CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT317-G1
PROJECT MANAGEMENT

SPRINT PLANNING REPORT

Project Title: RentMate

Prepared By: Gregory Ivan Onyx M. Badinas / RentMate / Scrum Master

Sprint Duration: 3

Sprint Duration: 10/19/2025 - 11/02/2025

Date of Submission: 11/30/2025

Table of Contents

1. Sprint Goal

Establish the core operational workflows of RentMate by implementing the Landlord Maintenance Management system, enabling Tenant Payment Submission and Verification, and deploying a Dynamic Tenant Dashboard for real-time status tracking.

2. Sprint Backlog Items

1. Create a secure Django view to retrieve and display all maintenance requests from the Supabase database.
2. Design and build the landlord's "Maintenance" page to display a list of all requests with their details.
3. Develop a Django view and UI controls that allow landlords to update the status of a maintenance request (e.g., Pending, In Progress, Completed).
4. Ensure the tenant's view of their request history is updated in real-time to reflect the new status set by the landlord.
5. Create the database model and a Django view to handle tenant payment submissions with proof of payment.
6. Develop the "Submit Payment" form for tenants using Django Forms, and implement server-side input validations.
7. Connect the payment form on the frontend to the Django view to ensure data storage and display success/failure messages.
8. Create a secure Django view to allow landlords to view the history of all tenant payment submissions.
9. Build the "Payment Verification" page for landlords to see a list of submitted payments with their details.
10. Develop a Django view and UI controls (Approve/Disapprove buttons) for landlords to update a payment's status.
11. Create a secure Django view to retrieve the logged-in tenant's data (lease end date, rent, payment status, open maintenance requests count).
12. Create the HTML structure and CSS for the main dashboard layout and widget containers.
13. Integrate Django template tags to dynamically display data passed from the view.

14. Implement logic to handle cases where data might not be available (e.g., zero maintenance requests).

3. Capacity Planning

Team: 3 Developers

Estimated Total Capacity: 105 Hours

Sprint Story Points: 36

4. Acceptance Criteria

1. Landlords can view all tenant maintenance requests and update their status (Pending/In Progress/Completed).
2. Status updates made by the Landlord are immediately reflected in the Tenant's view.
3. Tenants can access the "Proof of Payment" page, submit a form, and receive a success message.
4. Landlords can view submitted proofs and update the status to "Approved" or "Disapproved".
5. The Tenant Dashboard loads within 2 seconds and accurately displays the count of pending maintenance requests, current rent amount, lease end date, and payment status.
6. Unauthorized access (e.g., Tenants accessing Landlord views) is blocked and returns an error.

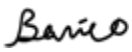
5. Risks & Dependencies

1. Potential delays or inconsistencies between Landlord updates and Tenant views if real-time logic fails.
2. Aggregating data from multiple tables for the Dynamic Dashboard may impact page load performance.

6. Team

Name	Role
Joseph James T. Banico	Lead Developer
Junpaul P. Arradaza	Developer
Julius C. Bargamento	Developer
Zander N. Aligato	Product Owner
Mechole Angelou M. Auditor	Business Analyst
Shane Nathan B. Archival	Business Analyst
Gregory Ivan Onyx M. Badinas	Scrum Master

7. Approval Sign-off

	Full Name	Signature	Date
Prepared By:	Gregory Ivan Onyx M. Badinas		11/29/2025
Developer	Junpaul P. Arradaza		11/29/2025
Developer	Julius C. Bargamento		11/29/2025
Lead Developer	Joseph James T. Banico		11/29/2025
Reviewed By:	Mechole Angelou M. Auditor		11/29/2025

Reviewed By:	Shane Nathan B. Archival		11/29/2025
Reviewed By:	Zander N. Aligato		11/29/2025
Approved By:	Mr. Joemarie C. Amparo		
Approved By:	Mr. Frederick Revilleza		