

# Figma Front-End Design Link

<https://www.figma.com/proto/gkRKT7xJLcE5aRIdrFKgTn/RentEase?node-id=6-484&node-type=frame&t=FkNWXcu796LykJEl-1&scaling=scale-down&content-scaling=fixed&page-id=0%3A1>

## Requirements for Both Landlords and Tenants

### Requirement 1: User Registration

Users should be able to register for a new account by providing the necessary information, such as username, email, password, and role (either a landlord or tenant).

#### Test Case 1: User Registration

1. **Objective:** Verify that users can successfully register for a new account as either a landlord or tenant.
2. **Action:** On the registration page, enter valid details (e.g., username, email, password) and select the role (tenant or landlord), then click "Register."
3. **Assert:** Ensure the user is registered and redirected to the login page with a confirmation message displayed.

#### Test Case 2: Invalid Email Format

1. **Objective:** Verify that the system rejects invalid email formats during registration.
2. **Action:** Enter a valid username and password but an invalid email, then click "Register."
3. **Assert:** Ensure that an appropriate error message is displayed, and registration is not processed.

### Requirement 2: User Login

Users (landlords and tenants) should be able to log in to their accounts using valid credentials.

#### Test Case 1: User Login

1. **Objective:** Verify that users can successfully log in to the system.
2. **Action:** On the login page, enter a valid email and password, then click "Login."

3. **Assert:** Ensure that the user is logged in and redirected to the appropriate dashboard (landlord or tenant dashboards are different).

## Test Case 2: Invalid Login Attempt

1. **Objective:** Verify that the system rejects invalid login credentials.
2. **Action:** Enter an incorrect email or password, then click "Login."
3. **Assert:** Ensure that an error message is displayed, and the user is not logged in.

# Requirements for Landlords

## Requirement 3: Property Management

Landlords should be able to add, edit, and delete rental properties.

### Test Case 1: Add Properties

1. **Objective:** Verify that landlords can add a new property.
2. **Action:** On the property management page, click "Add Property," fill in the details, and click "Save."
3. **Assert:** Ensure that the new property appears in the property list.

### Test Case 2: Edit Property

1. **Objective:** Verify that landlords can edit an existing property.
2. **Action:** On the property list page, click "Edit" for a property, change details, and click "Save."
3. **Assert:** Ensure that the updated details are reflected in the property list.

### Test Case 3: Delete Property

1. **Objective:** Verify that landlords can delete a property.
2. **Action:** On the property list page, click "Delete" for a property.
3. **Assert:** Ensure that the property is removed from the property list.

## Requirement 4: Tenant Management

Landlords should be able to view tenant details and communicate with them.

### Test Case 1: View Tenant Details

1. **Objective:** Verify that landlords can view tenant information.
2. **Action:** On the tenant management page, click on a tenant's name to view their details.
3. **Assert:** Ensure that the tenant's full information is displayed.

### Test Case 2: Send Message to Tenant

1. **Objective:** Verify that landlords can send messages to tenants.
2. **Action:** On the tenant details page, click "Send Message," type a message, and click "Send."
3. **Assert:** Ensure that the message is sent successfully.

### Test Case 3: View Message History

1. **Objective:** Verify that landlords can view message history with tenants.
2. **Action:** On the tenant details page, click "Message History."
3. **Assert:** Ensure that the previous conversations with the tenant are displayed.

## Requirements for Tenants

### Requirement 5: Submit Maintenance Request

Tenants should be able to submit maintenance requests with descriptions and photos.

### Test Case 1: Submit Maintenance Request

1. **Objective:** Verify that tenants can submit maintenance requests.
2. **Action:** On the maintenance request page, fill in the issue description, upload a photo, and click "Submit."
3. **Assert:** Ensure that the request is successfully submitted and a confirmation is displayed.

## Test Case 2: View Maintenance Request Status

1. **Objective:** Verify that tenants can view the status of their maintenance requests.
2. **Action:** On the maintenance requests page, view the list of submitted requests.
3. **Assert:** Ensure that the correct status (e.g., "In Progress," "Completed") is displayed.

## Test Case 3: Update Maintenance Request

1. **Objective:** Verify that tenants can update an existing maintenance request.
2. **Action:** On the maintenance requests page, click "Edit" for a request, update the details, and click "Save."
3. **Assert:** Ensure that the updated request is reflected in the request list.

## Requirement 6: Rent Payment

Tenants should be able to securely make rent payments and view payment history.

### Test Case 1: Make Rent Payment

1. **Objective:** Verify that tenants can make a rent payment.
2. **Action:** On the rent payment page, enter payment details and click "Pay."
3. **Assert:** Ensure that the payment is processed successfully and a confirmation message is displayed.

### Test Case 2: View Payment History

1. **Objective:** Verify that tenants can view their payment history.
2. **Action:** On the payment history page, view the list of past payments.
3. **Assert:** Ensure that all payments are listed correctly with their corresponding dates and amounts.

### Test Case 3: Set Up Auto-Pay

1. **Objective:** Verify that tenants can set up automatic rent payments.
2. **Action:** On the rent payment settings page, enable the auto-pay option and enter payment details.

3. **Assert:** Ensure that auto-pay is activated and future payments will be processed automatically.