Project Plan Lightning Talk

Small Equipment Checkout Locker Software

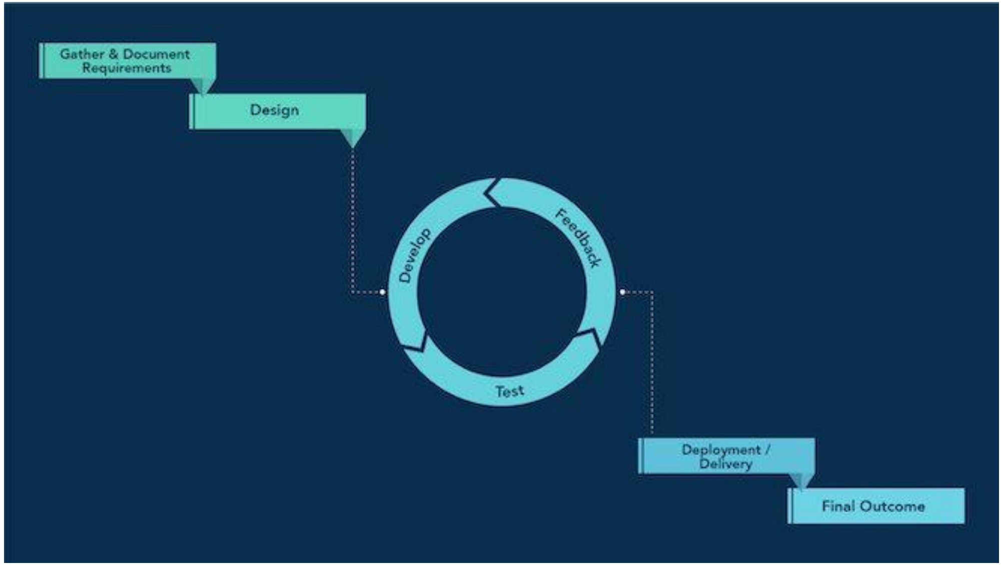
Ainara Machargo del Rio, Camille Cramer, Laura Mejia, Jon Gonzalez and Ben Johnson

sddec23-03

Problem Statement

Engineering students usually need certain materials/components to complete numerous assignments.
Since their workload is very heavy, they often need to remember to rent this material when the ETG is open. So when they start completing the assignment, they realize they still need that certain component, but the ETG is closed. With this set of lockers, we hope to help these students when they are desperate to complete their assignments on time before the next day when the ETG opens.

Problem Management Style



We will be using a hybrid waterfall + agile project management style. We are adopting this approach due to the dependencies between the different parts of our project.

Task Decomposition

- Front-end webpage
    - Allows users to login, select items to checkout, and open/close the locker
    - Includes admin page for ETG staff to monitor checkout history
- Database
    - Stores all user data needed for the application
- Raspberry Pi
    - Runs a server to connect the front-end and database
    - Operates the locker hardware as needed to checkout items

Project Schedule

- **Milestone #1:** Allow a user to open/close a locker remotely
- **Milestone #2:** Record basic locker history (opening/closing) in the database
- **Milestone #3:** Allow a user to open/close a specific locker and fully build out the API for all locker operations (open/close/turn on lights/check sensors)
- **Milestone #4:** Record all necessary user data (user, current time, reservation time, locker number, return time, and any other data requested from the client) from locker operations
- **Milestone #5:** Integrate NetID sign in order to access the locker
- **Milestone #6:** Set up touch display to operate the locker

Risks and Mitigation

- Raspberry Pi hardware failure
  - Could cause loss of user data
  - May need to set up software on new pi
- Network outage
  - Could prevent access to the locker
  - Client expects this to occur
- Software error
  - Bugs are almost impossible to avoid

- Creating a backup database
  - Ensures we can recover data in the event of an incident
- Dockerizing the application
  - Allows us to quickly redeploy the application on new hardware
- Allow touch display to interact with Pi without an internet connection
- Develop a thorough test plan
  - Helps us find bugs early

Personnel Effort Level

Tasks	Time
UI	40+ hours. UI must be intuitive and functional. Must be able to display login/welcome and main checkout locker pages using the ISU template.
User Authentication/User Data Recording	40+ hours. Must set up a backend with authentication, front-end login page, back-and-forth communication, and recording user data.
Database Structure	30+ hours. The database needs to be able to keep track of users, rental items, and locker information.
Raspberry Pi and Solenoid Connection	30+ hours. Pi must be able to operate the locker hardware. This includes opening and closing the locker, turning on and off the lights, and reading data from sensors.

QUESTIONS?