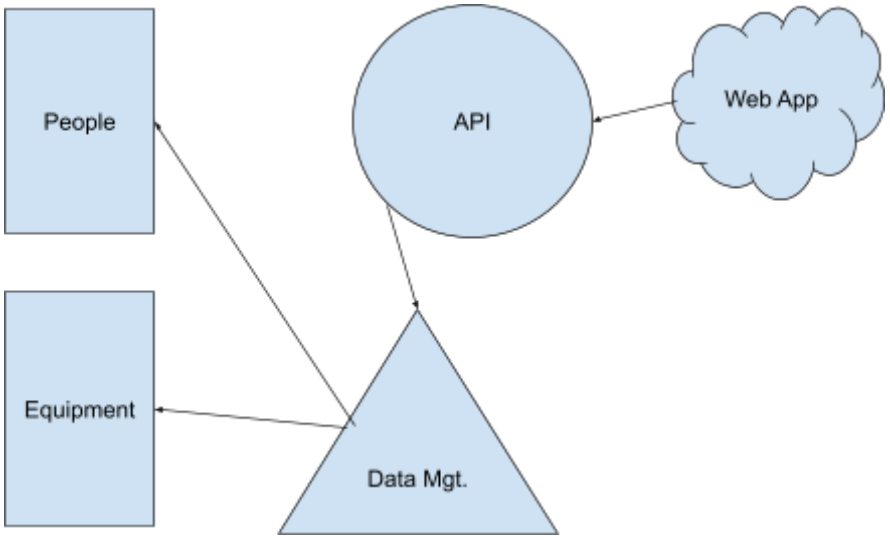
1. System Description
    a. The system is designed to streamline the process of lending and managing departmental equipment. It ensures that users can easily borrow available equipment while maintaining accurate records of lending activity. Admins oversee inventory, manage user roles, and enforce borrowing policies. In short, it coordinates equipment lending and borrowers for multiple instances of similar devices.
    b. Overview of the functionality:
        i. Admin users ("admins") can add, remove, and manage equipment in a catalog
        ii. Regular users can browse the catalog and request to borrow available equipment
        iii. The system tracks borrowed items, borrowing limits, and updates item availability
        iv. Notifications keep users informed about due dates and overdue items

2. Architectural Overview
    a. The system has a central API connecting database and the web application. The database has two main parts: Users (storing user info and roles) and Equipment (tracking items and availability). The API facilitates communication and ensures data consistency between users, equipment, and the web application.
    b. Major Components:
        i. API
        ii. Database
        iii. WebApp (FrontEnd)
        iv. Data Management Protocol
        v. Notification system(?)
    c. 

3. Functional requirements

EPIC 1 - Borrow
As a regular user
I want to borrow and return equipment through an intuitive interface
So that I can easily access departmental resources and track my borrowing history

EPIC 2 - Login Protocol
As a user
I want to securely log into the system and have my role (regular user or admin) recognized
So that I can access the features relevant to my permissions

EPIC 3 - Elevate (admin actions)
As an admin
I want to manage users, equipment, and generate reports
So that I can oversee and optimize the department's resource usage

EPIC 4 - Notification
As a system
I want to send automated reminders and notifications for equipment status updates
So that users and admins stay informed about due dates, check-ins, and overdue items

4. Non-Functional requirements

a. ***Security***
   The system must support secure user authentication, using encrypted protocols (e.g., HTTPS) to protect login credentials and user data.
b. ***Data Integrity***
   All changes to equipment or user data must be logged and verified to prevent inconsistencies in the database.
c. ***Notification Delivery***
   Automated reminders and notifications must be sent to users and admins within a couple minutes of triggering events (due dates, check-ins, or check-outs).
d. ***Unique identifiers***
   Each piece of equipment must be assigned a unique identifier, ensuring no duplication or conflicts in the tracking process.
e. ***Role-Based Permissions***
   Admins must be able to perform all actions available to regular users, as well as admin-specific tasks like managing equipment, users, and generating reports.

5. Technologies and Frameworks
    a. Python w/ Django (DRF)
        i. develop RESTful API to communicate with Web app
    b. SQL (PostgreSQL? SQLite?)
        i. interactive, real-time updatable user and equipment tables
    c. email client for sending notifications and reminders
    d. Docker? Containerization

6. Minimum Viable Product
    a. A service that can look into a database and check in/out equipment. Must have login to differentiate users from one another (to assign equipment) and from admin users. Must be accessible through an API and interact with web app (commands in common, server/client interaction)
    a. Users must be able to see the list of available equipment to check out and their current due dates for the equipment they have already borrowed.
    b. Admin users must be able to add equipment to a catalog which lists that equipment as available to admin and non-admin users for borrowing. Users must also be able to reserve equipment from the catalog for borrowing and the system must initiate pickup logistics. Users must be able to see which items are currently checked out to them.

7. Project Road Map

Sprint 1 - EPIC 2: Login
Focus on implementing the authentication system, allowing users to create new accounts and securely log in.

Sprint 2 - EPIC 1: Borrow
Develop the equipment borrowing system, enabling users to browse available equipment, request items, and update inventory status upon borrowing.

Sprint 3 - Integrate Login & Borrow
Ensure integration between authentication and borrowing functionalities. Users should only be able to request equipment once logged in, and role-based access should be enforced.

Sprint 4 - EPIC 3: Elevate
Implement admin features, allowing admins to manage equipment, assign user roles, generate reports, and verify returned equipment.

Sprint 5 - EPIC 4: Notification
Develop the notification system for automated reminders, overdue alerts, and real-time equipment status updates via email or other communication methods.

Sprint 6 - TBD (Final Adjustments & Additional Features)
Depending on progress, this sprint will focus on completing unfinished work from previous sprints. If all epics are completed, additional features may be explored.