meeting a week and found that this was lacking. Our solution

in the past was to ramp that up to 2 per week and we saw large improvements in productivity. Unfortunately, as was mentioned above, we've all been extremely busy and fell back to bad habits with 1 meeting a week. This has not completely destroyed our productivity as can be seen by the amount of progress we made, but it is still an issue. The solution here being to encourage our group to hold more meetings and to prepare for them with some progress reports on their work. This will help alleviate some of our problems above, as well as add transparency and efficiency to the team.

## 9.5 Sprint 5

### **What went well:**

- Even with exams, light work was done during study week and thus not everything was left to the last second.
- More communication with the team, more meetings
- Great cohesiveness and code annotations helped a lot when trying to bring features from the web to the companion app.

### **What we can work on and how/why it happened:**

- Setting up the hosted web app to be seamlessly integrated with the newly developed features created some issues. Happened since we didn't set global variables from the get go so requires a lot of back checks to change respective communication links.

This sprint was a daunting task as it overlapped with finals and many of us are taking multiple heavy classes this semester. With that in mind, like any sprint there were ups and downs, but we got through it well. On the positive end:

- Much of the work was done either before or after exams depending on each team member's exam schedule. Thankfully, everyone was quite cooperative and did their fair share. For those with later exams, more work was done during the study week, but those with exams sooner were also able to do some light work

before too. This set up the team well to finish the rest after exams and not feel too pressed last second.

- As mentioned earlier, communication and cooperation were at the forefront this sprint because of the time crunch. To achieve this, we brought our meeting schedule back up to par and made sure to communicate more over Discord, our main communication hub. This allowed for more efficient work at a time where it was the most necessary.
- Improved code documentation and adherence to coding practices as of late made the transition of features from the web app to the companion mobile app much easier. As we are using the React framework, code reusability

Like any other sprint or project, there are negative sides. Thankfully this sprint only one thing really stood out as an issue, but it was a noticeable one:

- Our web app was initially designed to run on our local machines while being linked to the hosted database. This allowed for seamless development for our team but when we had to switch to online hosting, issues arose. To remedy this, much work had to be done to change all the proper communication links across the application to host correctly online. This led to some minor bugs which were addressed quickly, but more importantly it took time away from new features and development. In retrospect, the solution is simple: hosting the web application earlier and planning ahead by using global variables. By hosting earlier, less work would be needed, as well as that work happening at a less critical time in the semester.

# 10 UI Prototypes

## 10.1 Sprint 2 UI Prototypes

**Note: We did not fully understand what was expected of us in the first sprint, thus this was what we submitted in a file. Future sprints were much better in this regard.**

Refer to this figma link:

<https://www.figma.com/file/GX1otYGVRcio8utNvIPXga/EstateFlow?type=design&node-id=0-1&mode=design&t=BXbz1gb3RR303Wm6-0>

## 10.2 Sprint 3 UI Prototypes

**Note: We again did not fully understand what was expected of us and chose to put the UI prototypes in the [Sprint 3 Release Plan](#). They can be found below the main table**

## 10.3 Sprint 4 UI Prototypes

**#9**

**As** a condo management company  
**I want** to be able to enter detailed information for each condo unit  
**So that** condo users and renters can have access to their units' information

Log in


**EstateFlow**

Home Features Why EstateFlow Pricing Contact [Login](#)

**Login**

I am a: Company ▾

Email Address\*

Password\*

[Don't have an account? Register](#)


**EstateFlow**

Home Features Why EstateFlow Pricing Contact [Welcome, CoolCompany](#) ▾


+

**36 Lee drive, HBB 3M6**

[View Employee List](#) [Units](#) [Requests](#)

Units


**EstateFlow**

Home Features Why EstateFlow Pricing Contact [Welcome, CoolCompany](#) ▾

**<<PROPERTY ADDRESS>> UNITS**



**UNIT 123**

2 parking spaces  
1000 square feet  
1 locker  
Owner: Not assigned

[Edit](#)

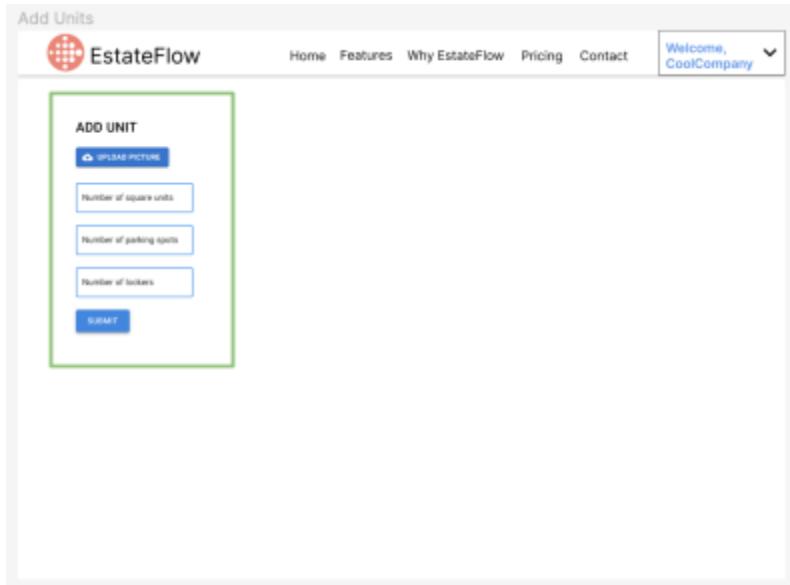


**UNIT 124**

2 parking spaces  
1000 square feet  
1 locker  
Owner: Sarah Abellard

[Edit](#)

[New](#)



Steps:

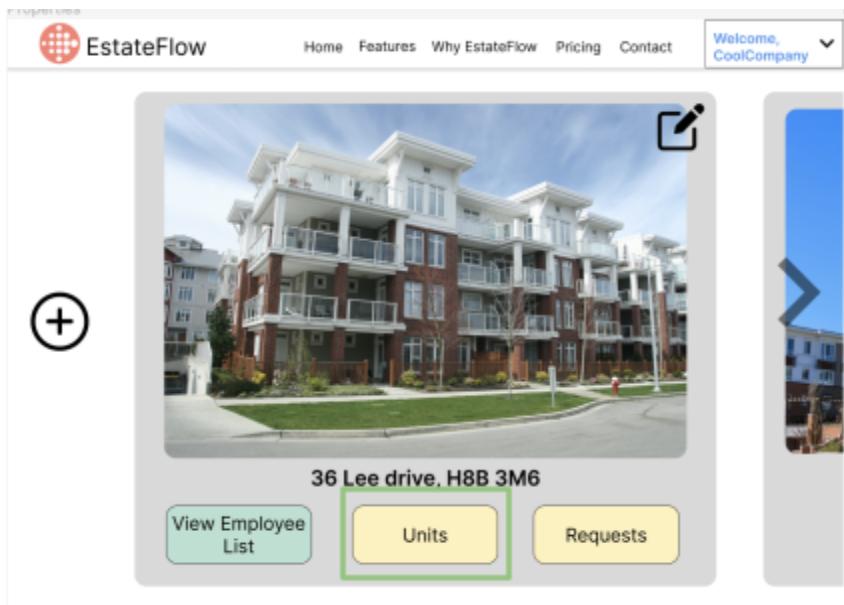
- 1- Condo management user signs in
- 2- Goes on property dashboard
- 3- Selects a property
- 4- Click on Add unit button
- 5- Fills the form with information on the unit
- 6- Sends form

**[#3]:**

**[As a condo management company**

**I want** to be able to send out registration keys to users

**So that** their accounts can be linked to their housing]



Property dashboard

Units

EstateFlow

Home Features Why EstateFlow Pricing Contact Welcome, CoolCompany

<<PROPERTY ADDRESS>> UNITS

Unit 123  
2 parking spaces  
1000 square feet  
1 bedroom  
Owner: Not assigned

Unit 124  
2 parking spaces  
1000 square feet  
1 bedroom  
Owner: Sarah Marshall

[Edit](#)

EstateFlow

Home Features Why EstateFlow Pricing Contact Welcome, CoolCompany

EDIT UNIT

2014 Forest Hill, Unit 123

Number of parking spaces  
2

Number of bedrooms  
1

Number of square feet  
1000

owner email  
sarah123@yahoo.ca

[SEND REGISTRATION KEY](#)

[SUBMIT](#)

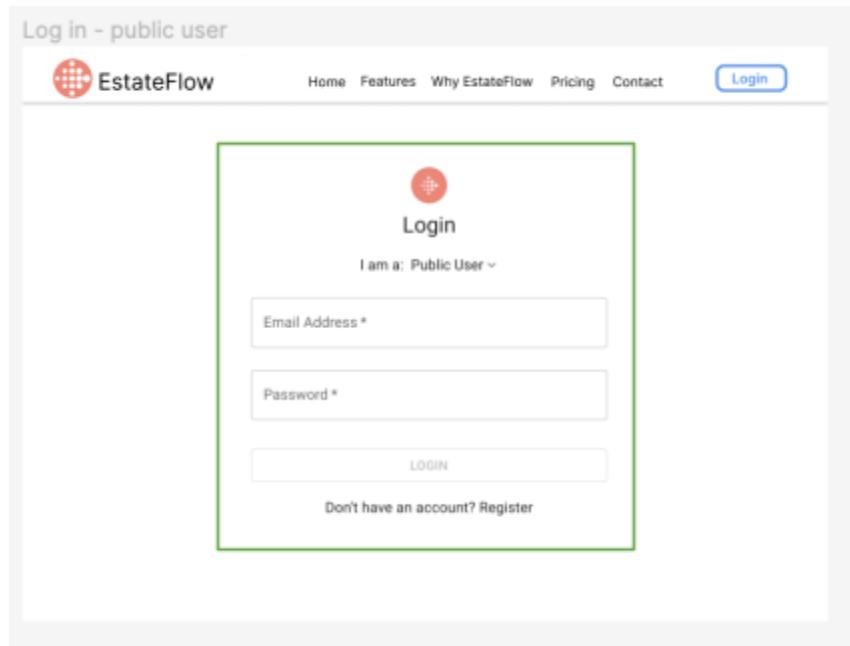
Steps:

- 1- Sign in as a condo management company
- 2- Navigate to property dashboard
- 3- Select property and click on units
- 4- Go on edit unit of the unit you would like to edit
- 5- Fill in the owner email
- 6- Click on send registration key
- 7- Submit form

**#18**

**As** a public user  
**I want** to be able to enter a registration key provided by the condo management company  
**So that** I can gain access to relevant tenant functionalities

Log in - public user



The screenshot shows the EstateFlow login page. At the top, there's a navigation bar with the EstateFlow logo, a search bar, and links for Home, Features, Why EstateFlow, Pricing, Contact, and Login. The main area is titled "Log in - public user". It features a "Login" button with a red circular icon above it. Below the button is a dropdown menu labeled "I am a: Public User". There are two input fields: "Email Address \*" and "Password \*". Below these fields is a "LOGIN" button. At the bottom of the form, there's a link "Don't have an account? Register". A green rectangular box highlights the entire login form area.



Steps:

- 1- Log in as public user
- 2- Navigate to registration key page

#5

**As** a condo owner or renter  
**I want** to have a good view (dashboard) of my property units  
**So that** they can be managed and I can keep track of issues

## Log in - public user



EstateFlow

Home Features Why EstateFlow Pricing Contact

Login



Login

I am a: Public User ▾

Email Address \*

Password \*

LOGIN

Don't have an account? Register

## Dashboard - User



Home Features Why EstateFlow Pricing Contact

Welcome, John ▾



36 Lee drive, H8B 3M6, Unit 201

[View Requests](#)

[Make request](#)

[Financial Information](#)

Steps:

- 1- Log in as public user
- 2- Navigate to property dashboard

#10

**As** a condo management company  
**I want** to be able to enter a condo fee per square foot, per parking lot and per locker  
**So that** unit owners can view costs and estimates

Log in

EstateFlow

Home Features Why EstateFlow Pricing Contact Login

Login

I am a: Company -

Email Address \*

Password \*

LOGIN

Don't have an account? Register

Properties

EstateFlow

Welcome, CoolCompany

+



+

36 Lee drive, H8B 3M6

View Employee List Units Requests



Edit listing page

EstateFlow Home Features Why EstateFlow Pricing Contact Welcome, CoolCompany

**Photo album**

Address: 36 Lee drive  
Postal Code: H8B 3M6  
Total Units: 12  
Parking spaces: 16

Amenities: 

- Indoor pool
- Includes indoor parking
- Newly renovated (2020)

Description: Beautiful newly renovated condo, in a great neighborhood. Schools nearby, great for new families. Includes an indoor pool and underground parking. Feel free to contact us for any inquiries.  
Phone number: 514-565-5859  
Email: Inquiries@Coolcompany.com

Price per square feet  
Price per parking space  
Price per locker

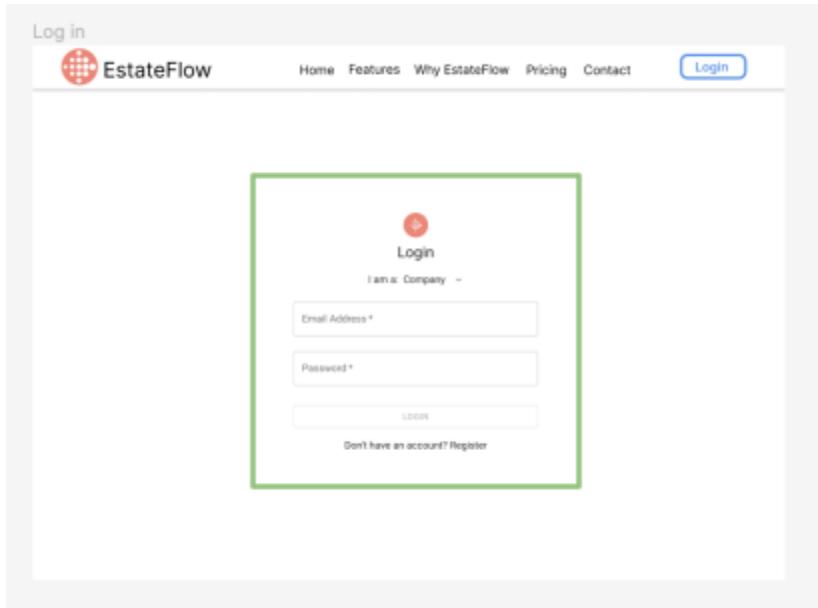
**Submit**

Steps:

- 1- Log in as a condo management company
- 2- Navigate to property dashboard
- 3- Click on Add property or edit property
- 4- Modify form on appropriate fields (price per square feet, parking space, locker)
- 5- Submit

**#11**

**As** a condo management company  
**I want** to be able to enter the cost of each operation for the financial system  
**So that** annual financial reports can be generated  
[series of images producing a successful outcome]



Dashboard

EstateFlow

Home Features Why EstateFlow Pricing Contact Welcome, CoolCompany

The dashboard has a header with the EstateFlow logo and navigation links. A "Welcome, Condo Management User!" message is displayed over a cityscape background. On the left is a sidebar titled "Navigation" with links to Property Dashboard, Manage Condo Files, Financial System, Reservation System, Employee, and Requests. The main area shows four cards: "Property Dashboard" (with a link to "Condo properties with employee list and request management features"), "Manage Condo Files" (with a link to "Here you can manage and access condo-related documents and files"), "Financial System" (with a link to "Overview of financial operations, such as billing, and payments"), and "Reservation System" (with a link to "Residents can book amenities and services available in the condo").

Welcome, Condo Management User!

Navigation

Property Dashboard

Manage Condo Files

Financial System

Reservation System

Employee

Requests

Property Dashboard

Condo properties with employee list and request management features >

Manage Condo Files

Here you can manage and access condo-related documents and files >

Financial System

Overview of financial operations, such as billing, and payments >

Reservation System

Residents can book amenities and services available in the condo >

Financial system

The screenshot shows the EstateFlow financial system interface. At the top, there's a navigation bar with links for Home, Features, Why EstateFlow, Pricing, Contact, and a welcome message for 'CoolCompany'. Below the navigation is a section titled 'FINANCIAL SYSTEM'.

**Outgoing Invoices**

Property Address	Unit	Amount	Status	Due Date
123 Main St	A101	100	Unpaid	2024-03-01
456 Elm St	B202	150	Paid	2024-03-29
789 Oak St	C303	120	Unpaid	2024-04-02

Rows per page: 100 < 1-3 of 3 >

**ADD BILL**

**Operational Costs**

Property Address	Amount	Description	Date
123 Main St	100	Operational cost 1	2024-03-17
456 Elm St	150	Operational cost 2	2024-03-18
789 Oak St	120	Operational cost 3	2024-03-19

Rows per page: 100 < 1-3 of 3 >

**ADD OPERATIONAL COST**

**GENERATE REPORT**

Add operational cost

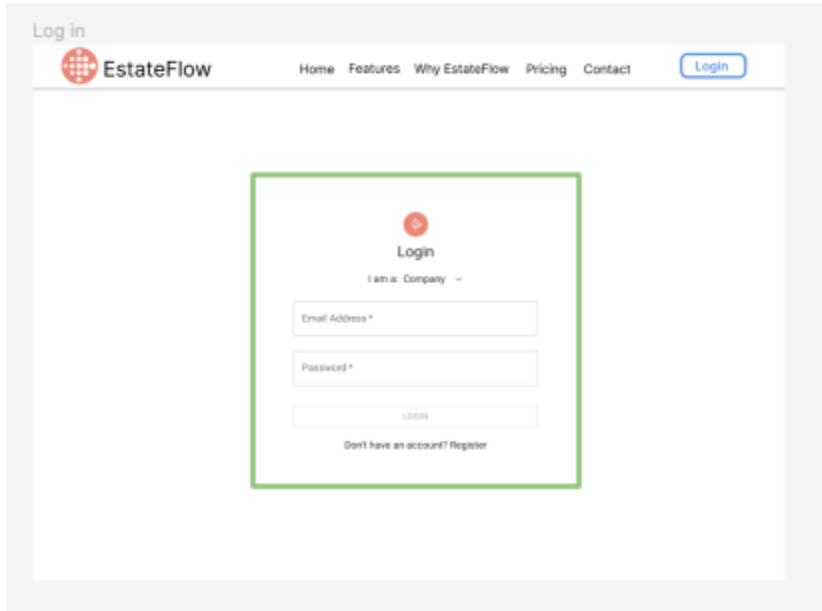
The screenshot shows a modal window titled 'Add operational cost'. It contains a form with fields for 'Property Address', 'Amount', 'Description', and 'Due Date' (a date input field set to 'yyyy-mm-dd'). There is also a 'Submit' button at the bottom of the form.

Steps:

- 1- Sign in as management company
- 2- Go to financial system
- 3- Click on add operational cost
- 4 - Fill form
- 5- Submit

**#40**

**As a condo owner  
I want to be able to view my requests that are assigned, open, or completed  
So that I can keep track of issue resolution**



The screenshot shows the EstateFlow properties page for a specific property. At the top, it says "Properties" and "Welcome, CoolCompany". The main content features a large image of a modern multi-story apartment complex with a "Edit" icon in the top right corner. Below the image is the address "36 Lee drive, H8B 3M6". At the bottom, there are three buttons: "View Employee List" (green), "Units" (yellow), and "Requests" (yellow, with a green rectangular border around it). To the left of the main content, there's a circular button with a plus sign (+). To the right, there's a vertical blue bar with a black arrow pointing right.

Open Requests - Management

EstateFlow

Home Features Why EstateFlow Pricing Contact Welcome, John

Request ID	Description	Date	Time	Status
1. Moving Out (Elevator Request)	-	18/03/2024	-	Awaiting Response
2. Deficiency in common areas	-	No more snacks in the common area, we...	-	Task assigned
3. Reporting a violation	-	Unit 4 has been practicing tap dancing during...	-	Resolved

ADD REQUEST

Steps:

- 1- Sign in as condo management company
- 2- Go to property dashboard
- 3-Go to requests to view properties

## 10.4 Sprint 5 UI Prototypes

#15

**As** As a public user  
**I want** I would like to be able to reserve a common space on my calendar

**So that** I can have guaranteed access to them when the time come.

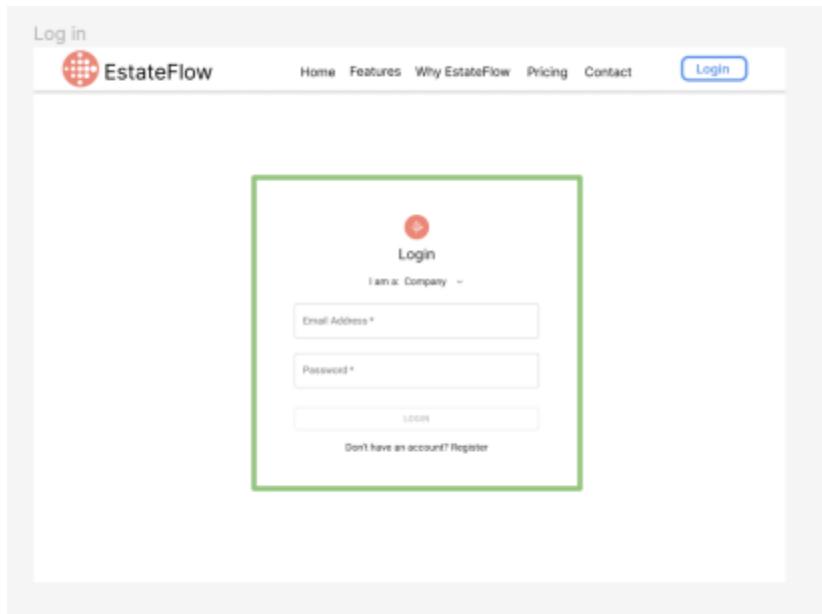
The screenshot shows a web-based reservation system for a property at 36 Lee drive, H8B 3M6. The top navigation bar includes links for Home, Features, Why EstateFlow, Pricing, Contact, and a welcome message for 'Welcome, John'. Below the navigation is a large image of a modern multi-unit residential building. The address '36 Lee drive, H8B 3M6' is displayed. A search bar with fields for Date, From, and To, along with a calendar and clock icon, is present. A dropdown menu for 'Reservation Type' is set to 'Event Hall'. A note specifies a 'Max Duration 3 hours'. A section titled 'Rules:' lists several guidelines: Occupancy Limit (Maximum 50 guests allowed), Noise Control (Adhere to quiet hours from [insert start time] to [insert end time]), Prohibited Activities (No illegal actions, gambling, or hazardous materials), Cleanup (Leave the space as found; dispose of all waste properly), and Food (All food must be prepared off-site; adhere to food safety regulations). A blue 'Submit' button is located at the bottom of the form.

Steps:

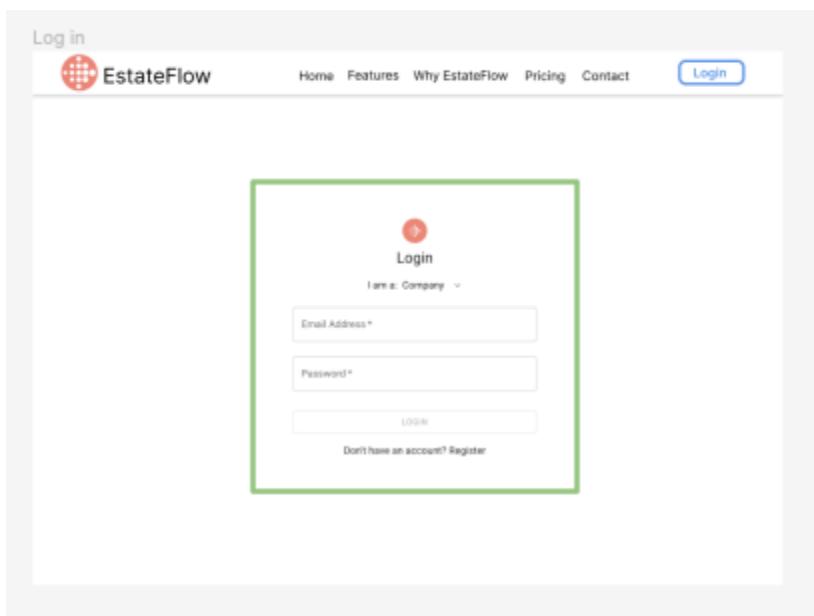
- 1- Log in as public user
- 2- Navigate to Dashboard page for user
- 3- Click on Make reservation
- 4 - Fill form
- 5- Submit

**#19**

**As** As a condo management company  
**I want** I would like to be able to set up common facilities to a simplified reservation system  
**So that** condo owners/renters can reserve the spaces.



**#9** As a condo management company, I want to be able to enter detailed information for each condo unit so condo users and renters can have access to the information for their respective units





Home Features Why EstateFlow Pricing Contact

Welcome,  
CoolCompany ▾

#### EDIT UNIT



2014 forest hill, Unit 123

Number of parking spaces

2

Number of lockers

1

Number of square feet

1000

Owner email

sarah123@yoohoo.ca

**SEND REGISTRATION KEY**

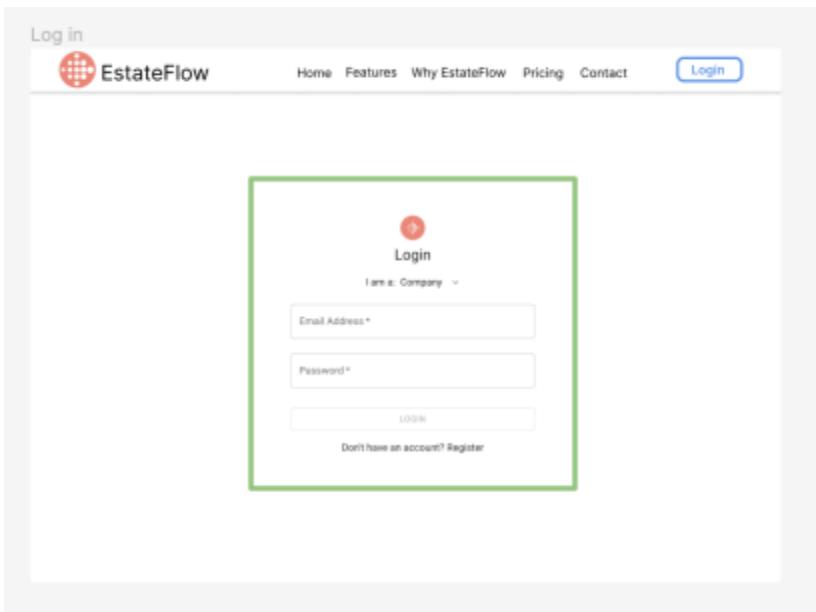
**SUBMIT**

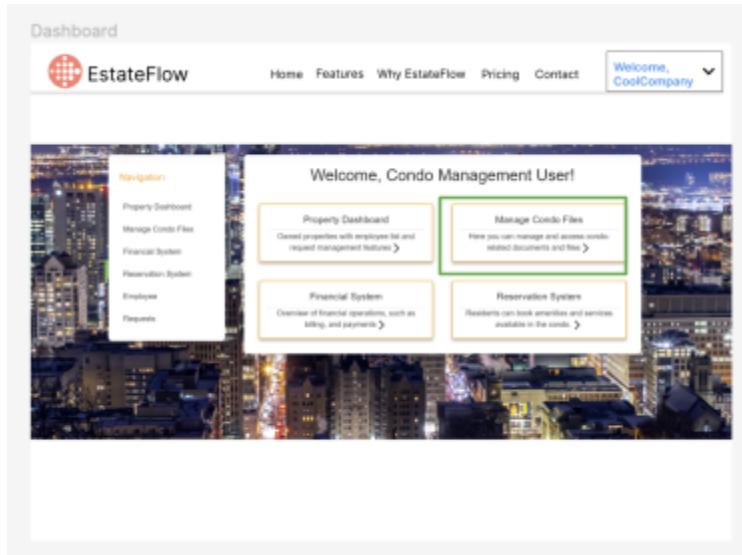
Date:

1- Sign in as condo management company

2- Add condo unit

**#8** As a condo management company, I want to be able to upload files for each property to be viewed by all condo owners of the property





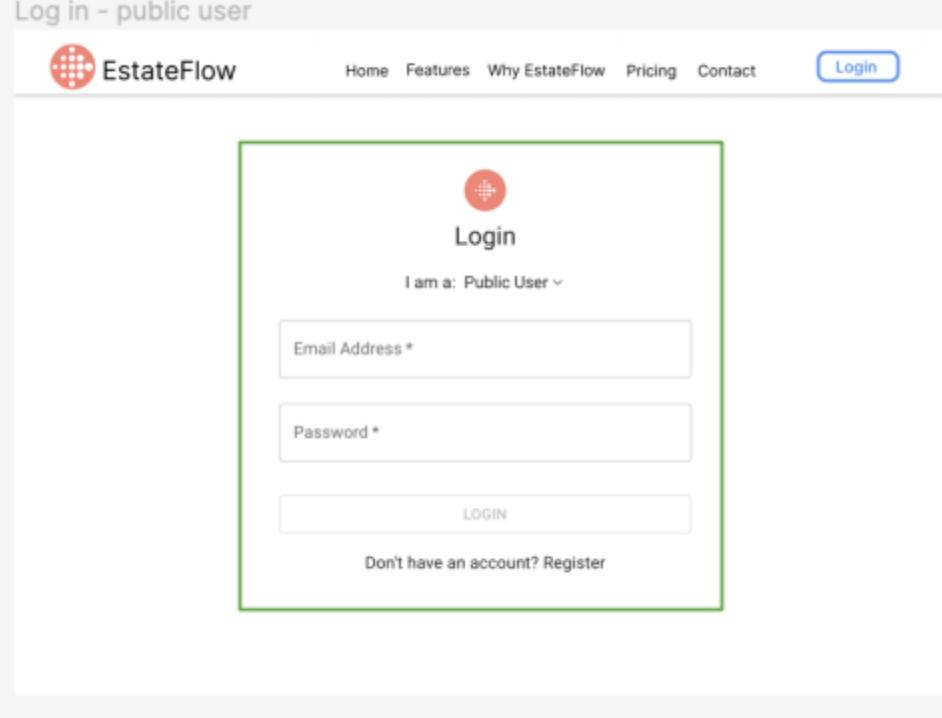
Reservation Company view:

The screenshot shows the EstateFlow interface for a reservation company. At the top, there's a navigation bar with links for Home, Features, Why EstateFlow, Pricing, Contact, and a dropdown for 'Welcome, John'. Below the navigation is a form titled 'Upload File to Property'. It has three main sections: a dropdown menu labeled 'Property' with a green border, a file input field labeled 'Choose File' with 'No file chosen' below it, and a blue 'UPLOAD FILE' button. The entire form is enclosed in a green border.

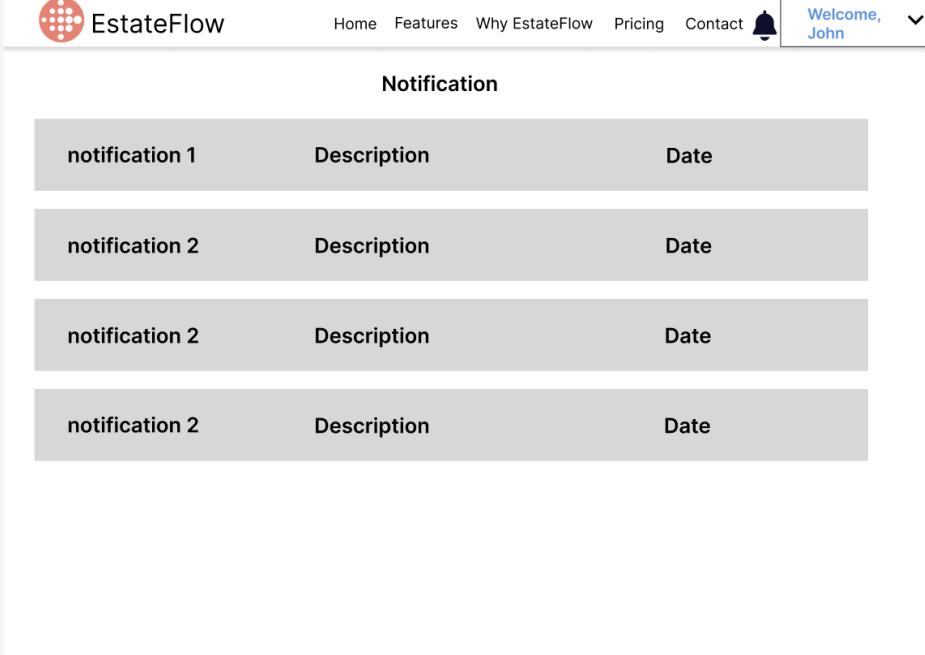
- 1- Sign in as a condo management company
- 2- Click on manage condo files
- 3- Click on property to choose the property
- 4- Choose file from directory
- 5- Click on upload file

**#13** As a public user, I want to be able to see any relevant notifications for me, so that I can be aware of things happening.

Log in - public user



The screenshot shows the EstateFlow login page. At the top, there is a navigation bar with links for Home, Features, Why EstateFlow, Pricing, Contact, and a blue 'Login' button. Below the navigation bar is a green-bordered login form. The form contains a red circular logo with a white cross-like pattern at the top, followed by the word 'Login'. Below this is a dropdown menu labeled 'I am a: Public User'. The form includes two input fields: 'Email Address \*' and 'Password \*', both with placeholder text. Below these fields is a 'LOGIN' button. At the bottom of the form, there is a link 'Don't have an account? Register'.

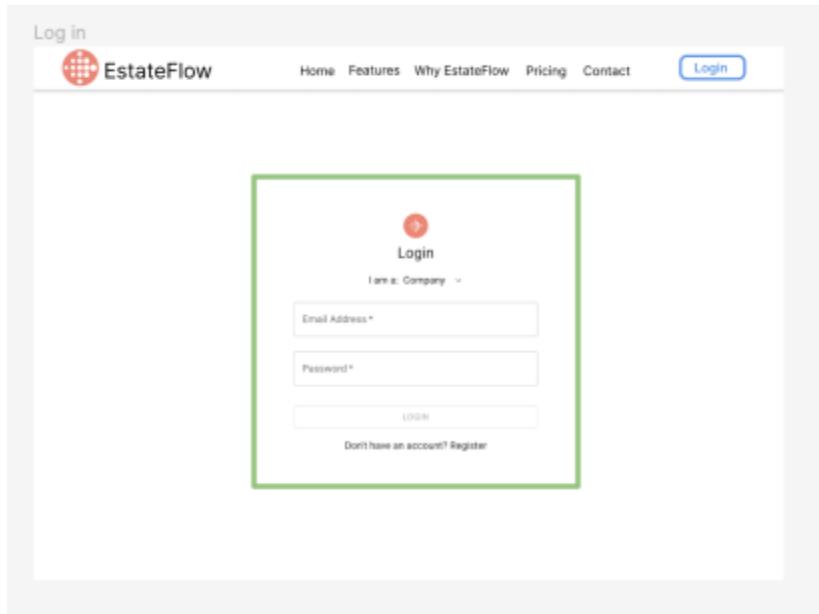


The screenshot shows the EstateFlow notification page. At the top, there is a navigation bar with links for Home, Features, Why EstateFlow, Pricing, Contact, and a 'Welcome, John' message next to a bell icon. Below the navigation bar is a section titled 'Notification' containing a table with four rows. The table has three columns: 'notification 1', 'Description', and 'Date'. Each row represents a notification entry.

notification 1	Description	Date
notification 2	Description	Date
notification 2	Description	Date
notification 2	Description	Date

- 1- Login as public user
- 2- Click on notification bell
- 3- View notifications

**#19** As a condo management company, I would like to be able to set up common facilities to a simplified reservation system so condo owners/renters can reserve the spaces



The screenshot shows the EstateFlow dashboard. At the top, it features a navigation bar with the EstateFlow logo, Home, Features, Why EstateFlow, Pricing, Contact, and a dropdown menu showing 'Welcome, CoolCompany'. Below the navigation is a main content area with a cityscape background. On the left, a sidebar titled 'Navigation' lists: Property Dashboard, Manage Condo Files, Financial System, Reservation System (which is highlighted with a green border), Employee, and Requests. The central area has a 'Welcome, Condo Management User!' message and four cards: 'Property Dashboard' (Shows properties with employee list and request management features), 'Manage Condo Files' (Shows where you can manage and access condo-related documents and fees), 'Financial System' (Shows overview of financial operations such as billing and payments), and 'Reservation System' (Shows where residents can book amenities and services available in the condo). A small 'CoolCompany' watermark is visible in the bottom right corner of the dashboard area.

**ADD AMENITY**

 [UPLOAD PICTURE](#)

[SUBMIT](#)

- 1- Sign in as condo management company
- 2- Click on reservation system
- 3- Fill in form

## APPENDIX A: Sources

Fernando, Jason. "What Are Stakeholders: Definition, Types, and Examples." *Investopedia*, Investopedia, [www.investopedia.com/terms/s/stakeholder.asp](http://www.investopedia.com/terms/s/stakeholder.asp). Accessed 7 Feb. 2024.

"Industry-Leading Property Management Software." *BuildingLink*, [www.buildinglink.io/en-ca/](http://www.buildinglink.io/en-ca/). Accessed 7 Feb. 2024.

"Optimize Property Management with One Platform." *Entrata*, [go.entrata.com/watch-demo.html?utm\\_medium=cpc&utm\\_source=capterra](http://go.entrata.com/watch-demo.html?utm_medium=cpc&utm_source=capterra). Accessed 7 Feb. 2024.

*Practical Law UK*, [uk.practicallaw.thomsonreuters.com/Browse/Home/PracticalLaw?contextData=\(sc.Default\)](http://uk.practicallaw.thomsonreuters.com/Browse/Home/PracticalLaw?contextData=(sc.Default)). Accessed 7 Feb. 2024.

Zoltan, Adina. "What Is a Technical Project Manager?: Toptal®." *Toptal Projects Blog*, Toptal, 14 Apr. 2021, [www.toptal.com/project-managers/technical/what-is-a-technical-project-manager](http://www.toptal.com/project-managers/technical/what-is-a-technical-project-manager).