System description:

CheckPoint is a software that will be used to manage the sharing of the computer science department resources. This will provide a way for students to check out equipment for use of technology. Through the software a user will be able to maintain their personal usage history and future equipment use. This software will be a way the computer science department can allow equipment to be used by ambitious students while holding them accountable for the equipment they borrow.

There will be two types of users: regular users and admins. Regular users can check out and check in equipment from a list of available technologies. This list will be monitored by admin users who will have access to add and remove equipment from this list. Regular users will also see their personal borrow history and current borrowed equipment. The Admin users will have full control over the availability of the equipment, management of other users, and the ability to send notifications to users borrowing equipment currently. Admin users can also generate reports that show current and past equipment usage, and reserve equipment for future classroom use. The Admin users will have these controls in addition to the regular user access.

Architecture Overview:

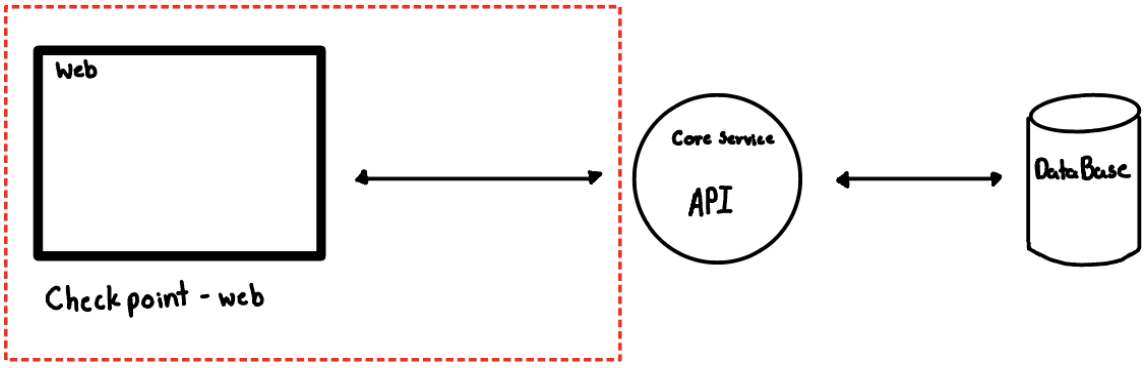
Description of design/architecture of the system

Checkpoint has 3-tier architecture consisting of a frontend (client-side UI), a backend (server-side logic), and a database (information storage). The frontend provides a UI for check-ins/check-outs, displays lists of equipment and their statuses and availability, lets users see their borrowed equipment history, lets admins manage equipment and user permissions, and handles logins/logouts. The backend processes requests from the frontend and interacts with the database, Provides RESTful APIs for frontend interaction, Implements business logic for check-in/check-out, reminders, and reports. The database is used to store user and equipment information.

List of all major system components

Checkpoint web will cover all system components until API at which case it will be handled by the checkpoint core service and API team.

System diagram - (make a new diagram)



Functional Requirements:

Epic 1: logging in and seeing regular user functionality

As a regular user,
I want to be able to log in and see what functionalities are available,
so that I can see what equipment I can borrow.

Epic 2: accessing history logs

As a regular user,
I want to have access to my own history log,
so that I can track my borrowing activities.

Epic 3: reserving equipment (checking out and checking in equipment)

As a regular user,
I want to be able to reserve an equipment,
so that I can set a period of time to reserve equipment.

Epic 4: admin control over all users and equipment

As an admin,
I want to have access to all user activities and equipment status,
so that I can accurately manage the process of borrowing and returning.

Epic 5: reminders and notifications

As an admin,
I want to be able to notify users when their needs to be checked in,
so that I can make sure students do not exceed deadlines for their checkouts.

- Check-In/Check-Out Equipment: Regular users can borrow and return equipment.
- See Personal Borrowing History: Regular users can view their personal history of borrowed equipment, including check-in/check-out dates, and due dates.
- List Department Equipment: Regular users can view a list of all department equipment including item status, availability dates, and other item details.
- Manage Equipment: Admins can add and remove departmental equipment from the system.
- Manage User Permissions: Admins can toggle user roles (regular vs. admin permissions).
- Generate Usage Reports: Admins can generate detailed reports showing the statuses of all equipment, the users that currently have equipment out on loan, and return dates.

- Automated Reminders and Notifications: The system should send automated reminders when due dates approach. Past-due notifications should be sent to both users and admins. Admins should also be alerted when equipment is checked-in and checked-out.
- Equipment Status Verification on Return: Following equipment return, admins should have the ability to assert that they have checked the equipment and that the most recent user returned the equipment in good condition. If the item was returned damaged, then admins can enter additional information describing the damage.

Non-functional Requirements:
- CheckPoint must be accessible via a web interface; however, a mobile application may optionally be developed to improve the user experience during the check-in/check-out process.
- Each piece of equipment must be assigned a unique identifier to ensure accurate tracking.
- Admins can perform all actions available to regular users, in addition to the actions that are only available to admins.
- All users must have login credentials before they can access the system.
- The system must support user authentication to ensure secure system usage and to differentiate between admins and regular users.

Technologies and Frameworks:
- HTML - website structure
- CSS - website styling
- JavaScript - website functionality and interactivity
- Flask - API requests
- Python, requests package

Minimum Viable Product:
- Check-In/Check-Out Equipment
- Regular users can borrow and return equipment .
- Regular users can view their borrowing history, including check-in/check-out dates and due dates.
- List Department Equipment
- Login
- Manage Equipment (Admin Only)
- Admins can add or remove departmental equipment.

Preliminary Road Map:
Sprint 1: structure all the basic html pages and start working on login page and start working on the equipment history page

Sprint 2: finalize login page and finalize the equipment history page and start on standard user controls (meet up with API and DB in this sprint)

Sprint 3: finalize standard users controls page, start and finalize personal equipment log (continue to meet with API and DB as needed)

Sprint 4: start and finish admin controls page. (meet with API and DB to insure the admin users will be distinguished separately from the regular users)

Sprint 5: check-in/ check-out and reservation pages, also start on notifications and reminders for equipment. (meet with API and DB to make sure the DB has a way to manage availability)

Sprint 6: make the complete product pretty.