F23-081-D-TranspaRent

Project Team

Usman Kamal 20I-0562
Ameera Haider 20I-0799
Ahmed Baig 20I-1884

Session 2020-2024

Supervised by
Dr. Qaiser Shafi

Co-Supervised by
Dr. Muhammad Asim

Department of Computer Science

National University of Computer and Emerging Sciences Islamabad, Pakistan

June, 2024

Table of Contents

Chapter 1

Introduction

The product specified in this document is a comprehensive rental property application called "TranspaRent." This application integrates blockchain technology, AI-powered features to streamline the rental process, ensuring efficiency, security, and transparency. The document is intended for various types of readers, including developers who will build the application, project managers overseeing its progress, marketing staff promoting the application, users interacting with the platform, testers ensuring its functionality and reliability, and documentation writers preparing user guides and technical manuals.

1.1 Problem Statement

The current rental property market in Pakistan faces significant challenges, including inefficiencies, lack of transparency, and legal ambiguities. Traditional methods of contract creation, identity verification, and payment processing are often time-consuming and prone to disputes. The absence of formalized rental agreements leaves both landlords and tenants vulnerable to misunderstandings and fraud. Existing real estate platforms primarily serve as listing services without ensuring comprehensive security or legal compliance. This system, "TranspaRent," is being developed to address these critical issues by providing a secure, transparent, and user-friendly rental platform. Integrating blockchain technology ensures that all transactions and agreements are securely recorded, eliminating the possibility of post-transaction fraud. AI-powered features offer competitive pricing suggestions and automate processes, enhancing efficiency. The platform also includes robust identity verification and a transparent dispute resolution mechanism, promoting trust and reducing conflicts. By streamlining the rental process and ensuring legal compliance, TranspaRent aims to redefine the rental experience, making it reliable and hassle-free for both property owners and tenants.

1.2 Scope

- Development of a comprehensive rental property application for Pakistan.
- Secure contract management using blockchain technology.
- Robust identity verification through integration with NADRA.
- AI-driven pricing suggestions based on historical data.
- Payment processing with a secure gateway.
- Transparent dispute resolution mechanism.
- Mutual reviewing system for landlords and tenants.
- Focus on rental properties within Islamabad.
- Adherence to local rental laws and regulations.

- Integration of external services like NADRA API and secure payment gateways.

1.3 Modules

1.3.1 User Management

The User Management module handles the registration, authentication, and profile management of users within the system. It ensures that users can securely access and manage their accounts and personal information.

- Secure login for user registration and authentication.
- User profile management for updating personal details and managing profiles.

1.3.2 Contract Management

The Contract Management module encompasses the generation, validation, and storage of tenant agreements, ensuring compliance with Islamabad's rental laws and regulations, and providing a user-friendly interface for creating agreements.

- Generate legally binding tenant agreements by inputting essential details.
- Validate contract details such as rent amount, duration, and terms to ensure adherence to Islamabad's rental laws and regulations.

- Store agreements securely on a blockchain for transparency and permanence.
- Provide a user-friendly interface for creating tenant agreements.

1.3.3 Identity Verification

Integrate with NADRA for robust tenant and landlord authentication. Allow users to upload and verify identity documents for authentication.

- Leverage NADRA integration for secure tenant and landlord verification.
- Provide functionality for users to upload and validate their identity documents.

1.3.4 Mutual Reviewing System

Provide access to property's history and creditworthiness assessments. Provide access to rental history and creditworthiness assessments.

- Enable tenants to view property reviews and ratings.

1.3.5 Price Recommendation

Leverage AI and historical data to recommend competitive rental prices, utilizing user-input property details for accurate suggestions.

- Utilize AI and historical data to recommend competitive rental prices.

- Allow users to input property details for accurate pricing recommendations.

1.3.6 Document Management

Ensure the secure storage and management of crucial rental contracts while enabling users to upload and verify necessary documents for validation purposes.

- Securely store and manage rental contracts.
- Allow users to upload and verify required documents for validation.

1.3.7 Payment Processing

Facilitate payments via a secure payment gateway while offering users access to their payment history and transaction tracking.

- Automate payments through a secure payment gateway.
- Provide payment history and transaction tracking for users.

1.3.8 Dispute Resolution

Facilitate transparent and unquestionable dispute resolution processes.

- Ensure clear and indisputable procedures for resolving disputes.
- Implement open and unquestionable methods for settling disputes.

1.4 User Classes and Characteristics

Class	Characteristic
User	Landlord and Tenant
Landlord	Property owner listing rentals. Around 200 initially. Features: property listing, contract management, payment, tenant reviews.
Tenant	Individual seeking rental properties. About 1000 initially. Features: property search, contract signing, automated payments, landlord reviews.

Chapter 2

Project Requirements

This chapter describes the functional and non-functional requirements of the project.

2.1 Requirement Gathering Techniques

2.1.1 High Level Use Cases

UC01: User Registration and Authentication	
Actor	Users
Type	Primary
Description	Users can register on the platform, providing necessary details. They can authenticate their identity through secure login methods, ensuring a safe and verified user base.

UC02: Create a Rental Agreement	
Actor	Landlord
Type	Primary
Description	Landlords can create a new rental agreement by entering essential details such as rent amount, duration, and terms. The system validates the information for accuracy and adherence to local rental laws. Once validated, the agreement is securely stored on the blockchain.

UC03: Review Rental History and Creditworthiness	
Actor	Users
Type	Primary

Description	Both landlords and tenants can access a mutual reviewing system. They can view each other's review history and creditworthiness assessments, aiding in the decision-making process for both parties.

UC04: Identity Verification	
Actor	Users
Type	Primary
Description	Users, both landlords, and tenants, can verify their identity by uploading necessary documents. System will extract ID card number from image of ID card. The system integrates with NADRA for robust authentication, ensuring a secure environment for all users.

UC05: AI-Based Pricing Suggestions	
Actor	Landlord
Type	Primary
Description	Landlords can input property details, and the system, utilizing AI algorithms and historical data, suggests competitive rental prices. This feature promotes fair pricing practices and helps landlords set appropriate rental rates.

UC06: Manage Rental Documents	
Actor	Users
Type	Primary
Description	Users can securely upload and manage essential rental documents, such as property deeds and identification papers. The system ensures the documents are valid and authentic, maintaining a transparent record for all parties.

UC07: Payments	
Actor	Tenant
Type	Primary
Description	Tenants can do payments through various payment methods. The system holds the payment until contract is confirmed and

	then send tot landlord. History and transaction tracking are available for users' reference, stored on blockchain.

UC08: Transparent Dispute Resolution	
Actor	Users
Type	Primary
Description	In case of disputes, a structured and transparent dispute resolution process is available. Both parties can submit their concerns, and the system facilitates resolution, ensuring fairness and trust between landlords and tenants.

2.1.2 Extended Use Cases

<table>
<tr><th colspan="2">UC01: User Registration and Authentication</th></tr>
<tr><td><b>Scope</b></td><td>TranspaRent</td></tr>
<tr><td><b>Level</b></td><td>Primary</td></tr>
<tr><td><b>Primary Actor</b></td><td>Users</td></tr>
<tr><td><b>Stakeholders and Interests</b></td><td><b>User:</b> Interested in a secure and user-friendly registration process, ensuring their data is protected.</td></tr>
<tr><td><b>Pre-Condition</b></td><td>The user has access to the rental property app and a stable internet connection.</td></tr>
<tr><td><b>Post Condition</b></td><td>The user is successfully registered and authenticated, gaining access to the platform's features.</td></tr>
<tr><td><b>Main Success Scenario</b></td><td><table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>User selects "Register" option</td><td></td></tr>
<tr><td></td><td>Displays registration form with fields for necessary details</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td>User enters registration details</td><td></td></tr>
<tr><td></td><td>Validates data in real-time; provides feedback</td></tr>
<tr><td>User submits registration form</td><td></td></tr>
<tr><td></td><td>Stores user information</td></tr>
</table></td></tr>
<tr><td><b>Extensions</b></td><td><b>User provides incorrect information during registration:</b> System displays appropriate error messages, prompting the user to correct the provided information.

<b>User attempts to register with an already registered email/phone number:</b> System notifies the user that the email/phone number is already associated with an existing account and prompts them to log in or recover their account.</td></tr>
</table>

UC02: Create a Rental Agreement	
Scope	TranspaRent
Level	Primary
Primary Actor	User
Stakeholders and Interests	**Landlord:** Interested in creating a legally binding and accurate rental agreement. **Tenants:** Interested in receiving a clear, detailed, and legally compliant rental agreement. **Regulatory Authorities:** Interested in ensuring the rental agreements comply with local rental laws and regulations.

<table>
<tr><td><b>Pre-Condition</b></td><td>The landlord is logged into the rental property app and has selected the option to create a new rental agreement. The required property details and tenant information are available.</td></tr>
<tr><td><b>Post Condition</b></td><td>A legally binding rental agreement is created, validated, and securely stored on the blockchain.</td></tr>
<tr><td><b>Main Success Scenario</b></td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>Landlord selects "Create Rental Agreement"</td><td></td></tr>
<tr><td></td><td>Displays rental agreement creation form</td></tr>
<tr><td>Landlord fills out agreement details</td><td></td></tr>
<tr><td></td><td>Validates data in real-time; provides feedback</td></tr>
<tr><td>Landlord submits the agreement form</td><td></td></tr>
<tr><td></td><td>Validates details for legal compliance; securely stores agreement on the blockchain; notifies both parties</td></tr>
</table>
</td></tr>
<tr><td><b>Extensions</b></td><td><b>Landlord provides incorrect or incomplete information:</b> System displays appropriate error messages, indicating the fields that need correction. The landlord can revise the information.</td></tr>
</table>

	Rental agreement details do not adhere to local rental laws: System informs the landlord about the specific issues that violate the local laws. Provides guidance on how to rectify the problem or adjust the terms to comply with regulations. **Tenant does not agree with the terms of the agreement:** System sends a notification to the landlord about the tenant's disagreement. Landlord and tenant can negotiate the terms, and the agreement can be edited and re-submitted for validation.

<table>
<tr><th colspan="2">UC03: Review Rental History and Creditworthiness</th></tr>
<tr><td><b>Scope</b></td><td>TranspaRent</td></tr>
<tr><td><b>Level</b></td><td>Primary</td></tr>
<tr><td><b>Primary Actor</b></td><td>Users</td></tr>
<tr><td><b>Stakeholders and Interests</b></td><td><b>Landlord:</b> Interested in assessing the potential tenant's review history and creditworthiness to make an informed decision.

<b>Tenants:</b> Interested in reviewing the landlord's history to ensure a trustworthy rental experience.</td></tr>
<tr><td><b>Pre-Condition</b></td><td>Both the landlord and tenant are registered users on the platform and have expressed interest in a specific rental property or tenant.</td></tr>
<tr><td><b>Post Condition</b></td><td>Both parties have reviewed each other's rental history reviews and creditworthiness, aiding in their decision-making process regarding the rental agreement.</td></tr>
<tr><td><b>Main Success Scenario</b></td><td><table><tr><th>Actor Action</th><th>System Response</th></tr></table></td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td>Landlord/Tenant selects a specific propety</td><td></td></tr>
<tr><td>Landlord/Tenant selects "Review History"</td><td></td></tr>
<tr><td></td><td>Shows summary of the selected property’s reviews</td></tr>
</table></td></tr>
<tr><td><b>Extensions</b></td><td><b>Property’s rental history or creditworthiness information is not available:</b> System informs the reviewer that there is no available information for the selected property, possibly because the property is new or hasn't participated in previous rental agreements on the platform.

<b>Landlord or tenant decides not to proceed based on the review:</b> System provides options for the user to continue their search for a suitable tenant/property or suggests alternative matches based on their preferences and criteria.</td></tr>
</table>

UC04: Identity Verification	
Scope	TranspaRent
Level	Primary
Primary Actor	Users
Stakeholders and Interests	**Landlord:** Interested in verifying the identity of potential tenants to ensure security and trustworthiness. **Tenants:** Interested in verifying the identity of landlords to ensure the legitimacy of the rental property. **Regulatory Authorities:** Interested in ensuring compliance with identity

<table>
<tr><td></td><td>verification regulations to prevent fraudulent activities.</td></tr>
<tr><td><b>Pre-Condition</b></td><td>The user (landlord or tenant) is registered on the platform and has accessed the identity verification section. The user has a valid government-issued ID card available for verification.</td></tr>
<tr><td><b>Post Condition</b></td><td>The user's identity is verified, and the verified status is updated in their profile. The user gains trust in the platform.</td></tr>
<tr><td><b>Main Success Scenario</b></td><td>
<table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>User selects "Identity Verification" and uploads ID card image</td><td></td></tr>
<tr><td></td><td>Processes the image to extract ID card number; prompts for additional information if necessary</td></tr>
<tr><td>NADRA verifies the ID card number</td><td></td></tr>
<tr><td></td><td>Receives verification status; if valid, marks user's identity as verified; sends verification confirmation to the user</td></tr>
</table>
</td></tr>
<tr><td><b>Extensions</b></td><td><b>User uploads an unclear or invalid ID card image:</b> System informs the user that the uploaded image is unclear or invalid. Requests the user</td></tr>
</table>

	to upload a clear, high-quality image for verification. **Additional information is required for verification:** System prompts the user for additional details such as address or date of birth. Ensures the user provides accurate and matching information to complete the verification process. **NADRA verification fails:** System notifies the user that their identity verification has failed and provides information on how to rectify the issue, such as contacting NADRA or verifying their ID card details.

UC05: AI-Based Pricing Suggestions	
Scope	TranspaRent
Level	Primary
Primary Actor	Landlord
Stakeholders and Interests	**Landlord:** Interested in setting competitive and fair rental prices for their properties to attract potential tenants and maximize rental income. **Tenants:** Interested in finding affordable rental properties that offer value for their budget. **Regulatory Authorities:** Interested in ensuring transparency in rental pricing, preventing price gouging, and promoting fair market practices.
Pre-Condition	The landlord is logged into the platform and has navigated to the pricing suggestions section. The landlord has accurate details about their property including size, location, amenities, and other relevant factors.

<table>
<tr><td><b>Post Condition</b></td><td>The landlord receives AI-based rental price suggestions based on the provided property details.</td></tr>
<tr><td><b>Main Success Scenario</b></td><td><table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>Landlord inputs property details</td><td></td></tr>
<tr><td></td><td>Processes input data; runs AI algorithms</td></tr>
<tr><td>Landlord reviews suggested rental prices</td><td></td></tr>
<tr><td></td><td>Displays list of suggested prices with market comparisons; allows landlord to confirm the selection</td></tr>
</table></td></tr>
<tr><td><b>Extensions</b></td><td><b>Landlord provides incomplete or inaccurate property details:</b> System informs the landlord about missing or incorrect details and prompts them to review and correct the information. Provides guidance on the required data for accurate pricing suggestions.

<b>Landlord disagrees with the suggested prices:</b> System allows the landlord to manually adjust the rental price based on their preferences. Provides guidance on factors affecting rental pricing to assist the landlord in making an informed decision.</td></tr>
</table>

UC06: Manage Rental Documents	
Scope	TranspaRent
Level	Primary

<table>
<tr><td><b>Primary Actor</b></td><td>Users</td></tr>
<tr><td><b>Stakeholders and Interests</b></td><td><b>Landlord:</b> Interested in uploading property-related documents for verification and transparency.
<b>Tenants:</b> Interested in providing necessary identification and other documents for the rental agreement process.
<b>Regulatory Authorities:</b> Interested in ensuring the authenticity and legality of the documents submitted during the rental process.</td></tr>
<tr><td><b>Pre-Condition</b></td><td>The user (landlord or tenant) is registered on the platform and has initiated the process of creating a rental agreement. Required documents, such as property deeds or identification papers, are available in digital format.</td></tr>
<tr><td><b>Post Condition</b></td><td>The uploaded documents are securely stored, verified for authenticity, and linked to the respective rental agreements, ensuring transparency and legal compliance.</td></tr>
<tr><td><b>Main Success Scenario</b></td><td><table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>User selects "Manage Rental Documents"</td><td></td></tr>
<tr><td></td><td>Displays an option to upload documents</td></tr>
<tr><td>User uploads necessary documents</td><td></td></tr>
<tr><td></td><td>Stores Documents</td></tr>
</table></td></tr>
<tr><td><b>Extensions</b></td><td><b>User uploads invalid or illegible documents:</b> System informs the user about the issue with the uploaded document and requests a re-upload.</td></tr>
</table>

	Provides guidelines for acceptable document formats and quality.

<table>
<tr><th colspan="2">UC07: Payments Processing</th></tr>
<tr><td><b>Scope</b></td><td>TranspaRent</td></tr>
<tr><td><b>Level</b></td><td>Primary</td></tr>
<tr><td><b>Primary Actor</b></td><td>Tenant</td></tr>
<tr><td><b>Stakeholders and Interests</b></td><td><b>Landlord:</b> Interested in setting up automated rent payments for convenience and ensuring timely payments to the landlord.
<b>Tenants:</b> Interested in receiving rent payments on time and being notified of successful payments.
<b>Regulatory Authorities:</b> Interested in ensuring transparency in rent payment processes and adherence to financial regulations.</td></tr>
<tr><td><b>Pre-Condition</b></td><td>The tenant is registered on the platform, has a linked payment method (e.g., bank account or credit card), and has initiated the process of setting up rent payments.</td></tr>
<tr><td><b>Post Condition</b></td><td>Automated rent payments are scheduled, notifications and reminders are set up, and payment history is recorded for both the tenant and the landlord.</td></tr>
<tr><td><b>Main Success Scenario</b></td><td><table>
<tr><th>Actor Action</th><th>System Response</th></tr>
<tr><td>Tenant selects "Automate Rent Payments"</td><td></td></tr>
<tr><td></td><td>Displays a form for payment setup</td></tr>
</table></td></tr>
</table>

<table>
<tr>
<td></td>
<td>
<table>
<tr><td>Tenant inputs payment details and confirms setup</td><td></td></tr>
<tr><td></td><td>Processes payment information; schedules automated payments; sends confirmation notifications</td></tr>
<tr><td></td><td>Sends automated notifications to tenant and landlord</td></tr>
</table>
</td>
</tr>
<tr>
<td><b>Extensions</b></td>
<td><b>Tenant provides incorrect or incomplete payment details:</b> System informs the tenant about the issues with the provided information and requests correction. Provides guidance on valid payment methods and formats.

<b>Tenant attempts to set up automated payments without a linked payment method:</b> System informs the tenant that a valid payment method needs to be linked first. Provides an option to link a payment method and proceed with the automated payment setup.

<b>Tenant attempts to set up automated payments for an incorrect rent amount or frequency:</b> System informs the tenant about the issue and requests correction. Provides guidance on the acceptable rent amount and payment frequency.</td>
</tr>
</table>

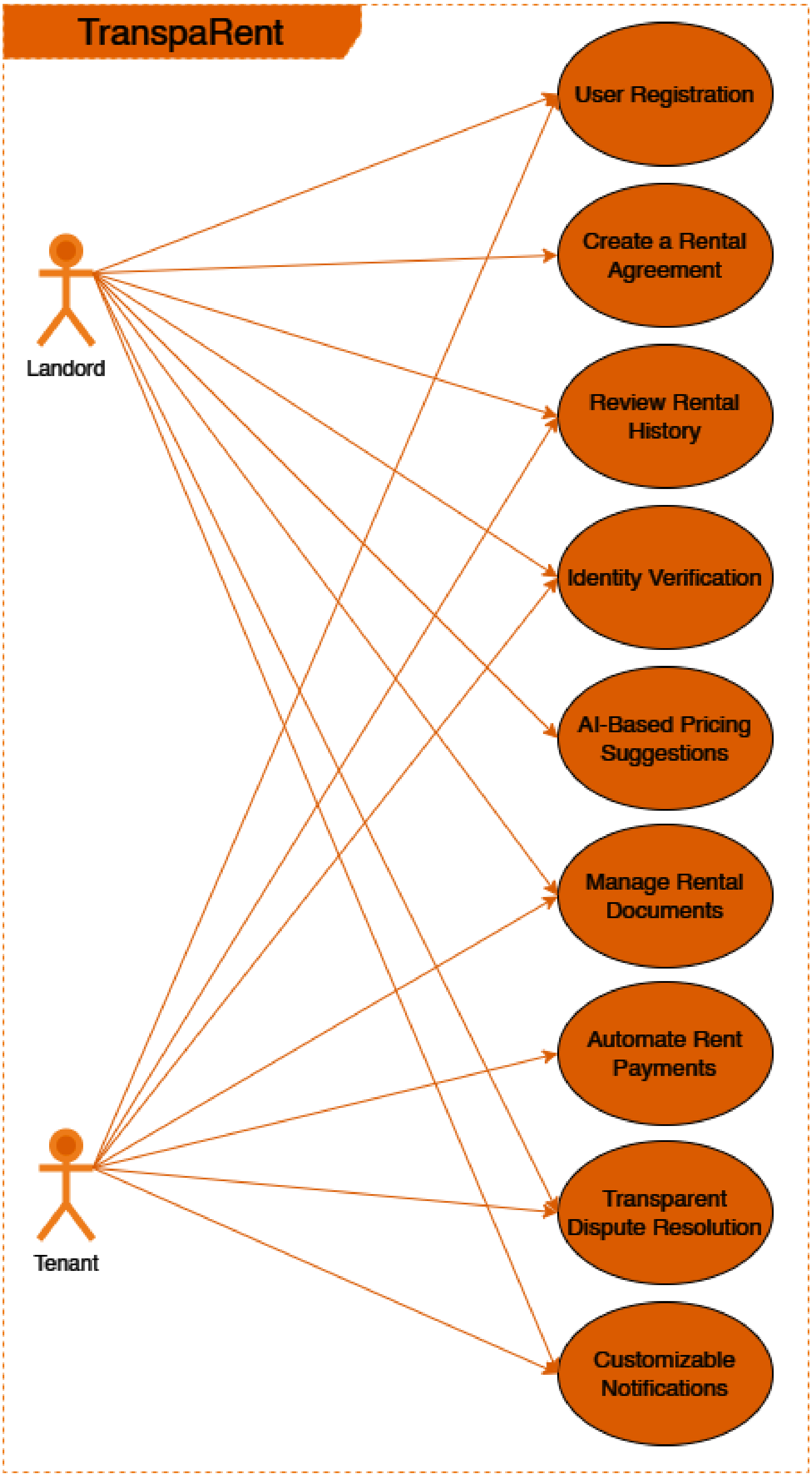
	Tenant misses a scheduled payment due to insufficient funds: System sends notifications to the tenant about the failed payment attempt. Provides options for the tenant to resolve the issue, such as updating payment information or making a manual payment.

UC08: Transparent Dispute Resolution	
Scope	TranspaRent
Level	Primary
Primary Actor	Users
Stakeholders and Interests	**Landlord:** Interested in resolving disputes efficiently and fairly to maintain a good landlord-tenant relationship and property management. **Tenants:** Interested in addressing concerns and disputes promptly to ensure a comfortable living environment and resolve any issues with the landlord. **Regulatory Authorities:** Interested in ensuring that dispute resolution processes are fair and transparent, protecting the rights of both landlords and tenants.
Pre-Condition	A dispute or concern has arisen between the landlord and tenant, and both parties have agreed to engage in the dispute resolution process provided by the platform.
Post Condition	The dispute is resolved, and an agreement or resolution is reached between the landlord and tenant. Both parties are notified of the resolution, ensuring clarity and understanding.

Main Success Scenario	**Actor Action**	**System Response**
	Landlord or tenant submits concerns and evidence	
		Provides a structured form for submission; allows attachment of relevant documents
	Both parties provide additional information if required	
		System analyzes concerns and evidence; facilitates communication
		System proposes a resolution to both parties
	Landlord and tenant agree on the proposed resolution	
		Confirms agreed resolution; updates relevant records; notifies both parties
Extensions	**One party fails to provide necessary information within the specified timeframe:** System sends reminders to the party to provide the required information. If the information is not provided within the	

	specified timeframe, the system proceeds with the available information. **System encounters complexities in the case that require legal expertise:** System informs the parties about the need for legal expertise. Provides options for involving a legal expert or mediator in the dispute resolution process. **Parties disagree with the proposed resolution:** System facilitates further communication between parties to address disagreements. If necessary, escalates the dispute to a higher level of authority or legal expert for resolution. Provides guidance on alternative resolution methods, such as arbitration or mediation.

2.1.3 Use Case Diagram



2.2 Functional Requirements

User Management Module:

1. User registration and authentication with secure login and password recovery.
2. User profile management allowing users to update their details and manage profiles.

Contract Management Module:

1. User-friendly interface for creating tenant agreements.
2. Validation of contract details for accuracy and legal compliance.
3. Secure storage and retrieval of tenant agreements on the blockchain.

Identity Verification Module:

1. Integration with NADRA for tenant and landlord authentication.
2. Upload and verification of identity documents for authentication.

Mutual Reviewing Module:

1. Access to property history analysis and creditworthiness assessments.
2. Tenants can view each property reviews and rental history.

Pricing Recommendation Module:

1. AI-driven competitive rental price suggestions based on historical data.
2. Property details input for accurate pricing recommendations.

Document Management Module:

1. Secure storage and management of essential rental contracts.

Payment Processing Module:

2. Automated rent payments through various payment methods.
3. Payment history and transaction tracking for users.

Dispute Resolution Module:

1. Structured and transparent dispute resolution processes.

2.3 Non-Functional Requirements

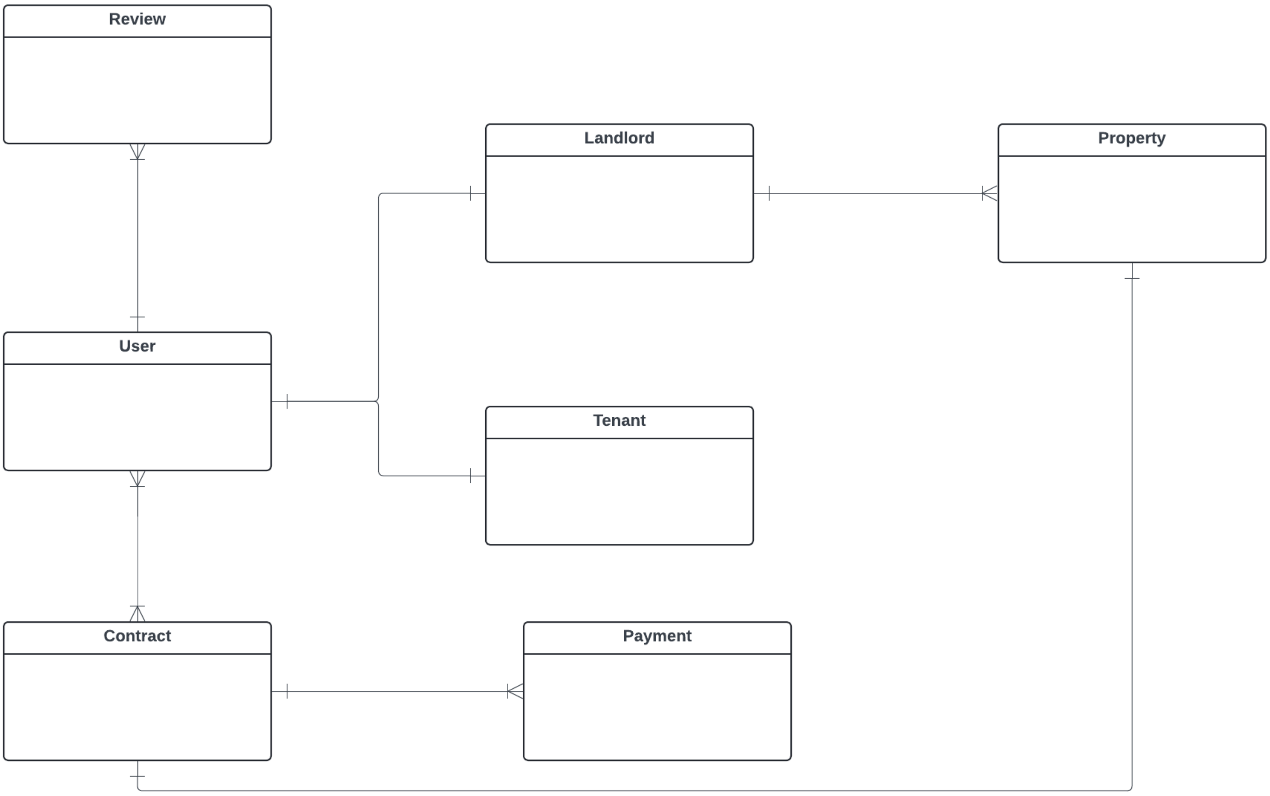
2. **Ease of Use (USE-1):** The system interface must be user-friendly and intuitive, allowing both landlords and tenants to perform tasks without extensive training.
3. **Efficiency (USE-2):** The system should provide quick response times to ensure efficient user interactions, especially during contract creation and retrieval.

4. **Error Handling (USE-3):** The system must provide clear error messages and guidance for users in case of incorrect inputs or system errors.
5. **Response Time (PERF-1):** The system should maintain low response times, with contract creation and retrieval processes taking no more than a few seconds.
6. **Scalability (PERF-2):** The system should be scalable to handle a growing user base and increasing data loads without performance degradation.
7. **Reliability (PERF-3):** The system must be highly reliable, with minimal downtime and data loss.
8. **Data Encryption (SEC-1):** All sensitive user data, including contracts and personal information, should be encrypted both in transit and at rest to protect against data breaches.
9. **Authentication (SEC-2):** Strong authentication mechanisms, such as multi-factor authentication, should be implemented to ensure user identity and prevent unauthorized access.
10. **Audit Trails (SEC-3):** The system should maintain detailed audit trails of user actions and interactions for security monitoring and compliance.
11. **Platform Compatibility (SCAL-1):** The system should be compatible with commonly used web browsers and mobile devices to ensure broad accessibility.

12. **Interoperability (SCAL-2):** It should allow for integration with third-party services, such as identity verification and payment gateways, to enhance functionality.
13. **Regulatory Compliance (COMP-1):** The system must comply with all relevant legal and regulatory requirements related to tenant agreements and data privacy, particularly in Islamabad.
14. **User Documentation (DOC-1):** Provide comprehensive user documentation and training materials to assist users in understanding and using the system effectively.
15. **Responsiveness (UX-1):** The system should provide a responsive design that adapts to various screen sizes and resolutions to enhance the user experience on both mobile and desktop devices.
16. **Feedback Mechanism (UX-2):** Include a feedback mechanism that allows users to report issues, suggest improvements, or seek assistance, promoting user engagement and satisfaction.
17. **Data Minimization (PRIV-1):** Collect only the minimum amount of data required for system functionality and adhere to privacy-by-design principles.
18. **Consent Mechanism (PRIV-2):** Implement mechanisms for obtaining explicit user consent for data processing and sharing, especially when integrating with third-party services.

19. **Technical Support (SUP-1):** Offer technical support and assistance to users, including a helpdesk or customer support portal.

2.4 Domain Model

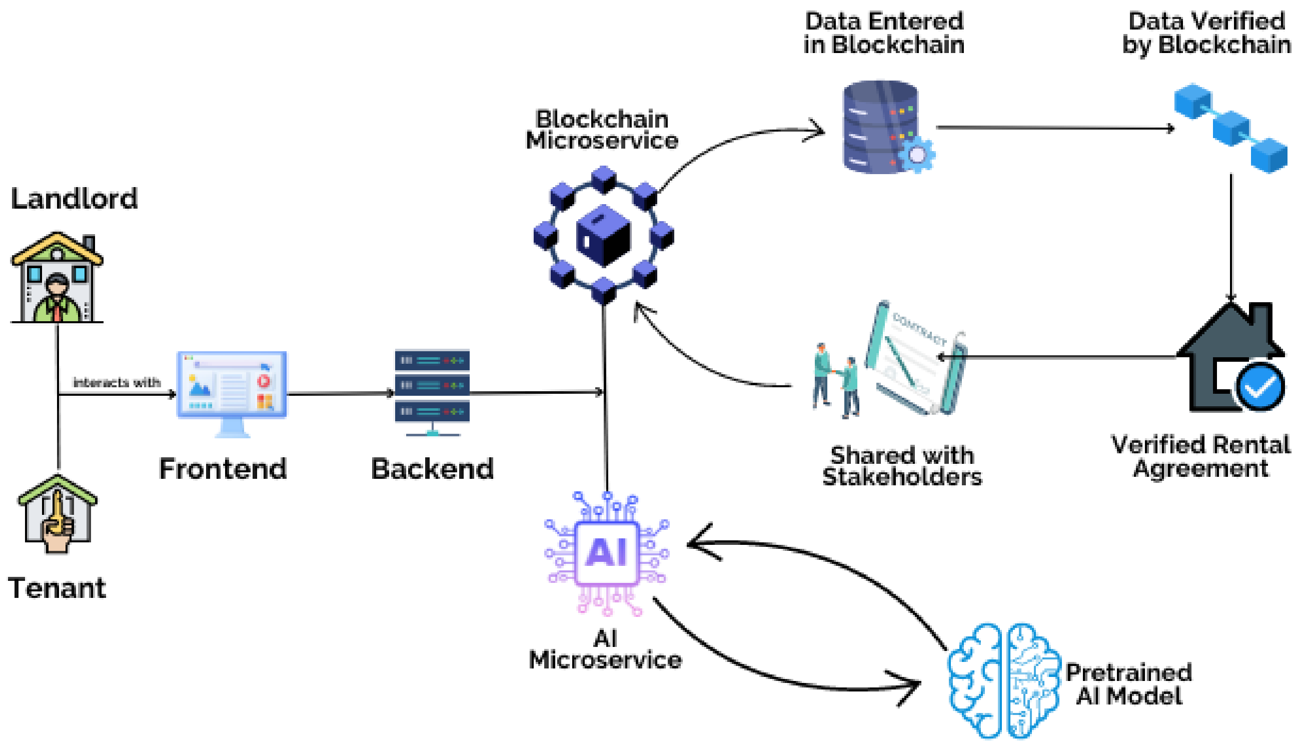


Chapter 3

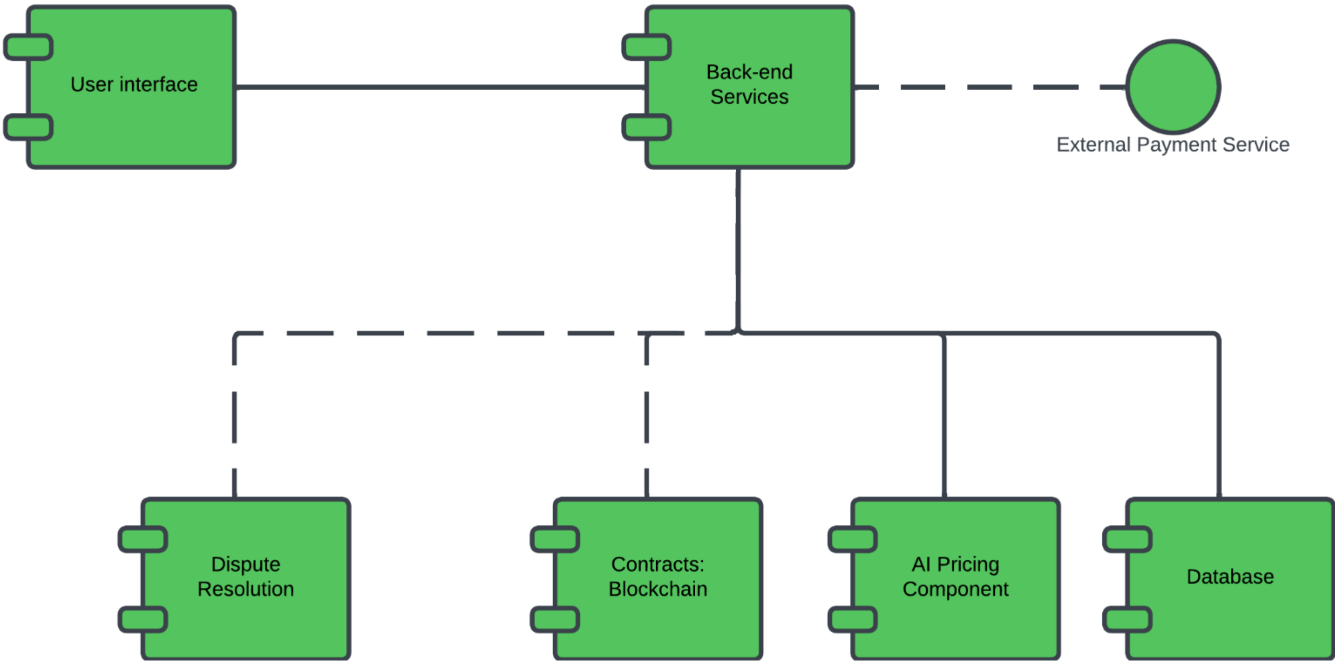
System Overview

3.1 Architectural Design

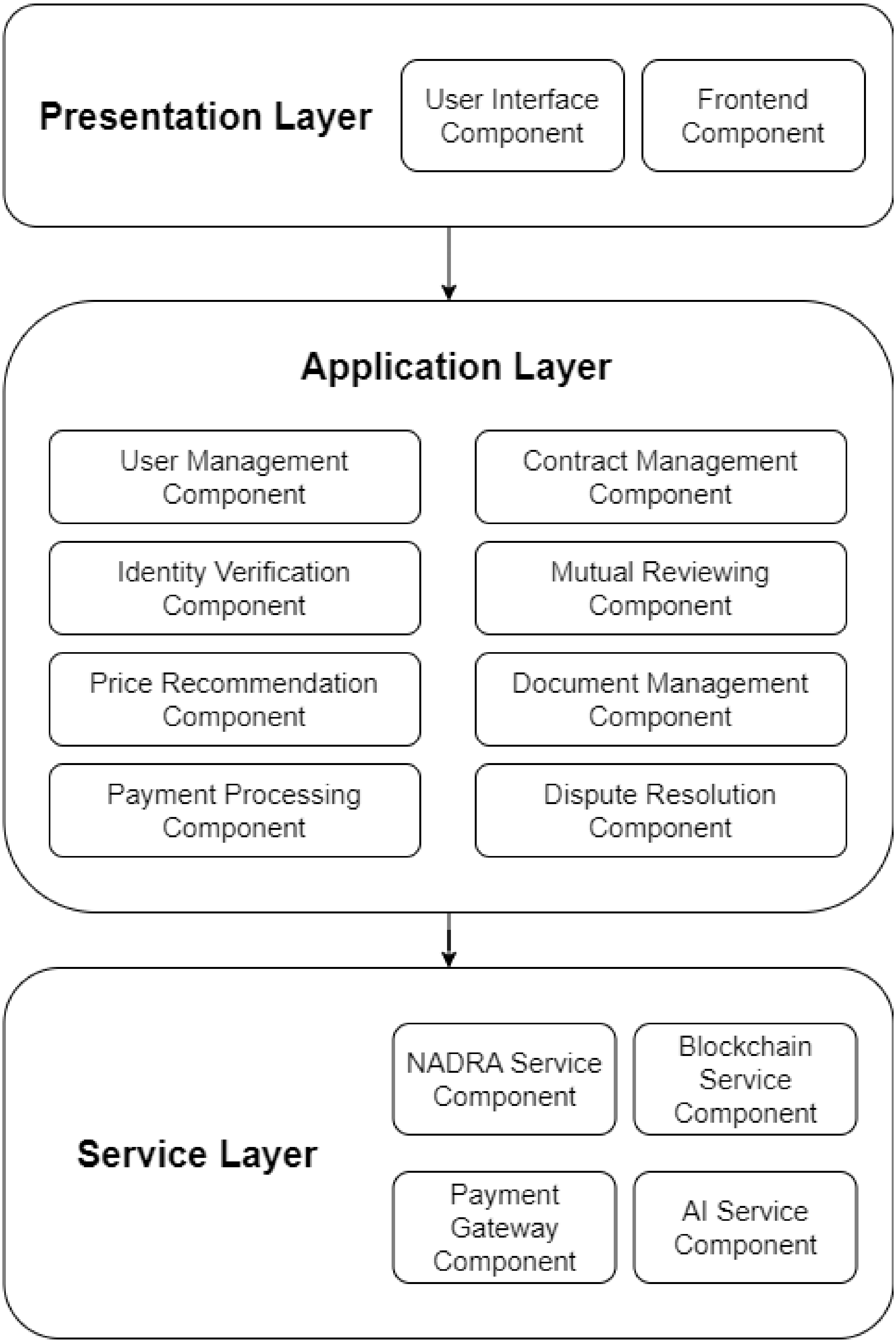
Architecture Diagram



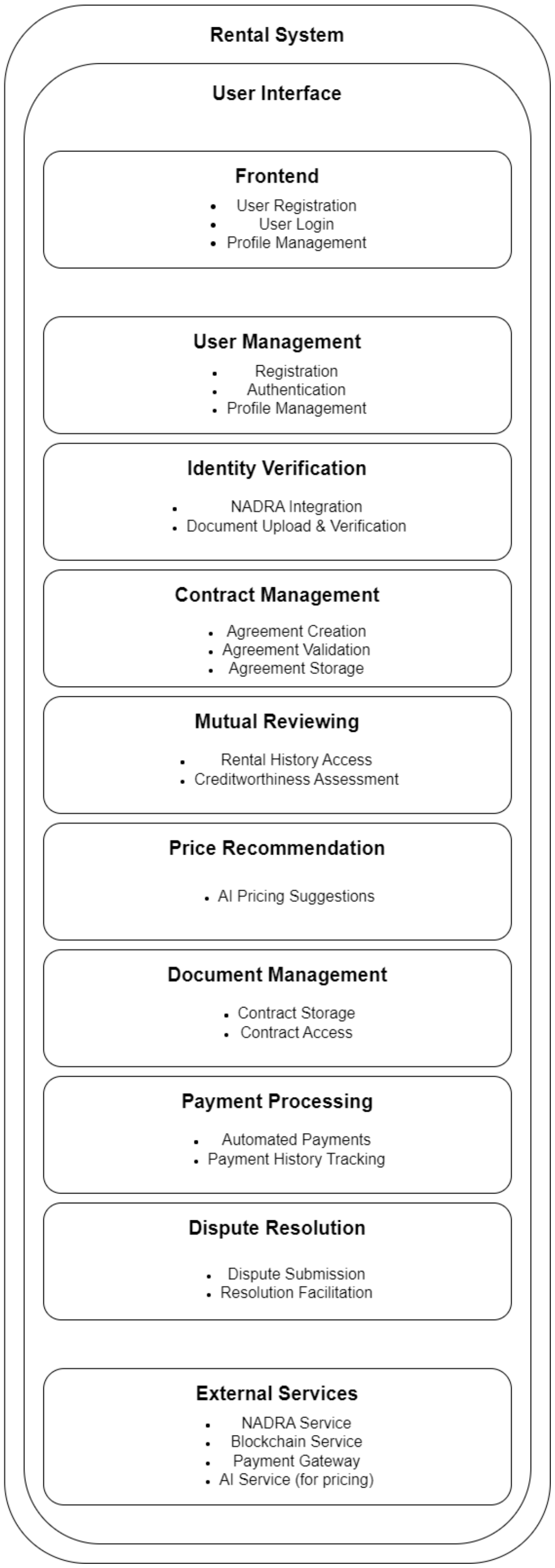
Component Diagram



Layer Diagram

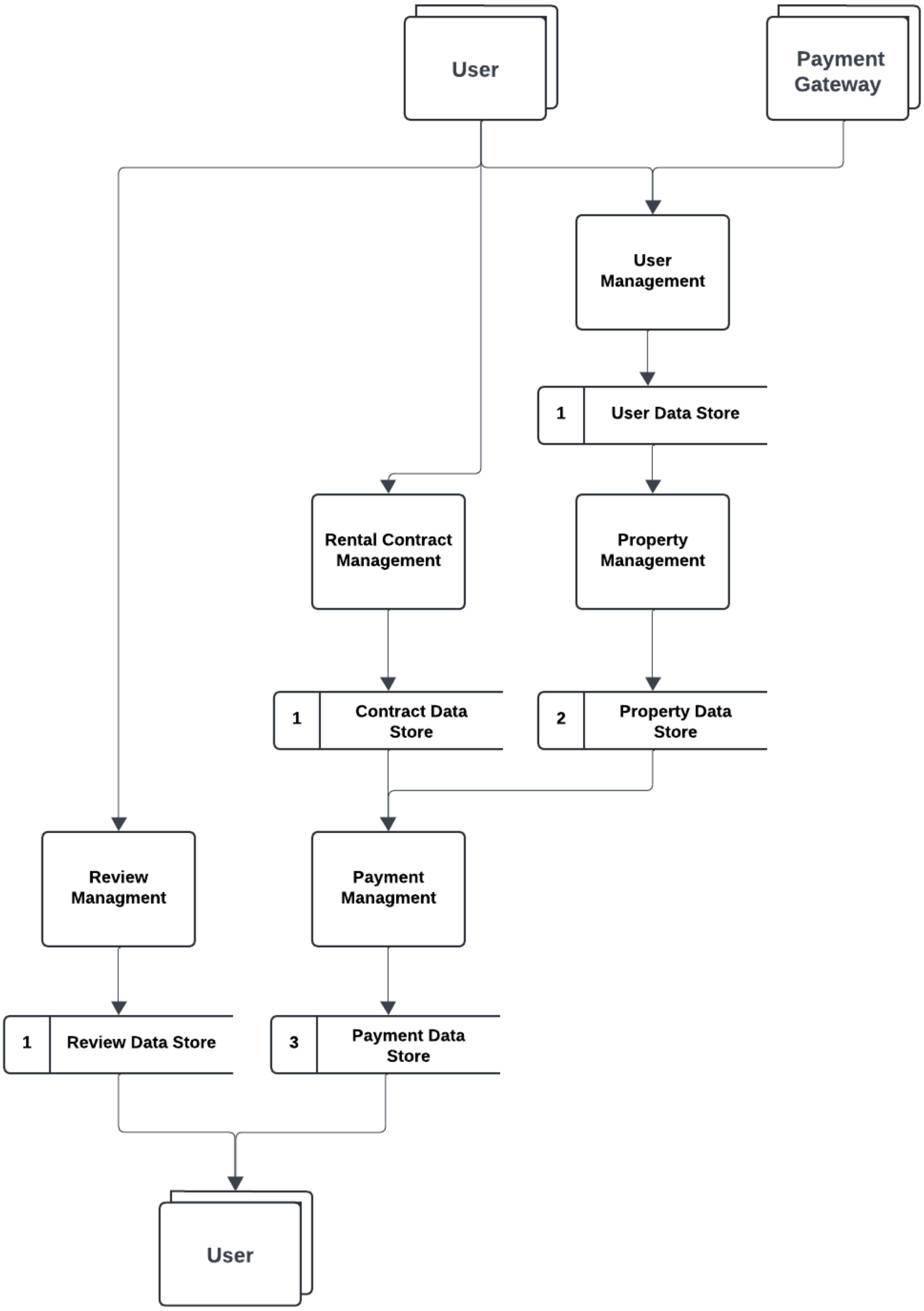

Presentation Layer
User Interface Component
Frontend Component
Application Layer
User Management Component
Contract Management Component
Identity Verification Component
Mutual Reviewing Component
Price Recommendation Component
Document Management Component
Payment Processing Component
Dispute Resolution Component
Service Layer
NADRA Service Component
Blockchain Service Component
Payment Gateway Component
AI Service Component

Overview Diagram



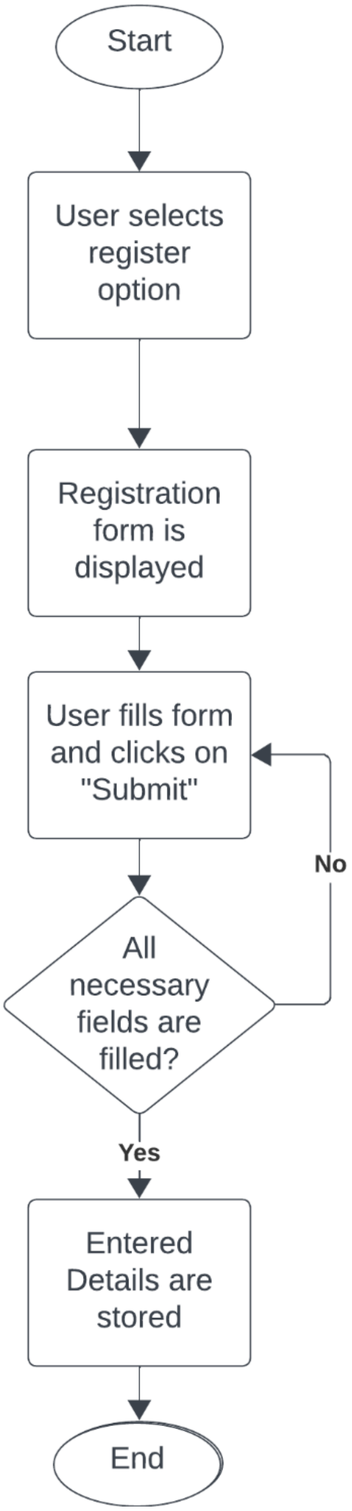
3.2 Design Models

Data Flow Diagram

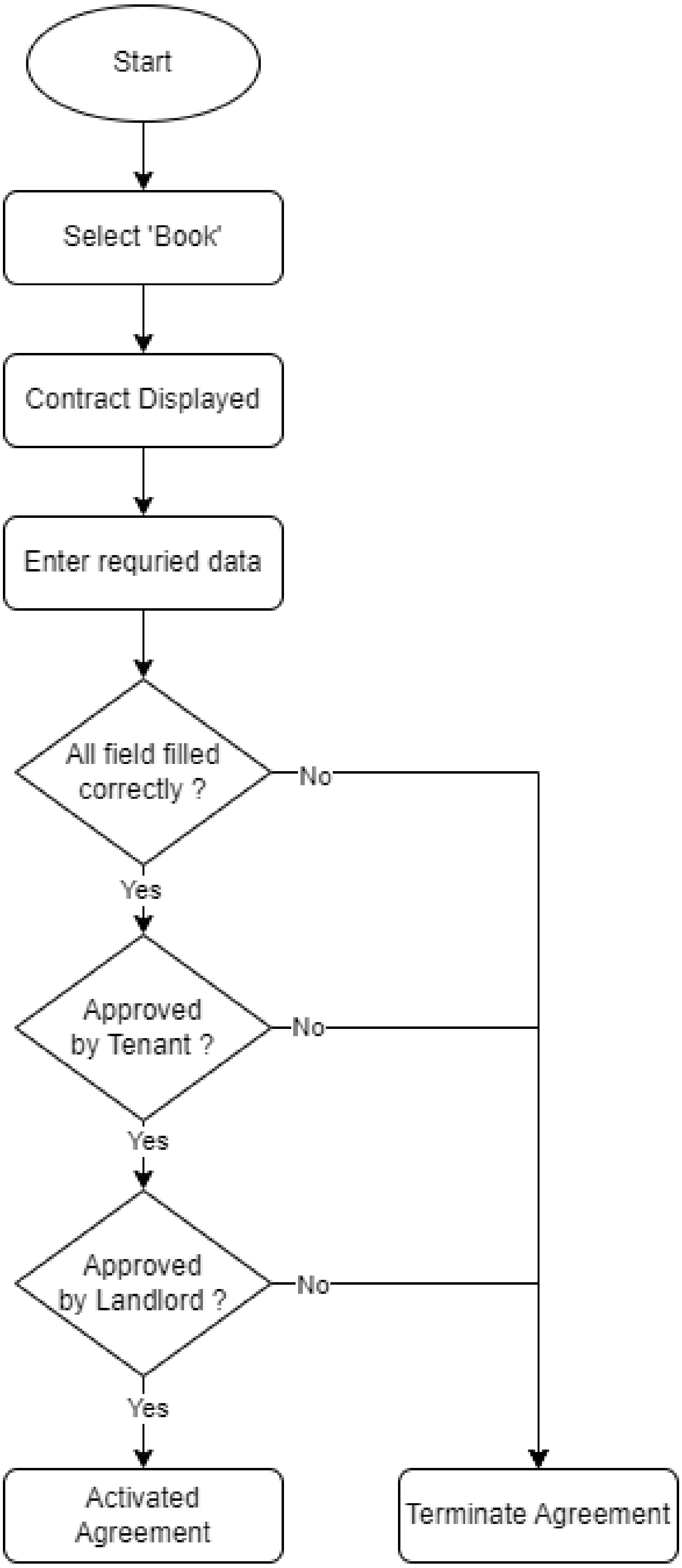


Flow Diagrams

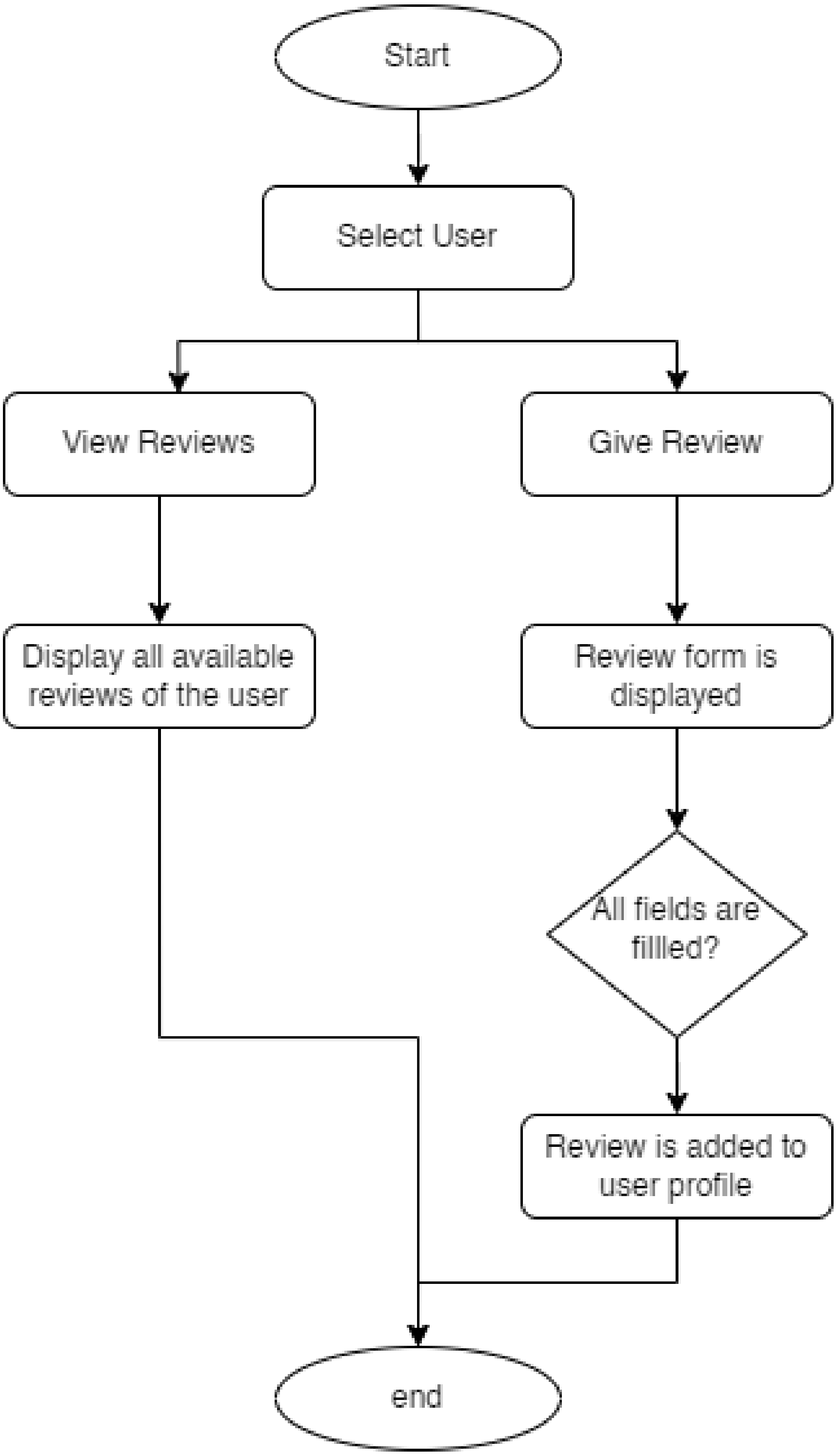
UC1


Start
User selects register option
Registration form is displayed
User fills form and clicks on "Submit"
All necessary fields are filled?
No
Yes
Entered Details are stored
End

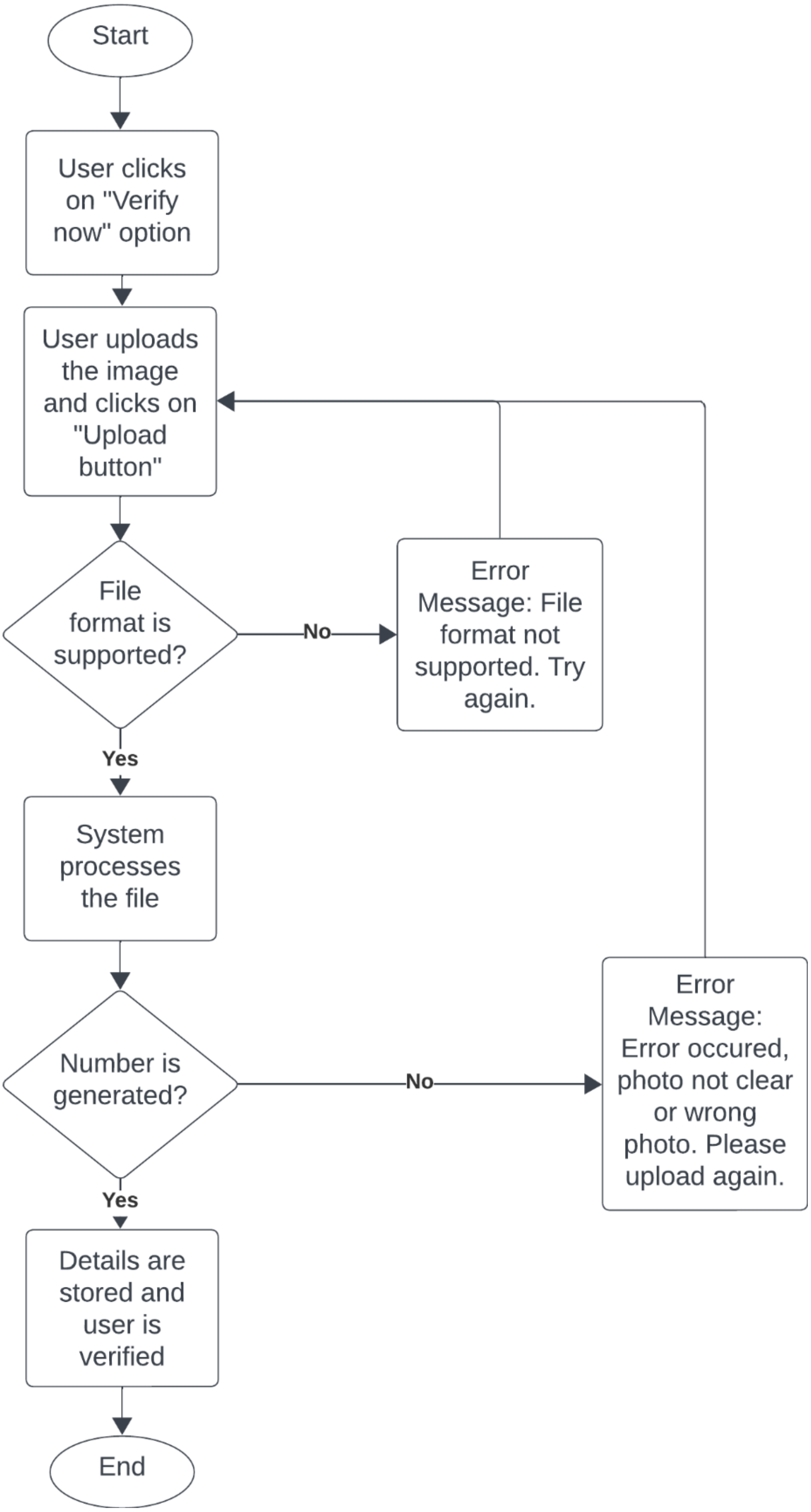
UC2


Start
Select 'Book'
Contract Displayed
Enter requried data
All field filled correctly ?
No
Yes
Approved by Tenant ?
No
Yes
Approved by Landlord ?
No
Yes
Activated Agreement
Terminate Agreement

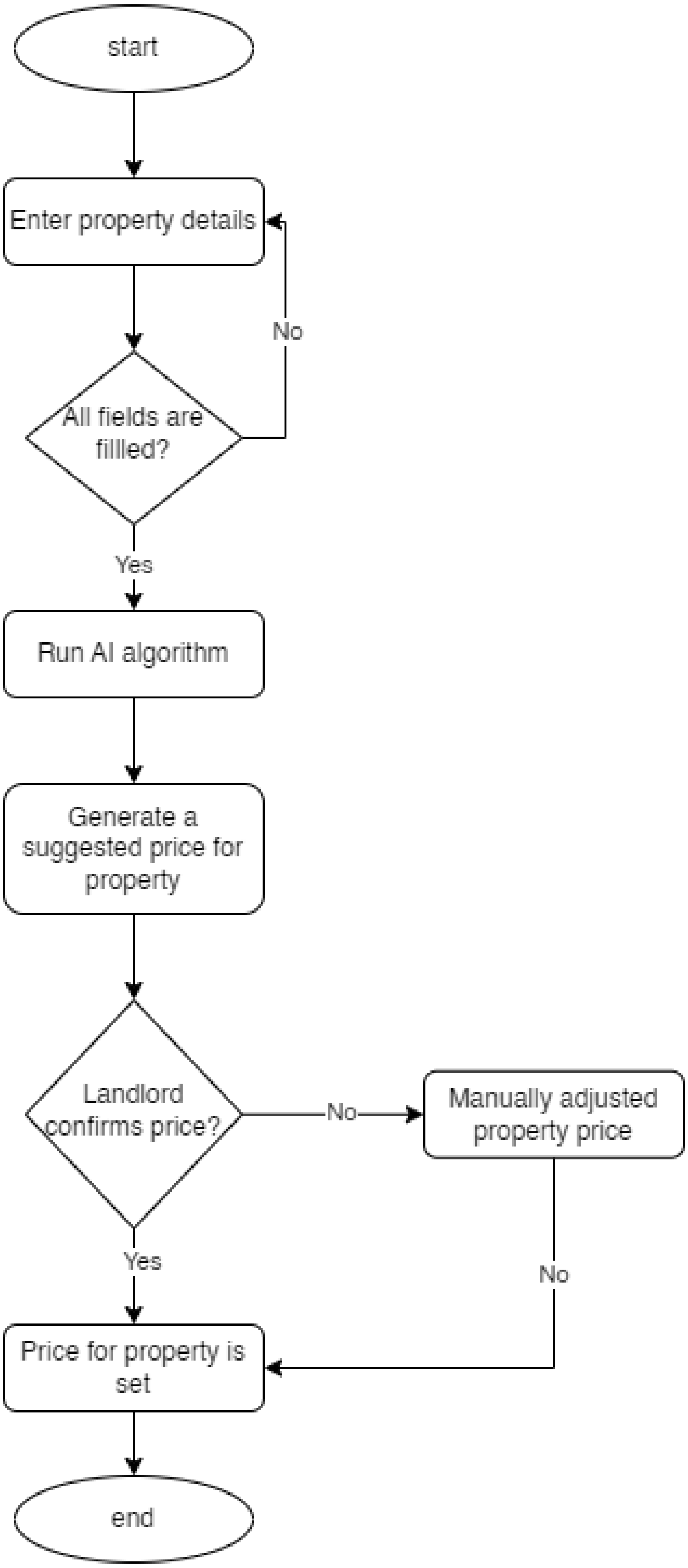
UC3


Start
Select User
View Reviews
Give Review
Display all available reviews of the user
Review form is displayed
All fields are fillled?
Review is added to user profile
end

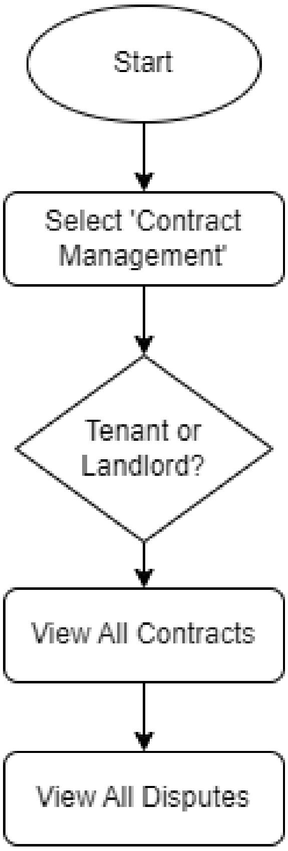
UC4


Start
User clicks on "Verify now" option
User uploads the image and clicks on "Upload button"
File format is supported?
No
Error Message: File format not supported. Try again.
Yes
System processes the file
Number is generated?
No
Error Message: Error occured, photo not clear or wrong photo. Please upload again.
Yes
Details are stored and user is verified
End

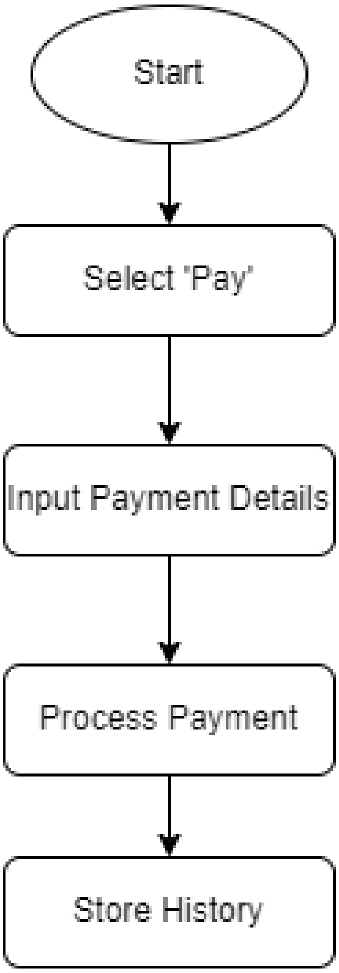
UC5


start
Enter property details
All fields are fillled?
No
Yes
Run AI algorithm
Generate a suggested price for property
Landlord confirms price?
No
Manually adjusted property price
Yes
No
Price for property is set
end

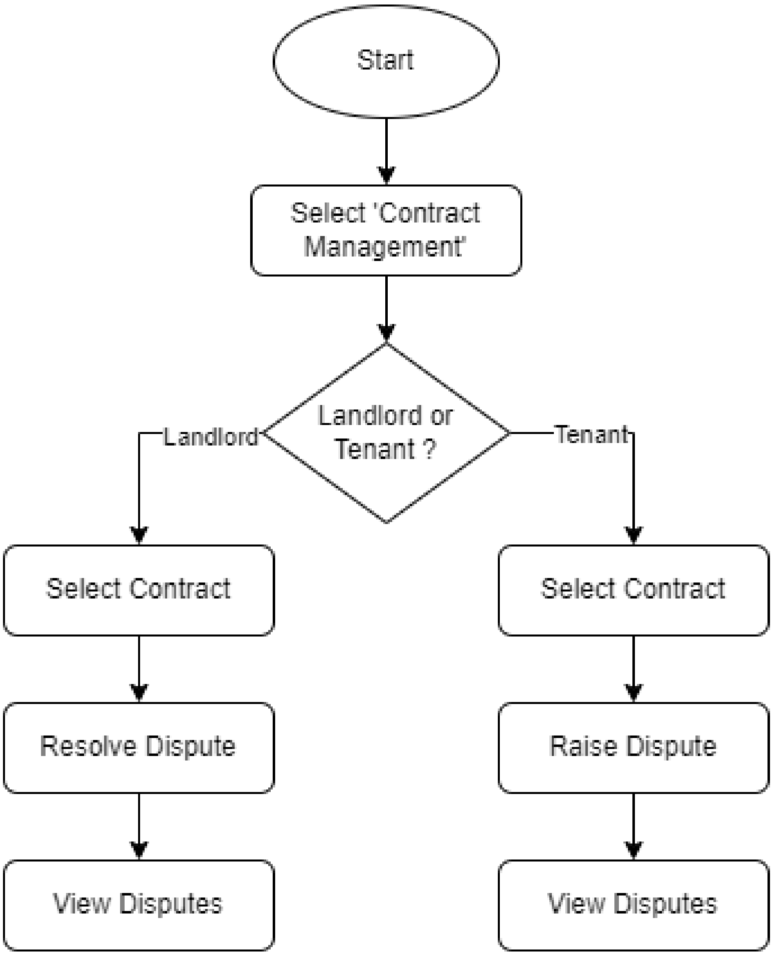
UC6


Start
Select 'Contract Management'
Tenant or Landlord?
View All Contracts
View All Disputes

UC7

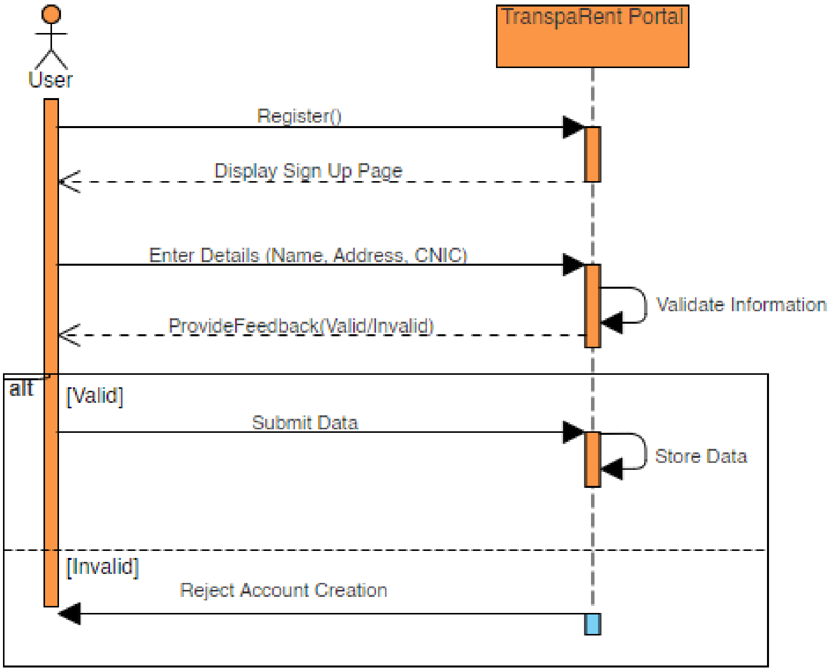

Start
Select 'Pay'
Input Payment Details
Process Payment
Store History

UC8

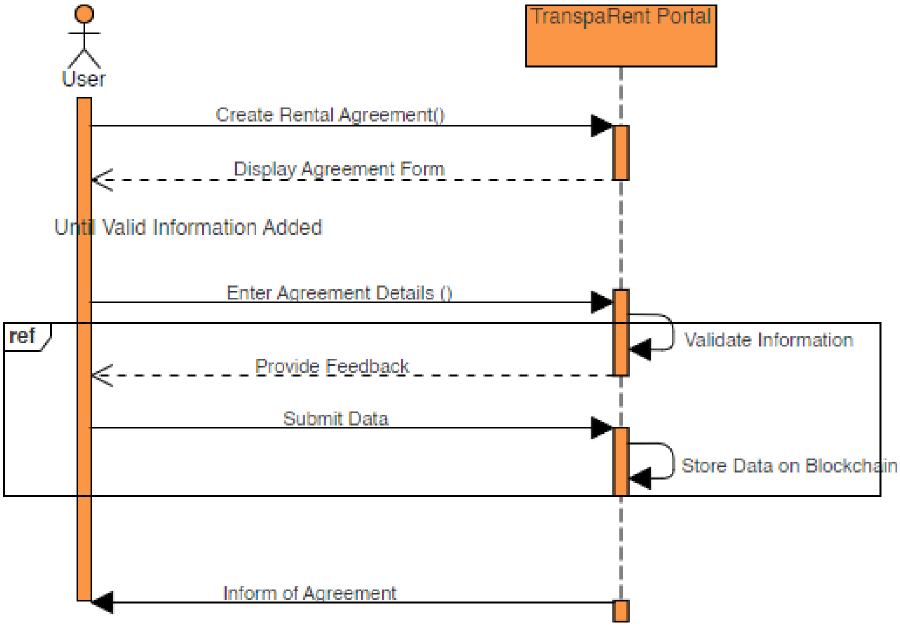

Start
Select 'Contract Management'
Landlord or Tenant ?
Landlord
Tenant
Select Contract
Select Contract
Resolve Dispute
Raise Dispute
View Disputes
View Disputes

System Sequence Diagrams

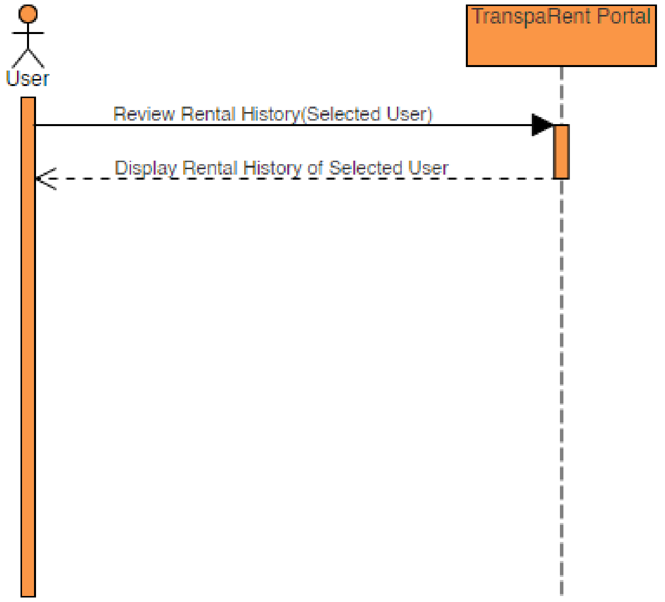
UC01: User Registration and Authentication



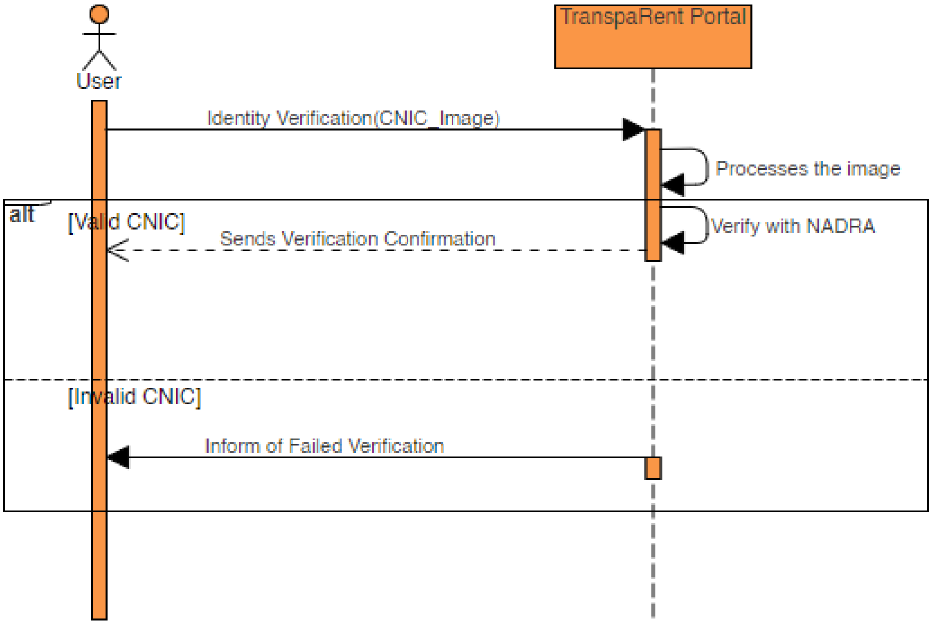
UC02: Create a Rental Agreement



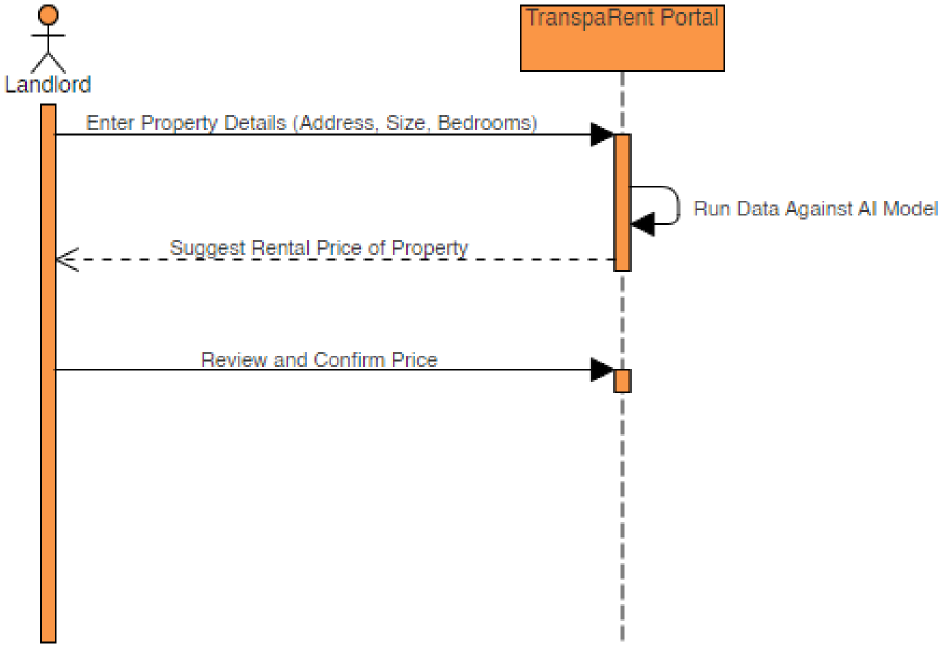
UC03: Review Rental History and Creditworthiness


User
TranspaRent Portal
Review Rental History(Selected User)
Display Rental History of Selected User

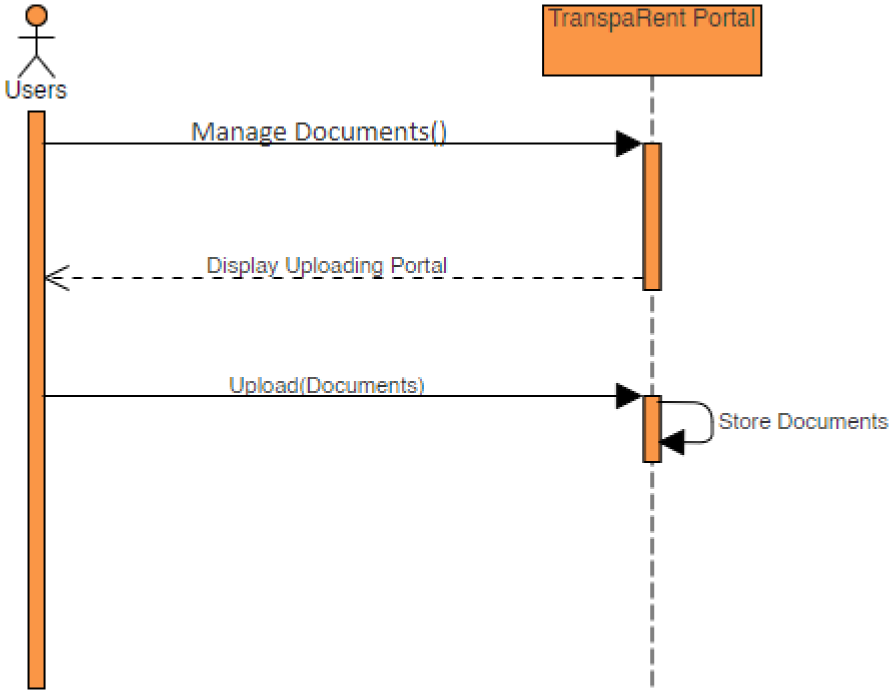
UC04: Identity Verification


User
TranspaRent Portal
Identity Verification(CNIC_Image)
Processes the image
alt
[Valid CNIC]
Verify with NADRA
Sends Verification Confirmation
[Invalid CNIC]
Inform of Failed Verification

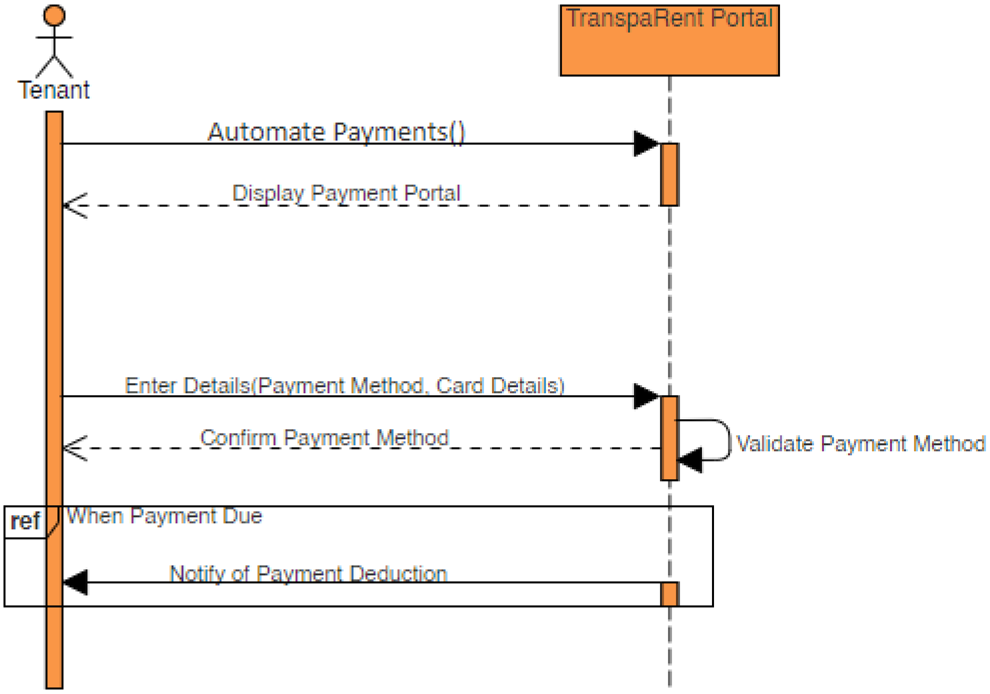
UC05: Get AI-Based Pricing Suggestions


Landlord
TranspaRent Portal
Enter Property Details (Address, Size, Bedrooms)
Run Data Against AI Model
Suggest Rental Price of Property
Review and Confirm Price

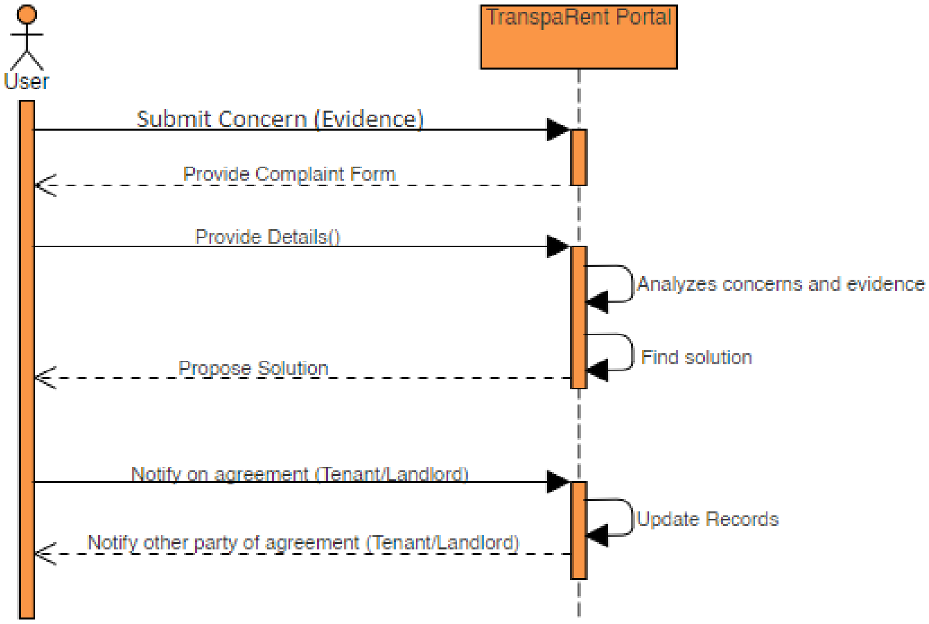
UC06: Manage Rental Documents



UC07: Automate Rent Payments

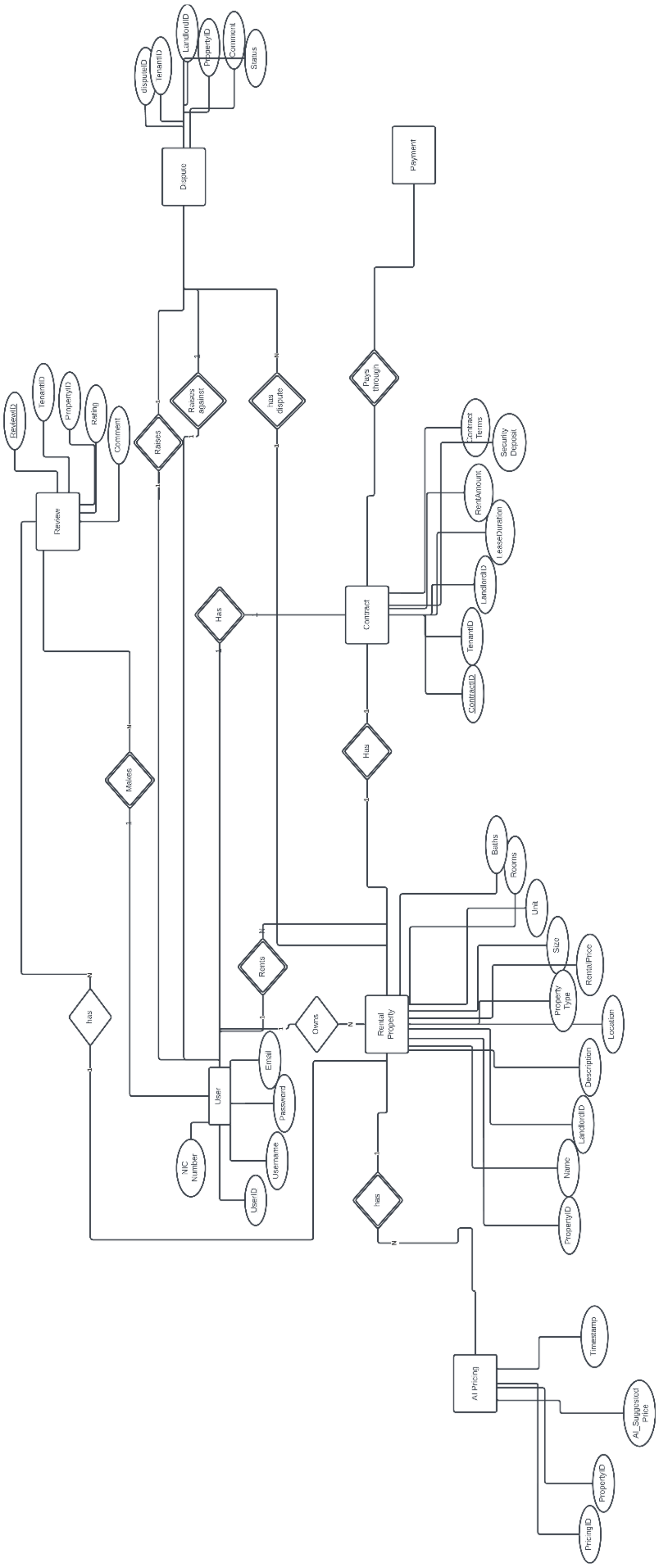


UC08: Transparent Dispute Resolution


User
TranspaRent Portal
Submit Concern (Evidence)
Provide Complaint Form
Provide Details()
Analyzes concerns and evidence
Find solution
Propose Solution
Notify on agreement (Tenant/Landlord)
Update Records
Notify other party of agreement (Tenant/Landlord)

3.3 Data Design

ER Diagram



Chapter 4

Implementation and Testing

4.1 Algorithm Design

User Registration and Authentication

- Pseudocode

```
Algorithm RegisterUser(email, password, userType)
  Input: email, password, userType
  Output: registrationStatus
  Begin
    if isValidEmail(email) and isValidPassword(password) then
      hashedPassword = hash(password)
      if userType == "Landlord" then
        saveToDatabase(email, hashedPassword, userType)
      else if userType == "Tenant" then
        saveToDatabase(email, hashedPassword, userType)
      end if
      return "Registration Successful"
    else
      return "Invalid Email or Password"
    end if
  End
```

- The algorithm takes an email, password, and user Type as inputs.
- It first validates the email and password.
- If valid, the password is hashed for security.
- Based on the user Type, the email and hashed password are saved to the database.
- The function returns a success message if registration is successful, otherwise an error message.

Create Rental Agreement

- Pseudocode

```
Algorithm CreateRentalAgreement(landlordID, tenantID, propertyDetails, rentAmount, duration, terms)
  Input: landlordID, tenantID, propertyDetails, rentAmount, duration, terms
  Output: agreementStatus
  Begin
    if validateDetails(propertyDetails, rentAmount, duration, terms) then
      agreement = generateAgreement(landlordID, tenantID, propertyDetails, rentAmount, duration, terms)
      blockchainStore(agreement)
      return "Agreement Created Successfully"
    else
```

```
        return "Validation Failed"
      end if
    End
```

- Inputs include landlord and tenant IDs, property details, rent amount, duration, and terms.
- The algorithm validates the input details.
- If valid, it generates a rental agreement and stores it on the blockchain.
- Returns a success message if the agreement is created, otherwise an error message.

AI-Based Pricing Suggestions

- Pseudocode

```
Algorithm AIPricingSuggestion(propertyDetails)
    Input: propertyDetails
    Output: suggestedPrice
    Begin
      historicalData = fetchHistoricalData(propertyDetails)
      model = trainModel(historicalData)
      suggestedPrice = model.predict(propertyDetails)
      return suggestedPrice
    End
```

- Takes property details as input.

- Fetches historical data relevant to the property.
- Trains a model using this historical data.
- The trained model predicts a suggested rental price for the property.
- Returns the suggested price.

4.2 External APIs/SDKs

API and Version	Description	Purpose of Usage	API
'web3' ^4.5.0	Ethereum JavaScript API.	Interacting with Ethereum blockchain and smart contracts.	'Web3()', 'web3.eth', 'web3.utils'

4.3 Testing Details

Once the system has been successfully developed, testing has to be performed to ensure that the system working as intended. This is also to check that the system meets the requirements stated earlier. Besides that, system testing will help in finding the errors that may be hidden from the user. The testing must be completed before it is deployed for use.

4.3.1 Unit Testing

Unit Test Case 1: Register User with Valid and Invalid Data

Objective: Ensure the registration form works correctly with valid and invalid data.

Steps:

1. Enter valid email, password, and userType.
2. Submit the form.
3. Verify successful registration.
4. Enter invalid email or password.
5. Submit the form.
6. Verify appropriate error message.

Unit Test Case 2: Create Rental Agreement with Valid and Invalid Details

Objective: Ensure the rental agreement creation works with valid and invalid details.

Steps:

1. Enter valid landlordID, tenantID, property details, rent amount, duration, and terms.
2. Submit the form.
3. Verify agreement creation and storage on the blockchain.
4. Enter invalid property details or terms.
5. Submit the form.
6. Verify appropriate validation error message.

Unit Test Case 3: AI Pricing Suggestion Accuracy

Objective: Ensure the AI model suggests accurate rental prices based on historical data.

Steps:

1. Enter property details.
2. Fetch historical data.
3. Train the model.
4. Verify the predicted price against expected price ranges.

Unit Test Case 4: Login as User with Valid and Invalid Credentials

Objective: Ensure the login form works correctly with valid and invalid credentials.

Steps:

1. Enter valid credentials and submit.
2. Verify successful login.
3. Enter invalid credentials and submit.
4. Verify appropriate error message.

Bibliography

A Complete Guide on Tenant and Landlord Rights in Pakistan. (n.d.). Retrieved from validlaws.com: https://validlaws.com/a-complete-guide-on-tenant-and-landlord-rights-in-pakistan/

Bhandari, A. (2020, November 28). Build your own Optical Character Recognition (OCR) System using Google's Tesseract and OpenCV. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/05/build-your-own-ocr-google-tesseractopencv/

crackingdemon. (2021, September 2). Register and Login System in MERN Stack. Retrieved from dev.to: https://dev.to/crackingdemon/register-and-login-system-in-mern-stack-1n98

Login and Registration Form with MERN stack. (2023, July 29). Retrieved from React.js Examples: https://reactjsexample.com/login-and-registration-form-with-mern-stack/

MUHAMMAD ZAIN UL ABIDEEN, M. T. (2022). Pakistani CNIC OCR System. Wah: CUI.

QUÂN, M. (2021). Extract data from image of an Id card. OCR. Retrieved from Kaggle: https://www.kaggle.com/code/minhquan23102000/extract-data-from-image-of-an-id-cardocr

Rehan, A. (2022, March 1). Here are the applicable tenant rights in Pakistan. Retrieved from Graana.com: https://www.graana.com/blog/tenant-rights-in-pakistan/

SAMUEL, F. S. (2023, May 15). How to Secure Your MERN Stack App with JWT-Based User Authentication and Authorization. Retrieved from freeCodeCamp: https://www.freecodecamp.org/news/how-to-secure-your-mern-stack-application/

Shrestha, A. (2020, August 30). User Authentication in MERN Stack — Part 1 (Backend). Retrieved from Medium: https://medium.com/swlh/user-authentication-using-mern-stack-part-1-backend-cd4d193f15b1

Zameen.com Property Data Pakistan. (2021, February 6). Kaggle. https://www.kaggle.com/datasets/huzzefakhan/zameencom-property-data-pakistan