Research Lab Equipment Booking System

Team Members:

Johnjacobjames Erdman (11771150)

Cole Logan (011813565)

Shaylin Smith (11755543)

Asa Fischer (011850410)

Instructor:

Parteek Kumar

2/23/25

CptS 451 – Intro to Database Systems

Abstract

The Research Lab Equipment Booking System Project's main objective is to create a database system that allows actions by different types of users. Other objectives include creating a frontend, backend, and hosting our entire system on a cloud service.

To briefly go over an overview of the database system. It stores equipment details that individual researchers can make reservations for. Admins approve these reservations and suppliers supply new equipment that is added to the database system. All types of users are able to view certain details about the database depending on their access level. For example, admins can view usage analytics for equipment, but all types of users can view equipment details.

The expected outcomes of this project include gaining experience in creating applied databases and applying databases to a full-stack project. In terms of gaining experience, we expect to go through the design process of database modeling and database design. We will create anything that is expected in the design process including entity-relationship diagrams, relational tables, etc. Finally, the project will produce a finished and interactable system with a fully usable front end to interact with the database.

Table of Contents

I. Introduction 4
I.I. Overview 4
I.II. Problem Statement 4
I.III. Objectives and Scope 4
I.IV. Importance and Impact 5
II. Requirements 5
II.I. Functional Requirements 5
II.II. User Stories 7
II.III. Nonfunctional Requirements 9
III. Database Design 10
III.I. ER Diagram 10
III.II. Conversion of ER Diagram to Relational Tables 11

I. Introduction

I.I. Overview

The project is a booking system for research lab equipment. Researchers will be able to view available equipment and reserve equipment that they want to use for a certain amount of time. A database will store the details of the equipment including specifications and availability. Reservations made by a researcher must be approved by an admin. Admins approve reservations and can view usage analytics to make sure the equipment is maintained well by researchers. All equipment is supplied by some supplier, and suppliers are the only ones that can create new equipment that will be approved by an admin. Any new type of user will have to register an account that will store their account type in the database (researcher, admin, supplier).

The purpose of this project is to apply a database system to a full stack application that we can deploy on a cloud service. The database system will have an interactive component with an implemented frontend that users will use to submit "actions" to the database. The context of the research lab equipment booking system allows us to gain experience with creating a real-world-like database system.

I.II. Problem Statement

Our project addresses a real-world-like database system that could be applied. Research laboratories often buy a lot of equipment from suppliers and labs tend to have many researchers working at them. Since researchers are often working on different tasks, equipment deteriorates over time. A database system can greatly benefit a research lab to allow researchers exclusive access to certain equipment over a finite period. It also allows accountability among researchers and prevents excessive damage to the equipment.

I.III. Objectives and Scope

Our project consists of four main objectives. Our first and most important objective is to apply a database system to our project. This includes creating several tables to store data about equipment, reservations, and different types of users. Our second objective is to implement actions that a certain type of user can enact on the database system. This will require some backend Python code and queries by the database to do. Our third and fourth objectives are a bit

more minor, but we will need to implement a front end and deploy the entire system on a cloud service respectively. These last two objectives aren't as important as the first two, but they will allow for a nicer user experience and allow others to test use of the system.

The scope of the project is to create a full stack application that is deployable on a cloud service. As stated before, a frontend, backend, and database system will be required for this project to have a completed product that is useable by others.

I.IV. Importance and Impact

The significance of our project is that it is solving a real-world-like problem that something like a research laboratory might want to implement in their lab. It's also significant to us because it allows us to gain experience with implementing a database system within a full stack project. As well as giving each of us individually more experience with team projects. The potential impact of this project is that it provides a potentially new solution for a booking system for research labs specifically.

II. Requirements

II.I. Functional Requirements

User Registration –

Our system must allow users to register and create accounts. This allows new accounts to be created and the enforcement of role-based access control.

User Verification –

Our system must allow users to be vetted and accepted/denied based on credentials. This makes it possible for users to gain access to their accounts and our site.

Login and Logout –

Our system must allow users to log in and log out with their credentials. This creates basic session security for user accounts.

View Equipment Details –

Our system must allow users to view all the details of any specific piece of equipment. For every piece of equipment, a user can be presented with a view containing relevant information.

Search Equipment –

Our system must allow users to query the list of equipment to find certain equipment. This creates a search system for the database, making usage simpler for users.

Reserve Equipment –

Our system must allow users to reserve a specific piece of equipment for a certain period. This is the primary functionality for our site so we must make the reservation process easy and intuitive.

Cancel Reservations –

Our system must allow users to cancel reservations they have made if they change their mind. This adds flexibility and ensures equipment being unused is freed up for others.

Usage Logs –

Our system must allow users to see the availability of equipment and who is using it. This gives users the opportunity to track equipment usage.

Admin Approval –

Our system must allow users to ask for admin approval for the use of equipment. This ensures equipment use is controlled and authorized by an admin user.

View Usage Analytics –

Our system must allow users (admins) to view patterns and trends based on analytics. This provides admins with the ability to see usage patterns and improve resource management.

Equipment Maintenance Schedule –

Our system must allow users to view the status of equipment to verify it works. This ensures that equipment is only available for reservation when it is in working condition.

User Profile Management –

Our system must allow users to update their profile to keep their information up to date. This ensures accurate user information.

Notification System –

Our system must allow users to subscribe to notifications to stay up to date on equipment. This provides the users with the ability to stay informed on their reservations and equipment details.

Booking Confirmation –

Our system must allow users to give their phone number/email for confirmation notifications. This gives the users added assurance that their reservation has been processed and stored.

Supplying –

Our system must allow suppliers to supply equipment to a research lab. This gives suppliers a way to add new equipment and allows users to have access to new equipment.

II.II. User Stories

User Role –

As a user, I want to register for an account so I can use the equipment reservation site.

As a user, I want my credentials checked at login so that unauthorized access isn't allowed.

As a user, I want to log in so that my information is saved and accessible.

As a user, I want to log out so that my account isn't used when I am not using it.

As a user, I want to be able to view available equipment so that I can choose which one I want.

As a user, I want to be able to search for equipment so that I can find the equipment I need.

As a user, I want to reserve equipment so that I can have access to it when I need it.

As a user, I want to cancel my reservation so that I do not keep the equipment reserved when I am not using it.

As a user, I want equipment to only be available when it is in working condition so that I don’t accidentally reserve a broken piece of equipment.

As a user, I want to be able to manage my profile so that it shows accurate information.

As a user, I want notifications so that I can be informed and up to date.

As a user, I want confirmation notifications so that I know that my equipment has been booked.

Admin Role –

As an admin, I want to be able to make my equipment available for viewing so that users can reserve them.

As an admin, I want my equipment to be up for reservation so that users can borrow it and use it as needed.

As an admin I want to track the equipment usage so I can maintain logs for accountability.

As an admin, I want to approve equipment use so that trusted users can use the equipment.

As an admin, I want to deny equipment use so that untrusted users cannot use equipment.

As an admin, I want to view usage analytics so that I can recognize patterns and improve efficiency.

As an admin, I want to ensure equipment is only up for reservation when in working condition so that users don’t accidentally reserve a broken piece of equipment.

Supplier Role –

As a supplier, I want to supply equipment so that users can reserve and use new equipment.

As a supplier, I want to view the equipment database, so that I can recognize if I need to supply certain equipment.

II.III. Nonfunctional Requirements

Scalability –

Our system must be able to handle increasing usage without issue. This encourages upholding great performance with higher demand.

Performance –

Our system must run smoothly and efficiently on all fronts. This ensures a smooth and efficient user experience.

Security –

Our system must be secure and not allow user data to be tampered with or seen. This allows us to protect user data and overall system integrity.

Reliability –

Our system must be available whenever a user may need it. This ensures minimal downtime and consistent availability of our site.

Maintainability –

Our system must be easily maintained, needing little supervision. This reduces maintenance costs, and the effort needed to keep it up and running.

Usability –

Our system must be easy to use and simple to learn. This enhances user satisfaction and reduces the learning curve for new users.

Compatibility –

Our system must be uniform across different devices. This allows us to provide consistent user experience across different screen sizes and devices.

Documentation –

Our system must be thoroughly documented for future reference. This facilitates a simple path for the layman to understand our system.

Modularity –

Our system must be developed in a modular manner for easier development, testing, and maintenance. This will simplify the development process and enhance overall flexibility.

Auditability –

Our system must keep logs to support audits. This provides transparency and accountability.

III. Database Design

III.I. ER Diagram

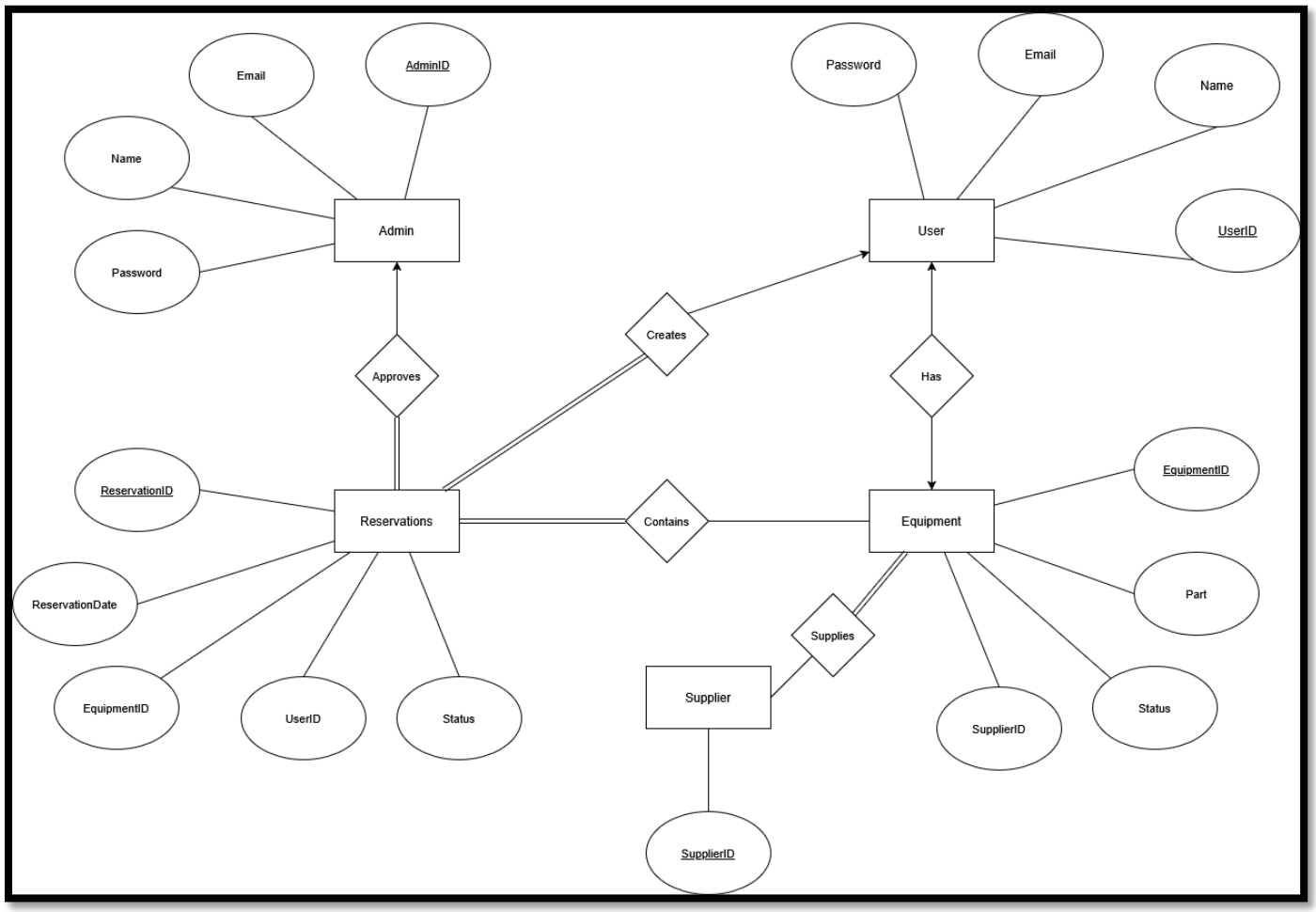

Figure 1: Entity-Relationship Diagram of Research Lab Equipment Booking System

III.II. Conversion of ER Diagram to Relational Tables

When we convert our ER diagram to relational tables, we will be creating five tables for entities. The entity tables will be User, Admin, Supplier, Reservations, and Equipment. Each of the entity tables is a direct translation from the entity's attributes into table format. A few of the tables will have additional columns from their attributes due to having one-to-many or one-to-one relationships with a different entity. Below will be the table structure for each of these tables including their column names, variable type, and constraints.

User		
UserID	Int	Primary Key
Name	Varchar	Not Null
Email	Varchar	Not Null, Unique
Password	Varchar	Not Null

Admin		
AdminID	Int	Primary Key
Name	Varchar	Not Null
Email	Varchar	Not Null, Unique
Password	Varchar	Not Null

Reservation		
ReservationID	Int	Primary Key
ReservationDate	Date	Not Null
EquipmentID	Int	Foreign Key (references Equipment(EquipmentID))
UserID	Int	Foreign Key (references User(UserID))
Status	Varchar	Not Null
AdminID	Int	Foreign Key (references Admin(AdminID))

Equipment		
EquipmentID	Int	Primary Key
Part	Varchar	Not Null
Status	Bool	Not Null
SupplierID	Int	Foreign Key (references Supplier(SupplierID))

UserID	Int	Foreign Key (references User(UserID))

Supplier		
SupplierID	int	Primary Key

For each of the five relationships in our ER diagram, there will be an additional two tables, since there's only two many-to-many relationships. The other three relationships do not need additional tables because the many-to-one relationships inherit a primary key from the "one" table to the "many" tables. Similarly with the one-to-one relationship, one of the tables will inherit the primary key from the other table. These additional tables will be junction tables for equipment to suppliers and for reservations to equipment as well. These additions will allow us to bridge the information between the two tables and uniquely identify each pair for the many-to-many relationship. Below are the additional tables.

Equipment-Supplier		
(EquipmentID, SupplierID)	n/a	Primary Key
EquipmentID	int	Foreign Key (references Equipment(EquipmentID))
SupplierID	int	Foreign Key (references Supplier(SupplierID))

Equipment-Reservation		
(EquipmentID, ReservationID)	n/a	Primary Key
EquipmentID	int	Foreign Key (references Equipment(EquipmentID))
ReservationID	int	Foreign Key (references Reservation(ReservationID))