EstateFlow

Condo Management Made Easy

Sarah Abellard - 40184667
Andy Sun - 40192040
Barthan Thirunavukarasu - 40098158
Abisan Uthamacumaran - 40078084
Yousef Enein - 40115494

Ziyi Wang - 40167122
Fatoumata Binta Barry - 40213443
Samuel Collette - 40175048
Arshpreet Singh - 40172137
Jeremie Garzon - 40062316

Website:
https://estate-api-production.up.railway.app/

Github:
https://github.com/Irisvella/SOEN-390-W2024

Project Background

Problem Statement

- Access to Information: There is a lack of centralized access to essential property-related information.
- Challenges in communication, tracking costs, and booking amenities

Market or Industry Context

- Growing need for digital solutions in real estate management
- Increasing number of condo owners and rental users seeking streamlined services

Project Objectives and Goals

- Develop a user-friendly condo management app and companion website
- Provide a centralized dashboard for property overview and financial administration
- Implement features like personal profiles, property profiles, and reservation systems for amenities

Project Scope

- **Our Solution**

  A comprehensive digital platform designed to simplify condo management through a user-friendly app and website.

- **Target Audience or Users**
  - Condo Owners: Access property details, financial statuses, and make service requests.
  - Renters: Enjoy similar functionalities as condo owners, including reservations.
  - Condo Management Companies: Manage multiple properties, set up financial operations, and oversee reservations.

Development Process Timeline

Development Process Overview:

1. **Requirements Gathering and Analysis (RGA)**
   - Sprint 1: Initial requirements and user stories, risk assessment.
   - Sprint 2: Updated requirements, focus on meaningfulness and formality.
   - Continuous minor updates and reviews in subsequent sprints.
2. **System Design (SD)**
   - Sprint 1: Initial software architecture document (SAD), domain model.
   - Sprint 2 and 3: Updated SAD and risk assessments, enhanced design documents.
   - Sprint 4: Further refinement in design, deployment diagram enhancements.
3. **Implementation (Impl)**
   - Sprint 1: Early prototype development.
   - Sprint 2-5: Continuous implementation, including UI prototypes, and feature expansions.
   - Focus on code quality, design patterns, and implementation of functionalities like the reservation system.
4. **Testing (Test)**
   - Sprint 2: Initial testing plans and prototype usability.
   - Sprint 3-5: Enhanced testing with a focus on UI, system coverage, and code quality.
   - Continuous integration of automated testing and code reviews.

In summary without realising , each sprint really had a focus on one of the steps of the software development process

This graph depicts which process was emphasized in each sprint:

- Sprint 1: Heavy on RGA and SD, minimal Impl, no Testing.
- Sprint 2: All areas covered, with a strong emphasis on further RGA and SD.
- Sprint 3: Strong focus on Impl and Test while continuing SD.
- Sprint 4: Balanced focus across all areas, with incremental advancements in each.
- Sprint 5: Emphasis on finalizing Implementation and comprehensive Testing.

Requirements Gathering and Analysis

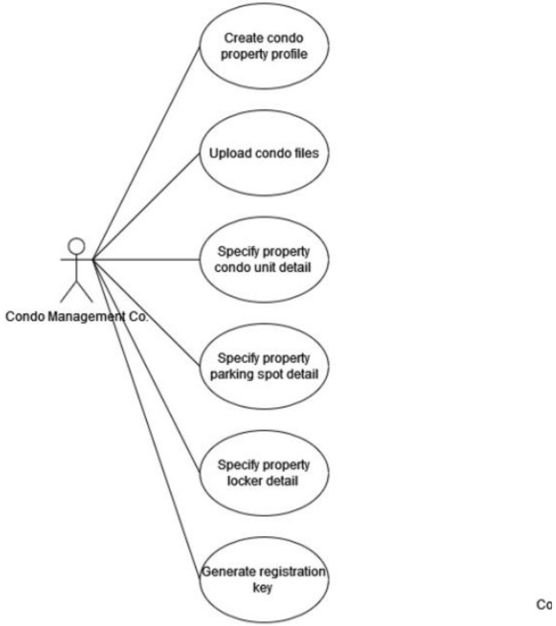
Requirements Gathering and Analysis

- Set a project scope and identify the problem statement
- Identify stakeholders and users, and both their concerns
- Break down requirements
  - Generate user stories
- Set up a tool to track these user stories
  - Github issues and backlog document

System Architecture Design

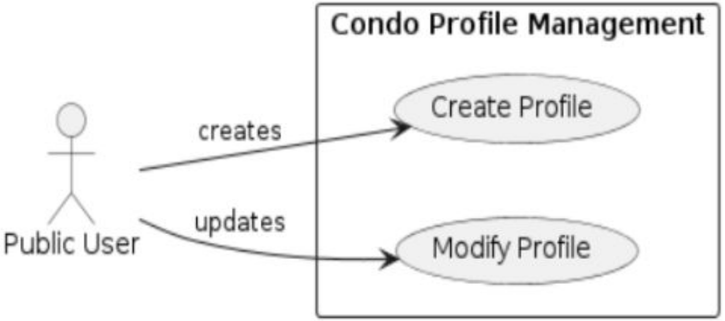
Use Case Diagrams

Use Case 1 : A condo management company can create and manage profiles for their properties


Create condo property profile
Upload condo files
Specify property condo unit detail
Specify property parking spot detail
Specify property locker detail
Generate registration key
Condo Management Co.
Co

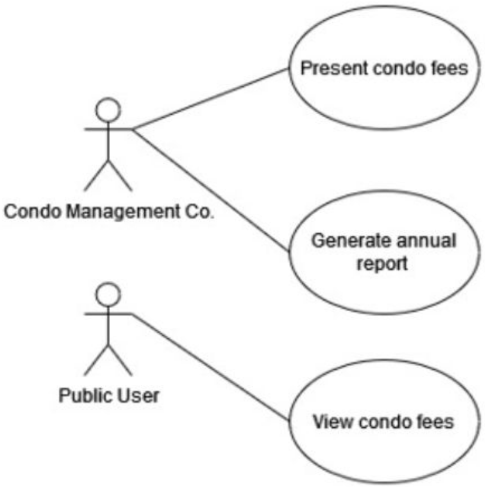
Use Case 3: Calculating condo unit financials

Use Case 2: A condo user can manage their condo profile

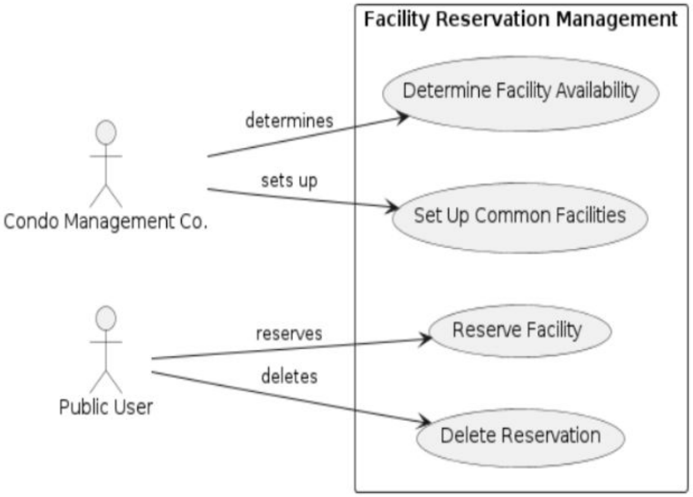


Use Case Diagrams

Use Case 3: Calculating condo unit financials

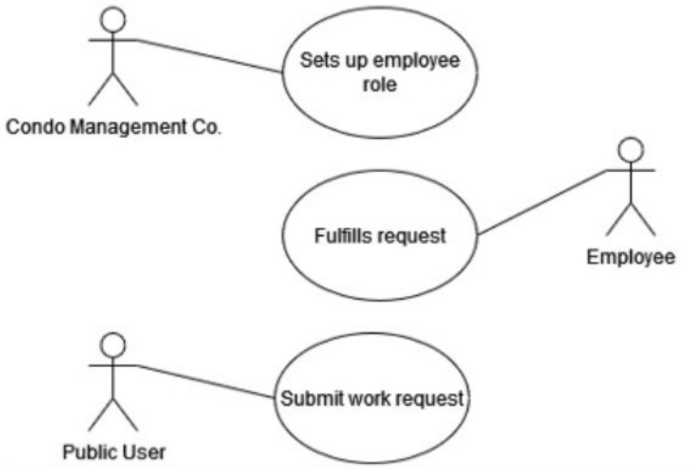


Use Case 4: Common Facilities Reservation Use Case Diagram

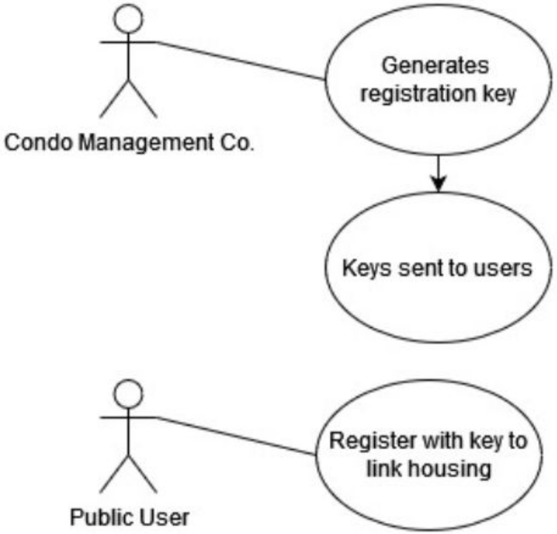


Use Case Diagrams

Use case 5: Work requests are treated according to type

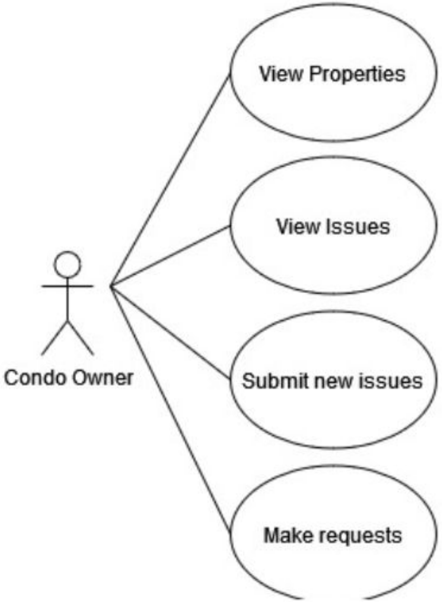


Use case 6: registration keys sent and used

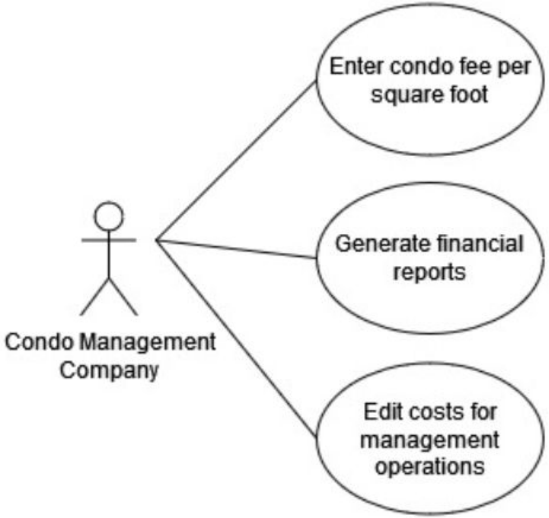


Use Case Diagrams

Use case 7: Condo Owner accesses their features through their dashboard

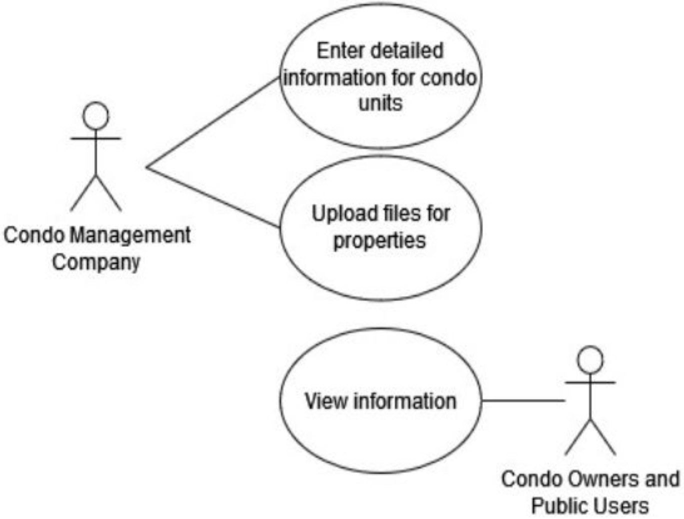


Use Case 8: Condo management company using the financial systems features

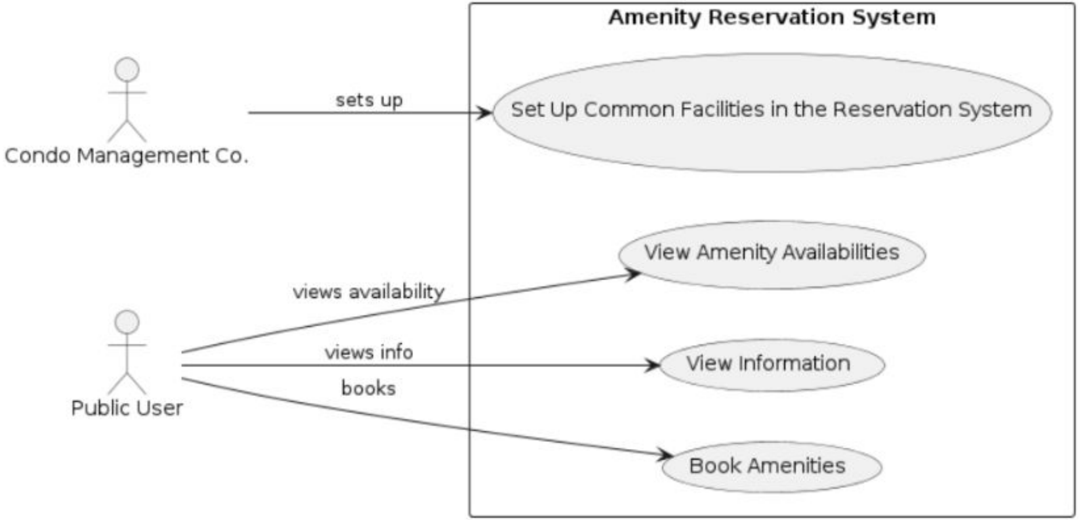

Enter condo fee per square foot
Generate financial reports
Edit costs for management operations
Condo Management Company

Use Case Diagrams

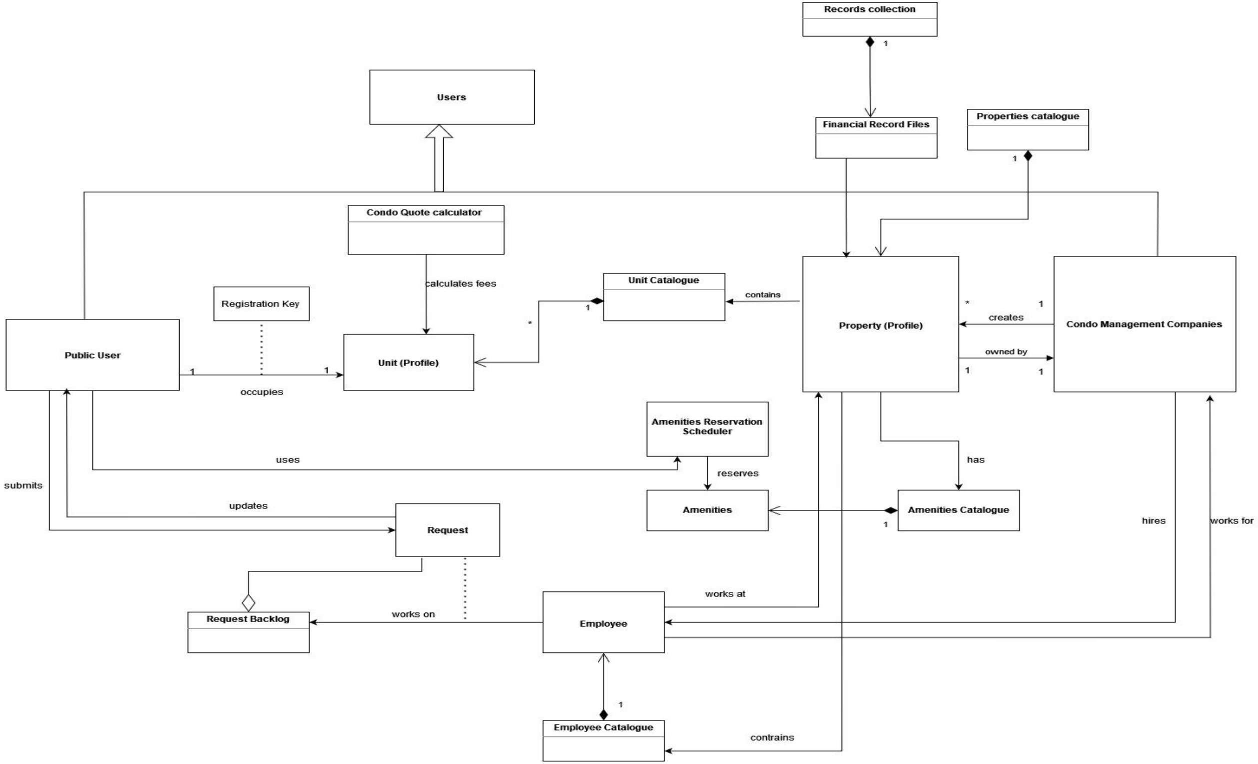
Use Case 9: Condo management company uploading important condo information to be viewable by other users


Enter detailed information for condo units
Upload files for properties
Condo Management Company
View information
Condo Owners and Public Users

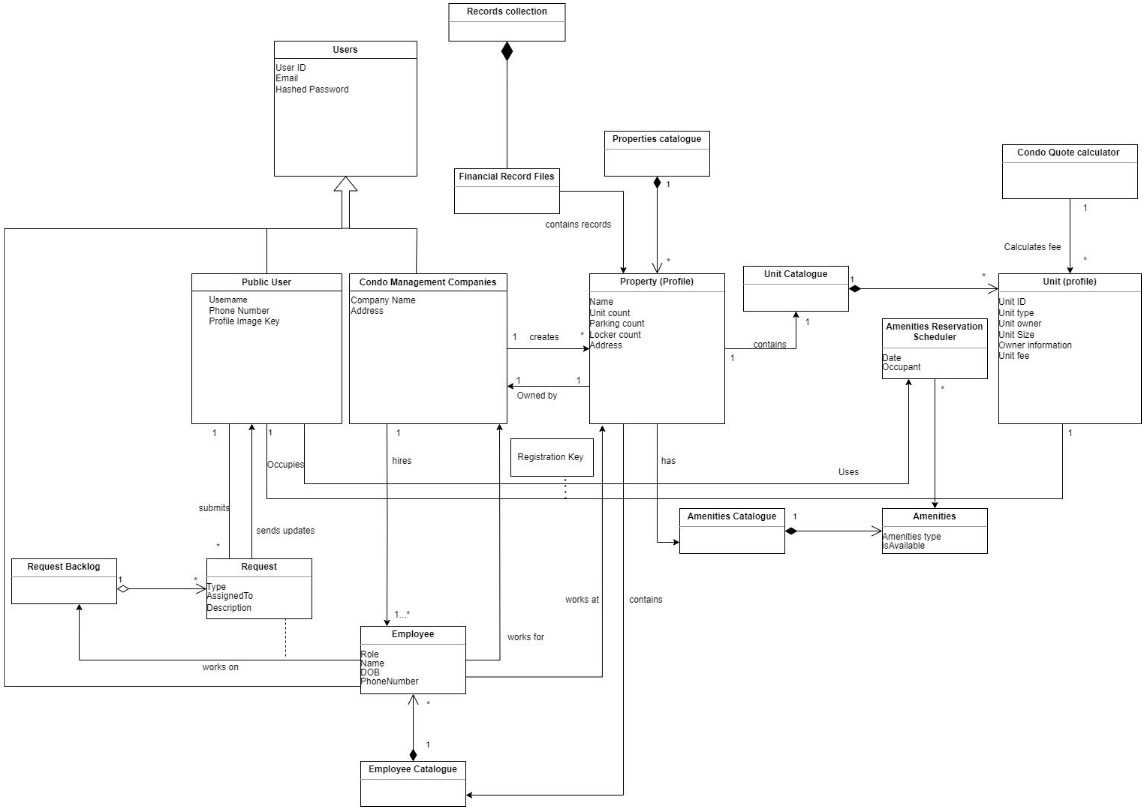
Use Case 10: View, Check Availability, and Book Amenities Use Case Diagram


Amenity Reservation System
Condo Management Co.
sets up
Set Up Common Facilities in the Reservation System
Public User
views availability
View Amenity Availabilities
views info
View Information
books
Book Amenities

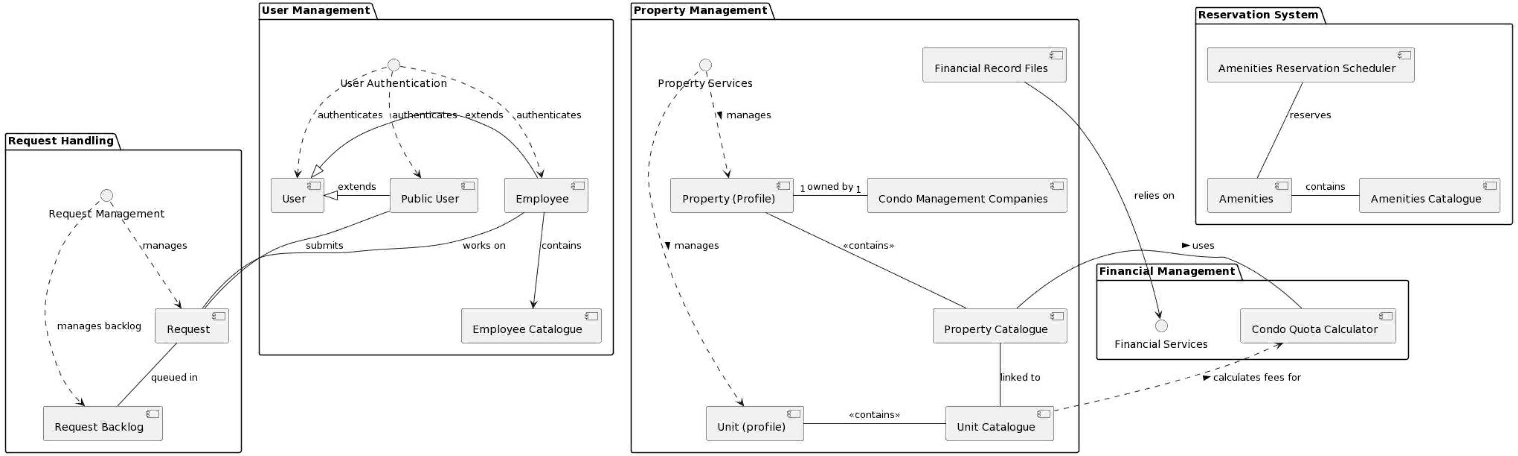
Domain Model



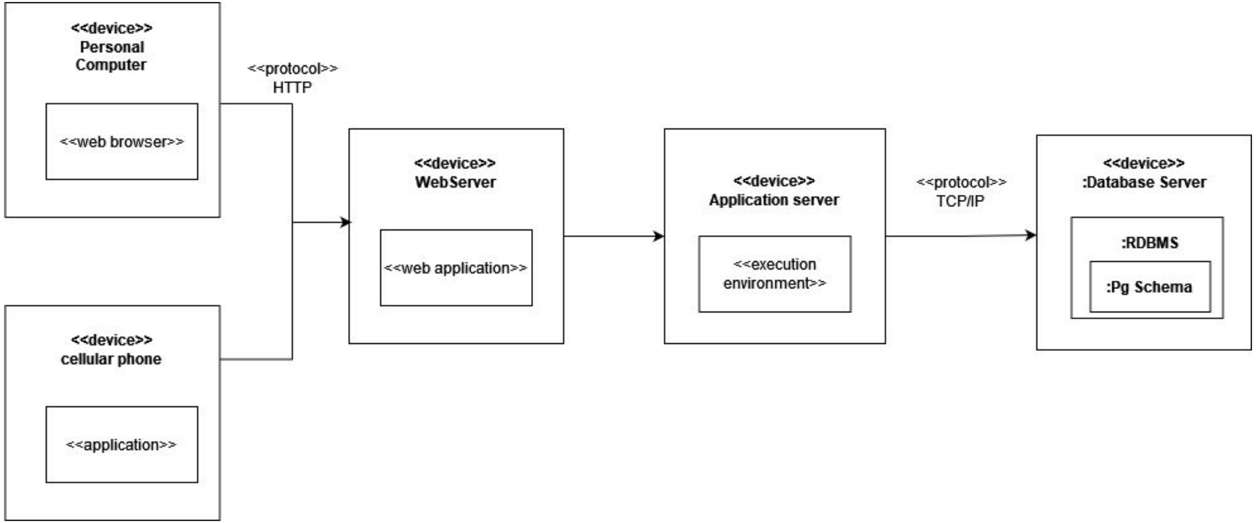
Class Diagram



Component Diagram



Deployment Diagram



Implementation

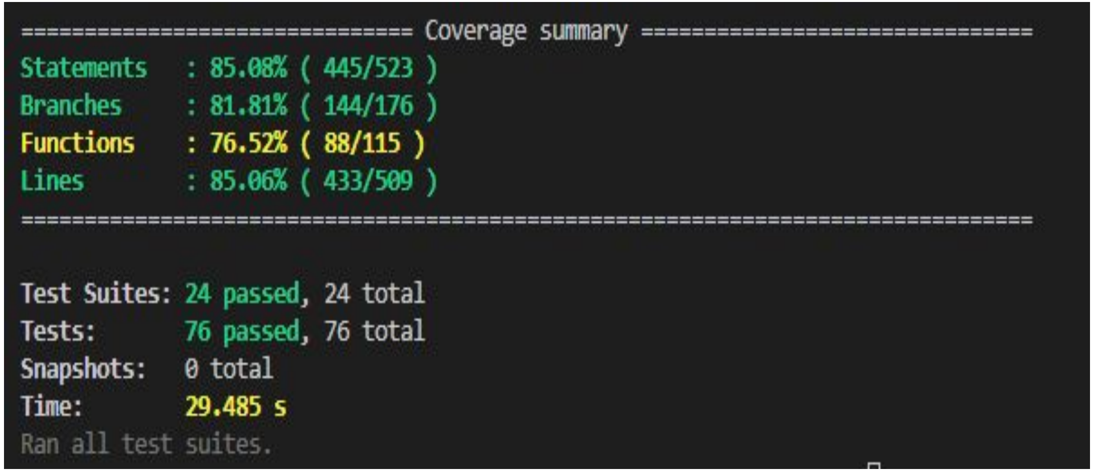
Implementation

- Coding based on aforementioned design
  - Modular, atomic commits, good documentation, etc...
- Integrating smaller components and making sure they work in the larger system
- Debugging and troubleshooting any errors that appear
- Both create good code documentation, but also update design documentation to any changes or decisions made.

Testing

Testing and QA

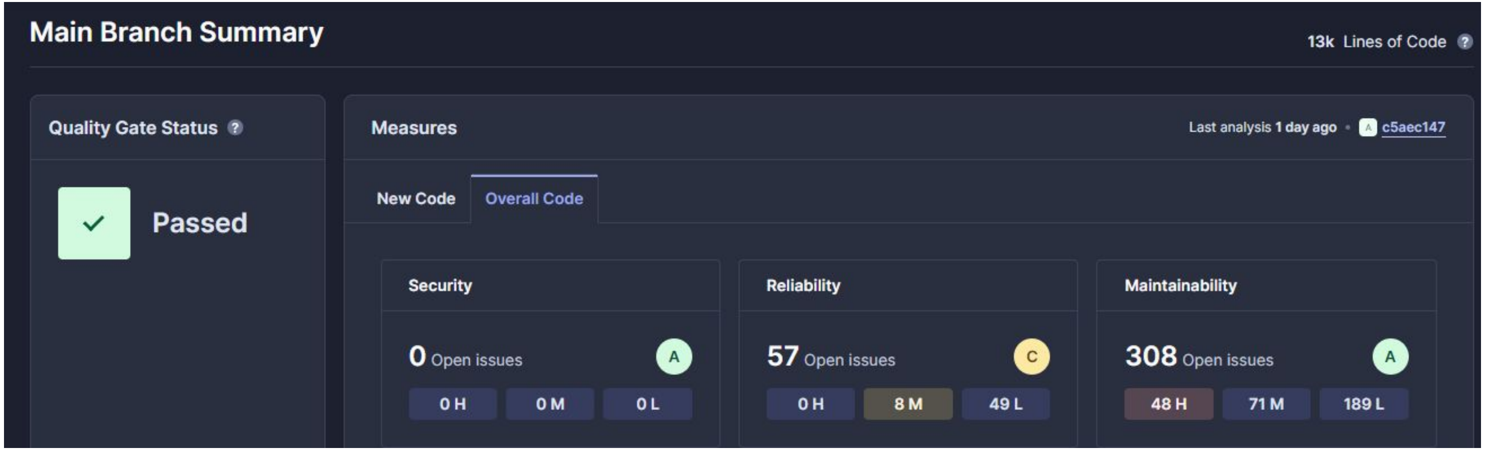
- Testing with Jest
- Automated with Github Actions
- Testing evolved from Sprint 1 to Sprint 5 to achieve 80% coverage in unit tests.



```
=============================== Coverage summary ===============================
Statements   : 85.08% ( 445/523 )
Branches     : 81.81% ( 144/176 )
Functions    : 76.52% ( 88/115 )
Lines        : 85.06% ( 433/509 )
================================================================================

Test Suites: 24 passed, 24 total
Tests:       76 passed, 76 total
Snapshots:   0 total
Time:        29.485 s
Ran all test suites.
```

Testing and QA



Main Branch Summary

13k Lines of Code

Quality Gate Status

Passed

Measures

Last analysis 1 day ago • A c5aec147

Security

0 Open issues A

0 H 0 M 0 L

Reliability

57 Open issues C

0 H 8 M 49 L

Maintainability

308 Open issues A

48 H 71 M 189 L

Quality Metrics with SonarCloud

What we learned

- Effective communication during synchronous and asynchronous activities
  - Communication improved throughout the duration of the project
  - Weekly meetings
  - Meeting minutes
- Task progression tracking via Github issues
- The importance to start a project with a proper documentation; models, diagrams, mockups

What we could improve on

- Follow a strict code convention
  - Perhaps implement a linter
- Use of more header comments
  - As the project grew, it became confusing to keep track of which files did what
- Task tracking tool

Conclusion

- Delivered a comprehensive condo management system across both web and mobile platforms, encompassing essential functionalities.
- Enhanced team skills in agile development, responsive web design, and mobile application development.
- Successfully addressed challenges related to system integration, cross-platform functionality, and data security.
- Acknowledgments: Thanks to Professor, Teaching Assistant and team members.

Future Plans

- Expand to iOS and Linux to increase accessibility.
- Implement AI analytics for smarter property and financial management.
- Enhance security with advanced encryption and regular audits.
- Prepare for international markets by localizing the app in multiple languages.
- Integrate with smart home technologies to improve property management.