Compiled Report

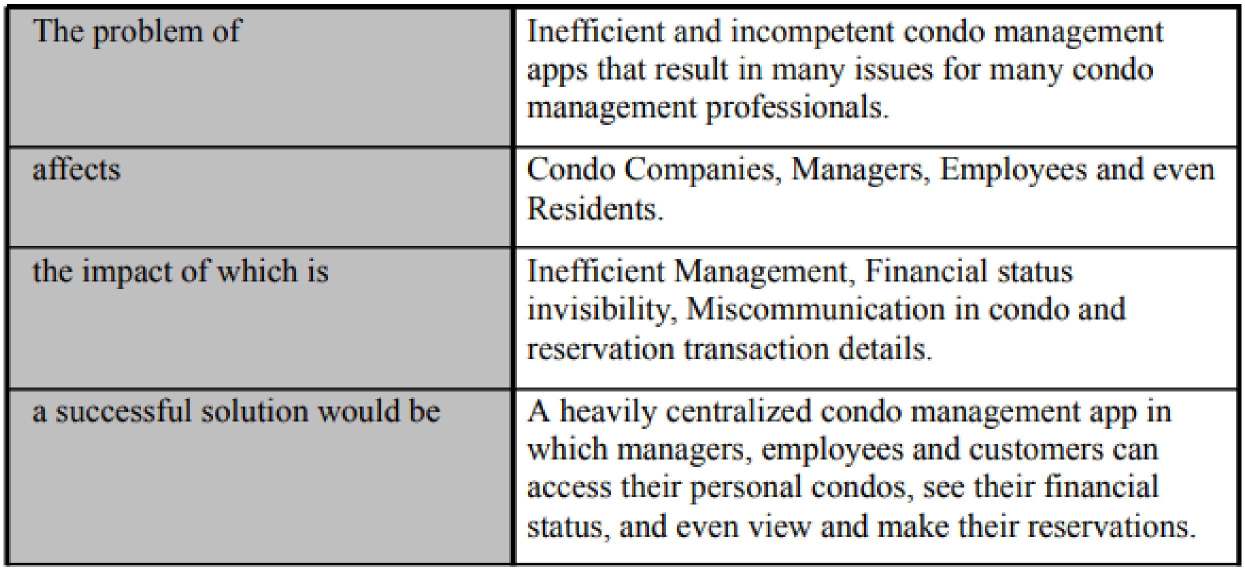
Product Vision

Introduction

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

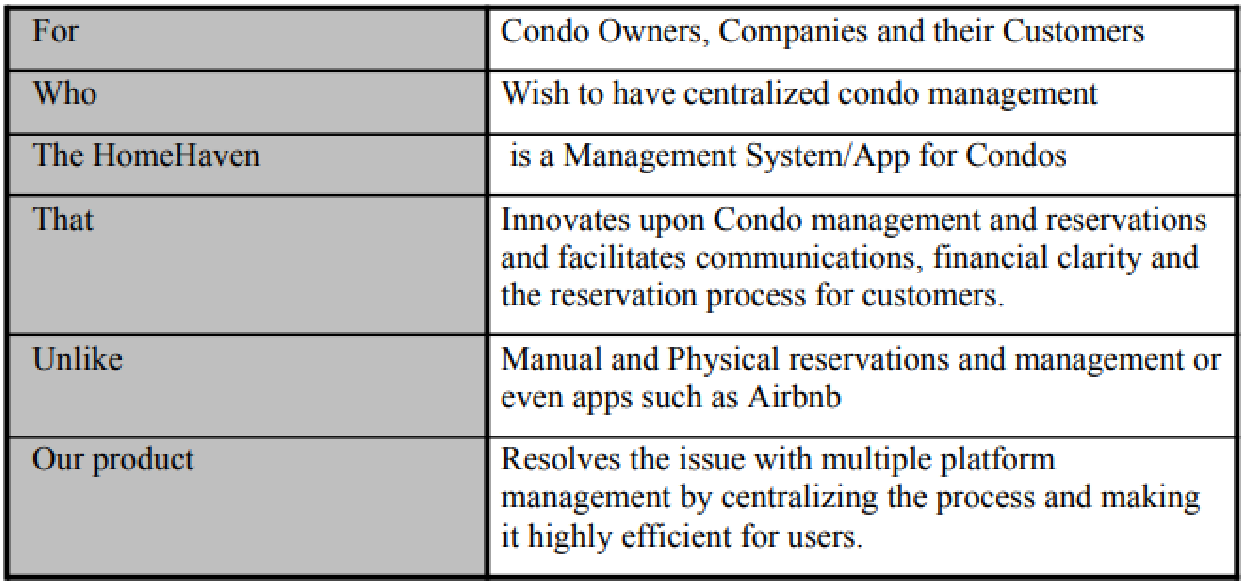
Positioning

Problem Statement



The problem of	Inefficient and incompetent condo management apps that result in many issues for many condo management professionals.
affects	Condo Companies, Managers, Employees and even Residents.
the impact of which is	Inefficient Management, Financial status invisibility, Miscommunication in condo and reservation transaction details.
a successful solution would be	A heavily centralized condo management app in which managers, employees and customers can access their personal condos, see their financial status, and even view and make their reservations.

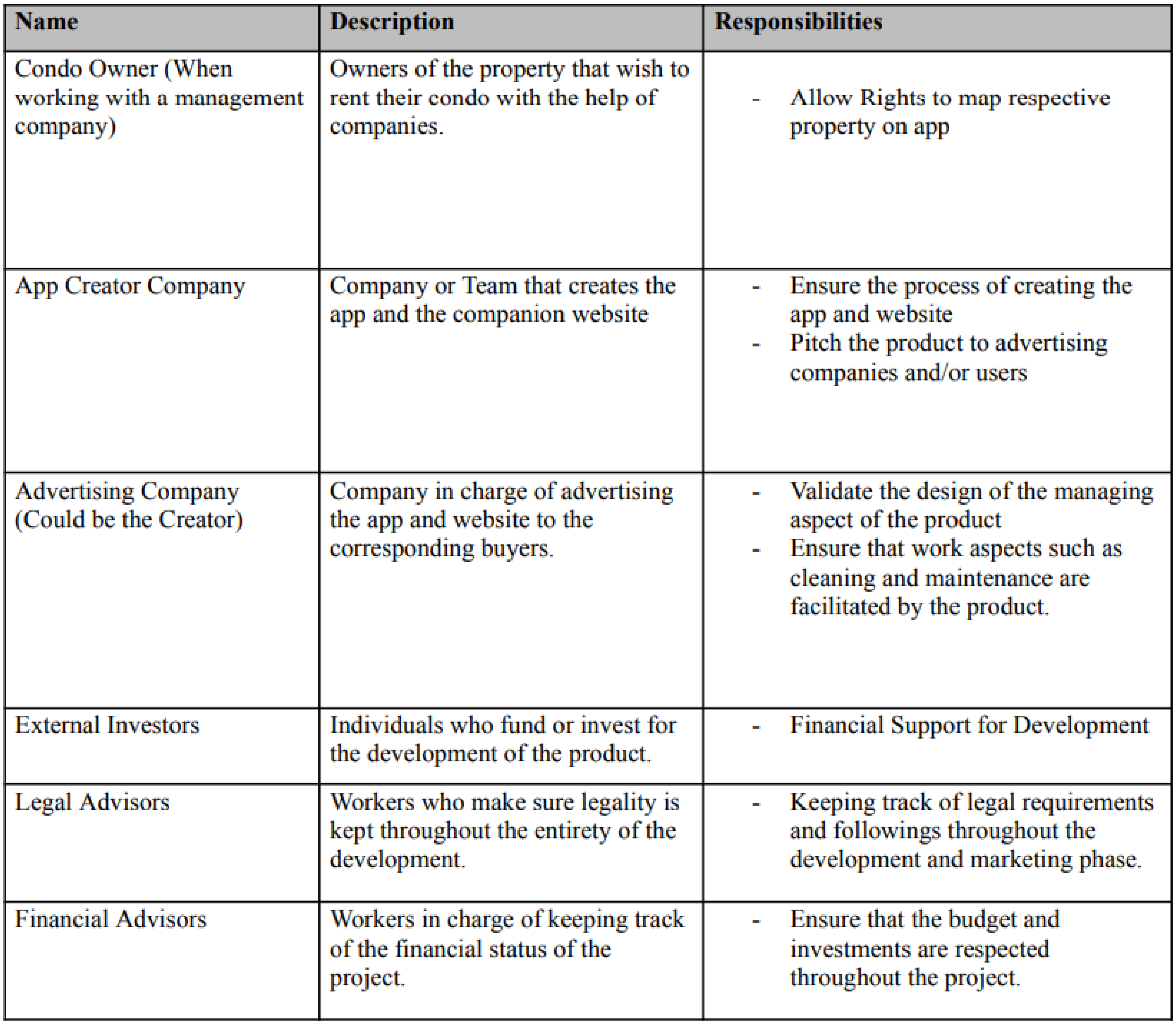
Product Position Statement



For	Condo Owners, Companies and their Customers
Who	Wish to have centralized condo management
The HomeHaven	is a Management System/App for Condos
That	Innovates upon Condo management and reservations and facilitates communications, financial clarity and the reservation process for customers.
Unlike	Manual and Physical reservations and management or even apps such as Airbnb
Our product	Resolves the issue with multiple platform management by centralizing the process and making it highly efficient for users.

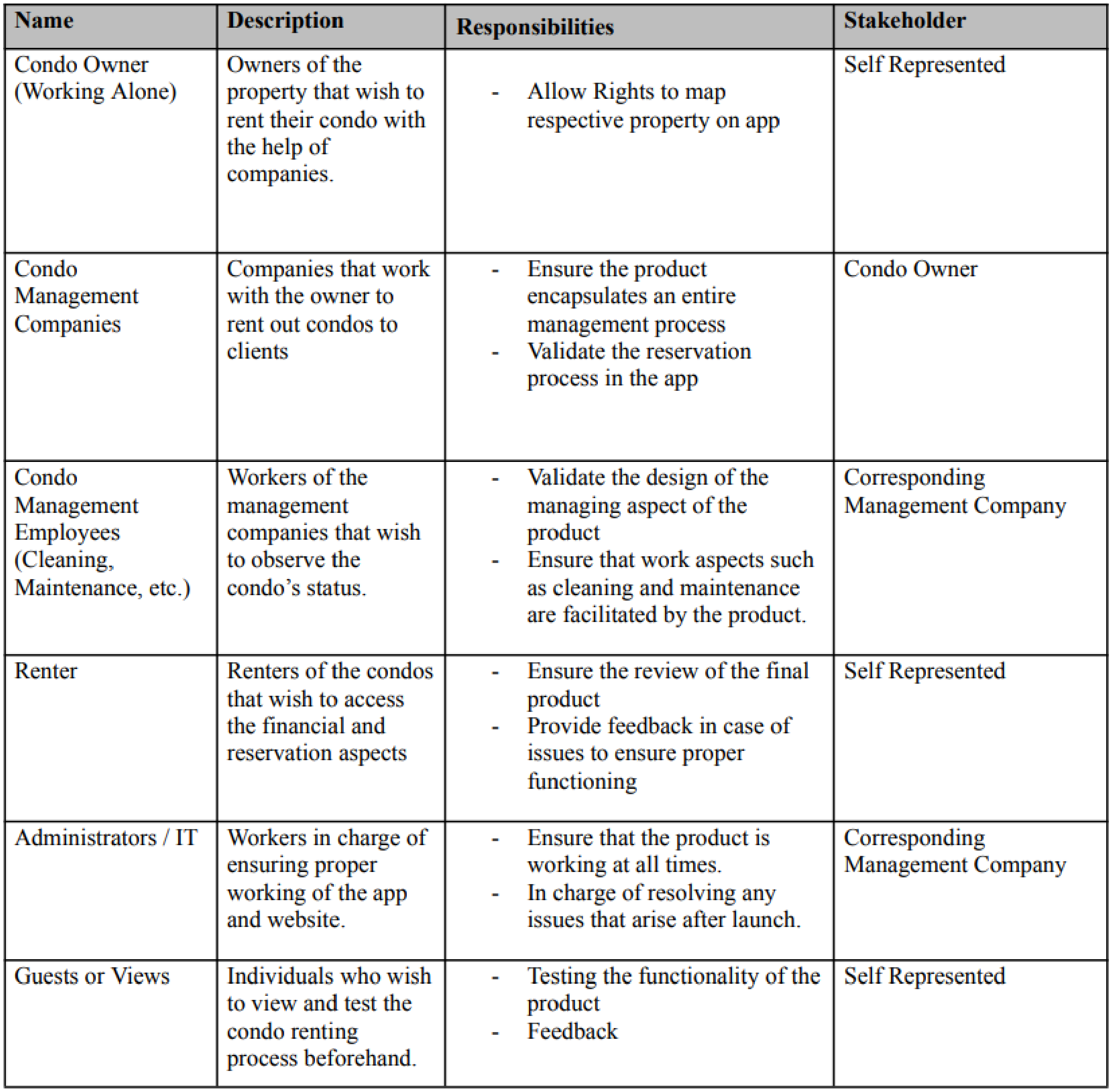
Stakeholder and User Descriptions

Stakeholder Summary



Name	Description	Responsibilities
Condo Owner (When working with a management company)	Owners of the property that wish to rent their condo with the help of companies.	- Allow Rights to map respective property on app
App Creator Company	Company or Team that creates the app and the companion website	- Ensure the process of creating the app and website - Pitch the product to advertising companies and/or users
Advertising Company (Could be the Creator)	Company in charge of advertising the app and website to the corresponding buyers.	- Validate the design of the managing aspect of the product - Ensure that work aspects such as cleaning and maintenance are facilitated by the product.
External Investors	Individuals who fund or invest for the development of the product.	- Financial Support for Development
Legal Advisors	Workers who make sure legality is kept throughout the entirety of the development.	- Keeping track of legal requirements and followings throughout the development and marketing phase.
Financial Advisors	Workers in charge of keeping track of the financial status of the project.	- Ensure that the budget and investments are respected throughout the project.

User Summary



Name	Description	Responsibilities	Stakeholder
Condo Owner (Working Alone)	Owners of the property that wish to rent their condo with the help of companies.	- Allow Rights to map respective property on app	Self Represented
Condo Management Companies	Companies that work with the owner to rent out condos to clients	- Ensure the product encapsulates an entire management process - Validate the reservation process in the app	Condo Owner
Condo Management Employees (Cleaning, Maintenance, etc.)	Workers of the management companies that wish to observe the condo's status.	- Validate the design of the managing aspect of the product - Ensure that work aspects such as cleaning and maintenance are facilitated by the product.	Corresponding Management Company
Renter	Renters of the condos that wish to access the financial and reservation aspects	- Ensure the review of the final product - Provide feedback in case of issues to ensure proper functioning	Self Represented
Administrators / IT	Workers in charge of ensuring proper working of the app and website.	- Ensure that the product is working at all times. - In charge of resolving any issues that arise after launch.	Corresponding Management Company
Guests or Views	Individuals who wish to view and test the condo renting process beforehand.	- Testing the functionality of the product - Feedback	Self Represented

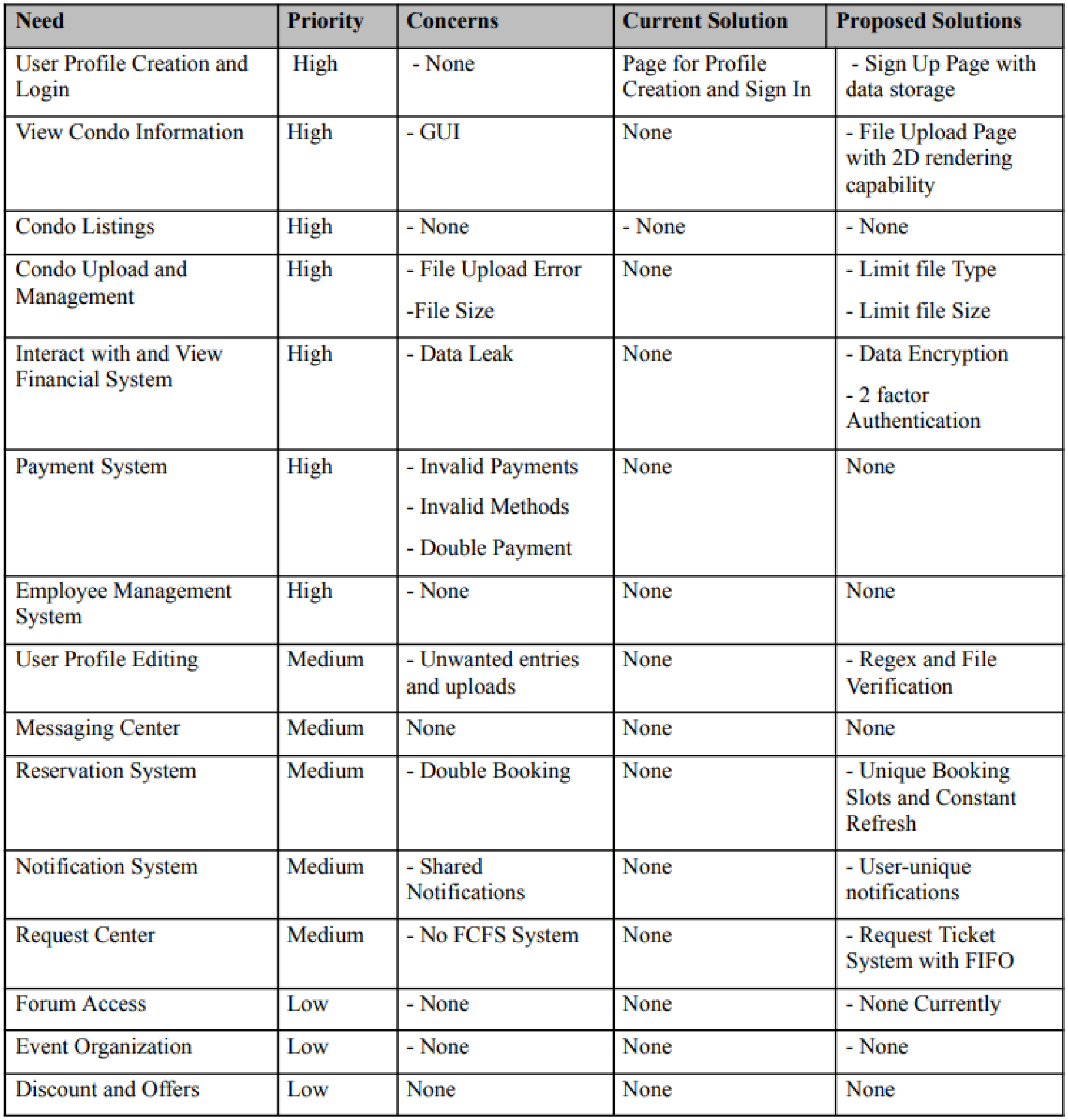
User Environment

- Number of People Involved: The number of people varies depending on the user. For the management company and its employees, it varies on the number of teams involved and the size of the company itself. For the renters, it is the amount of individuals that rent out the condos managed by each company.
- Task Cycles: For the management companies, each cycle lasts as long as they work with the condo owner. Activities are mostly done on a daily to weekly basis. This includes maintenance, cleaning and the like. These cycles are prone to change as they depend on the condo owner and company's wishes. For the renters, each cycle lasts as long as they

rent a condo. Their activities are done daily in most cases. For rent, it’s usually done monthly.

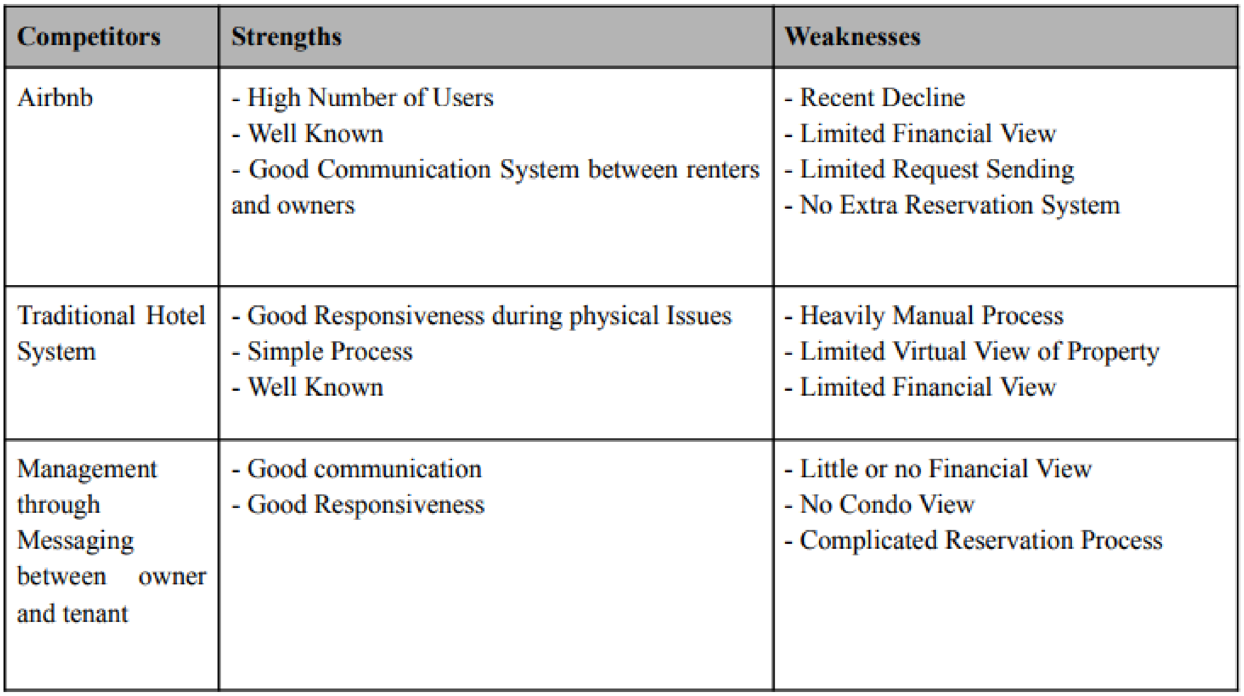
- Environmental Constraints: The users would require a device capable of running the app or website with internet access.
- Platforms: For the app, modern mobile platforms such as iOS and Android are supported. For the websites, Windows, Linux and MacOS are supported with the use of any browser.
- Other Applications: HomeHaven is able to interact with external apps such as the App Store and Play Store for the download process. Gmail access is also available for profile creation, login as well as an external notification link.

Key Stakeholder or User Needs



Need	Priority	Concerns	Current Solution	Proposed Solutions
User Profile Creation and Login	High	- None	Page for Profile Creation and Sign In	- Sign Up Page with data storage
View Condo Information	High	- GUI	None	- File Upload Page with 2D rendering capability
Condo Listings	High	- None	- None	- None
Condo Upload and Management	High	- File Upload Error -File Size	None	- Limit file Type - Limit file Size
Interact with and View Financial System	High	- Data Leak	None	- Data Encryption - 2 factor Authentication
Payment System	High	- Invalid Payments - Invalid Methods - Double Payment	None	None
Employee Management System	High	- None	None	None
User Profile Editing	Medium	- Unwanted entries and uploads	None	- Regex and File Verification
Messaging Center	Medium	None	None	None
Reservation System	Medium	- Double Booking	None	- Unique Booking Slots and Constant Refresh
Notification System	Medium	- Shared Notifications	None	- User-unique notifications
Request Center	Medium	- No FCFS System	None	- Request Ticket System with FIFO
Forum Access	Low	- None	None	- None Currently
Event Organization	Low	- None	None	- None
Discount and Offers	Low	None	None	None

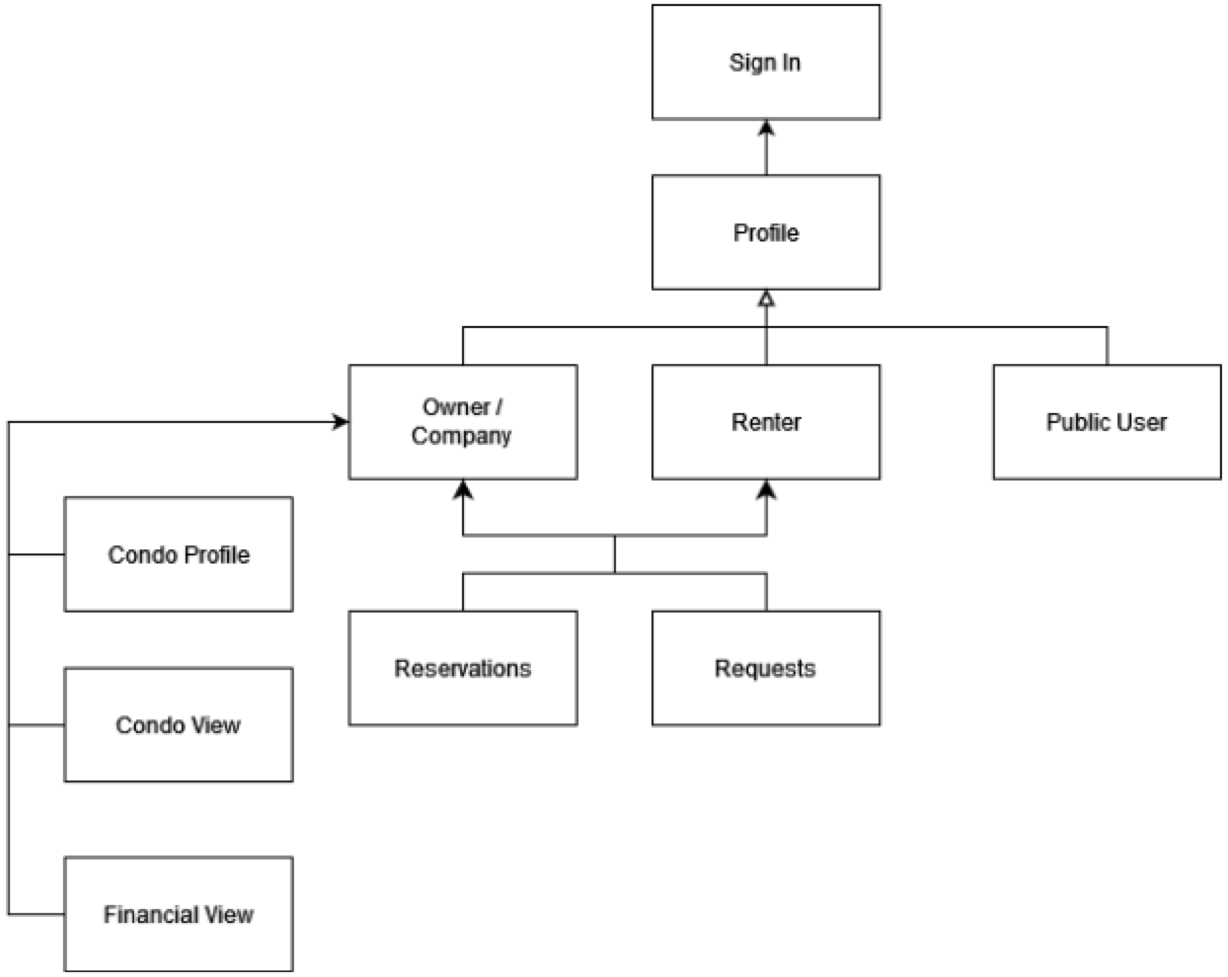
Alternatives and Competition



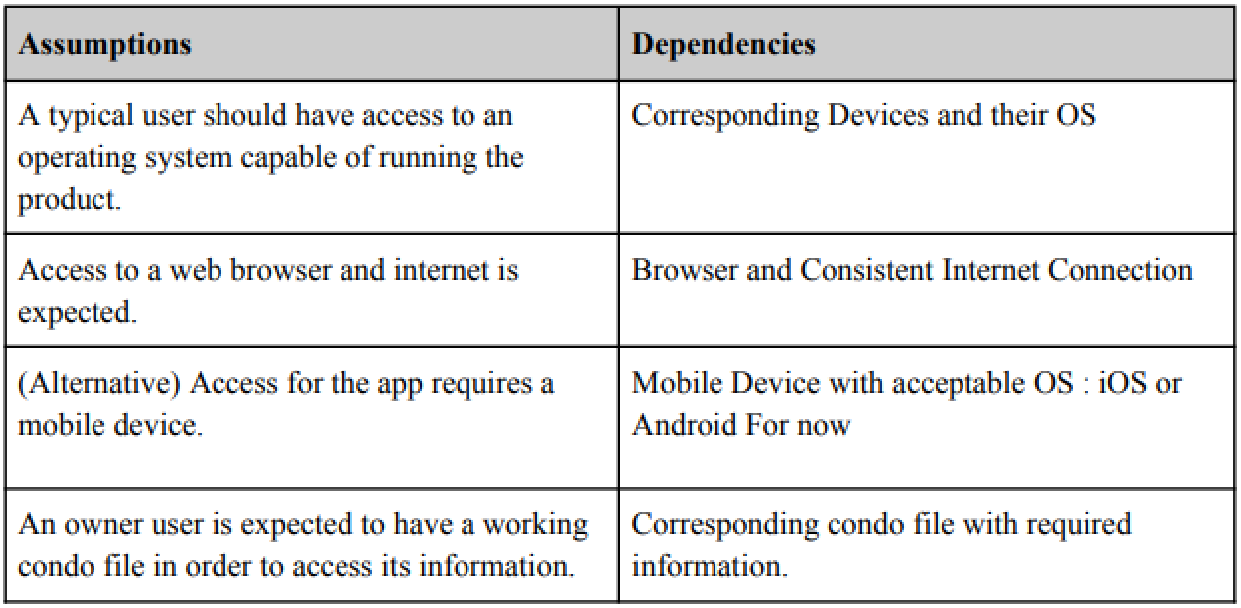
Competitors	Strengths	Weaknesses
Airbnb	- High Number of Users - Well Known - Good Communication System between renters and owners	- Recent Decline - Limited Financial View - Limited Request Sending - No Extra Reservation System
Traditional Hotel System	- Good Responsiveness during physical Issues - Simple Process - Well Known	- Heavily Manual Process - Limited Virtual View of Property - Limited Financial View
Management through Messaging between owner and tenant	- Good communication - Good Responsiveness	- Little or no Financial View - No Condo View - Complicated Reservation Process

Product Overview

Product Perspective



Assumptions and Dependencies



Assumptions	Dependencies
A typical user should have access to an operating system capable of running the product.	Corresponding Devices and their OS
Access to a web browser and internet is expected.	Browser and Consistent Internet Connection
(Alternative) Access for the app requires a mobile device.	Mobile Device with acceptable OS : iOS or Android For now
An owner user is expected to have a working condo file in order to access its information.	Corresponding condo file with required information.

Product Features

Base Features

Login
Any public user can sign up and log in to the website or app. This is ensured as long as their corresponding sign-up is valid.

Custom Profile Access
Users can edit their profile with a profile picture, username, contact email and phone number.

Dashboard View
Users registered as Condo owners in the system can have a rendered view of their properties. This includes a general view as well as information such as financial status, payments, requests, etc.

Property Profiles
Companies and Owners can create profiles for their properties. This enables access to add and edit information such as name, parking, locker, address, tenants, registration keys, etc.

Financial System with Simplified View
Companies and Owners have access to a generalized view of the financial status of their properties. This includes information like fee per square foot, parking spot fees, operational budget information as well as the ability to generate an annual report for properties.

Reservation System
A reservation system in which owners and companies can create listings for common areas and facilities for reservation while the renters can access these reservations

Request System
A ticket-based request system through which a tenant can request aid, maintenance or change. The owning user can view requests and act upon them.

Additional Features

Condo Community Posting
Condo renters can communicate and make posts about their condos and lives for others to see and interact with.

Community Events Tab and Calendar
Condo owners and residents can see current events and have a calendar to view events that are organized each day.

Suggestion Box
Condo residents can input suggestions and recommendations for the owners to see. These messages can be done anonymously.

Emergency Alerts
Condo owners can send emergency alerts in case of fires, safety hazards, etc. This emergency alert will notify all tenants through their app notifications.

Parking System
Tenants can register their vehicles in specified parking spots. This will use an FCFS system in which a parking spot owner can remove and change as needed.

Other Product Requirements

Direct Messaging
Users will be able to message each other. This is especially useful between owners and tenants.

Notification System
Users will have access to a notification system through which they can view their reservation status or messages.

Languages
The application will be available in English. Other languages may be added in the future.

Accessibility
Features like dark mode and the zoom lens throughout the pages will be accessible.

Linked Sign Up
Users can sign up directly with their Gmail account or manually using other emails.

User Stories & Requirements

User Stories

The user stories completed in each sprint are shown in this section.

Sprint 2

Public User Stories

1	Create Public Profile		
As a public user, I want to create a unique public profile clearly and familiarly.			
User Profile Feature			
Must	9	Low Risk	5

2	Public User Login		
As a public user, I want to log in to the app. The app will authenticate my login information and proceed to log in if the information is correct.			
Login Feature			
Must	9	Low Risk	4

3	Update Public Profile Details		
As a public user, I want to be able to update my profile details. This includes a profile picture, user name, contact email, and phone number.			
User Profile Feature			
Should	6	Medium Risk	5

Sprint 3

Public User Stories

4	Register as a Condo Owner		
As a public user, I want to become a condo owner by providing a registration key given by the condo management company			
Registration Feature			
Should	7	Low Risk	3

5	Register as a Rental User		
As a public user, I want to become a user rental by providing a registration key given by the condo management company			
Registration Feature			
Should	7	Low Risk	3

Condo Owner User Stories

6	Condo Owner Login		
As a condo owner, I want to log in to view all the reserved information.			
Login Feature			
Must	9	Low Risk	3

7	View Properties Dashboard		
As a condo owner, I want to be able to view my properties with each of their general information, balance owed, and payment due date on the main page/screen so that I can quickly navigate between multiple properties without having to click through pages.			
Dashboard Feature			
Should	7	Medium Risk	6

Condo Management Company Manager Stories

13	Add Files to the Property Account

As a condo manager, I want to upload PDF files to the property account so that the condo owners and rental users can see them.			
Property Profile Feature			
Must	8	Low Risk	4

14	Modify Unit Information		
As a condo manager, I want to modify the information associated with a unit of the property so that changes to unit id, size, unit owner, unit occupant, and condo fees can accurately reflect new changes to the unit.			
Property Profile Feature			
Must	8	Low Risk	3

15	Modify Parking Spot Information		
As a condo manager, I want to modify the information associated with a parking spot on the property so that changes to parking spot id, spot owner, spot occupant, and spot fees can accurately reflect new changes to the parking spot.			
Property Profile Feature			
Must	8	Low Risk	3

16	Modify Locker Information		
As a condo manager, I want to modify the information associated with a locker on the property so that changes to locker id, locker owner, locker occupant, and locker fees can accurately reflect new changes to the parking spot.			
Property Profile Feature			
Must	8	Low Risk	3

19	Modify Financial Fees		
As a condo manager, I want to be able to modify the square footage fee, the locker fee, and the parking spot fee so that all units' fees are updated at once.			
Financial Record Feature			
Must	7	Low Risk	4

20	Calculate Financial Budget		
As a condo manager, I want to see the property's operational budget (total fees from occupied units, parking spots, and lockers) so that I can control costs and still generate a profit.			
Financial Record Feature			
Must	6	Low Risk	4

21	Generate Annual Report		
As a condo manager, I want to be able to retrieve an annual report of the financial record of a property for a given year so that I can have an overview of the property's finances year-to-year.			
Financial Record Feature			
Must	7	Low Risk	4

Sprint 4

Condo Owner User Stories

8	Add Property from Dashboard		
As a condo owner, I want to add new properties from my dashboard so that, once I have a new property's registration key, I can quickly add the new property without navigating through the app/website.			
Dashboard Feature			
Should	Low Value	Low Risk	6

10	Create Request		
As a property user, I want to be able to make requests to the property's management so that I can have particular issues and/or requests fulfilled by the property's employees.			
Requests Feature			
Must	8	Low Risk	5

Condo Management Company Manager Stories

11	Create Property Account		
As a condo manager, I want to be able to create a profile for a new property so that I can track the property's units, parking spots, lockers, facility reservations and their current occupants.			
Registration Feature			
Must	9	Low Risk	5

12	Property Account Login		
As a condo manager, I want to log into the property account to access and modify all the reserved information.			
Login Feature			
Must	9	Low Risk	3

17	Create Owner Registration Key		
As a condo manager, I want to be able to create an owner registration key for a unit so that a new unit owner may access their unit information and update the unit's information.			
Registration Key Feature			
Must	8	Low Risk	5

18	Create Rental Registration Key		
As a condo manager, I want to be able to create a rental registration key for a unit so that a new rental user may access their unit information and update the unit's information.			
Registration Key Feature			
Must	8	Low Risk	5

24	Assign Requests		
As a condo manager, I want requests submitted by client users to be automatically sent to the appropriate employee(s) depending on their stated "request type" so that no one on the management profile needs to assign them one by one.			
Requests Feature			
Should	7	Low Risk	4

Requirements

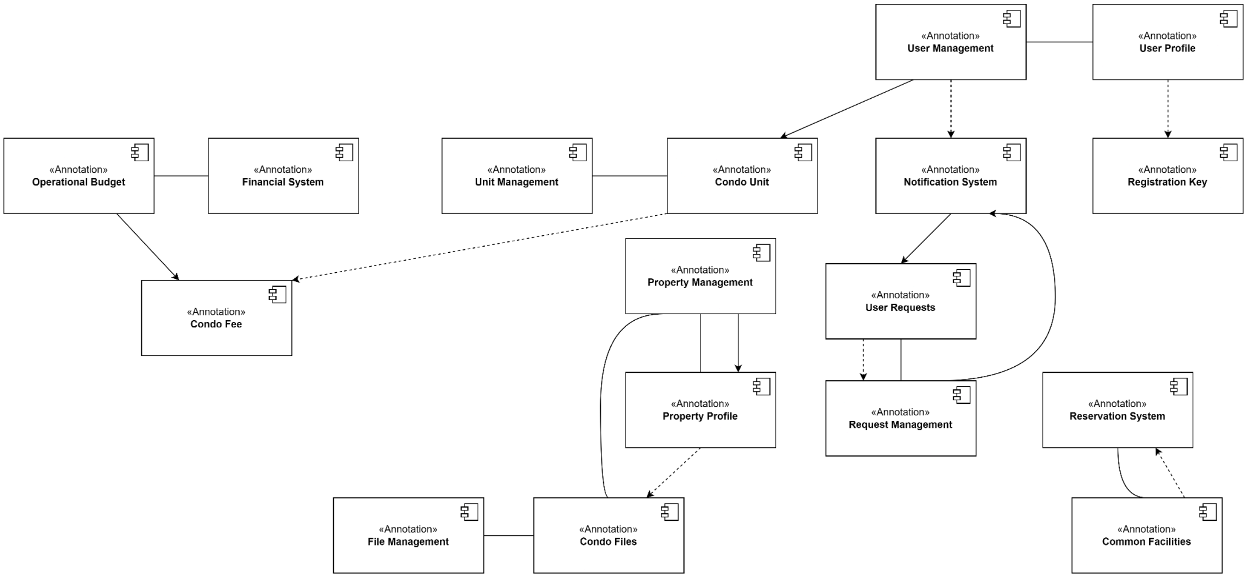
- Login Page
  - Description: The landing page for any user is the login page. On this page, the user can enter their credentials to log in or navigate to the signup page if they do not have an account.
  - Associated User Stories: 2, 6, 12, 26
- Sign Up Page
  - Description: For new condo renters or owners, first create a public user account through the signup page. On this page, they must fill out their credentials (including but not limited to email, username, password, name, last name, phone number).
  - Associated User Stories: 1,
- Dashboard
  - Description: Once logged in, users will access their dashboard. Their dashboard should show their properties' information, the current balance and the due date for each property, each property's building's facilities, and the reservation interface for the facilities.
  - Associated User Stories: 7, 8, 9,
- Profile Page
  - Description: The user should be able to access their profile page. On this page, they can modify their information (including but not limited to email, phone number, username, password, and profile picture). It is also on this page that a user will enter a registration key to become a condo owner or rental user.
  - Associated User Stories: 3, 4, 5
- Request Page
  - Description: The user should be able to access a request page apart from the dashboard. On the request page, a user may submit a request to the management company associated with the user's selected property. Each request should include the type of request (which will determine the employee that receives it) and a description. The type of request should be selected by the user from a drop-down menu with predetermined choices and the description should be a text box.
  - Associated User Stories: 10, 24
- Company Profile Page
  - Description: If the user logs in with a company manager account, they should have access to the company profile page. On the company profile page, they can see all property accounts under the company along with a summary of relevant information about the property (including but not limited to address, number of units, parking spots, lockers, facilities, number of owners and rental users). The

    company manager user can create a new property account which would include its login information on this page. Upon creating a property, the company manager user needs to enter the relevant information for the property account (including but not limited to address, number of units, parking spots, lockers, and facilities).
  - Associated User Stories: 11,
- Property Profile Page
  - Description: If the user logs in with a property account, they should reach the property profile page. On this page, multiple tabs are accessible: a units tab, a parking spots tab, a lockers tab, a facilities tab, a files tab, and a registration tab. For the first three tabs, the user can see a list of all units/parking spots/lockers along with their information. The user can select any of them and modify their information. The facilities tab displays a list of all available facilities of the property with their information. On this tab, the user may create new facilities. The files tab allows the user to see all public and private PDF documents. The user may upload new PDF documents. For any documents on the list, the user may change its privacy setting and/or delete it if desired. For the registration tab, the user can generate a registration key to be given to a new owner or rental user. Before generating the key, the user needs to select a unit from a list of available ones along with parking spot(s) and/or locker(s).
  - Associated User Stories: 13, 14, 15, 16, 17, 18, 22
- Financial Management Page
  - Description: The property profile can access the financial management page where they can view the financial information of the property. The user can see the current rates the property is charging its clients such as the square footage fee, the locker fee, and the parking spot fee. The user can update these fees so they'll be reflected on all client's balances. The user can also view the expenses of the property. The page will display a financial budget of the net income of the property. The page will also provide an annual financial report to the property account where all income and expenses of the property for the past year will be compiled by the date of the income or expense along with a graph showing the profit of the property.
  - Associated User Stories: 19, 20, 21
- Employee Management Page
  - Description: The property profile can access the employee management page where they can view all employees of the property along with their information. The user may view how many requests each employee has pending and their work schedule. The user may also create new employee accounts by setting up their login credentials.
  - Associated User Stories: 23,
- Property Notification Page

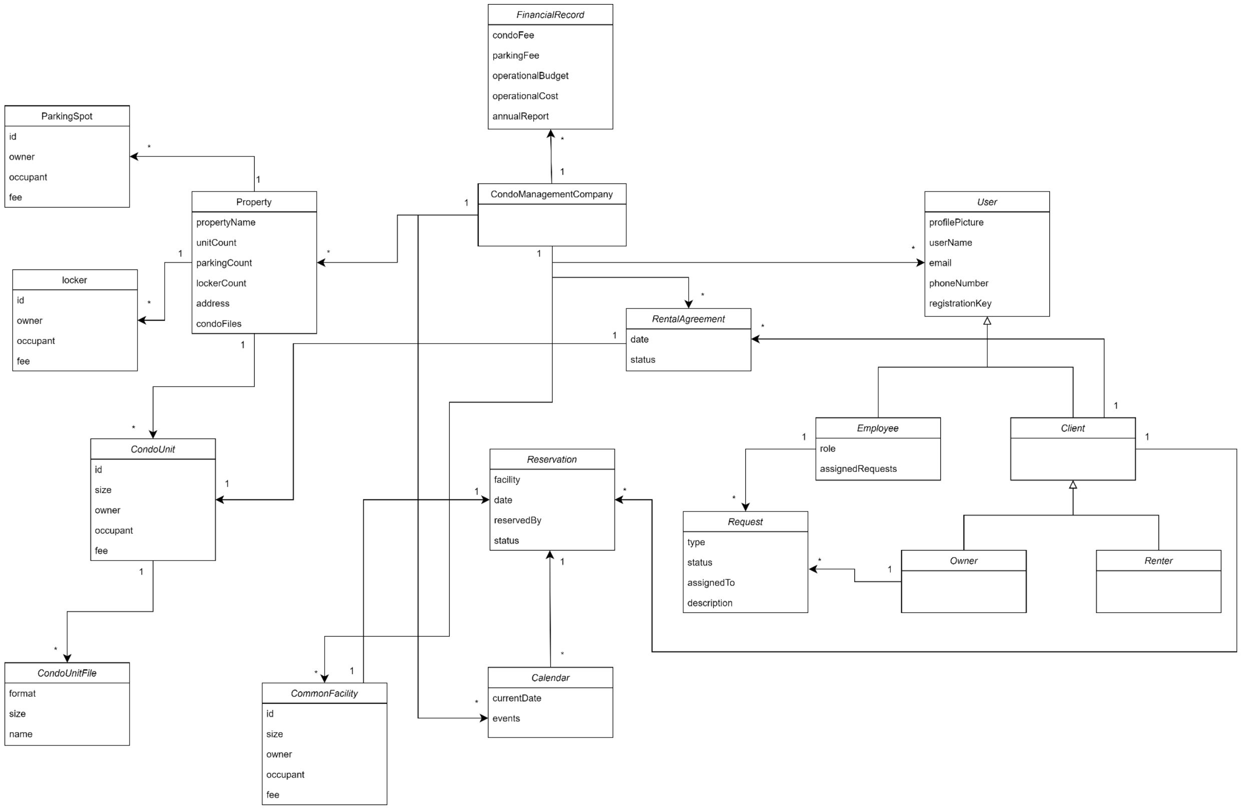
    - Description: The property profile should have access to the property notification page. On this page, they should see two tabs: pending and completed requests. The pending requests are a list of requests from client users that have been assigned to any property employees. On the completed requests tab, they will see a list of all completed requests. On both tabs, each request includes their information such as type of request, client message, status, employee assigned, employee message, time of submission, and time of response.
    - Associated User Stories: 25,
- Employee Notification Page
    - Description: If the user logs in with an employee account, they should have access to the employee notification page. On this page, they should see two tabs: pending and completed requests. The pending requests are a list of requests from client users that have been assigned to the given employee. When the employee clicks on one, they can select to confirm or deny the request and, in either case, send a response to the client. Once the employee confirms or denies the request, it is moved to the list of completed requests in its tab.
    - Associated User Stories: 27,
- Forum Page
    - Description: Both client accounts and property accounts would have access to the forum page. On this page, a list of forum posts will be displayed in order of how recently the post was published. The property account may decide to pin one or multiple posts to the top of the list. Clicking on a post will open its full text, and the comments, and give the user the option to comment under it.
    - Associated User Stories: 28, 29

System Architecture

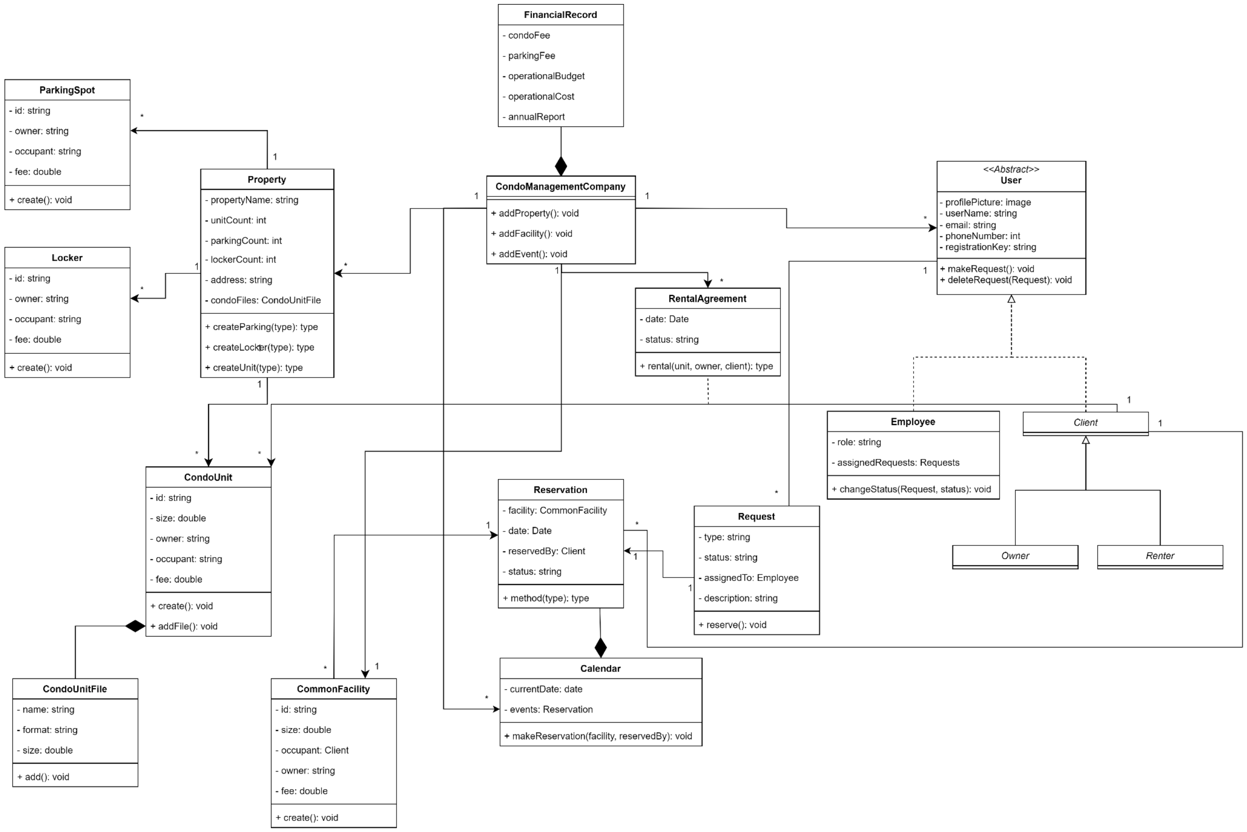
Component Diagram



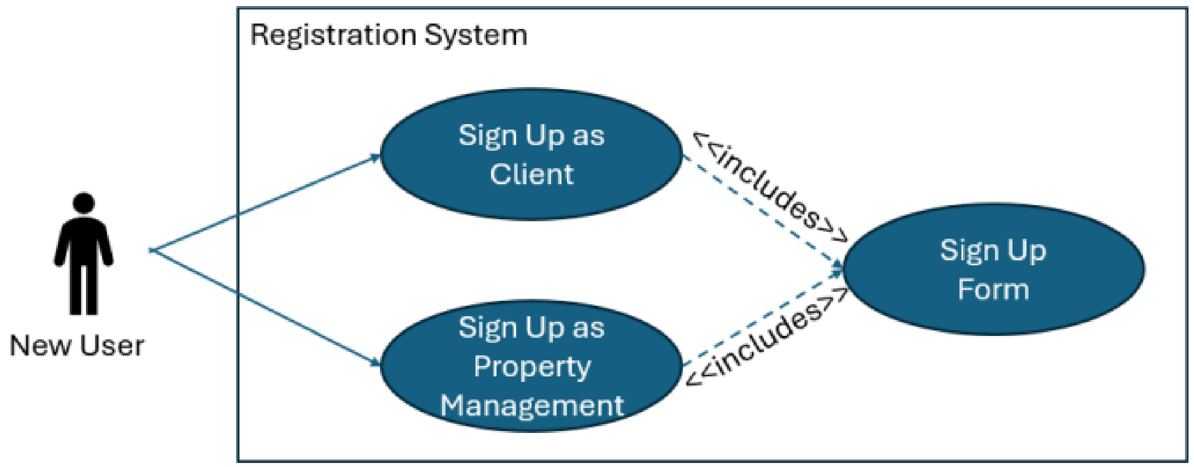
Domain Model

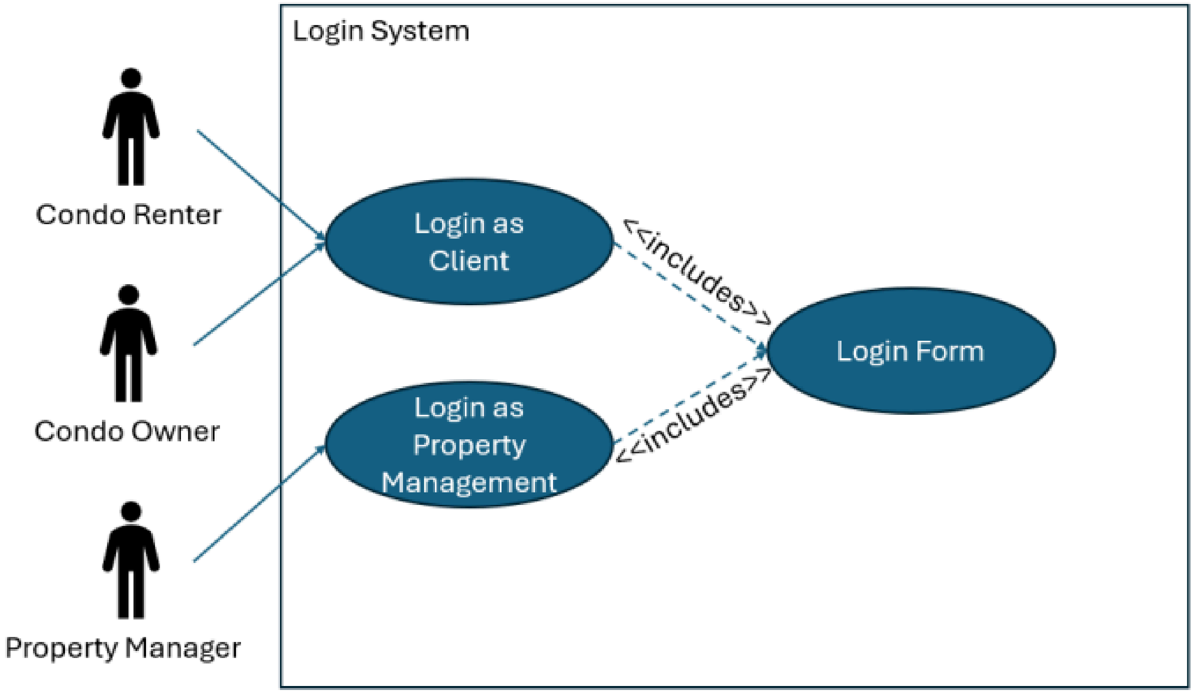


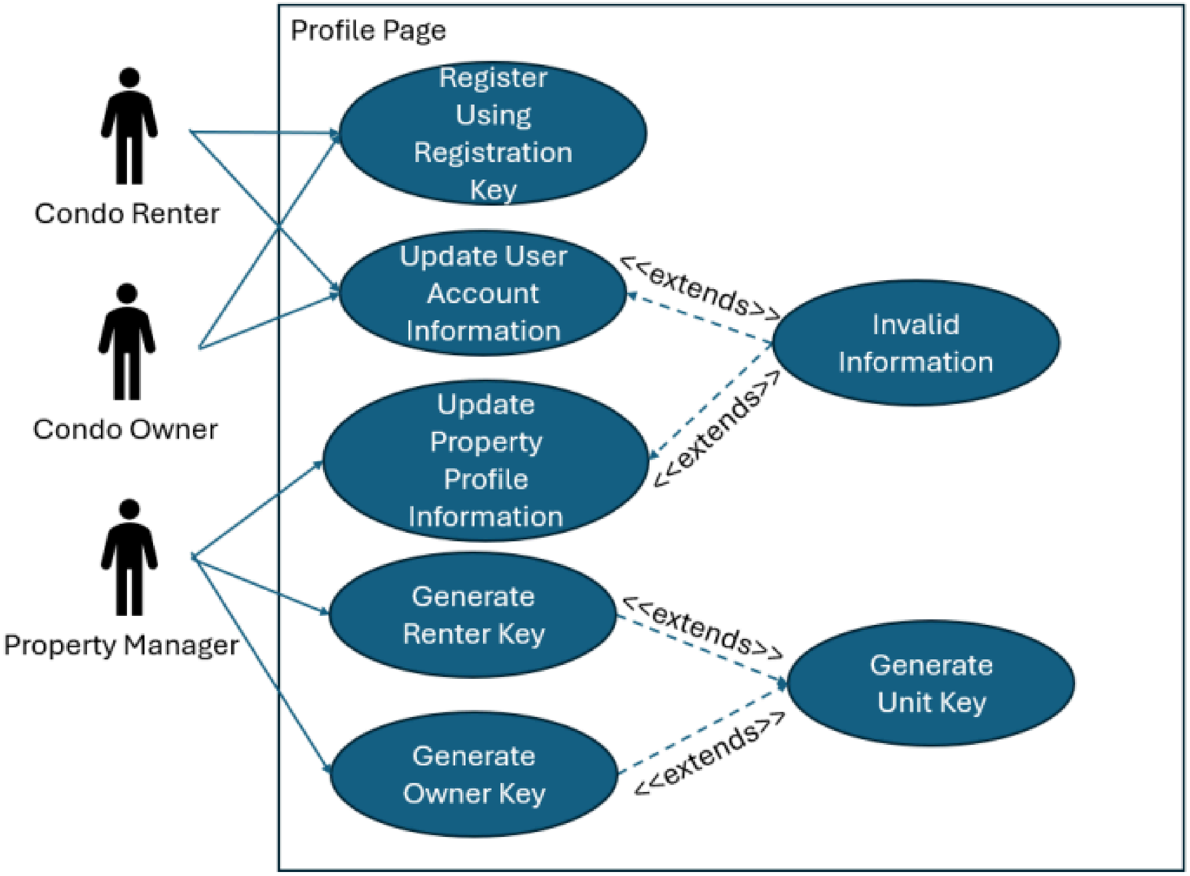
Class Diagram

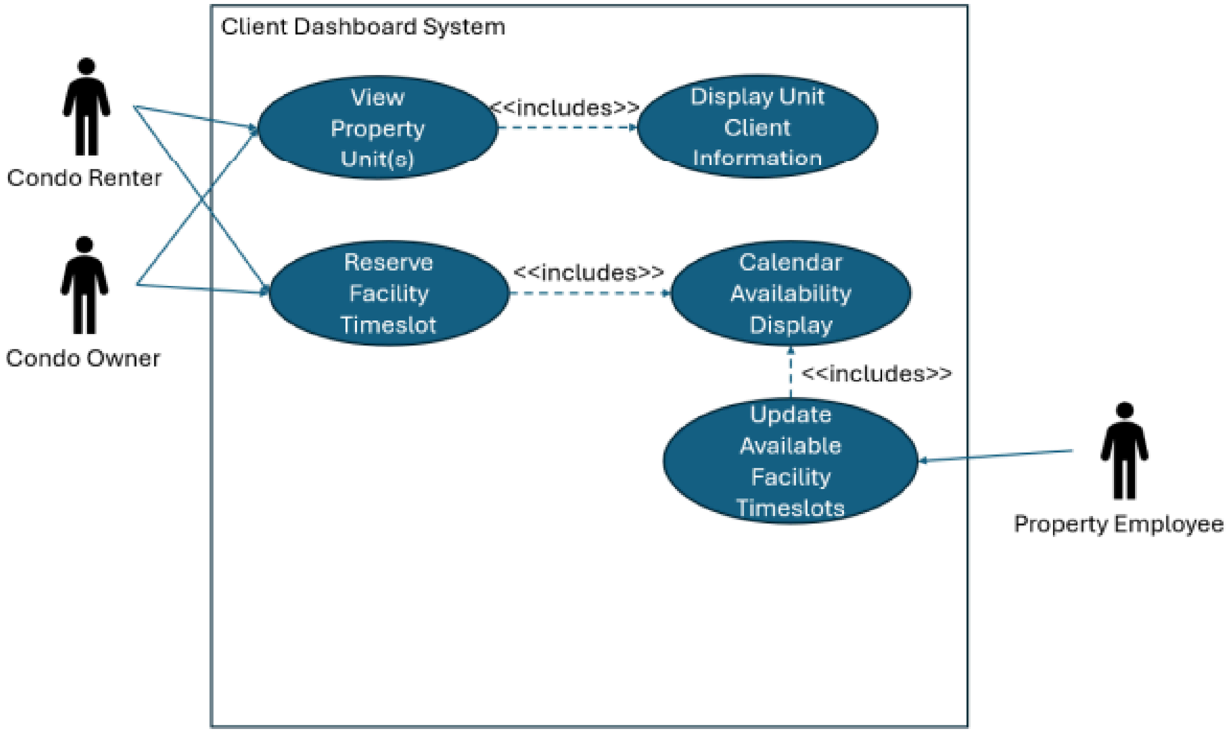


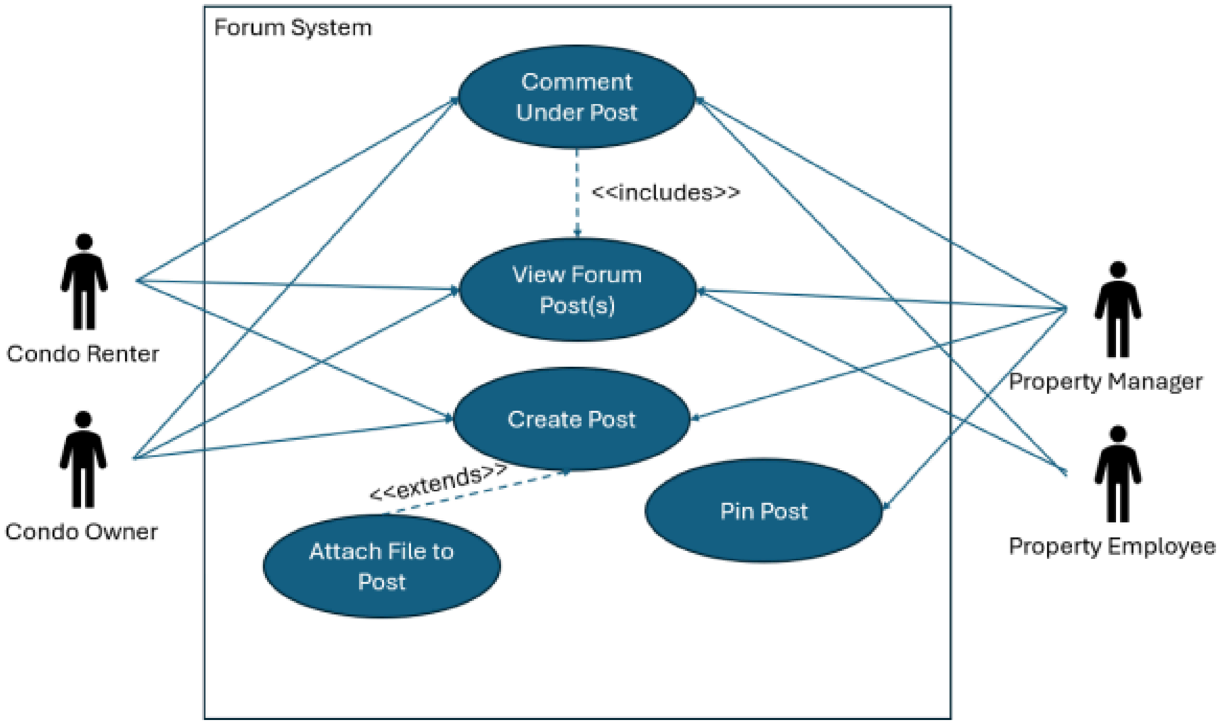
Use Case Diagrams






Profile Page
Condo Renter
Condo Owner
Property Manager
Register Using Registration Key
Update User Account Information
Update Property Profile Information
Generate Renter Key
Generate Owner Key
Invalid Information
Generate Unit Key
<<extends>>
<<extends>>
<<extends>>
<<extends>>


Client Dashboard System
Condo Renter
Condo Owner
View Property Unit(s)
<<includes>>
Display Unit Client Information
Reserve Facility Timeslot
<<includes>>
Calendar Availability Display
<<includes>>
Update Available Facility Timeslots
Property Employee


Forum System
Comment Under Post
<<includes>>
View Forum Post(s)
Create Post
<<extends>>
Attach File to Post
Pin Post
Condo Renter
Condo Owner
Property Manager
Property Employee

Testing

Testing Plan

This section outlines a comprehensive testing plan that incorporates unit, integration, and system testing methodologies, leveraging the Jest testing framework alongside automation strategies for efficient test execution. Jest, a delightful JavaScript Testing Framework focusing on simplicity, offers features such as a zero-configuration setup, instant feedback, and fast execution of tests. Integrating Jest into our testing plan will ensure a streamlined and effective testing process, enhancing code quality and reliability.

Scope

- Unit Testing: Isolating and testing individual units/components for correctness.
- Integration Testing: Verifying the interactions between different units/components.
- System Testing: Testing the complete and integrated software to verify it meets requirements.

Tools and Technology

- Jest: Primary framework for writing and executing tests.
- CI/CD Pipeline (e.g., Jenkins, GitHub Actions): Automate testing and deployment processes.
- Mocking Libraries (e.g., jest.mock): For simulating interactions with modules or external services.

Test Strategy

1. Unit Testing
    - Objective: Test individual units/components in isolation to ensure they function correctly.
    - Approach: Use Jest to write and execute tests for each component. Utilize mocking and spies to isolate the component from its dependencies.
    - Execution: Developers are responsible for writing unit tests as they develop features. Continuous Integration (CI) pipelines will automatically run these tests on every commit to the version control system.
2. Integration Testing
    - Objective: Ensure that integrated components work together as intended.
    - Approach: Identify critical integration points within the application and write tests that simulate the interaction between these components using Jest.
    - Execution: Integration tests will be executed after successful completion of unit tests in the CI pipeline. They can also be run manually for specific components during development.

3. System Testing
    - Objective: Validate the system as a whole against requirements.
    - Approach: Use Jest in conjunction with end-to-end testing tools (e.g., Puppeteer, Selenium) to write tests that simulate real user scenarios.
    - Execution: System tests will be executed as part of the deployment phase in the Continuous Deployment (CD) pipeline, ensuring the software is tested in an environment that closely mirrors production.

Automation Strategy

- Continuous Integration (CI): Integrate Jest tests into the CI pipeline to run automatically on every push to the main/version control branch, ensuring immediate feedback on the impact of changes.
- Continuous Deployment (CD): Configure the CD pipeline to run system tests automatically before deployment, ensuring that only thoroughly tested code is deployed to production.
- Scheduled Runs: Schedule regular test runs for system tests to identify issues that external dependencies or environment changes might introduce over time.

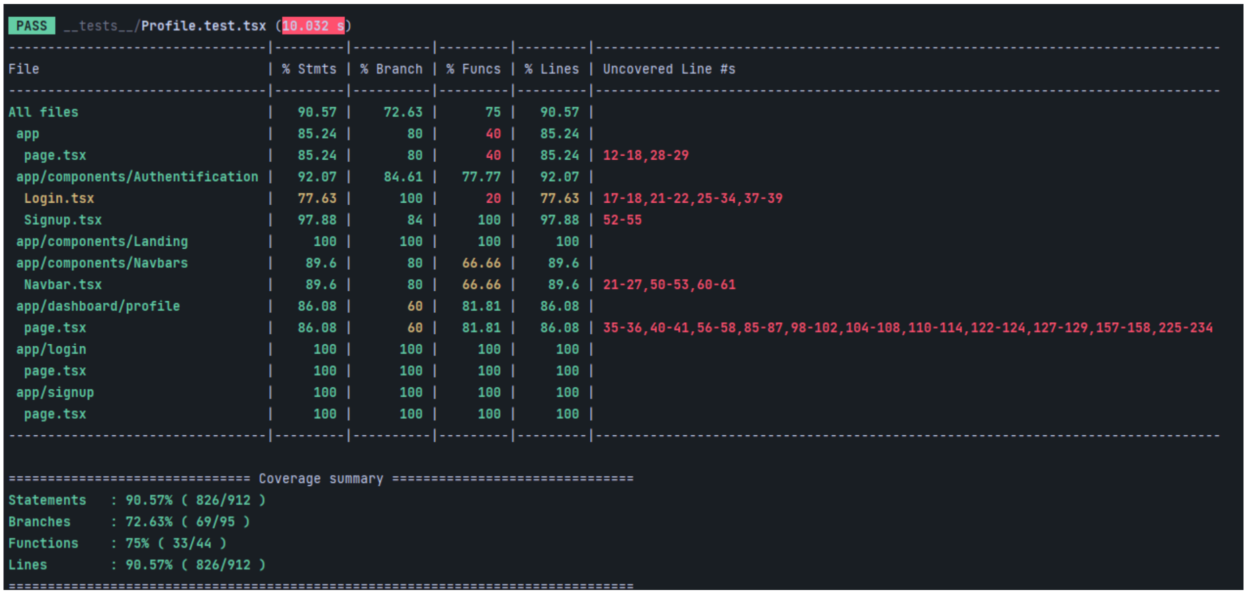
Best Practices

- Test Naming: Follow a consistent naming convention that clearly describes what each test does.
- Test Data Management: Use separate test data from production data, ensuring tests do not interfere with live environments.
- Version Control Integration: Integrate the testing suite with version control systems to ensure tests are versioned alongside the code they test.
- Documentation: Maintain thorough documentation for test cases, including the purpose and expected outcomes, to facilitate maintenance and understanding

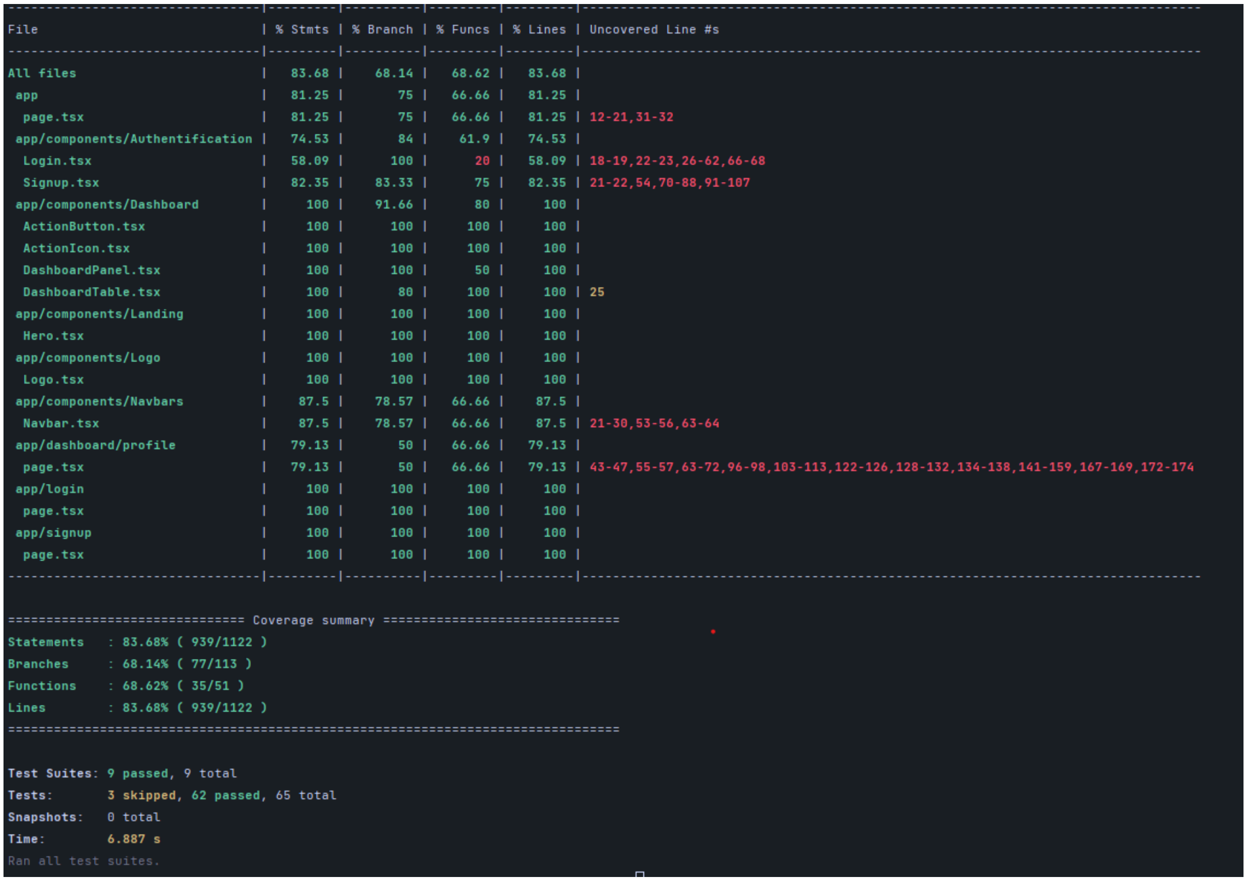
Implementing a structured testing plan using Jest and automation will significantly improve the quality and reliability of the software. By covering unit, integration, and system testing, we can ensure that individual components, their interactions, and the entire system function as expected. Automating these tests within CI/CD pipelines not only streamlines the development process but also enables quick detection and resolution of issues, leading to a more efficient and error-free development lifecycle.

Testing Reports

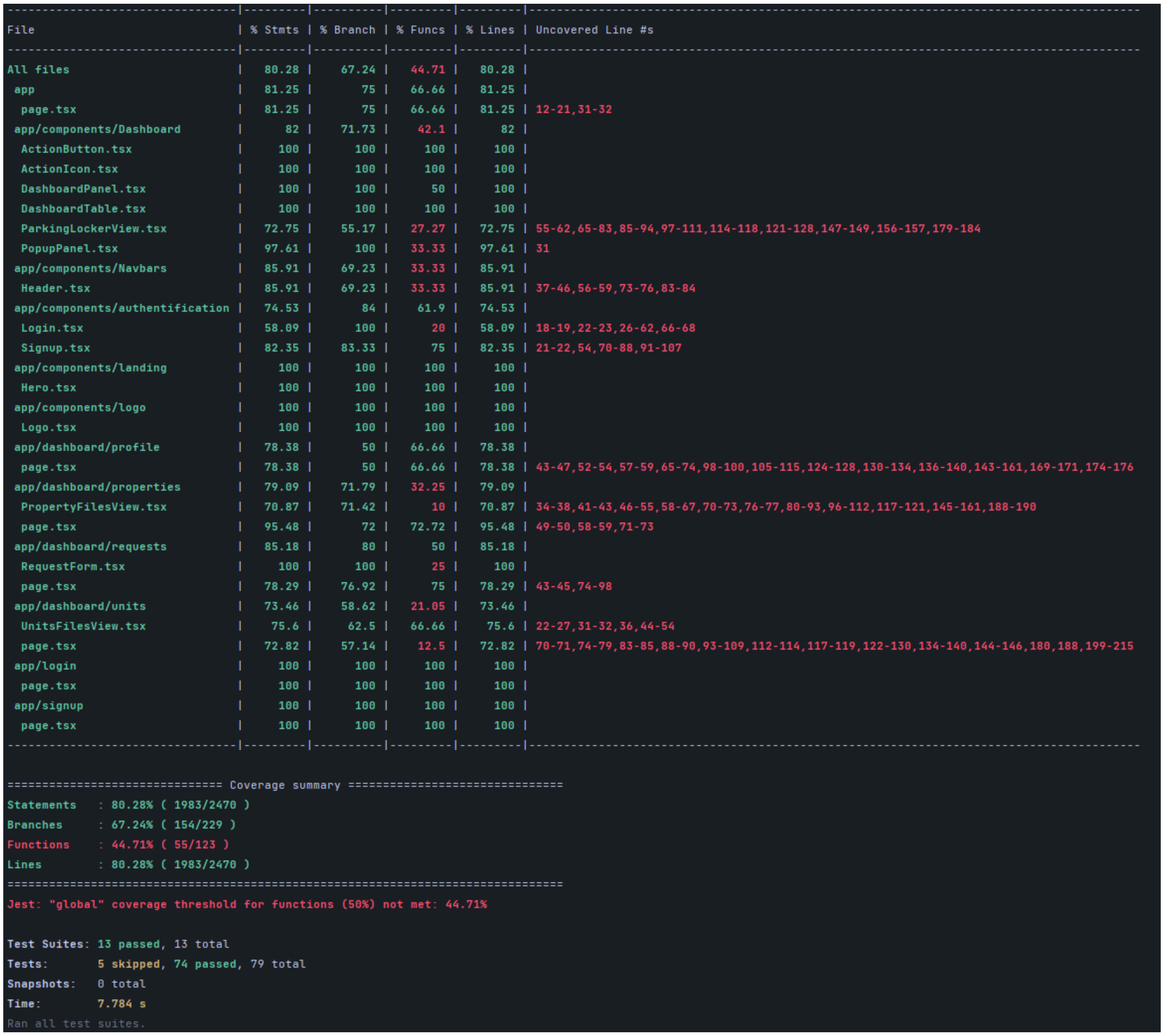
Sprint 3



```
PASS  __tests__/Profile.test.tsx (10.032 s)
----------------------------------|---------|----------|---------|---------|------------------------------------------------------------------------------------
File                              | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------------------|---------|----------|---------|---------|------------------------------------------------------------------------------------
All files                         |   90.57 |    72.63 |      75 |   90.57 |
 app                              |   85.24 |       80 |      40 |   85.24 |
  page.tsx                        |   85.24 |       80 |      40 |   85.24 | 12-18,28-29
 app/components/Authentification  |   92.07 |    84.61 |   77.77 |   92.07 |
  Login.tsx                       |   77.63 |      100 |      20 |   77.63 | 17-18,21-22,25-34,37-39
  Signup.tsx                      |   97.88 |       84 |     100 |   97.88 | 52-55
 app/components/Landing           |     100 |      100 |     100 |     100 |
 app/components/Navbars           |    89.6 |       80 |   66.66 |    89.6 |
  Navbar.tsx                      |    89.6 |       80 |   66.66 |    89.6 | 21-27,50-53,60-61
 app/dashboard/profile            |   86.08 |       60 |   81.81 |   86.08 |
  page.tsx                        |   86.08 |       60 |   81.81 |   86.08 | 35-36,40-41,56-58,85-87,98-102,104-108,110-114,122-124,127-129,157-158,225-234
 app/login                        |     100 |      100 |     100 |     100 |
  page.tsx                        |     100 |      100 |     100 |     100 |
 app/signup                       |     100 |      100 |     100 |     100 |
  page.tsx                        |     100 |      100 |     100 |     100 |
----------------------------------|---------|----------|---------|---------|------------------------------------------------------------------------------------

=============================== Coverage summary ===============================
Statements   : 90.57% ( 826/912 )
Branches     : 72.63% ( 69/95 )
Functions    : 75% ( 33/44 )
Lines        : 90.57% ( 826/912 )
================================================================================
```

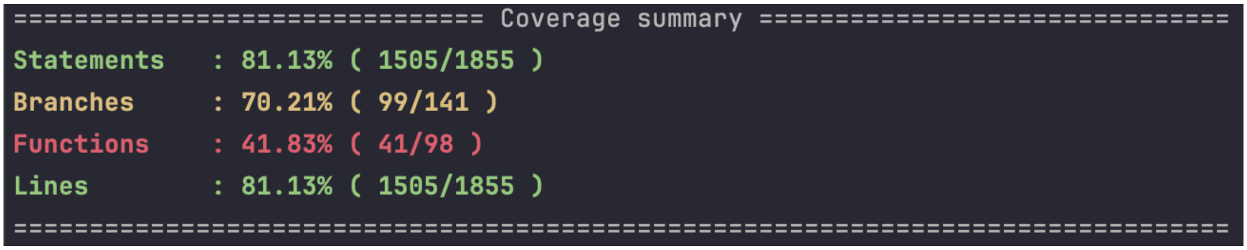


```
----------------------------------|---------|----------|---------|---------|---------------------------------------------------------------------------------
File                              | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------------------|---------|----------|---------|---------|---------------------------------------------------------------------------------
All files                         |   83.68 |    68.14 |   68.62 |   83.68 |
 app                              |   81.25 |       75 |   66.66 |   81.25 |
  page.tsx                        |   81.25 |       75 |   66.66 |   81.25 | 12-21,31-32
 app/components/Authentification  |   74.53 |       84 |    61.9 |   74.53 |
  Login.tsx                       |   58.09 |      100 |      20 |   58.09 | 18-19,22-23,26-62,66-68
  Signup.tsx                      |   82.35 |    83.33 |      75 |   82.35 | 21-22,54,70-88,91-107
 app/components/Dashboard         |     100 |    91.66 |      80 |     100 |
  ActionButton.tsx                |     100 |      100 |     100 |     100 |
  ActionIcon.tsx                  |     100 |      100 |     100 |     100 |
  DashboardPanel.tsx              |     100 |      100 |      50 |     100 |
  DashboardTable.tsx              |     100 |       80 |     100 |     100 | 25
 app/components/Landing           |     100 |      100 |     100 |     100 |
  Hero.tsx                        |     100 |      100 |     100 |     100 |
 app/components/Logo              |     100 |      100 |     100 |     100 |
  Logo.tsx                        |     100 |      100 |     100 |     100 |
 app/components/Navbars           |    87.5 |    78.57 |   66.66 |    87.5 |
  Navbar.tsx                      |    87.5 |    78.57 |   66.66 |    87.5 | 21-30,53-56,63-64
 app/dashboard/profile            |   79.13 |       50 |   66.66 |   79.13 |
  page.tsx                        |   79.13 |       50 |   66.66 |   79.13 | 43-47,55-57,63-72,96-98,103-113,122-126,128-132,134-138,141-159,167-169,172-174
 app/login                        |     100 |      100 |     100 |     100 |
  page.tsx                        |     100 |      100 |     100 |     100 |
 app/signup                       |     100 |      100 |     100 |     100 |
  page.tsx                        |     100 |      100 |     100 |     100 |
----------------------------------|---------|----------|---------|---------|---------------------------------------------------------------------------------

=============================== Coverage summary ===============================
Statements   : 83.68% ( 939/1122 )
Branches     : 68.14% ( 77/113 )
Functions    : 68.62% ( 35/51 )
Lines        : 83.68% ( 939/1122 )
================================================================================

Test Suites: 9 passed, 9 total
Tests:       3 skipped, 62 passed, 65 total
Snapshots:   0 total
Time:        6.887 s
Ran all test suites.
```



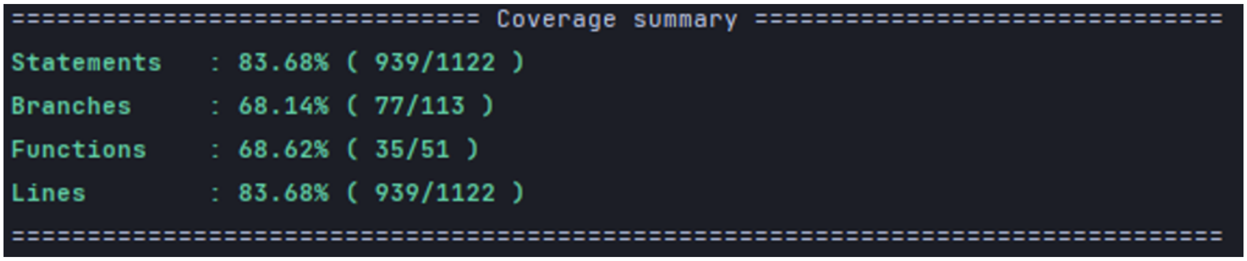
```
---------------------------------|---------|----------|---------|---------|----------------------------------------------------------------------------------------
File                             | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
---------------------------------|---------|----------|---------|---------|----------------------------------------------------------------------------------------
All files                        |   80.28 |    67.24 |   44.71 |   80.28 |
 app                             |   81.25 |       75 |   66.66 |   81.25 |
  page.tsx                       |   81.25 |       75 |   66.66 |   81.25 | 12-21,31-32
 app/components/Dashboard        |      82 |    71.73 |    42.1 |      82 |
  ActionButton.tsx               |     100 |      100 |     100 |     100 |
  ActionIcon.tsx                 |     100 |      100 |     100 |     100 |
  DashboardPanel.tsx             |     100 |      100 |      50 |     100 |
  DashboardTable.tsx             |     100 |      100 |     100 |     100 |
  ParkingLockerView.tsx          |   72.75 |    55.17 |   27.27 |   72.75 | 55-62,65-83,85-94,97-111,114-118,121-128,147-149,156-157,179-184
  PopupPanel.tsx                 |   97.61 |      100 |   33.33 |   97.61 | 31
 app/components/Navbars          |   85.91 |    69.23 |   33.33 |   85.91 |
  Header.tsx                     |   85.91 |    69.23 |   33.33 |   85.91 | 37-46,56-59,73-76,83-84
 app/components/authentification |   74.53 |       84 |    61.9 |   74.53 |
  Login.tsx                      |   58.09 |      100 |      20 |   58.09 | 18-19,22-23,26-62,66-68
  Signup.tsx                     |   82.35 |    83.33 |      75 |   82.35 | 21-22,54,70-88,91-107
 app/components/landing          |     100 |      100 |     100 |     100 |
  Hero.tsx                       |     100 |      100 |     100 |     100 |
 app/components/logo             |     100 |      100 |     100 |     100 |
  Logo.tsx                       |     100 |      100 |     100 |     100 |
 app/dashboard/profile           |   78.38 |       50 |   66.66 |   78.38 |
  page.tsx                       |   78.38 |       50 |   66.66 |   78.38 | 43-47,52-54,57-59,65-74,98-100,105-115,124-128,130-134,136-140,143-161,169-171,174-176
 app/dashboard/properties        |   79.09 |    71.79 |   32.25 |   79.09 |
  PropertyFilesView.tsx          |   70.87 |    71.42 |      10 |   70.87 | 34-38,41-43,46-55,58-67,70-73,76-77,80-93,96-112,117-121,145-161,188-190
  page.tsx                       |   95.48 |       72 |   72.72 |   95.48 | 49-50,58-59,71-73
 app/dashboard/requests          |   85.18 |       80 |      50 |   85.18 |
  RequestForm.tsx                |     100 |      100 |      25 |     100 |
  page.tsx                       |   78.29 |    76.92 |      75 |   78.29 | 43-45,74-98
 app/dashboard/units             |   73.46 |    58.62 |   21.05 |   73.46 |
  UnitsFilesView.tsx             |    75.6 |     62.5 |   66.66 |    75.6 | 22-27,31-32,36,44-54
  page.tsx                       |   72.82 |    57.14 |    12.5 |   72.82 | 70-71,74-79,83-85,88-90,93-109,112-114,117-119,122-130,134-140,144-146,180,188,199-215
 app/login                       |     100 |      100 |     100 |     100 |
  page.tsx                       |     100 |      100 |     100 |     100 |
 app/signup                      |     100 |      100 |     100 |     100 |
  page.tsx                       |     100 |      100 |     100 |     100 |
---------------------------------|---------|----------|---------|---------|----------------------------------------------------------------------------------------

=============================== Coverage summary ===============================
Statements   : 80.28% ( 1983/2470 )
Branches     : 67.24% ( 154/229 )
Functions    : 44.71% ( 55/123 )
Lines        : 80.28% ( 1983/2470 )
================================================================================
Jest: "global" coverage threshold for functions (50%) not met: 44.71%

Test Suites: 13 passed, 13 total
Tests:       5 skipped, 74 passed, 79 total
Snapshots:   0 total
Time:        7.784 s
Ran all test suites.
```

Sprint 4



```
=============================== Coverage summary ===============================
Statements   : 81.13% ( 1505/1855 )
Branches     : 70.21% ( 99/141 )
Functions    : 41.83% ( 41/98 )
Lines        : 81.13% ( 1505/1855 )
================================================================================
```

Sprint 5



```
=============================== Coverage summary ===============================
Statements   : 83.68% ( 939/1122 )
Branches     : 68.14% ( 77/113 )
Functions    : 68.62% ( 35/51 )
Lines        : 83.68% ( 939/1122 )
================================================================================
```

Retrospectives

Sprint 1

Before starting the sprint, task division went notably well. Everyone selected their assignments, ensuring no overlap in responsibilities. The spirit of collaboration was strong; team members were promptly available to assist others as needed, enhancing our overall efficiency.

However, our time management could have been better. We initiated the sprint too close to the deadline which introduced unnecessary stress and haste into our process. Although we completed all tasks on time, we had little opportunity to review our work due to time constraints.

Moving forward, improving our time management will be our primary focus. By starting earlier, we aim to enhance the quality of our sprints and reduce stress, allowing for a more thoughtful and measured approach to our projects.

Sprint 3

What went well?
Our team coordination and communication this time around was good. The task separation was done and everyone got something to work on and help each other out when needed. We had channels for separate parts of our project where members could ask for help if they were stuck on something or needed advice.

What could have gone better?
Our time management could have gone better. In the end, we got everything done by the deadline, however, it was quite a lot of work to do with how short of time we gave ourselves.

What should change?
We need to start our work way ahead of the deadline of each remaining sprint. This would ensure there is less stress to be had trying to rush everything last minute while maintaining a high-quality project. Working on everything early would also mean we could peer review each

other's work even more than we already do, find problems we wouldn't normally have and generally refine our project.

Sprint 4

What Went Well:
Successfully implemented encryption, and access control to mitigate unauthorized access or data breaches. (Risk management) . Our testing coverage is always above 80%, even with new features being added every sprint. Deadlines were met, leaving time to refine and update documents properly. We successfully had a presentation with the product owner for this sprint.

What Could Have Gone Better:
Updating all our diagrams would have been useful as it would have helped us manage our code even more easily and helped the product owner better understand the presentation and application. We also need to have proper management to prevent financial and legal issues.

Changes for Next Sprint:
We will work on updating our diagrams for each new feature or refinement. We will strengthen our staff and management ensuring everything complies with the law.

Deployment Plan

The app will be deployed on Vercel. Link to HomeHaven

Infrastructure Components

- Web server: Vercel optimizes delivery and performance by automatically serving your application from a globally dispersed network of servers.
- Application Server: To create backend services without having to manage a separate application server, Vercel manages serverless functionalities.
- Database Server: Oversees the management of all transactional data storage, including financial records, property details, and user data. Data security and integrity are guaranteed.
- File Storage: Manages bulky static files, such as PDF budgets, board meeting minutes, and condo declarations.
- Email server: Enables communication between users by providing them with alerts, registration keys, and notifications.

Service Breakdown

- User Management Service: Oversees profile maintenance, user registration, and authentication. The security of user sessions is likewise guaranteed by this service.

- Property Management Service: The component tasked with maintaining and updating records for properties, parking spaces, and lockers. It manages file uploads as well as file access.
- Financial System: Manages every financial operation, such as recording payments, computing condo fees, and generating annual budgets and financial reports.
- Reservation System: Coordinates reservations for common facilities, keeping track of events and enabling reservations.
- Notification Service: Notifies users on a timely basis about general updates, request statuses, and account activity.

Security Measures

- SSL/TLS Encryption: Applied automatically by Vercel for every application that has been deployed, guaranteeing safe data transfer.
- Firewalls: Implemented to secure data from unwanted access and secure the network.
- Data Encryption: To protect data privacy, sensitive information is encrypted and kept in files and databases. Make sure sensitive data is stored securely by handling it within application logic.
- Authentication and Authorization: Robust mechanisms are in place to enforce access restrictions and confirm user identities. Overseen by outside providers and incorporated into the serverless functions.

Backup and Recovery

- Data Backups: To guarantee that data can be restored in the case of a data loss scenario, regular backups of databases and important file storage are carried out. Managed by the outside database supplier, guaranteeing that standard backup procedures are followed.
- Disaster Recovery Plans: Extensive plans are set up to ensure that services can be continued even in the event of serious failures, with minimal downtime. We guarantee high availability and fault tolerance by combining Vercel's deployment techniques with the resilience of cloud providers.

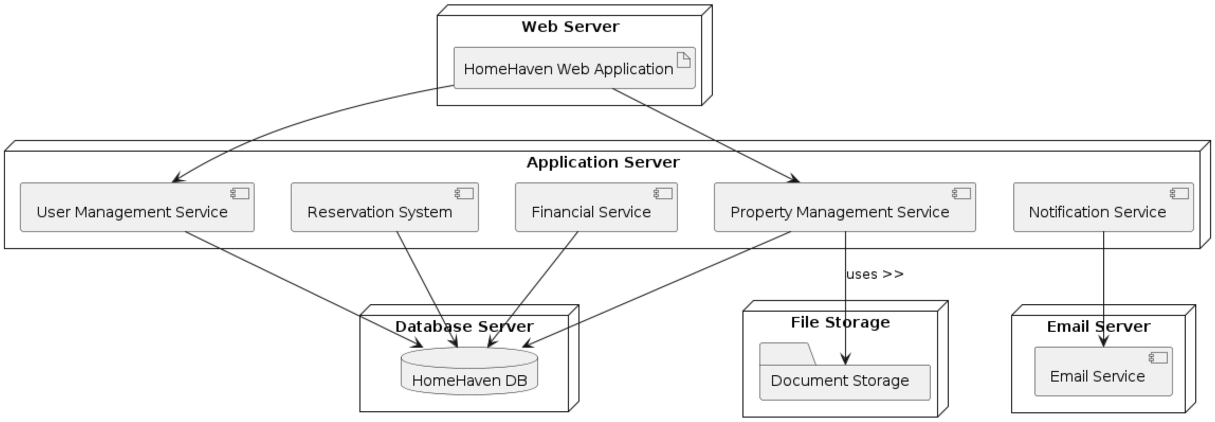
Monitoring and Maintenance

- System Monitoring: To detect and fix problems early, a server's health, performance indicators, and application logs are continuously monitored. Vercel is equipped with built-in analytics tools.
- Log Management: The methodical gathering, examination, and preservation of logs to support security audits and debugging.

Performance and Scalability

- Load Balancers: Automatically handled by Vercel’s edge network to improve availability and responsiveness by equitably distributing incoming traffic among multiple servers.
- Resource Scaling: Since Vercel handles the instantiation and scaling of serverless functions, the serverless architecture naturally allows scalability based on request load.

Deployment Diagram



Drive Archive

Link to the Archive