UITS

utu ill b b an thy past

An initiative of PHP Family

University of Information Technology & Sciences

# CSE 356

# Software Engineering and System Analysis Lab

## Project Report


**Submitted By**

Sazzad Hossain (0432220005101103)

A. O. M. Ramim Chowdhury (0432220005101146)

Suraia Akter Mim (0432220005101108)

Samin Yeaser Rafid (043231000510110)

**Submitted To**

Khandoker Nosiba Arifin

Lecturer

Department of CSE, UITS


CSE

University of Information Technology and Sciences

## 1. Executive Summary

The House Renting System is a comprehensive web-based platform designed to streamline the rental property management process. This system connects landlords, property managers, and tenants through an efficient online platform that automates property listing, tenant screening, application processing, and lease management.

- **Comprehensive Platform:** Successfully designed a full-featured rental management system
- **Multi-User Support:** Implemented role-based access for tenants, landlords, and administrators
- **Automated Processes:** Streamlined rental workflows from property listing to lease generation
- **Security Focus:** Integrated multi-factor authentication and secure data handling
- **Mobile Responsive:** Ensured accessibility across all device types

**Technical Stack:**

- **Frontend:** HTML5, CSS, Bootstrap, JavaScript
- **Backend:** JavaScript (Node.js)
- **Database:** MySQL 8.0+
- **Development Environment:** VS Code, XAMPP
- **Third-party Integrations:** Background check services, payment gateways

## 2. Project Overview

### 2.1 Problem Statement

Traditional property rental processes involve manual paperwork, time-consuming property searches, and inefficient communication between landlords and tenants. These challenges result in operational inefficiencies, delayed processes, and poor user experiences.

### 2.2 Solution Approach

Our House Renting System addresses these challenges by providing:

- **Centralized Platform:** Single point of access for all rental activities
- **Automated Workflows:** Streamlined processes from property listing to lease signing
- **Enhanced Communication:** Integrated messaging and notification systems
- **Data-Driven Insights:** Analytics and reporting capabilities for informed decision-making

### 2.3 Target Users

- **Primary Users:** Tenants (property seekers), Landlords (property owners), Property Managers
- **Secondary Users:** System Administrators
- **Age Range:** 18-70 years with varying technical expertise levels
- **Usage Context:** Multi-device access (desktop, tablet, mobile)

## 3.2 System Components

### 3.2.1 Frontend Components

- **User Interface Layer:** Responsive web interfaces for different user roles
- **Client-Side Logic:** JavaScript for dynamic interactions and form validation
- **Styling Framework:** Bootstrap for consistent and mobile-responsive design

### 3.2.2 Backend Components

- **Authentication Module:** User registration, login, and session management
- **Property Management Module:** CRUD operations for property listings
- **Application Processing Module:** Rental application workflow management
- **Communication Module:** Messaging and notification systems
- **Integration Module:** Third-party service integrations

### 3.2.3 Database Components

- **User Management:** User profiles, authentication data, and preferences
- **Property Catalog:** Property listings, images, and availability status
- **Application Tracking:** Rental applications and processing status
- **Communication Logs:** Message history and notification records

# 4. Functional Requirements Analysis

### 4.1 Core Functional Requirements

**FR-1: User Registration and Authentication**

- **Description:** Secure user account creation and login system
- **Priority:** High
- **Actors:** All Users
- **Inputs:** Email, password, user type, personal information
- **Outputs:** User account, authentication tokens
- **Business Rules:**

    - Unique email addresses
    - Password complexity requirements
    - Email verification mandatory

### FR-2: Property Management

- **Description:** Comprehensive property listing and management system
- **Priority:** High
- **Actors:** Landlords, Property Managers
- **Inputs:** Property details, images, rental terms, pricing
- **Outputs:** Published listings, management interface
- **Business Rules:**
    - Mandatory property information fields
    - Image upload limits and formats
    - Automatic availability status updates

### FR-3: Property Search and Discovery

- **Description:** Advanced property search with filtering capabilities
- **Priority:** High
- **Actors:** Tenants
- **Inputs:** Search criteria, filters, location preferences
- **Outputs:** Filtered property results, detailed property pages
- **Business Rules:**
    - Real-time availability checking
    - Location-based search radius
    - Price range validation

### FR-4: Rental Application Processing

- **Description:** Online application submission and tracking system
- **Priority:** High
- **Actors:** Tenants, Landlords
- **Inputs:** Application data, supporting documents
- **Outputs:** Application confirmations, status updates

- **Business Rules:**
    - Complete application requirements
    - Document format restrictions
    - Application status workflow

### 4.2 Secondary Functional Requirements

#### FR-5: Communication System

- **Description:** Integrated messaging between users
- **Priority:** Medium
- **Features:** Direct messaging, automated notifications, email integration

#### FR-6: Maintenance Request Management

- **Description:** Tenant maintenance request submission and tracking
- **Priority:** Medium
- **Features:** Request categorization, priority levels, status tracking

#### FR-7: Reporting and Analytics

- **Description:** Data insights and performance reporting
- **Priority:** Low
- **Features:** Property performance metrics, user activity reports, financial summaries

## 6.2 Database Schema Details

### 6.2.1 Key Design Decisions

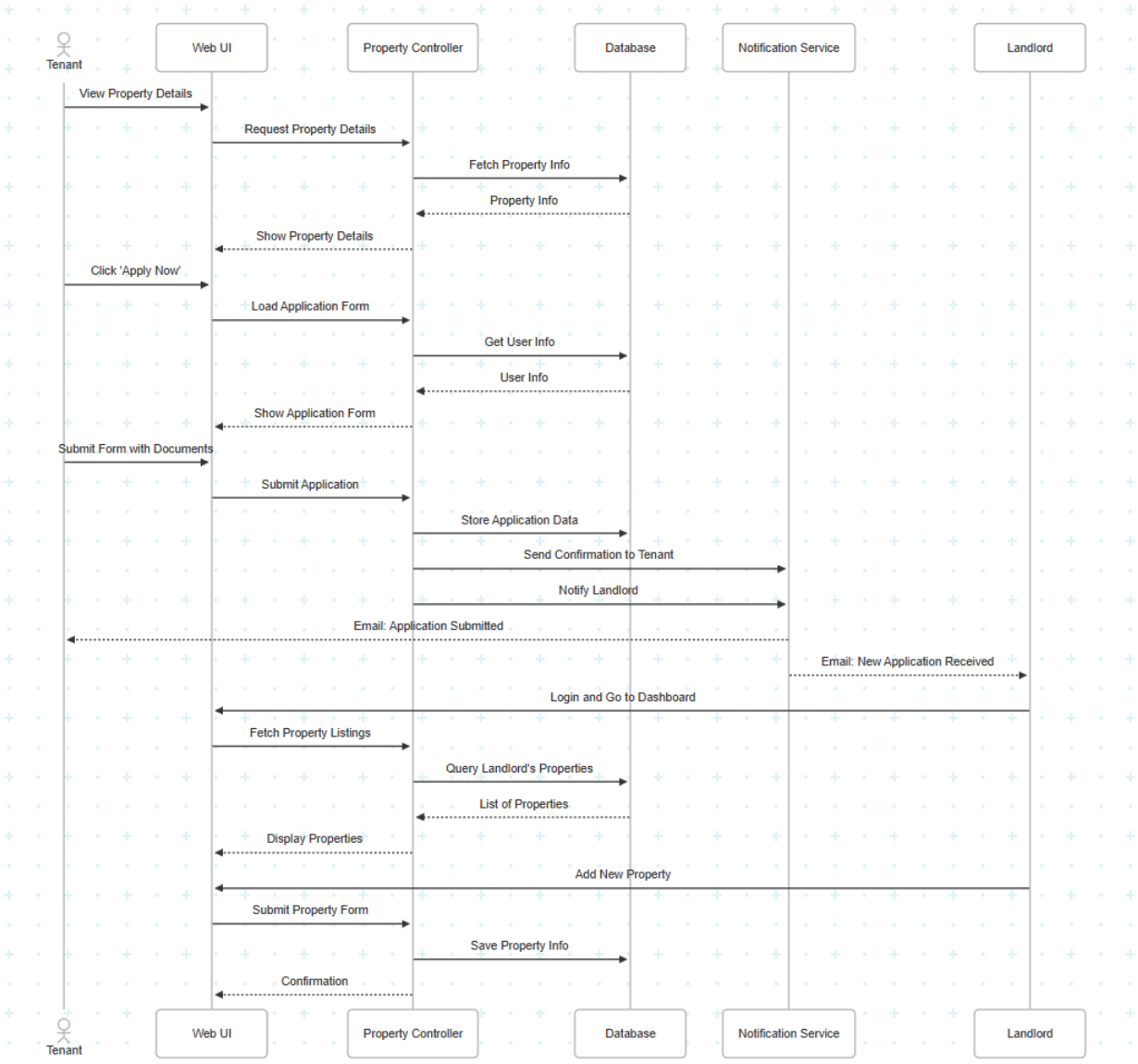
- **Primary Keys:** UUID-based for better scalability and security
- **Foreign Keys:** Properly defined relationships with cascade options
- **Data Types:** Optimized for performance and storage efficiency
- **Indexes:** Strategic indexing on frequently queried columns
- **JSON Fields:** Used for flexible, schema-less data storage

#### 6.2.2 Data Integrity Measures

- **Constraints:** NOT NULL, UNIQUE, and CHECK constraints
- **Referential Integrity:** Foreign key relationships maintained
- **Data Validation:** Application-level and database-level validation
- **Backup Strategy:** Daily incremental, weekly full backup.

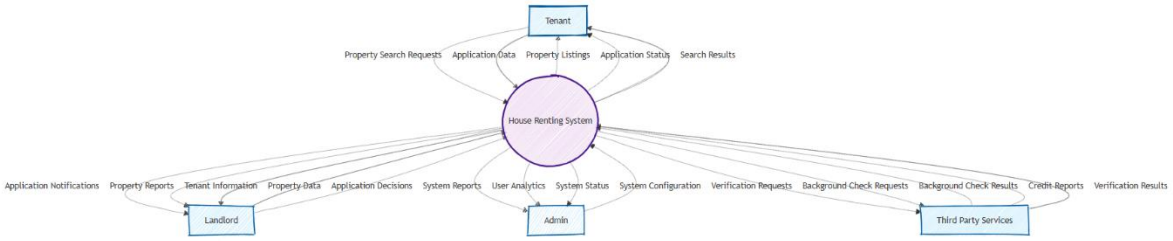
# 5. System Design

## 5.1 Use sequence Diagram

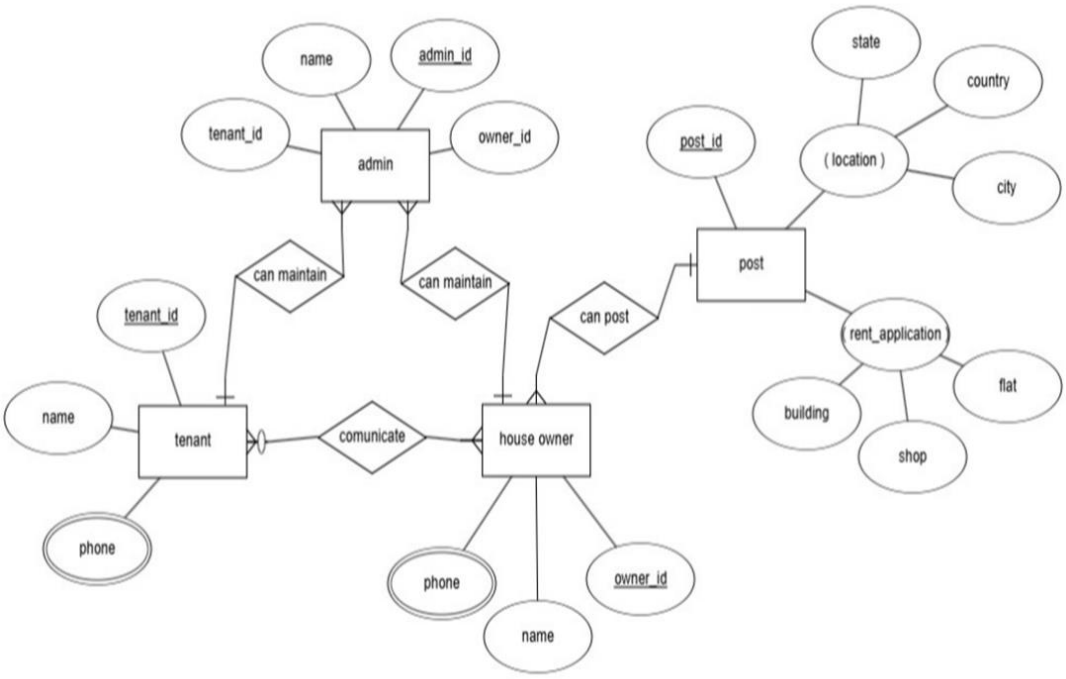


## 5.1 Class Diagram:

## 5.1 Data Flow Diagram:

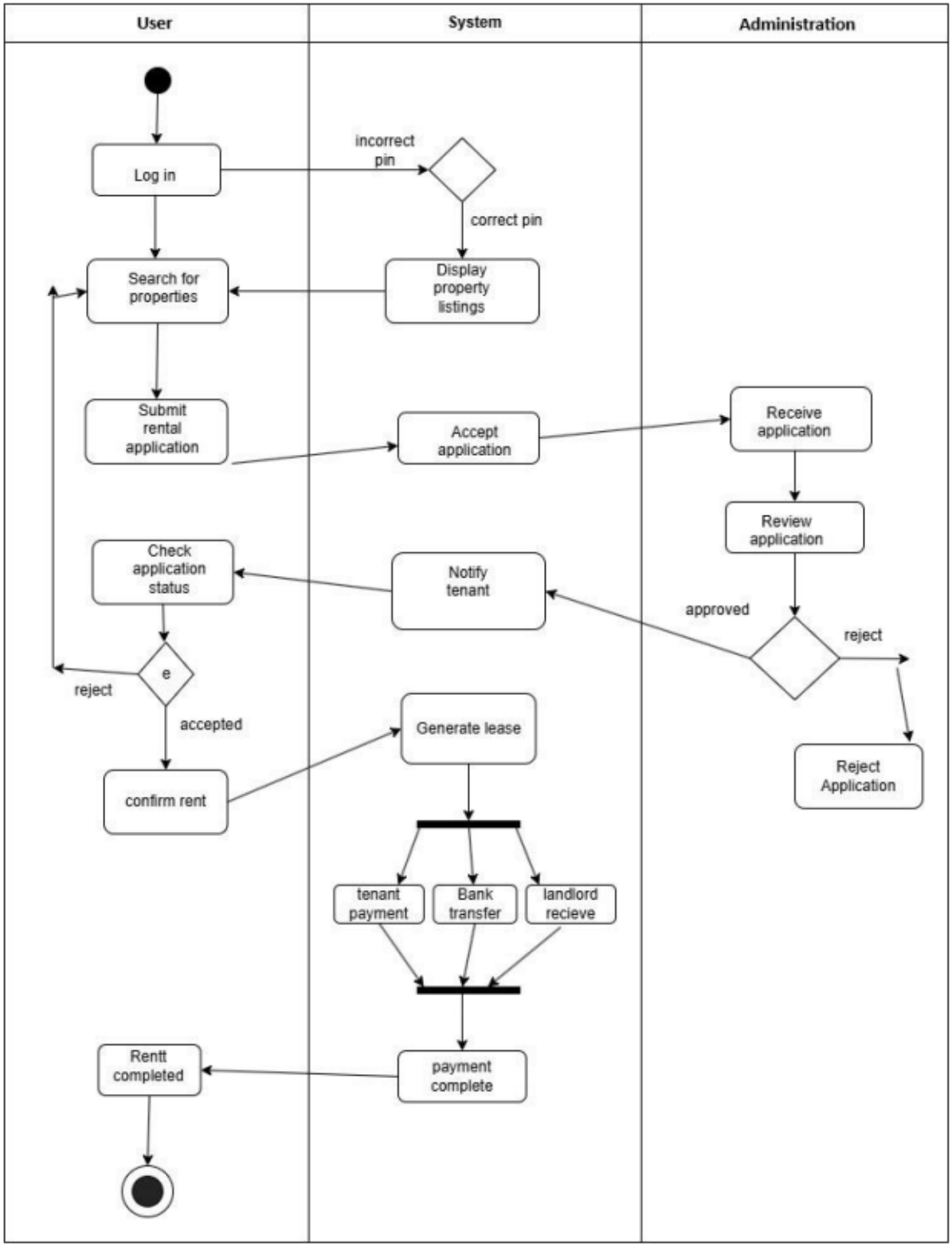



## 5.1 ER Diagram:

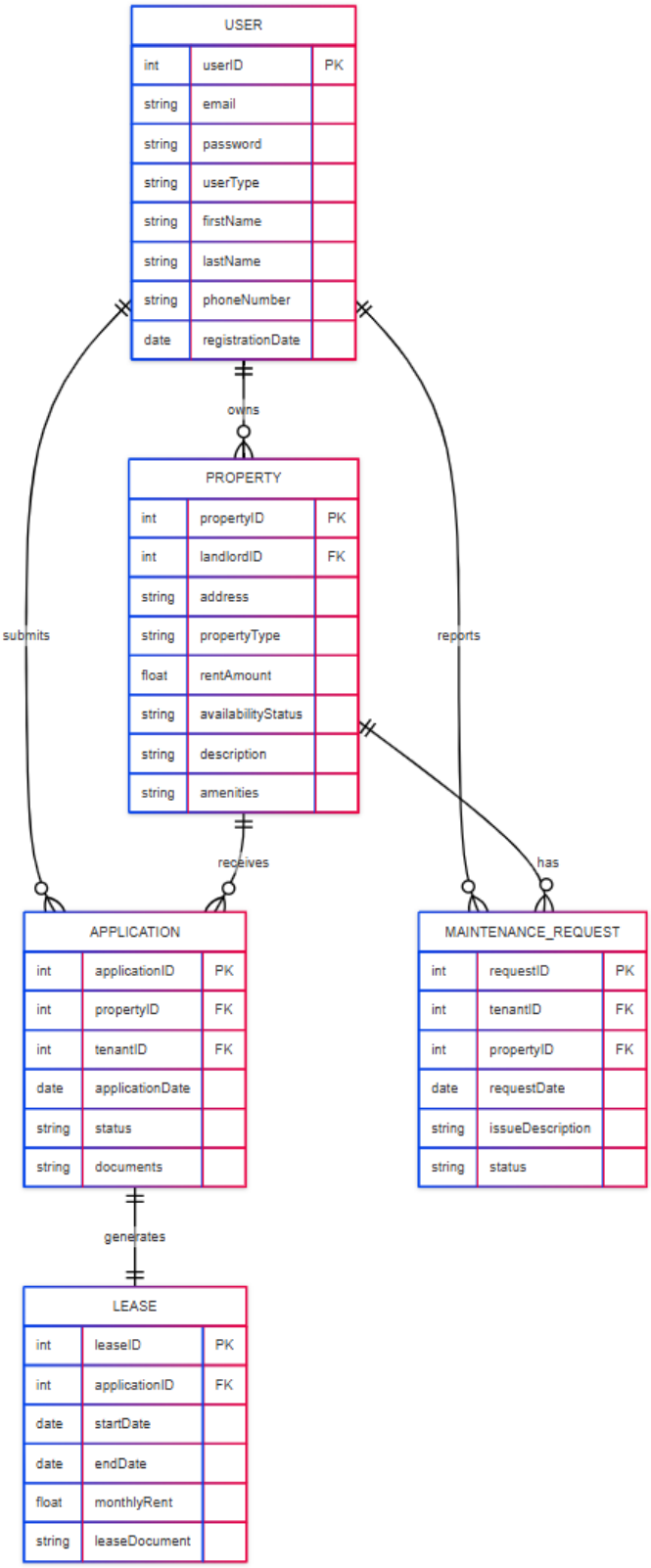


## 5.1 Use Case Diagram:

## 5.1 Activity Diagram:



## 5.1 Activity Diagram:



## 7. User Interface Design

### 7.1 Design Principles

- **Responsive Design:** Mobile-first approach using Bootstrap framework
- **User-Centered Design:** Intuitive navigation and clear information hierarchy
- **Accessibility:** WCAG 2.1 AA compliance for users with disabilities
- **Consistency:** Uniform design language across all interfaces
- **Performance:** Optimized loading times under 3 seconds

## 8. Testing Strategy

### 8.1 Testing Methodology

### 8.2 Testing & Results

**We tested various features under different conditions:**

| Feature | Test Case | Result |
|---|---|---|
| **Login** | **Valid credentials** | **Passed** |
| **Property Search** | **Filter by price/location** | **Passed** |
| **Application Submission** | **Valid input** | **Passed** |
| **Maintenance Request** | **Submit & update status** | **Passed** |

**All modules passed unit and integration tests.**

**10. Conclusion:** The system successfully digitalizes the property rental workflow, making it easier for landlords and tenants to interact. It minimizes manual work, improves communication, and brings transparency to the renting process.