

CS335 Group Project 2025

Group: 32

Topic: Real Estate & Property Management

Group Members: Seamus Mullan (23434744),
Kevin Walsh (23447842), Sean Conlon
(23373296)

Work completed by group members:

- **Seamus Mullan** (23434744)
 - o Report Writing and Editing
 - o Overview
 - o 4 User Stories
 - o Models, Descriptions and UML Diagrams
 - o User Interface Mock-Up
- **Kevin Walsh** (23447842)
 - o 3 User Stories
 - o Functional Requirements
 - o Non-Functional Requirements
- **Sean Conlon** (23373296)
 - o 3 User Stories
 - o System Tests
 - o Navigation Flows

Should you have any issue reading the contained information, the original docx, drawio, svg, png, pptx and pdf files are available at https://github.com/SeamusMullan/CS335_Group_Project

Overview

Breakdown of requirements

This project requires us to complete the following with relation to our topic:

- | | |
|-------------------------------|---------------------|
| - 10 User Stories | - UI Mock-up |
| - Functional Requirements | - System Tests |
| - Non-Functional Requirements | - Navigations Flows |
| - UML Models / Diagrams | |

Final project choice

We decided to create a report on a mobile app for tenants, landlords, maintenance staff and other relevant individuals to simplify management of properties, paying and monitoring bills / payments, and the completion of works on premises owned by landlords or property managers.

Modern property management is slowly skewing towards a more technical process compared to the paper based systems in the past. Landlords, Lawyers, Property Managers, Accountants and Contractors are using digital forms of documentation more and more, providing us with a solid reason to create a software system to manage and combine all this information.

Our decision to create a mobile application was fuelled by the fact that almost every person has a mobile device like a phone or tablet in 2025. This combined with the rapid digitalization of documentation, payments and site management reinforced our reasoning and solidify our idea.

The following pages contain our planned design and implementation for this application, with careful attention to details and a focus on our project's topic.

Part 1: Software Requirements

User Stories

US01 - Landlord / Owner

Description

- **As a:** Landlord
- **I want to:**
 - Manage Rent Payments
 - View all Tenants
 - View Property Information
 - Know value of property at a glance
 - Manage all properties (Update / Remove)
 - View Tenant Requests / Issue
- **So that:**
 - Be more efficient in work
 - Make work easier
 - View and make changes quickly

Acceptance Criteria

- ☐ **Criterion 1:** View Property Data incl. Rent payments, Tenants, etc.
- ☐ **Criterion 2:** Update above mentioned data
- ☐ **Criterion 3:** Communicate with relevant people

Tasks

- ☐ **Task 1:** Check if Tenant A has paid rent this month
- ☐ **Task 2:** Check if issues have been resolved
- ☐ **Task 3:** Sign in to relevant accounts

Test Cases

1. **Test Case 1:**
 - **Input/Test Data:** Username and Password
 - **Expected Output:** Log in to landlord page
2. **Test Case 2:**
 - **Input/Test Data:** Access / Read Tenants & Property Information
 - **Expected Output:** User Should be able to read relevant info

US02 - Property Manager

Description

- **As a:** Property Manager
- **I want to:**
 - See all the properties I manage
 - See who lives in each property
 - Access and CRUD documents relating to their properties
 - See who the landlord is, contact them
- **So that:**
 - They can manage multiple properties quickly
 - See all the documents they need in once place
 - They can contact people in their buildings

Acceptance Criteria

- ☐ **Criterion 1:** Can Log In and access all the documents for every property
- ☐ **Criterion 2:** Can contact all the landlords, tenants & others in their building
- ☐ **Criterion 3:** Can Add and remove properties, update their information, etc.

Tasks

- ☐ **Task 1:** Keep Documents for a property up to date
- ☐ **Task 2:** Contact landlord and tenants
- ☐ **Task 3:** Edit documents for owned properties

Test Cases

3. **Test Case 1:**
 - **Input/Test Data:** Log In
 - **Expected Output:** Access to Admin Panel with property documents
4. **Test Case 2:**
 - **Input/Test Data:** Contact the Landlord using messaging system
 - **Expected Output:** Landlord should receive message from Property Manager

US03 - Accountant

Description

- **As a:** Accountant
- **I want to:**
 - Examine Accounting records
 - Track rent payments
 - Know how money is being spent on each property
 - communicate with Property manager
- **So that:**
 - Give advice to Property manager
 - Ensure all financial records are correct
 - Ensure compliance with tax laws

Acceptance Criteria

- ☐ **Criterion 1:** View financial records about properties
- ☐ **Criterion 2:** Communicate with Landlord and property manager
- ☐ **Criterion 3:** Update financial records

Tasks

- ☐ **Task 1:** Inform property manager of potential financial risks
- ☐ **Task 2:** compare income and expenditure of properties
- ☐ **Task 3:** Have access to all relevant data

Test Cases

5. **Test Case 1:**
 - **Input/Test Data:** **Username and password
 - **Expected Output:** Log in to an account with access to accounting information
6. **Test Case 2:**
 - **Input/Test Data:** Change financial report
 - **Expected Output:** See new edited report

US04 - Admin Assistant

Description

- **As an:** Admin Assistant
- **I want to:**
 - View information on relevant properties
 - Establish contact with Tenants / other staff
 - Respond to any queries the Tenants may have
 - See which Tenants have paid rent
- **So that:**
 - Effectively manage Tenants
 - Assist Landlords in managing Tenants

Acceptance Criteria

- ☐ **Criterion 1:** View property / Tenant details
- ☐ **Criterion 2:** Communicate with relevant people
- ☐ **Criterion 3:** Update property details

Tasks

- ☐ **Task 1:** Respond to Tenant queries / concerns
- ☐ **Task 2:** Follow up on tenants who did not pay rent
- ☐ **Task 3:** Sign into relevant account

Test Cases

7. **Test Case 1:**
 - **Input/Test Data:** Username and password
 - **Expected Output:** Log into Admin Assistant account
8. **Test Case 2:**
 - **Input/Test Data:** View information on relevant properties
 - **Expected Output:** User should be able to read relevant information

US05 - Accountant

Description

- **As an:** Accountant
- **I want to:** Manage and access relevant documents for all the properties my landlord / property manager owns
- **So that:** I can provide accurate financial information regarding the properties

Acceptance Criteria

- ☐ **Criterion 1:** Ability to see all documentation regarding property, owners, and tenants
- ☐ **Criterion 2:** Able to contact tenants and other important people for information
- ☐ **Criterion 3:** Able to export documents to process using other software (Excel)

Tasks

- ☐ **Task 1:** Download valuation documents for all properties
- ☐ **Task 2:** Contact other people involved in property
- ☐ **Task 3:** Make requests for more docs or to update documents

Test Cases

9. **Test Case 1:**
 - **Input/Test Data:** Contact person
 - **Expected Output:** Person receives message
10. **Test Case 2:**
 - **Input/Test Data:** Download all documents for a property
 - **Expected Output:** Documents are downloaded

US06 - Admin Assistant

Description

- **As a:** Admin Assistant
- **I want to:**
 - review how the app functions
 - conduct tests within the app
 - Communicate with users
- **So that:**
 - Help users with small issues
 - Ensure everything is working correctly
 - Get feedback from users
 - provide updates to fix errors within the app

Acceptance Criteria

- ☐ **Criterion 1:** Update the app
- ☐ **Criterion 2:** Review how the app handles payments
- ☐ **Criterion 3:** *be able to communicate with users*

Tasks

- ☐ **Task 1:** Review payment between tenant and landlord
- ☐ **Task 2:** Provide a fix to a previous issue
- ☐ **Task 3:** Help user with login issues

Test Cases

11. **Test Case 1:**
 - **Input/Test Data:** Login details
 - **Expected Output:** Admin panel which has access to all app features
12. **Test Case 2:**
 - **Input/Test Data:** Create a change in the app
 - **Expected Output:** The app should be updated for all users

US07 - Maintenance Staff

Description

- **As a:** Maintenance staff
- **I want to:**
 - Quickly fix any issues Tenants may have
 - Contact Landlord to provide external parts if required
 - Have a way to sort issues based on priority
 - Be able to mark issues as solved
 - Be able to bill landlords/tenants for labour and materials
- **So that:**
 - Respond to tenant issues in a timely fashion
 - Give support to Tenants based on importance

Acceptance Criteria

- ☐ **Criterion 1:** View and sort Tenant complaints
- ☐ **Criterion 2:** Update status of complaints (show if resolved / give an estimate as to when staff will be in to resolve issue)
- ☐ **Criterion 3:** Be paid by appropriate people for labour and materials

Tasks

- ☐ **Task 1:** View Tenant requests and sort by importance
- ☐ **Task 2:** Mark issues as resolved
- ☐ **Task 3:** Log into relevant accounts

Test Cases

13. **Test Case 1:**
 - **Input/Test Data:** Username / password
 - **Expected Output:** Log into maintenance staff account
14. **Test Case 2:**
 - **Input/Test Data:** Mark issue as resolved
 - **Expected Output:** Landlord can see that an issue is resolved

US08 - Contractor / Service Provider

Description

- **As a:** Contractor / Service Provider
- **I want to:**
 - Gain access to the buildings plans and information to carry out works
 - Request time slots to do works in the properties
 - Contact the Landlord and Property Manager in the event that more issues arise
 - Get Payment from Accountant / Property Manager
- **So that:**
 - My works can be completed quickly and effectively
 - I can get paid for my work
 - The tenants can live in a safe and suitable home
 - The Landlord / Property Manager can maintain their properties

Acceptance Criteria

- ☐ **Criterion 1:** Access to the building documents (Plans, Door Codes, Etc.)
- ☐ **Criterion 2:** Ability to request time for works or repairs
- ☐ **Criterion 3:** Access to means of payment

Tasks

- ☐ **Task 1:** Request time for works
- ☐ **Task 2:** Request / Manage Payments
- ☐ **Task 3:** Contact Landlord / Property Manager

Test Cases

15. **Test Case 1:**
 - **Input/Test Data:** Attempt to Download Pay slip
 - **Expected Output:** Relevant Pay slip for period of works downloaded.
16. **Test Case 2:**
 - **Input/Test Data:** Contact LL / PM to request works / inform of issue
 - **Expected Output:** Landlord / Property Manager receive message

US09 - Real Estate Agent

Description

- **As a:** Real Estate Agent
- **I want to:**
 - View Property details
 - Create property listings
 - Communicate with property owners and landlords
- **So that:**
 - Find the right property for clients
 - Find potential buyers for properties
 - Talk with property owners and landlords about specific properties
 - Connect new tenants with landlords

Acceptance Criteria

- ☐ **Criterion 1:** View Property information
- ☐ **Criterion 2:** Create property listings
- ☐ **Criterion 3:** Contact property owners

Tasks

- ☐ **Task 1:** Create a new property listing
- ☐ **Task 2:** Talk with a property owner about their property details
- ☐ **Task 3:** Search for a specific type of property for a client

Test Cases

17. **Test Case 1:**
- **Input/Test Data:** User Account login
 - **Expected Output:** A marketplace panel where the user can search for and view property details as well as being able to message users
18. **Test Case 2:**
- **Input/Test Data:** Create a new property listing
 - **Expected Output:** A new listing should show for other users

US10 - Tenant

Description

- **As a:** Tenant
- **I want to:**
 - Be able to pay rent
 - Check if rent has already been paid for the month
 - View information on the property (e.g. address, postcode, apartment number)
 - Report issue with the property
 - Direct line of communication with relevant people
- **So that:**
 - Make any issues with the property easier
 - Effectively communicate with landlord

Acceptance Criteria

- ☐ **Criterion 1:** Easily able to pay rent
- ☐ **Criterion 2:** Communicate with relevant people
- ☐ **Criterion 3:** Report issues with the property

Tasks

- ☐ **Task 1:** Pay rent and see if rent has been paid
- ☐ **Task 2:** Able to make complaints about property
- ☐ **Task 3:** Sign into relevant accounts

Test Cases

19. **Test Case 1:**
- **Input/Test Data:** Username and Password
 - **Expected Output:** Log into Tenant page
20. **Test Case 2:**
- **Input/Test Data:** View property information
 - **Expected Output:** User can read property information
21. **Test Case 3:**
- **Input/Test Data:** Report an issue with the property
 - **Expected Output:** Landlord / maintenance staff can see and respond to issue

Functional Requirements

When beginning to create the functional requirements I started by reviewing the user stories that the team had written. Specifically I identified the most common points described by the acceptance criteria for each type of user and focused on the most important requirements to satisfy as many of these criteria as possible. I then made a short list of the most important features the app would need to function which included user logins, communications and property data. From here I worked to complete descriptions of these initial requirements and Identified the related user stories for each.

After these basic requirements I looked to some of the first diagrams to identify other important requirements such as payments and work scheduling. At this point it was important to begin identifying some non-functional requirements for the system to run smoothly. I reviewed the functional requirements that I had already completed and tried to match at least one non-functional requirement to match each functional requirement. For example a functional requirement such as a user login would also need a non-functional security requirement in order to protect user personal data and passwords.

Throughout the duration of the project I continued to iterate upon these requirements to make sure I added any that were previously overlooked as well as updating the initial requirements to match the project direction. This was very important to do as I needed to ensure each requirement was still relevant to the project. Lastly I reviewed the UI Mock-ups to ensure each requirement was correctly shown and also to add any further requirements that I had missed earlier.

FR01 - User Login

Description:

Allow users to access to their specific account by entering their login credentials.

Related User Stories:

All

FR02 - Property Data

Description:

Property Data should be available to the relevant users. Landlords and Property owners should be able to view/update data for their properties. Accountants should have access to financial data. Tenants should also see relevant info about their homes.

Related User Stories:

US01 - Landlord / Owner
US02 - Property Manager
US03 - Accountant
US10 – Tenant

FR03 - Direct Messages

Description:

Users should have a way to communicate with each other from within the app through a dedicated messaging feature

Related User Stories:

All

FR04 - Tenant issues

Description:

Allow tenants to raise issues with specific properties notifying landlords and maintenance staff who can interact with this issue and track it from within the app until it is fixed.

Related User Stories:

US10 - Tenant
US01 - Landlord / Owner
US07 - Maintenance Staff

FR05 - Work scheduling

Description:

Ability for landlords to allocate time for work/maintenance to take place within a property. Maintenance staff and contractors should be able to view this too.

Related User Stories:

US01 - Landlord / Owner
US07 - Maintenance Staff
US08 - Contractor / Service Provider

FR06 - Payments

Description:

Users should be able to have a balance on the app and be able to send money to the relevant users. Tenants should be able to pay rent, Landlords should be able to pay maintenance staff and contractors.

Related User Stories:

US01 - Landlord / Owner
US07 - Maintenance Staff
US08 - Contractor / Service Provider
US10 - Tenant

Non-Functional Requirements

NFR01 - Login

Type: Security

Description:

Handling of user credentials must be done correctly

Measurement/Criteria:

Ensure data is being handled in accordance with local regulations and company policy

NFR02- Property Data

Type: Performance

Description:

Property data must be stored efficiently to allow the app to function properly with a large amount of individual properties

Measurement/Criteria:

Testing app performance with varying amounts of property data.

NFR03 - Direct messages

Type: Performance

Description:

Sent messages must be received as fast as possible to allow for quick communication between users

Measurement/Criteria:

Compare time between sent and received messages

NFR04 - Payments

Type: Security

Description:

Payments between users need to be processed safely and securely and to guarantee safety for the users.

Measurement/Criteria:

Payment features should be ensure compliance with regulations

NFR05 - Work Scheduling

Type: Usability

Description:

Maintenance staff and contractors will be interacting with the work scheduling features frequently. The user interface should be designed in clear way for these features should be simplified to accommodate for all types of people using these features

Measurement/Criteria:

Test app with real world users

NFR06 - Tenant Issues

Type: Usability

Description:

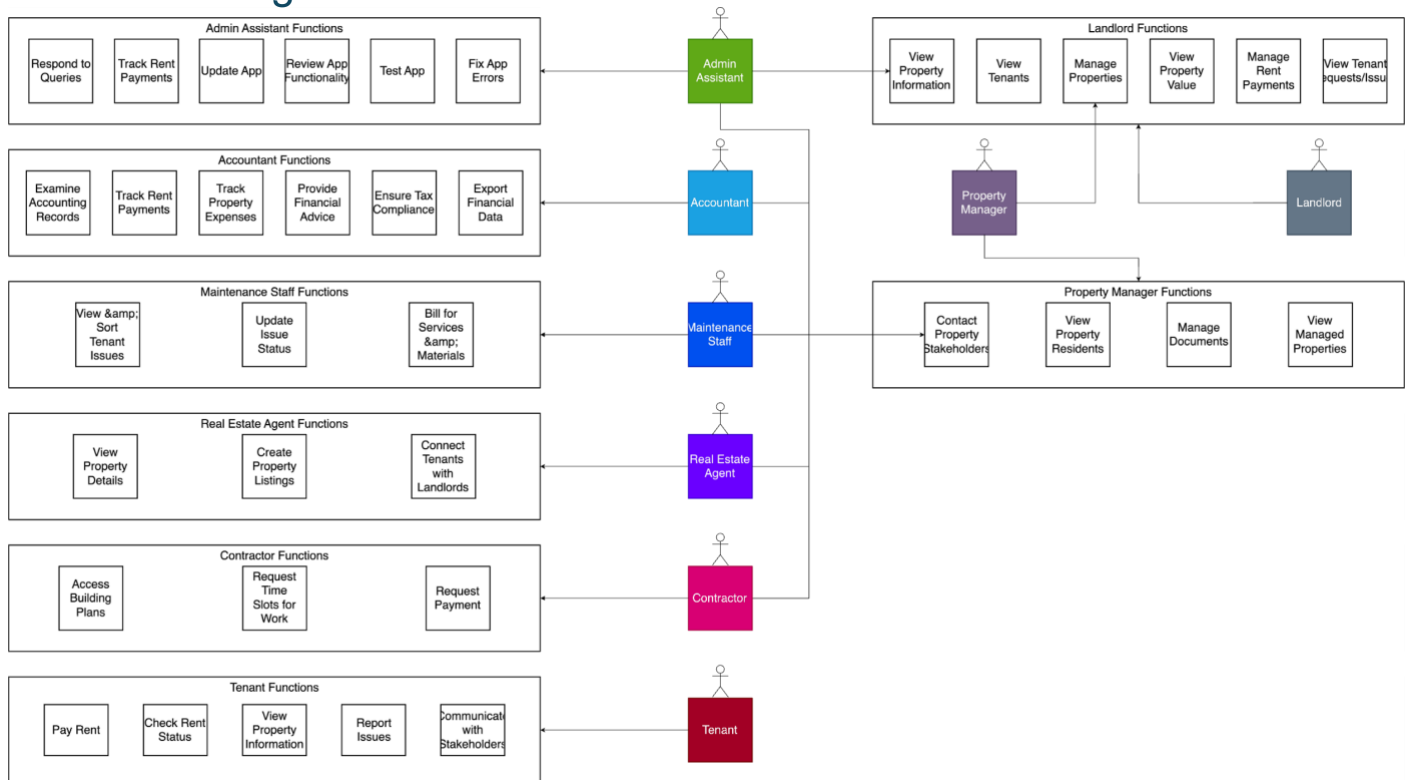
It should be easy for tenants to report issues with a property to the relevant parties. It should be easy and intuitive for a user to find the buttons to report issues to landlords.

Measurement/Criteria:

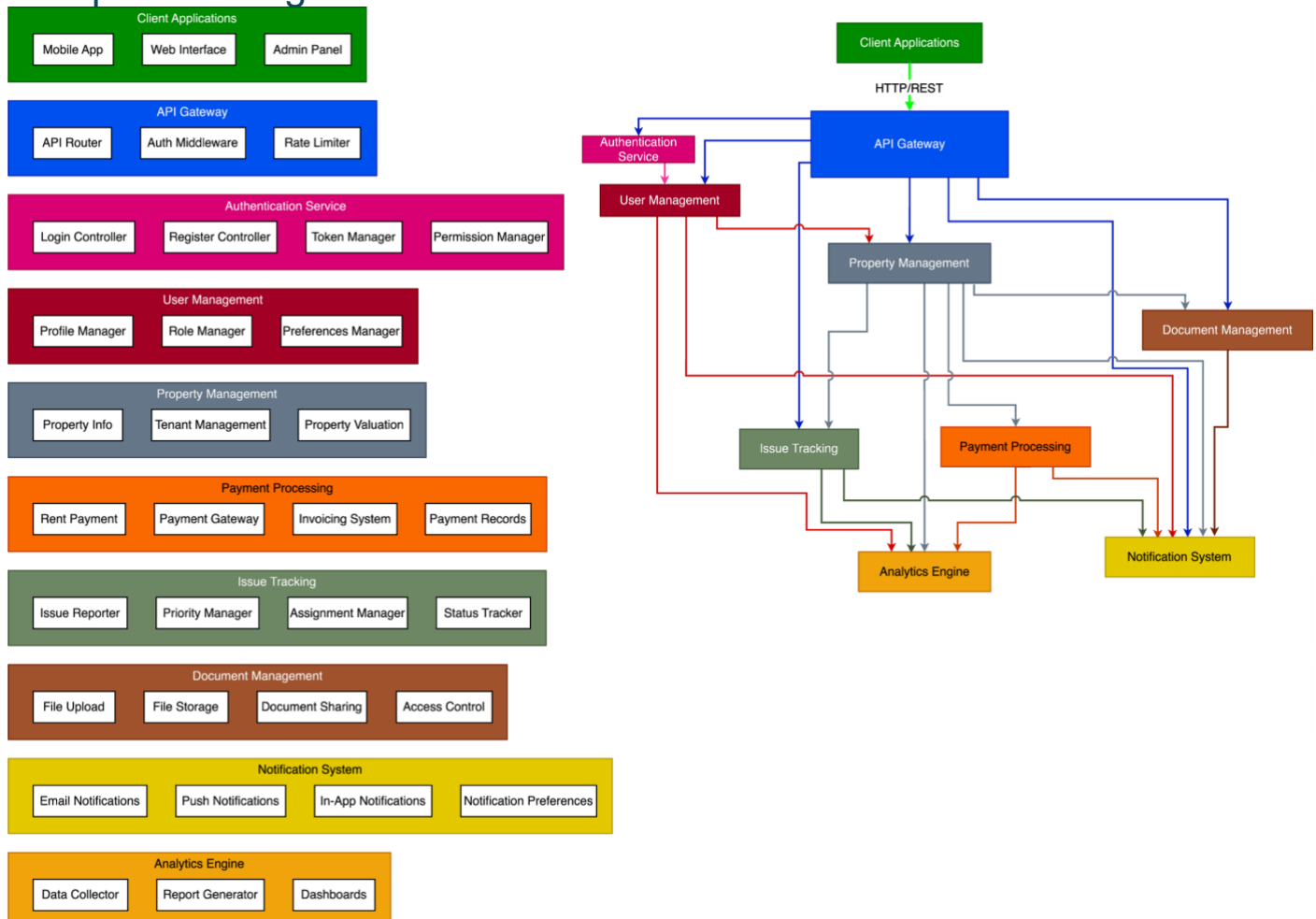
Test app with many types of tenants

Part 2: Models, Descriptions and UML Diagrams

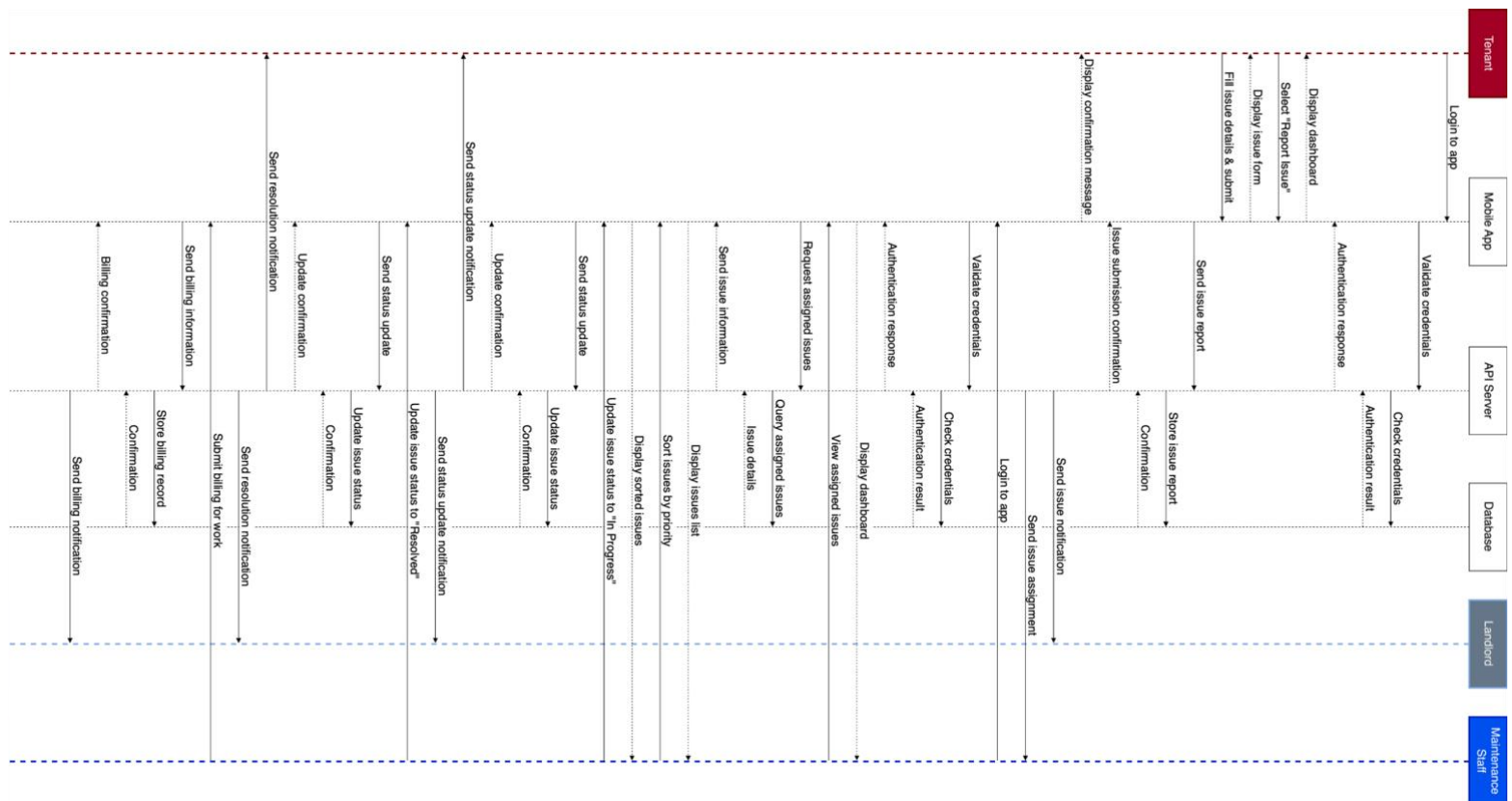
Use Case Diagram



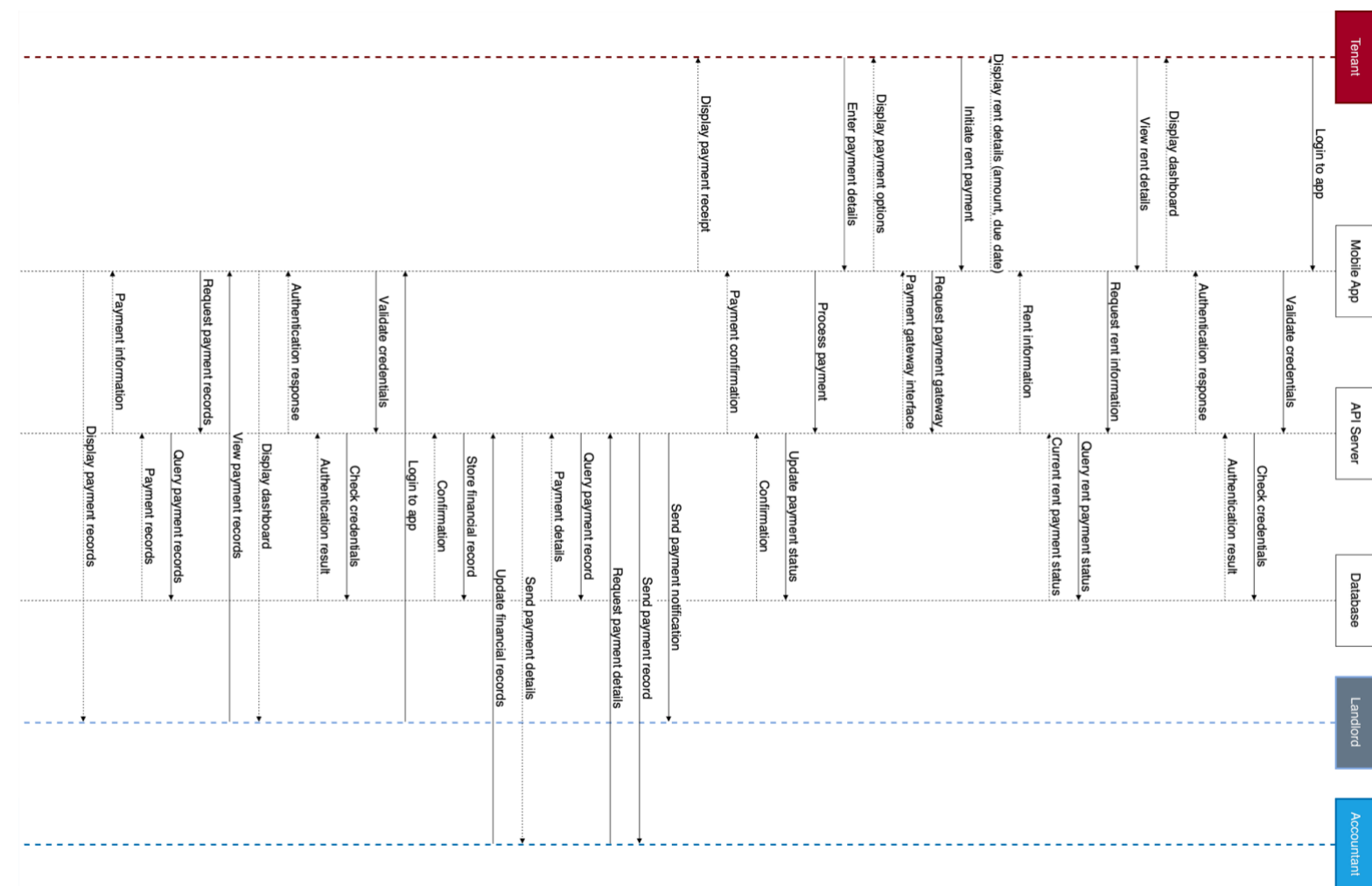
Component Diagram



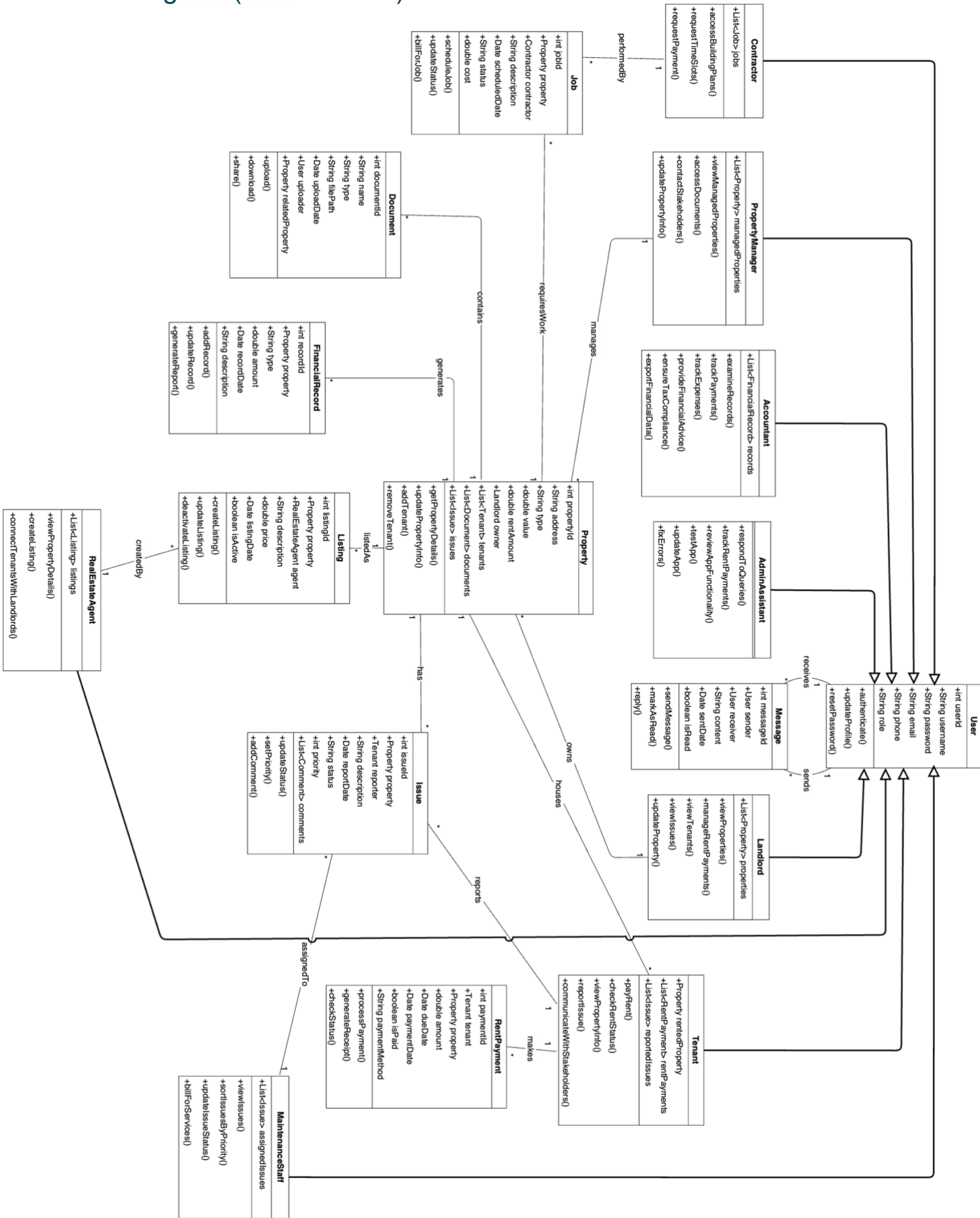
Sequence Diagram - Issue Reporting (Rotated to fit)



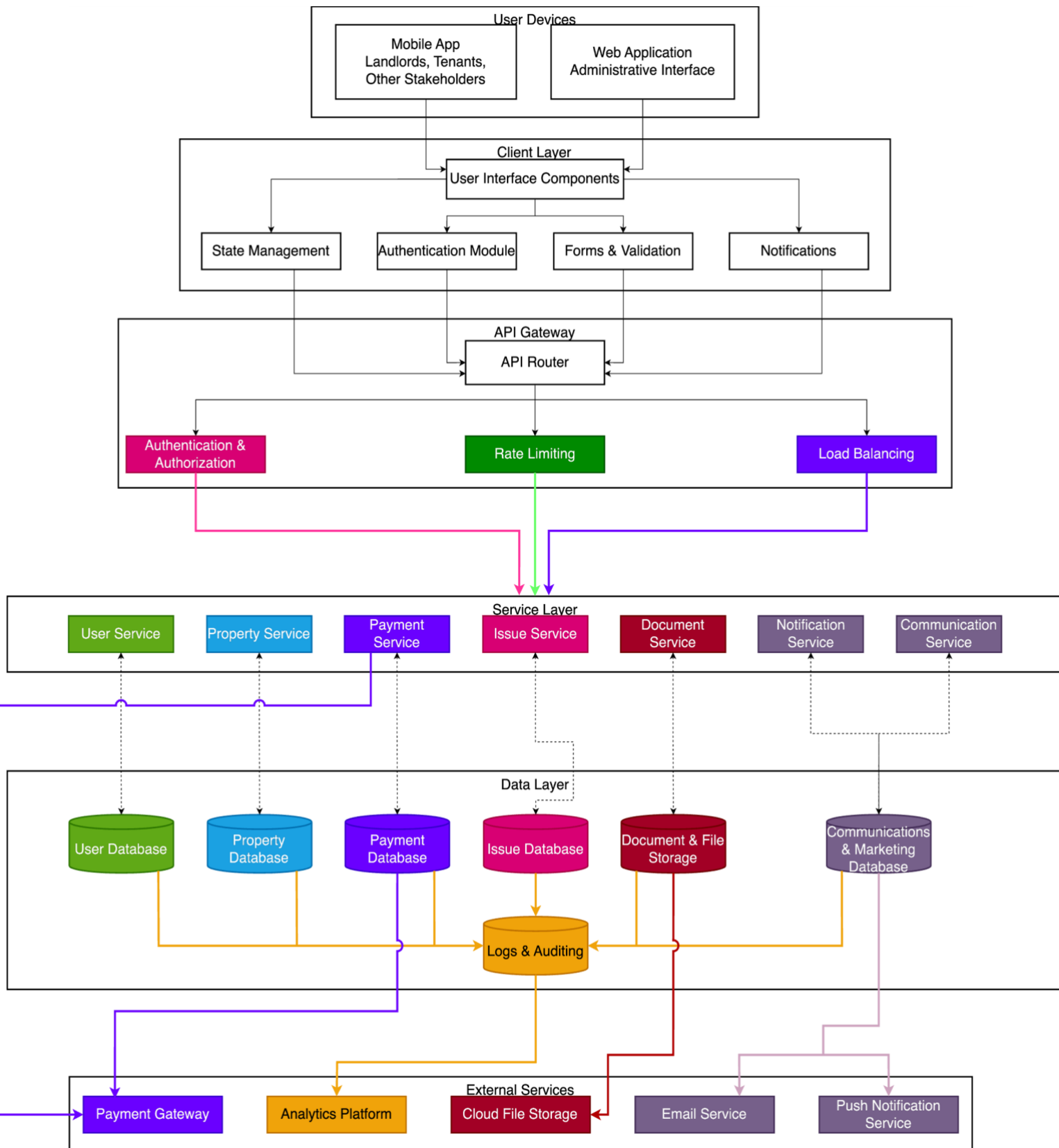
Sequence Diagram - Renting (Rotated to fit)



Class Diagram (Rotated to fit)



System Architecture Diagram

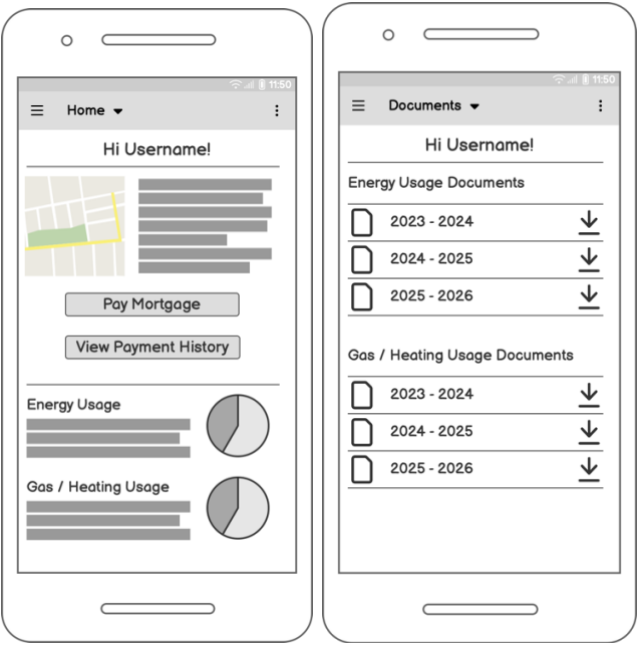


Part 3: User Interface Mock-up

General Pages



Tenant Pages



Landlord Pages



Maintenance Staff Pages



Part 4: System Tests and User Requirements

System Tests

System Tests

The System tests given here are based on the user stories and the functional / nonfunctional requirement. Tests contain their objective, users that they apply to, steps to completion and expected result.

Test 1 (All users):

Objective:

To verify the functionality of the sign up system.

Test Steps:

1. User selects their role (e.g. Landlord, Tenant etc.)
2. User keys in their username.
3. User keys in their password twice.

Expected Output:

The user should enter into the panel that they selected at startup. Their username and password is saved in the database and they can log in again at a later stage with these details. If the conditions are not met, then an error message should be displayed that informs the user what is incorrect. User is then told to try again.

Test 2 (All users):

Objective:

To verify the correct functionality of the log in system.

Test steps:

1. User keys in their username.
2. User keys in their password.
3. User selects "Log In".

Expected output:

User logs in to their account. They gain access to the panel that is relevant to their account(e.g. Landlords get their landlord specific UI). If the user enters an incorrect username or password, then the user should be denied access to their account. The user should then see text informing them that either the username or password was incorrect and prompted to sign up or try again.

Test 3 (All users):

Objective:

Ensure correct functionality of the chat feature.

Test steps

1. Select username of account you want to chat with
2. Select "start a chat".
3. Enter a message to send.
4. Select send message.

Output:

A chat is started between the two users where both can send and receive messages. Every user should have access to this chat feature and be able to chat with any other user.

Test 4(All users):

Objective:

To view information about a property.

Test Steps:

1. User selects the "View Property information" on the relevant property.

Expected Output:

Property information such as address, landlords contact information, availability is displayed. For the Landlord and accountants only, a property valuation should also appear. For the landlord, Maintenance workers and the building contractors / service providers only, property building plans should be available to download / view.

Test 5 (Tenant, Landlord, Maintenance staff):

Objective:

To ensure correct functionality of Tenant reports. Tenants should be able to report issues and Landlords should be able to view reports made by the user. If action is necessary, maintenance staff should also be able to view the report.

Test Steps:

1. User selects "report an issue".
2. User briefly explains their issue with the property.
3. User selects "submit".

Expected Output:

Tenants' issue is logged in the database and the Landlord receives it in their inbox. The receiver can read it in its entirety as it was written by the tenant. If relevant to the Maintenance staff, they will also receive it.

Test 6 (Tenants, Landlord, Maintenance staff):

Objective:

Allow users to pay relevant people for fees due. Examples of fees a tenant might pay are rent, utilities, damages, deposits, goods or services.

Test steps:

1. User selects "payments due".
2. They select a fee they have to pay.
3. Input card information.
4. Select "pay".

Expected output

Relevant user should be paid and the database should be updated to reflect this change. The payment should be removed from the payments due page. Both users receive a confirmation of payment.

Test 7 (Tenants, Landlord, Maintenance staff):

Objective:

Request payment from other users

Test steps:

1. Select "request payment".
2. Input the username of the person you are requesting payment from.
3. Input the amount you are owed.
4. Give a tag to the request outlining what it is for.(e.g. If requesting rent money, tag it as "rent").
5. Select request.

Expected output:

The relevant user should receive the billing request on their account and have the option to pay the money.

Test 8 (Landlords):

Objective:

Allow the landlord to update property information if needed.

Test steps:

1. Select the relevant property.
2. Select "edit details".
3. User keys in new any changes that need to be made

Expected output:

User receives a confirmation that the details have been changed. The changes will be updated on the database and displayed to any user viewing the property details.

Test 9 (Landlords):

Objective:

Allow landlords to add new properties to their account.

Test steps:

Navigation Flows (Scaled to fit)



1. User selects "add new property".
2. User inputs relevant documentation and information about property.

Output:

The landlord's new property is added to their account and they receive a confirmation notification. This property is then added to the database.

Test 10 (Accountants):

Test Objective:

View payment / billing record of a User

Test Steps:

1. Select the account of the relevant User.
2. Select "View payment records"

Expected Results/Output:

Payment records should be shown to the user.

Test 11 (Landlord, Building contractors / service providers, Maintenance staff):

Objective:

Allow users to place tags on reported issues. This is for the purpose of notifying others as to the current status of the issue.

Test Steps:

1. User selects the relevant issue.
2. User selects the current tag (default is "not started")
3. User selects any different tag.

Expected output:

The tag on the issue is changed on the database and updated for all users involved.