

Sprint 5

Vision Statement

Software Product Vision

1. Introduction

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

The introduction of the Vision document provides an overview of the entire document. It includes the purpose, and references of this Vision document.

1.1 References

https://docs.google.com/document/d/1W5KY6c0_mEQ_bhYRdYB1SbMT_YfOyvo7/edit?usp=sharing&ouid=11043965938531188893&rtpof=true&sd=true

2. Positioning

2.1. Problem Statement

The problem of	Managing a given set of condos within a property
affects	The Condo Management Companies, the Condo Owners and Condo Renters
the impact of which is	(Confusion) Communication and ease of access to relevant information between the Condo Management Companies, the Condo Owners and Condo Renters
a successful solution would be	A Condo Management App to simplify communication and payment between Condo Management Companies, Condo Owners and Condo Renters

2.2. Product Vision Statement

For	Condo Management Companies, the Condo Owners and Condo Renters
Who	Want an easy way to communicate or make payment

The Condo Management System	is a software product
That	Simplifies owning, renting and managing Condos
Unlike	[primary competitive alternative listed in section 3.5]
Our product	Contains a simplified financial system for payment and is more user friendly(intuitive)

3. Stakeholder and User Descriptions

3.1. Stakeholder Summary

Name	Description	Responsibilities
Project manager	Oversees the development of the system.	<ul style="list-style-type: none"> • Make estimations about project: Cost, Time and Effort • Monitor the projects progress • Ensure that project meets requirements
Product owner	They provide the directions about the system's purpose based on the users needs and business goals.	<ul style="list-style-type: none"> • Ensure that the app's features and vision are well defined. • Ensures that end product meets end users needs • Communicate with both end users and stakeholders • Translate feedback into features
Development team	Team that will code and program the application.	<ul style="list-style-type: none"> • Ensure that the requirements set by product owners are turned into functional products
Marketing team	Team that will come up with a strategy to promote the app on the market.	<ul style="list-style-type: none"> • Ensure that there will be a market demand for the proposed features • Study the market and target audience to optimize success of app
Investors	People that are funding the creation of this application.	<ul style="list-style-type: none"> • Fund the project in exchange of a return on investments
IT department	People that maintain the software and hardware of the application.	<ul style="list-style-type: none"> • Ensures that the system will be maintainable

3.2. User Summary

Name	Description	Responsibilities	Stakeholder
Condo Management Companies	Companies that own condos and that are trying to put them on sale or rent.	<ul style="list-style-type: none"> • Create profiles for their condos and manage those profiles • Have a reservation system for the facilities in their condos • Set up roles for their employees 	Their interest will be represented by the product owners and marketing team.
Future and actual eCondo owners	<p>People that currently own a single or multiple units of a condo.</p> <p>People that will use the app to look into buying a unit of a condo or that already own units.</p>	<ul style="list-style-type: none"> • Manage their properties • Create and manage a user profile • Keep registration key secure 	Their interest will be represented by the product owners
Future and actual eCondo renters	<p>People that are currently renting out a unit of a condo.</p> <p>People that will use the app to look into renting a unit of a condo or that are already renting units.</p>	<ul style="list-style-type: none"> • Manage their properties • Create and manage a user profile • Keep registration key secure 	Their interest will be represented by the product owners
Employees of the condo management companies	They will use the app to take on roles (manager, support staff, etc) and assist condo unit owners.	<ul style="list-style-type: none"> • View the requests submitted by the condo owners and assist them • Manage condo : Keep it clean and safe 	Their interests will be represented by the product owners and IT department.
Real Estate Broker	They will use the app to find and represent sellers (Condo Management Company) or buyers (Potential buyer or renter).	<ul style="list-style-type: none"> • Intermediate between seller and buyer • Facilitate transactions 	Their interest will be represented by the product owners
Public users	They will use the app to look into buying or renting a unit of a condo.	<ul style="list-style-type: none"> • Create and manage a user profile • Use app in accordance with its guidelines 	Their interest will be represented by the product owners and marketing team.

3.3. User Environment

- Users will need browser enabled device
- App needs to integrate Gmail so that users can log in using their gmail account
- Users need to have an adequate device such as a smartphone, tablet, laptop or desktop device to access the app and its features. They need to be able to open documents, pictures and videos sent by the app and other users.
- Users will be required to have the proper updates to be able to run the app on their smartphone or tablet
- Users will require internet connectivity to access all the features offered by the app. In case they don't have internet connectivity, they'll have access to emergency features and content they have saved for offline viewing. A secure internet connection is suggested to protect the personal information on the user's profile.
- App needs to integrate a secure financial system to allow safe transactions (ex: auto deposit, down payments, reimbursements, etc).
- App will require a team (4-10 people) to maintain it daily and make sure that there are no breaches of securities since it contains a lot of personal information.
- App needs to have a built in calculator that can calculate the price per unit and the remaining balance to be paid for a unit. This feature can be implemented by a team of 2-4 people.
- App needs to track the financial status of each condo management company, users and renters and provide them a financial report detailing all expenses, balances, etc. The financial report can be provided monthly and on request. This feature can be implemented by a team of 2 to 4 people.
- The payment portal of the app will require connection to the users banks websites.

3.4. Key Stakeholder or User Needs

Need	Priority	Concerns	Current Solution	Proposed Solutions
Create user profiles	High	Connect UI to database and store information for future logins	Connect user input from login page to database using MongoDB	None

Property dashboard	Medium	Intuitive UI, not cluttered but display enough information	None	Create a Prototype UI and obtain feedback from users
Create profiles for properties	High	Ensure there is no overlap in important information for condo files of each properties such as parking spot, locker, etc	None	Implement testing to check for overlap of information
Simplified financial system	Medium	Minute-to-minute update of condo fees for each unit	None	Implement observers to update changes to condo's balance
Simplified reservation system	Medium	Connect reservation system to make reservations for common facilities	None	Create a reservation calendar for common facilities

3.5. Alternatives and Competition

Competition	Advantages	Disadvantages
Buildium	Comprehensive functionality User-friendly interface	Expensive Steep user learning curve
Condo Control Central	Visitor management system Announcement features	Limited accounting features Website issues
AppFolio	Robust financial features	Lacking specific condo management features
TOPS[ONE]	Full suite of features Customizable reporting and analytics	Less intuitive UI Steep learning curve
Yardi Voyager	Scalable for large properties portfolio	Many potentially unnecessary features Expensive

4. Product Overview

4.1. Product perspective

The Condo Management System is a self-contained software system used in service of the condo owners, renters and management company.

The Condo Management System consists of a login page, a database to store user and condo data and different UIs for a variety of activities such as payment, requests, etc.

The Condo Management System can be used by condo owners and renters to get notified of payments, make requests to the management company, store various related information and communicate with other condo owners and renters.

The Condo Management system is comprised of several subsystems including:

- User Management subsystem
- Property Management subsystem
- Financial subsystem
- Reservation subsystem
- Authentication subsystem
- Database subsystem

External Subsystems:

- MongoDB ; external database subsystem
- RESTful API
- Fetch API/Axios

4.2. Assumptions and Dependencies

- One profile per condo
- The Management Company, Owners and Renters are the only people that will access this system
- Each owner and renter is responsible for their created account that is correctly linked to their respective condos
- The software is developed through the MERN stack and any changes made to the software will be using the same framework
- The system is compatible with a range of modern web browsers.
- The Condo Management System is designed to be deployed on various platforms and operating systems without significant modifications needed.

5. Product Features

- 5.1. User profiles: The condo management companies, condo owners and condo renters will be able to create and manage user profiles which will display their picture, personal information and contact information.
- 5.2. Sign up : Condo management companies, future condo owners and condo renters can create an account using their unique email and a password to list condos and have access to the condos listed on the application.
- 5.3. Log in: Condo management companies, present and future condo owners and renters can securely log in using their email and a password. A two-factor

- authentication can be implemented to make their accounts more secure since they'll have personal information in the system.
- 5.4. Messaging system: Users will be able to communicate with each other via a messaging system when they see an interesting offer that they want to discuss with the condo management companies and owners.
 - 5.5. Property dashboard: Condo owners will be able to see all the properties that they own including necessary information about their properties, their financial status and the requests that were submitted related to their properties.
 - 5.6. Property profile: Condo management companies can create and manage a profile for the property they are listing, this profile will display pictures of the property and vital information such as the price, size, parking and lockers available.
 - 5.7. Notification page: Users can see in a tab all the new messages they've received and all the updates since they last used the app.
 - 5.8. Search feature: Users looking to buy or rent a condo can look for condos based on certain features such as the location and price.
 - 5.9. Search filters: To further ease the search of the ideal condo, users can use filters to narrow down their search. These filters can be the size of the unit, available facilities, units on sale, etc.
 - 5.10. Reservation system: Users will be able to rent common facilities provided by they own or rent through a reservation system. This reservation system will also display all the availability of the facilities to ease the reservation procedure for the user.
 - 5.11. Request submission center: Condo owners can place requests though this system to request access to their condo, to an elevator for moving in or out, etc.
 - 5.12. Registration key maker: Condo management companies can create randomized registration keys through the app to send to their clients.
 - 5.13. Financial system: Condo management companies can set the price of their units, parking spots and lockers through this system, which will calculate the total price that will be presented to the buyer or renter.
 - 5.14. Employee management: Condo management companies can set different roles for their employees for each property that they own.
 - 5.15. Request assignment: When a condo owner submits a request, the request will be sent to the corresponding employee that can take care of it.
 - 5.16. Emergency alert system: Condo management companies can send emergency alerts to their unit owners and renters like when the water will be shut off, elevator will be unavailable, etc.
 - 5.17. Calendar: Condo owners and renters can see the important dates and events happening in their condo.

6. Other Product Requirements

- 6.1. The Condo management app and website is available on MacOs, Linux, Android, iOS and Windows. It's important to concentrate on the web version before the app because it is simpler. The app should have the same theme and layout as the web version.
- 6.2. The language of the app can be changed to English or French. This has a high priority because this feature will bring in more users.
- 6.3. Users can sign in the app using their Gmail account or an SSO (Single Sign On). This has high priority because users have their personal information on the app and using their Gmail account or an SSO offers more security to their condo management account.

S.A.D.

SOEN 390 - Software Engineering Team Design Project

Team 18

Deliverable 3

Architecture Description of Concordia Condo Management System (CCMA) for Condominium Administrative and Community Engagement Platform

Winter 2024

Student Name	Student ID

Jeremy Rimokh	40110746
Ishaï Cohen	40188880
Eden Oliel	40211989
Alexander El Ghawi	40200062
Joud Babik	40031039
Minh Duc Vu	40166824
Nicholas Werugia	40131956
Yash Patel	40175454
Patrick MacEachen	40209790
Vithujanan Vigneswaran	40157822

Contents

Contents	2
List of Figures	4
1 Introduction	5
1.1 Identifying information	5
1.2 Supplementary information	5
1.3 Other information	7
1.3.2 Architecture evaluations	10
1.3.3 Rationale for key decisions	10
2 Stakeholders and concerns	11
2.1 Stakeholders and their Concerns	11
3 Viewpoints	13
3.1 <Viewpoint Name>	15
3.2 Overview	15
3.3 Concerns and stakeholders	15
3.3.1 Concerns	15
3.3.2 Typical stakeholders	16
3.3.3 "Anti-concerns" (optional)	16
3.4 Model kinds+	16
I) Model kind languages or notations (optional)	17
II) Model kind metamodel (optional)	17
III) Model kind templates (optional)	18

3.5.2 <Model Kind Name> operations (optional)	18
3.5.3 <Model Kind Name> correspondence rules	18
3.6 Operations on views	18
3.7 Correspondence rules	18
3.8 Examples (optional)	19
3.9 Notes (optional)	19
3.10 Sources	19
3.1 Functional Viewpoint	19
3.1.1 Overview	19
3.1.2 Concerns and Stakeholders	19
3.1.2.1 Concerns	19
3.1.2.2 Stakeholders	19
3.1.3 Model Kinds	19
3.1.4 Behavioural Model	19
3.1.4.1 Behavioural Model Conventions	19
3.1.4.2 Behavioural Model Correspondence Rules	20
3.1.5 Operation on Views	20
3.1.6 Correspondence Rules	20
3.1.7 Sources	20
3.2 Development Viewpoint	20
3.2.1 Overview	20
3.2.2 Concerns and Stakeholders	20
3.2.2.1 Concerns	20
3.2.2.2 Stakeholders	20
3.2.3 Model Kinds	20
3.2.4 Structural Model	21
3.2.4.1 Structural Model Conventions	21
3.2.4.2 Structural Model Correspondence Rules	21
3.2.5 Operation on Views	21
3.2.6 Correspondence Rules	21
3.2.7 Sources	21
3.3 Deployment Viewpoint	21
3.3.1 Overview	21
3.3.2 Concerns and Stakeholders	21
3.3.2.1 Concerns	21
3.3.2.2 Stakeholders	21
3.3.3 Model Kinds	22
3.3.4 Deployment Diagram	22
3.3.4.1 Deployment Diagram Conventions	22
3.3.4.2 Deployment Diagram Correspondence Rules	22
3.3.5 Operations on Views	22
3.3.6 Correspondence Rules	22
3.3.7 Sources	22
4 Views+	22

4.1 View: Logical View	23
4.1.1 Models+	23
4.1.2 Class Model and Domain Model	24
4.1.3 Known Issues with View	26
4.2 View: Development View	26
4.2.1 Models+	26
4.2.2 Component Model	26
4.2.3 Known Issues with View	27
4.3 View: Process view	27
4.3.1 Models+	28
4.3.3 Known Issues with View	31
4.4 View: Physical view	31
4.4.1 Models+	31
4.4.2 Deployment model	31
4.4.3 Known Issues with View	31
4.5 View: Scenario view	31
4.5.1 Models+	31
4.5.2 Use Case model	31
Scenario for sign up use case:	32
4.5.3 Known Issues with View	35
5 Consistency and correspondences	35
5.1 Known inconsistencies	35
5.2 Correspondences in the AD	35
5.3 Correspondence rules	35

List of Figures

1	The components of the 4 + 1 view model - Sprint 1	21
2	Class diagram for the project - Sprint 1	22
3	Domain model for the project from - Sprint 1	23
4	Component diagram for the project submitted - Sprint 1	24
5	Activity diagram for the Sign Up Process - Sprint 1	26
6	Activity diagram for Property Profile Creation - Sprint 1	27
7	Activity diagram for adding a fee rate for a unit - Sprint 2	28
8	Activity diagram for updating or creating a unit - Sprint 2	29
9	Activity diagram for creating a new property and uploading document-Sprint 2	30
10	Activity diagram for reserving a facility - Sprint 3	31
11	Activity diagram for adding a new facility - Sprint 3	32
12	Deployment diagram for our application - Sprint 3	33
13	Use case scenario for user sign up - sprint 1	35
14	Use case scenario for Financial system operations- sprint 1	36
15	Use case diagram for property management operations - sprint 1	38

List of Tables

1	List of all student authors	7
2	Architecture Decision 1 - Overall Architecture	8
3	Architecture Decision 2 - Cloud-Based Infrastructure	8
4	Architecture Decision 3 - MVC Architecture	9
5	Architecture Decision 4 - Single Page Application (SPA)	9
6	Architecture Decision 5 - RESTful API Design	10
7	Architecture Decision 6 - Use of Frameworks (React, Django, MongoDB)	10

1 Introduction

This chapter describes introductory information items of the AD, including identifying and supplementary information.

1.1 Identifying information

The Architecture Description of Concordia Condo Management System (CCMA) for Condominium Administrative and Community Engagement Platform is a condominium management system, which encompasses an interactive web platform and a

complementary mobile application. This digital ecosystem is designed to serve the administrative needs and community engagement of condominium residents and management, facilitating various functions such as facility booking, maintenance requests, and communication. As a software-intensive system, it integrates various software components and services to provide a seamless user experience and support the operational processes of condo management. The system is developed as an academic project under Concordia University, demonstrating the application of software architecture principles in real-world scenarios.

1.2 Supplementary information

Project Summary

This document outlines the development and implementation of a comprehensive condominium management website, designed to facilitate the efficient management of condo-related activities and services. The project is initiated under the auspices of Concordia University, supervised by Dr. Jinqiu Yang. The website aims to provide a centralized platform for condo residents, management, and staff to communicate, manage facilities, and access important information with ease.

Date of Issue: January 18th 2023

Status: In Development

Authors: Listed in table 1

Reviewers: Lin Ling (Teaching Assistant) at Concordia university

Approving Authority: Lin Ling (Teaching Assistant) and Dr. Jinqiu Yang

Issuing Organization: Concordia University

Change History: This is the first version of the document submitted for sprint 1

Scope: the scope of the project is detailed below

Context: The project is developed as part of the coursework for SOEN 390, aiming to address the practical needs of condo management through a digital solution.

Version Control Information: Managed through GitHub repository "THE-390"

Configuration Management Information: Document versioning and updates are tracked via GitHub to ensure consistency and collaboration among development team members.

Project Scope

The scope of the condo management website includes:

- **User Management:** Secure registration and login mechanisms for residents, condo management, and staff. Profiles for each user category with relevant permissions and access levels.
- **Communication Platform:** A forum or bulletin board for announcements, queries, and feedback between residents and the management.
- **Facility Booking and Management:** An interface for booking common facilities, such as meeting rooms, gym slots, and event spaces. Management can update facility status and availability.
- **Maintenance Requests:** A portal for residents to submit maintenance requests and track their status. Integration with management's workflow for efficient handling and resolution.
- **Document Repository:** A secure location for storing and accessing condominium documents, policies, and meeting minutes.
- **Mobile App:** To ensure accessibility and convenience, a mobile application will complement the website, providing users with on-the-go access to features and notifications.
- **Integration and Scalability:** The platform will be designed for easy integration with existing management tools and scalable to accommodate future requirements.
- **Version Control:** Development is managed through the GitHub repository "THE-390", accessible at [\[https://github.com/JRB958/THE-390.git\]](https://github.com/JRB958/THE-390.git), ensuring a collaborative and transparent development process.

This project is designed to enhance the living experience of condo residents by streamlining management processes and fostering a connected community. Through careful planning and execution, it aims to set a standard for digital condo management solutions.

Student Name	Student ID
Jeremy Rimokh	40110746
Ishaï Cohen	40188880
Eden Oliel	40211989
Alexander El Ghawi	40200062
Joud Babik	40031039
Minh Duc Vu	40166824
Nicholas Werugia	40131956
Yash Patel	40175454

Patrick MacEachen	40209790
Vithujanan Vigneswaran	40157822

Table 1: List of all student authors.

1.3 Other information

The MVC-architected Condo Management System centralizes condominium operations, enabling user profile creation and property management through a streamlined dashboard interface. This AD serves as a roadmap for navigating the system's architecture, highlighting its core functionalities and the rationale behind key design choices.

Evaluations of the architecture affirm its capability to support the system's robust feature set while maintaining scalability and performance. Decisions made throughout the design process, particularly the adoption of MVC, were anchored in ensuring a modular, maintainable, and secure framework for the Condo Management System.

Interrelationships between the architecture's various views and models are clearly defined, ensuring a comprehensive understanding of the system's structure and operation for all stakeholders involved in the project.

Related requirements	The condominium management system shall provide a seamless interface for users to manage their property, communicate with management, book facilities, and submit maintenance requests.
Alternative	Standalone desktop application.
Constraint	<ol style="list-style-type: none"> 1. The system must be easily accessible without extensive installation procedures. 2. It needs to be compatible with various devices used by the condo residents and management staff.
Assumption	Most users have access to the internet and are more accustomed to using web and mobile applications rather than desktop applications.
Architecture Decision	
Identifier	#1
Description	The system will be developed as a cross-platform web application that is complemented by a mobile application for both Android and iOS devices.
Rationale	1. A web application does not require users to install

	<p>software, making it instantly accessible via a web browser.</p> <p>2. A mobile application ensures accessibility for users on-the-go and provides notifications directly to their smartphones.</p> <p>3. Cross-platform compatibility ensures that regardless of the device (PC, tablet, smartphone), users can access the system with their preferred device.</p> <p>4. The development team is proficient in web and mobile application development, which aligns with the skill set required for the project.</p>
--	---

Table 2: Architecture Decision 1 - Overall Architecture

Related requirements	The system must be highly available, scalable, and capable of handling varying loads without service degradation.
Alternative	On-premises hosting solution.
Constraint	<p>1. Limited physical infrastructure resources.</p> <p>2. Need for a scalable solution that can adapt to fluctuating usage patterns.</p>
Assumption	The user base will grow, and the system will require scaling in response to the increased demand.
Architecture Decision	
Identifier	#2
Description	Implement the web application on a cloud-based infrastructure using a provider such as AWS, Azure, or Google Cloud.
Rationale	<p>1. Cloud providers offer on-demand resource scaling to handle increased traffic without physical hardware constraints.</p> <p>2. High availability configurations provided by cloud services ensure minimal downtime and reliable user access.</p> <p>3. Cloud services include disaster recovery capabilities, securing data and services against potential incidents.</p>

Table 3: Architecture Decision 2 - Cloud-Based Infrastructure

Related requirements	The system must separate the data model, user interface, and control logic to facilitate maintainability and scalability.
Alternative	Monolithic architecture.
Constraint	<ol style="list-style-type: none"> 1. The team's familiarity with the MVC pattern. 2. The need for a clear separation of concerns to simplify future updates and maintenance.
Assumption	Development using MVC will improve code quality and make future enhancements easier and more manageable.
Architecture Decision	
Identifier	#3
Description	Employ an MVC (Model-View-Controller) architecture to structure the application into logical chunks for better manageability and separation of concerns.
Rationale	<ol style="list-style-type: none"> 1. MVC architecture enhances the ability to test components in isolation, improving overall system reliability. 2. It provides a clear framework for development, making it easier for new developers to understand the codebase. 3. Separation of concerns facilitates easier maintenance and scalability of the application over time.

Table 4: Architecture Decision 3 - MVC Architecture

Related requirements	The user interface must provide a fluid and interactive experience, minimizing page reloads and delay.
Alternative	Traditional multi-page web application.
Constraint	<ol style="list-style-type: none"> 1. The need for an application-like user experience within a web browser. 2. Ensuring compatibility across all modern browsers.
Assumption	Users expect a fast and seamless interaction with the

	web application, akin to native desktop or mobile applications.
Architecture Decision	
Identifier	#4
Description	Develop the web interface as a Single Page Application (SPA) using a JavaScript framework like React.
Rationale	<ul style="list-style-type: none"> 1. SPAs offer a more dynamic and responsive user experience by only reloading parts of the page in response to user actions. 2. This approach minimizes server round trips, reducing the overall latency of the application. 3. It allows for the application's UI to be more responsive and provides the end-user with a smooth and uninterrupted experience.

Table 5: Architecture Decision 4 - Single Page Application (SPA)

Related requirements	The system must enable integration with various services and client applications while adhering to industry standards.
Alternative	SOAP-based web services.
Constraint	<ul style="list-style-type: none"> 1. The need for a stateless architecture that can handle numerous clients simultaneously. 2. Requirement for a scalable and flexible backend service that can evolve over time.
Assumption	RESTful services are widely accepted and understood, making them a default choice for web API development.
Architecture Decision	
Identifier	#5
Description	Implement a RESTful API design for backend services, facilitating ease of integration and adherence to web standards.
Rationale	<ul style="list-style-type: none"> 1. RESTful APIs use standard HTTP methods, making them easy to implement and consume by various clients. 2. They support a stateless communication mechanism,

	<p>which is ideal for web-based services where numerous clients may interact with the system concurrently.</p> <p>3. RESTful APIs provide a flexible way to deliver data in multiple formats, such as JSON or XML, accommodating different client needs.</p>
--	--

Table 6: Architecture Decision 5 - RESTful API Design

Architecture Decision	
Identifier	#6
Description	The web application will utilize React for the frontend to build a component-based user interface, Django as the backend framework for robust and scalable web services, and MongoDB for a flexible, schema-less database.
Rationale	<ol style="list-style-type: none"> React's component-based architecture allows for reusable UI components, making the frontend scalable and easier to maintain. Django offers a comprehensive set of tools for rapid

Table 7: Architecture Decision 6 - Use of Frameworks (React, Django, MongoDB)

1.3.2 Architecture evaluations

Tables 2 to 7 highlight evaluations and descriptions for Architectural decisions for sprint 1

1.3.3 Rationale for key decisions

Tables 2 to 7 highlight rationale for Architectural decisions for sprint 2

2 Stakeholders and concerns

2.1 Stakeholders and their Concerns

1. **Project manager** : A person that oversees the development of the system.
 - a. Scalability : Can the system handle a growing number of users and new updates ?
 - b. Feasibility : Are the resources (developers and budget) allocated to develop the system sufficient?
 - c. Feasibility : Can the system be deployed in a reasonable time?
2. **Product Owner** : A person that provides the directions about the system's purpose based on the users needs and business goals.
 - a. Scalability : Can the system handle a growing number of users and new updates ?
 - b. Security : Can the system protect users from malicious users and threats ?
 - c. Flexibility: Can the system tolerate customization to meet end users needs?
 - d. Interoperability : Can the system support integration with other systems and services?
3. **Development Team** : A team consisting of people that will code and program the application.
 - a. Performance : Will the system be able to handle heavy traffic and offer a smooth experience?
 - b. Flexibility: Can the system tolerate customization to meet end users needs?
 - c. Scalability : Can the system handle a growing number of users and new updates ?
 - d. Testability : Can the system be tested without much effort?
 - e. Security : Can the system protect users from malicious users and threats ?
 - f. Flexibility : Does the system run similarly on any device?
4. **Marketing Team** : A team consisting of people that will come up with a strategy to promote the app on the market.

- a. Interoperability : Can the system support integration with other systems and services?
 - b. Scalability : Can the system handle a growing number of users and new updates ?
 - c. Security : Can the system protect users from malicious users and threats ?
5. **Investors** : A person that is funding the creation of this application.
- a. Scalability : Can the system survive on the market against its competitors and quickly growing trends?
 - b. Usability : Is there a demand on the market for the features offered by the system?
6. **IT Department** : Team consisting of people that maintain the software and hardware of the application.
- a. Performance : Will the system be able to handle heavy traffic and offer a smooth experience?
 - b. Portability : Can the system run on numerous platforms?
 - c. Reliability : Can the system still run some features when there is no network connectivity?
 - d. Flexibility : Does the system run similarly on any device?
7. **Condo Management Company** : A company that owns condos and that is trying to put them on sale or rent.
- a. Security : Can the system protect users from malicious users and threats ?
 - b. Usability : Is the sign up process and property profile creation simple?
 - c. Interoperability : Can the system support integration with other systems and services?
 - d. Usability : Is it easy to customize a profile that was created ?
8. **Future Condo Owner** : A person that is looking to buy a new home.
- a. Security : Can the system protect users from malicious users and threats ?
 - b. Usability : Is the sign up process and user profile creation simple?
 - c. Interoperability : Can the system support integration with other systems and services?
9. **Current Condo Owner** : A person that has purchased units of a condo.
- a. Security : Can the system protect users from malicious users and threats ?
 - b. Usability : Is it easy to customize a profile that was initially created ?
10. **Future Condo Renter** : A person that is looking to rent a new home.
- a. Security : Can the system protect users from malicious users and threats ?
 - b. Usability : Is the sign up process and user profile creation simple?

- c. Interoperability : Can the system support integration with other systems and services?
11. **Current Condo Renters** : A person that is currently renting a unit of a condo.
- a. Security : Can the system protect users from malicious users and threats ?
 - b. Usability : Is it easy to customize a profile that was initially created ?
12. **Employee of the Condo Management Company** : A person that was assigned by the condo management company to treat their customers requests.
- a. Security : Can the system protect users from malicious users and threats ?
 - b. Usability : Is the sign up process and user profile creation simple?
 - c. Interoperability : Can the system support integration with other systems and services?
 - d. Usability : How can the system be used to fulfill requests submitted by a condo owner?
13. **Real Estate Brokers** : A person that helps either the condo management company or the client to sell or buy a property.
- a. Security : Can the system protect users from malicious users and threats ?
 - b. Usability : Is it easy to get in contact with clients?
14. **Hacker** : A person that is trying to take advantage of the app by obtaining access to unauthorized personal information.
- a. Security : Can the system protect users from malicious users and threats ?
15. **Competitors** : A company that views the app as a threat for their business.
- a. Security : Which information should be available to the public ?
 - b. Security : Can the system be reverse engineered easily?
16. **Data Brokers** : An organization that collects personal data to sell them later.
- a. Security : Can the system protect users from malicious users and threats ?
 - b. Security : Which information should be available to the public ?

3 Viewpoints

3.1 Functional Viewpoint

3.1.1 Overview

The functional viewpoint describes what the system does from the viewpoint of the user. It defines how the system will execute the required functions and how the exposed interfaces interact with each other.

3.1.2 Concerns and Stakeholders

3.1.2.1 Concerns

From a functional viewpoint, the concerns are:

- How does the user interact with the system to accomplish a task?
- What are the critical use cases of the system?
- What are the main modules or subsystems of the system and how do they interact with each other?

3.1.2.2 Stakeholders

The typical stakeholders of this view are:

- Product owner
- Development team
- Future and current condo owners/renters
- Condo Management Company
- Employees

3.1.3 Model Kinds

For this viewpoint, the kind of diagrams used are the use case diagram and the activity diagram. Both are classified into the behavioral model..

3.1.4 Behavioural Model

3.1.4.1 Behavioural Model Conventions

The UML conventions were followed to create the diagrams..

3.1.4.2 Behavioural Model Correspondence Rules

- An actor should be associated with at least one use case
- Actors are properly identified. They should have different roles.
- Avoid redundant use cases.
- Actions are well defined.

3.1.5 Operation on Views

- Define actors that need to be included in the diagram
- Define use cases

- Associate actors with use cases

3.1.6 Correspondence Rules

- The scenario view will be described by the use case diagram.
- The process view will be described by the activity diagram.

3.1.7 Sources

- <https://www.viewpoints-and-perspectives.info/home/viewpoints/functional-view-point/>
- <https://www.javatpoint.com/uml-diagrams>

3.2 Development Viewpoint

3.2.1 Overview

The development viewpoint describes how the system has been developed. It focuses on the development process, the methodologies used and the software artifacts created during the process.

3.2.2 Concerns and Stakeholders

3.2.2.1 Concerns

From a development viewpoint, the concerns are:

- What are the main components or modules of the system?
- How are components and modules connected and organized?
- What is the model or pattern used for the system structure?
- Can the system be tested without much effort?
- Can the system protect users from malicious users and threats ?
- Does the system run similarly on any device?

3.2.2.2 Stakeholders

From a development viewpoint, the stakeholders are:

- Development team
- IT department
- Condo Management Company
- Product Owner

3.2.3 Model Kinds

For this viewpoint, the diagrams used are the domain model, component diagram and the class diagram. They are classified as structural diagrams, which means the only model kind for this viewpoint is the structural model.

3.2.4 Structural Model

3.2.4.1 Structural Model Conventions

The structural diagrams all follow the UML convention. They describe the structure of the system.

3.2.4.2 Structural Model Correspondence Rules

- Classes are well defined
- Classes have associations with other classes
- Associations should have a multiplicity
- Activity diagram should be thorough.
- Conceptual classes in the domain model come from the requirements

3.2.5 Operation on Views

- Define classes
- Define the associations and multiplicity between classes
- Make a list of possible user actions
- Define key concepts and relationships

3.2.6 Correspondence Rules

- The class diagram will define the logical view.
- The domain model will define the logical view.
- The component diagram will describe the logical view.

3.2.7 Sources

- <https://www.viewpoints-and-perspectives.info/home/viewpoints/development/>
- <https://www.javatpoint.com/uml-diagrams>

3.3 Deployment Viewpoint

3.3.1 Overview

The deployment viewpoint focuses on how the system is deployed, and what type of environment the system runs on. It also describes any third-party requirements and physical constraints for the deployment process.

3.3.2 Concerns and Stakeholders

3.3.2.1 Concerns

From a deployment viewpoint, the concerns are:

- How scalable is the deployment architecture?
- How is the performance of the deployment architecture?
- How is load balancing handled?
- What are the security measures put in place to protect data?

3.3.2.2 Stakeholders

From a deployment viewpoint, the stakeholders are:

- Development Team
- IT Department

3.3.3 Model Kinds

The model used to describe this viewpoint is the deployment diagram.

3.3.4 Deployment Diagram

3.3.4.1 Deployment Diagram Conventions

To construct the deployment diagram, the UML conventions must be followed.

3.3.4.2 Deployment Diagram Correspondence Rules

- Relate software elements and the corresponding hardware elements they are deployed to.
- Associate nodes with each other to describe their interactions
- Each software element should belong to only one node
- Indicate how load balancing is handled

3.3.5 Operations on Views

- Identify key software and hardware elements
- Associate nodes and indicate messages they pass
- Show the artifacts in the nodes

3.3.6 Correspondence Rules

- The physical view will be described by the deployment diagram

3.3.7 Sources

- <https://www.viewpoints-and-perspectives.info/home/viewpoints/deployment/>
- <https://www.javatpoint.com/uml-diagrams>

4 Views

Much of the material in an AD is presented through its architecture views. Each view follows the conventions of its governing viewpoint. A view is made up of architectural models.

The Logical View in the 4+1 View Model primarily addresses the functional requirements of the system, providing a high-level view of the application's core structure and its behavior from a user's perspective. It is typically modeled using class diagrams, which showcase the system's key classes, interfaces, and the relationships between them, encapsulating the system's principal functionality. This view is crucial for understanding the object model of the application and is often the focus for application developers as it guides the application's development in terms of key abstractions and their interactions.

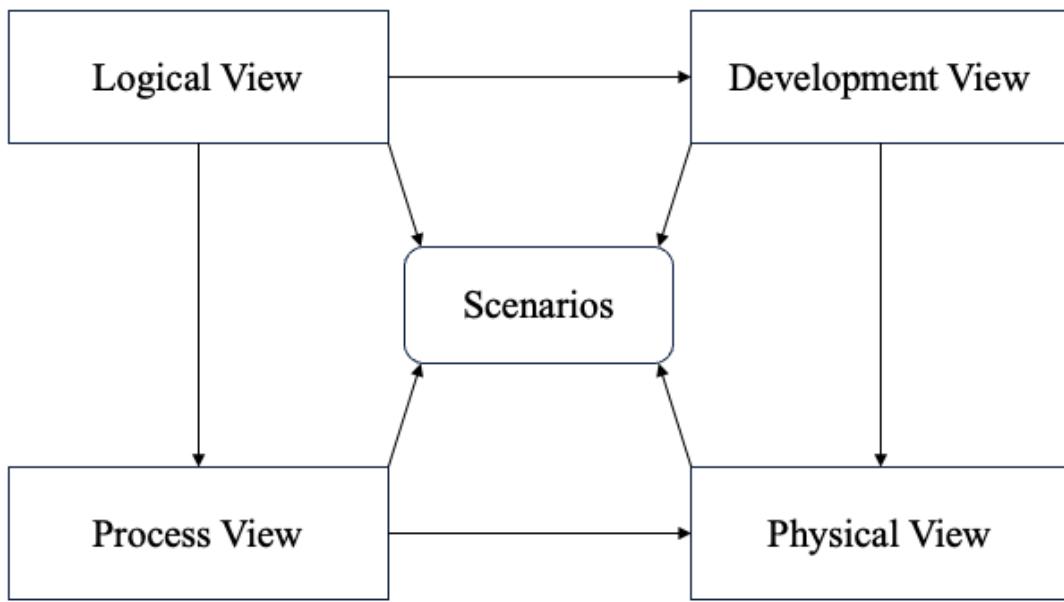


Figure 1 The components of the 4 + 1 view model - Sprint 1

4.1 View: Logical View

The Logical View in the 4+1 Model governed by the functional viewpoint details the system's structure through classes, interfaces, and their interactions, focusing on end-user functionality and object responsibilities. It maps out the object-oriented design, highlighting aspects like inheritance and object relationships, which are crucial for developers.

4.1.1 Models+

The logical view can be represented by the multiple models, including the Class model, the sequence model, or State model.

4.1.2 Class Model and Domain Model



Figure 2: Class diagram for the project - Sprint 1

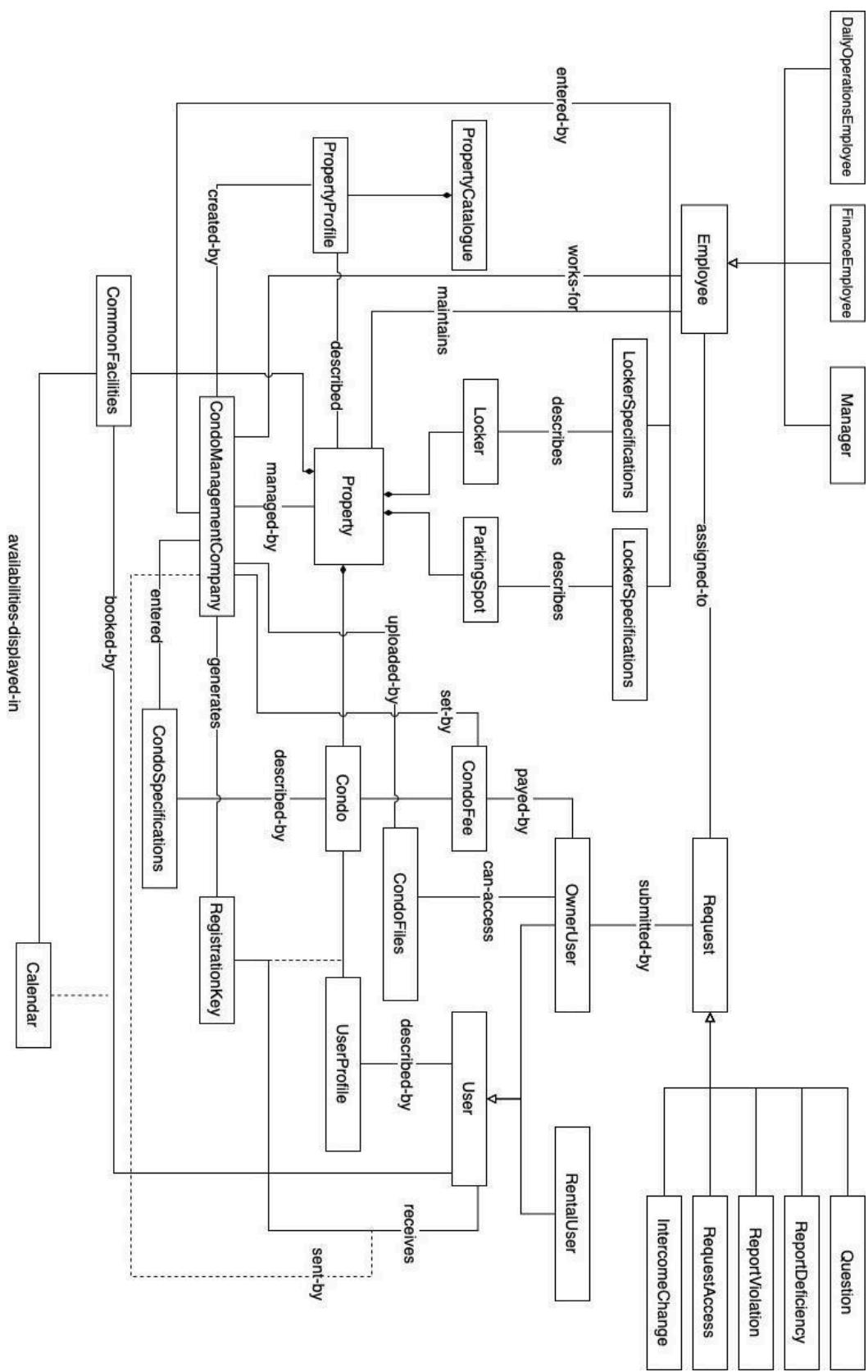


Figure 3: Domain model for the project from - Sprint 1

4.1.3 Known Issues with View

Multiplicities of the classes in the domain and class models need to be added as well as the

4.2 View: Development View

The Development View in the 4+1 View Model which is governed by the development viewpoint delineates the system's structure from the developer's perspective, focusing on software management aspects like module organization, source code layering, and component partitioning. It uses diagrams such as component and package diagrams to provide a roadmap for the system's software build, its dependencies, and its development environment setup.

4.2.1 Models+

The development view can be represented with the package model which outlines how code and components are organized into packages or modules and the component model which will be presented.

4.2.2 Component Model

The Component Model within the Development View represents the modular parts of a system's software architecture, detailing how each component functions, interacts with others, and its dependencies. It serves as a blueprint for developers, illustrating the breakdown of the system into manageable, reusable, and interchangeable parts that encapsulate specific functionality.

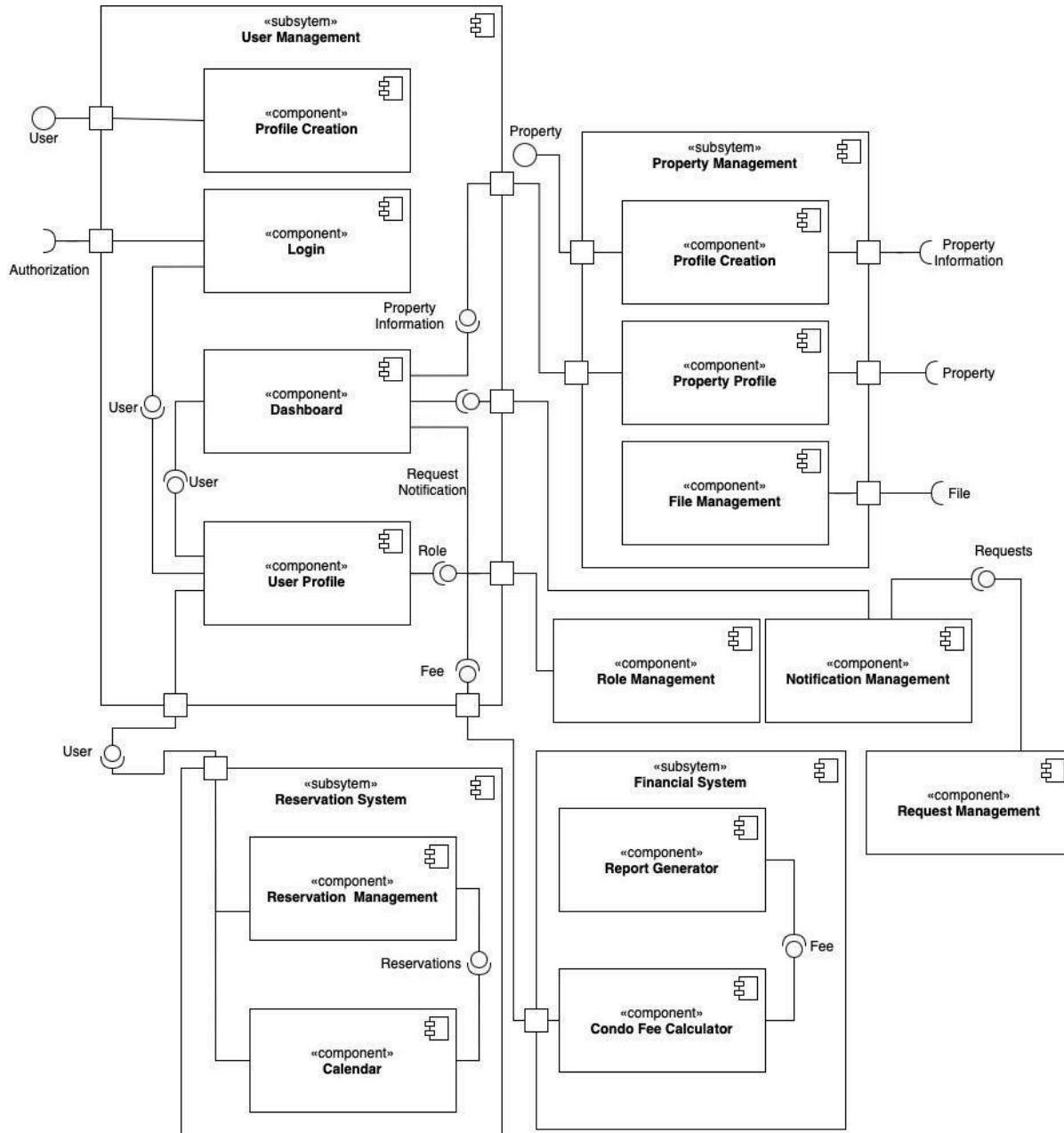


Figure 4 component diagram for the project submitted - Sprint 1

4.2.3 Known Issues with View

nothing for now

4.3 View: Process view

The Process View of the 4+1 View Model which is governed by the behavioral viewpoint captures the dynamic aspects of the system, illustrating how runtime processes interact and manage data flow. It focuses on the concurrency, synchronization, and communication between various components, often represented through activity and sequence diagrams, to ensure the system's scalability and performance.

4.3.1 Models+

The process view can be represented by the concurrency model which details the concurrency mechanisms and synchronization aspects of a system. In addition, the activity model also represent the view.

4.3.2 Activity Model

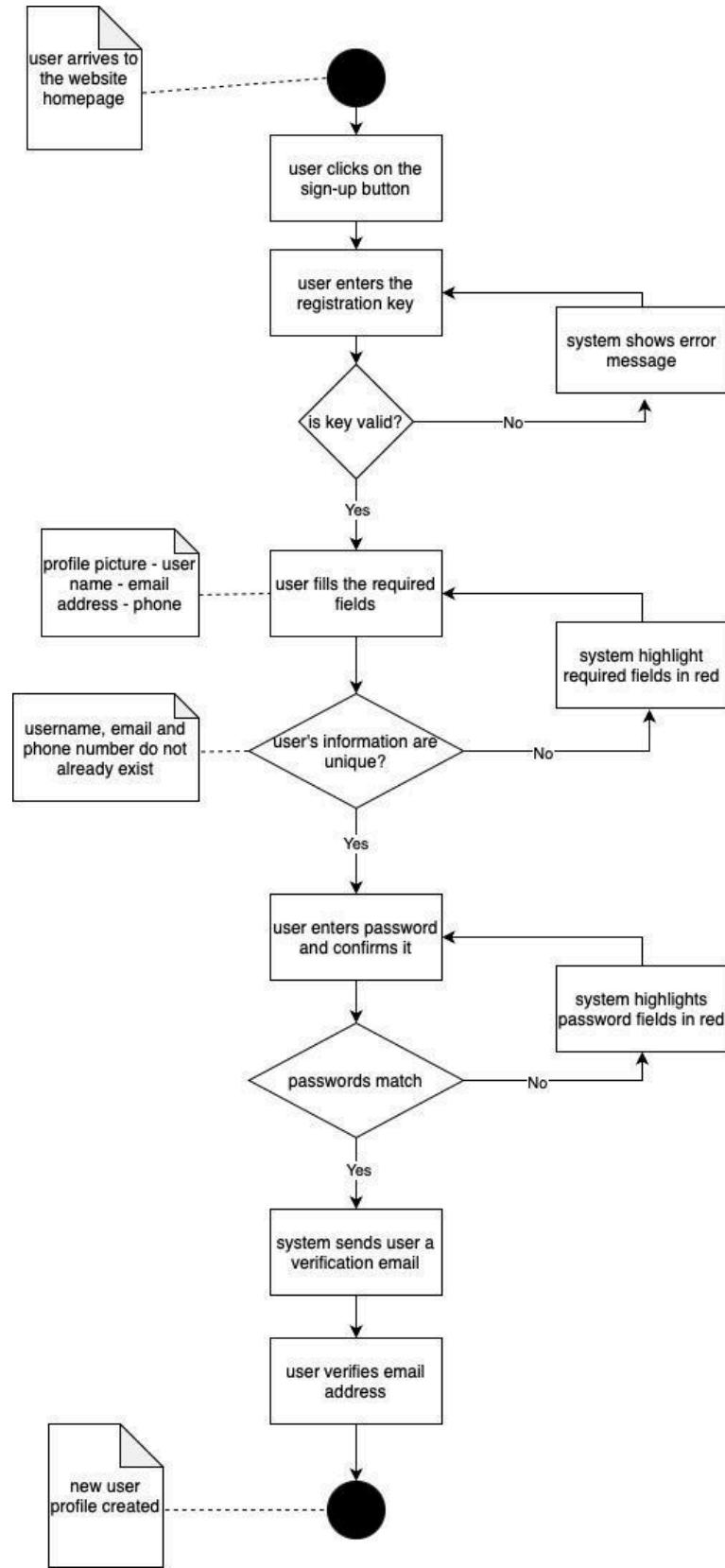


Figure 5 Activity diagram for the Sign Up Process - Sprint 1

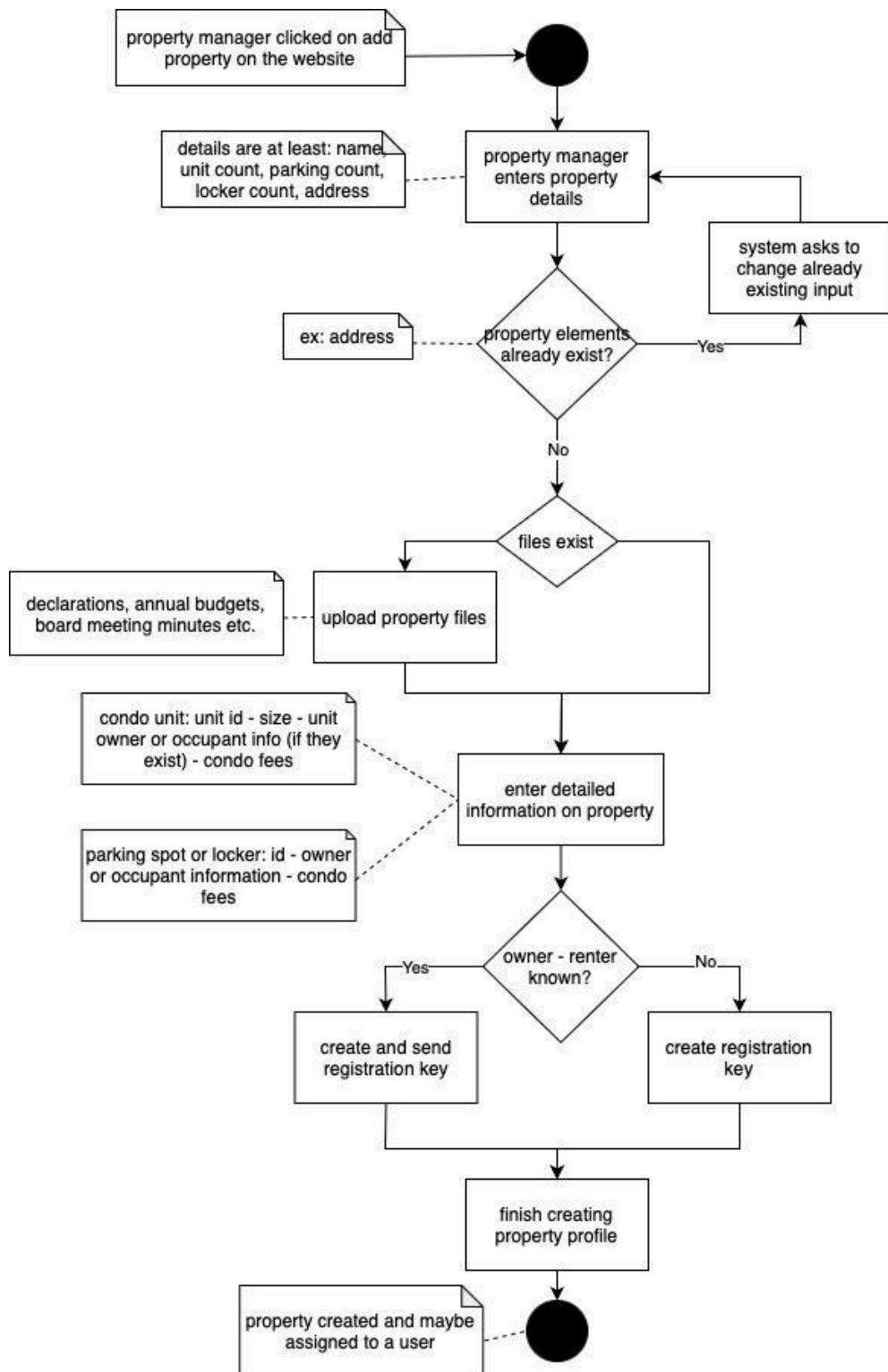


Figure 6: Activity diagram for Property Profile Creation - Sprint 1

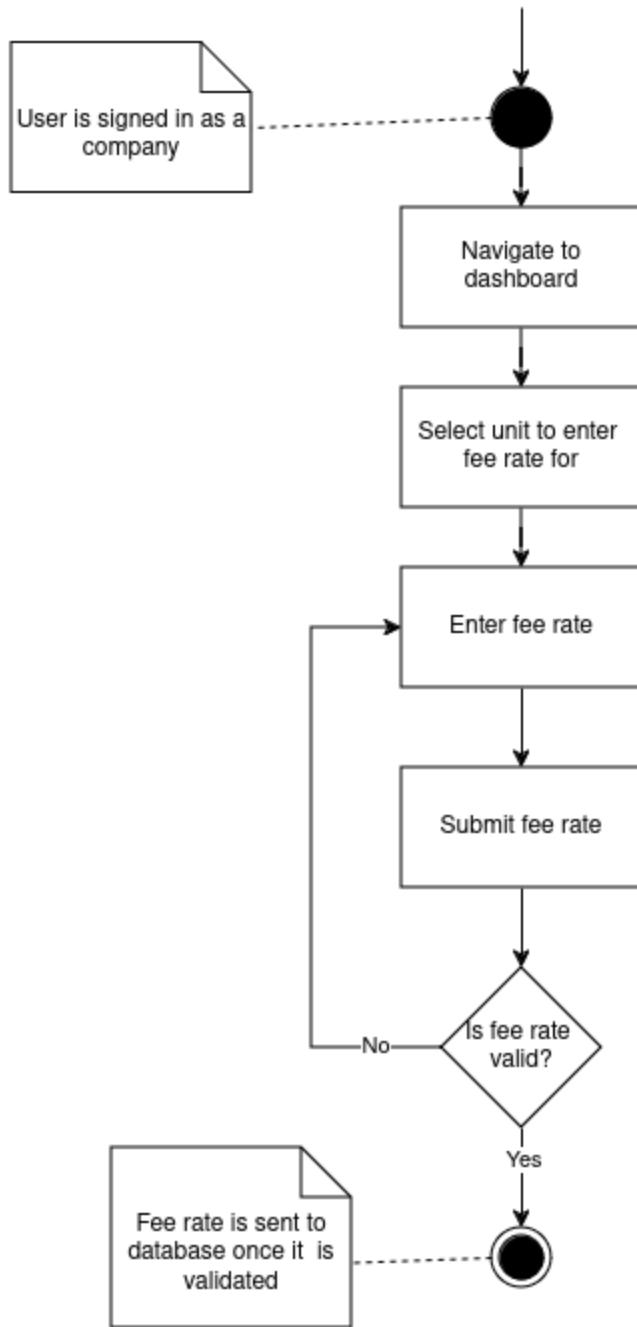


Figure 7: Activity diagram for adding a fee rate for a unit - Sprint 2

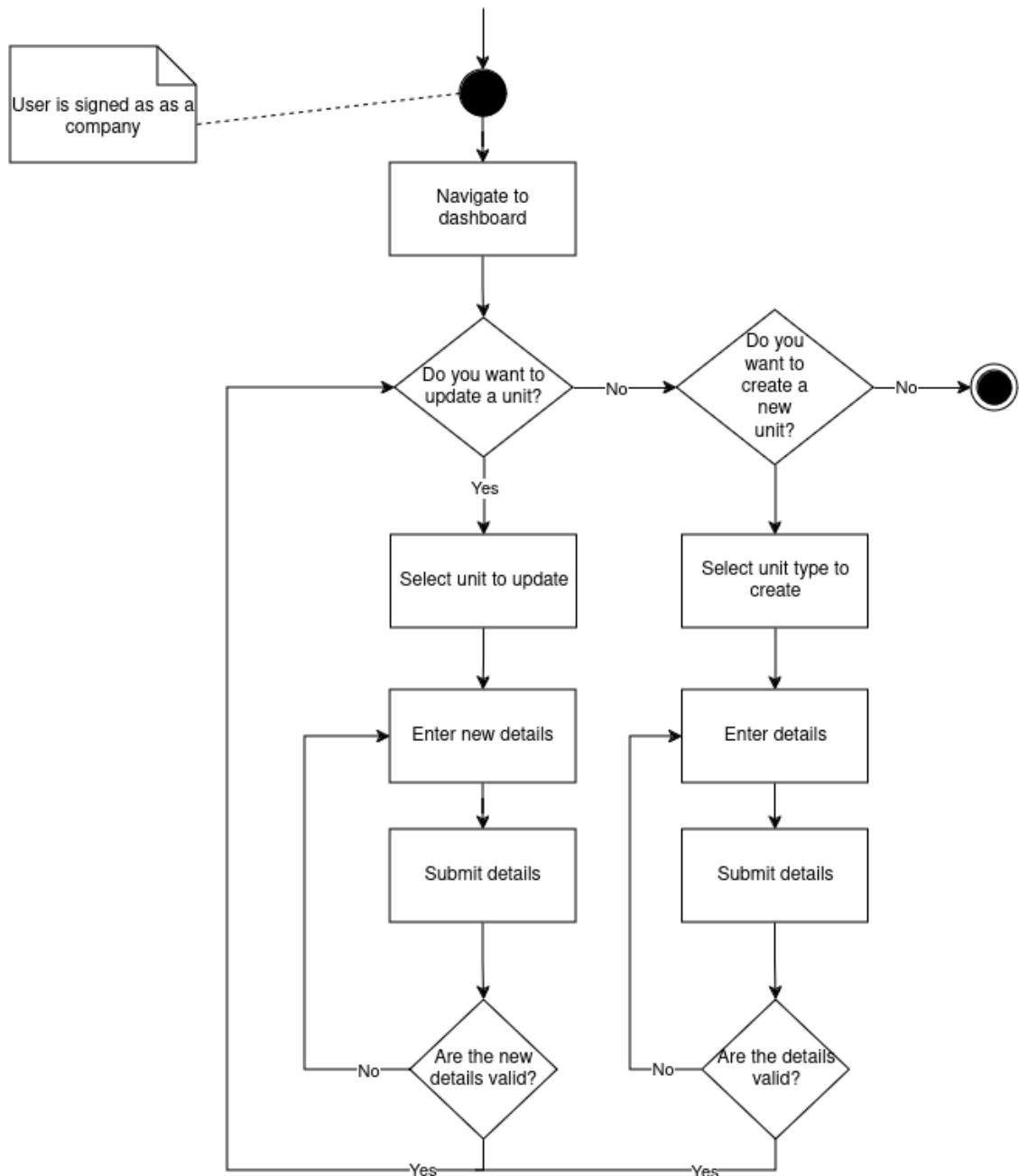


Figure 8: Activity diagram for updating or creating a unit - Sprint 2

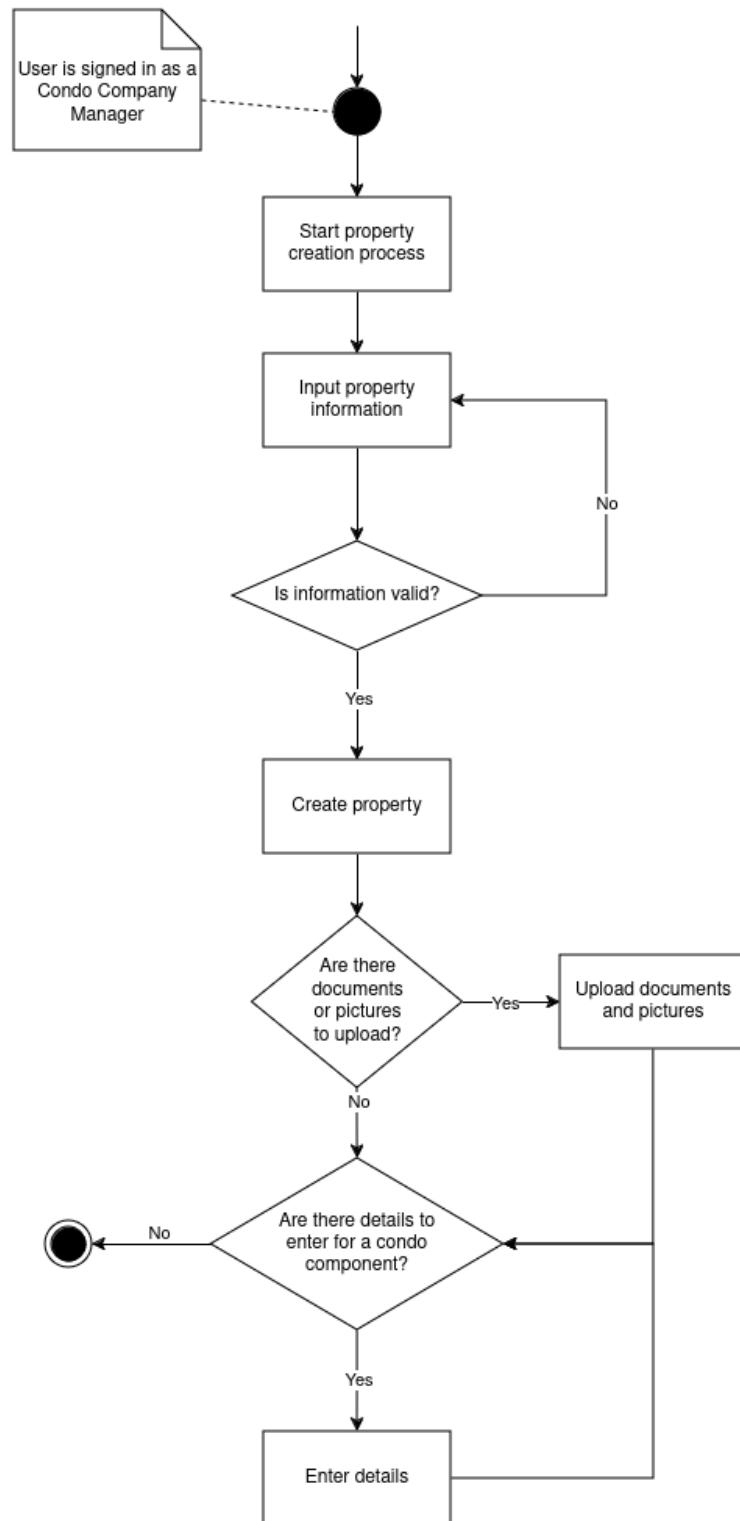


Figure 9: Activity diagram for creating a new property and uploading documents - Sprint 2

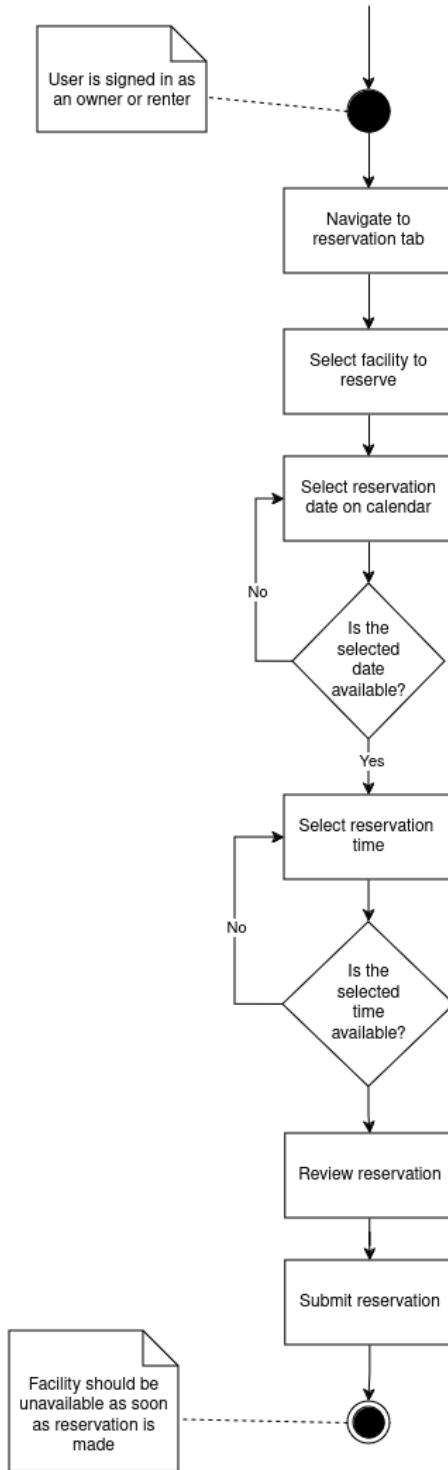


Figure 10: Activity diagram for reserving a facility - Sprint 3

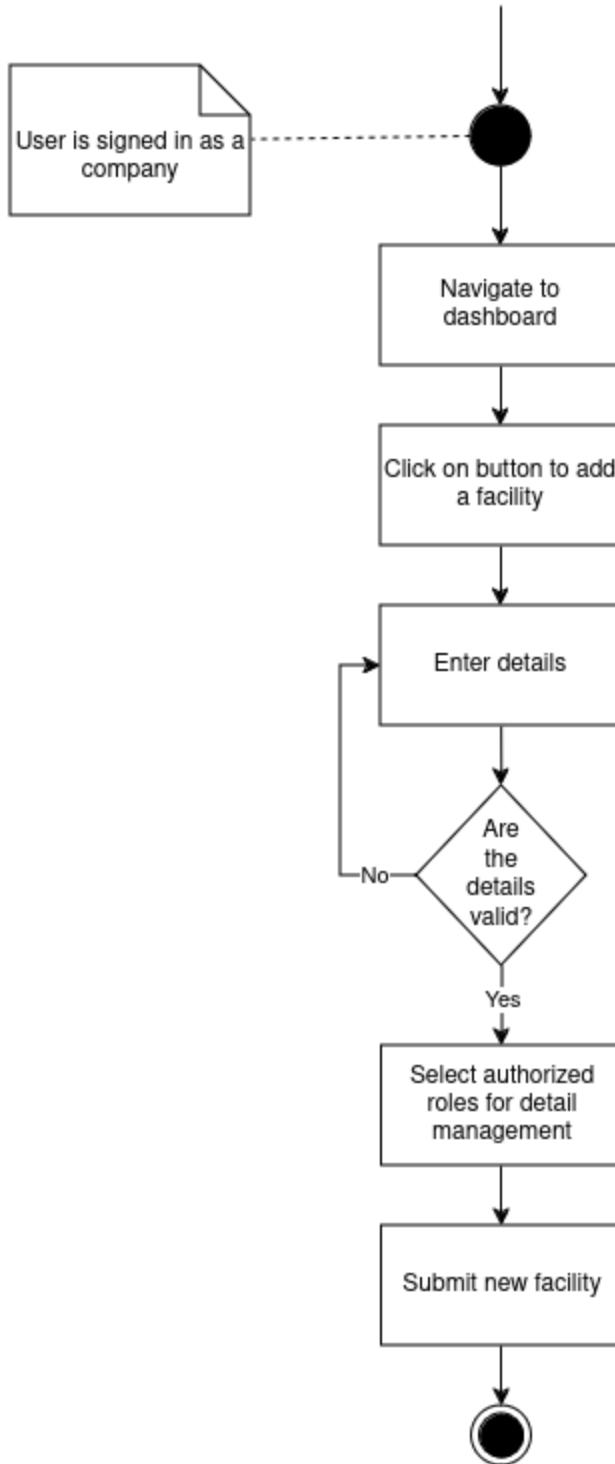


Figure 11: Activity diagram for adding a new facility - Sprint 3

4.3.3 Known Issues with View

There's more diagrams to be added for later stages of the project

4.4 View: Physical view

The details of this information will be as specified by the organization and/or project. See §1 for examples of identifying and supplementary information.

Views have their own identifying and supplementary information distinct from ADs because they may be developed and evolve separately over the lifetime of a project.

4.4.1 Models+

The deployment model represents the physical view which is in turn governed by the deployment viewpoint.

4.4.2 Deployment model

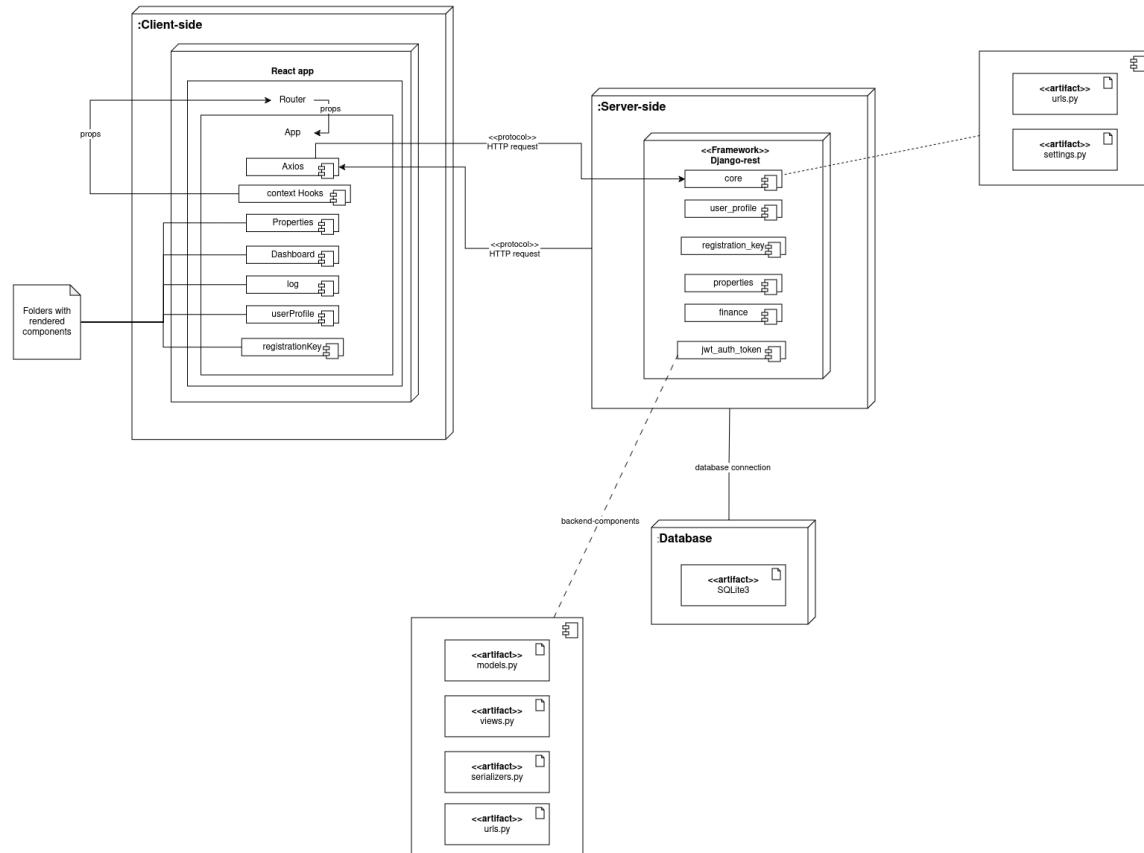


Figure 12: Deployment diagram for our application - Sprint 3

4.4.3 Known Issues with View

the model is yet to be created to fit the physical view

4.5 View: Scenario view

The Scenario View, governed by the usability viewpoint is central to the 4+1 View Model, provides a practical illustration of the system's architecture through real-world use cases, detailing how various elements interact to fulfill specific user-driven tasks. It serves as a

means to validate the architecture, ensuring that it meets the user requirements and stakeholder expectations. This view often uses use case diagrams and sequence diagrams to depict the interaction between the users and the system, highlighting the system's behavior in various scenarios.

4.5.1 Models+

The scenario view can be represented by the interaction model which illustrates the flow of activities and communication between system components during user interactions, using diagrams to detail the sequence of events. It validates system behavior, ensuring alignment with user requirements and operational objectives in real-world scenarios. as well as the use case model discussed below

4.5.2 Use Case model

The Use Case Model captures and organizes the functional requirements of a system by detailing the interactions between actors (users or external systems) and the system itself to achieve specific goals. It is expressed through use case diagrams, which provide a high-level overview of the system's functionality and clarify the context and scope of each interaction.

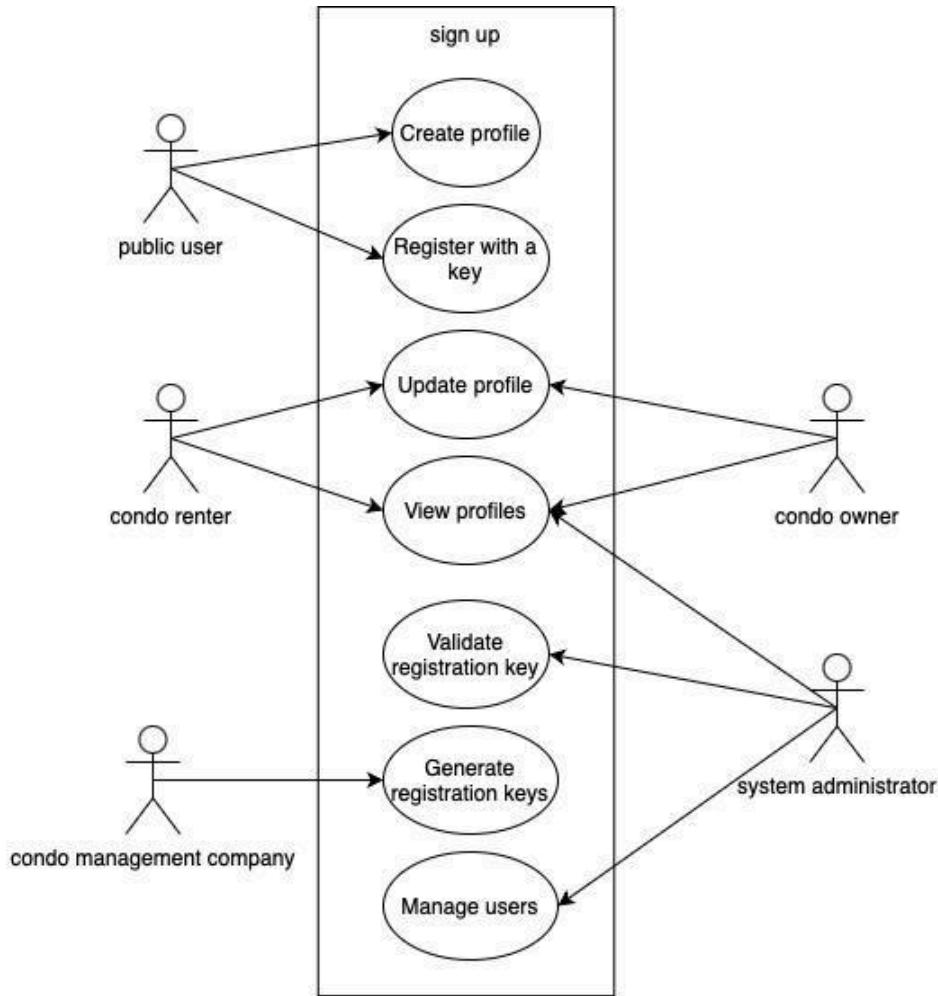


Figure 13: Use case scenario for user sign up - sprint 1

Scenario for sign up use case:

1. A **Public User** accesses the system and chooses to create a profile.
2. The user provides the necessary details (profile picture, username, email, phone number) and creates their profile.
3. The user then receives a registration key from their **Condo Management Company**.
4. The user enters this key into the system to register either as a **Condo Owner** or a **Rental User**.
5. The system validates the key.
6. Upon successful validation, the user's status is updated to either Condo Owner or Rental User, and they can now access additional functionalities like updating their profile.
7. The **System Administrator** oversees these operations, ensuring everything runs smoothly and managing user accounts as needed.

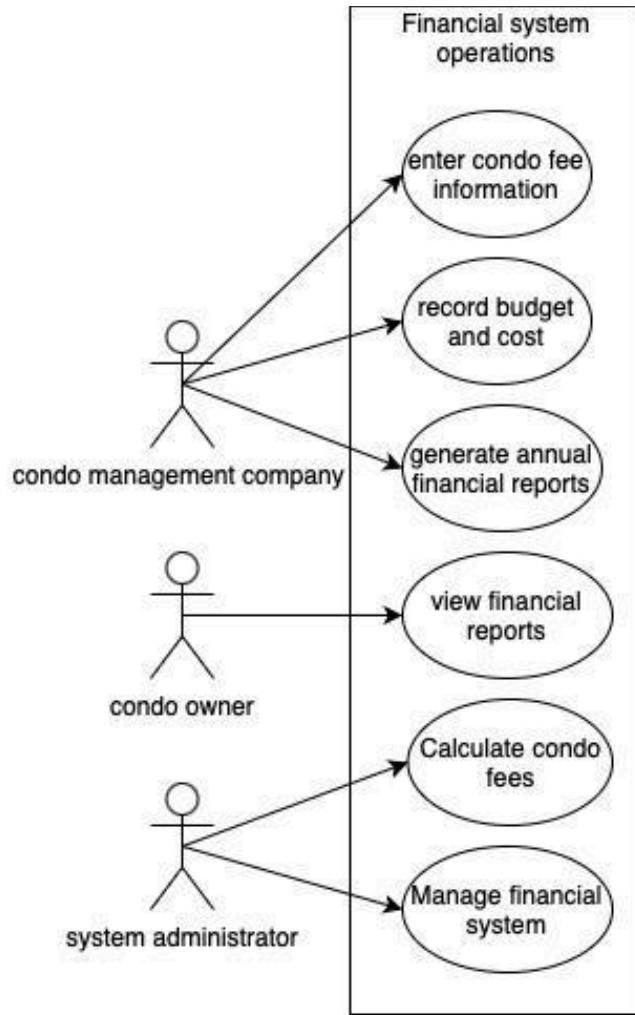


Figure 14: Use case scenario for Financial system operations- sprint 1

Scenario for financial system operations:

1. The **Condo Management Company** enters the condo fees per square foot and per parking spot into the system.
2. The system calculates the total condo fee for each unit based on these rates and the specific details of each unit.
3. The management company also records and enters the operational budget, which includes the total collected condo fees, and inputs the cost for each operation.
4. At the end of the year, the company uses the system to generate an annual financial report, which includes details of all the condo fees collected and other financial data.
5. **Condo Owners** can access this report to understand the financial status of their property, including the total fees paid and how these funds are being utilized.
6. The **System Administrator** oversees the system to ensure that all financial data is correctly processed and that the system remains secure and functional.

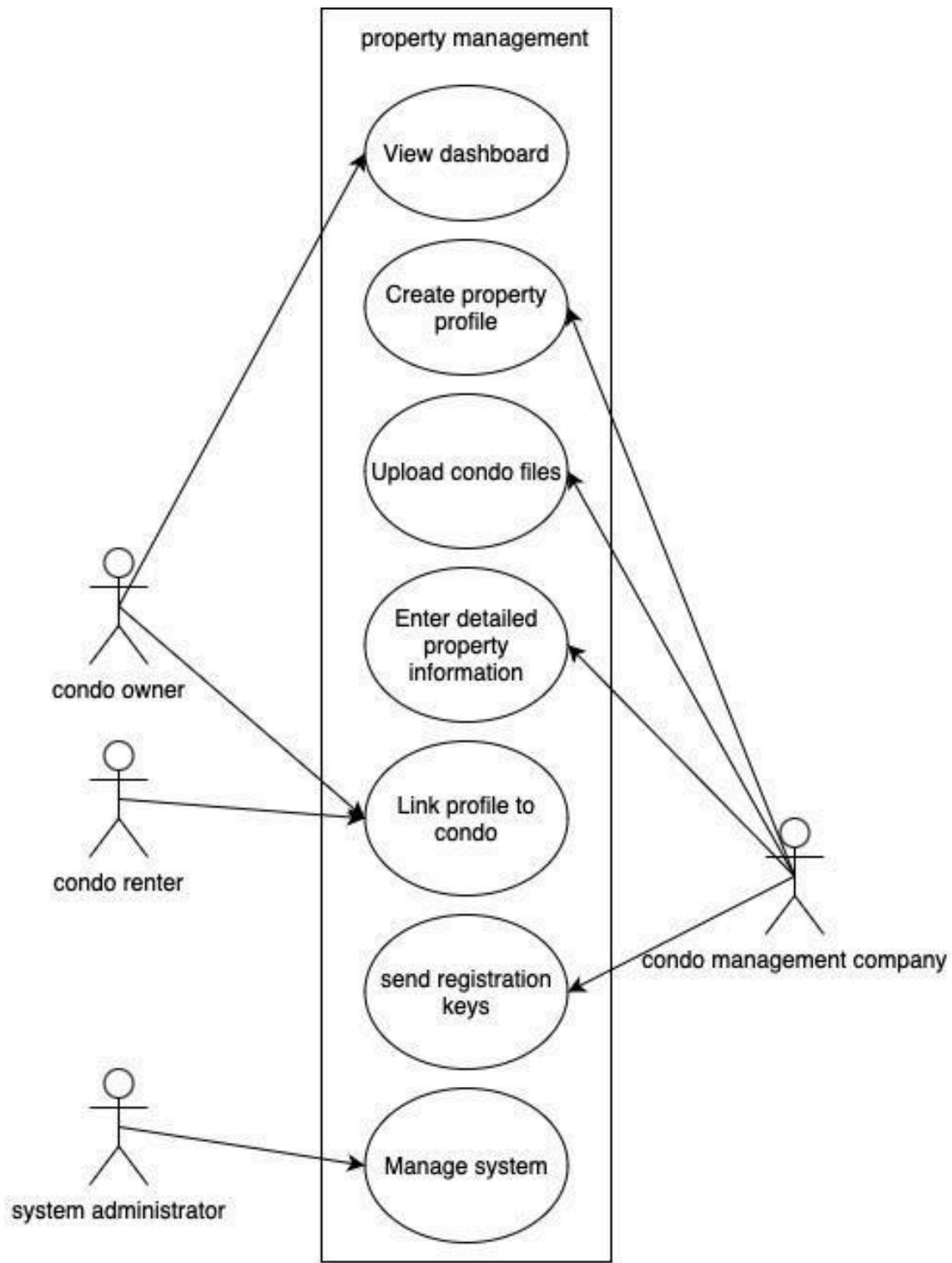


Figure 15: Use case diagram for property management operations - sprint 1

Scenario for property management use case:

1. The **Condo Management Company** creates profiles for each property, including all relevant details.
2. They upload necessary condo files to the system, ensuring they are accessible to condo owners.
3. Detailed information about each condo unit, parking spot, and locker is entered into the system.
4. The management company sends registration keys to **Condo Owners** and **Rental Users**.
5. **Condo Owners** and **Rental Users** receive these keys and use them to link their specific units with their profiles on the system.
6. **Condo Owners** can view their dashboard, which provides comprehensive information about their properties, including financial status and other personal details.
7. The **System Administrator** oversees the entire process, ensuring the system's integrity and smooth operation.

4.5.3 Known Issues with View

more associations between operations might be possible and more use case diagrams can be implemented in the next phases of the project

5 Consistency and correspondences

This chapter describes consistency requirements, recording of known inconsistencies in an AD, and the use and documentation of correspondences and correspondence rules.

5.1 Known inconsistencies

- In the activity diagrams, there is a slight inconsistency about when the registration key is provided to the client. It's given both during the sign up process and during the purchase process.
- There is a slight consistency in the naming convention between certain diagram but it's not major because it's easily understandable that both names represent the same thing.

5.2 Correspondences in the AD

- There is a GUI representing a User Profile that users can interact with, which corresponds to the User Profile class in the Class Diagram and Domain Model as initially planned.

5.3 Correspondence rules

- There needs to be a clear and consistent naming convention between the database tables and the user profiles they are related to
- Any changes in the profile schema in the database should reflect on the profile page view in the website
- The user interface elements need to align with their corresponding system functionality (The Reservation page should be linked to the functionality that is in charge of handling reservation requests)
- Any interface definition needs to have a corresponding GUI element. (The reservation interface needs to be linked to a reservation page a user can interact with)

Risk Assessment and Management Plan (RMP)

A risk is an event or condition that, if it occurs, could have a positive or negative effect on a project's objectives. Risk Management is the process of identifying, assessing, responding to, monitoring, and reporting risks. This Risk Management Plan defines how risks associated with the Condo Management System project will be identified, analyzed, and managed. It outlines how risk management activities will be performed, recorded, and monitored throughout the lifecycle of the project and provides templates and practices for recording and prioritizing risks.

The Risk Management Plan is created by the project manager in the Planning Phase of the CDC Unified Process and is monitored and updated throughout the project. The intended audience of this document is the project team, project sponsor and management.

Risk identification will involve the project team, appropriate stakeholders, and will include an evaluation of environmental factors, organizational culture and the project management plan including the project scope. Careful attention will be given to the project deliverables, assumptions, constraints, WBS, cost/effort estimates, resource plan, and other key project documents.

A Risk Management Log will be generated and updated as needed and will be stored electronically in the project library located at
https://drive.google.com/drive/u/0/folders/1Yaso52WHZf5TxmD2KLVr0bEh4Nu_fUrg

List of risks across user stories:

Sprint - 1

US-1: Public User Sign-Up

Data security: Storing user details in MongoDB poses a risk of data breaches if proper security measures are not implemented, in which case sensitive user information could be compromised.

API vulnerability: Developing a sign-up API opens up the system to potential security vulnerabilities such as injection attacks, parameter tampering, or unauthorized access. Without proper input validation and testing, attackers could exploit weaknesses in the API to gain unauthorized access to user accounts or inject malicious code.

Front-end vulnerability: Implementing the front-end React component for the sign-up form introduces the risk of malicious scripts injected through the form that could be executed within users' browsers.

Validation failure: Client-side validation for the sign-up form inputs may not be sufficient to prevent all types of incomplete or improper data submission, which can result in malformed data being sent to the server, leading to errors or inconsistencies in the user database.

US-2: Public User Login

Authentication vulnerabilities: Risk of brute force attacks, session fixation, or token hijacking which can potentially grant unauthorized access to user accounts.

Front-end vulnerability: Implementing the front-end React component for the sign-up form introduces the risk of malicious scripts injected through the form that could be executed within users' browsers.

JWT security: Issues such as insecure token storage, insufficient token expiration policies, or improper validation of tokens could lead to security breaches, including unauthorized access or session hijacking.

Password Security: If passwords are not properly encrypted using a strong cryptographic algorithm, they could be vulnerable to password cracking techniques, exposing sensitive user information.

US-3: Signed in Public User Create Profile

Data security: Expanding the user schema in MongoDB to include profile details increases the amount of personal information stored in the system. If proper security measures are not implemented, sensitive user information could be compromised.

API vulnerability: Adding API endpoints for retrieving and updating user profiles exposes the system to security risks such as injection attacks, unauthorized access, or data manipulation. Attackers could exploit weaknesses in the API to gain unauthorized access to user accounts or inject malicious code.

Front-end vulnerability: Implementing a front-end React component for designing the profile management page introduces the risk of malicious scripts injected through the form to be executed within users' browsers or Cross-Site Request Forgery (CSRF)

Insecure file upload: Without proper validation and sanitization of uploaded files, attackers could upload malicious files to the server, leading to security breaches and compromising the system

Validation failure: Client-side validation for the required profile details may not be sufficient to prevent all types of incomplete or improper data submission, which can result in malformed data being sent to the server, leading to errors or inconsistencies in the user database or injection attacks.

US-4: Enter Registration Key to Assign User Roles

Registration key security: Keys may not be securely generated, transmitted, and validated. Weaknesses in key generation algorithms or improper handling of keys during transmission could lead to guessing attacks or unauthorized access to restricted data.

Role assignment vulnerabilities: If role assignment logic is not implemented securely, attackers could manipulate the registration key or exploit vulnerabilities in the role assignment process to gain elevated privileges or access unintended features or data.

Data security: Risks of updating the database schema include unauthorized modification or access to user roles if proper access controls and encryption mechanisms are not implemented. Insecure database configurations or vulnerabilities in the schema update process could lead to data breaches or integrity issues.

UX impact: Risks include user frustration or abandonment if the registration key input process is confusing or cumbersome. Poorly designed registration key interfaces or inadequate user guidance could lead to usability issues and dissatisfaction among users.

Impact Probability	Low	Medium	High
Low	Validation failure Insecure file upload Role assignment vulnerability	API vulnerability Front-end vulnerability Validation failure Registration key security	Data security
Medium	JWT security	UX impact	Password security
High	Insufficient testing		Authentication vulnerabilities

Table [1]: Risk management chart (Sprint - 1)

Risk ID	Risk Type and Description	Risk Score	Resolved in Sprint	Strategy and Effectiveness
US-1.0	Technical Management External Budget Schedule Etc.	Low Medium High	Sprint 1	Mitigate Accept Avoid Transfer
US-1.1	Technical risk Data security	Medium	Sprint 1	Mitigate
US-1.2	External risk API Vulnerability	Low	Sprint 1	Mitigate
US-1.3	External risk Front-end vulnerability	Low	Sprint 1	Avoid
US-1.4	Technical risk Validation failure	Low	Sprint 1	Mitigate
US-1.5	Technical risk	Medium	Sprint 1	Mitigate

	Insufficient testing			
US-2.1	External risk Authentication vulnerability	High	Sprint 1	Mitigate
US-2.2	External risk Front-end vulnerability	Low	Sprint 1	Avoid
US-2.3	External risk JWT security	Low	Sprint 1	Mitigate
US-2.4	Technical risk Password security	High	Sprint 1	Mitigate
US-2.5	Technical risk Insufficient testing	Medium	Sprint 1	Mitigate
US-3.1	Technical risk Data security	Medium	Sprint 1	Mitigate
US-3.2	External risk API Vulnerability	Low	Sprint 1	Mitigate
US-3.3	External risk Front-end vulnerability	Low	Sprint 1	Avoid
US-3.4	Technical risk Insecure file upload	Low	Sprint 1	Mitigate
US-3.5	Technical risk Validation failure	Low	Sprint 1	Mitigate
US-4.1	Technical risk Registration key security	Low	Sprint 1	Mitigate
US-4.2	Technical risk Role assignment vulnerability	Low	Sprint 1	Mitigate
US-4.3	External risk Data security	Medium	Sprint 1	Mitigate
US-4.4	Technical risk Insufficient testing	Medium	Sprint 1	Mitigate
US-4.5	External risk UX-Impact	Medium	Sprint 1	Mitigate

Table [2]: List of identified risks (Sprint - 1)

Sprint - 2

US-5: Create Owner's Property Dashboard

Data privacy: Displaying property details on the dashboard may pose risks to data privacy if sensitive information such as property addresses, ownership status, or financial data is exposed to unauthorized users.

Authentication vulnerabilities: Risks of unauthorized access or data manipulation exist if authentication mechanisms for accessing the dashboard are not implemented securely. Weaknesses in authentication processes could lead to unauthorized users gaining access to sensitive property information.

Front-end vulnerability: Implementing the front-end React components for the dashboard introduces the risk of code injection or Cross-Site Scripting (XSS) attacks if proper input validation and sanitization measures are not implemented. Attackers could exploit vulnerabilities in the front-end code to execute malicious scripts within users' browsers.

Data integrity: Risks include data inconsistencies or inaccuracies on the dashboard if data validation and verification processes are not robust. Inaccurate property details or outdated information could impact the decision-making process for property owners and users accessing the dashboard.

Insufficient testing: Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability.

US-6: Create Property Profile by Condo Company Manager

Data security: Risks arise from storing property profile details in the system, including sensitive information such as property addresses, unit numbers, and owner contact details. Without proper security measures, such as encryption and access controls, there's a risk of unauthorized access or data breaches.

Authorization vulnerabilities: Risks of unauthorized access to property profile creation functionalities if access control mechanisms are not implemented effectively. Weaknesses

in authorization logic could allow unauthorized users to create or modify property profiles, leading to data manipulation or integrity issues.

Front-end vulnerability: Implementing the front-end React components for property profile creation introduces the risk of code injection or Cross-Site Scripting (XSS) attacks if proper input validation and sanitization measures are not implemented. Attackers could exploit vulnerabilities in the front-end code to execute malicious scripts within users' browsers.

Data validation failure: Risks of data inconsistency or inaccuracy in property profiles if validation processes for input data are not robust. Incomplete or improper data submission could lead to errors or inconsistencies in property details, impacting the reliability of the information stored in the system.

Insufficient testing: Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability

US-8: Condo Manager Enter Details For Every Component in the Property

Data integrity: Risks arise from entering and managing details for every component in the property, including fixtures, utilities, and amenities. Without robust data validation and verification processes, there's a risk of data inconsistencies or inaccuracies, impacting the reliability of property information stored in the system.

Authorization vulnerabilities: Risks of unauthorized access to component details entry functionalities if access control mechanisms are not implemented effectively. Weaknesses in authorization logic could allow unauthorized users to view or modify component details, leading to data manipulation or integrity issues.

Front-end vulnerability: Implementing the front-end React components for entering component details introduces the risk of code injection or Cross-Site Scripting (XSS) attacks if proper input validation and sanitization measures are not implemented. Attackers could exploit vulnerabilities in the front-end code to execute malicious scripts within users' browsers.

Data security: Risks of unauthorized access or data breaches if sensitive component details, such as maintenance schedules, inspection records, or warranty information, are not adequately protected. Proper security measures, such as encryption and access controls, are essential to prevent unauthorized access to sensitive property information.

Insufficient testing: Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components.

Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability

US-9: Implement Registration Key Distribution From Condo Managers to Public Users

Security of registration keys: Risks arise from the generation, distribution, and validation of registration keys, including risks of key guessing attacks, unauthorized access, or data breaches if keys are not securely generated, transmitted, and validated. Weaknesses in key generation algorithms or improper handling of keys during distribution could lead to unauthorized access to the system or guessing attacks.

Role assignment vulnerabilities: Risks of unauthorized access or privilege escalation if role assignment logic for registration keys is not implemented securely. Attackers could manipulate registration keys or exploit vulnerabilities in the role assignment process to gain elevated privileges or access unintended features or data.

Data security: Risks associated with updating the database schema to include registration key details, including risks of unauthorized modification or access to key data. Insecure database configurations or vulnerabilities in the schema update process could lead to data breaches or integrity issues if proper access controls and encryption mechanisms are not implemented.

Insufficient testing: Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability

Impact	Low	Medium	High
Probability			
Low	<ul style="list-style-type: none"> • Authentication Vulnerabilities 	<ul style="list-style-type: none"> • Front-end vulnerability 	<ul style="list-style-type: none"> • Security of registration keys
Medium	<ul style="list-style-type: none"> • Role assignment vulnerabilities 	<ul style="list-style-type: none"> • Data validation failure 	<ul style="list-style-type: none"> • Data security • Authorization vulnerabilities • Data integrity • Data privacy
High	<ul style="list-style-type: none"> • Insufficient testing 		

Table [3]: Risk management chart (Sprint - 2)

Risk ID	Risk Type and Description	Risk Score	Resolved in Sprint	Strategy and Effectiveness
US-5.1 #56	Data Integrity Data Privacy	High	Sprint 2	Mitigate
US-5.2 #57	Front-End Vulnerability Data Privacy	Medium	Sprint 2	Mitigate
US-5.3 #58	Front-End Vulnerability Data Privacy	Medium	Sprint 2	Mitigate
US-5.4 #59	Authentication Vulnerabilities Data Security	High	Sprint 2	Mitigate
US-5.5 #60	Authentication Vulnerabilities Front-End Vulnerability	Low	Sprint 2	Avoid
US-5.6 #62	Authentication Vulnerabilities Data Privacy	High	Sprint 2	Mitigate
US-5.7 #63	Insufficient testing	Low	Sprint 2	Avoid
US-6.1 #64	Data Security Data Validation Failure	Medium	Sprint 2	Mitigate
US-6.2 #65	Authorization Vulnerabilities Data Validation Failure	Low	Sprint 2	Mitigate
US-6.3 #66	Front-End Vulnerability Data Validation Failure	Medium	Sprint 2	Mitigate
US-6.4 #67	Front-End Vulnerability Data Validation Failure	Medium	Sprint 2	Mitigate
US-6.5	Data Validation Failure	Medium	Sprint 2	Mitigate

#68				
US-6.6 #69	Insufficient testing	Low	Sprint 2	Avoid
US-8.1 #70	Data Integrity Authorization Vulnerabilities Front-End Vulnerabilities	High	Sprint 2	Mitigate
US-8.2 #71	Data Integrity Authorization Vulnerabilities Front-End Vulnerabilities	High	Sprint 2	Mitigate
US-8.3 #72	Data Integrity Authorization Vulnerabilities Front-End Vulnerabilities	High	Sprint 2	Mitigate
US-8.4 #73	Front-End Vulnerabilities Data Integrity	Medium	Sprint 2	Mitigate
US-8.5 #74	Data integrity	Low	Sprint 2	Accept
US-8.6 #75	Insufficient Testing	Low	Sprint 2	Avoid
US-9.1 #76	Security of Registration Role Assignment Vulnerabilities	Low	Sprint 2	Mitigate
US-9.2 #77	Security of Registration	Low	Sprint 2	Mitigate
US-9.3 #78	Data security Role assignment vulnerabilities	Medium	Sprint 2	Mitigate
US-9.4 #79	Data Security	Medium	Sprint 2	Mitigate
US-9.5 #80	Insufficient Testing	Low	Sprint 2	Avoid

Table [4]: List of identified risks (Sprint - 2)

Sprint - 3

US-7: Condo Manager Upload File for Each Property

File integrity: Risks arise from uploading and managing files for each property, including risks of data corruption, loss, or unauthorized access if proper file integrity checks and access controls are not implemented. Without robust file validation mechanisms, there's a risk of uploading corrupted or malicious files, compromising the integrity of property data stored in the system.

Authorization vulnerabilities: Risks of unauthorized access to file upload functionalities if access control mechanisms are not implemented effectively. Weaknesses in authorization logic could allow unauthorized users to upload or access files, leading to data manipulation or security breaches.

Front-end vulnerability: Implementing the front-end React components for file upload introduces the risk of code injection or Cross-Site Scripting (XSS) attacks if proper input validation and sanitization measures are not implemented. Attackers could exploit vulnerabilities in the front-end code to execute malicious scripts within users' browsers.

Data security: Risks of unauthorized access or data breaches if sensitive files, such as property documents, contracts, or inspection reports, are not adequately protected. Proper security measures, such as encryption and access controls, are essential to prevent unauthorized access to sensitive property information stored in uploaded files.

Insufficient testing: Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability.

US-10: Implement Condo Fee Rate Entry

Data accuracy: Risks arise from entering and managing condo fee rates, including risks of data entry errors, inconsistencies, or inaccuracies if proper validation and verification processes are not implemented. Without adequate controls, there's a risk of entering incorrect fee rates, leading to financial discrepancies or incorrect billing for condo owners.

Authorization vulnerabilities: Risks of unauthorized access to fee rate entry functionalities if access control mechanisms are not implemented effectively. Weaknesses in authorization logic could allow unauthorized users to modify fee rates, leading to financial manipulation or disputes.

Front-end vulnerability: Implementing the front-end React components for fee rate entry introduces the risk of code injection or Cross-Site Scripting (XSS) attacks if proper input validation and sanitization measures are not implemented. Attackers could exploit vulnerabilities in the front-end code to execute malicious scripts within users' browsers.

Data security: Risks of unauthorized access or data breaches if fee rate data is not adequately protected. Proper security measures, such as encryption and access controls, are essential to prevent unauthorized access to sensitive financial information related to condo fee rates.

Insufficient testing: Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability.

US-11: Calculate and Present Condo Fees

Accuracy of calculations: Risks arise from calculating condo fees, including the potential for errors or discrepancies in fee calculations if the underlying algorithms or formulas are not implemented correctly. Incorrect fee calculations could lead to financial inaccuracies, billing discrepancies, or dissatisfaction among condo owners.

Data integrity: Risks of data corruption or loss if proper data validation and error handling mechanisms are not implemented during fee calculation processes. Without adequate safeguards, there's a risk of processing invalid or incomplete data, leading to incorrect fee calculations or system failures.

Performance issues: Risks of performance degradation or system overload during fee calculation processes, especially if dealing with large datasets or complex calculations.

Inefficient algorithms or resource-intensive calculations could lead to system slowdowns, timeouts, or unresponsive behavior, impacting user experience and system reliability.

Transparency and auditability: Risks of insufficient transparency or auditability in fee calculation processes, which could lead to mistrust or disputes among condo owners. Clear documentation and logging of fee calculation steps are essential to ensure accountability and facilitate auditing and reconciliation processes.

Insufficient testing: Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability.

US-12: Record Operational Budget and Costs

Data accuracy and completeness: Risks arise from recording operational budgets and costs, including the potential for inaccuracies or omissions if proper validation and verification processes are not in place. Without adequate controls, there's a risk of recording incorrect budget figures or missing essential cost data, leading to financial discrepancies or mismanagement of funds.

Security of financial data: Risks of unauthorized access or data breaches if operational budget and cost records are not adequately protected. Proper security measures, such as encryption, access controls, and regular security audits, are essential to prevent unauthorized access to sensitive financial information.

Compliance with regulations: Risks of non-compliance with financial regulations or reporting standards if budget and cost recording processes do not adhere to legal requirements. Failure to comply with regulations could lead to penalties, fines, or legal liabilities for the organization.

Integration with accounting systems: Risks associated with integrating budget and cost records with accounting systems or financial software. Incompatible formats, data discrepancies, or errors during data transfer could lead to synchronization issues or inconsistencies between systems, impacting financial reporting and analysis.

Insufficient documentation and audit trail: Risks of inadequate documentation or audit trail for budget and cost recording processes, which could hinder transparency, accountability, and auditability. Clear documentation of recording procedures, approvals, and changes is essential to ensure traceability and facilitate financial audits and reviews.

US-13: Generate Annual Financial Reports

Data accuracy and completeness: Risks arise from generating annual financial reports, including the potential for inaccuracies or missing data if proper validation and verification processes are not in place. Without adequate controls, there's a risk of including incorrect financial figures or omitting essential information, leading to unreliable reports and misinformed decision-making.

Timeliness of reporting: Risks of delays or missed deadlines in generating annual financial reports, especially if the reporting process is manual or reliant on inefficient workflows. Delays in reporting could impact stakeholders' ability to make informed decisions or comply with regulatory requirements, leading to reputational damage or legal consequences.

Compliance with reporting standards: Risks of non-compliance with financial reporting standards or regulatory requirements if annual financial reports do not adhere to prescribed formats or guidelines. Failure to comply with reporting standards could result in penalties, fines, or legal liabilities for the organization.

Data security and confidentiality: Risks of unauthorized access or disclosure of sensitive financial information contained in annual reports. Proper security measures, such as encryption, access controls, and data anonymization, are essential to protect confidential financial data from unauthorized disclosure or misuse.

Auditability and transparency: Risks of insufficient auditability or transparency in the generation process of annual financial reports, which could hinder accountability and scrutiny. Clear documentation of reporting procedures, data sources, and assumptions is necessary to facilitate financial audits and reviews, ensuring the integrity and reliability of the reports.

Impact Probability	Low	Medium	High
Low	<ul style="list-style-type: none"> • Performance Issues 	<ul style="list-style-type: none"> • Front-end Vulnerability • Transparency and Auditability 	<ul style="list-style-type: none"> • Data Integrity • Timeliness of Reporting
Medium	<ul style="list-style-type: none"> • Insufficient Testing • Integration with Accounting Systems 	<ul style="list-style-type: none"> • File Integrity • Compliance with Regulations 	<ul style="list-style-type: none"> • Authorization Vulnerabilities • Data Accuracy • Data Accuracy and Completeness • Compliance with Reporting Standards
High	<ul style="list-style-type: none"> • Accuracy of Calculations 	<ul style="list-style-type: none"> • Insufficient Documentation and Audit Trail • Auditability and Transparency 	<ul style="list-style-type: none"> • Data Security • Security of Financial Data • Data Security and Confidentiality

Table [5]: Risk management chart (Sprint - 3)

Risk ID	Risk Type and Description	Risk Score	Resolved in Sprint	Strategy and Effectiveness
US-7.1 #	File integrity	Medium	Sprint 3	Mitigate
US-7.2 #	Authorization vulnerabilities	High	Sprint 3	Mitigate
US-7.3 #	Front-end vulnerability	Low	Sprint 3	Mitigate
US-7.4 #	Data security	High	Sprint 3	Mitigate
US-7.5 #	Insufficient testing	Low	Sprint 3	Avoid
US-10.1 #	Data accuracy	High	Sprint 3	Mitigate
US-10.2 #	Authorization vulnerabilities	High	Sprint 3	Mitigate
US-10.3 #	Front-end vulnerability	Low	Sprint 3	Mitigate

US-10.4 #	Data security	High	Sprint 3	Mitigate
US-10.5 #	Insufficient testing	Low	Sprint 3	Avoid
US-11.1 #	Accuracy of calculations	Low	Sprint 3	Mitigate
US-11.2 #	Data integrity	Medium	Sprint 3	Mitigate
US-11.3 #	Performance issues	Low	Sprint 3	Avoid
US-11.4 #	Transparency and auditability	Low	Sprint 3	Mitigate
US-11.5 #	Insufficient testing	Low	Sprint 3	Avoid
US-12.1 #	Security of financial data	High	Sprint 3	Mitigate
US-12.2 #	Data accuracy and completeness	High	Sprint 3	Mitigate
US-12.3 #	Compliance with regulations	Medium	Sprint 3	Mitigate
US-12.4 #	Integration with accounting systems	Low	Sprint 3	Mitigate
US-12.5 #	Insufficient documentation and audit trail	High	Sprint 3	Avoid
US-13.1 #	Timeliness of reporting	Medium	Sprint 3	Accept
US-13.2 #	Data accuracy and completeness	High	Sprint 3	Accept
US-13.3 #	Compliance with reporting standards	High	Sprint 3	Mitigate
US-13.4 #	Data security and confidentiality	High	Sprint 3	Mitigate
US-13.5 #	Auditability and transparency	High	Sprint 3	Avoid

Table [6]: List of identified risks (Sprint - 3)

Sprint - 4

US-14: Set Up Common Facilities Management

Infrastructure readiness: Risks associated with setting up common facilities management include the readiness of infrastructure such as physical spaces, equipment, and utilities. Delays or deficiencies in infrastructure preparation could hinder the effective management of common facilities, leading to operational disruptions or inefficiencies.

Resource allocation: Risks related to allocating sufficient resources, including personnel, budget, and technology, for managing common facilities. Inadequate resources may result in suboptimal facility management practices, compromising the quality of service provided to occupants or users of the facilities.

Compliance with regulations: Risks of non-compliance with regulatory requirements or building codes in the setup of common facilities management. Failure to adhere to regulations could result in fines, penalties, or legal liabilities for the organization, as well as safety or health hazards for occupants.

Integration with existing systems: Risks associated with integrating common facilities management systems with existing property management or operational systems. Compatibility issues, data discrepancies, or technical challenges during integration could disrupt facility management processes or data flow, affecting service delivery and decision-making.

User training and adoption: Risks of insufficient user training or resistance to adopting new facility management processes or technologies. Inadequate training programs or lack of user engagement may impede the effective utilization of common facilities management systems, limiting their potential benefits and efficiency gains.

US-15: Calender-Like Interface for Facility Reservation

User interface design: Risks associated with the design and implementation of a calendar-like interface for facility reservation. Poor user interface design may lead to usability issues, such as difficulty in navigating the calendar, confusion in booking

facilities, or inconsistencies in displaying reservation information, resulting in user dissatisfaction and decreased adoption.

Concurrency and conflicts: Risks of managing concurrent reservation requests and resolving conflicts in booking facilities. Without proper concurrency control mechanisms, multiple users may attempt to reserve the same facility simultaneously, leading to conflicts or overbooking situations that could disrupt scheduling and cause frustration among users.

Data integrity and security: Risks related to ensuring the integrity and security of reservation data in the calendar interface. Vulnerabilities such as data corruption, unauthorized access, or data loss could compromise the accuracy and confidentiality of reservation information, impacting the reliability and trustworthiness of the system.

Integration with backend systems: Risks associated with integrating the calendar interface with backend systems for facility management and reservation tracking. Compatibility issues, data synchronization errors, or communication failures between the frontend interface and backend databases may result in inconsistencies or inaccuracies in reservation data, affecting the overall functionality and usability of the system.

Scalability and performance: Risks of scalability and performance bottlenecks in handling large volumes of reservation requests or concurrent user interactions. Inadequate system scalability or inefficient resource utilization may lead to slow response times, system crashes, or degraded performance during peak usage periods, negatively impacting user experience and satisfaction.

US-16: Display Real-Time Availability in Reservation System

Data synchronization: Risks associated with ensuring real-time synchronization of availability data across the reservation system. Inaccurate or delayed updates to availability status may lead to conflicts or inconsistencies in reservation requests, resulting in double bookings or missed opportunities for users to reserve facilities, ultimately impacting user satisfaction and system reliability.

Concurrency control: Risks of managing concurrent access to availability information by multiple users. Without proper concurrency control mechanisms, simultaneous reservation requests may lead to conflicts or race conditions, where users may inadvertently book the same facility or encounter errors due to inconsistent data states, leading to frustration and confusion.

Performance optimization: Risks related to optimizing system performance for real-time availability updates. Processing and propagating availability changes in a timely manner require efficient algorithms and resource utilization to minimize latency and ensure responsiveness, especially during periods of high user activity or frequent reservation updates.

User interface responsiveness: Risks associated with displaying real-time availability information to users in the reservation system interface. Ensuring that availability status updates are promptly reflected in the user interface without perceptible delays or inconsistencies is crucial for providing a seamless and intuitive booking experience, as any lag or inaccuracies may undermine user trust and satisfaction.

Data integrity and security: Risks of maintaining the integrity and security of availability data in the reservation system. Vulnerabilities such as unauthorized access, data tampering, or system breaches could compromise the accuracy and confidentiality of availability information, leading to unauthorized bookings, data loss, or privacy violations, posing significant risks to system reliability and user trust.

US-17: Manage Reservations on a First-Come-First-Serve Basis

Fairness and equity: Risks associated with ensuring fair and equitable treatment of users in the reservation process. Implementing a first-come-first-serve (FCFS) policy may lead to challenges in prioritizing reservation requests based on their submission time, potentially resulting in dissatisfaction or disputes among users who perceive the allocation of facilities as unfair or biased.

Concurrency and contention: Risks of managing concurrent reservation requests and potential contention for limited resources. In scenarios where multiple users submit reservation requests simultaneously for the same facility or time slot, conflicts may arise, requiring robust concurrency control mechanisms to prevent double bookings, race conditions, or inconsistent outcomes, which could undermine user trust and system reliability.

System scalability: Risks related to the scalability of the reservation system to handle increasing demand and workload. As the number of reservation requests grows, the system must efficiently process and manage incoming requests while maintaining responsiveness and performance, without experiencing bottlenecks, degradation, or service disruptions that could impact user experience and satisfaction.

Feedback and communication: Risks associated with providing timely feedback and communication to users regarding the status of their reservation requests. Ensuring clear and transparent communication channels for notifying users about the outcome of their reservation attempts, including confirmations, rejections, or waitlist notifications, is essential for managing user expectations and fostering trust in the reservation process.

Compliance and regulations: Risks of compliance with relevant policies, regulations, or contractual obligations governing reservation management. Adhering to legal requirements, organizational policies, or contractual agreements regarding reservation procedures, data privacy, or accessibility standards is critical for mitigating legal liabilities, reputational risks, and potential sanctions or penalties that could arise from non-compliance.

US-18: Design Role-Based Access Control System

Complexity in role definition: Risks associated with defining and managing roles within the access control system. Designing a role-based access control (RBAC) system requires careful consideration of user roles, permissions, and hierarchical structures to ensure that access rights are properly assigned and enforced. Inadequate role definition or misalignment between roles and organizational responsibilities may lead to access inconsistencies, security vulnerabilities, or administrative overhead, impacting system integrity and usability.

Granularity and flexibility: Risks related to achieving the right balance between granularity and flexibility in access control policies. Striking a balance between fine-grained access control to enforce least privilege principles and flexible role assignment to accommodate dynamic user roles and evolving organizational needs is challenging. Overly restrictive access policies may hinder user productivity, while overly permissive policies may increase the risk of unauthorized access, data breaches, or privilege escalation, necessitating careful policy design and review.

Access control enforcement: Risks associated with the effective enforcement of access control policies across the system. Implementing robust mechanisms for access validation, authorization, and audit logging is essential for preventing unauthorized access attempts, privilege abuse, or security breaches. Failure to properly enforce access controls may result in data exposure, unauthorized modifications, or unauthorized actions by malicious insiders or external attackers, compromising system confidentiality, integrity, and availability.

Integration and interoperability: Risks of integrating the RBAC system with existing infrastructure, applications, and identity management solutions. Ensuring seamless interoperability and compatibility with diverse platforms, protocols, and authentication mechanisms requires thorough integration testing, validation, and configuration management. Incompatibilities, misconfigurations, or interoperability issues may disrupt access control operations, user authentication, or provisioning workflows, leading to service disruptions, user frustration, or security vulnerabilities.

Compliance and governance: Risks related to compliance with regulatory requirements, industry standards, or organizational policies governing access control practices. Adhering to data protection regulations, privacy laws, or industry-specific mandates regarding access rights, user consent, and data handling is crucial for mitigating legal risks, reputational damage, or financial penalties associated with non-compliance.

Implementing effective access controls, audit trails, and governance frameworks is essential for demonstrating compliance, accountability, and risk management practices to stakeholders and regulatory authorities.

US-19: Predefined Roles and Permissions Setup

Role definition accuracy: Risks associated with accurately defining predefined roles and permissions to align with the needs and responsibilities of different user groups within the system. The definition of roles must be comprehensive, clear, and reflective of organizational requirements to ensure that users are assigned appropriate access rights and privileges. Risks include ambiguities in role definitions, inconsistencies in permission assignments, or inadequate consideration of user roles' functional requirements, which could lead to unauthorized access, data breaches, or operational inefficiencies.

Permission granularity and control: Risks related to the granularity and control of permissions assigned to predefined roles within the system. The permission model must support fine-grained access control to restrict users' actions and privileges based on their roles and responsibilities. Risks include overly permissive permissions, insufficient segregation of duties, or lack of controls to enforce least privilege principles, resulting in elevated security risks, data exposure, or regulatory non-compliance.

Role-based access control (RBAC) implementation: Risks associated with the implementation of role-based access control mechanisms to enforce predefined roles and permissions effectively. The RBAC system must be robust, scalable, and configurable to accommodate changes in user roles, organizational structures, and access requirements over time. Risks include implementation flaws, vulnerabilities in access control logic, or misconfigurations that could result in access violations, privilege escalation, or unauthorized actions by users, compromising system security and integrity.

User role mapping and assignment: Risks related to accurately mapping user identities to predefined roles and managing role assignments dynamically. The role assignment process must be automated, auditable, and consistent to ensure that users are granted the appropriate level of access based on their roles, responsibilities, and organizational affiliations. Risks include errors in role mapping, manual oversight, or inconsistencies in access provisioning, leading to unauthorized access, data leaks, or compliance violations.

Role evolution and maintenance: Risks associated with the evolution and maintenance of predefined roles and permissions over time. The role definition process must be iterative, adaptive, and responsive to changing organizational needs, regulatory requirements, and security threats. Risks include role proliferation, role creep, or lack of governance mechanisms to review, update, and deprecate roles periodically, resulting in role redundancy, confusion, or inefficiencies in access management and enforcement. Regular

audits, role lifecycle management, and role-based training programs can help mitigate these risks and ensure the continued effectiveness of the role-based access control system.

Impact Probability	Low	Medium	High
Low	<ul style="list-style-type: none"> • Infrastructure readiness • Fairness and equity 	<ul style="list-style-type: none"> • Resource allocation • Feedback and communication 	<ul style="list-style-type: none"> • Granularity and flexibility • User role mapping and assignment
Medium	<ul style="list-style-type: none"> • Compliance with regulations • Integration with backend systems • Complexity in role definition 	<ul style="list-style-type: none"> • User interface design • Data synchronization • System scalability • Role evolution and maintenance 	<ul style="list-style-type: none"> • Concurrency and conflicts • Access control enforcement • Role-based access control (RBAC) implementation
High	<ul style="list-style-type: none"> • Integration with existing systems • User training and adoption • User interface responsiveness 	<ul style="list-style-type: none"> • Scalability and performance • Integration and interoperability 	<ul style="list-style-type: none"> • Data integrity and security

Table [7]: Risk management chart (Sprint - 4)

Risk ID	Risk Type and Description	Risk Score	Resolved in Sprint	Strategy and Effectiveness
US-14.1 #	Infrastructure readiness	Low	Sprint 4	Accept
US-14.2 #	Resource allocation	Low	Sprint 4	Mitigate
US-14.3 #	Compliance with regulations	Low	Sprint 4	Avoid
US-14.4 #	Integration with existing systems	Medium	Sprint 4	Accept
US-14.5 #	User training and adoption	Medium	Sprint 4	Transfer

US-15.1 #	User interface design	Medium	Sprint 4	Mitigate
US-15.2 #	Concurrency and conflicts	High	Sprint 4	Avoid
US-15.3 #	Data integrity and security	High	Sprint 4	Avoid
US-15.4 #	Integration with backend systems	Low	Sprint 4	Mitigate
US-15.5 #	Scalability and performance	High	Sprint 4	Transfer
US-16.1 #	Data synchronization	Medium	Sprint 4	Mitigate
US-16.2 #	Concurrency control	High	Sprint 4	Mitigate
US-16.3 #	Performance optimization	High	Sprint 4	Accept
US-16.4 #	User interface responsiveness	Medium	Sprint 4	Mitigate
US-16.5 #	Data integrity and security	High	Sprint 4	Avoid
US-17.1 #	Fairness and equity	Low	Sprint 4	Transfer
US-17.2 #	Concurrency and contention	High	Sprint 4	Mitigate
US-17.3 #	System scalability	High	Sprint 4	Transfer
US-17.4 #	Feedback and communication	Low	Sprint 4	Avoid
US-17.5 #	Compliance and regulations	Low	Sprint 4	Avoid
US-18.1 #	Complexity in role definition	Low	Sprint 4	Avoid
US-18.2 #	Granularity and flexibility	Medium	Sprint 4	Mitigate
US-18.3 #	Access control enforcement	High	Sprint 4	Mitigate
US-18.4 #	Integration and interoperability	High	Sprint 4	Transfer
US-18.5 #	Compliance and governance	Low	Sprint 4	Avoid
US-19.1 #	Role definition accuracy	Medium	Sprint 4	Avoid
US-19.2 #	Permission granularity and control	Medium	Sprint 4	Mitigate

US-19.3 #	Role-based access control implementation	High	Sprint 4	Mitigate
US-19.4 #	User role mapping and assignment	Medium	Sprint 4	Mitigate
US-19.5 #	Role evolution and maintenance	Medium	Sprint 4	Mitigate

Table [8]: List of identified risks (Sprint - 4)

Sprint - 5

US-20: Manage Employee Roles Interface

Role assignment errors: Risks associated with inaccuracies or inconsistencies in assigning roles to employees through the management interface. The interface for managing employee roles may be prone to user errors, such as selecting incorrect roles, assigning excessive permissions, or overlooking critical access requirements. These errors could result in unauthorized access, data breaches, or compliance violations, impacting system security and integrity.

Complexity in role management: Risks related to the complexity of managing employee roles effectively through the interface. As the organization grows or undergoes structural changes, the number of roles, permissions, and access requirements may increase, leading to greater complexity in role management. Without intuitive and efficient interfaces for role assignment, modification, or revocation, administrators may struggle to maintain accurate and up-to-date role assignments, increasing the risk of access control errors or inconsistencies.

Access control misconfigurations: Risks of misconfiguring access control settings or policies through the management interface. Administrators responsible for configuring employee roles may inadvertently misconfigure access control settings, such as granting excessive privileges, applying incorrect permissions, or overlooking security best practices. These misconfigurations could create security vulnerabilities, exploit opportunities, or compliance gaps, exposing the system to unauthorized access, data leakage, or malicious activities.

Auditability and accountability: Risks associated with the lack of auditability and accountability features in the role management interface. Without robust audit trails, logging mechanisms, or access control monitoring capabilities, it may be challenging to track changes to role assignments, identify unauthorized modifications, or detect suspicious activities. Inadequate auditability and accountability features could hinder incident response efforts, forensic investigations, or compliance audits, reducing the organization's ability to detect, mitigate, or prevent security incidents.

User training and awareness: Risks stemming from insufficient user training or awareness regarding role management practices and interface usage. Users responsible for managing employee roles may lack the necessary knowledge, skills, or awareness to perform their roles effectively, increasing the likelihood of errors, omissions, or misconfigurations in

role assignments. Providing comprehensive training, documentation, or guidance on role management best practices and interface functionalities is essential for mitigating user-related risks and enhancing overall system security and compliance.

US-21: Secure and Scalable System

Security vulnerabilities: Risks associated with potential security vulnerabilities in the system architecture, design, or implementation. As the system scales to accommodate a larger user base or increased data processing requirements, it may become more susceptible to security threats, such as unauthorized access, data breaches, or denial-of-service attacks. Failure to address security vulnerabilities proactively through measures such as threat modeling, code reviews, or penetration testing could expose the system to exploitation, compromising sensitive data or disrupting system operations.

Data privacy and compliance: Risks related to maintaining data privacy and compliance with regulatory requirements as the system scales. Handling sensitive user data or processing personal information imposes legal and regulatory obligations regarding data protection, privacy, and security. Failure to implement robust data privacy controls, encryption mechanisms, or compliance monitoring measures could result in non-compliance penalties, legal liabilities, or reputational damage to the organization.

Performance bottlenecks: Risks of performance bottlenecks or degradation as the system scales to handle increased user traffic or data volume. Inadequate capacity planning, resource allocation, or system optimization strategies may lead to performance issues, such as slow response times, system downtime, or service interruptions. Mitigating performance risks requires careful monitoring, performance testing, and optimization efforts to ensure that the system can scale effectively while maintaining optimal performance levels.

Infrastructure dependencies: Risks associated with dependencies on external infrastructure or third-party services for system scalability. Relying on external providers for cloud hosting, storage, or other infrastructure components introduces dependencies that may impact system availability, reliability, or performance. Service outages, network disruptions, or changes in service-level agreements (SLAs) could affect the system's ability to scale seamlessly, requiring contingency plans, redundancy measures, or alternative deployment strategies to mitigate risks and ensure uninterrupted service delivery.

Cost management: Risks related to managing costs associated with system scalability and resource provisioning. Scaling infrastructure resources, such as compute instances, storage, or network bandwidth, to accommodate increased demand or workload fluctuations can incur additional costs. Failure to anticipate, monitor, or control scalability-related expenses could lead to budget overruns, financial constraints, or

resource shortages, affecting the organization's ability to sustainably scale the system while managing operational costs effectively. Implementing cost optimization strategies, such as usage monitoring, resource allocation policies, or pricing negotiations with service providers, is essential for mitigating cost-related risks and ensuring long-term scalability and financial viability.

US-22: Owners can Submit Requests for Services from Employees

Service request processing: Risks associated with the processing and management of service requests submitted by owners to employees. As owners submit requests for various services, such as maintenance, repairs, or amenities, there is a risk of delays, miscommunication, or errors in processing these requests efficiently. Failure to establish clear procedures, workflows, or communication channels for handling service requests could result in delays in service delivery, dissatisfaction among owners, or disruptions in property management operations.

Data security and privacy: Risks related to the security and privacy of owner-submitted service requests and associated data. Service requests may contain sensitive information, such as personal details, property locations, or service preferences, which must be handled securely to prevent unauthorized access, data breaches, or privacy violations. Inadequate data encryption, access controls, or data handling practices could expose owner information to unauthorized parties, compromising confidentiality and trust in the system.

Resource allocation and scheduling: Risks of inefficient resource allocation and scheduling for fulfilling service requests submitted by owners. Property management teams must allocate resources, such as personnel, equipment, or materials, effectively to address service requests in a timely manner. Failure to optimize resource allocation, prioritize requests based on urgency or importance, or maintain accurate scheduling records could lead to inefficiencies, service delays, or conflicts in resource utilization, impacting owner satisfaction and property operations.

Communication breakdowns: Risks of communication breakdowns between owners, employees, and management teams involved in the service request process. Effective communication is essential for conveying service request details, status updates, or resolution actions between stakeholders, ensuring transparency, accountability, and responsiveness in service delivery. Lack of clear communication channels, protocols, or escalation procedures could result in misunderstandings, unresolved issues, or dissatisfaction among owners and employees, undermining trust and collaboration within the property management ecosystem.

Service quality and performance: Risks related to maintaining service quality and performance standards while fulfilling owner requests. Property management teams must ensure that service requests are addressed promptly, accurately, and satisfactorily to meet owner expectations and contractual obligations. Failure to deliver high-quality services, adhere to service level agreements (SLAs), or address owner feedback and complaints effectively could lead to reputational damage, loss of business, or legal liabilities for the property management company. Implementing robust quality assurance measures, performance metrics, and service improvement initiatives is crucial for mitigating risks and enhancing owner satisfaction and loyalty.

US-23: Employee (Paul) Accessing Assigned Request From Condo Owners

Data Security: There's a risk of unauthorized access to sensitive information if proper access controls are not implemented. If Paul can access requests beyond his assigned scope, it could lead to privacy breaches and data leaks.

Confidentiality Breach: If Paul can view requests from condo owners that are meant to be confidential or restricted to other employees or managers, it could result in a breach of confidentiality.

Data Integrity: If Paul has the ability to modify or tamper with requests that he shouldn't have access to, it could lead to data integrity issues and affect the accuracy and reliability of the information stored in the system.

Miscommunication: Allowing Paul to access requests from condo owners may result in miscommunication or misunderstandings if he acts on requests that are not within his assigned responsibilities or if he provides incorrect information.

US-24: Create User Notification Page or Updates

Notification delivery reliability: Risks associated with ensuring the reliable delivery of notifications to users regarding important updates, announcements, or events. The notification system must be robust enough to handle high volumes of notifications efficiently and deliver them to users in a timely manner. Risks include system failures, network issues, or software bugs that could result in delays, missed notifications, or inconsistent delivery, leading to user frustration and dissatisfaction with the platform.

Content accuracy and relevance: Risks of providing users with inaccurate, outdated, or irrelevant information through notifications. The content of notifications, including updates, alerts, or reminders, must be carefully curated and validated to ensure its accuracy, relevance, and usefulness to users. Risks include errors in content generation, inappropriate targeting or segmentation of notifications, or lack of mechanisms to verify

the validity and currency of information, which could undermine user trust and confidence in the platform.

User privacy and data protection: Risks related to the privacy and security of user data and preferences used for delivering notifications. The notification system must comply with privacy regulations and industry standards to protect user information from unauthorized access, disclosure, or misuse. Risks include data breaches, unauthorized data sharing, or inadequate safeguards for storing and transmitting sensitive user data, leading to privacy violations, legal liabilities, or reputational damage for the platform.

Notification overload and fatigue: Risks of overwhelming users with excessive or intrusive notifications, leading to notification fatigue and disengagement. The frequency, timing, and relevance of notifications must be carefully managed to avoid overwhelming users with unnecessary or irrelevant information. Risks include overzealous notification strategies, lack of user control over notification settings, or failure to respect user preferences and boundaries, resulting in user dissatisfaction, decreased engagement, or even opt-out from the notification system.

Impact	Low	Medium	High
Probability			
Low	<ul style="list-style-type: none"> Notification overload and fatigue 	<ul style="list-style-type: none"> Cost management User training and awareness 	<ul style="list-style-type: none"> Infrastructure dependencies Data security and privacy
Medium	<ul style="list-style-type: none"> Role assignment errors Content accuracy and relevance 	<ul style="list-style-type: none"> Communication breakdowns Data privacy and compliance Auditability and accountability 	<ul style="list-style-type: none"> Security vulnerabilities Service quality and performance
High	<ul style="list-style-type: none"> Access control misconfigurations Notification delivery reliability 	<ul style="list-style-type: none"> Service request processing 	<ul style="list-style-type: none"> Performance bottlenecks

Table [9]: Risk management chart (Sprint - 5)

Risk ID	Risk Type and Description	Risk Score	Resolved in Sprint	Strategy and Effectiveness
US-20.1 #	Role assignment errors	Low	TBD	Accept

US-20.2 #	Complexity in role management	Medium	TBD	Avoid
US-20.3 #	Access control misconfigurations	High	TBD	Mitigate
US-20.4 #	Auditability and accountability, User training and awareness	High	TBD	Mitigate
US-21.1 #	Security vulnerabilities	High	TBD	Mitigate
US-21.2 #	Data privacy and compliance	High	TBD	Mitigate
US-21.3 #	Performance bottlenecks	Medium	TBD	Avoid
US-21.4 #	Infrastructure dependencies, Cost Management	High	TBD	Mitigate
US-22.1 #	Data security and privacy	High	TBD	Mitigate
US-22.2 #	Communication breakdowns	Medium	TBD	Mitigate
US-22.3 #	Service request processing	Medium	TBD	Mitigate
US-22.4 #	Resource allocation and scheduling	High	TBD	Mitigate
US-22.5 #	Service quality and performance	Low	TBD	Avoid
US-23.1 #	Confidentiality Breach	Low	TBD	Mitigate
US-23.2 #	Miscommunication	Meidum	TBD	Avoid
US-23.3 #	Data Security	High	TBD	Mitigate
US-23.4 #	Miscommunication	Medium	TBD	Mitigate
US-23.5 #	Data Integrity	High	TBD	Mitigate
US-24.1 #	Notification delivery reliability	Low	TBD	Avoid
US-24.2 #	Notification overload and fatigue	Low	TBD	Accept
US- 24.3 #	Content accuracy and relevance	Medium	TBD	Mitigate
US-24.4 #	User privacy and data protection	High	TBD	Mitigate
US-24.5 #	Content accuracy and relevance	Medium	TBD	Mitigate

Table [10]: List of identified risks (Sprint - 5)

Retrospective

Sprint 4's Triumphs and Trials

Introduction

During Sprint 4 of the Condo Management Systems project, our team focused on improving the reservation and access control features of our condo management application. The sprint aimed to implement robust functionalities for managing common facilities, a calendar-like interface for facility reservations, real-time availability updates, and a role-based access control system. We strived to align our development efforts with the main features required, ensuring all functionalities are user-friendly and efficient.

What went wrong

1 - Challenges in MongoDB Schema Design for Common Facilities

Our team faced significant challenges with the initial MongoDB schema design intended for storing data about common facilities. The schema was not optimized, which led to redundant data storage and inefficient data retrieval operations. This inefficiency notably slowed down system performance when handling queries for multiple facilities. The team addressed this by refactoring the schema, which improved data handling and system response times. Going forward, we plan to incorporate more rigorous planning and peer review stages in our development process to identify and rectify such issues earlier.

2 - Complexities with Role-Based Access Control Implementation

The integration of RBAC with existing systems introduced several challenges that required additional resources and time, thus impacting our sprint timeline. The development team responded by enhancing collaboration with our security specialists and conducting extensive testing to ensure both functionality and security.

What went right

1 - Successful Implementation of the Calendar-like Interface

One of the major successes in Sprint 4 was the development and implementation of a calendar-like interface for managing reservations of common facilities. This success was largely due to effective teamwork and clear task division among our front-end developers. The interface was integrated smoothly with the backend, resulting in a functional and user-friendly reservation system.

2 - Efficient Real-Time Updates for Facility Availability

Our backend team excelled in utilizing APIs to fetch and update the reservation data accurately, which greatly enhanced user experience by providing up-to-date availability information and reducing booking conflicts. This was made possible by our strong focus on robust backend development and leveraging advanced technologies to ensure responsiveness and reliability of the application.

3 - Robust Validation and Error Handling in API Development

We implemented comprehensive input validation and sophisticated error handling mechanisms, which played a crucial role in minimizing data-related errors and improving the overall stability of the application. The effectiveness of these enhancements was a direct result of our team's commitment to rigorous testing and detailed code reviews. This meticulous approach ensured that all API endpoints functioned reliably before being integrated into the live environment.

4 - Effective Use of Agile Practices for Task Management

Our team's application of agile methodologies significantly contributed to our success in Sprint 4. By consistently implementing agile practices, such as daily stand-ups, sprint planning, and retrospective meetings, we maintained a high level of team alignment and transparency. This approach allowed us to adapt swiftly to changes and feedback, ensuring that all team members were aware of priorities and task progress.

Conclusion

Sprint 4 was a challenging yet fruitful phase in our project development cycle. We learned valuable lessons in handling complex system integrations and the importance of robust initial planning. The successes in user interface development and real-time data handling have set a strong foundation for future enhancements. The main takeaway for the team is the critical role of clear communication and early testing in managing complex features effectively.

Sprint 5 Release Plan

Sprint (5) Release Plan

User story 20 – Manage Employee Roles Interface: 3 USP

- Design and implement a React component for managing employee roles within the condo management platform.
- Integrate this component with the Django backend to fetch available roles and permissions and to assign roles to employees.
- Create a user-friendly UI that allows for easy assignment of roles and modification of permissions for individual employees.
- Ensure the interface supports adding new employees to the system and changing their roles as necessary.

User Story 21 – Secure and Scalable System: 5 USP

- Integrate the RBAC system with the existing authentication mechanisms to enforce secure access control based on assigned roles.
- Implement logging and auditing features to track changes to roles and permissions, ensuring compliance and facilitating reviews.
- Test the system for scalability, ensuring it can handle an increasing number of employees and roles as the company grows.
- Review and update security protocols to protect against unauthorized access and ensure data privacy.

User Story 22 – Owners Can Submit Requests for Services from Employees: 3 USP

- Extend the MongoDB schema to include request types and details.
- Implement API endpoints in Django for request submission and assignment based on request type.
- Design and develop a request submission form in React.
- Create a system in Django to automatically assign requests to employees based on their roles and the type of request.

- Update the role-based access control to include permissions for managing different types of requests.

User Story 23 – Employee (Paul) Accessing Assigned Request from Condo Owners: 7 USP

- Develop a dashboard in React for employees to view their assigned requests.
- Implement filtering and sorting of requests in the dashboard based on type, date, and status.
- Create Django views and API endpoints for updating the status of requests.
- Integrate notification functionality in Django to alert employees of new assignments.
- Adjust MongoDB schema to track request status updates and notifications.

User Story 24 – Create User Notification Page For Updates: 5 USP

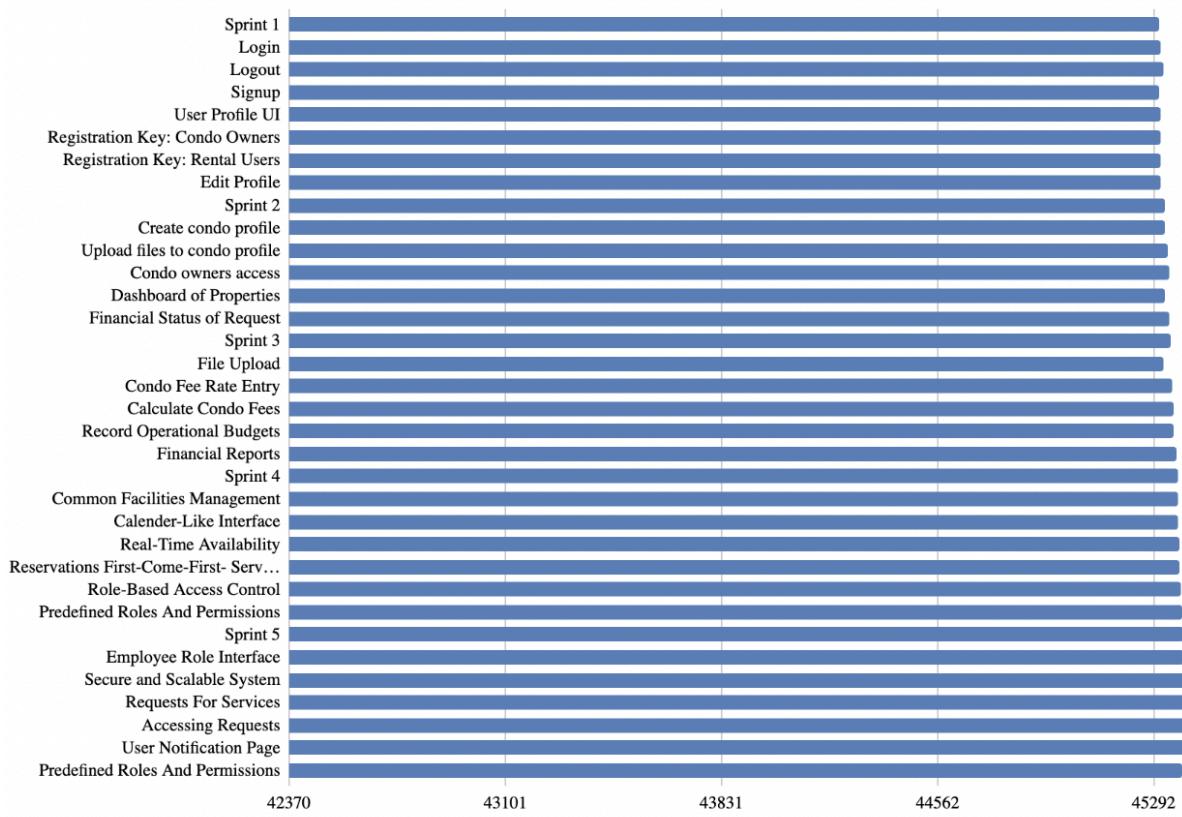
- Design and implement a notifications schema in MongoDB to store notification data related to user activities.
- Develop a notification system in Django that triggers updates for activities such as new requests, status changes, and assignments.
- Create a React component for the notification page that displays a list of notifications, sorted by the most recent.
- Integrate the React notification page with Django backend to fetch and display notifications.
- Update the role-based access control system to ensure users only receive notifications relevant to their role and activities.

User Story ID	User Story Points (USP)	Priority	Status
US-20 (#32)	3 USP	• Medium	• Backlog
US-21 (#33)	5 USP	• High	• Backlog
US-22 (#18)	3 USP	• Medium	• Backlog
US-23 (#19)	7 USP	• Medium	• Backlog

US-24 (#20)	5 USP	• Low	• Backlog
Total USP	23 USP		

Task Name	Responsible	Start	End	Days	Status
Sprint 1	Joud	1/18	2/7	20	Complete
Login	Joud	1/25	2/3	9 days	Complete
Logout	Jeremy	2/1	2/7	6 days	Complete
Signup	Yashi	1/18	1/25	7 days	Complete
User Profile UI	Patrick	1/25	2/3	9 days	Complete
Registration Key: Condo Owners	Nicholas	1/26	2/7	12 days	Complete
Registration Key: Rental Users	Nicholas	1/26	2/7	12 days	Complete
Edit Profile	Joud	1/25	2/7	13 days	Complete
Sprint 2	Jeremy	2/8	2/28	20 days	Complete
Create condo profile	Patrick	2/8	2/9	1 days	Complete
Upload files to condo profile	Ishai	2/17	2/21	4 days	Complete
Condo owners access	Alex	2/22	2/24	2 days	Complete
Dashboard of Properties	Eden	2/9	2/16	7 days	Complete
Financial Status of Request	Vithu	2/22	2/28	6 days	Complete
Sprint 3	Ishai	2/29	3/21	21 days	Complete
File Upload	Patrick	2/2	3/3	3 days	Complete
Condo Fee Rate Entry	Yashi	3/3	3/7	4 days	Complete
Calculate Condo Fees	Eden	3/7	3/10	3 days	Complete

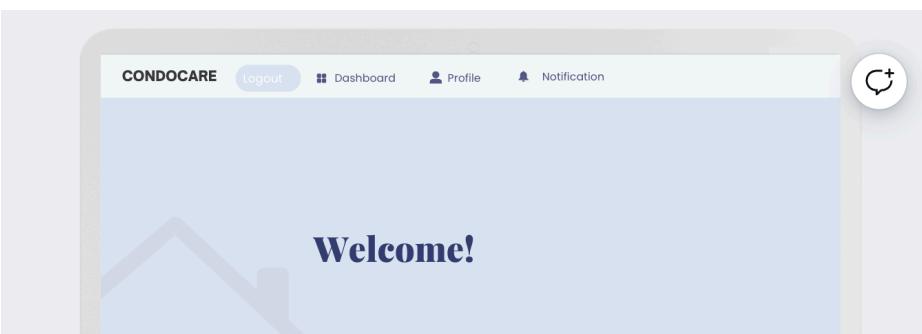
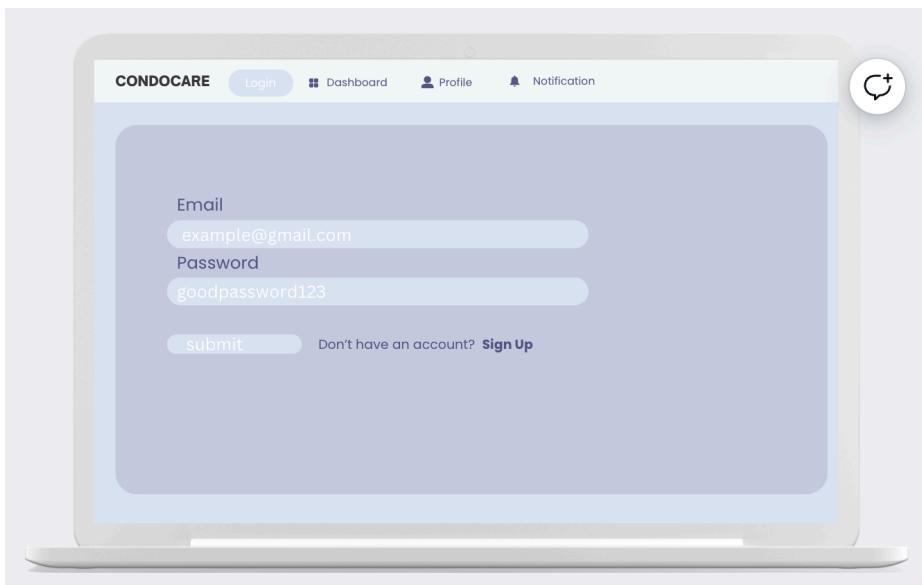
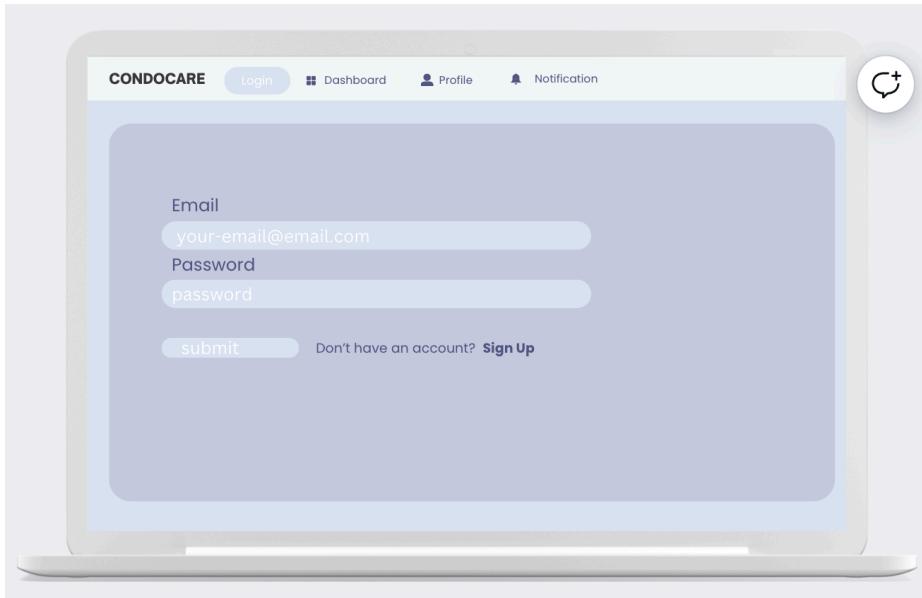
Record Operational Budgets	Minh	3/8	3/16	8 days	Complete
Financial Reports	Joud	3/16	3/21	5 days	Complete
Sprint 4	Nicholas	3/22	4/11	21 days	Complete
Common Facilities Management	Yashi	3/22	3/26	4 days	Complete
Calender-Like Interface	Patrick	3/25	3/30	5 days	Complete
Real-Time Availability	Vithu	3/30	4/2	3 days	Complete
Reservations First-Come-First-Serve Basis	Alexander	3/30	4/7	9 days	Complete
Role-Based Access Control	Joud	4/1	4/5	4 days	Complete
Predefined Roles And Permissions	Ishai	4/5	4/11	6 days	Complete
Sprint 5	Patrick	4/12	5/1	19 days	Not Started
Employee Role Interface	Alexander	4/12	4/16	4 days	Not Started
Secure and Scalable System	Minh	4/16	4/22	8 days	Not Started
Requests For Services	Yashi	4/18	4/22	4 days	Not Started
Accessing Requests	Jeremy	4/23	4/27	4 days	Not Started
User Notification Page	Eden	4/27	5/1	4 days	Not Started

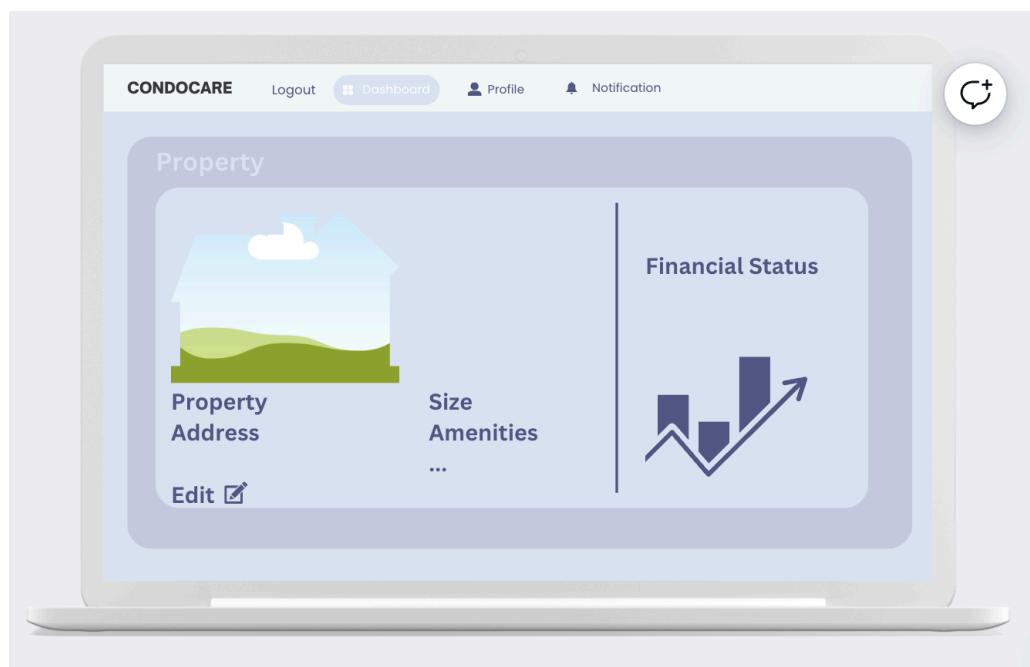
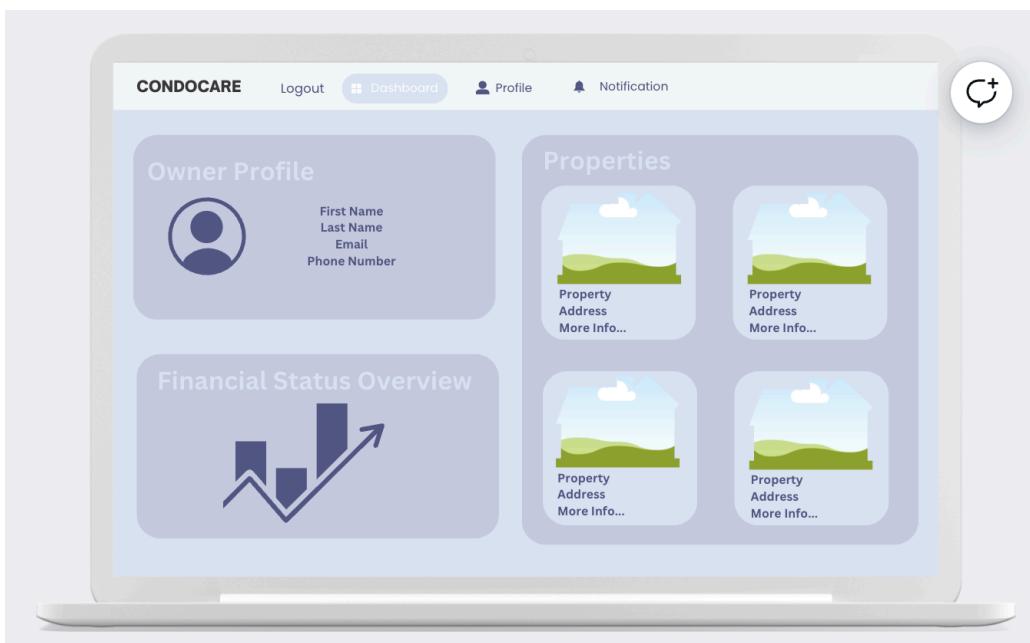


UI Prototype

UI Prototypes for Sprint 2

User Story 5: Create Owner's Property Dashboard





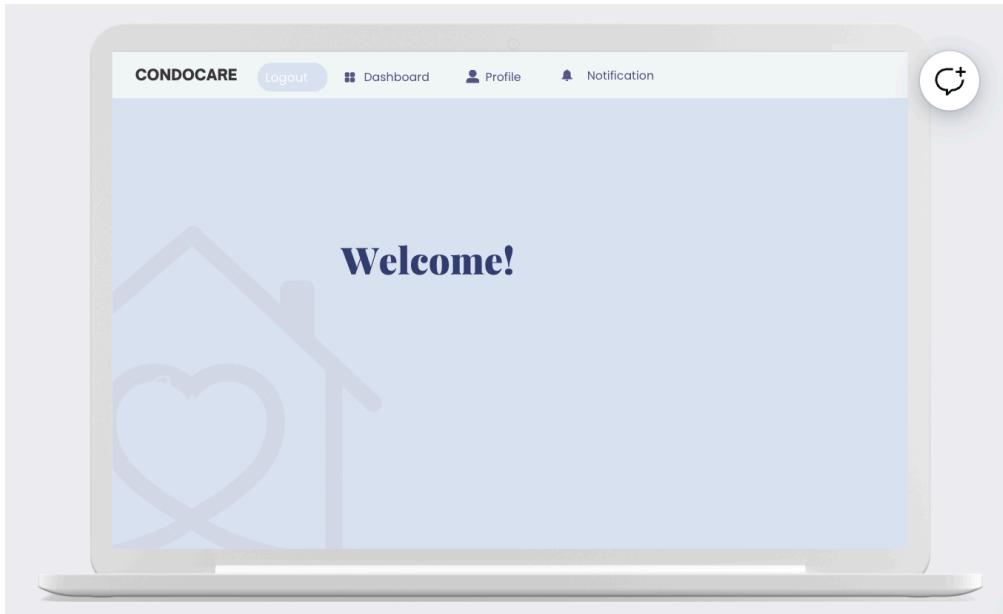


Steps:

1. User Logs In
2. User is redirected to Home Page
3. User selects Dashboard
4. User is redirected to the Dashboard
5. User can view properties, owner profile, and financial status.
6. User can select to view more information on a property by selecting “More Info...”
7. User is redirected to page where they can view full property information

UI Prototypes for Sprint 3

User Story 13: Create Owner Notification Page/Updates



A prototype of a mobile application notifications screen. On the left, there is a list of notifications, each showing a "Request Name" and a status message like "Status : 'Request Assigned,' 'Status Updated,' or 'New Request Submitted.'". Each notification has "Delete" and "Mark as Read" options. On the right, there is a "Filter" section containing two checkboxes: "Status" and "Type". The top of the screen has a navigation bar with the "CONDOCARE" logo, "Logout", "Dashboard", "Profile", and "Notification" buttons. In the top right corner, there is a circular icon containing a "C+" symbol.

Steps:

1. Once logged in, User selects “Notifications”
2. User is redirected to Notifications Page
3. User can view and refresh notifications.
4. User can select a filter.
5. User can delete and mark as read.

User Story 7: Condo Manager can upload files for each property

The screenshot shows the CondoCare software interface. At the top, there's a header with the CondoCare logo and an Admin dropdown menu. Below the header, the property "Estate Alpha" is displayed, located in Greenwich, London. The property features a modern building with blue and orange panels. On the left, there's a section titled "Upload Files" with a placeholder for PDF files (maximum 250 MB). To the right, there are sections for "Units", "Parking Spots", and "Lockers", each listing details for specific units or spots.

The Buckingham Suite	The Buckingham Suite	The Buckingham Suite
Address: 123 Main St.	Address: 123 Main St.	Address: 123 Main St.
Location: Downtown	Location: Downtown	Location: Downtown
Price: \$12000000	Price: \$12000000	Price: \$12000000
Size: 1000 sqft	Size: 1000 sqft	Size: 1000 sqft

Parking Spots			
Level: 2	Slot Number: 12	Size: 200 sqft	Price: \$50000

Lockers		
Location: Basement	Number: 3	Size: 50 sqft

Copyright © CondoCare

Steps:

1. Once logged in with the admin account, the manager can upload files using the designated prompt

User Story 10 : Implement Condo Fee Rate Entry

CondoCare Admin

Estate Alpha



Condo Fee

Price per square foot	1000 \$	EDIT
Price per parking spots	1000 \$	EDIT
Price per locker	1000 \$	EDIT

CONFIRM

Cube67 Condos



Price per square foot	1000 \$	EDIT
Price per parking spots	1000 \$	EDIT
Price per locker	1000 \$	EDIT

CONFIRM

Copyright © CondoCare

Steps:

1. Once logged in with an admin account, the manager can edit the price of each property owned

2. Using the edit button, the manager can set a new price for the rate per square foot, of each parking spots and locker
3. When done editing the price, the confirm button needs to be pressed to save the new prices

User Story 11 : Calculate and Present Condo Fees:

Calculate Condo Fees

Price per square foot

1000 \$

Net Area

sqft

Price per parking spots

1000 \$

Number of parking spots

Price per lockers

1000 \$

Number of lockers

Price of property

\$

Down payment

\$

Interest rate

%

Amortization

years

CALCULATE

Monthly payment

\$

Steps:

1. This feature can be used by anyone - A simple calculator
2. The price per square foot, parking spots and lockers cannot be changed and values come where set by the condo manager in User Story 10
3. Enter size of the unit, number of parking spots and lockers desired.
4. Initial value of property will be shown
5. Enter down payment, interest rate and amortization years
6. Press calculate

7. Monthly payment will be shown

User Story 12 : Record Operational Budget and Costs:

The screenshot shows the CondoCare software interface. At the top, there's a header bar with the CondoCare logo and an Admin dropdown menu. Below the header, on the left, is a sidebar titled "Properties" containing four items: P1, P2, P3, and P4. The main area is titled "Operational Budget and Costs". It features a table with two columns: "Type" (dropdown menu) and "Amount" (text input field with a dollar sign). There are also "EDIT" and "CONFIRM" buttons. Below the table is a "Description" input field. In the top right corner of the main area, there are "+" and "-" buttons. At the bottom of the screen, the text "Copyright © CondoCare" is visible.

Steps:

1. Once logged in with an admin account, the manager can enter operational budgets and costs for each of his company's properties
2. Press + to create a new sheet
3. In the sheet, choose using the Type dropdown menu between budget and cost
4. Enter amount
5. Enter description of the budget or costs (Ex: If it's budget : City grant. If it's costs : Renovate main entrance)
6. Press confirm to save changes
7. If a mistake is made, use Edit button to update values
8. Press - to remove a costs or budget

UI Prototypes for Sprint 4

To run prototype :

<https://www.figma.com/file/cSIJqJVYmX3EbemKhSEWcz/Sprint-3?type=design&node-id=0%3A1&mode=design&t=CKFvzLupI3XxheLx-1>

User Story 14 : Set up common facilities management

The screenshot shows the CondoCare application's "Common Facilities Management" section. On the left, there's a sidebar for "Greenland" with an address: 123 Rue Mackay, Montreal, Quebec H3G 1H8. The main area displays a "Spa" facility. It includes an "Add Facility" button, an "EDIT" button, a "Ø" button, and a "Remove" button. The facility details show "Opening hours: Mon - Fri : 8am - 10pm Sat -Sun : 7am - 5pm". A descriptive text states, "An oasis of calm where people actively do something for the relaxation of body, mind, and spirit." Below this, a "Available:" section lists time slots: 8:00 am, 9:00 am, 10:00 am, and 2:00 pm. At the bottom, there's a form for adding a new facility: "Facility name" (with a "Remove" button), "Add Opening hours:", "Add Description:", and a field for "Enter interval of time : [] hour (Ex: 30min = 0.5 hour)". A "CONFIRM" button is at the bottom right of the form.

Steps :

1. Once logged in with an admin account, the manager can set up the common facilities that can be reserved for a particular condo
2. The manager can add a facility using the “Add Facility” button”
3. Enter facility name, opening hours, description and reservation time slots interval
4. Press “confirm” to save the changes
5. Facility that has been created can be removed using the “Remove” button
6. “Edit” button is used to edit the information for a facility already created
7. “Ø” button is used to make a facility unavailable temporarily without removing it (people cannot reserve the facility at the moment)

User Story 15 & 16 : Calendar-like interface for facility reservation & Display real-time availability in reservation system

The screenshot shows a user interface for booking a facility. At the top, there's a header bar with 'CondoCare' on the left and 'Profile' / 'Dashboard' / 'LOGOUT' on the right. Below the header, the title 'Greenland : Book a facility' is centered. The main area is divided into three sections: 'Choose a facility' (with options for Gym, Spa, Playroom, and Pool, where 'Gym' is highlighted), 'Pick a day' (a calendar for September 2021 showing days from 1 to 30, with '19' highlighted in red), and 'Pick a time' (a grid of time slots: 8:00 am, 9:00 am, 1:00 pm, 2:00 pm, 4:00 pm, and 7:00 pm, with '1:00 pm' highlighted). A large 'SCHEDULE' button is located at the bottom right of the time selection section. At the bottom center of the page, the text 'Copyright © CondoCare' is visible.

Steps :

1. Once logged in with a user account, a unit owner or renter can reserve a facility at their convenient time through this page
2. Choose a facility from the options displayed
3. A calendar and time slots for the current date will appear
4. Choose a day and time
5. Press "Schedule" to confirm choice

User Story 17 : Manage reservations on a first-come-first-serve basis

CondoCare Profile Dashboard LOGOUT

Greenland : Book a facility

Choose a facility :

Gym
Spa
Playroom
Pool

Pick a day

September 2021

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Pick a time

8: 00 am9: 00 am2: 00 pm

7: 00 pm

SCHEDULE

Copyright © CondoCare

ERROR

The chosen time is not available anymore.

Please refresh your page.

Choose a facility :

Gym
Spa
Playroom
Pool

Pick a day**Pick a time**

8: 00 am 9: 00 am 2: 00 pm
7: 00 pm

SCHEDULE

Copyright © CondoCare

Steps :

1. In the picture for US 15 & 16, there are 6 time slots available and in the current picture for there are 4 available
2. When a user tries to reserve a facility, only the available time slots will be presented
3. When a user confirms a time slot, it will be removed from the available options for the next person that tries to book it
4. If two people try to book the same time slot at the same time, the second person that tries to book the time slot will be get an error message asking them to refresh their page to view the new available time slots

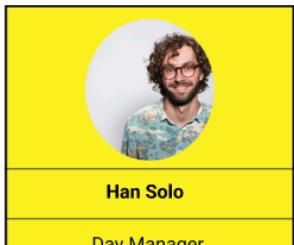
UI Prototypes for Sprint 5

To run the prototype :

<https://www.figma.com/file/cSIJqJVYmX3EbemKhSEWcz/Sprint-3?type=design&node-id=276-768&mode=design&t=LwRj5dXflWNnDqiO-0>

User Story 20 : Manage Employee Roles Interface

Employee Management

[Add employee](#)

Han Solo
Day Manager
Email : han.solo@gmail.com
Phone : (514) 123 - 4567

[EDIT](#) [Remove](#)



John Wick
Night Janitor
Email : ilovepuppies@gmail.com
Phone : (514) 666 - 3333

[EDIT](#) [Remove](#)

Insert Picture

Name :

Role :

Email :

Phone :

Shift ▾

[CONFIRM](#)

Copyright © CondoCare

Steps :

1. The following actions can only be done by a manager, owner or someone with the proper clearing.
2. Press on “Add Employee” to create an employee and assign them a role.
3. Fill up the necessary info and press “Confirm”. When choosing a shift, you will get the chance of day and night and the colour of the card will be set accordingly.
4. After an employee card is created, the role can be modified using the “Edit” button.
5. An employee can also be removed using the “Remove” button.

User Story 22 : Owners can submit requests for services from employees

Employee Task Request

[Add Request](#)[Greenland](#)[Sky](#)[BikiniBottom](#)

Title :

Assign : [Employee](#) ▾

Description :

Task : [Type](#) ▾

Due date : DD MM YYYY

CONFIRM

Send Catering Invoice

[EDIT](#)
[DELETE](#)

Send catering invoice from Grand Opening event on April 1st 2024

Assigned : • Bob
• Sandy

DUE DATE : April 12th 2024

Copyright © CondoCare

Steps :

1. The following actions can only be done by an owner.
2. The owner needs to choose one of his condos first.
3. Use the “Add request” button to create a request form.
4. The employee dropdown button will display the names of the employees working at that Condo.
5. Task dropdown button will display the type of the task, ex : Billing, Cleaning, Misc.

User Story 23: Employee accessing assigned requests from condo owners

Open tasks :

The screenshot shows the 'My Task Center' page of the CondoCare application. At the top, there is a navigation bar with 'CondoCare' on the left and 'Profile' (with a dropdown arrow), 'Dashboard', and 'LOGOUT' on the right. Below the navigation bar, the title 'My Task Center' is centered. To the right of the title are two buttons: one for sorting by 'Date' (with an upward arrow icon) and another for filtering by 'Type' (with a dropdown arrow icon). On the left side of the main area, there are two large buttons: 'OPEN' (light blue background) and 'CLOSED' (white background). The 'OPEN' button is currently selected. In the center, there are two task cards. The first task card is titled 'Send Catering Invoice' and contains the following details: 'Send catering invoice from Grand Opening event on April 1st 2024', 'Assigned : • Bob • Sandy', and 'DUE DATE : April 12th 2024'. It has a 'CLOSE' button in the top right corner. The second task card is titled 'Clean Front Desk' and contains the following details: 'Wipe all the counter and screens on the front desk. Use All-purpose cleaner', 'Assigned : • Bob', and 'DUE DATE : April 5th 2024'. It also has a 'CLOSE' button in the top right corner.

Closes tasks :

My Task Center

↑ Date

Type ▾

OPEN

CLOSED

Do inventory count

REOPEN

Count all the boxes in the backroom

Assigned : • Bob
• Patrick

DUE DATE : April 12th 2024

Welcome new tenants

REOPEN

Help the tenants and give them a quick tour.

Assigned : • Bob

DUE DATE : April 5th 2024

Copyright © CondoCare

Steps :

1. An employee has access to a dashboard displaying the tasks that have been assigned to them.
2. The “Open” and “Closed” buttons will show their open and closed tasks.
3. In the Open tasks tab, they can close a task if it’s completed.
4. The tasks that were closed will move to the Closed tasks tab.
5. In the Closed tasks tab, they can reopen a task that was marked completed.
6. There is a date filter button. When the arrow is up, the tasks are in the order they came in and when the button is pressed, the arrow will turn upside down to display the tasks from the most recent to the oldest.
7. Type is a filter button to display by the type of task. Ex: Misc, Billing, Cleaning.

User Story 24: Create a user notification page or update

The screenshot shows a user interface for a 'Task Notification Center'. At the top, there is a header bar with the 'CondoCare' logo on the left and 'Profile', 'Dashboard', and 'LOGOUT' links on the right. Below the header, the main title 'Task Notification Center' is centered. Underneath the title are two filter buttons: '↑ Date' and 'Type ▾'. The main content area displays three notifications, each in its own box:

- New Task Assigned : Do inventory count** (April 9th 2024)
Assigned : • Bob
• Patrick
- Task Closed : Welcome New Tenants** (April 9th 2024)
Assigned : • Bob
- Status Updated : Send flyers (Reopened)** (April 7th 2024)
Assigned : • Patrick
• Sandy

Copyright © CondoCare

Steps :

1. The owner has a notification page displaying all the new tasks that were created, closed and reopened.
2. Each notification will display the title, the assignees and the date the notification came through.

Sprint 5 Testing Plan

SOEN390 - Testing Plan for sprint #5

Testing Tools:

- **Cypress:** Cypress is an end-to-end testing framework for frontend, offering a clear syntax and real-time test execution. It simulates user interactions in browsers, supports browser APIs, and facilitates efficient test creation with built-in debugging tools. Next sprint we will further increase this test by adding more.
- **Jest:** Jest is a JavaScript testing framework focused on simplicity and speed, ideal for unit testing in a React project. It features built-in runners, assertion libraries, and mocking capabilities, with support for parallel execution and snapshot testing. Next sprint we will further increase this test by adding more.
- **Unittest & Django base tests:** We used Unittest to verify the functionality of backend functions, APIs, and processing logic. Test cases cover input validation, data processing, and response generation. Pytest runs alongside the development environment, providing immediate feedback on code changes.
- **Integration Testing with MSW:** MSW (Mock Service Worker) intercepts HTTP requests made by the frontend to simulate server responses. Configure MSW to return predefined fake data for different endpoints. Integration tests ensure that frontend components interact correctly with backend APIs. MSW testing will be added further in Sprint #5
- **Backend integration testing with Django Client:** Django Client simulates HTTP requests to backend endpoints without actually sending them over the network. Use test cases to verify the integration of different backend components, such as views, serializers, and models. Integration tests cover scenarios such as data retrieval, manipulation, and validation. Next sprint we will further increase this test by adding more.

Testing Approach and coverage:

Unit Testing:

- Objective: Ensure that individual components and modules function correctly in isolation, with a focus on the adding any missing testing during Sprint #5
- Coverage Goals: Maintain or increase code coverage to above 80% for new and modified code, ensuring that recently added functionalities are thoroughly tested.

Integration Testing:

- Objective: Test the interaction between new and existing components, especially focusing on any APIs, data flow, and user interactions that have not been tested in

previous sprints..

- Coverage Goals: Cover all new integration points and critical paths affected by recent changes, with special attention to web deployment.

System Testing:

- Objective: Validate the application's overall functionality, user experience, and performance, ensuring that Sprint #5 web development integrates seamlessly with the current app..
- Coverage Goals: Execute comprehensive tests covering all user stories that have been implemented so far, including web deployment of the app.

Acceptance Testing:

- Objective: Confirm that the application meets all updated business requirements and is deployed without any bugs, with a specific focus on user satisfaction and compliance with the updated product vision.
- Approach: Convert acceptance criteria defined by Business Analysts and Product Owners into automated tests, prioritizing critical user journeys and new requirements introduced in Sprint #5.

Metrics and Reporting:

- Code Coverage: Aim for at least 80% coverage across unit and integration tests, with detailed reports on areas lacking coverage for targeted improvements.
- Bug Density and Fix Rate: Monitor and report on the number of defects per thousand lines of code and the rate at which they are resolved, aiming for continuous quality improvement.
- User Story Completion: Track the completion and acceptance testing success rate of user stories associated with Sprint #5, ensuring that all features deliver the expected value.

Code Management

Code Management Report

14-04-24

TABLE OF CONTENTS

<u>1 . Bug reports/fixing.....</u>	<u>3</u>
<u>2. Correct use of design patterns.....</u>	<u>15</u>
<u>3. Code Coverage.....</u>	<u>18</u>
<u>4. Feature Branching.....</u>	<u>20</u>
<u>5. Defect/Bug Tracking Analysis.....</u>	<u>22</u>
<u>6. Atomic commits.....</u>	<u>24</u>
<u>7. Linking of commits to bug reports/features.....</u>	<u>27</u>
<u>8. Coding Style.....</u>	<u>28</u>
<u>9. Git History Commit Refactoring.....</u>	<u>28</u>

1 . Bug reports/fixing

Bug ID	114		
Originator	Joud Babik	Email: j_babik@live.concordia	Signature: github
Submit Date	March 4, 2024		
Summary	Creation of units is not working. There is an issue with the api calls and endpoints need to be tested.		
Severity	major		

Product	website
Component	View component of the properties model
Version	
Platform	PC
OS	Windows, Mac and Linux
Browser	Google Chrome, Firefox
URL	https://github.com/THE-390-Team/THE-390/issues/114

Table [1] : Bug report and fixing breakdown - Sprint 2

Bug ID	117		
Originator	Patrick MacEachen	Email: patrickmaceachen9@gmail.com	Signature :github
Submit Date	March 5, 2024		

Summary	When the page is refreshed the state of the authentication status is reset. The headers on the page reverts back to what they would be if the user was not logged in.
Severity	minor
Product	website
Component	Issue in the authorization context component in the frontend
Version	
Platform	PC
OS	Windows, Mac and Linux
Browser	Google Chrome, Firefox
URL	https://github.com/THE-390-Team/THE-390/issues/117

Table [2] : Bug report and fixing breakdown - Sprint 3

Bug ID	158		
Originator	Minh Duc Vu	Email: minhductdn4@gmail.com	Signature :github

Submit Date	March 19, 2024
Summary	The forms for creating units in properties are not validating numerical values correctly. It checks the length of the number and not the value.
Severity	minor
Product	website
Component	Issue in the form validation component
Version	
Platform	PC
OS	Windows, Mac and Linux
Browser	Google Chrome, Firefox
URL	https://github.com/THE-390-Team/THE-390/issues/158

Table [3] : Bug report and fixing breakdown - Sprint 3

Bug ID	161
---------------	-----

Originator	Yash Patel	Email: yashpatel1815@hotmail.com	Signature :github
Submit Date	March 19, 2024		
Summary	Submitting the form to create a property is rejected and the server will respond with error 400 "bad request"		
Severity	normal		
Product	website		
Component	This is a frontend and backend issue. At the frontend, the post request is not being sent with the correct method. The property view component does not parse forms with images properly.		
Version			
Platform	PC		
OS	Windows, Mac and Linux		
Browser	Google Chrome, Firefox		
URL	https://github.com/THE-390-Team/THE-390/issues/161		

Table [4] : Bug report and fixing breakdown - Sprint 3

Bug ID	74		
Originator	Joud Babik	Email: j_babik@live.concordia	Signature :github
Submit Date	February 19, 2024		
Summary	Form is not being validated on frontend and backend for data integrity.		
Severity	minor		
Product	website		
Component	Issue in the frontend form component and backend view component.		
Version			
Platform	PC		
OS	Windows, Mac and Linux		
Browser	Google Chrome, Firefox		
URL	https://github.com/THE-390-Team/THE-390/issues/74		

Table [5] : Bug report and fixing breakdown - Sprint 3

Bug ID	115		
Originator	Joud Babik	Email: j_babik@live.concordia	Signature :github
Submit Date	March 4, 2024		
Summary	All forms need to have their input types matched to the backend type to avoid errors.		
Severity	minor		
Product	website		
Component	Issue in backend view component		
Version			
Platform	PC		
OS	Windows, Mac and Linux		
Browser	Google Chrome, Firefox		

URL	https://github.com/THE-390-Team/THE-390/issues/115
------------	---

Table [6] : Bug report and fixing breakdown - Sprint 3

Bug ID	120		
Originator	Yash Patel	Email: yashpatel1815@hotmail.co m	Signature :github
Submit Date	March 16, 2024		
Summary	Only public users are able to change their profile pictures. Other user types should be able to do the same.		
Severity	normal		
Product	website		
Component	Issue in backend model and view for the user profiles.		
Version			
Platform	PC		

OS	Windows, Mac and Linux
Browser	Google Chrome, Firefox
URL	https://github.com/THE-390-Team/THE-390/issues/120

Table [7] : Bug report and fixing breakdown - Sprint 3

Bug ID	165		
Originator	Joud Babik	Email: j_babik@live.concordia	Signature :github
Submit Date	March 20, 2024		
Summary	After logging in, properties are not loaded on the dashboard on the first visit.		
Severity	trivial		
Product	website		
Component	Issue with the components fetching data and rendering UI elements.		
Version			

Platform	PC
OS	Windows, Mac and Linux
Browser	Google Chrome, Firefox
URL	https://github.com/THE-390-Team/THE-390/issues/165

Table [8] : Bug report and fixing breakdown - Sprint 3

Bug ID	169			
Originator	Joud Babik	Email: j_babik@live.concordia	Signature :github	
Submit Date	March 21, 2024			
Summary	Sending the registration key does not work on all devices. It only works on the pc of the person who wrote the backend code.			
Severity	major			
Product	website			

Component	Issue with some backend component	
Version		
Platform	PC	
OS	Windows, Mac and Linux	
Browser	Google Chrome, Firefox	
URL	https://github.com/THE-390-Team/THE-390/issues/169	

Table [9] : Bug report and fixing breakdown - Sprint 3

Bug ID	176			
Originator	Patrick MacEachen	Email: patrickmaceachen9@gmail.com	Signature :github	
Submit Date	March 21, 2024			
Summary	After a brief period of being on the website, a 404 error appears due to the refresh token no longer working. Users are required to logout and sign back in for them to continue working.			

Severity	trivial	
Product	website	
Component	Issue with some backend component	
Version		
Platform	PC	
OS	Windows, Mac and Linux	
Browser	Google Chrome, Firefox	
URL	https://github.com/THE-390-Team/THE-390/issues/176	

Table [10] : Bug report and fixing breakdown - Sprint 3

Bug ID	175			
Originator	Patrick MacEachen	Email: patrickmaceachen9@gmail.com	Signature :github	

Submit Date	March 21, 2024	
Summary	After receiving the registration key for a condo unit, the user tries to register the condo by entering the registration key. This is where an error happens. The system cannot locate the key within the database.	
Severity	trivial	
Product	website	
Component	Issue with some backend component	
Version		
Platform	PC	
OS	Windows, Mac and Linux	
Browser	Google Chrome, Firefox	
URL	https://github.com/THE-390-Team/THE-390/issues/176	

Table [10] : Bug report and fixing breakdown - Sprint 3

2. Correct use of design patterns

The MVC pattern is used in the backend to simplify the management of the code. The model component is represented by django models. In our case, those are the classes that are in the models.py files. The view component is represented by django views, which are in the views.py files. This controls what the user sees and sends the user input to the controller component. The controller component is represented by the django views as well. It handles the input sent by the view, updates the model and can do other tasks such as input validation.

```
class User(AbstractBaseUser, PermissionsMixin):
    """
    Custom user model for authentication.
    """

patrickmac3, 3 weeks ago | 1 author (patrickmac3)
class Role(models.TextChoices):
    PUBLIC = 'PUBLIC', 'Public'
    EMPLOYEE = 'EMPLOYEE', 'Employee'
    COMPANY = 'COMPANY', 'Company'

email = models.EmailField(_('email address'), unique=True)
role = models.CharField(max_length=50, choices=Role.choices, default=Role.PUBLIC)
first_name = models.CharField(max_length=150, blank=True)
last_name = models.CharField(max_length=150, blank=True)
created_at = models.DateTimeField(default=timezone.now)
is_staff = models.BooleanField(default=False)
is_active = models.BooleanField(default=True)
objects = CustomUserManager()

USERNAME_FIELD = 'email'
REQUIRED_FIELDS = ['role', 'first_name', 'last_name']
```

patrickmac3, 3 weeks ago

Figure 1: MVC example: User model

```

class PublicProfileViewSet(ModelViewSet):
    """
    A ViewSet for managing public profiles.
    """

    queryset = PublicProfile.objects.all()
    serializer_class = PublicProfileSerializer
    lookup_field = 'user'
    parser_classes = [MultiPartParser, FormParser, JSONParser]

    """
    List the various units
    """
    def get_condo_units(self, request, **kwargs):
        """
        Retrieve the condo units associated with a user.

        Args:
            request (HttpRequest): The HTTP request object.
            **kwargs: Additional keyword arguments.

        Returns:
            Response: The HTTP response object containing the serialized condo units.

        Raises:
            Http404: If the user or condo units are not found.
        """
        user_id = self.kwargs.get('user_id', None)
        if not user_id:
            return Response({'error': "Missing user_id"}, status=status.HTTP_400_BAD_REQUEST)
        try:
            profile = PublicProfile.objects.get(user_id=user_id)
        except PublicProfile.DoesNotExist:
            return Response({'error': "User not found"}, status=status.HTTP_404_NOT_FOUND)
        if not profile.condo_units:
            return Response({'error': "No condo units found for this user"}, status=status.HTTP_404_NOT_FOUND)
        serializer = CondoUnitSerializer(profile.condo_units, many=True)
        return Response(serializer.data, status=status.HTTP_200_OK)

```

Figure 2: MVC example: user view and part of controller

The singleton pattern is utilised in the frontend to create a singular instance of an axios object called "axiosInstance". This object is created with a predetermined base url, timeout and headers. By using the singleton pattern we ensure that this specific configuration is always used.

```
import axios from 'axios';

const baseURL = 'http://localhost:8000/';

const axiosInstance = axios.create({
  baseURL: baseURL,
  timeout: 5000,
  headers: {
    Authorization: localStorage.getItem('access_token')
      ? 'JWT ' + localStorage.getItem('access_token')
      : null,
    'Content-Type': 'application/json',
    accept: 'application/json',
  },
})
```

Figure 3: Example of singleton pattern of axiosInstance with configuration

Another pattern used in the frontend is the context design pattern. This pattern is used to maintain states and to share information with the entire system. It is utilised in AuthContext.js, ProfileContext.js, PropertyContext.js. These contexts share information about the authorization state of the user, the user information and any property related to the user.

```

const AuthContext = createContext();

export function useAuth() {
    return useContext(AuthContext);
}

export function AuthProvider(props) {

    const [authUser, setAuthUser] = useState({
        first_name: '',
        last_name: '',
    })
    const [isLoggedIn, setIsLoggedIn] = useState(false)

    // check local storage for token, and update the login state
    // accordingly to avoid loss of login state on refresh
    const checkAuthState = () => {
        const access_token = localStorage.getItem("access_token");
        if (access_token) {
            axiosInstance.defaults.headers["Authorization"] = "JWT " + access_token;
            setIsLoggedIn(true);
            setAuthUser(localStorage.getItem("ID"));
        } else {
            setIsLoggedIn(false);
            setAuthUser(null);
        }
    };

    const value = {
        authUser,
        setAuthUser,
        isLoggedIn,
        setIsLoggedIn,
        checkAuthState
    }

    return (
        <AuthContext.Provider value={value} > {props.children} </AuthContext.Provider>
    )
}

```

Figure 4: Authorization context as example of context pattern

3. Code Coverage

Tool: codecov.io

link : <https://app.codecov.io/gh/THE-390-Team/THE-390>

Code coverage gives us a coverage tree indicator in percentage to display even better our code coverage:

Code Coverage Tree



1%

There also exists a section in the main README.md that upon each push into the repo will update to show the recent reports from testing.

Testing Reports from Workflow

```
| frontend@0.1.0 coverage react-scripts test --watchAll=false --coverage
```

File	% Stmt	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	1.11	0	3.29	1.13	
src	0	100	0	0	
App.js	0	100	0	0	25-30
index.js	0	100	100	0	10-12
src/api	0	0	0	0	
axios.js	0	0	0	0	3-83
src/components	44.44	100	50	44.44	
Content.js	100	100	100	100	
Footer.js	100	100	100	100	
Header.js	0	100	0	0	12-14
LargeTitle.js	0	100	0	0	4-5
src/components/createProperty	0	0	0	0	
CreateLocker.js	0	0	0	0	13-185
CreateParking.js	0	0	0	0	13-188
CreateProperty.js	0	0	0	0	13-156
CreateUnit.js	0	0	0	0	13-189

4. Feature Branching

Within our repository we follow a feature branching conventions

Branching for tasks:

(*with user story*)

main

→ development

→ BRANCH-US(*user_story_number*)-(*user_story_description*)

→ TASK-US(*user_story_number*)-(*task_description*)-(*task_issue_number*)

(*without user story*)

main

→ development

→ TASK-NO-US-(*task_description*)

Branching for bugs:

(*with user story*)

main

→ development

→ BRANCH-US(*user_story_number*)-(*user_story_description*)

→ BUG-US(*user_story_number*)-(*bug_description*)-(*bug_issue_number*)

(*without user story*)

main

→ development

→ BUG-NO-US-(*bug_description*)

Issue for tasks:

TASK-US(number) or TASK-NO-US.

BUG-US(number) or BUG-NO-US.

Following the above structure we are able to create feature branching ensuring a branch is utilized to perform a specific feature. Each user story gets its own branch and then we branch from this user story into this branch to create a task. There are also NO-US bugs and issues and these can be used for non user story specific tasks (i.e.: updating the CI/CD pipeline).

5. Bug reports fixing techniques

For Bug Fixes:

Bug Issue Creation:

- Ensure that the bug convention is followed for issue creation (BUG-US(number) or BUG-NOUS).
- Check if the bug is clearly described, including details such as when it was noticed, steps to reproduce, and any relevant context or observations.

Applying Labels:

- Confirm that appropriate labels are applied to the bug issue, including priority, bug, and severity (if applicable).

Branch Naming:

- Ensure the branch name follows the convention BUG-US(number)-(description of user story)-(issue bug number) or BUG-NO-US-(short title)-(issue bug number) and matches the corresponding bug issue number.

Committing Changes:

- Check if commit messages are descriptive and include the bug issue number.

Pull Request for Bug Fix:

- Confirm that the pull request "relates to" the corresponding bug issue.
- Review the pull request description for detailed information on the bug, steps to reproduce, and the fix implemented.
- Ensure that the severity of the bug is mentioned in the pull request description.
- Check for the inclusion of screenshots illustrating the bug before and after the fix (if applicable).
- Include "Fixed" in the pull request title.
- Verify that the user story number is present in the pull request title and matches the branch name.

5. Defect/Bug Tracking Analysis

When a bug is found, the developer creates an issue and adds a bug label to it for quick and efficient tracking as can be seen in the image below.

The screenshot shows a list of open bugs in a software repository. The search bar at the top contains the query "is:open is:issue label:bug". There are 17 labels and 3 milestones listed. A green button for "New issue" is visible. The list includes the following items:

- BUG-US9-sending registration keys doesn't work on all devices (backend, bug, P1)
- BUG-NO-US-properties-do-not-load-on-the-first-visit-to-dashboard (bug, frontend, P2)
- BUG-NOUS Unable to create new property (bug, P0)
- BUG-NOUS Fix users not being able to change pfp (backend, bug)
- BUG-NOUS Match input types to backend for all input react formsto avoid server errors (bug, enhancement, frontend, help wanted, P0, task)
- BUG-US8 Implement validation on both the frontend and backend to ensure data integrity. (backend, bug, frontend, P0)

Each item shows the issue number, creation date, assignee, and a small icon representing the labels.

Design Quality

Date: Thu Mar 21 03:13:10 UTC 2024

Code Quality Indicators

quality gate	passed	bugs	1	code smells	98	duplicated lines	0.4%	lines of code	2.7k	reliability	C
security	E	technical debt	4h	maintainability	A	vulnerabilities	1				

Figure [1] : Code Quality Indicators

These metrics can be viewed on the repository upon every time a branch is merged into another. This will be updated.

The tool used is: **SonarCloud** is a cloud-based code analysis service designed to detect coding issues in 30+ languages, frameworks and IaC platforms.

(<https://docs.sonarsource.com/sonarcloud/#:~:text=SonarCloud%20Documentation-,What%20is>

[SonarCloud languages frameworks and platforms.\)](#)

There is also a dashboard that goes more in depth and can show exactly where these bugs metrics occur:
<https://sonarcloud.io/organizations/the-390-team/projects>

Here is a quick snapshot of it

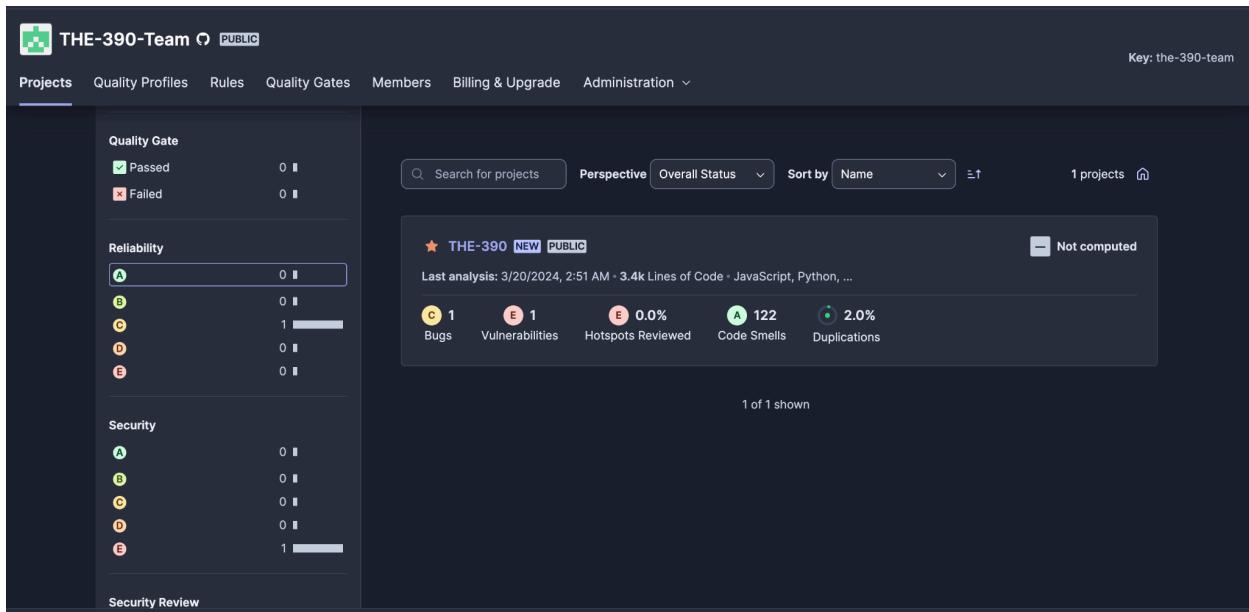


Figure [1] : Code Quality Indicators

6. Atomic commits

Each commit is atomic in nature implementing at times a single feature over multiple files to ensure consistency throughout

Task US6 change dashboard based on role #11 #129

Merged NotTh3ta merged 11 commits into [development](#) from [TASK-US4-CHANGE-DASHBOARD-BASED-ON-ROLE-#4](#) 2 days ago

Conversation 4 Commits 11 Checks 0 Files changed 7

JRB958 commented 3 days ago · edited

in this PR the dashboard will only shows what the role of the user allows

before:

- any user role (PUBLIC or COMPANY) will show the same properties (all properties in the db)

after:

- COMPANY profiles will only see the PROPERTIES related to their company id
- PUBLIC profiles will only see the UNITS related to their id

Notes:

- to test company please use : Joud.babik@gmail.com 123qweasd to login
- to test public please user: test@example.com password123 to login
- the parking and storage will be outputted once the db model can support them

related to:
[#11](#) [#7](#)

JRB958 added 11 commits 3 days ago

- remove the id from local storage when logout
- add three fetch methods and update an existing one: ...
- move the user role and the method to fetch it to ProfileContext
- add a fetch role to get the user role right after login
- fix accidental deletion made on the last commit on this file
- update the property container to support propertycard and unitcard
- update the unit card to support both property output and unit output ...
- update the dashboard to use the profile context for user role fetching
- fix issue of having the previous user data after logout
- Merge branch 'development' into TASK-US4-CHANGE-DASHBOARD-BASED-ON-ROLE-...
- merge development into the branch before the PR

Figure[3]

In the above image, we can see the same task as the commit in figure [X], each line identified

with a serial number such as 'ecb43af' represents each atomic commit made within the pull request for the specific task.

Pull Request Reviewing:

Upon every pull request we ensure that 2 people are reviewing another person's work. This review has one person working on a similar task and another person ensuring code quality and documentation.

The code quality portion of the review follows a strict review process which is outlined as follows:

For Tasks:

Task Issue Creation:

- Ensure that a task issue exists for the completion with the convention TASK-US(number) or TASK-NOUS.
- Check if the task objectives and outcomes are clearly outlined within the issue.

Branch Naming:

- Verify that the branch name follows the convention US(number)-(description of user story) or NOUS-(short title) and matches the corresponding task issue number.

Committing Changes:

- Check if commit messages are descriptive and include the task issue number.

Pull Request for Task Completion:

- Ensure that the pull request "relates to" the corresponding task issue.
- Review the pull request description for a detailed overview of the task objectives and outcomes.
- Confirm that the severity label is applied (if applicable).
- Ensure the inclusion of any relevant details, such as potential impacts or considerations.
- Verify that the user story number is present in the pull request title and matches the branch name

For Tasks:

- Issue: Ensure the pull request relates to the corresponding task issue.
- Branch Name: Verify that the branch name follows the convention (TASK-US(number)-(short title) or NOUS-(short title)).
- Commits: Check if commit messages are descriptive and include the task issue number.
- Description: Review the pull request description for a detailed overview of the task objectives and outcomes.
- Upon closing an Issue Task, we reference the Pull Request that was responsible for the implementation of the Task

Here is an image of how reviewing plays out on a real pull request:

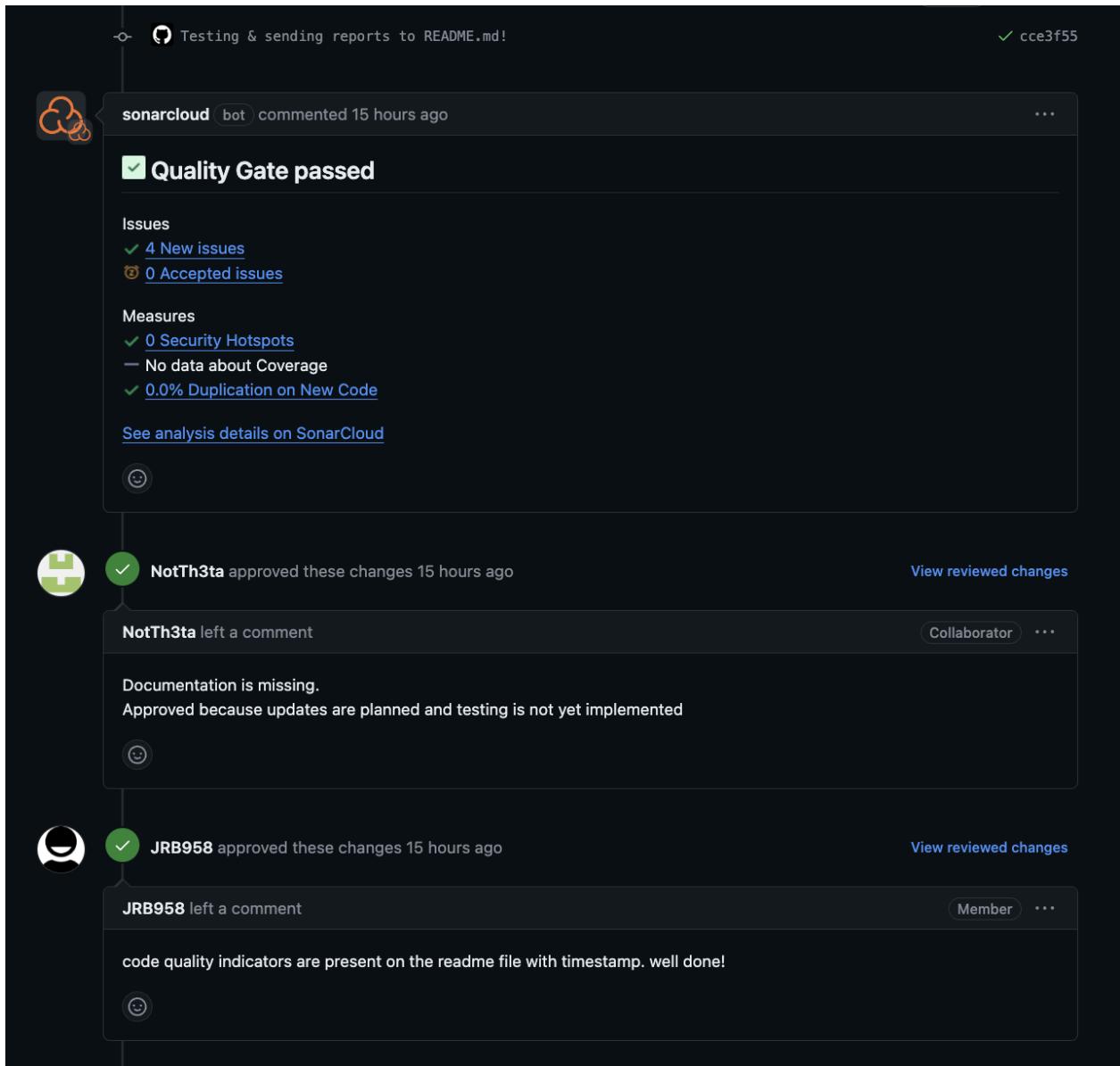


Figure [4]

As can be seen in the above, 2 code reviews are required by our team to push to the repo and we also have configured sonarcloud bot extension which gives us a quick rundown on what to fix on our repo and if something needs to be fixed.

7. Linking of commits to bug reports/features

The screenshot shows a GitHub issue page for a bug titled "BUG-US9-sending registration keys doesn't work on all devices #169". The issue was opened by JRB958 13 hours ago. The description includes a note that "only backend device can run the feature well (create a registration key and submitting it in PR #170)". The issue has been assigned to patrickmac3 and JRB958, and labeled with "backend", "bug", and "P1". It has been mentioned by JRB958 and referenced in a commit by NicholasWahome. A commit from TASK-US9 input field to send registration key 78 (#170) has been merged. The issue is currently "Merged". Notifications are turned off, and there are 2 participants. The footer includes links to GitHub Community Guidelines and privacy settings.

Figure [2] : Bug Linking

Each bug follows a naming convention that starts with the word BUG, followed by the user story number and a brief description of the bug.
We then link to the user story or specific task the bug in the description.
Priority labes are then added to the bug to give more feedback on it.
This creates commit linking between the bug and the different commits made to it.

8. Coding Style

The coding style of the team is to work on a user story that is then going to have its dedicated branch. From that branch creation, each member works on their specific task and has to ensure that they review their code and add commits iteratively. Upon their task being completed, they create a pull request to merge into the user story branch and once the user story tasks are all completed. It gets pushed into the development branch.

9. Git History Commit Refactoring

dde1c83 - github-actions[bot], 10 hours ago : Testing & sending reports to README.md!

1 4 README.md

03e3e31 - Nicholas W, 10 hours ago : Remove document

Removing the google link to the repo as it is no longer necessary. This is a VERY SMALL change so no 2 review needed

0 3 .github/workflows/Testing.yml

eedb962 - github-actions[bot], 2 days ago : Testing & sending reports to README.md!

1 1 README.md

bdfca6f - github-actions[bot], 3 days ago : Testing & sending reports to README.md!

1 1 README.md

c743859 - JRB958, 3 days ago : Task us14 common facilities 186 (#194)

Added page for common facilities, access through property page

2be1a3f - github-actions[bot], 3 days ago : Testing & sending reports to README.md!

1 1 README.md

e935a80 - MinhDuc1711, 3 days ago : Merge branch 'development' into
TASK-US14-COMMON-FACILITIES-186

ece6abf - github-actions[bot], 3 days ago : Testing & sending reports to README.md!

14 12 README.md

a962528 - MinhDuc1711, 3 days ago : Removed TestCreate.test

0 66 frontend/src/components/createProperty/TestCreate.test.js
delete mode 100644 frontend/src/components/createProperty/TestCreate.test.js

6f0ff1c - MinhDuc1711, 3 days ago : Revert readme changes

1 1 README.md

7a4479c - MinhDuc1711, 3 days ago : Temporarily removed testCreate.test

59 59 frontend/src/components/createProperty/TestCreate.test.js

8d11808 - MinhDuc1711, 3 days ago : Merge branch 'TASK-US14-COMMON-FACILITIES-186'
of <https://github.com/JRB958/THE-390> into TASK-US14-COMMON-FACILITIES-186

6e4c424 - MinhDuc1711, 3 days ago : Added common facilities page

2 0 frontend/src/App.js
94 0 frontend/src/components/commonFacilities/CommonFacilities.js
9 0 frontend/src/components/property/PropertyPage.js
create mode 100644 frontend/src/components/commonFacilities/CommonFacilities.js

8ce443f - github-actions[bot], 3 days ago : Testing & sending reports to README.md!

1 1 README.md

2fad9ee - MinhDuc1711, 3 days ago : added simple jest tests

```
66    0    frontend/src/components/createProperty/TestCreate.test.js
179    0    frontend/src/components/dashboard/DashBoard.test.js
create mode 100644 frontend/src/components/createProperty/TestCreate.test.js
create mode 100644 frontend/src/components/dashboard/DashBoard.test.js
```

30d4358 - github-actions[bot], 5 days ago : Testing & sending reports to README.md!

```
1    1    README.md
```

e0d6b74 - patrickmac3, 5 days ago : Update registration endpoints to use PATCH method (#185)

Fixed issue with registering a condo unit to public profile by updating the request method used.

previously, this was done using a GET request, and i updated it to a PATCH request, since using this endpoint will update information concerning the public profile and the condo unit involved.

even though it they are not used yet, i made the same change for registering a parking and storage unit to a user with the register key.

Specific changes:

- update method to PATCH in public profile urls, for endpoint using the concerned view methods
- updated React.js api to use patch rather than get

Related Issue: #175

eb54b97 - github-actions[bot], 5 days ago : Testing & sending reports to README.md!

```
1    3    README.md
```

2390ecf - patrickmac3, 5 days ago : Merge branch 'development' into BUG-US9-problem-using-registration-keys-175

745357e - github-actions[bot], 10 days ago : Testing & sending reports to README.md!

```
1    1    README.md
```

ae3f62c - Nicholas W, 10 days ago : Delete myfile.txt

Deletion of test file for CI/CD pipeline

#184

0 1 myfile.txt
delete mode 100644 myfile.txt

59efbd9 - github-actions[bot], 13 days ago : Testing & sending reports to README.md!

1 1 README.md

6f9552b - patrickmac3, 13 days ago : Update registration endpoints to use PATCH method

3 3 backend/user_profile/urls.py
1 1 frontend/src/components/registrationKey/SubmitRegistrationButton.js

8ec3f11 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

1 1 README.md

56a34bd - Nicholas W, 3 weeks ago : Admin Commit

No user story

0 2 .github/workflows/Testing.yml

9435f8c - Nicholas W, 3 weeks ago : Admin Commit

Very Small Change for code coverage no user story nothing is attached to this it is not within the scope for delivery

0 4 .github/workflows/Testing.yml

f261934 - Nicholas W, 3 weeks ago : Admin Commit

This type of commit is small in nature and does not contribute to any user story and is only to test a functionality in the github actions to check code coverage

3 1 .github/workflows/Testing.yml

8a5cb84 - Nicholas W, 3 weeks ago : Update Testing.yml

1 1 .github/workflows/Testing.yml

fe9b203 - Nicholas W, 3 weeks ago : Admin Push

7 3 .github/workflows/Testing.yml

dc7bbf6 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

1 1 README.md

5c7103c - Nicholas W, 3 weeks ago : Test no us integration testing base 180 (#181)

Created unit test for frontend for

- Financial Public
- Total

Relates: #180

cbc2006 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

88 178 README.md

06d9d12 - Nicholas W, 3 weeks ago : Update Testing.yml

3 3 .github/workflows/Testing.yml

9ba4388 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

178 86 README.md

7cdb7fd - Nicholas W, 3 weeks ago : Backend into read me

1 1 .github/workflows/Testing.yml

868b1b2 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

1 1 README.md

43440e5 - Nicholas W, 3 weeks ago : Making Migrations

3 0 .github/workflows/Testing.yml

09c1945 - Nicholas W, 3 weeks ago : Admin Push

2 1 .github/workflows/Testing.yml

b02d53c - Ishaï Cohen, 3 weeks ago : TASK-NO-US-uploading-the-submission-file

Uploaded master doc of all compiled files into PDF.

- - SOEN_390_Master_Doc.pdf
create mode 100644 SOEN_390_Master_Doc.pdf

002cac6 - Nicholas W, 3 weeks ago : Update Testing.yml

1 1 .github/workflows/Testing.yml

9a39037 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

8 8 README.md

e23e24d - Nicholas W, 3 weeks ago : Uncomment Admin

1 1 .github/workflows/Testing.yml

71a11a8 - Nicholas W, 3 weeks ago : Update Testing.yml

2 2 .github/workflows/Testing.yml

73493e1 - Nicholas W, 3 weeks ago : Update Testing.yml

2 2 .github/workflows/Testing.yml

1158ed1 - Nicholas W, 3 weeks ago : Administrative Commit

Administrative Commit

7 2 .github/workflows/Testing.yml

250728d - NicholasWahome, 3 weeks ago : Quick commit

1 1 frontend/src/components/dashboard/FinancialPublic.test.js

b0f1505 - NicholasWahome, 3 weeks ago : Bringing back the yml file

- The yml was deleted from the repository
It is being brought back with more documentation

155 0 .github/workflows/Testing.yml
create mode 100644 .github/workflows/Testing.yml

7a7c1f3 - NicholasWahome, 3 weeks ago : Merge branch 'development' into TEST-NO-US-Integration-Testing-Base

aef59fd - NicholasWahome, 3 weeks ago : Adding 2 tests

- One test for Total.js
- One test for FinancialPublic.js

32 0 frontend/src/components/dashboard/FinancialPublic.test.js
23 0 frontend/src/components/dashboard/financial/Total.test.js
create mode 100644 frontend/src/components/dashboard/FinancialPublic.test.js
create mode 100644 frontend/src/components/dashboard/financial/Total.test.js

ce50c98 - Nicholas W, 3 weeks ago :
BRANCH-US12-RECORD-OPERATIONAL-BUDGET-COST-28 (#179)

Very minor PR, noticed I accidentally deleted a the operation component just before the last PR and I just fixed it.

related user story: #28
8c2fc82 - JRB958, 3 weeks ago : fix accidental deletion of the operation component

3 1 frontend/src/App.js

a9efa93 - Nicholas W, 3 weeks ago : Branch us11 implement condo fee rate entry 27 (#178)

PR

create UI for:
fields to enter condo fee rate when the property is created

fields to show condo fees for units in properties and unit cards

related story #27

9f52418 - JRB958, 3 weeks ago : Merge branch 'development' into
BRANCH-US11-IMPLEMENT-CONDO-FEE-RATE-ENTRY-27

6954b3b - JRB958, 3 weeks ago : fix some extra merge conflicts

```
0      195  .github/workflows/Testing.yml
30     30   frontend/src/components/createProperty/CreateProperty.js
delete mode 100644 .github/workflows/Testing.yml
```

f8de95d - JRB958, 3 weeks ago : Merge branch 'development' into
BRANCH-US11-IMPLEMENT-CONDO-FEE-RATE-ENTRY-27

68cdcaf - MinhDuc1711, 3 weeks ago : Branch us12 record operational budget cost 28 (#177)

in this PR:

- company employees can have an operational page that allows them to enter costs and compare them to unit condo fees.
- this PR only has UI template functionality will be added later when time permits.

-small bug: low severity and priority: the "operation" on the navigation bar doesn't show right away, requires navigation on the website to show us (briefly not visible to company and briefly visible to public) which is not the intended design.

related us:

#28

9710d81 - JRB958, 3 weeks ago : Merge branch
'BRANCH-US12-RECORD-OPERATIONAL-BUDGET-COST-28' of
<https://github.com/JRB958/THE-390> into
BRANCH-US12-RECORD-OPERATIONAL-BUDGET-COST-28

7b18806 - JRB958, 3 weeks ago : remove the testing.yml file

```
0      195  .github/workflows/Testing.yml
```

delete mode 100644 .github/workflows/Testing.yml

ae7c9b5 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

7 6 README.md

80e6c3b - JRB958, 3 weeks ago : Merge branch 'TASK-US12-OPERATION-PAGE-134' into BRANCH-US12-RECORD-OPERATIONAL-BUDGET-COST-28

4a9e912 - JRB958, 3 weeks ago : fix some issues from merging

3 2 frontend/src/components/navigationBar/NavigationBar.js
1 1 frontend/src/components/registrationKey/SubmitRegistrationButton.js

03deace - JRB958, 3 weeks ago : merge dev into branch

65f0796 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

1 2 README.md

6dec3e3 - Nicholas W, 3 weeks ago : BRANCH-US13 generate annual financial reports 29 (#174)

in this PR:

- use the UI that was designed in #135 to implement api calls and receive the financial data dynamically from the backend instead of hard coding a sample as before.

the user story #29 is not fully completed, but the available features will be merged for the demo a

related US:

#29

9a2996b - Nicholas W, 3 weeks ago : Merge branch 'development' into BRANCH-US13-GENERATE-ANNUAL-FINANCIAL-REPORTS-29

82dbafa - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

53 50 README.md

630133e - JRB958, 3 weeks ago : remove outdated test file

0 14 frontend/src/components/dashboard/financial/Financial.test.js
delete mode 100644 frontend/src/components/dashboard/financial/Financial.test.js

3566dd0 - JRB958, 3 weeks ago : Merge branch
'BRANCH-US13-GENERATE-ANNUAL-FINANCIAL-REPORTS-29' into
TASK-US13-PUBLIC-CREATE-REACT-COMPONENT-TO-SHOW-FINANCIAL-REPORTS-29

3b1801a - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

1 1 README.md

db0c479 - Yash Patel, 3 weeks ago : Task no us add code cov extension 172 (#173)

Adding Code coverage Report + other other reports

Relates: #172

b883e2f - JRB958, 3 weeks ago : update the endpoint with the new endpoint made for financial reporting from development

6 6 frontend/src/components/dashboard/financial/Financial.js

446a9da - JRB958, 3 weeks ago : Merge branch 'development' into
BRANCH-US13-GENERATE-ANNUAL-FINANCIAL-REPORTS-29

6c07673 - JRB958, 3 weeks ago : create api call to get the financial information from the server

5 4 frontend/src/components/dashboard/DashBoard.js
120 191 frontend/src/components/dashboard/financial/Financial.js

9fb7104 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

7 2 README.md

20beed2 - Nicholas W, 3 weeks ago : Adding link to code coverage

#172
7 1 .github/workflows/Testing.yml

47a46a5 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

1 1 README.md

b6bdd2a - Nicholas W, 3 weeks ago : Switched position of code coverage

Relates: #172
7 7 .github/workflows/Testing.yml

6d348f0 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

1 1 README.md

ca8c3e9 - Nicholas W, 3 weeks ago : Code Coverage Report

- Added lines for code coverage uploading

#172
8 0 .github/workflows/Testing.yml

3795043 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

1 1 README.md

4dc58c8 - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

13 9 README.md

fcc69a4 - Nicholas W, 3 weeks ago : Update Testing.yml

6 41 .github/workflows/Testing.yml

a18e2a0 - Nicholas W, 3 weeks ago : TASK-US9 input field to send registration key 78 (#170)

this PR has:

- Creation of "send key" button visible only for company employees' dashboard to send a registration key to a user for a specific available unit which sends it by email to the user.
- Creation of "submit key" button visible only to public profiles where they add the registration key and they get assigned to a unit as renters or owners.
- Minor bug fix for updating company user profile
- Minor UI updates like adding page titles clearly
- updated the e2e cypress testing to pass the check with property addition (property-units)

Notes: there's a known bug that doesn't allow the registration key to work from my computer, but works from the backend developer's computer. it's being worked on currently #169

related Issues: #78

9e0c27d - github-actions[bot], 3 weeks ago : Testing & sending reports to README.md!

3 3 README.md

20341ac - Nicholas W, 3 weeks ago : Update Testing.yml

6 40 .github/workflows/Testing.yml

7670bac - MinhDuc1711, 3 weeks ago : Task us12 add expenses 142 (#171)

- updated the unit models to add an operational_expense field representing the expenses needed for a single unit
- updated the property profile model to add a name
- updated the Company Finance View to also return operational expenses and produce a full report on the budget of a company

related issue: #142

f56b1c1 - JRB958, 4 weeks ago : Merge branch 'development' into BRANCH-US13-GENERATE-ANNUAL-FINANCIAL-REPORTS-29

2899584 - JRB958, 4 weeks ago : minor styling and refractoring

```
11    9    frontend/src/components/dashboard/DashBoard.js
53    7    frontend/src/components/dashboard/{ => financial}/Financial.js
0     0    frontend/src/components/dashboard/{ => financial}/Financial.test.js
32    0    frontend/src/components/dashboard/financial/Total.js
3     3    frontend/src/components/property/PropertyPage.js
1     1    frontend/src/index.css
rename frontend/src/components/dashboard/{ => financial}/Financial.js (80%)
rename frontend/src/components/dashboard/{ => financial}/Financial.test.js (100%)
create mode 100644 frontend/src/components/dashboard/financial/Total.js
```

d79959f - Nicholas W, 4 weeks ago : Update Testing.yml

```
2     2    .github/workflows/Testing.yml
```

a5594f0 - Nicholas W, 4 weeks ago : ByPass

```
2     2    .github/workflows/Testing.yml
```

79e660b - Nicholas W, 4 weeks ago : Update Testing.yml

```
17    9    .github/workflows/Testing.yml
```

842ed64 - JRB958, 4 weeks ago : configure the workflow with user name and email

```
2     0    .github/workflows/Testing.yml
```

50bc99f - JRB958, 4 weeks ago : further changes to drop it below previous 9%

```
4     33   frontend/cypress/e2e/PropertyTesting.cy.js
45    0    frontend/cypress/support/commands.js
```

f68b69d - patrickmac3, 4 weeks ago : Refactor CompanyFinanceView in finance/views.py

```
14    7    backend/finance/views.py
```

021bc73 - patrickmac3, 4 weeks ago : Refactor finance calculation logic in
CompanyFinanceView

69 30 backend/finance/views.py
1 1 backend/user_profile/urls.py

d52f8fd - patrickmac3, 4 weeks ago : Add 'operational_expense' and 'image' fields to UnitSerializer and update fields in PropertyProfileSerializer

2 2 backend/properties/serializers.py

92433ac - patrickmac3, 4 weeks ago : Update models.py: Set 'name' field to allow null values and add 'operational_expense' field to Unit model

5 3 backend/properties/models.py

d9663ea - JRB958, 4 weeks ago : more fixes to the testing file to have less repition:

8 33 frontend/cypress/e2e/PropertyTesting.cy.js
14 5 frontend/cypress/support/commands.js
1 1 frontend/src/components/dashboard/DashBoard.js

541f653 - JRB958, 4 weeks ago : change the login function in e2e cypress to a custom command to pass the duplicated code failure

10 1 frontend/cypress/cypress/support/commands.js
6 27 frontend/cypress/e2e/PropertyTesting.cy.js
8 1 frontend/cypress/support/commands.js
2 0 frontend/cypress/support/index.js

b4bed40 - Nicholas W, 4 weeks ago : Name Typo

1 1 .github/workflows/Testing.yml

921be80 - Nicholas W, 4 weeks ago : Adding my name

2 2 .github/workflows/Testing.yml

1d6c649 - JRB958, 4 weeks ago : Merge branch 'development' into TASK-US9-INPUT-FIELD-TO-SEND-REGISTRATION-KEY-78

0a63df8 - JRB958, 4 weeks ago : fix a side bug noticed for update company profile info

38 16 frontend/src/components/userProfile/UserProfile.js

9e70927 - JRB958, 4 weeks ago : update the e2e cyperss tests to pass based on improvements

10 11 frontend/cypress/e2e/AppTesting.cy.js
135 162 frontend/cypress/e2e/PropertyTesting.cy.js
1 1 frontend/src/components/nagationBar/NavigationBar.js
3 3 frontend/src/components/property/PropertyPage.js

19b09ff - Nicholas W, 4 weeks ago : ByPass Rules

2 2 .github/workflows/Testing.yml

540b622 - Nicholas W, 4 weeks ago : Local User ByPAss

2 2 .github/workflows/Testing.yml

a9da8aa - Nicholas W, 4 weeks ago : ByPass

0 1 .github/workflows/Testing.yml

b42c4b2 - Nicholas W, 4 weeks ago : By Pass

These are small changes

23 10 .github/workflows/Testing.yml

1bfa98c - eo2000, 4 weeks ago : Update task branch

6f51198 - Nicholas W, 4 weeks ago : By pass

1 3 .github/workflows/Testing.yml

77a7a95 - Nicholas W, 4 weeks ago : Adding Rule GitHUB TOKEN

This changes are very small and do not require approval or pull requests
2 0 .github/workflows/Testing.yml

6b07437 - Nicholas W, 4 weeks ago : Pull Request Creation (By Passing)

11 2 .github/workflows/Testing.yml

bbf9b8b - eo2000, 4 weeks ago : Task - US13 react component for financial info

149 16 frontend/src/components/dashboard/Financial.js

e86f6c6 - patrickmac3, 4 weeks ago : NO-US SonarCloud Code Analysis (#164)

- Adding the default sonar cloud for github actions.
- This should create upon every commit to the development 'branch' or 'main' a current analysis of code

Relates: #163

528bab7 - JRB958, 4 weeks ago : Merge branch 'development' into
TASK-US9-INPUT-FIELD-TO-SEND-REGISTRATION-KEY-78

aaa0fa1 - JRB958, 4 weeks ago : Fixed - Bug nous unable to create new property 161 (#168)

Bug occurs when trying to create a new property. It returns a status code 400: bad request.

This is a severe bug, it disables a main feature of the site.

Form parsers were added to the views.py, since this form takes in an image file.

In the react code, the file input is handled differently using handleImageChange.

Related to Bug #161

cd27b51 - Nicholas W, 4 weeks ago : Overriding Everything

1 1 .github/workflows/Testing.yml

7103144 - JRB958, 4 weeks ago : NO-US Integrate reports into README using workflows 162 (#118)

Purpose: Ensure that the read me updates github push made by repository committers.

This pull request creates README.md within the repository.

Adding indicators such as

-Quality Gate Status

-Bugs

-Code Smells

-Duplicated Lines (%)

-Lines of Code

-Reliability Rating

-Security Rating

-Technical Debt

-Maintainability Rating

-Vulnerabilities

It ensures per every commit made within the repo that these new values are reflected and add testing reports that are up to date in a convenient location

Relates: #162

cce3f55 - github-actions[bot], 4 weeks ago : Testing & sending reports to README.md!

15 0 README.md

ea9e33e - Nicholas W, 4 weeks ago : Fixed "" type

11 11 .github/workflows/Testing.yml

c05e77c - Nicholas W, 4 weeks ago : Adding Parameters

Quality Gate Status

Bugs

Code Smells

Duplicated Lines (%)

Lines of Code

Reliability Rating

Security Rating

Technical Debt

Maintainability Rating

Vulnerabilities

16 1 .github/workflows/Testing.yml

0d7e408 - github-actions[bot], 4 weeks ago : Add submission link, Codacy badge, and testing reports

8 0 README.md

c84d990 - Nicholas W, 4 weeks ago : Backward cd ..

1 0 .github/workflows/Testing.yml

52b080e - Nicholas W, 4 weeks ago : Update Testing.yml

21 21 .github/workflows/Testing.yml

7808013 - Nicholas W, 4 weeks ago : Update Testing.yml

2 2 .github/workflows/Testing.yml

fd15489 - Nicholas W, 4 weeks ago : Update Testing.yml

23 16 .github/workflows/Testing.yml

4fd381f - github-actions[bot], 4 weeks ago : initial commit

45 91 README.md

d26a83f - Nicholas W, 4 weeks ago : Replacing with frontend

1 1 .github/workflows/Testing.yml

c3ce76a - Nicholas W, 4 weeks ago : Replacing README.md

3 3 .github/workflows/Testing.yml

75203f3 - github-actions[bot], 4 weeks ago : initial commit

48 109 frontend/README.md

ba63306 - Nicholas W, 4 weeks ago : Update Testing.yml

2 2 .github/workflows/Testing.yml

1000d1d - Nicholas W, 4 weeks ago : Update Testing.yml

2 2 .github/workflows/Testing.yml

d518d0f - Vithu, 4 weeks ago : Add an upload documents button in the property page

33 5 frontend/src/components/property/PropertyPage.js

68d3268 - Vithu, 4 weeks ago : Add unit fees section in the unit's info card

1 1 frontend/src/components/property/PropertyCard.js

c0ee8f8 - Vithu, 4 weeks ago : Add fee rate input section for add property form

44 14 frontend/src/components/createProperty/CreateProperty.js

d7ea0c2 - Nicholas W, 4 weeks ago : Fixing typo

2 2 .github/workflows/Testing.yml

28d0089 - Nicholas W, 4 weeks ago : Update Testing.yml

1 0 .github/workflows/Testing.yml

9a01958 - Nicholas W, 4 weeks ago : Update Testing.yml

6 0 .github/workflows/Testing.yml

7aa4faa - Nicholas W, 4 weeks ago : Update Testing.yml

6 1 .github/workflows/Testing.yml

dd7b5f5 - Nicholas W, 4 weeks ago : Fixed Typo

1 1 .github/workflows/Testing.yml

6fd6021 - Nicholas W, 4 weeks ago : Update Testing.yml

63 51 .github/workflows/Testing.yml

cb206c0 - Nicholas W, 4 weeks ago : Trying to add System tests automatically

3 1 .github/workflows/Testing.yml

0fde2b5 - github-actions[bot], 4 weeks ago : initial commit

1 1 myfile.txt

6fc069e - Nicholas W, 4 weeks ago : Update Testing.yml

4 7 .github/workflows/Testing.yml

60970dc - Nicholas W, 4 weeks ago : Running all unbuntu

2 2 .github/workflows/Testing.yml

a89d611 - Nicholas W, 4 weeks ago : Update Testing.yml

6 28 .github/workflows/Testing.yml

609e35e - Nicholas W, 4 weeks ago : Test

149 149 .github/workflows/Testing.yml

91a87a9 - github-actions[bot], 4 weeks ago : initial commit

```
1 0 myfile.txt
create mode 100644 myfile.txt
```

c25d39c - Nicholas W, 4 weeks ago : Update Testing.yml

151 125 .github/workflows/Testing.yml

eb6e28a - JRB958, 4 weeks ago : Task us11 endpoint for retrieving company property fee 133 (#167)

-added view for calculating a companies revenue based on their properties and their property fees for units

-updated the endpoints accordingly

related issue: <https://github.com/THE-390-Team/THE-390/issues/133>

b7a1bb6 - Nicholas W, 4 weeks ago : Update Testing.yml

6 8 .github/workflows/Testing.yml

7e4251b - Yash Patel, 4 weeks ago : Change in file field is handled with handleImageChange.

12 5 frontend/src/components/createProperty/CreateProperty.js

d63844d - Nicholas W, 4 weeks ago : Update Testing.yml

1 0 .github/workflows/Testing.yml

40ab380 - Nicholas W, 4 weeks ago : Update Testing.yml

1 1 .github/workflows/Testing.yml

c253d89 - patrickmac3, 4 weeks ago : Update email host password

1 1 backend/core/settings.py

c411795 - Nicholas W, 4 weeks ago : Update Testing.yml

2 2 .github/workflows/Testing.yml

73162c3 - patrickmac3, 4 weeks ago : Add CompanyFinanceView to retrieve company finance information

77 1 backend/finance/views.py
5 1 backend/user_profile/urls.py

1843754 - patrickmac3, 4 weeks ago : Add default value for fee_rate field in PropertyProfile model and added a name field

2 2 backend/properties/models.py

c5da8bf - Nicholas W, 4 weeks ago : Github Repo addition

1 1 .github/workflows/Testing.yml

f3427a9 - Nicholas W, 4 weeks ago : Update Testing.yml

1 1 .github/workflows/Testing.yml

1932e3c - Nicholas W, 4 weeks ago : Update Testing.yml

1 0 .github/workflows/Testing.yml

05c4c57 - Nicholas W, 4 weeks ago : Update Testing.yml

1 2 .github/workflows/Testing.yml

f2d2768 - JRB958, 4 weeks ago : create a page title component

16 0 frontend/src/components/LargeTitle.js
create mode 100644 frontend/src/components/LargeTitle.js

925ba35 - JRB958, 4 weeks ago : add css for page title

19 1 frontend/src/index.css

7abf702 - JRB958, 4 weeks ago : replace page title

3 3 frontend/src/components/dashboard/DashBoard.js

dc13705 - JRB958, 4 weeks ago : add page title

2 0 frontend/src/components/log/Login.js

7dd26df - JRB958, 4 weeks ago : add page title

2 0 frontend/src/components/log/SignUp.js

5b41d01 - JRB958, 4 weeks ago : add page title to signup company

2 1 frontend/src/components/log/SignUpCompany.js

989e306 - JRB958, 4 weeks ago : cast the load of the api call to int to match the models

3 3 frontend/src/components/registrationKey/SendRegistrationButton.js

6870669 - JRB958, 4 weeks ago : create a component that provides a button to submit a registration key the component does an api call awaiting a debug

83 0 frontend/src/components/registrationKey/SubmitRegistrationButton.js
create mode 100644 frontend/src/components/registrationKey/SubmitRegistrationButton.js

f3d843e - MinhDuc1711, 4 weeks ago : Added operation folder and operation page. Added necessary dependencies and datepicker Added link to operation from nav-bar. Minor changes to FinancialPublic UI

92 0 frontend/package-lock.json

```
2      1      frontend/package.json
2      0      frontend/src/App.js
13     4      frontend/src/components/dashboard/FinancialPublic.js
9      0      frontend/src/components/navigationBar/NavigationBar.js
142    0      frontend/src/components/operationCost/Operation.js
create mode 100644 frontend/src/components/operationCost/Operation.js
```

376c1d8 - JRB958, 4 weeks ago : update page title and add a (send key) button for the registration key

```
7      18     frontend/src/components/userProfile/UserProfile.js
```

aa76caf - Yash Patel, 4 weeks ago : Added form parsers to handle the property image upload

```
2      1      backend/properties/views.py
```

0a1856f - Nicholas W, 4 weeks ago : Update With git pull instead of orinig

```
3      3      .github/workflows/Testing.yml
```

de861f9 - Nicholas W, 4 weeks ago : Update Testing.yml

```
0      14     .github/workflows/Testing.yml
```

d5989f1 - Nicholas W, 4 weeks ago : Update Testing.yml

```
14     0      .github/workflows/Testing.yml
```

cde3d01 - Nicholas W, 4 weeks ago : Update PropertyTesting.cy.js

```
1      1      frontend/cypress/e2e/PropertyTesting.cy.js
```

1adcf23 - Nicholas W, 4 weeks ago : Update PropertyTesting.cy.js

```
36     36     frontend/cypress/e2e/PropertyTesting.cy.js
```

9ffd529 - Nicholas W, 4 weeks ago : Disabling Test Frontend

11 11 frontend/cypress/e2e/AppTesting.cy.js

9619d72 - Nicholas W, 4 weeks ago : Removing unit tests for testing purposes

15 14 .github/workflows/Testing.yml

e764979 - Nicholas W, 4 weeks ago : Update Secrets

3 3 .github/workflows/Testing.yml

5294f04 - Nicholas W, 4 weeks ago : Update Testing.yml

6 0 .github/workflows/Testing.yml

dca73b0 - Nicholas W, 4 weeks ago : Added ECHO + KEYS IN ENVIRONMENT FOR AWS SETUP

3 0 .github/workflows/Testing.yml

2a7e590 - Nicholas W, 4 weeks ago : Update Testing.yml

0 1 .github/workflows/Testing.yml

6f79d95 - Nicholas W, 4 weeks ago : Update Testing.yml

0 5 .github/workflows/Testing.yml

7638c07 - Nicholas W, 4 weeks ago : Update Testing.yml

10 10 .github/workflows/Testing.yml

b1ab249 - Nicholas W, 4 weeks ago : Updating keys

15 2 .github/workflows/Testing.yml

09d9621 - Nicholas W, 4 weeks ago : Update Testing.yml

1 0 .github/workflows/Testing.yml

293b60a - Nicholas W, 4 weeks ago : Add Environment Variable

2 0 .github/workflows/Testing.yml

96d6d96 - Nicholas W, 4 weeks ago : Update Testing.yml

2 0 .github/workflows/Testing.yml

e80379e - Nicholas W, 4 weeks ago : Added acess key

4 0 .github/workflows/Testing.yml

b28fb63 - NicholasWahome, 4 weeks ago : Merge branch 'development' into NOUS-reportsIntoReadMe

afdf6f7d - Nicholas W, 4 weeks ago : Fixed syntax error

2 2 .github/workflows/Testing.yml

b759926 - Nicholas W, 4 weeks ago : Update Formatting

1 1 .github/workflows/Testing.yml

8d27827 - Nicholas W, 4 weeks ago : Update Testing.yml

1 1 .github/workflows/Testing.yml

8f9ef77 - Nicholas W, 4 weeks ago : Adding coverage_report to readme

12 12 .github/workflows/Testing.yml

d247b6b - Nicholas W, 4 weeks ago : Update Testing.yml

16 20 .github/workflows/Testing.yml

1f2ea87 - MinhDuc1711, 4 weeks ago : Minor changes to dashboard UI

2 3 frontend/src/components/dashboard/DashBoard.js
7 10 frontend/src/components/dashboard/FinancialPublic.js

aef1e4b - MinhDuc1711, 4 weeks ago : Added Financial accordion to dashboard, moved financial card for public to FinancialPublic.js

4 3 frontend/src/components/dashboard/DashBoard.js
83 0 frontend/src/components/dashboard/FinancialPublic.js
create mode 100644 frontend/src/components/dashboard/FinancialPublic.js

5115bfa - Nicholas W, 4 weeks ago : TEST

19 14 .github/workflows/Testing.yml

b66d210 - Nicholas W, 4 weeks ago : Update to file

PERFORMING TEST

24 21 .github/workflows/sonarqube.yml

cc441ec - Nicholas W, 4 weeks ago : Updating the secrets key for sonar

PERFORMING TEST

1 2 .github/workflows/sonarqube.yml

6f8d0a0 - Nicholas W, 4 weeks ago : NO-US SonarCloud Code Analysis

- Adding the default sonar cloud for github actions.
- This should create upon every commit to the development 'branch' or 'main' a current analysis of code

Relates: #163

66 0 .github/workflows/sonarqube.yml
create mode 100644 .github/workflows/sonarqube.yml

32c2dfc - JRB958, 4 weeks ago : update the db with user profiles to test registration

- - backend/db.sqlite3

4b307d4 - JRB958, 4 weeks ago : create a component that allows sending of registration key to users it's still not complete can select a unit and user having issues with the endpoint

164 0 frontend/src/components/registrationKey/SendRegistrationButton.js
create mode 100644 frontend/src/components/registrationKey/SendRegistrationButton.js

3782e98 - JRB958, 4 weeks ago : fix a typo eventKey -> eventkey that was causing a console error

4 4 frontend/src/components/navigationBar/NavigationBar.js

d242241 - JRB958, 4 weeks ago : fix an error in a form that was not controlled and causing console error

25 26 frontend/src/components/log/SignUp.js

2bd9d44 - JRB958, 4 weeks ago : add the (send a key) button to the dashboard of a company employee

3 2 frontend/src/components/dashboard/DashBoard.js

bf37e92 - JRB958, 4 weeks ago : update the db to have some data for testing financial properties

- - backend/db.sqlite3

8bc3642 - JRB958, 4 weeks ago : Fix front end input validation to properly check the length of user input (BUG) (#159)

- Addressed issue where input data size validation was incorrect.
- Modified it for assessing input size to ensure accurate validation.
- Overall increasing front-end interface & experience.
- Severity: Low

- Relates to task: #68

- Relates to US: #6

01146b4 - patrickmac3, 4 weeks ago : Task api for secure uploading and serving of property pictures 141 (#160)

Pictures are uploaded and served from an s3 bucket.

.env file must be downloaded from google drive and inserted in the backend folder.

It is in the "development" folder in the "THE-390" google drive.

Rename "env" to ".env" once its added to backend.

Alongside the profile pictures, the images for properties, condo units, storage units and parking units are stored in the same bucket.

![image](https://github.com/THE-390-Team/THE-390/assets/116601995/f815a15f-6ae9-441e-ae
cf-8b33cd0a3cb0)

![image](https://github.com/THE-390-Team/THE-390/assets/116601995/9f41092c-3c51-4286-95
30-6b8909dc366e)

related US: #11

ef2232d - MinhDuc1711, 4 weeks ago : TASK-US6-#68-Bug fix: Fixed validation to properly check the length of the input

3	3	frontend/src/components/createProperty/CreateLocker.js
3	3	frontend/src/components/createProperty/CreateParking.js
1	1	frontend/src/components/createProperty/CreateProperty.js
3	3	frontend/src/components/createProperty/CreateUnit.js

2a21b8d - Yash Patel, 4 weeks ago : Changed width for unit listgroups

3	17	frontend/src/components/property/PropertyPage.js
---	----	--

7c4f4b8 - Yash Patel, 4 weeks ago : Merge branch 'development' into
TASK-api-for-secure-uploading-and-serving-of-property-pictures-141 Removed duplicate in
propertypage

45b10e9 - MinhDuc1711, 4 weeks ago : TASK-US4 Update the UI for the website to be more dynamic and colourful-4 (#157)

This PR has:

- compile all the UI work including (change theme, form checking, page titles)

- now the website has a lighter theme with better organization of properties for the ease of view

- possible bug fix needed: at first visit to the dashboard, it doesn't show any properties until visited again after going to the profile page and back.

related stories: #4 #156

c50960b - JRB958, 4 weeks ago : fix some strange mistake after merging, duplicate components deleted

1 15 frontend/src/components/dashboard/DashBoard.js

eaf76a1 - Yash Patel, 4 weeks ago : Profile picture is now visible

- - backend/db.sqlite3
3 8 frontend/src/utils/hooks/ProfileContext.js

45830ee - JRB958, 4 weeks ago : Merge branch 'development' into ENHANCEMENT-UPDATE-UI-FOR-WEBSITE

5ae9616 - JRB958, 4 weeks ago : resolve merge conflicts from the development branch before making a pr

40 43 README.md
14 1 backend/core/settings.py
1 1 backend/core/urls.py
- - backend/db.sqlite3
3 0 backend/finance/admin.py
6 0 backend/finance/apps.py
18 0 backend/finance/models.py
3 0 backend/finance/tests.py

```
3      0    backend/finance/views.py
134    15    backend/properties/models.py
4      2    backend/properties/serializers.py
6      1    backend/registration_key/admin.py
60     0    backend/registration_key/migrations/0001_initial.py
109    23    backend/registration_key/models.py
48     0    backend/registration_key/serializers.py
465    1    backend/registration_key/tests.py
22     0    backend/registration_key/urls.py
187    2    backend/registration_key/views.py
65     8    backend/user_profile/models.py
4      0    backend/user_profile/urls.py
132    6    backend/user_profile/views.py
8      1    frontend/src/App.js
29     29   frontend/src/components/createProperty/CreateUnit.js
29     7    frontend/src/components/dashboard/DashBoard.js
4      0    frontend/src/components/log/LogOut.js
24     3    frontend/src/components/property/PropertyCard.js
61     12   frontend/src/components/property/PropertyContainer.js
4      35   frontend/src/components/property/PropertyPage.js
3      1    frontend/src/components/userProfile/UserProfile.js
3      0    frontend/src/index.css
17     1    frontend/src/utils/hooks/AuthContext.js
18     5    frontend/src/utils/hooks/ProfileContext.js
81     65   frontend/src/utils/hooks/PropertyContext.js
create mode 100644 backend/finance/admin.py
create mode 100644 backend/finance/apps.py
create mode 100644 backend/finance/models.py
create mode 100644 backend/finance/tests.py
create mode 100644 backend/finance/views.py
create mode 100644 backend/registration_key/migrations/0001_initial.py
create mode 100644 backend/registration_key/serializers.py
create mode 100644 backend/registration_key/urls.py
```

12c1588 - MinhDuc1711, 4 weeks ago : TASK-US5-Enhancement add page titles-58 (#154)

in this PR:

add a page title to every page to ensure ease of navigation for the users

add a title for the Home Screen with a welcoming message

related to #2 #5 #6 #7 #11 #58

16c77ed - MinhDuc1711, 4 weeks ago : Task us6 frontend form validation 68 (#155)

in this PR:

create form validation for ALL forms currently on the website to ensure no input will be invalid:

- this will avoid any bad request sent to the server due to type incompatibility
- ensure the user enters all the needed information

related to #68

7ce2b07 - Yash Patel, 4 weeks ago : Task US6 change dashboard based on role #11 (#129)

in this PR the dashboard will only shows what the role of the user allows

before:

- any user role (PUBLIC or COMPANY) will show the same properties (all properties in the db)

after:

- COMPANY profiles will only see the PROPERTIES related to their company id
- PUBLIC profiles will only see the UNITS related to their id

Notes:

- to test company please use : Joud.babik@gmail.com 123qweasd to login
 - to test public please user: test@example.com password123 to login
- the parking and storage will be outputted once the db model can support them

related to:

#11 #7

2b7c572 - JRB958, 4 weeks ago : merge parent branch

1 1 frontend/src/components/createProperty/CreateLocker.js

7cf17f5 - JRB958, 4 weeks ago : add comments to changes

1 0 frontend/src/components/dashboard/DashBoard.js
1 0 frontend/src/components/dashboard/Financial.js

2 1 frontend/src/components/dashboard/SubmittedRequests.js
6 5 frontend/src/components/userProfile/UserProfile.js
3 1 frontend/src/screens/HomeScreen.js

2c32f3d - Yash Patel, 4 weeks ago : storage unit and parking unit images are being shown

2 0 frontend/src/components/property/PropertyPage.js

8821670 - MinhDuc1711, 4 weeks ago : Added input requirements for all fields in a form, added default province value, added additional constraints to match backend model, further minor fixes

31 10 frontend/src/components/createProperty/CreateLocker.js
34 10 frontend/src/components/createProperty/CreateParking.js
23 20 frontend/src/components/createProperty/CreateProperty.js
39 15 frontend/src/components/createProperty/CreateUnit.js
14 13 frontend/src/components/log/SignUp.js
14 13 frontend/src/components/log/SignUpCompany.js
1 1 frontend/src/components/property/PropertyPage.js

b576667 - Yash Patel, 4 weeks ago : Added image field to unit serializers. Property page shows default condo unit image

- - backend/db.sqlite3
3 3 backend/properties/serializers.py
2 1 frontend/src/components/property/PropertyPage.js

75a17f7 - Yash Patel, 4 weeks ago : added fields for parking, condo and storage unit images

- - backend/db.sqlite3
28 0 backend/properties/migrations/0005_condounit_image_parkingunit_image_storageunit_image.py
3 0 backend/properties/models.py
create mode 100644
backend/properties/migrations/0005_condounit_image_parkingunit_image_storageunit_image.py

e94ed4e - Yash Patel, 4 weeks ago : renamed propertyImage to image

```
-      -      backend/db.sqlite3
18    0
backend/properties/migrations/0004_rename_propertyimage_propertyprofile_image.py
1    1      backend/properties/models.py
1    1      backend/properties/serializers.py
1    1      frontend/src/components/property/PropertyCard.js
2    2      frontend/src/utils/hooks/PropertyContext.js
create mode 100644
backend/properties/migrations/0004_rename_propertyimage_propertyprofile_image.py
```

5f7070a - Yash Patel, 4 weeks ago : Added property image field and did migration

```
-      -      backend/db.sqlite3
24    0      backend/properties/migrations/0003_propertyprofile_fee_rate_and_more.py
1    0      backend/properties/models.py
1    1      backend/properties/serializers.py
73    0
backend/registration_key/migrations/0002_rename_is_activate_condoregistrationkey_is_active_
and_more.py
create mode 100644
backend/properties/migrations/0003_propertyprofile_fee_rate_and_more.py
create mode 100644
backend/registration_key/migrations/0002_rename_is_activate_condoregistrationkey_is_active_
and_more.py
```

de7f1e2 - Yash Patel, 4 weeks ago : Properties now have a default image shown on the card

```
1    1      frontend/src/components/property/PropertyCard.js
3    1      frontend/src/utils/hooks/PropertyContext.js
```

d18d0df - Yash Patel, 4 weeks ago : Added settings and requirements for connecting to s3

```
1    1      backend/.gitignore
16   6      backend/core/settings.py
-      -      backend/media/avatar_images/testpfp.jpg
1    0      backend/requirements.txt
delete mode 100644 backend/media/avatar_images/testpfp.jpg
```

c8afe4 - Yash Patel, 4 weeks ago : removed local storage location for media. Added amazon storage location

9 6 backend/core/settings.py

cc1a6aa - Yash Patel, 4 weeks ago : Added python packages for amazon s3

8 0 backend/requirements.txt

f0b9dd5 - Vithu, 4 weeks ago : Modified font of Your Requests title

1 1 frontend/src/components/dashboard/SubmittedRequests.js

3259194 - Vithu, 4 weeks ago : Modified font of Financial Details title

1 1 frontend/src/components/dashboard/Financial.js

2500b86 - Vithu, 4 weeks ago : Modified font of Properties title

1 1 frontend/src/components/dashboard/DashBoard.js

bc6bb76 - Vithu, 4 weeks ago : Adding a title to the User Profile screen

3 0 frontend/src/components/userProfile/UserProfile.js

44300e6 - Vithu, 4 weeks ago : Adding a title to the Home screen

4 0 frontend/src/screens/HomeScreen.js

fb4d1b5 - MinhDuc1711, 4 weeks ago : Added validation for login and signups

80 29 frontend/src/components/log/Login.js

197 61 frontend/src/components/log/SignUp.js

172 38 frontend/src/components/log/SignUpCompany.js

561c87b - MinhDuc1711, 4 weeks ago : Edited Create locker/parking/property/unit

15 11 frontend/src/components/createProperty/CreateLocker.js

15 11 frontend/src/components/createProperty/CreateParking.js
13 10 frontend/src/components/createProperty/CreateProperty.js
14 10 frontend/src/components/createProperty/CreateUnit.js

445c017 - JRB958, 4 weeks ago : Task us9 registration key testing 80 (#128)

1. Tested endpoints for creating and sending registration keys
2. verifies registration key data is consistent
3. verifies creation of registration key

Related Issue: #80

195138b - patrickmac3, 4 weeks ago : Task us11 calculate condo fees 125 (#130)

Added Finance app to calculate property fees for individual units
added fee rate for property profile models
Updated unit serializer

related issue: #125 , #131

381c36d - patrickmac3, 4 weeks ago : Refactor property and unit models and serializers

10 3 backend/finance/models.py
130 20 backend/properties/models.py
0 10 backend/properties/serializers.py

76ab8c6 - JRB958, 4 weeks ago : merge development into the branch before the PR

c7f44f5 - JRB958, 4 weeks ago : Merge branch 'development' into
TASK-US4-CHANGE-DASHBOARD-BASED-ON-ROLE-#4

ede316f - JRB958, 4 weeks ago : fix issue of having the previous user data after logout

3 0 frontend/src/components/log/LogOut.js
1 1 frontend/src/components/property/PropertyContainer.js
11 1 frontend/src/utils/hooks/PropertyContext.js

4e43a85 - patrickmac3, 4 weeks ago : Add property fee field and calculate property fee in
serializers

9 0 backend/properties/models.py

14 2 backend/properties/serializers.py

f38efec - patrickmac3, 4 weeks ago : Add finance app to INSTALLED_APPS

```
1 0 backend/core/settings.py
3 0 backend/finance/admin.py
6 0 backend/finance/apps.py
11 0 backend/finance/models.py
3 0 backend/finance/tests.py
3 0 backend/finance/views.py
create mode 100644 backend/finance/admin.py
create mode 100644 backend/finance/apps.py
create mode 100644 backend/finance/models.py
create mode 100644 backend/finance/tests.py
create mode 100644 backend/finance/views.py
```

0cc7518 - JRB958, 4 weeks ago : update the dashboard to use the profile context for user role fetching

7 11 frontend/src/components/dashboard/DashBoard.js

d1e5b2e - JRB958, 4 weeks ago : update the unit card to support both property output and unit output based on user type

24 3 frontend/src/components/property/PropertyCard.js

16fb37d - JRB958, 4 weeks ago : update the property container to support propertycard and unitcard

62 61 frontend/src/components/property/PropertyContainer.js

9d4a99e - JRB958, 4 weeks ago : fix accidental deletion made on the last commit on this file

9 2 frontend/src/utils/hooks/PropertyContext.js

c4e2c2e - JRB958, 4 weeks ago : add a fetch role to get the user role right after login

3 1 frontend/src/components/userProfile/UserProfile.js

b8dc9cf - JRB958, 4 weeks ago : move the user role and the method to fetch it to ProfileContext

18 5 frontend/src/utils/hooks/ProfileContext.js

dca21a2 - JRB958, 4 weeks ago : add three fetch methods and update an existing one: update the fetch method that used to get all properties to get only company specific properties create three fetch calls to get the condo, parking and locker units linked to a specific public user

60 11 frontend/src/utils/hooks/PropertyContext.js

497b146 - MinhDuc1711, 4 weeks ago : Added form validation to create property/parking/locker/unit. Added comments for readability

117 47 frontend/src/components/createProperty/CreateLocker.js
111 39 frontend/src/components/createProperty/CreateParking.js
122 48 frontend/src/components/createProperty/CreateProperty.js
108 28 frontend/src/components/createProperty/CreateUnit.js

ecb43af - JRB958, 4 weeks ago : remove the id from local storage when logout

1 0 frontend/src/components/log/LogOut.js

7feaf61 - patrickmac3, 4 weeks ago : implemented tests for Registration key endpoints

465 1 backend/registration_key/tests.py

2422abe - patrickmac3, 4 weeks ago : Add status code to response in registration key views

3 3 backend/registration_key/views.py

fb5d9a2 - patrickmac3, 4 weeks ago : Task us9 api endpoints registration key 77 (#124)

1. created endpoints for creating the various unit registration keys
2. created endpoints for registering unit to user (public profile) through the registration key

1a65ab3 - JRB958, 4 weeks ago : only company profiles can create property (#123)

Small PR concerning what can be seen from the dashboard based on the user type (PUBLIC or COMPANY)

before: public users were able to create properties because they had access to the "add property" button

after:

- implement dashboard view based on the type of user (public or company)
- the option to create the property only shows for "COMPANY" roles now

Relates: #4

eb0bd8c - JRB958, 4 weeks ago : add comments to previous PR commits

3 0 frontend/src/components/dashboard/DashBoard.js
2 1 frontend/src/components/property/PropertyPage.js

bd195ba - patrickmac3, 4 weeks ago : Refactor registration key URL patterns

12 2 backend/registration_key/urls.py

a24ae88 - patrickmac3, 4 weeks ago : Fix registration key email sending

3 3 backend/registration_key/views.py

33c338b - patrickmac3, 4 weeks ago : added comments for registration key viewsets

36 2 backend/registration_key/views.py

d10bdb - JRB958, 4 weeks ago : change the api call from property page to the property context when seeing a single property page

3 35 frontend/src/components/property/PropertyPage.js
4 54 frontend/src/utils/hooks/PropertyContext.js

dc67149 - JRB958, 4 weeks ago : Merge branch 'development' into TASK-US4-CHANGE-DASHBOARD-BASED-ON-ROLE-#4

92a8816 - JRB958, 4 weeks ago : Bugfix refresh page 117 (#122)

add a method that checks for tokens in local storage on every page load
in App.js

a83d9ee - patrickmac3, 4 weeks ago : Add registration endpoints to public profile

4 0 backend/user_profile/urls.py

4a348b6 - patrickmac3, 4 weeks ago : Add registration methods to PublicProfileViewSet

132 6 backend/user_profile/views.py

3c78857 - patrickmac3, 4 weeks ago : Add company.send_registration_key() method call in registration key views

4 0 backend/registration_key/views.py

30b9d27 - patrickmac3, 4 weeks ago : Fix serializers in registration_key module

4 4 backend/registration_key/serializers.py

236e447 - JRB958, 4 weeks ago : only company profiles can create property

24 2 frontend/src/components/dashboard/DashBoard.js

c4e0580 - JRB958, 4 weeks ago : add method to check for tokens in local storage

3 2 frontend/src/App.js

5 4 frontend/src/utils/hooks/AuthContext.js

1a1049d - JRB958, 4 weeks ago : add method to check for tokens in local storage

6 0 frontend/src/App.js

16 1 frontend/src/utils/hooks/AuthContext.js

44c5c70 - JRB958, 4 weeks ago : add titles to create unit pages, change the styling file

12 0 frontend/src/bootstrap.min-3.css

```
8      2      frontend/src/components/createProperty/CreateLocker.js
9      1      frontend/src/components/createProperty/CreateParking.js
11     0      frontend/src/components/createProperty/CreateUnit.js
1      1      frontend/src/index.js
create mode 100644 frontend/src/bootstrap.min-3.css
```

1caff2c - JRB958, 4 weeks ago : property page layout change

```
74     29     frontend/src/components/property/PropertyPage.js
```

4167847 - JRB958, 4 weeks ago : update the nagivation bar

```
1      3      frontend/src/components/Header.js
82     0      frontend/src/components/nagivationBar/NavigationBar.js
create mode 100644 frontend/src/components/nagivationBar/NavigationBar.js
```

07c1004 - JRB958, 4 weeks ago : add the grid container css for the property page

```
26     0      frontend/src/index.css
```

670abb1 - JRB958, 4 weeks ago : layout changes and refractoring

```
6      31     frontend/src/components/Header.js
10     8      frontend/src/components/dashboard/DashBoard.js
1      1      frontend/src/components/dashboard/Financial.js
1      1      frontend/src/components/dashboard/SubmittedRequests.js
1      1      frontend/src/components/property/PropertyCard.js
4      55     frontend/src/components/property/PropertyContainer.js
```

0584cb5 - JRB958, 4 weeks ago : change default route to homepage after login

```
1      1      frontend/src/components/log/Login.js
```

6d645a2 - JRB958, 4 weeks ago : change the address presentation

```
5      4      frontend/src/components/userProfile/UserProfile.js
```

dd42a7e - JRB958, 4 weeks ago : add padding to homescreen

1 1 frontend/src/screens/HomeScreen.js

823dbda - JRB958, 4 weeks ago : create a carousel for homepage

1 1 frontend/src/screens/homeScreenCarousel/CarouselHomeScreen.js

6fd54a2 - patrickmac3, 4 weeks ago : Add StorageRegistrationKeyView and update ParkingRegistrationKeyView

2 1 backend/registration_key/urls.py
41 1 backend/registration_key/views.py

831b734 - patrickmac3, 4 weeks ago : Add ParkingRegistrationKeyView to urls.py and views.py

4 1 backend/registration_key/urls.py
41 0 backend/registration_key/views.py

ca70dc9 - patrickmac3, 4 weeks ago : Update RegistrationKey model and generate_key method

10 12 backend/registration_key/models.py

39c8731 - patrickmac3, 4 weeks ago : Add registration key URL pattern

1 1 backend/core/urls.py

e4d8db8 - patrickmac3, 4 weeks ago : Add condo registration key functionality and endpoint

8 0 backend/registration_key/urls.py
68 2 backend/registration_key/views.py
create mode 100644 backend/registration_key/urls.py

f089893 - patrickmac3, 4 weeks ago : Add serializers for registration key models

48 0 backend/registration_key/serializers.py
create mode 100644 backend/registration_key/serializers.py

0de1d44 - patrickmac3, 4 weeks ago : Task us9 generating registration key 76 (#119)

1. Created Abstract Registration key for linking a public profile user to a unit
2. Create Concrete Registration keys that inherit from abstract Registration Key model
3. implemented method for generating the key value and method for creating the registration key object.
4. Added method to Company Profile model for sending the registration key to a given user's email

Related Issue: #76

90dcb23 - patrickmac3, 4 weeks ago : Add ParkingRegistrationKey and StorageRegistrationKey to admin.py and rename StorageUnitRegistrationKey to StorageRegistrationKey

3 1 backend/registration_key/admin.py
1 1 backend/registration_key/models.py

339035c - patrickmac3, 4 weeks ago : TODO: Update destination email address in CompanyProfile model

1 0 backend/user_profile/models.py

b5e01a7 - patrickmac3, 4 weeks ago : removed circular import error of registration key manager

0 1 backend/registration_key/models.py

24355b6 - patrickmac3, 4 weeks ago : Fix typo in is_owner and is_active fields and added string representation of the inheriting classes

17 4 backend/registration_key/models.py

25cf83a - Nicholas W, 4 weeks ago : Update Testing.yml

2 2 .github/workflows/Testing.yml

5f8366c - Nicholas W, 4 weeks ago : Update Testing.yml

9 5 .github/workflows/Testing.yml

380f157 - NicholasWahome, 4 weeks ago : Update Testing.yml

9 0 .github/workflows/Testing.yml

009c55f - JRB958, 4 weeks ago : create a caresoul for homepage

1 0 frontend/src/App.js
9 3 frontend/src/screens/HomeScreen.js
80 0 frontend/src/screens/homeScreenCarousel/CarouselHomeScreen.js
- - frontend/src/screens/homeScreenCarousel/property.jpeg
- - frontend/src/screens/homeScreenCarousel/property1.png
- - frontend/src/screens/homeScreenCarousel/property2.jpeg

create mode 100644 frontend/src/screens/homeScreenCarousel/CarouselHomeScreen.js

create mode 100644 frontend/src/screens/homeScreenCarousel/property.jpeg

create mode 100644 frontend/src/screens/homeScreenCarousel/property1.png

create mode 100644 frontend/src/screens/homeScreenCarousel/property2.jpeg

9e5854e - JRB958, 4 weeks ago : Update README.md

correct the format of the tables in the readme

40 43 README.md

90233d1 - patrickmac3, 4 weeks ago : added comments to registration model and user profile model files

55 5 backend/registration_key/models.py
51 7 backend/user_profile/models.py

828cca9 - patrickmac3, 4 weeks ago : registered key to admin site and slight modification to registration key manager

2 2 backend/registration_key/admin.py
10 11 backend/registration_key/models.py

ca24447 - patrickmac3, 4 weeks ago : added functionnality for company to send registration key to user

```
12    1    backend/core/settings.py  
13    1    backend/user_profile/models.py
```

54ba017 - patrickmac3, 5 weeks ago : implemented registration key model

created unique registration key for condo, storage and parking units

```
1    0    backend/core/settings.py  
-    -    backend/db.sqlite3  
4    1    backend/registration_key/admin.py  
60   0    backend/registration_key/migrations/0001_initial.py  
50   23   backend/registration_key/models.py  
create mode 100644 backend/registration_key/migrations/0001_initial.py
```

2fd74c8 - MinhDuc1711, 5 weeks ago : Task us5 api calls for properties from the react 60 (#116)

api calls made from createUnit, createParking, createLocker #114 #105
#106 #107

f8f7f24 - JRB958, 5 weeks ago : changes to pass tests the tests are a little outdated and need update for now will correct them later

```
-    -    backend/db.sqlite3  
127  112  frontend/cypress/e2e/PropertyTesting.cy.js  
10   10   frontend/src/components/createProperty/CreateProperty.js  
2    2    frontend/src/components/createProperty/CreateUnit.js
```

1fd7469 - JRB958, 5 weeks ago : more minor changes to let the tests pass

```
4    1    backend/README.md  
1    1    frontend/cypress/e2e/AppTesting.cy.js
```

6081ae0 - JRB958, 5 weeks ago : fix testing for pr

```
66   0    backend/properties/migrations/0001_initial.py  
52   0    backend/properties/migrations/0002_initial.py  
87   0    backend/user_profile/migrations/0001_initial.py  
create mode 100644 backend/properties/migrations/0001_initial.py  
create mode 100644 backend/properties/migrations/0002_initial.py  
create mode 100644 backend/user_profile/migrations/0001_initial.py
```

3c35b23 - JRB958, 5 weeks ago : change line attributes in alphabetical order in frontend for codacy

```
15    15    frontend/src/components/createProperty/CreateLocker.js  
15    15    frontend/src/components/createProperty/CreateParking.js  
19    19    frontend/src/components/createProperty/CreateUnit.js  
1     1     frontend/src/components/property/PropertyPage.js
```

5c908c3 - JRB958, 5 weeks ago : change line lengths in backend readme for codacy

```
40    18    backend/README.md
```

f259f51 - patrickmac3, 6 weeks ago : updated backend read me for cloning and running the server

```
129    2    backend/README.md
```

16085dc - JRB958, 6 weeks ago : remove migration files

```
0     60    backend/properties/migrations/0001_initial.py  
0     37    backend/properties/migrations/0002_initial.py  
0     60    backend/properties/migrations/0003_condounit_extra_information_condounit_owner_and_more.py  
0     28    backend/properties/migrations/0004_rename_owner_condounit_public_profile_and_more.py  
0     84    backend/user_profile/migrations/0001_initial.py  
0     28    backend/user_profile/migrations/0002_companyprofile_avatar_employeeprofile_avatar_and_more.py  
1     1     frontend/src/components/createProperty/CreateLocker.js  
delete mode 100644 backend/properties/migrations/0001_initial.py  
delete mode 100644 backend/properties/migrations/0002_initial.py  
delete mode 100644  
backend/properties/migrations/0003_condounit_extra_information_condounit_owner_and_more.py  
delete mode 100644  
backend/properties/migrations/0004_rename_owner_condounit_public_profile_and_more.py  
delete mode 100644 backend/user_profile/migrations/0001_initial.py
```

delete mode 100644
backend/user_profile/migrations/0002_companyprofile_avatar_employeeprofile_avatar_and_more.py

b51bb23 - JRB958, 6 weeks ago : Merge remote-tracking branch 'origin' into
TASK-US5-api-calls-for-properties-from-the-REACT-60

d41499b - JRB958, 6 weeks ago : implement successful api calls to create a storage unit

60 35 frontend/src/components/createProperty/CreateLocker.js

b113bef - JRB958, 6 weeks ago : implement successful api calls to create a parking unit

63 49 frontend/src/components/createProperty/CreateParking.js

3ca3359 - JRB958, 6 weeks ago : file clean up and comments for CreateUnit.js

15 9 frontend/src/components/createProperty/CreateUnit.js

e11a46d - JRB958, 6 weeks ago : BUGFIX fix the api call bug to create a new condo unit #114

2 3 frontend/src/components/createProperty/CreateUnit.js

8f4f7c0 - JRB958, 6 weeks ago : update create unit trying to make the request work bug#114

7 1 frontend/src/components/createProperty/CreateParking.js

20 13 frontend/src/components/createProperty/CreateUnit.js

0 4 frontend/src/components/property/PropertyCard.js

1c61fc3 - JRB958, 6 weeks ago : change the back button to dynamic -1

7 3 frontend/src/components/property/PropertyPage.js

1f7bcde - JRB958, 6 weeks ago : update property card output to match db models

7 8 frontend/src/components/property/PropertyCard.js

b9b8d49 - JRB958, 6 weeks ago : update the property state with corresponding db model

1 1 frontend/src/utils/hooks/PropertyContext.js

e4b0945 - Nicholas W, 6 weeks ago : Fixed Table for tests

12 261 README.md

12cc9c4 - Nicholas W, 6 weeks ago : Update Testing.yml

1 1 .github/workflows/Testing.yml

8950e8b - Nicholas W, 6 weeks ago : Removing Dist Packages from Coverage Report

2 2 .github/workflows/Testing.yml

6c8351d - Nicholas W, 6 weeks ago : Update README.md

0 4 README.md

cbcc25a - JRB958, 6 weeks ago : Add files via upload

- - Sprint_2/Release Plan for Sprint 3.pdf
- - Sprint_2/Risk Assessment and Management Plan (RMP).pdf
- - Sprint_2/SAD THE-390.pdf
- - Sprint_2/Software Product Vision.pdf
- - Sprint_2/Sprint 2 retrospective.pdf
- - Sprint_2/Testing Plan for sprint 3.pdf
- - Sprint_2/UI Prototype.pdf
- - Sprint_2/User Stories_THE-390 - Sheet1.pdf

create mode 100644 Sprint_2/Release Plan for Sprint 3.pdf

create mode 100644 Sprint_2/Risk Assessment and Management Plan (RMP).pdf

create mode 100644 Sprint_2/SAD THE-390.pdf

create mode 100644 Sprint_2/Software Product Vision.pdf

create mode 100644 Sprint_2/Sprint 2 retrospective.pdf

create mode 100644 Sprint_2/Testing Plan for sprint 3.pdf

create mode 100644 Sprint_2/UI Prototype.pdf

create mode 100644 Sprint_2/User Stories_THE-390 - Sheet1.pdf

c524593 - JRB958, 6 weeks ago : Task us5 api calls for properties from the react 60 (#113)

- update the property based on the db using the api calls

1d410c2 - JRB958, 6 weeks ago : Merge remote-tracking branch 'origin' into
TASK-US5-api-calls-for-properties-from-the-REACT-60

a914eab - JRB958, 6 weeks ago : link the api calls with the new endpoints

3	3	backend/user_profile/serializers.py
16	5	frontend/src/components/createProperty/CreateLocker.js
17	5	frontend/src/components/createProperty/CreateParking.js
85	85	frontend/src/components/createProperty/CreateProperty.js
27	24	frontend/src/components/createProperty/CreateUnit.js
1	1	frontend/src/components/log/SignUpCompany.js
18	13	frontend/src/components/property/PropertyPage.js
1	0	frontend/src/utils/hooks/ProfileContext.js
7	6	frontend/src/utils/hooks/PropertyContext.js

5fd315f - Nicholas W, 6 weeks ago : Test Reports

343 0 README.md

d5ef9b1 - Nicholas W, 6 weeks ago : Delete unnecessary

0 10 .github/workflows/Testing.yml

ebb5872 - Nicholas W, 6 weeks ago : Secret Github

Integrated with Github Secret

11 0 .github/workflows/Testing.yml

14eeaa1 - JRB958, 6 weeks ago : US8 US3 Unit Tests Multi us 51 75 (#112)

Verify individual commit comments

Summary: Creation of unit testing for 5 components in the repository +
integration of them in the yaml workflow

Relates: #51, #75

ab49135 - JRB958, 6 weeks ago : Merge remote-tracking branch 'origin' into TASK-US5-api-calls-for-properties-from-the-REACT-60

3d46b6f - Nicholas W, 6 weeks ago : Minor Typo Fix

1 1 .github/workflows/Testing.yml

e4a3ef8 - NicholasWahome, 6 weeks ago : Small Update

Retesting with yaml file workflow to ensure that it does run correctly before getting submitting for approval

Relates: #51, #75

1 1 .github/workflows/Testing.yml

097617a - NicholasWahome, 6 weeks ago : Multi-Modification of files

Testing.yml -> modified to incorporate jest testing

modified the package.json

8 1 .github/workflows/Testing.yml
1453 301 frontend/package-lock.json
7 1 frontend/package.json

7db22f6 - JRB958, 6 weeks ago : update createunit to use the updated endpoints for condo-unit creation

3 5 frontend/src/App.js
0 12 frontend/src/components/createProperty/CreateProperty.js
15 15 frontend/src/components/createProperty/CreateUnit.js
2 2 frontend/src/components/property/PropertyCard.js
7 9 frontend/src/components/property/PropertyPage.js

5b073ff - NicholasWahome, 6 weeks ago : Adding Unit Testing (Frontend)

- Unit Testing for the Content
- Unit Testing for Dashboard
- Unit Testing for SubmittedRequests
- Unit Testing for Footer

- Unit Testing for HomeScreen

In order to run use:

npm test

Relates: #51, #75

```
5    0    frontend/jest.config.js
5    0    frontend/package.json
15   0    frontend/src/components/Content.test.js
9    0    frontend/src/components/Footer.test.js
14   0    frontend/src/components/dashboard/Financial.test.js
13   0    frontend/src/components/dashboard/SubmittedRequests.test.js
15   0    frontend/src/screens/HomeScreen.test.js
create mode 100644 frontend/jest.config.js
create mode 100644 frontend/src/components/Content.test.js
create mode 100644 frontend/src/components/Footer.test.js
create mode 100644 frontend/src/components/dashboard/Financial.test.js
create mode 100644 frontend/src/components/dashboard/SubmittedRequests.test.js
create mode 100644 frontend/src/screens/HomeScreen.test.js
```

b6b8b03 - patrickmac3, 6 weeks ago : Task us5 endpoints owner property units 59 (#111)

created endpoints to allow CRUD operations for units belonging or rented out to a public profile

1. updated unit models by adding a public profile user as a foreign key, the size and extra_information
2. updated PublicProfileViewSet to add endpoints for listing the various units related to a public profile
3. updated serializers accordingly
4. for other crud methods for the units, the endpoints have already been generated by the model view set and the simple router

related issue: #59

ed5a395 - patrickmac3, 6 weeks ago : fixed failed django test

```
1    1    backend/properties/models.py
```

f93d6f4 - patrickmac3, 6 weeks ago : Merge remote-tracking branch 'origin' into TASK-US5-endpoints-owner-property-units-59

606cf0a - patrickmac3, 6 weeks ago : minor code clean up

```
2      1      backend/user_profile/urls.py  
24     9      backend/user_profile/views.py
```

a4d003c - patrickmac3, 6 weeks ago : created endpoints for public profile to list their various units

- make function and endpoint for listing each unit for a given user
- added fields to the unit models (size, profile, extra_information)

```
-      -      backend/db.sqlite3  
60     0      backend/properties/migrations/0003_condounit_extra_information_condounit_owner_and_more.py  
28     0      backend/properties/migrations/0004_rename_owner_condounit_public_profile_and_more.py  
5      0      backend/properties/models.py  
9      3      backend/properties/serializers.py  
5      1      backend/user_profile/serializers.py  
5      1      backend/user_profile/urls.py  
43     1      backend/user_profile/views.py  
create mode 100644  
backend/properties/migrations/0003_condounit_extra_information_condounit_owner_and_more.py  
create mode 100644  
backend/properties/migrations/0004_rename_owner_condounit_public_profile_and_more.py
```

e954b13 - JRB958, 6 weeks ago : update README file with Codacy

```
4      1      README.md
```

84be9a8 - JRB958, 6 weeks ago : Added properties testing for front end using cypress and unit testing for backend (#110)

f73a101 - Nicholas W, 6 weeks ago : Modifications to comments

```
2      3      .github/workflows/Testing.yml
```

bdb4385 - Nicholas W, 6 weeks ago : Added Comments

3 2 .github/workflows/Testing.yml

c3e6622 - MinhDuc1711, 6 weeks ago : Merge remote-tracking branch 'origin' into
TASK-US8-property_testing-75

cfde602 - MinhDuc1711, 6 weeks ago : switch location in unit from a int field to a char field

1 1 backend/properties/models.py

7321e06 - Nicholas W, 6 weeks ago : Update Testing.yml

0 5 .github/workflows/Testing.yml

758db2d - Nicholas W, 6 weeks ago : Efficiency Improvement

Fixed 2 lines of code to ensure the workflow goes faster

4 2 .github/workflows/Testing.yml

e5d2b6c - Nicholas W, 6 weeks ago : Update Testing.yml

Artifcat option not necessary

0 4 .github/workflows/Testing.yml

2fec4df - MinhDuc1711, 6 weeks ago : Merge remote-tracking branch 'origin' into
TASK-US8-property_testing-75

cbcd202 - Nicholas W, 6 weeks ago : Forget one of them

Fixed bug

1 0 .github/workflows/Testing.yml

dbbe0d8 - Nicholas W, 6 weeks ago : Fixed Bug on run

Added run to every step in order to run the yml file

2 0 .github/workflows/Testing.yml

6380245 - MinhDuc1711, 6 weeks ago : Fixed backend unit test for properties

7 8 backend/properties/tests.py

0abebbb - Nicholas W, 6 weeks ago : Update and rename systemTestingWF.yml to Testing.yml

23 11 .github/workflows/{systemTestingWF.yml => Testing.yml}
rename .github/workflows/{systemTestingWF.yml => Testing.yml} (70%)

a264d30 - MinhDuc1711, 6 weeks ago : Merge remote-tracking branch 'origin' into
TASK-US8-property_testing-75

f385927 - Nicholas W, 6 weeks ago : Status badge per push

1 1 README.md

e718147 - Nicholas W, 6 weeks ago : Fixing Bug with workflow

2 2 .github/workflows/systemTestingWF.yml

35dfdaa - Nicholas W, 6 weeks ago : Adding Status Badge of workflows

3 1 README.md

e45e65d - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

1 1 .github/workflows/systemTestingWF.yml

7e70ef6 - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

Removing the process for cypress

0 7 .github/workflows/systemTestingWF.yml

1fb2f90 - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

10 6 .github/workflows/systemTestingWF.yml

aeb3093 - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

2 0 .github/workflows/systemTestingWF.yml

53faf70 - JRB958, 6 weeks ago : Task us8 create endpoints for property units 71 (#104)

1. modified inherited list and create methods for all unit (condo, parking, storage) to easily create and list units for property
 - other crud methods were already in the backend with the model viewsets for each unit model
 2. added id fields to the unit serializer
- 283d772 - patrickmac3, 6 weeks ago : added comments for view modifications added id field to unit serializer

1 1 backend/properties/serializers.py
10 7 backend/properties/views.py

b2bbfc2 - Nicholas W, 6 weeks ago : New Cypress Report Structure

Added Cypress Report
3 2 .github/workflows/systemTestingWF.yml

a0ef5ea - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

1 11 .github/workflows/systemTestingWF.yml

c144055 - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

2 1 .github/workflows/systemTestingWF.yml

4daf902 - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

9 10 .github/workflows/systemTestingWF.yml

a8e1fd5 - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

10 9 .github/workflows/systemTestingWF.yml

047010c - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

2 0 .github/workflows/systemTestingWF.yml

fd06202 - JRB958, 6 weeks ago : company employee can create a property that belongs to his company

104 109 frontend/src/components/createProperty/CreateProperty.js
47 26 frontend/src/components/createProperty/CreateUnit.js
56 22 frontend/src/components/property/PropertyPage.js
3 2 frontend/src/utils/hooks/PropertyContext.js

6568513 - MinhDuc1711, 6 weeks ago : Merge remote-tracking branch 'origin' into TASK-US8-property_testing-75

f061883 - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

2 0 .github/workflows/systemTestingWF.yml

437ae54 - patrickmac3, 6 weeks ago : updated endpoints for listing and creating units for a property

- - backend/db.sqlite3
6 1 backend/properties/urls.py
57 8 backend/properties/views.py

aef307b - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

0 4 .github/workflows/systemTestingWF.yml

ae9a26a - MinhDuc1711, 6 weeks ago : Added testing for properties backend. no api endpoint testing yet

35 1 backend/properties/tests.py

3140815 - JRB958, 6 weeks ago : Update systemTestingWF.yml

test comment commit
1 0 .github/workflows/systemTestingWF.yml

e48c404 - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

1 1 .github/workflows/systemTestingWF.yml

0ca322c - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

1 1 .github/workflows/systemTestingWF.yml

bebbbb2 - patrickmac3, 6 weeks ago : Ensure secure file upload for the users picture and make user profile modifiable (#103)

Updated the form in the user profile to save profile picture. Picture is saved locally in backend/media/avatar_pictures.

The profile picture field is called "avatar" and is part of the Profile model.

The PFP upload input was part of a form to update the user profile, therefore I modified the form implementation to enable the user to modify their profile.

Response status is verified to check if profile picture was uploaded.

Resolves: #49

0e3dbae - Nicholas W, 6 weeks ago : NOUS-DirectUpdateToMain

3 1 .github/workflows/systemTestingWF.yml

7cad61f - Yash Patel, 6 weeks ago : Added pillow to requirements

2 0 backend/requirements.txt

642e585 - JRB958, 6 weeks ago : add a sign up page for a company profile to control permissions after

2 0 frontend/src/App.js

6 5 frontend/src/components/log/SignUpCompany.js

6c12547 - JRB958, 6 weeks ago : add a check for user role before fetching from a specific endpoint #60

58 18 frontend/src/utils/hooks/ProfileContext.js

61cb713 - Yash Patel, 6 weeks ago : Response status is checked to see if changes were saved

- - backend/db.sqlite3
7 1 frontend/src/components/userProfile/UserProfile.js

265f343 - MinhDuc1711, 6 weeks ago : Updated package-lock.json and package.json files

16 3 frontend/package-lock.json
1 0 frontend/package.json

5583b07 - MinhDuc1711, 6 weeks ago : Merge remote-tracking branch 'origin' into
TASK-US8-property_testing-75

a405f38 - JRB958, 6 weeks ago : add new sign up page for company sign up updated the sign up forms with photo input add link from the sign in page to new sign up page must confirm db models for photo must confirm db models for company name related to issue #100 related to US#11 #24

1 1 frontend/src/components/log/Login.js
33 33 frontend/src/components/log/SignUp.js
244 0 frontend/src/components/log/SignUpCompany.js
4 2 frontend/src/utils/hooks/ProfileContext.js
create mode 100644 frontend/src/components/log/SignUpCompany.js

852120a - JRB958, 6 weeks ago : minor updates to the property card to match databse models related to #60 and #64

1 1 frontend/src/components/property/PropertyCard.js

c3064d8 - JRB958, 6 weeks ago : update the property context with better state default for #60

25 2 frontend/src/utils/hooks/PropertyContext.js

eb1da94 - Yash Patel, 6 weeks ago : Error being caught from axios patch requests

7 1 frontend/src/components/userProfile/UserProfile.js

447ec64 - Yash Patel, 6 weeks ago : Merge branch 'main' into
TASK-US3-ensure-secure-file-upload-for-the-users-picture-49

61562f4 - Yash Patel, 6 weeks ago : Implemented ability to change all profile information

- - backend/db.sqlite3
12 3 frontend/src/components/userProfile/UserProfile.js

d8f7273 - Yash Patel, 6 weeks ago : User is able to update profile picture

- - backend/db.sqlite3
- - backend/media/avatar_images/testpfp.jpg
41 36 frontend/src/components/userProfile/UserProfile.js
create mode 100644 backend/media/avatar_images/testpfp.jpg

3ad9311 - Nicholas W, 6 weeks ago : NO-US Reports for testing

- Adding report files to main repo as txt files
28 26 .github/workflows/systemTestingWF.yml

9c11010 - Nicholas W, 6 weeks ago : Approving this merge

The validity of the tests is correct.

Related Issue: #45

7237da9 - Nicholas W, 6 weeks ago : Delete .github/workflows/pylint.yml

Not useful anymore. Not necessary
0 23 .github/workflows/pylint.yml
delete mode 100644 .github/workflows/pylint.yml

c9198ed - patrickmac3, 6 weeks ago : Merge remote-tracking branch 'origin' into
TASK-US1-testing-api-endpoints-45

77fa104 - patrickmac3, 6 weeks ago : created tests for user creation endpoints (for public user)

implemented tests for valid and invalid sign up scenarios

- valid sign up
- invalid due to email already in use
- invalid due to invalid email
- invalid due to missing password

Related Issue: #45

```
-      -      backend/db.sqlite3  
0      27     backend/jwt_auth_token/tests.py  
45      52     backend/user_profile/tests.py
```

ea108a0 - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

```
31      2      .github/workflows/systemTestingWF.yml
```

20a2ac4 - Yash Patel, 6 weeks ago : Added MultiPartParser and FormParser to allow img upload

```
-      -      backend/db.sqlite3  
5      2      backend/user_profile/views.py
```

b15ed70 - Yash Patel, 6 weeks ago : Added avatar to the profile serializer

```
-      -      backend/db.sqlite3  
1      1      backend/user_profile/serializers.py
```

3ef074c - Yash Patel, 6 weeks ago : Set avatar in profile context

```
1      1      frontend/src/utils/hooks/ProfileContext.js
```

7a11a45 - Yash Patel, 6 weeks ago : Added media url

```
3      1      backend/core/urls.py
```

2fccf5d - Yash Patel, 6 weeks ago : Added avatar field to Profile model. Migrated

```
-      -      backend/db.sqlite3
-      -      backend/media/avatar_images/pp.jpg
28    0
backend/user_profile/migrations/0002_companyprofile_avatar_employeeprofile_avatar_and_mo
re.py
1      0      backend/user_profile/models.py
create mode 100644 backend/media/avatar_images/pp.jpg
create mode 100644
backend/user_profile/migrations/0002_companyprofile_avatar_employeeprofile_avatar_and_mo
re.py
```

b1ed55f - Yash Patel, 6 weeks ago : Merge branch 'main' into
TASK-US3-ensure-secure-file-upload-for-the-users-picture-49

df747fb - Yash Patel, 6 weeks ago : Added location and url for media files

```
5      0      backend/core/settings.py
```

437de95 - JRB958, 6 weeks ago : start changes needed to access property-page from the
dashboard

```
1      1      frontend/src/App.js
2      2      frontend/src/components/property/PropertyCard.js
4      3      frontend/src/utils/hooks/PropertyContext.js
```

0e02423 - JRB958, 6 weeks ago : Create pylint.yml (#97)

Adding pylint to repository

9a3e5fb - Nicholas W, 6 weeks ago : Create pylint.yml

```
23     0      .github/workflows/pylint.yml
create mode 100644 .github/workflows/pylint.yml
```

5bea727 - MinhDuc1711, 6 weeks ago : Added Testing of Create Property form and Property
page navigation

```
93     35     frontend/cypress/e2e/PropertyTesting.cy.js
-      -      frontend/cypress/fixtures/property-image.png
```

1 0 frontend/src/components/createProperty/CreateProperty.js
1 1 frontend/src/components/property/PropertyPage.js
create mode 100644 frontend/cypress/fixtures/property-image.png

fe5cb19 - JRB958, 6 weeks ago : minor changes and notes added for later to CreateProperty.js

4 3 frontend/src/components/createProperty/CreateProperty.js

09af54e - JRB958, 6 weeks ago : change the card to match db models changed key names in obj to match db updated some elements in the card to match current db until fixed related to #60

11 6 frontend/src/components/property/PropertyCard.js

6e53d37 - JRB958, 6 weeks ago : GET properties from db it show the properties well the card doesn't show the information in the db well will adjust the cards related to #60

57 49 frontend/src/components/property/PropertyContainer.js
52 51 frontend/src/utils/hooks/PropertyContext.js

7692c5a - JRB958, 6 weeks ago : create property using api successfully the company field must be existing company profile id need to add some extra fields in model

20 15 frontend/src/components/createProperty/CreateProperty.js

1604ba7 - JRB958, 6 weeks ago : implement api calls for adding a new property need to be tested when apis are available here

130 131 frontend/src/components/createProperty/CreateProperty.js

9f90f01 - MinhDuc1711, 6 weeks ago : Added testing for unit/parking/locker create form and navigation to these from profile

83 63 frontend/cypress/e2e/PropertyTesting.cy.js
3 3 frontend/src/components/property/PropertyPage.js

0b38eb5 - Yash Patel, 6 weeks ago : Merge branch 'main' into TASK-US3-ensure-secure-file-upload-for-the-users-picture-49

758ceba - Yash Patel, 6 weeks ago : Picture input only allows png and jpeg

1 0 frontend/src/components/userProfile/UserProfile.js

07bba57 - Yash Patel, 6 weeks ago : Removed modal, user profile already has pfp upload

0 30 frontend/src/components/UploadPFPModal.js
delete mode 100644 frontend/src/components/UploadPFPModal.js

b85a4fb - Yash Patel, 6 weeks ago : Remove duplicate "name"

0 2 backend/properties/migrations/0001_initial.py

5a15657 - Yash Patel, 6 weeks ago : Added file input type to modal

2 2 frontend/src/components/UploadPFPModal.js

b980a94 - Nicholas W, 6 weeks ago : Task us8 create unit/parking/locker 72 (#96)

add UI related to user story #24 and task #72 to be able to:

- create property
- create unit
- create parking
- create locker

3e780d9 - JRB958, 6 weeks ago : make migrations to pass the test on main

60 0 backend/properties/migrations/0001_initial.py
37 0 backend/properties/migrations/0002_initial.py
84 0 backend/user_profile/migrations/0001_initial.py
create mode 100644 backend/properties/migrations/0001_initial.py
create mode 100644 backend/properties/migrations/0002_initial.py
create mode 100644 backend/user_profile/migrations/0001_initial.py

20fdf1b - JRB958, 6 weeks ago : Merge remote-tracking branch 'origin' into
TASK-US8-Create-unit/parking/locker-72 removed conflict file

e8436c3 - Yash Patel, 6 weeks ago : Merge branch 'main' into
TASK-US3-ensure-secure-file-upload-for-the-users-picture-49

faf18fb - MinhDuc1711, 6 weeks ago : Initial Property Testing file (WIP)

```
79      0      frontend/cypress/e2e/PropertyTesting.cy.js
create mode 100644 frontend/cypress/e2e/PropertyTesting.cy.js
```

48b01ff - MinhDuc1711, 6 weeks ago : Merge branch 'TASK-US8-Create-unit/parking/locker-72'
of https://github.com/JRB958/THE-390 into TASK-US8-property_testing-75

6ba8984 - MinhDuc1711, 6 weeks ago : Removed redundant CondoProfile, minor code changes

```
0      123    frontend/src/components/createProperty/CondoProfile.js
1      3      frontend/src/components/createProperty/CreateLocker.js
1      3      frontend/src/components/createProperty/CreateParking.js
0      3      frontend/src/components/createProperty/CreateUnit.js
delete mode 100644 frontend/src/components/createProperty/CondoProfile.js
```

ba34426 - JRB958, 6 weeks ago : organize files in directory and add minor changes added a
cancel button to every (add) component correct small button UI

```
4      4      frontend/src/App.js
0      123    frontend/src/components/CondoProfile.js
123    0      frontend/src/components/createProperty/CondoProfile.js
16     10     frontend/src/components/{ => createProperty}/CreateLocker.js
16     10     frontend/src/components/{ => createProperty}/CreateParking.js
0      0      frontend/src/components/{property => createProperty}/CreateProperty.js
11     3      frontend/src/components/{ => createProperty}/CreateUnit.js
1      1      frontend/src/components/dashboard/DashBoard.js
delete mode 100644 frontend/src/components/CondoProfile.js
create mode 100644 frontend/src/components/createProperty/CondoProfile.js
rename frontend/src/components/{ => createProperty}/CreateLocker.js (81%)
rename frontend/src/components/{ => createProperty}/CreateParking.js (81%)
rename frontend/src/components/{property => createProperty}/CreateProperty.js (100%)
rename frontend/src/components/{ => createProperty}/CreateUnit.js (85%)
```

133b95c - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

UPDATE no branch relation

6 5 .github/workflows/systemTestingWF.yml

63f5a28 - Nicholas W, 6 weeks ago : Update systemTestingWF.yml

Adding Coverage Report - No branch association

3 1 .github/workflows/systemTestingWF.yml

d4f7b8a - MinhDuc1711, 6 weeks ago : Merge branch 'TASK-US8-Create-unit/parking/locker-72' of https://github.com/JRB958/THE-390 into TASK-US8-property_testing-75

a1a9770 - MinhDuc1711, 6 weeks ago : Added create property, added buttons to different pages, changed Link to useContext navigate

2	1	frontend/src/App.js
9	1	frontend/src/components/dashboard/DashBoard.js
179	33	frontend/src/components/property/CreateProperty.js
14	4	frontend/src/components/property/PropertyPage.js
0	5	frontend/src/utils/hooks/PropertyContext.js

ce83eed - JRB958, 6 weeks ago : removed merge conflicts to add the creation of property to the existing dashboard related to #72 #60

5cdfe6b - MinhDuc1711, 6 weeks ago : Created registration_key folder in backend

0	0	backend/registration_key/__init__.py
3	0	backend/registration_key/admin.py
6	0	backend/registration_key/apps.py
0	0	backend/registration_key/migrations/__init__.py
32	0	backend/registration_key/models.py
3	0	backend/registration_key/tests.py
3	0	backend/registration_key/views.py

create mode 100644 backend/registration_key/__init__.py
create mode 100644 backend/registration_key/admin.py
create mode 100644 backend/registration_key/apps.py
create mode 100644 backend/registration_key/migrations/__init__.py
create mode 100644 backend/registration_key/models.py
create mode 100644 backend/registration_key/tests.py
create mode 100644 backend/registration_key/views.py

ca8ab8d - JRB958, 6 weeks ago : small fix for hard code to make it work temp

3 2 frontend/src/App.js

f02cef8 - JRB958, 6 weeks ago : create property context add property context functions to GET all properties and by id still all hard coded, waiting on confirmation of endpoint some refractoring to code related to #71 #59 #60

1 2 frontend/src/App.js
65 44 frontend/src/components/property/PropertyPage.js
106 17 frontend/src/utils/hooks/PropertyContext.js

8af7707 - JRB958, 6 weeks ago : refractorg property card and navigation to property page

4 38 frontend/src/components/property/PropertyCard.js

df97137 - JRB958, 6 weeks ago : make small adjustments to layout and efficiency

1 15 frontend/src/components/dashboard/UserInfo.js
3 3 frontend/src/screens/HomeScreen.js

613bbbc4 - JRB958, 6 weeks ago : updated user information component in dashboard there was no need to make an api call again after using the context state for user profile in the react app

0 72 backend/properties/migrations/0001_initial.py
0 37 backend/properties/migrations/0002_initial.py
0 84 backend/user_profile/migrations/0001_initial.py
5 3 frontend/src/components/dashboard/UserInfo.js
delete mode 100644 backend/properties/migrations/0001_initial.py
delete mode 100644 backend/properties/migrations/0002_initial.py
delete mode 100644 backend/user_profile/migrations/0001_initial.py

cd1efed - JRB958, 6 weeks ago : Merge branch
'TASK-US5-api-calls-for-properties-from-the-REACT-60' into
TASK-US5-create_property_page-58

62b0919 - JRB958, 6 weeks ago : fix the property page routing to be reached from property card the routing was moved to property card the property page is hard coded now will implememnt api calls in a different branch

```
3    1    frontend/src/App.js
1    0    frontend/src/components/DashBoard.js
4    3    frontend/src/components/property/PropertyCard.js
25   6    frontend/src/components/property/PropertyPage.js
```

afb387b - Ghawi25, 6 weeks ago : final update for now for createProperty and propertyPage

```
1    1    frontend/src/App.js
10   0    frontend/src/components/property/CreateProperty.js
11   10   frontend/src/components/property/PropertyPage.js
1    1    frontend/src/screens/HomeScreen.js
```

e09e300 - MinhDuc1711, 6 weeks ago : Merge branch
'TASK-US8-Create-unit/parking/locker-72' of https://github.com/JRB958/THE-390 into
TASK-US8-Create-unit/parking/locker-72

039043c - MinhDuc1711, 6 weeks ago : Resolved conflicts for merging from main before PR

200fbea - MinhDuc1711, 6 weeks ago : minor variable name change

```
2    2    frontend/src/components/CreateUnit.js
```

ac62977 - JRB958, 6 weeks ago : Task us6 property endpoints 65 (#95)

Made django API endpoints for property profile CRUD operations

1. implemented serializer for each of the properties app models
2. implemented a view for each of the properties app models
3. made endpoints for crud operations

Related Issue: #65

e6e7c18 - JRB958, 6 weeks ago : resolve the conflicts from merging main

53c7237 - patrickmac3, 6 weeks ago : migrations

```
-      -      backend/db.sqlite3
60    0      backend/properties/migrations/0001_initial.py
37    0      backend/properties/migrations/0002_initial.py
84    0      backend/user_profile/migrations/0001_initial.py
create mode 100644 backend/properties/migrations/0001_initial.py
create mode 100644 backend/properties/migrations/0002_initial.py
create mode 100644 backend/user_profile/migrations/0001_initial.py
```

646b492 - patrickmac3, 6 weeks ago : undid removal of property context

24 24 frontend/src/utils/hooks/PropertyContext.js

64f808e - patrickmac3, 6 weeks ago : removed property context file content

24 24 frontend/src/utils/hooks/PropertyContext.js

e1cf4cd - patrickmac3, 6 weeks ago : fixed conflicts

```
0      60      backend/properties/migrations/0001_initial.py
0      37      backend/properties/migrations/0002_initial.py
0      84      backend/user_profile/migrations/0001_initial.py
delete mode 100644 backend/properties/migrations/0001_initial.py
delete mode 100644 backend/properties/migrations/0002_initial.py
delete mode 100644 backend/user_profile/migrations/0001_initial.py
```

7853bf2 - JRB958, 6 weeks ago : update profile context, and correct migration files

```
-      -      backend/db.sqlite3
1      1      backend/properties/migrations/0001_initial.py
1      1      backend/properties/migrations/0002_initial.py
1      1      backend/user_profile/migrations/0001_initial.py
0      1      frontend/src/utils/hooks/ProfileContext.js
```

3b5d941 - patrickmac3, 6 weeks ago : added tests

26 2 backend/jwt_auth_token/tests.py
92 6 backend/user_profile/tests.py

22447ef - JRB958, 7 weeks ago : correct one of the merge conflicts

0 2 frontend/src/utils/hooks/ProfileContext.js

88a3be7 - JRB958, 7 weeks ago : resolved the merge conflict after merging from main

d85e470 - JRB958, 7 weeks ago : minor changes to fix small issues removed some development code that was stopping not allowing compilation in propertyContext updated UserProfile useEffect

2 2 frontend/src/components/userProfile/UserProfile.js
15 15 frontend/src/utils/hooks/PropertyContext.js

2ef6bf9 - JRB958, 7 weeks ago : TASK US2 Conduct tests for the login functionality (#93)

test
- successful login scenario
- invalid email
- invalid password

Related Issue: #39

482cdea - JRB958, 7 weeks ago : removed conflicting files from the data base

e3da9eb - patrickmac3, 7 weeks ago : Task us5 api integration with react context 60 (#92)

Implemented the new REACT Context for "user profile" that uses updated API calls #60

c97e0f6 - patrickmac3, 7 weeks ago : cypress test checking

added test@example.com user into db for test

- - backend/db.sqlite3
13 19 backend/properties/migrations/0001_initial.py
37 0 backend/properties/migrations/0002_initial.py
4 7 backend/user_profile/migrations/0001_initial.py
create mode 100644 backend/properties/migrations/0002_initial.py

4510a9c - patrickmac3, 7 weeks ago : testing for login functionnality

test

- successful login scenario
- invalid email
- invalid password

```
-       -      backend/.coverage
-       -      backend/db.sqlite3
13     19      backend/properties/migrations/0001_initial.py
37     0       backend/properties/migrations/0002_initial.py
4      7       backend/user_profile/migrations/0001_initial.py
55     1       backend/user_profile/tests.py
create mode 100644 backend/properties/migrations/0002_initial.py
```

c6318df - Joud Babik, 7 weeks ago : updated the branch with the main branch before merging back to main

aeaf5f8 - patrickmac3, 7 weeks ago : db

```
-       -      backend/db.sqlite3
```

4dff046 - patrickmac3, 7 weeks ago : made endpoint for more efficiently creating property profile

include company id in endpoint

```
1      0      .gitignore
-       -      backend/db.sqlite3
6      1      backend/properties/admin.py
13    19      backend/properties/migrations/0001_initial.py
37    0       backend/properties/migrations/0002_initial.py
23    0       backend/properties/views.py
4      7       backend/user_profile/migrations/0001_initial.py
1      1       backend/user_profile/urls.py
create mode 100644 backend/properties/migrations/0002_initial.py
```

248c87c - Joud Babik, 7 weeks ago : add the updated api calls with the new endpoints to the context the user profile fields now make the api call from the updated context

6 40 frontend/src/components/UserProfile.js

```
3      0    frontend/src/index.js
37     14   frontend/src/utils/hooks/ProfileContext.js
15     15   frontend/src/utils/hooks/PropertyContext.js
```

11787ba - Joud Babik, 7 weeks ago : fix the state to have all the needed parts dquote> the state was missing the profile attributes so the PATCH call dquote> was not working properly #58 #60

```
6      0    frontend/src/components/log/SignUp.js
```

b8919e8 - patrickmac3, 7 weeks ago : US5 django endpoints for properties app

implemented endpoints for crud operations on the PropertyProfile, CondoUnit, ParkingUnit, StorageUnit

1. implemented serializers for each of the models
2. implemented views for each of the models, extra for the property-profile (main task of this issue)
3. created the associated endpoints

Related Issue: # 59

```
2      1    backend/.gitignore
2      1    backend/core/urls.py
-      -    backend/db.sqlite3
18     2    backend/properties/models.py
41     0    backend/properties/serializers.py
15     0    backend/properties/urls.py
67     1    backend/properties/views.py
3      0    backend/user_profile/models.py
10     2    backend/user_profile/urls.py
create mode 100644 backend/properties/serializers.py
create mode 100644 backend/properties/urls.py
```

69d16a1 - patrickmac3, 7 weeks ago : Task us6 property profile schema #64 (#89)

1. changed auth app to jwt_auth_token due to issues with the naming
2. implemented a Property Profile Schema
3. implemented basic models for condo, parking and storage units

issue #64

4b53b60 - patrickmac3, 7 weeks ago : Update db.sqlite3

- - backend/db.sqlite3

11824d6 - Nicholas W, 7 weeks ago : Test (#90)

Testing with new inputs to update Cypress Testing

Relates to no issue

(Just for testing purposes)

e87a967 - NicholasWahome, 7 weeks ago : Test

Testing with new inputs to update Cypress Testing

Relates to no issue

(Just for testing purposes)

3 3 frontend/cypress/e2e/AppTesting.cy.js

a72b33c - patrickmac3, 7 weeks ago : update sqllite

- - backend/db.sqlite3

795c6f5 - patrickmac3, 7 weeks ago : Update db.sqlite3

- - backend/db.sqlite3

aaa6d98 - patrickmac3, 7 weeks ago : sqllite

- - backend/db.sqlite3

2468ac8 - patrickmac3, 7 weeks ago : sqllite file

- - backend/db.sqlite3

2fd5cac - patrickmac3, 7 weeks ago : migrations for changes schema

- - backend/db.sqlite3

e0f0584 - patrickmac3, 7 weeks ago : created property profile schema

implemented property profile model
implemented basic models for condo, parking and storage units

```
1    0    backend/core/settings.py
0    0    backend/properties/__init__.py
3    0    backend/properties/admin.py
6    0    backend/properties/apps.py
66   0    backend/properties/migrations/0001_initial.py
0    0    backend/properties/migrations/__init__.py
54   0    backend/properties/models.py
3    0    backend/properties/tests.py
3    0    backend/properties/views.py
10   1    backend/user_profile/models.py
create mode 100644 backend/properties/__init__.py
create mode 100644 backend/properties/admin.py
create mode 100644 backend/properties/apps.py
create mode 100644 backend/properties/migrations/0001_initial.py
create mode 100644 backend/properties/migrations/__init__.py
create mode 100644 backend/properties/models.py
create mode 100644 backend/properties/tests.py
create mode 100644 backend/properties/views.py
```

d7c87dc - patrickmac3, 7 weeks ago : due to the name of the auth app, there were some problems, therefore i switched it to jwt_auth_token

```
2    1    backend/core/settings.py
1    1    backend/core/urls.py
0    0    backend/{auth => jwt_auth_token}/__init__.py
0    0    backend/{auth => jwt_auth_token}/admin.py
2    2    backend/{auth => jwt_auth_token}/apps.py
0    0    backend/{auth => jwt_auth_token}/migrations/__init__.py
0    0    backend/{auth => jwt_auth_token}/models.py
1    1    backend/{auth => jwt_auth_token}/serializers.py
0    0    backend/{auth => jwt_auth_token}/tests.py
0    0    backend/{auth => jwt_auth_token}/views.py
rename backend/{auth => jwt_auth_token}/__init__.py (100%)
rename backend/{auth => jwt_auth_token}/admin.py (100%)
rename backend/{auth => jwt_auth_token}/apps.py (58%)
rename backend/{auth => jwt_auth_token}/migrations/__init__.py (100%)
rename backend/{auth => jwt_auth_token}/models.py (100%)
rename backend/{auth => jwt_auth_token}/serializers.py (97%)
rename backend/{auth => jwt_auth_token}/tests.py (100%)
rename backend/{auth => jwt_auth_token}/views.py (100%)
```

3add000 - patrickmac3, 7 weeks ago : Task us5 create property card #57 (#88)

create property card to be shown in the dashboard.

created a pull from main before I made this request

e6a9057 - Joud Babik, 7 weeks ago : modify tests and header form to pass the required e2d tests

16 14 frontend/cypress/e2e/AppTesting.cy.js
2 2 frontend/src/components/Header.js

7dab7d6 - Joud Babik, 7 weeks ago : resolve conflicts from main -> this branch

1d41492 - Joud Babik, 7 weeks ago : finalize initial layout for owner's dashboard contains the properties, personal info, financial info, request info the information would be updated in later sprints.

6 21 frontend/src/components/dashboard/DashBoard.js
6 3 frontend/src/components/dashboard/Financial.js
27 0 frontend/src/components/dashboard/SubmittedRequests.js
18 52 frontend/src/components/dashboard/UserInfo.js
2 3 frontend/src/components/property/PropertyContainer.js
22 39 frontend/src/components/userProfile/UserProfile.js
7 1 frontend/src/index.js
5 2 frontend/src/utils/hooks/AuthContext.js
36 14 frontend/src/utils/hooks/ProfileContext.js
create mode 100644 frontend/src/components/dashboard/SubmittedRequests.js

06a08e1 - JRB958, 7 weeks ago : Updated Schema To support User Roles (#86)

User Story 4 - Issue # 54

Updated Schema to support Roles:

1. implemented a base user model
 - When a base user model instance is created and saved, an associated profile model is also saved depending on the user role
2. Implemented PublicProfile, EmployeeProfile, CompanyProfile
3. Updated Views
 - Made ViewSet for each new model for CRUD operations

4. Updated Serializers

- Made Serializer for each new model

@NicholasWahome

a959561 - patrickmac3, 7 weeks ago : Delete .github/workflows/djan.yml (#84)

Deleting this file as it not important for testing and contains no code

82ce8c6 - patrickmac3, 7 weeks ago : updated django test to follow changes. achieved 91% code coverage

```
-      -      backend/.coverage  
-      -      backend/db.sqlite3  
51    205    backend/user_profile/tests.py
```

fee351d - patrickmac3, 7 weeks ago : updated e2e testing in cypress for testing login/logout - modified login info to existing user in db

```
1      1      frontend/cypress/e2e/AppTesting.cy.js
```

31fc226 - patrickmac3, 7 weeks ago : updated endpoints used in frontend to match backend adjustements

```
5      4      frontend/src/components/UserProfile.js  
1      1      frontend/src/components/log/LogOut.js  
1      0      frontend/src/components/log/Login.js  
30    14      frontend/src/components/log/SignUp.js
```

c4a31d7 - patrickmac3, 7 weeks ago : created new app "auth" for cleaner code

```
0      0      backend/auth/__init__.py  
3      0      backend/auth/admin.py  
6      0      backend/auth/apps.py  
0      0      backend/auth/migrations/__init__.py  
3      0      backend/auth/models.py  
22    0      backend/auth/serializers.py  
3      0      backend/auth/tests.py  
26    0      backend/auth/views.py  
2      1      backend/core/settings.py  
2      1      backend/core/urls.py  
-      -      backend/db.sqlite3
```

```
4      1      backend/user_profile/models.py
20     3      backend/user_profile/serializers.py
21    24      backend/user_profile/views.py
create mode 100644 backend/auth/__init__.py
create mode 100644 backend/auth/admin.py
create mode 100644 backend/auth/apps.py
create mode 100644 backend/auth/migrations/__init__.py
create mode 100644 backend/auth/models.py
create mode 100644 backend/auth/serializers.py
create mode 100644 backend/auth/tests.py
create mode 100644 backend/auth/views.py
```

7091454 - MinhDuc1711, 7 weeks ago : Limit create unit/parking/locker to logged in users

```
11     4      frontend/src/components/Header.js
```

2064642 - Joud Babik, 7 weeks ago : add UserInfo and Finanical components to Dashboard layout and api calls still need to be done

```
-      -      backend/db.sqlite3
2      2      frontend/src/App.js
15     4      frontend/src/components/{ => dashboard}/DashBoard.js
24     0      frontend/src/components/dashboard/Financial.js
80     0      frontend/src/components/dashboard/UserInfo.js
1      0      frontend/src/components/property/PropertyContainer.js
3      3      frontend/src/components/{ => userProfile}/UserProfile.js
rename frontend/src/components/{ => dashboard}/DashBoard.js (55%)
create mode 100644 frontend/src/components/dashboard/Financial.js
create mode 100644 frontend/src/components/dashboard/UserInfo.js
rename frontend/src/components/{ => userProfile}/UserProfile.js (98%)
```

465ec08 - Ghawi25, 7 weeks ago : Property Page design Created Property page to be used/accesed from dashboard needs refactoring later

```
2      0      frontend/src/App.js
29     0      frontend/src/components/property/CreateProperty.js
1      0      frontend/src/components/property/PropertyCard.js
71     0      frontend/src/components/property/PropertyPage.js
3      3      frontend/src/screens/HomeScreen.js
create mode 100644 frontend/src/components/property/PropertyPage.js
```

8e75aed - Joud Babik, 7 weeks ago : update the property card changed the information to be smaller in #57 and will create property page this will be used in the dashboard #58

```
-      -      backend/db.sqlite3
2      0      frontend/src/components/DashBoard.js
44     8      frontend/src/components/property/PropertyCard.js
4      3      frontend/src/components/property/PropertyContainer.js
34     0      frontend/src/utils/hooks/ProfileContext.js
30     0      frontend/src/utils/hooks/PropertyContext.js
create mode 100644 frontend/src/utils/hooks/ProfileContext.js
create mode 100644 frontend/src/utils/hooks/PropertyContext.js
```

4328745 - patrickmac3, 7 weeks ago : updated logout endpoint in react frontend

endpoint used in react logout for blacklisting tokens is changed from user-profile/logout/blacklist to logout/blacklist

<https://github.com/JRB958/THE-390/pull/86#pullrequestreview-1902507615>

```
-      -      backend/db.sqlite3
1      1      frontend/src/components/log/LogOut.js
```

a5ff24c - patrickmac3, 7 weeks ago : Updated Schema To support User Roles

User Story 4 - Issue # 54

Updated Schema to support Roles:

- implemented a base user model
- implemented PublicProfile, EmployeeProfile and CompanyProfile
- Update Views and Serializers
 - Made ViewSet for each model which allows for CRUD operations
 - Made Serializer for each new model

```
-      -      backend/core/__pycache__/_init__.cpython-311.pyc
-      -      backend/core/__pycache__/_settings.cpython-311.pyc
-      -      backend/core/__pycache__/_urls.cpython-311.pyc
-      -      backend/core/__pycache__/_wsgi.cpython-311.pyc
2      2      backend/core/settings.py
4      3      backend/core/urls.py
-      -      backend/db.sqlite3
-      -      backend/user_profile/__pycache__/_init__.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_admin.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_apps.cpython-311.pyc
```

```
-     - backend/user_profile/__pycache__/models.cpython-311.pyc
-     - backend/user_profile/__pycache__/serializers.cpython-311.pyc
-     - backend/user_profile/__pycache__/tests.cpython-311.pyc
-     - backend/user_profile/__pycache__/urls.cpython-311.pyc
-     - backend/user_profile/__pycache__/views.cpython-311.pyc
6    10  backend/user_profile/admin.py
53   5   backend/user_profile/migrations/0001_initial.py
0    18  backend/user_profile/migrations/0002_alter_userprofile_registration_key.py
0    29
backend/user_profile/migrations/0003_userprofile_groups_userprofile_is_superuser_and_more.
py
0    18  backend/user_profile/migrations/0004_userprofile_phone_number.py
-     - backend/user_profile/migrations/__pycache__/0001_initial.cpython-311.pyc
-
backend/user_profile/migrations/__pycache__/0002_alter_userprofile_registration_key.cpython-
311.pyc
-
backend/user_profile/migrations/__pycache__/0003_userprofile_groups_userprofile_is_superuser_and_more.cpython-311.pyc
-
backend/user_profile/migrations/__pycache__/0004_userprofile_phone_number.cpython-311.py
c
-     - backend/user_profile/migrations/__pycache__/_init__.cpython-311.pyc
62   28  backend/user_profile/models.py
34   7   backend/user_profile/serializers.py
8    9   backend/user_profile/urls.py
32   61  backend/user_profile/views.py
delete mode 100644 backend/core/__pycache__/_init__.cpython-311.pyc
delete mode 100644 backend/core/__pycache__/settings.cpython-311.pyc
delete mode 100644 backend/core/__pycache__/urls.cpython-311.pyc
delete mode 100644 backend/core/__wsgi.cpython-311.pyc
delete mode 100644 backend/user_profile/__pycache__/_init__.cpython-311.pyc
delete mode 100644 backend/user_profile/__pycache__/_admin.cpython-311.pyc
delete mode 100644 backend/user_profile/__pycache__/apps.cpython-311.pyc
delete mode 100644 backend/user_profile/__pycache__/models.cpython-311.pyc
delete mode 100644 backend/user_profile/__pycache__/serializers.cpython-311.pyc
delete mode 100644 backend/user_profile/__pycache__/tests.cpython-311.pyc
delete mode 100644 backend/user_profile/__pycache__/urls.cpython-311.pyc
delete mode 100644 backend/user_profile/__pycache__/views.cpython-311.pyc
delete mode 100644
backend/user_profile/migrations/0002_alter_userprofile_registration_key.py
delete mode 100644
backend/user_profile/migrations/0003_userprofile_groups_userprofile_is_superuser_and_more.
py
```

```
delete mode 100644 backend/user_profile/migrations/0004_userprofile_phone_number.py
delete mode 100644
backend/user_profile/migrations/__pycache__/0001_initial.cpython-311.pyc
delete mode 100644
backend/user_profile/migrations/__pycache__/0002_alter_userprofile_registration_key.cpython-311.pyc
delete mode 100644
backend/user_profile/migrations/__pycache__/0003_userprofile_groups_userprofile_is_superuser_and_more.cpython-311.pyc
delete mode 100644
backend/user_profile/migrations/__pycache__/0004_userprofile_phone_number.cpython-311.py
c
delete mode 100644 backend/user_profile/migrations/__pycache__/_init__.cpython-311.pyc
```

6e1ff5d - Joud Babik, 7 weeks ago : create property card with cards container the container will be used in the dashboard created preliminary dashboard to test the container

```
19    13  frontend/package-lock.json
2     1   frontend/package.json
4     0   frontend/src/App.js
-     -   frontend/src/assets/condo-photo.jpg
-     -   frontend/src/assets/condo-photo1.png
-     -   frontend/src/assets/condo-photo2.png
2     2   frontend/src/bootstrap.min.css
0     12  frontend/src/bootstrap2.min.css
29    0   frontend/src/components/DashBoard.js
7     22  frontend/src/components/Header.js
1     3   frontend/src/components/UserProfile.js
3     2   frontend/src/components/log/Login.js
0     0   frontend/src/components/property/CreateProperty.js
98    0   frontend/src/components/property/PropertyCard.js
73    0   frontend/src/components/property/PropertyContainer.js
1     1   frontend/src/index.css
1     2   frontend/src/index.js
2     1   frontend/src/screens/HomeScreen.js
create mode 100644 frontend/src/assets/condo-photo.jpg
create mode 100644 frontend/src/assets/condo-photo1.png
create mode 100644 frontend/src/assets/condo-photo2.png
delete mode 100644 frontend/src/bootstrap2.min.css
create mode 100644 frontend/src/components/DashBoard.js
create mode 100644 frontend/src/components/property/CreateProperty.js
create mode 100644 frontend/src/components/property/PropertyCard.js
create mode 100644 frontend/src/components/property/PropertyContainer.js
```

0e80ed8 - MinhDuc1711, 7 weeks ago : Updated CreateLocker and CreateParking

```
3      3      frontend/src/components/CreateLocker.js
4      4      frontend/src/components/CreateParking.js
```

6744ca8 - MinhDuc1711, 7 weeks ago : Add files via upload

```
7      3      frontend/src/App.js
127    0      frontend/src/components/CreateLocker.js
127    0      frontend/src/components/CreateParking.js
36     37     frontend/src/components/CreateUnit.js
2      0      frontend/src/components/Header.js
create mode 100644 frontend/src/components/CreateLocker.js
create mode 100644 frontend/src/components/CreateParking.js
```

735bcae - Yash Patel, 7 weeks ago : Changed styling of modal

```
22     21     frontend/src/components/UploadPFPModal.js
```

d53dd71 - Yash Patel, 7 weeks ago : Created Front end for uploading profile picture

```
29     0      frontend/src/components/UploadPFPModal.js
create mode 100644 frontend/src/components/UploadPFPModal.js
```

5ae4cd5 - Nicholas W, 7 weeks ago : Delete .github/workflows/djan.yml

Deleting this file as it not important for testing and contains no code
0 1 .github/workflows/djan.yml
delete mode 100644 .github/workflows/djan.yml

5cc5daf - JRB958, 7 weeks ago : Update AppTesting.cy.js

try modifying test to match new layout
0 5 frontend/cypress/e2e/AppTesting.cy.js

d8ab246 - Joud Babik, 7 weeks ago : form routing to createunit components

```
4      0      frontend/src/App.js
```

```
127 0 frontend/src/components/CreateUnit.js
2 6 frontend/src/components/Header.js
create mode 100644 frontend/src/components/CreateUnit.js
```

b585be8 - MinhDuc1711, 7 weeks ago : Add files via upload

```
123 0 frontend/src/components/CondoProfile.js
create mode 100644 frontend/src/components/CondoProfile.js
```

985843a - JRB958, 7 weeks ago : implement authorized view for (#82)

logout and profile only if logged in

login only if logged out

related to #2 #5 #6

a8bf3e8 - Joud Babik, 8 weeks ago : implement authorized view for logout and profile only if logged in login only if logged out related to #2 #5 #6

```
2 0 .gitignore
2 1 backend/.gitignore
- - backend/db.sqlite3
23 12 frontend/src/components/Header.js
5 1 frontend/src/components/log/LogOut.js
10 9 frontend/src/components/log/Login.js
4 0 frontend/src/index.js
0 0 frontend/src/screens/SignScreen.js
23 0 frontend/src/utils/hooks/AuthContext.js
create mode 100644 .gitignore
delete mode 100644 frontend/src/screens/SignScreen.js
create mode 100644 frontend/src/utils/hooks/AuthContext.js
```

315404f - JRB958, 10 weeks ago : submit the deliverables for sprint 1

- - Sprint_1/Deliverable #1/Software Product Vision.pdf
- - Sprint_1/Deliverable #10/Testing Plan.pdf
- - Sprint_1/Deliverable #3/Diagrams/activity
diagram/property-profile-creation-activity-diagram.jpg
- - Sprint_1/Deliverable #3/Diagrams/activity diagram/sign-up-activity-diagram.jpg
- - Sprint_1/Deliverable #3/Diagrams/class diagram/Sprint1ClassDiagram.jpg
- - Sprint_1/Deliverable #3/Diagrams/component
diagram/soen390-componentDiagram.jpg
- - Sprint_1/Deliverable #3/Diagrams/domain model/domain-model-final.jpg

76 0 Sprint_1/Deliverable #3/Diagrams/use case
diagrams/Request-role-management-use-case.drawio
- - Sprint_1/Deliverable #3/Diagrams/use case
diagrams/create-profile-Login-use-case.jpg
- - Sprint_1/Deliverable #3/Diagrams/use case
diagrams/financial-system-operation-use-case.jpg
- - Sprint_1/Deliverable #3/Diagrams/use case
diagrams/property-management-use-case.jpg
106 0 Sprint_1/Deliverable #3/Diagrams/use case
diagrams/reservation-use-case.drawio
- - Sprint_1/Deliverable #3/SAD/SAD THE-390.pdf
- - Sprint_1/Deliverable #4/Risk Analysis Table.pdf
- - Sprint_1/Deliverable #4/Risk Management Plan.pdf
- - Sprint_1/Deliverable #7/Sprint 1 Retrospective.pdf
- - Sprint_1/Deliverable #8/Release plan for sprint 2.pdf
- - Sprint_1/Deliverable #9/UI Prototype.pdf
create mode 100644 Sprint_1/Deliverable #1/Software Product Vision.pdf
create mode 100644 Sprint_1/Deliverable #10/Testing Plan.pdf
create mode 100644 Sprint_1/Deliverable #3/Diagrams/activity
diagram/property-profile-creation-activity-diagram.jpg
create mode 100644 Sprint_1/Deliverable #3/Diagrams/activity
diagram/sign-up-activity-diagram.jpg
create mode 100644 Sprint_1/Deliverable #3/Diagrams/class diagram/Sprint1ClassDiagram.jpg
create mode 100644 Sprint_1/Deliverable #3/Diagrams/component
diagram/soen390-componentDiagram.jpg
create mode 100644 Sprint_1/Deliverable #3/Diagrams/domain model/domain-model-final.jpg
create mode 100644 Sprint_1/Deliverable #3/Diagrams/use case
diagrams/Request-role-management-use-case.drawio
create mode 100644 Sprint_1/Deliverable #3/Diagrams/use case
diagrams/create-profile-Login-use-case.jpg
create mode 100644 Sprint_1/Deliverable #3/Diagrams/use case
diagrams/financial-system-operation-use-case.jpg
create mode 100644 Sprint_1/Deliverable #3/Diagrams/use case
diagrams/property-management-use-case.jpg
create mode 100644 Sprint_1/Deliverable #3/Diagrams/use case
diagrams/reservation-use-case.drawio
create mode 100644 Sprint_1/Deliverable #3/SAD/SAD THE-390.pdf
create mode 100644 Sprint_1/Deliverable #4/Risk Analysis Table.pdf
create mode 100644 Sprint_1/Deliverable #4/Risk Management Plan.pdf
create mode 100644 Sprint_1/Deliverable #7/Sprint 1 Retrospective.pdf
create mode 100644 Sprint_1/Deliverable #8/Release plan for sprint 2.pdf
create mode 100644 Sprint_1/Deliverable #9/UI Prototype.pdf

d1d04f0 - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

1 1 .github/workflows/systemTestingWF.yml

33812a0 - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

26 15 .github/workflows/systemTestingWF.yml

9cd57f0 - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

4 4 .github/workflows/systemTestingWF.yml

2059687 - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

13 23 .github/workflows/systemTestingWF.yml

47256dd - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

8 11 .github/workflows/systemTestingWF.yml

b56f28a - NicholasWahome, 10 weeks ago : New commit for Testing

17 15 frontend/cypress/e2e/AppTesting.cy.js

2 2 frontend/src/components/UserProfile.js

f36893f - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

21 18 .github/workflows/systemTestingWF.yml

ab8a268 - NicholasWahome, 10 weeks ago : Merge branch 'main' of
<https://github.com/JRB958/THE-390>

7f606a4 - NicholasWahome, 10 weeks ago : Updates

-Tried to fix workflow

-Header.js & UserProfile.js

```
16    0    .github/workflows/systemTestingWF.yml  
2     2    frontend/src/components/Header.js  
1     1    frontend/src/components/UserProfile.js
```

490f222 - patrickmac3, 10 weeks ago : cypress test - fix bug

```
0     4    frontend/cypress/cypress.config.js  
1     1    frontend/cypress/support/e2e.js
```

7e6153e - NicholasWahome, 10 weeks ago : Cypress Tests

- Entering the site and navigating to the login page
- Entering the site and navigating to the profile page
- Filling out the sign-up form and submitting successfully
- Filling out the login form and submitting successfully
- Opening the profile editing modal and submitting successfully
- Logging out

```
4     0    frontend/cypress/cypress.config.js  
60    39   frontend/cypress/e2e/AppTesting.cy.js  
1     0    frontend/cypress/support/e2e.js  
2     0    frontend/src/App.js  
4     2    frontend/src/components/UserProfile.js  
1     0    frontend/src/components/log/LogOut.js  
3     0    frontend/src/components/log/Login.js  
14    4    frontend/src/components/log/SignUp.js
```

a6a0ea1 - patrickmac3, 10 weeks ago : Fixed doubled code error

when merging backend-signup with main, some pieces of code were doubled causing an error
when running the react frontend

```
1     2    frontend/src/components/log/Login.js  
0    118   frontend/src/components/log/SignUp.js
```

028953d - Nicholas W, 10 weeks ago : Backend signup (#21)

- backend Logic for Logging in and signing up

505f5ac - Nicholas W, 10 weeks ago : Merge branch 'main' into backend-signup

2c37a3a - Nicholas W, 10 weeks ago : Rename djan to djan.yml

```
0      0      .github/workflows/{djan => djan.yml}  
rename .github/workflows/{djan => djan.yml} (100%)
```

d28ab54 - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

```
1      0      .github/workflows/systemTestingWF.yml
```

443f449 - Nicholas W, 10 weeks ago : Update and rename django.yml to djan

```
1      0      .github/workflows/djan  
0      30     .github/workflows/django.yml  
create mode 100644 .github/workflows/djan  
delete mode 100644 .github/workflows/django.yml
```

63a27f2 - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

```
3      9      .github/workflows/systemTestingWF.yml
```

c4aafab - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

```
1      0      .github/workflows/systemTestingWF.yml
```

b610dd0 - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

```
14     2      .github/workflows/systemTestingWF.yml
```

a89aaf3 - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

```
0      3      .github/workflows/systemTestingWF.yml
```

febe117 - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

```
8      2      .github/workflows/systemTestingWF.yml
```

bf2cc45 - Nicholas W, 10 weeks ago : Update systemTestingWF.yml

5 1 .github/workflows/systemTestingWF.yml

a719205 - Nicholas W, 10 weeks ago : Rename systemTestingWF to systemTestingWF.yml

0 0 .github/workflows/{systemTestingWF => systemTestingWF.yml}
rename .github/workflows/{systemTestingWF => systemTestingWF.yml} (100%)

11f6fb6 - Nicholas W, 10 weeks ago : Create systemTestingWF

20 0 .github/workflows/systemTestingWF
create mode 100644 .github/workflows/systemTestingWF

abcaef0 - Nicholas W, 10 weeks ago : Create django.yml

30 0 .github/workflows/django.yml
create mode 100644 .github/workflows/django.yml

2195830 - Nicholas W, 10 weeks ago : Delete .github/workflows directory

0 29 .github/workflows/Cypress.yml
delete mode 100644 .github/workflows/Cypress.yml

b9583ab - Joud Babik, 10 weeks ago : add the google drive link for sprint 1 submission

3 0 README.md
create mode 100644 README.md

9c083e8 - patrickmac3, 10 weeks ago : Logout Button + Backend Tests

Frontend:

- added logout button
 - removes auth. token

backend:

- added comments to code
- added some tests

- - backend/.coverage
- - backend/db.sqlite3

```
-    - backend/user_profile/__pycache__/tests.cpython-311.pyc
-    - backend/user_profile/__pycache__/urls.cpython-311.pyc
-    - backend/user_profile/__pycache__/views.cpython-311.pyc
34   45  backend/user_profile/tests.py
2     2  backend/user_profile/urls.py
31   21  backend/user_profile/views.py
15   13  frontend/src/App.js
29   17  frontend/src/components/Header.js
6     1  frontend/src/components/UserProfile.js
24   0   frontend/src/components/log/LogOut.js
6     2  frontend/src/components/log/Login.js
8     5  frontend/src/components/log/SignUp.js
2     2  frontend/src/index.js
create mode 100644 frontend/src/components/log/LogOut.js
```

581919d - JRB958, 10 weeks ago : Update README.md

```
2     2  frontend/README.md
```

1e95d54 - JRB958, 10 weeks ago : Update README.md

```
8     6  frontend/README.md
```

5ada322 - Nicholas W, 10 weeks ago : Update Cypress.yml

Added npm start to yml start file

```
3     0  .github/workflows/Cypress.yml
```

be94b71 - Nicholas W, 10 weeks ago : Update Cypress.yml

```
5     2  .github/workflows/Cypress.yml
```

6aca8fd - patrickmac3, 10 weeks ago : System Tests Configuration (#15)

- Tests forms entries
- Tests link
- Doesn't test form submissions

Tech used: Cypress

to run cypress do:

```
-" npx cypress run "
```

Relates: #7, #5, #6

767cd46 - Nicholas W, 10 weeks ago : Update Cypress.yml

2 10 .github/workflows/Cypress.yml

2c1dc23 - Nicholas W, 10 weeks ago : Update Cypress.yml

1 1 .github/workflows/Cypress.yml

9539157 - Nicholas W, 10 weeks ago : Create Cypress.yml

31 0 .github/workflows/Cypress.yml

create mode 100644 .github/workflows/Cypress.yml

ed6afbf - NicholasWahome, 10 weeks ago : System Tests Configuration

- Tests forms entries
- Tests link
- * Doesn't test form submissions *

tech used: Cypress

to run cypress do:

-" npx cypress run "

Relates: #7, #5, #6

10	0	frontend/cypress.config.js
10	0	frontend/cypress/cypress.config.js
5	0	frontend/cypress/cypress/fixtures/example.json
25	0	frontend/cypress/cypress/support/commands.js
20	0	frontend/cypress/cypress/support/e2e.js
65	0	frontend/cypress/e2e/AppTesting.cy.js
5	0	frontend/cypress/fixtures/example.json
25	0	frontend/cypress/support/commands.js
20	0	frontend/cypress/support/e2e.js
5	0	frontend/cypress/support/index.js
1991	84	frontend/package-lock.json
5	0	frontend/package.json
4	4	frontend/src/components/log/Login.js
52	66	frontend/src/components/log/SignUp.js
create mode 100644 frontend/cypress.config.js		

```
create mode 100644 frontend/cypress/cypress.config.js
create mode 100644 frontend/cypress/cypress/fixtures/example.json
create mode 100644 frontend/cypress/cypress/support/commands.js
create mode 100644 frontend/cypress/cypress/support/e2e.js
create mode 100644 frontend/cypress/e2e/AppTesting.cy.js
create mode 100644 frontend/cypress/fixtures/example.json
create mode 100644 frontend/cypress/support/commands.js
create mode 100644 frontend/cypress/support/e2e.js
create mode 100644 frontend/cypress/support/index.js
```

35f2ed7 - patrickmac3, 10 weeks ago : Testing

Testing for views and models

```
-    -    backend/.coverage
-    -    backend/core/__pycache__/settings.cpython-311.pyc
-    -    backend/db.sqlite3
3    0    backend/requirements.txt
-    -    backend/user_profile/__pycache__/admin.cpython-311.pyc
-    -    backend/user_profile/__pycache__/models.cpython-311.pyc
-    -    backend/user_profile/__pycache__/serializers.cpython-311.pyc
-    -    backend/user_profile/__pycache__/tests.cpython-311.pyc
-    -    backend/user_profile/__pycache__/urls.cpython-311.pyc
-    -    backend/user_profile/__pycache__/views.cpython-311.pyc
2    0    backend/user_profile/models.py
231   1    backend/user_profile/tests.py
1     46   backend/user_profile/views.py
create mode 100644 backend/.coverage
create mode 100644 backend/user_profile/__pycache__/tests.cpython-311.pyc
```

dfd0c5d - patrickmac3, 10 weeks ago : Update db.sqlite3

```
-    -    backend/db.sqlite3
```

2fb667d - patrickmac3, 10 weeks ago : Connecting frontend and backend for signup, login and user profile

- create user from sign up form
- login using login form
- see all user profile details from user profile page

```
-    -    backend/core/__pycache__/settings.cpython-311.pyc
-    -    backend/core/__pycache__/urls.cpython-311.pyc
```

```
-      -      backend/db.sqlite3
-      -      backend/user_profile/__pycache__/_init__.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_admin.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_apps.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_models.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_serializers.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_urls.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_views.cpython-311.pyc
18  0      backend/user_profile/migrations/0004_userprofile_phone_number.py
-      -      backend/user_profile/migrations/__pycache__/_0001_initial.cpython-311.pyc
-
backend/user_profile/migrations/__pycache__/_0002_alter_userprofile_registration_key.cpython-311.pyc
-
backend/user_profile/migrations/__pycache__/_0003_userprofile_groups_userprofile_is_superuser_and_more.cpython-311.pyc
-
backend/user_profile/migrations/__pycache__/_0004_userprofile_phone_number.cpython-311.py
c
-      -      backend/user_profile/migrations/__pycache__/_init__.cpython-311.pyc
1  0      backend/user_profile/models.py
1  1      backend/user_profile/serializers.py
47  0      backend/user_profile/views.py
1  1      frontend/src/api/axios.js
153 63     frontend/src/components/UserProfile.js
52  30     frontend/src/components/log/Login.js
204 136    frontend/src/components/log/SignUp.js
create mode 100644 backend/user_profile/migrations/0004_userprofile_phone_number.py
create mode 100644
backend/user_profile/migrations/__pycache__/_0004_userprofile_phone_number.cpython-311.py
c
```

b50b038 - patrickmac3, 10 weeks ago : Merge branch 'signin-bootstrap' into backend-signup

0d3fe52 - patrickmac3, 10 weeks ago : Merge branch 'signin-bootstrap' of
<https://github.com/JRB958/THE-390> into signin-bootstrap

20c0656 - patrickmac3, 10 weeks ago : updated user profile model and created view for retrieving authenticated users details

```
-      -      backend/core/__pycache__/_settings.cpython-311.pyc
```

```

-   -   backend/core/__pycache__/urls.cpython-311.pyc
-   -   backend/db.sqlite3
-   -   backend/user_profile/__pycache__/_init__.cpython-311.pyc
-   -   backend/user_profile/__pycache__/_admin.cpython-311.pyc
-   -   backend/user_profile/__pycache__/_apps.cpython-311.pyc
-   -   backend/user_profile/__pycache__/_models.cpython-311.pyc
-   -   backend/user_profile/__pycache__/_serializers.cpython-311.pyc
-   -   backend/user_profile/__pycache__/_urls.cpython-311.pyc
-   -   backend/user_profile/__pycache__/_views.cpython-311.pyc
9   6   backend/user_profile/migrations/0001_initial.py
18  0   backend/user_profile/migrations/0002_alter_userprofile_registration_key.py
29  0
backend/user_profile/migrations/0003_userprofile_groups_userprofile_is_superuser_and_more.
py
-   -   backend/user_profile/migrations/__pycache__/_0001_initial.cpython-311.pyc
-   -
backend/user_profile/migrations/__pycache__/_0002_alter_userprofile_registration_key.cpython-
311.pyc
-
backend/user_profile/migrations/__pycache__/_0003_userprofile_groups_userprofile_is_superuser_
and_more.cpython-311.pyc
-   -   backend/user_profile/migrations/__pycache__/_init__.cpython-311.pyc
23  16   backend/user_profile/models.py
1    2    backend/user_profile/serializers.py
2    1    backend/user_profile/urls.py
46  8    backend/user_profile/views.py
create mode 100644
backend/user_profile/migrations/0002_alter_userprofile_registration_key.py
create mode 100644
backend/user_profile/migrations/0003_userprofile_groups_userprofile_is_superuser_and_more.
py
create mode 100644
backend/user_profile/migrations/__pycache__/_0002_alter_userprofile_registration_key.cpython-
311.pyc
create mode 100644
backend/user_profile/migrations/__pycache__/_0003_userprofile_groups_userprofile_is_superuser_
and_more.cpython-311.pyc

```

d5eeef9 - Joud Babik, 2 months ago : add user profile shows profile details can edit profile details for now no backend connection

```

2    1   frontend/src/App.js
-   -   frontend/src/assets/pp.jpg

```

```
6      7      frontend/src/components/Header.js
180    0      frontend/src/components/UserProfile.js
2      2      frontend/src/components/log/Login.js
13     12     frontend/src/components/log/SignUp.js
create mode 100644 frontend/src/assets/pp.jpg
create mode 100644 frontend/src/components/UserProfile.js
```

4db4023 - patrickmac3, 2 months ago : Merge branch 'backend-signup' into signin-bootstrap

789b245 - JRB958, 2 months ago : Signin bootstrap (#10)

React frontend for signing in and creating user

2efe39d - Joud Babik, 2 months ago : implement axios into sign in/up added some functionality to the two forms will merge and test with backend

```
29    0      frontend/package-lock.json
1      0      frontend/package.json
87    0      frontend/src/api/axios.js
58    24     frontend/src/components/log/Login.js
57    17     frontend/src/components/log/SignUp.js
create mode 100644 frontend/src/api/axios.js
```

041e552 - patrickmac3, 2 months ago : modified serializer for user profile

```
-      -      backend/core/__pycache__/settings.cpython-311.pyc
-      -      backend/core/__pycache__/urls.cpython-311.pyc
3      2      backend/core/urls.py
-      -      backend/db.sqlite3
-      -      backend/user_profile/__pycache__/_init__.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_admin.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_apps.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_models.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_serializers.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_urls.cpython-311.pyc
-      -      backend/user_profile/__pycache__/_views.cpython-311.pyc
7      7      backend/user_profile/admin.py
-      -      backend/user_profile/migrations/__pycache__/_0001_initial.cpython-311.pyc
-      -      backend/user_profile/migrations/__pycache__/_init__.cpython-311.pyc
1      1      backend/user_profile/serializers.py
5      7      backend/user_profile/views.py
```

2369f67 - Joud Babik, 2 months ago : replace styling with bootstrap

```
246 0 frontend/package-lock.json
3 0 frontend/package.json
0 38 frontend/src/App.css
21 18 frontend/src/App.js
0 8 frontend/src/App.test.js
12 0 frontend/src/bootstrap.min.css
12 0 frontend/src/bootstrap2.min.css
17 0 frontend/src/components/Content.js
18 0 frontend/src/components/Footer.js
39 0 frontend/src/components/Header.js
32 0 frontend/src/components/log/Login.js
100 0 frontend/src/components/log/SignUp.js
2 12 frontend/src/index.css
1 5 frontend/src/index.js
0 1 frontend/src/logo.svg
0 13 frontend/src/reportWebVitals.js
12 0 frontend/src/screens/HomeScreen.js
0 0 frontend/src/screens/SignScreen.js
0 5 frontend/src/setupTests.js
delete mode 100644 frontend/src/App.css
delete mode 100644 frontend/src/App.test.js
create mode 100644 frontend/src/bootstrap.min.css
create mode 100644 frontend/src/bootstrap2.min.css
create mode 100644 frontend/src/components/Content.js
create mode 100644 frontend/src/components/Footer.js
create mode 100644 frontend/src/components/Header.js
create mode 100644 frontend/src/components/log/Login.js
create mode 100644 frontend/src/components/log/SignUp.js
delete mode 100644 frontend/src/logo.svg
delete mode 100644 frontend/src/reportWebVitals.js
create mode 100644 frontend/src/screens/HomeScreen.js
create mode 100644 frontend/src/screens/SignScreen.js
delete mode 100644 frontend/src/setupTests.js
```

878e5eb - patrickmac3, 2 months ago : backend setup for user registr

created-endpoints for user registration and login

```
2 1 backend/.gitignore
- - backend/core/__pycache__/_init__.cpython-311.pyc
- - backend/core/__pycache__/settings.cpython-311.pyc
```

```
-    - backend/core/__pycache__/urls.cpython-311.pyc
-    - backend/core/__pycache__/wsgi.cpython-311.pyc
67    1 backend/core/settings.py
12   16 backend/core/urls.py
-    - backend/db.sqlite3
1     0 backend/requirements.txt
0     0 backend/user_profile/__init__.py
-    - backend/user_profile/__pycache__/_init__.cpython-311.pyc
-    - backend/user_profile/__pycache__/_admin.cpython-311.pyc
-    - backend/user_profile/__pycache__/_apps.cpython-311.pyc
-    - backend/user_profile/__pycache__/_models.cpython-311.pyc
-    - backend/user_profile/__pycache__/_serializers.cpython-311.pyc
-    - backend/user_profile/__pycache__/_urls.cpython-311.pyc
-    - backend/user_profile/__pycache__/_views.cpython-311.pyc
14    0 backend/user_profile/admin.py
6     0 backend/user_profile/apps.py
36    0 backend/user_profile/migrations/0001_initial.py
0     0 backend/user_profile/migrations/_init__.py
-    - backend/user_profile/migrations/__pycache__/_0001_initial.cpython-311.pyc
-    - backend/user_profile/migrations/__pycache__/_init__.cpython-311.pyc
50    0 backend/user_profile/models.py
18    0 backend/user_profile/serializers.py
3     0 backend/user_profile/tests.py
11    0 backend/user_profile/urls.py
37    0 backend/user_profile/views.py
create mode 100644 backend/user_profile/__init__.py
create mode 100644 backend/user_profile/__pycache__/_init__.cpython-311.pyc
create mode 100644 backend/user_profile/__pycache__/_admin.cpython-311.pyc
create mode 100644 backend/user_profile/__pycache__/_apps.cpython-311.pyc
create mode 100644 backend/user_profile/__pycache__/_models.cpython-311.pyc
create mode 100644 backend/user_profile/__pycache__/_serializers.cpython-311.pyc
create mode 100644 backend/user_profile/__pycache__/_urls.cpython-311.pyc
create mode 100644 backend/user_profile/__pycache__/_views.cpython-311.pyc
create mode 100644 backend/user_profile/admin.py
create mode 100644 backend/user_profile/apps.py
create mode 100644 backend/user_profile/migrations/0001_initial.py
create mode 100644 backend/user_profile/migrations/_init__.py
create mode 100644
backend/user_profile/migrations/__pycache__/_0001_initial.cpython-311.pyc
create mode 100644 backend/user_profile/migrations/_pycache__/_init__.cpython-311.pyc
create mode 100644 backend/user_profile/models.py
create mode 100644 backend/user_profile/serializers.py
create mode 100644 backend/user_profile/tests.py
create mode 100644 backend/user_profile/urls.py
```

```
create mode 100644 backend/user_profile/views.py
```

```
b4426ea - Joud Babik, 2 months ago : set up react app
```

```
23    0    frontend/.gitignore
108   0    frontend/README.md
18208 0    frontend/package-lock.json
38    0    frontend/package.json
-     -    frontend/public/favicon.ico
43    0    frontend/public/index.html
-     -    frontend/public/logo192.png
-     -    frontend/public/logo512.png
25    0    frontend/public/manifest.json
3     0    frontend/public/robots.txt
38   0    frontend/src/App.css
25   0    frontend/src/App.js
8    0    frontend/src/App.test.js
13   0    frontend/src/index.css
17   0    frontend/src/index.js
1    0    frontend/src/logo.svg
13   0    frontend/src/reportWebVitals.js
5    0    frontend/src/setupTests.js
create mode 100644 frontend/.gitignore
create mode 100644 frontend/README.md
create mode 100644 frontend/package-lock.json
create mode 100644 frontend/package.json
create mode 100644 frontend/public/favicon.ico
create mode 100644 frontend/public/index.html
create mode 100644 frontend/public/logo192.png
create mode 100644 frontend/public/logo512.png
create mode 100644 frontend/public/manifest.json
create mode 100644 frontend/public/robots.txt
create mode 100644 frontend/src/App.css
create mode 100644 frontend/src/App.js
create mode 100644 frontend/src/App.test.js
create mode 100644 frontend/src/index.css
create mode 100644 frontend/src/index.js
create mode 100644 frontend/src/logo.svg
create mode 100644 frontend/src/reportWebVitals.js
create mode 100644 frontend/src/setupTests.js
```

```
d78da71 - patrickmac3, 3 months ago : Update README.md
```

```
91    0    backend/README.md
```

b450036 - patrickmac3, 3 months ago : django rest framework setup

```
1    0    backend/.gitignore
0    0    backend/README.md
0    0    backend/core/__init__.py
-    -    backend/core/__pycache__/_init__.cpython-311.pyc
-    -    backend/core/__pycache__/_settings.cpython-311.pyc
-    -    backend/core/__pycache__/_urls.cpython-311.pyc
-    -    backend/core/__pycache__/_wsgi.cpython-311.pyc
16   0    backend/core/asgi.py
124  0    backend/core/settings.py
22   0    backend/core/urls.py
16   0    backend/core/wsgi.py
0    0    backend/db.sqlite3
22   0    backend/manage.py
8    0    backend/requirements.txt
create mode 100644 backend/.gitignore
create mode 100644 backend/README.md
create mode 100644 backend/core/__init__.py
create mode 100644 backend/core/__pycache__/_init__.cpython-311.pyc
create mode 100644 backend/core/__pycache__/_settings.cpython-311.pyc
create mode 100644 backend/core/__pycache__/_urls.cpython-311.pyc
create mode 100644 backend/core/__pycache__/_wsgi.cpython-311.pyc
create mode 100644 backend/core/asgi.py
create mode 100644 backend/core/settings.py
create mode 100644 backend/core/urls.py
create mode 100644 backend/core/wsgi.py
create mode 100644 backend/db.sqlite3
create mode 100644 backend/manage.py
create mode 100644 backend/requirements.txt
```

End of Document