
SOFTWARE REQUIREMENTS SPECIFICATION

For

Tenant–Landlord Information and Management System (TLIMS)

Prepared by:

E.T.C Extra-Terrestrial Coders

**Escoreal, Drexzel C.
Ticmon, Mariel M.
Corcelles, Jenneth D.**

**Submitted in fulfillment requirements for Integrative
Programming and Technologies 2 (IT305)**

August 16, 2025

Table of Contents

Table of Contents	ii
1. Introduction	
1.1 Purpose	1
1.2 Intended Audience	1
1.3 Product Scope	1
1.4 Definitions, Acronyms, and Abbreviations	1
2. Overall Description	2
2.1 User Characteristics	2
2.2 Constraints	3
2.3 Assumptions and Dependencies	3
3. Requirements Specifications	4
3.1 Functional Requirements	4
3.2 Non-Functional Requirements	4
3.3 External Interface Requirements	6
3.4 System Models	6
3.5 Database Design	7
3.6 Implementation	7

1. Introduction

1.1 Purpose

The Tenant–Landlord Information and Management System (TLIMS) is a web-based platform designed to simplify the process of finding and renting boarding houses, apartments, and rooms near J.H. Cerilles State College (JHCSC) - Pagadian Campus for students, while enabling landlords to post and manage their properties. The system integrates location mapping, notifications, and secure account management to enhance efficiency, transparency, and communication in the rental process. TLIMS aims to provide JHCSC-Pagadian students with a convenient way to search for accommodations near the campus and landlords with a platform to advertise their boarding houses, apartments, or rooms, facilitating direct communication and streamlined transaction management.

1.2 Intended Audience

- **Students/Tenants:** JHCSC-Pagadian students seeking boarding houses, apartments, or rooms near the campus.
- **Landlords:** Property owners near JHCSC-Pagadian who wish to rent out boarding houses, apartments, or individual rooms to students.

1.3 Product Scope

TLIMS provides a comprehensive platform for JHCSC-Pagadian students to search, filter, and view listings of boarding houses, apartments, or rooms near the campus. It offers landlords a dashboard to manage property postings, including pricing, availability, and photos. The system includes secure registration and login functionality for tenants and landlords to protect user data. Integration with Google Maps API enables accurate display of property locations, and notification features keep users informed about new listings, inquiries, bookings, and updates. TLIMS streamlines communication and transactions between tenants and landlords, making the rental process organized, transparent, and user-friendly.

1.4 Definitions, Acronyms, and Abbreviations

- **TLIMS:** Tenant–Landlord Information and Management System
- **Tenant:** A JHCSC-Pagadian student seeking to rent a boarding house, apartment, or room.
- **Landlord:** A property owner near JHCSC-Pagadian offering boarding houses, apartments, or rooms for rent.
- **API:** Application Programming Interface
- **GUI:** Graphical User Interface
- **JHCSC:** J.H. Cerilles State College - Pagadian Campus

2. Overall Description

TLIMS is a web-based application that simplifies the rental process for JHCSC-Pagadian students and landlords. It allows landlords to post listings for boarding houses, apartments, or rooms, including details such as location, price, and photos. Students can browse, search, and filter these listings based on preferences and contact landlords directly. The system enhances rental record management, and notifications ensuring an efficient and transparent experience.

2.1 User Characteristics

- **Students/Tenants:** JHCSC-Pagadian students with basic computer and internet skills, interested in renting accommodations near the campus.
- **Landlords:** Property owners familiar with posting information online, managing rental properties, and providing accurate details about their boarding houses, apartments, or rooms.

2.2 Constraints

- The system requires an internet connection for web access.
- Google Maps API usage is subject to API key limitations.
- Users must have valid email addresses to register.
- Data storage is limited by the hosting server capacity.

2.3 Assumptions and Dependencies

- Users have access to devices (e.g., desktops, laptops, or mobile devices) running modern web browsers.
- Landlords will provide accurate and up-to-date information about their properties.
- Students and landlords will actively use notification features for communication.
- System performance depends on the availability of the Google Maps API and server uptime.

3. Specific Requirements

3.1 Functional Requirements

- **User Registration & Login:** Secure account creation with role-based access for tenants and landlords.
- **Property Posting (Landlord):** Landlords can add, edit, or delete listings for boarding houses, apartments, or rooms, including details like price, description, and photos.
- **Property Listings (Student):** Students can browse, search, and filter listings based on preferences (e.g., price, proximity to JHCSC, availability).
- **Map Integration:** Display accurate locations of properties using Leaflet API.
- **Notifications:** Send alerts for new listings, inquiries, bookings, or updates.
- **Inquiries:** Allow students to send inquiries to landlords about specific listings.
- **Booking:** Enable students to book rooms, with landlords managing booking confirmations.

3.2 Non-functional Requirements

- **Performance:** Search and filter operations respond within 2-3 seconds.
- **Security:** Implement encrypted login (e.g., bcrypt for passwords) and secure database handling to protect user data.
- **Usability:** Provide a simple, intuitive, and responsive interface for both desktop and mobile users.
- **Reliability:** Ensure system uptime of at least 95%.
- **Scalability:** Support an increasing number of users and listings, starting with the JHCSC-Pagadian community.

3.3 External Interface Requirements

3.3.1 User Interfaces

- **Student Interface:** A web-based dashboard for students to search, filter, view property details, send inquiries, and manage bookings.
- **Landlord Interface:** A dashboard for landlords to add, update, or remove property listings and manage inquiries and bookings.
- **Graphical User Interface (GUI):** Responsive design compatible with desktop and mobile browsers, ensuring ease of use.

3.3.2 Hardware Interfaces

- **Client Devices:** Desktops, laptops, and mobile devices with internet access and modern web browsers.
- **Server:** A web server with sufficient processing power, RAM, and storage to support concurrent users.

3.3.3 Software Interfaces

- **Operating System Compatibility:** Compatible with Windows, Linux, and Android/iOS (via browsers).
- **Database:** MySQL or similar RDBMS for structured data storage.
- **Web Technologies:** HTML, CSS, JavaScript (front-end), and PHP or Node.js (back-end).
- **External APIs:** Map Integration (Leaflet) API for location-based services.

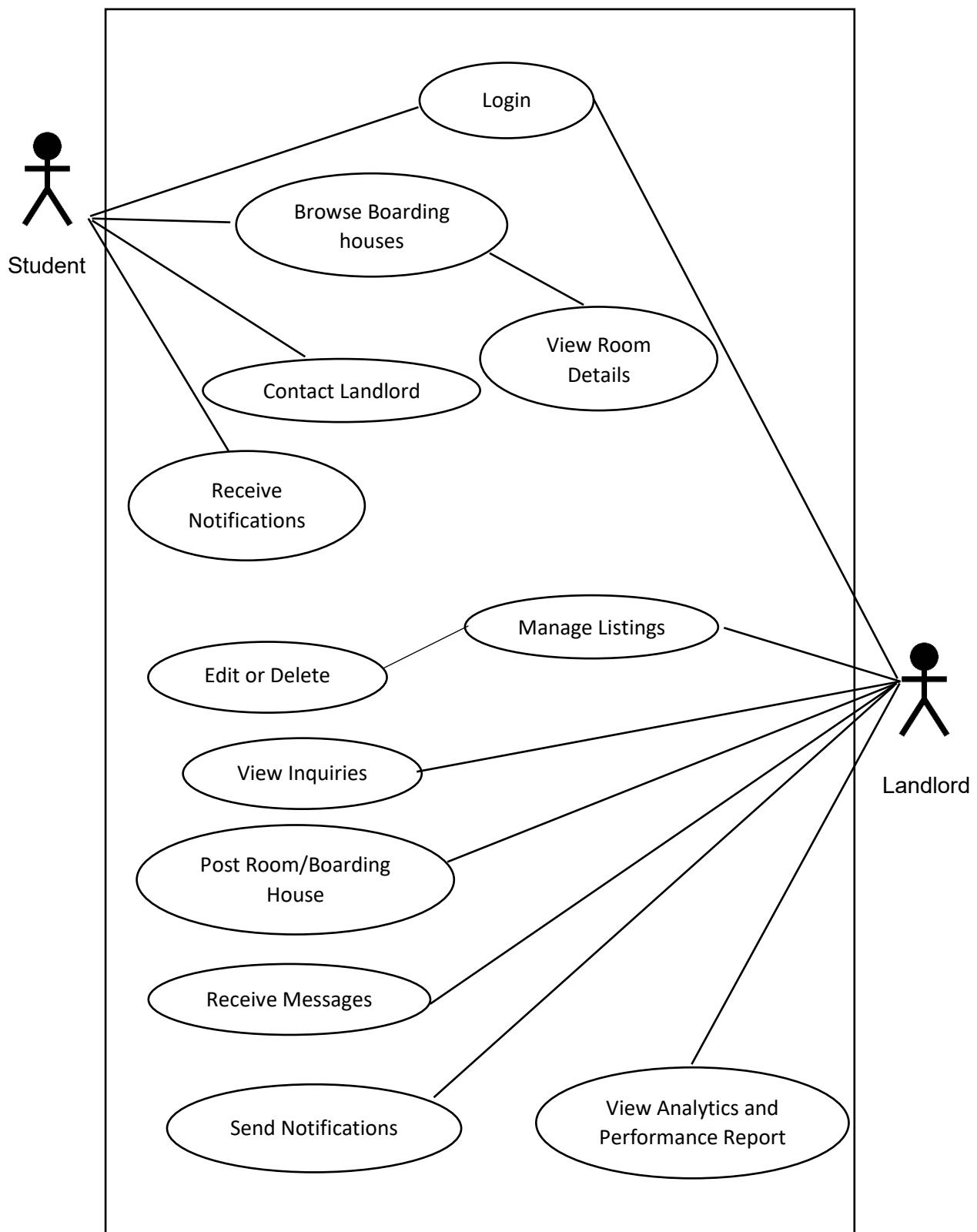
3.3.4 Communication Protocols

- **HTTP/HTTPS:** For secure client-server communication.
- **TCP/IP:** For reliable data transmission over the internet.

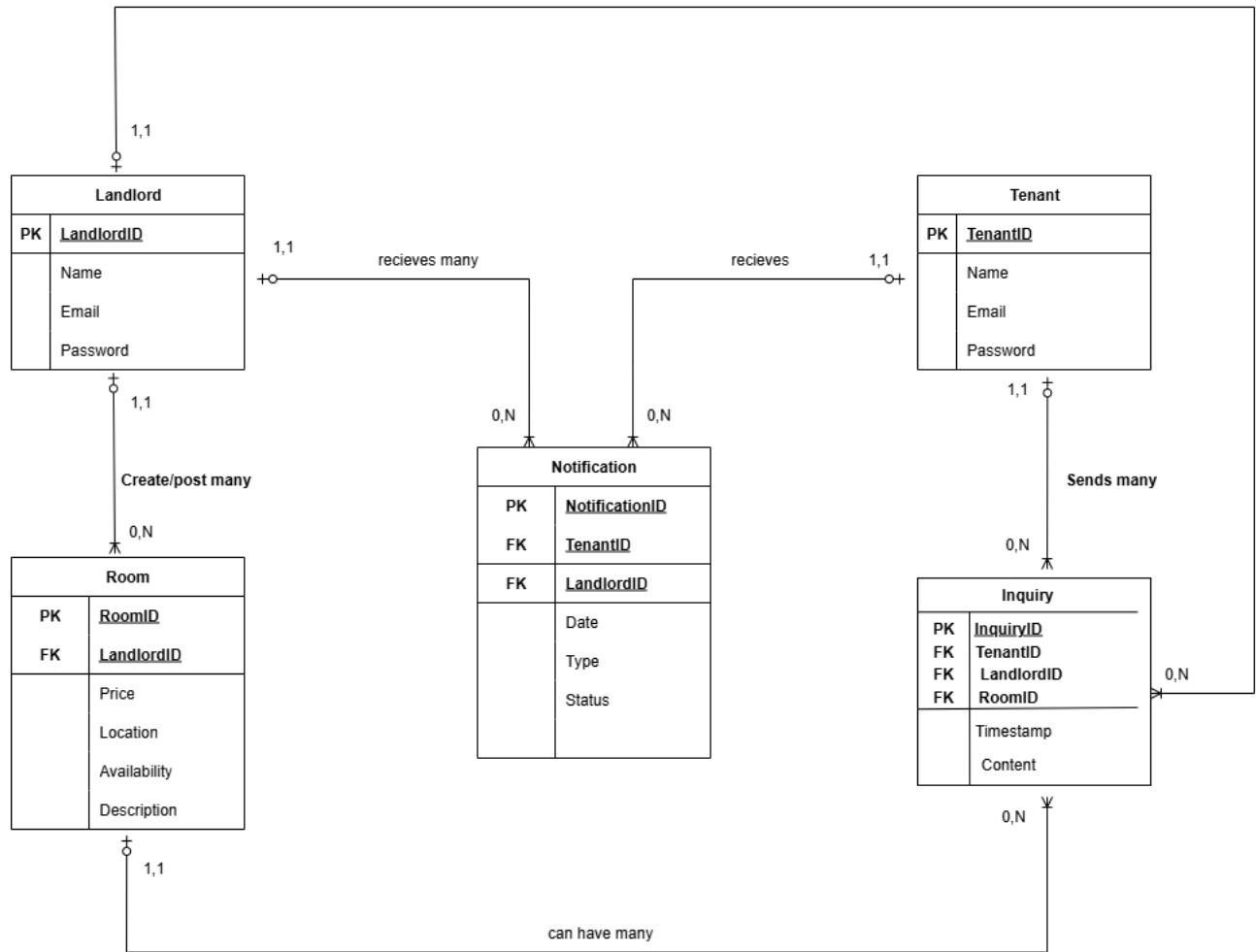
3.4. System Models

This section outlines the core models for TLIMS, including Use Case Diagram and ERD(Entity-Relationship Diagram).

Use Case Diagram



Entity–Relationship Diagram (ERD)

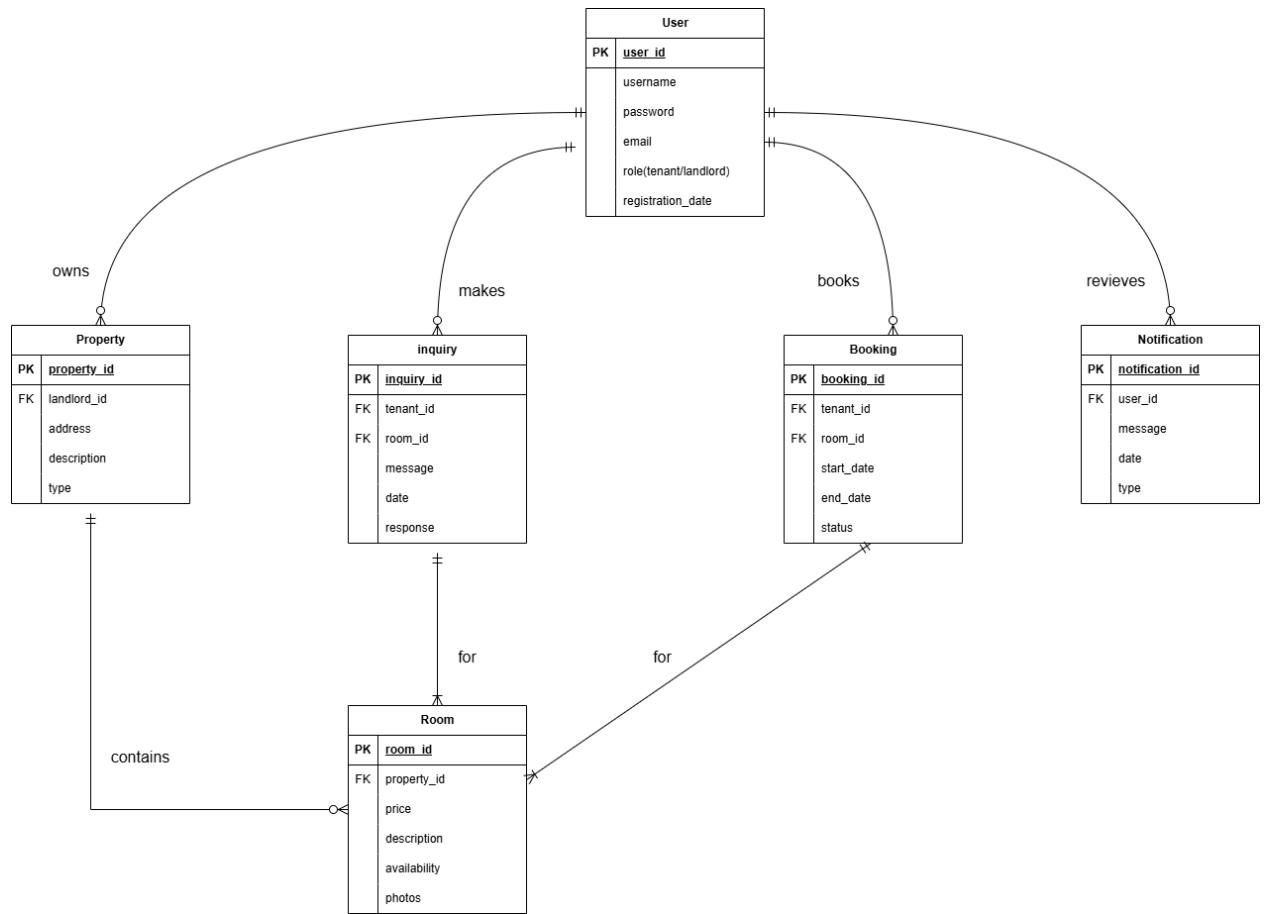


3.5 Database Design

Based on feedback from the previous meeting, the Entity-Relationship Diagram (ERD) has been revised to include Booking entities to support rental transactions and record-keeping, ensuring streamlined management as stated in the purpose. The database is normalized to the third normal form (3NF) to minimize redundancy.

Revised Entity-Relationship Diagram (ERD)

The ERD is described in text format below, with entities, attributes, and relationships. A visual diagram will be developed later as requested.



The revised ERD replaces the Boarding House entity with a more general Property entity to accommodate boarding houses, apartments, and rooms, as clarified. The type attribute distinguishes between property types. Primary keys (PK) and foreign keys (FK) ensure referential integrity. MySQL will be used with indexes on frequently searched fields (e.g., availability, price, type) to optimize performance.

3.6 Implementation

TLIMS will be developed using a full-stack web development approach. The front-end will use HTML, CSS, and JavaScript for responsive interfaces. The back-end will use PHP for server-side logic, handling requests, authentication, and API integrations. MySQL will manage the database, with tables corresponding to the ERD entities.

Key Implementation Steps:

- Set up the development environment using XAMPP (local testing) or a cloud server.
- Implement user authentication with PHP sessions and hashed passwords (e.g., bcrypt).
- Integrate Google Maps API using JavaScript SDK for property location display.
- Develop CRUD operations for property postings and listings using PHP and AJAX for seamless updates.
- Implement a notification system using WebSockets or polling for real-time alerts.
- Ensure security through HTTPS, input validation, and SQL injection prevention.
- Follow agile methodology with iterative sprints, incorporating feedback from JHCSC-Pagadian students and landlords.

