**Research Lab Equipment Booking System**

*Partial Report*

Christian Manangan 11755967, Kyle Hawkins 11759096, Ethan Frazier 11792962

Dr. Kumar

02/23/2025

CptS 451: Introduction to Database Systems

**Abstract**: The Research Lab Equipment Booking System aims to make it easier for students and researchers to reserve lab equipment. Users can register and login, view equipment availability, book equipment, and track their past reservations. Administrators oversee bookings, enforce rules, and generate reports to optimize resource use. The system includes user authentication, equipment tracking, and real time updates to prevent scheduling conflicts and ensure fair access. This project focuses on developing a database-driven solution with a three-tier client server architecture, ensuring scalability, security, and maintainability. The system will be created using Render as the cloud platform, and PostgreSQL as the relational database. These two technologies were chosen because Render integrates seamlessly with PostgreSQL which is very convenient for database creation, smooth workflow, and ease of access of information. Key features include role-based access control, an approval process for reserving equipment, notifications, and detailed usage logs. By creating a system in this manner, it reduces inefficiencies, helps prevent scheduling conflicts, and overall improves the accessibility and management of lab resources.

**TABLE OF CONTENTS**

[I. Introduction 4](#_Toc191226637)

[I.1. Overview of the Project 4](#_Toc191226638)

[I.2. Problem Statement 4](#_Toc191226639)

[I.3. Objectives and Scope of the Project 4](#_Toc191226640)

[I.4. Importance and Potential Impact 4](#_Toc191226641)

[II. Function and Non-Functional Requirements 4](#_Toc191226642)

[II.1. Functional Requirements 4](#_Toc191226643)

[II.2. Writing User Stories 5](#_Toc191226644)

[II.3. Non-Functional Requirements 6](#_Toc191226645)

[III. Database Design: ER Diagram & Tables 8](#_Toc191226646)

[III.1. Entity-Relationship (ER) Diagram 8](#_Toc191226647)

[III.2. Conversion of ER Diagram to Relational Tables 8](#_Toc191226648)

# Introduction

# Overview of the Project

The project is a lab equipment rental system that allows users to log on to an online website and create reservations for the equipment they desire. The reservations and usage logs of those who used the equipment will all be handled internally by the system and can be manually edited by a system admin with high tier credentials. Students and Researchers have the same level of clearance in the database and can rent any equipment that is available in the system.

# Problem Statement

Developing a secure and efficient way for students and researchers to reserve equipment with ease and no conflict. If equipment is broken and should not be used by another lab member, the online system for rentals and tracking equipment will display whether an item is broken or not and can then be properly removed and disposed of instead of someone not knowing the equipment is broken- and mistakenly reserving it for a potentially risky experiment.

# Objectives and Scope of the Project

The most important goal of this system is to effectively rent equipment and gear to any person who has logged onto the site and requested equipment that is available and not rented by someone else. The scope of this project is somewhat limited in a grand scheme of who can access it, and the global usage of this system as it is only for a laboratory, designed to help with equipment rental, and tracking. Many people may be a part of the lab, and the database can hold a lot of records, but users that access this system will all be a part of the same category and in the same location at one time.

# Importance and Potential Impact

Science lab equipment can be very limited, or hard to track when the facility does not use an efficient database to store records of the equipment usage and last renter. Without a system to track this information, trying to keep watch of the equipment with physical notes and people’s word of mouth on where something has gone is inefficient. Using a database is much faster, and secure for renting equipment to users who have credentials they can verify through the system and log their equipment condition and time used so that all other users may view that information and avoid confusion about availability.

# Function and Non-Functional Requirements

# Functional Requirements

Functional Requirements are key features, actions, and workflows of what the system must do. Below is a detailed and exhaustive list of functional requirements we defined.

1. **User Registration & Authentication**: Users such as students, researchers, and administrators must be able to register and authenticate using a secure login system.
2. **Equipment Listing & Details**: The system must be able to store a list of lab equipment along with its details like specifications and availability status, in a retrievable and modifiable way like a database.
3. **Equipment Booking System**: Users should be able to reserve equipment for a specific time slot based on availability status
4. **Usage Logging System**: The system must maintain logs of equipment usage. This may include user details and timestamps
5. **Booking Approval**: The system must allow administrators to review and to make a decision for booking requests
6. **Usage Reports & Analytics**: Administrators should be able to generate a report on user activity and equipment usage, and booking patterns
7. **User Can View Reservation Statuses**: Users should be able to see a page that shows a list with the status of their booking reservation requests and see if it's pending, denied, or approved.
8. **Admin Approve/Deny Reservation Requests**: An admin should be able to make decisions on pending reservation requests to either approve or deny requests.
9. **Admin View Reservation Requests**: An admin should be able to see a page of reservation requests that are pending approval.
10. **Lab Equipment Status Is Shared and Accurate**: The availability status of equipment should be accurately reflected/shared and seen by all users.

# Writing User Stories

User stories describe different interactions of users with the system. Below are all the user stories the team has come up with based on the functional requirements.

1. As a student, I want to register as a student through the system so that I can verify my role as a student and reserve equipment.
2. As a researcher, I want to register as a researcher through the system so that I can verify my role as a researcher and have priority reserving equipment.
3. As an administrator, I want to register as an admin through the system so that I can handle administrative tasks like generating usage reports.
4. As a student, researcher, or administrator, I want to log into the system securely so that I can access my account and perform role-specific actions.
5. As a student or researcher, I want to be able to view equipment along with its details so that I can rent and achieve my academic goals.
6. As a student or researcher, I want to be able to see if equipment is available or not, so that I can better plan for what I can do.
7. As a student or researcher, I want to be able to reserve equipment for a specific time slot based on availability status so that I can reserve equipment in an orderly and structured manner.
8. As an student or researcher or administrator, I want to be able to view system logs of equipment usage that includes user details and timestamps so that I can use that information to improve scheduling and identify misuse
9. As an administrator, I want to view a list of current booking requests, so that I can handle and see equipment demands.
10. As an administrator, I want to review and make decisions for booking requests so that I can prevent misuse and maintain resource availability
11. As an administrator, I want the system to store and to be able to see usage analytics, so that I can better perform administrative duties and understand equipment demands.
12. As an administrator, I want to be able to see a user's equipment rental history, so that I can understand and track use.
13. As a student or researcher, I want to see a page that shows a list with the status of their booking reservation so that I can see if it’s pending, denied, or approved
14. As a student or researcher, I want to be notified of whether my booking request was approved or denied so that I can plan accordingly.
15. As a student or researcher, I want to modify or cancel my booking request so that others can reserve the equipment
16. As an administrator, I want to see a page of reservation requests that are pending approval so that I can be informed about the amount of booking requests and make decisions.
17. As a student or researcher, I want to see the status of equipment so that I can see if equipment is available for a given day or has already been booked by another user for that date.

# Non-Functional Requirements

Non-functional requirements outline the operational qualities of the system, such as performance, system availability, maintainability, and security. The system needs to ensure user experience to ensure that it meets quality standards beyond core functionality. Details of non-functional requirements are as follows.

|  |  |  |
| --- | --- | --- |
| **Non-Functional Requirements** |  | **Description** |
| **Security** |  | **User data such as passwords should be encrypted and stored** |
| **Usability** |  | **Users should be able to use the system intuitively and with minimal** |
| **Scalability** |  | **System must be able to accommodate various expansions** |
| **Availability** |  | **System must be up and running for the majority of time** |
| **Maintainability** |  | **Bug fixes and updates should be simple and easy for the system** |
| **Compliance** |  | **Meets legal and regulatory requirements** |
| **Loose Coupling of UI and Database** |  | **The systems UI should be loosely coupled to the database meaning that if the underlying way the data is stored in the database changes, it won’t also require remaking the UI.** |
| **Role Based Access** |  | **The system should ensure role-based access and views so that unauthorized users can't see or perform administrative functions.** |
| **Performance** |  | **System must process booking requests in a quick/timely manner, no longer than ½ a minute for a user's booking request to be propagated and viewable by the admin page.** |
| **Accuracy** |  | **The accuracy of the status of data should always be above 99%, across all user devices. Meaning that all users should be able to see the same, accurate, data and details for the lab equipment.** |

# Database Design: ER Diagram & Tables

# Entity-Relationship (ER) Diagram

An ER diagram represents the database schema with entities, attributes, and relationships. Primary keys are attributes that are underlined. The cardinality (1:1, 1:M, M:1, and M:M) for each entity relationship set are labeled on the diagram. The ER diagram below was created using a UML tool Draw.io.

A diagram of a company

AI-generated content may be incorrect.

# Conversion of ER Diagram to Relational Tables

To turn the ER diagram into tables we make the squares into tables, and relationships that are many to many tables too. Then we make the attributes into columns. The primary keys are the underlined or id fields. Foreign keys are the relationships between entities. Then we normalized the tables and made multi-value attributes into their own tables. Please see on the next page our relational tables.

A diagram of a computer

AI-generated content may be incorrect.