**FINAL PROJECT**

**Project Kickoff & Requirements Gathering**

**Objective**

Students will identify the system objectives, scope, and stakeholders of their chosen project. They will practice gathering requirements through stakeholder interaction and document both functional and non-functional requirements. Students will also create a context diagram and high-level system overview.

**Instructions for Students**

**1. Team Formation**

* + **Project Lead** – Jimmilyn H. Pascua
  + **Documenter** – Ruby Ann V. Maramot
  + **Diagrammer** – Wildelene R. Consolacion
  + **Presenter** – Mary Rose U. Pardilla

**2. Stakeholder Interview**

Interview questions:

* How do you currently manage tenant information, room assignments, and payment records?
* What common problems do you face in managing tenant records and tracking payments manually?
* Would automated email notifications for due or overdue balances be helpful to you and your tenants?
* How do you currently generate financial reports, and what difficulties do you encounter?
* What features would make a digital management system easy and convenient for you (charts, summaries, secure login, etc.)?
* How important is it to have secure access and data backup for tenant and financial records?
* What concerns do you have about transitioning from a manual system to a digital solution?

**3. Requirements Documentation**

**Functional Requirements (FRs):**

1. The system shall allow the landlord to securely log in using Laravel Breeze authentication.
2. The system shall allow the landlord to add, edit, delete, and view details of houses, rooms, tenants, and payments.
3. The system shall send automated email notifications to tenants regarding due or overdue payments.
4. The system shall allow filtering of payment records by month for easier tracking.
5. The system shall generate reports summarizing tenant payments for monthly, quarterly, and annual periods.
6. The system shall display visual charts showing tenant distribution and monthly payment summaries.
7. The system shall allow the landlord to print total payment reports.
8. The system shall track real-time payment statuses and outstanding balances.

**Non-Functional Requirements (NFRs):**

1. The system shall provide a user-friendly interface that requires minimal training for the landlord.
2. The system shall ensure secure handling of tenant and financial data using encryption and authentication protocols.
3. The system shall be available 99% of the time to ensure accessibility for financial tracking.
4. The system shall be able to support growing numbers of tenants and records without performance issues.
5. The system shall be modular, allowing updates and bug fixes with minimal downtime.
6. The system shall integrate with Google Email API for sending notifications

**4. Create a Context Diagram**

A screenshot of a computer

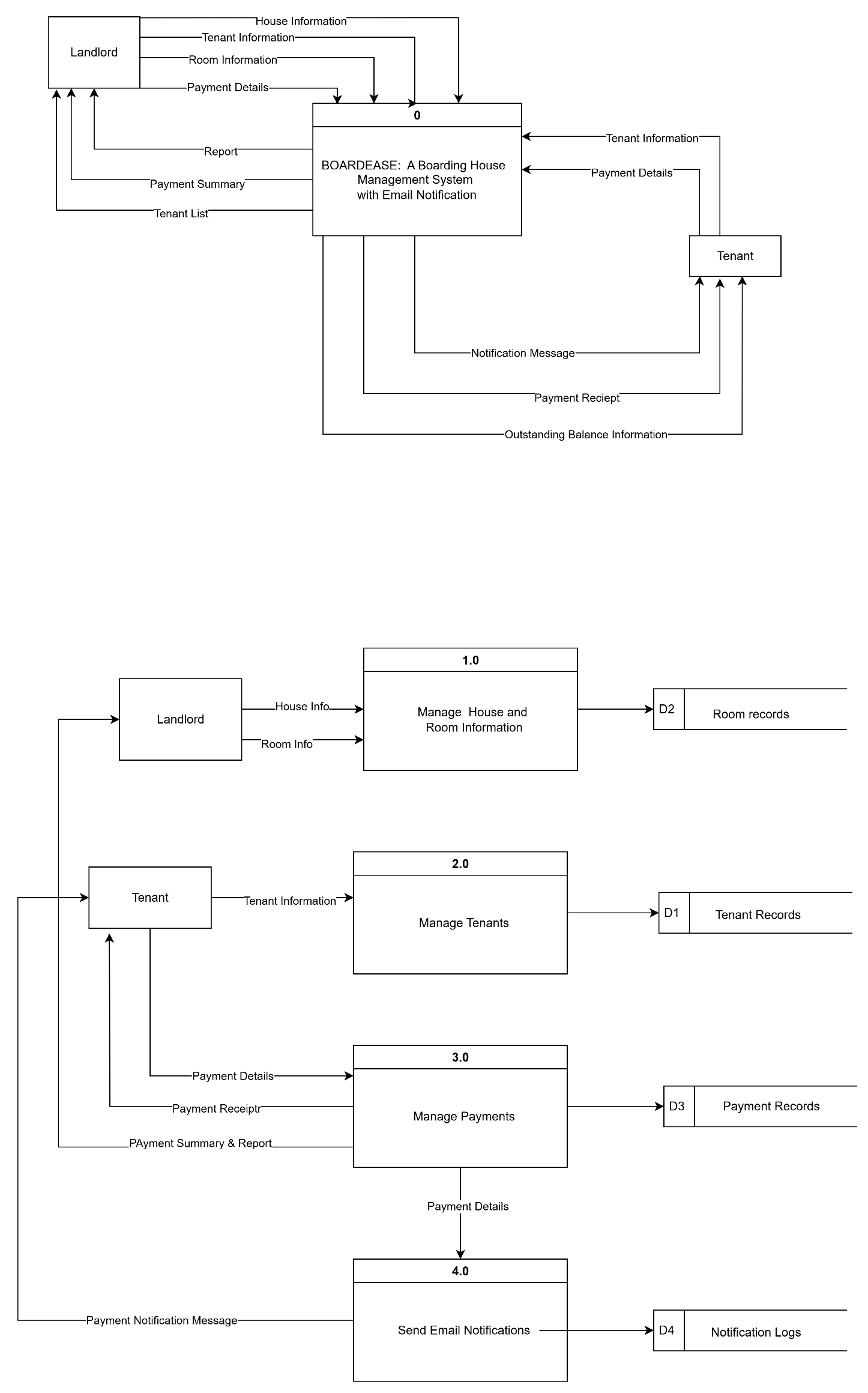
AI-generated content may be incorrect.

**5. High-Level System Overview**

BoardEase digitizes operations by replacing the existing paper-based process with a secure, user-friendly solution.

The system allows the landlord to manage tenants, rooms, and rental payments through a centralized dashboard. It features automated email notifications for tenants with due or overdue payments, real-time payment tracking, and filtering options for monthly financial analysis. Landlords can also generate detailed payment reports for monthly, quarterly, and annual use.

The platform provides visual charts for easy reference on tenant distribution and finances, ensuring data is accessible and actionable. Security is prioritized through Laravel Breeze authentication and encryption of tenant and financial data. By automating manual tasks and providing detailed insights, the system enhances efficiency, reduces errors, and supports long-term planning for boarding house operations.

1. **Data Flow Diagram**

**Modules:**

Authentication and Security

House and Room Management

Tenant Management

Payment Management

Email Notifications

Notification Module

Reports and Analytics

**API Modules:**

Google email API

Charts and Visualization API (chart.js)

Laravel Breeze