

Sprint (3) Release Plan

User story 7 – Condo Manager Upload File for Each Property: 9 USP

- Update the property schema in MongoDB to include a field for storing file references
- Develop a Django API endpoint for file uploads
- Integrate a file upload component in the React frontend within the property profile management page.
- Modify the React component to display a list of uploaded files for each property.
- Write tests to verify that files are correctly uploaded and associated with the property profiles.

User Story 10 – Implement Condo Fee Rate Entry: 6 USP

- Develop a feature in the financial module to allow entry of condo fee rates per square foot and per parking spot.
- Ensure the data model in MongoDB supports storing these rates associated with each property.
- Create a Django API endpoint to handle the submission of condo fee rates.
- Design a React component for condo management companies to enter and update these rates.
- Implement validation on the input to ensure the rates are positive numbers.

User Story 11 – Calculate and Present Condo Fees: 9 USP

- Implement a calculation engine in Django that automatically computes the condo fee for each unit based on its size, the number of parking spots, and the entered rates.
- Update the unit detail view in React to present the calculated condo fees to unit owners.
- Ensure the calculation takes place every time there is a change in unit size, parking spot allocation, or fee rates.
- Develop a Django RESTful API endpoint that provides the calculated condo fees on request.
- Write unit tests to ensure accurate calculation of condo fees under various scenarios.

User Story 12 – Record Operational Budget and Costs: 12 USP

- Expand the financial module to include functionality for recording operational budgets and costs.
- Design and implement a MongoDB schema to store operational budgets, collected condo fees, and specific costs for activities and maintenance operations.
- Create Django RESTful API endpoints for entering and updating these financial details.
- Develop a React interface for condo management companies to input and manage operational budgets and costs.
- Ensure data integrity with validation checks for budget and cost entries.

User Story 13 – Generate Annual Financial Reports: 9 USP

- Develop a reporting tool within the financial system to generate annual financial reports.
- The tool should aggregate collected condo fees and operational costs for each property.
- Design a Django RESTful API endpoint to trigger report generation and retrieval.
- Create a React component for condo management companies to generate and view these reports.
- Ensure the report includes detailed financial insights and is formatted for clarity and professionalism.

User Story ID	User Story Points (USP)	Priority	Status
US7 (#22)	9 USP	● Medium	● Backlog
US10 (#26)	6 USP	● Medium	● Backlog
US11(#27)	9 USP	● Low	● Backlog
US12(#28)	12 USP	● Medium	● Backlog
US13(#29)	9 USP	● Low	● Backlog
Total USP	45 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	Not Started
File Upload	Patrick	2/2	3/3	3 days	Not Started
Condo Fee Rate Entry	Yashi	3/3	3/7	4 days	Not Started
Calculate Condo Fees	Eden	3/7	3/10	3 days	Not Started
Record Operational Budgets	Minh	3/8	3/16	8 days	Not Started
Financial Reports	Joud	3/16	3/21	5 days	Not Started

