# Milestone 2

## 1. Functional Requirements Using User Stories

### User Registration and Verification

**As a student**, I want to be able to register as a student so that I can access features for lab equipment booking.

**As a teacher**, I want to be able to register as a teacher so that I can manage or book equipment for classes.

**As a lab manager**, I want to be able to register as a lab manager so that I can define what equipment is available.

### User Authentication

**As a student**, I want to log in with my information so that I can access equipment booking for my assigned lab section.

**As a teacher**, I want to log in with my information so that I can access the equipment booking for my assigned lab section.

**As a lab manager**, I want to log in with my information so that I can access equpment inventory for my lab section(s).

**As a system administrator**, I want to enforce role-based access control so that only verified users can book equipment, create equipment, etc.

### View Equipment and Availability

**As a student, teacher, or lab manager**, I want to be able to see a real-time view of of available equipment and their status for my lab section(s) so that I can optomize my bookings

**As a system administrator,** I want to be able view the available equipment for all lab sections so that I can oversee all bookings

### Equipment Inventory Management

**As a lab manager**, I want to maintain an updated inventory of what equipment is avaliable for my lab and add/remove equpment as nessisary so that the inventory is kept up-to-date

## Search and Filter Equipment

**As a registered user,** I want to be able to search and filter equipment so that I can find equipment I want to book

## Booking System

**As a student or teacher,** I want to be able to reserve equipment for my lab section(s) so that I can use that equipment

**As a student or teacher,** I want to be able to modify or cancel bookings so that I can have flexibility if schedules change or other events occur

**As a lab manager,** I want to be able to cancel user's bookings/reservations for my lab section(s) so that I can prevent booking of equipment for various reasons (eg equipment out of date, equipment broke between reservation time and the booking time etc)

**As a system administrator**, I want to be able to cancel user's bookings/reservations for any lab section so that I can solve any system-wide booking issues that may occur.

**As a system administrator**, I want to ensure that bookings are linked to users and equipment using foreign keys so that transactional integrety can be maintained

## Usage Analytics

**As a lab manager,** I want to be able to view usage analytics so that I can optomize inventory for people's needs

## Export Booking Data

**As a system administrator,** I want to be able to export bookings logs into CSV/JSON formats so that external data proccesing can be conducted

## Time-Slot Validation

**As a registered user,** I want the system to enforce time-slot validation so that overlapping bookings are auto-rejected.

## 2. Identify Key Entity Sets and Relationships

### User (Generalization)

Stores user details.
**Attributes:** `user_id`, `name`, `email`

### Student (Specialization of User)

Describes a student

**Attributes:** `student_id` (fk to user), `major`, `enrollment year`

### Teacher (Specialization of User)

Describes a teacher

**Attributes:** `teacher_id` (fk to user), `department`

### Lab Manager (Specialization of User)

Describes a lab manager

**Attributes:** `lab_manager_id` (fk to user), `department`

### System Admin (Specialization of User)

Describes a system admin

**Attributes:** `system_admin_id` (fk to user), `access_level`

### Equipment

Describes lab equipment.
**Attributes:** `equipment_id`, `equipment_name`, `equipment_type`, `status`, `lab_section(fk)`

### Booking

Manages reservations.
**Attributes:** `booking_id`, `user_id (fk)`, `equipment_id (fk)`, `start_time`, `end_time`, `status`.

### Lab Section

Describes a lab section.
**Attributes:** `lab_section_id`, `teacher (fk)`, `lab_manager (fk)`

## Student Section

Represents the realationship between students and lab sections

**Attributes:** `student_id` (fk to student), `section_id` (fk to lab section)

## Analytics Report

Stores aggregated usage data

**Attributes:** `report_id, report_type,date,data`

| Relationship | Connectivity | Participation Constraints |
|---|---|---|
| User / Student | 1:1 | User: partial<br>Student: total |
| User / Teacher | 1:1 | User: partial<br>Teacher: total |
| User / Lab Manager | 1:1 | User: partial<br>Lab Manager: total |
| User / System Admin | 1:1 | User: partial<br>System Admin: total |
| User / Booking | 1:M | User: partial<br>Booking: total |
| Equipment / Lab Section | 1:1 | Equipment: total<br>Lab Section: partial |
| Equipment / Booking | 1:M | Equipment: partial<br>Booking: total |
| Lab Section / Teacher | M:1 | Lab Section: total<br>Teacher: partial |
| Lab Section / Lab Manager | M:1 | Lab Section: total<br>Teacher: partial |
| Student Section / Student | M:1 | Student Section: total<br>Student: partial |
| Student Section / Lab Section | M:1 | Student Section: total<br>Lab Section: partial |
| Booking / Analytics Report | M:N | Booking: partial<br>Analytics Report: partial |

# 3. ER Diagram
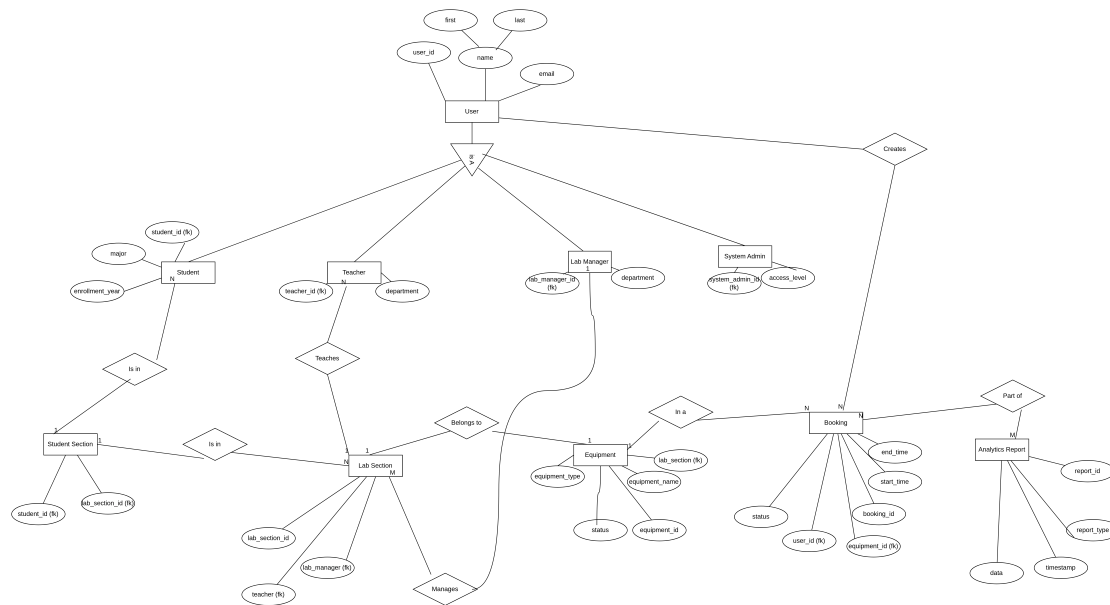


# 4. Justification

The **User** entity is generalized, with specialized entities (**Student, Teacher, Lab Manager, System Admin**) to enforce role-specific attributes and permissions. This choice was made due to each of these rolls interacting differently with bookings and equipment. The **Equipment** entity stores lab-specific items, linked to **Lab Sections** to ensure proper allocation. The **Booking** entity is crucial for managing reservations, tracking which users have reserved which equipment and for what duration.

## Connectivity and Justification

- **1:1 Relationships:**

    o A **User** can only belong to one specialized role, ensuring clear role-based access control.

    o **Equipment** belongs to a single **Lab Section**, preventing duplication across multiple sections.

- **1:M Relationships:**

    o A **User** can make multiple **Bookings**, but each **Booking** must belong to a single **User** to maintain accountability.

o A **Teacher** or **Lab Manager** can be responsible for multiple **Lab Sections**, but a **Lab Section** is assigned a single **Teacher** and a single **Lab Manager.**

- **M:N Relationships:**

  o **Students** are enrolled in multiple **Lab Sections**, and each **Lab Section** can have multiple **Students**. The **Student Section** table models this relationship.

  o **Analytics Reports** aggregate data from multiple **Bookings**, allowing system-wide and lab-specific usage tracking.

## Alignment with Functional Requirements

The schema enforces **role-based access control**, ensuring that only authorized users can book, manage, or oversee equipment. **Time-slot validation** is supported through booking constraints, preventing overlapping reservations. **Search and filtering** are streamlined by associating equipment with specific labs. **Usage analytics** and **exportable reports** provide lab managers and system administrators with data-driven insights for optimizing inventory management.

## Assumptions

- Each user belongs to exactly one role.

- Equipment cannot be shared between lab sections.