

Software Requirement Specification

Url: <https://github.com/Abelkelemu/SWEProject.git>

Introduction

The Rental Property Management System (RPMS) aims to provide a comprehensive software solution to streamline and optimize the process of managing house rentals for property owners and tenants. This document outlines the use cases and requirements for the RPMS, which will automate key tasks such as property listing, tenant screening, and communication between stakeholders.

Use Case:



RPMS: Use Case Description

Use Case Number: 1
Name User CRUD
Brief description This use case allows new users to create an account and sign up for the Rental Property Management System.
Actors User (Tenant/Property Owner)

Preconditions		
None		
Flows of Events:		
1. Basic Flows		
1.1.0 Sign Up		
Step	User Actions	System Actions
1	The new user accesses the system's sign-up page.	The system presents a registration form with fields for the user's name, email, password, and other necessary details.
2	The user fills out the registration form with the required information. And the user submits the form and requests the system to create the account.	The system validates the user's information, checks for duplicate accounts, and creates a new user profile. The system sends a verification email to the user's provided email address.
3.	The user verifies the account by clicking on the verification link in the email.	The system confirms the successful account verification and informs the user of successful registration.
Postconditions		
The user's account is successfully created, verified, and ready for use.		
Business Rules		
Each user must have a unique email address for registration..		
1.1.1 Sign In		
Step	User Actions	System Actions

1	The registered user accesses the system's sign-in page.	The system presents a login form with fields for the user's email and password.
2	The user enters their email and password into the login form. The user submits the form and requests the system to authenticate the credentials.	The system verifies the user's email and password combination. If the credentials are valid, the system logs the user in and redirects them to the user dashboard. If the credentials are invalid, the system notifies the user of the authentication failure and prompts them to re-enter the correct credentials.
Postconditions		
The user is successfully signed in and can access their account features.		
Business Rule		
The user must provide the correct email and password combination to sign in.		

Use Case Number: 2
Name Property CRUD
Brief description Property owners can create and manage listings for their rental properties.
Actors Property Owner
Preconditions
The property owner must be logged in to the system
Flows of Events:

1. Basic Flows		
1.0 Add property		
Step	User Actions	System Actions
1	The property owner enters property details and uploads images. The property owner sets rental rates and availability dates	The system saves the new property. The property listing is published and made visible to potential tenants.
Use Case Number: 3		
Name Tenant Screening		
Brief description Property owners can screen and evaluate potential tenants before finalizing rental agreements.		
Actors Property owner		
1.1 Tenant Sreening		
Step	User Actions	System Actions
1	Property owner makes a decision to accept or reject a tenant.	The system notifies the tenant about the property owner's decision regarding the screening/
Use Case Number: 4		
Name: Property Owner Monthly Subscription		
Brief Description: This use case allows property owners to subscribe to a monthly subscription plan to access premium features and services offered by the Rental Property Management System.		
Actors: Property Owner		
Step	User Actions	System Actions
1	The property owner logs into their account on the system.	The system presents the property owner with a subscription page, displaying various subscription plans and their respective benefits.

2	The property owner selects the desired monthly subscription plan.	The system calculates the subscription fee based on the chosen plan and presents the payment options (e.g., credit card, PayPal, etc.).
3	The property owner provides the necessary payment information and confirms the subscription purchase.	The system processes the payment and activates the selected subscription plan for the property owner's account.

Non-Functional Requirements:

- The system should be accessible via a web interface to accommodate users' mobility requirements.
- The system should support multiple platforms, such as Windows, macOS, iOS, and Android, ensuring a consistent user experience across devices.
- The system should adhere to industry standards for data security and privacy.
- The system should be responsive, ensuring fast loading times and smooth user interactions.
- The system should have an intuitive and user-friendly interface, allowing stakeholders to navigate and perform tasks efficiently.

Conclusion:

The Software Requirement Specification (SRS) for the Rental Property Management System (RPMS) has been developed to provide a clear and comprehensive overview of the system's functionalities and requirements. The RPMS aims to streamline and optimize the management of rental properties for both property owners and tenants. The use case descriptions have outlined the key user interactions and system actions, ensuring that all critical scenarios are addressed.

The RPMS will allow new users to create accounts and sign up for the system while existing users can easily sign in to access their account features. Property owners will have the ability to add and manage property listings, including property details, images, rental rates, and availability dates. Additionally, property owners can screen and evaluate potential tenants before finalizing rental agreements, ensuring a secure and reliable tenant selection process.

One significant feature of the RPMS is the Property Owner Monthly Subscription, which enables property owners to subscribe to premium features and services offered by the system. This subscription plan will provide enhanced functionalities to improve the property management experience for property owners.

The SRS also highlights the importance of non-functional requirements, such as

accessibility, platform support, data security, responsiveness, and user-friendliness. These requirements ensure that the RPMS delivers a seamless and secure experience to all stakeholders across different devices and platforms.

In conclusion, the Rental Property Management System's SRS serves as a foundation for the development team to design, implement, and deliver a robust and efficient software solution. By adhering to the outlined use cases and requirements, the RPMS aims to provide an intuitive and feature-rich platform for effective rental property management. The development team can utilize this document as a reference to create a successful Rental Property Management System that meets the needs of property owners and tenants alike.

GitHub URL for the RPMS project: [<https://github.com/Abelkelemu/SWEProject.git>]