ETG Checkout Locker

Design Document

Group Number: sddec23-03
Client: ISU Electronics and Technology Group
Advisers: Matt Post
Team Members/Roles:
Ainara Machargo Del Rio: Full Stack Developer
Camille Cramer: Backend Developer
Jon Gonzalez: Frontend Developer
Laura Mejia: Frontend Developer
Ben Johnson: Backend Developer
Team Email: sddec23-03@iastate.edu
Team Website: http://sddec23-03.sd.ece.iastate.edu/

Revised: April 23/v1.0

Executive Summary

Development Standards & Practices Used

- This project will be developed with a hybrid agile/waterfall approach
- The following IEEE standards for software will be followed:
    - IEEE 23026-2015, Systems and software engineering - Engineering and management of websites for systems, software, and services.
    - IEEE 26515-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering - Developing information for users in an agile environment.
    - IEEE 7002-2022 - IEEE Standard for Data Privacy Process.

Summary of Requirements

- A working web server that runs on a Raspberry Pi.
    - Capable of operating hardware to open and close specific lockers.
    - Records all transactions in a database.
    - Authenticates users with their NetID login or ISU ID card.
    - Compatible with both a web frontend and a touch screen display.
- A frontend for the web server where users can login and reserve items.
    - Allows users to save an item to pick up later.
    - Authenticated with NetID login.
- A GUI to run on a touchscreen where users can select items to check out.
    - Displays current items available in the locker.
    - Authenticates users with their ISU ID card.

Applicable Courses from Iowa State University Curriculum

- COM S 309
- COM S 228
- CPR E 288
- COM S 363
- S E 319

New Skills/Knowledge acquired that was not taught in courses

- Technical Research.
- Defining project requirements.
- Creating documentation for a project.
- Analyzing trade offs for system design.
- Operating hardware from a web server.

Table of Contents

1 Team

1.1 Team Members

- Ainara Machargo del Rio.
- Camille Cramer.
- Jon Gonzalez.
- Laura Mejia.
- Ben Johnson.

1.2 Required Skill Sets for Your Project

- HTML, CSS, and JavaScript programming experience.
- MySQL and MariaDB knowledge.
- Python and Flask programming experience.
- API and System Design experience.
- Raspberry Pi experience.

1.3 Skill Sets covered by the Team

- HTML, CSS, and Javascript: Jon Gonzalez and Laura Mejia.
- MySQL and MariaDB: Ainara Machargo Del Rio, Jon Gonzalez, and Camille Cramer.
- Python and Flask: Ben Johnson, Camille Cramer, and Ainara Machargo del Rio.
- API and System Design: Ben Johnson and Camille Cramer.
- Raspberry Pi: Ben Johnson.

1.4 Project Management Style Adopted by the team

- We will be adopting a hybrid agile/waterfall approach for managing our project. This will allow us to set targets for sprints, while completing certain parts of the project in order.

1.5 Initial Project Management Roles

- Ainara Machargo Del Rio: Full Stack Developer.
- Camille Cramer: Backend Developer.
- Jon Gonzalez: Frontend Developer.
- Laura Mejia: Frontend Developer.
- Ben Johnson: Backend Developer.

2 Introduction

2.1 Problem Statement

ETG is a great place to rent out equipment for ECpE courses, but it has its limitations. ETG closes when people still need to check out equipment for their courses, and rentals take away precious time of ETG employees to focus on their work. On occasion, ETG staff might take some time to attend students that need rentals, which can frustrate students. Despite this, ECpE students still

need to get their equipment to complete their projects, and ETG employees need to go home after a long day of work. To solve this, our team is going to work on creating a small equipment locker that would be placed in the TLA so that students no longer need to struggle with ETG wait times or closing times, and ETG staff have more time to focus on their other tasks.

2.2 Requirements & Constraints

Functional requirements:

1. The equipment locker should be able to keep track of the inventory of equipment available for check-out and return.
2. The system should allow students to easily check-out and return equipment through a self-service process via a touch screen.
3. The equipment locker should open itself according to the equipment that is being checked out.
4. The system should be able to identify each individual student and ensure that they are authorized to check-out equipment via a ISU ID scanner.
5. The system should be able to send email alerts to students when equipment is checked out, due or overdue for return.
6. The system should have a reservation portal that students can access from their browser that will allow them to reserve the equipment for a specified amount of hours so no other students can check that specific equipment out while the reservation is going on.
7. ETG staff should have a website that allows them to view what equipment is available, who is currently renting what equipment, who has equipment due or overdue, and the ability to add and remove equipment from the locker system.

Resource requirements:

1. The lockers should be large enough to fit the small equipment.
2. The touch screen should be responsive and large enough to allow students to easily check out by interacting with the interface.
3. The equipment locker should be sturdy and durable to withstand frequent use.
4. The system should have a reliable power source to ensure proper functionality.

Qualitative aesthetics requirements:

1. The system's user interface should be intuitive and easy to use.
2. The locker should be able to fit into the surrounding environment in the TLA seamlessly.

Economic/market requirements:

1. The equipment locker should be cost-effective to install and maintain.

Environmental requirements:

1. The locker and system should be energy-efficient to reduce the environmental impact.

UI requirements:

1. The interface should have clear instructions and visual cues to guide users through the check-out and return process.
2. The reservation website should use the same HTML template that is used university-wide for uniformity and familiarity of the user.
3. The admin site ETG staff can use should be easy to navigate and make the adding and removing of equipment quick and easy. The information about rentals should be easily accessible and readable.

Constraints:

1. Testing can be difficult as we need the Raspberry Pi in hand to do it. This can push back some of the progress we make since we can only know if it truly works once we test it on the Raspberry Pi in the Senior Design lab.
2. Working with Student information means working with sensitive data. Therefore in order to be cautious, we will need special permission from the University before we can test student log-ins using their NetID. Getting these permissions will take out some of the time we could be using to test our project.

2.3 Engineering Standards

IEEE 23026-2015, Systems and software engineering - Engineering and management of websites for systems, software, and services information This standard is likely to apply to our project because it has to deal with how to manage and create a website. The scope states that this standard should be used by those responsible for managing a website. Our project will require us to create a website and maintain it for the duration of our project, until it is given control to the ETG staff.

IEEE 26515-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering — Developing information for users in an agile environment
This standard is likely to apply to our project because the standard lays out how a project should be developed using the agile methodology. As a team, we decided that we would be implementing the agile methodology in the development of our project.

IEEE 7002-2022 - IEEE Standard for Data Privacy Process
The scope of this standard states that this is meant to be implemented when projects use personal data that needs to be/ should be protected. This standard likely applies to our project because to authenticate that a user is allowed to check out equipment, the student will have to scan their student id, and the net-id associated with the student will be recorded in the database. The student's net-id should be considered personal identifiable data

2.4 Intended Users and Uses

Our intended users are ECpE students and ETG staff.

ECpE students will use the small equipment checkout locker when:

1. ETG is closed and the student needs to check out equipment to complete an assignment.
2. ETG staff are busy and do not have time to help the student with a checkout.
3. A student wants to quickly receive equipment without human interaction.
4. A student is in the TLA studying and can access equipment within the room.

ETG staff can use the small equipment locker when:

1. They are busy with other work and can tell students to check out their equipment using the small equipment locker.
2. Easily keep track of what equipment is available, who has it, and when do they have to return it.

3 Project Plan

3.1 Project Management/Tracking Procedures

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. In order to effectively test the application, we need each part to be at least partially functional. Therefore it makes sense to use a waterfall method in some fashion. However, the flexibility that the agile style provides makes sense for our project, since several of the requirements may change depending on the client's needs. Therefore, we feel that it makes sense to combine the 2 project management styles for this project.

Our team will be using Github to track the progress of our project. We will be making 'issues' on Github, where we will describe the next task and assign members of the team to complete them. Since our project is mainly software based, we will be uploading our code to the team's repository. Github will track all our progress through our 'pushes' and distinct branches that we will create while we work on the project.

3.2 Task Decomposition

- **Page Structure:**
    - Home/Login Page: This HTML page uses the ISU template to briefly overview the small equipment locker checkout system and prompt users to log in with their Iowa State identification information.
    - Main Checkout Locker Page: This HTML page uses the ISU template to display all components available to checkout (similar to how the ETG website does now). Users click on their desired component and enter the date/time they need it. Lastly, the user must agree to the terms and conditions by clicking a checkbox.
    - Confirmation Page: This HTML page will display that the locker is open and their rental has been processed correctly. It will include the confirmation number and when the rental item(s) should be returned.
    - Confirmation Email: The user/lender will receive a confirmation email that has details about the rental such as the confirmation number, when the item was taken from the locker and what day it's supposed to be returned.
- **Database Structure:**

- The database would be a SQL relational database. This database will have 7 tables:
  - User - Stores users' net ID
  - Rental - Keeps track of details regarding a rental a user makes, such as the rental start and end date and status.
  - Item - Stores information about items that can be in the locker and how many of them are available.
  - Rental Item - Keeps track of a specific item that is being rented.
  - Item type - Stores details about the type and category an item can be.
  - Locker - Contains information about lockers and their availability.
  - Locker availability - Stores specific information about the locker's availability such as the start and end date.

  These tables will reference each other using foreign keys as depicted in the ER diagram in section 4.3.1 to allow these tables to interact and to be able to correctly keep track of the information that is needed in the database.

- **Raspberry Pi and Solenoid Connection:**
  - Flask Backend: The Pi will be running a Flask backend to handle both web server requests and opening and closing the locker. The API will read and write to the database and will call the Locker Controller methods as needed.

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

- **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

3.4 Project Timeline/Schedule

Task name	Due date	Percent of Project	Hours Requ...
To do			
Milestone #4	Apr 4 – Sep 4	16%	50
Milestone #5	Aug 21 – Sep 4	16%	30
Add task...		SUM 32%	SUM 80
Done			
Milestone #6	Apr 18 – May 2	16%	30
Milestone #3	Apr 4 – 18	16%	40
Milestone #2	Mar 21 – Apr 4	18%	40
Milestone #1	Mar 13 – 21	18%	30
Add task...		SUM 68%	SUM 140

Figure 1. Milestone Breakdown

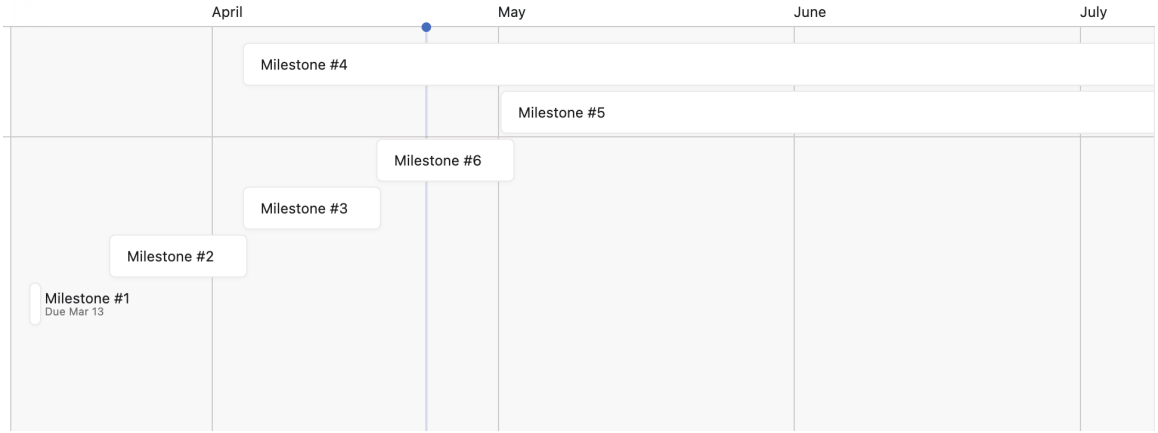

Figure 2. Milestone Calendar

3.5 Risks And Risk Management/Mitigation

Each risk is annotated with the expected probability of it occurring over the lifespan of the product

Page Structure:

- Risks:
  - Unresponsive pages | 0.5
  - Unable to checkout a component that appears available | 0.5
  - No email confirmation | 0.5
- Possible Solutions:

- Test the website regularly to ensure there are no bugs that might make it unresponsive.
-

Database Structure:

- Risks:
  - Performance issues due to high user traffic | 0.5
  - System downtime due to hardware failure | 0.4
  - Inadequate data backups | 0.5
- Possible Solutions:
  - Optimize database queries and design, implement caching, and scaling up any server resources.
  - Monitor the PI's system performance and health and take redundancy measures.
  - Regularly backup data to a cloud service or external storage device and routinely perform data integrity checks.

Raspberry Pi and Solenoid Connection:

- Risks:
  - Hardware Failure: There is the possibility of a hardware failure on the Pi. This could require the software to be set up on a new Pi | 0.5
  - Network Failure: The project will depend on the ISU network. If there is an outage, that will impact the locker's ability to function properly. |0 .9
- Possible Solutions:
  - Create a backup database on a separate server to have a live copy of all user data.
  - Dockerize the application to allow it to be quickly spun up on a new machine.
  - Allow the touch display to interact with the pi without needing a network connection.

3.6 Personnel Effort Requirements

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.

3.7 Other Resource Requirements

- Lockers and Raspberry Pi - They have been provided to our group by ETG.
- External memory or cloud service - In order to back up the information stored in the database, we are considering having an external memory or using a cloud service to be able to ensure that no information is lost.

4 Design

4.1 Design Context

4.1.1 Broader Context

This project is designed for Iowa State students and faculty members. The project will only affect the needs of students, depending on the tools or materials they need for their classes.

Area	**Description**	**Examples**
Public health, safety, and welfare	Our project affects the mental health of students. By providing a way to access equipment outside of ETG's business hours, students will be less stressed about making their project deadlines.	By providing a way to access equipment outside of ETG's business hours, students will be less stressed about making their project deadlines.
Global, cultural, and social	Our project affects the social community of the ECpE department at ISU by improving access to learning tools and materials.	Students could check out tools for classes after ETG closes.
Environmental	Our project can indirectly help the environment by reducing waste. There are not many environmental impacts associated with this project.	Reduce waste associated with item storage
Economic	With our project we will be helping ECPE save hundreds of work hours every semester. Renting equipment at ETG takes a lot of time that staff could be using for projects they may be working on, this project will take care of that. This will also help the student save money since they won't have to rely on buying	By reducing the interactions between ECPE students, ETG staff will be able to handle any complex requests they may get and work on any other projects they may have as well. Students won't have to depend on ETG hours or having their own

	their own equipment if they need it after ETG is closed.	equipment if they need something for a class. This will help the student save money and time as well.

4.1.2 User Needs

Iowa State Engineering Students need a way to access components at any time to complete assignments on time and succeed in their classes.

ETG employees need a way to provide Iowa State students with components used in their assignments without being open 24/7.

4.1.3 Prior Work/Solution

Similar to our project are the Amazon Lockers used for users to pick up their packages. When users order something from Amazon, they can have it shipped to the nearest/desirable Amazon hub locker. When their package is delivered, they will receive an email containing a unique code and instructions to pick up their package at the chosen location. Once they enter this unique code, a locker will open so they can grab their package and leave.

Advantages:

- Ability to pick up the package at the user's schedule.
- Secrecy. Your package will not be left on your front door for others to see but in a safe locker for you or whoever you decide to pick it up.

Shortcomings:

- Some of these lockers are placed inside stores or apartment buildings and have limited available hours.

Our solution:

Pros:

- Accessibility to lockers 24/7 since they will be placed in Coover hall, and the building is open anytime for students/faculty.
- More secure. Students must scan their student ID rather than using a unique code.

Cons:

- Only 35 lockers.

4.1.4 Technical Complexity

1. The design consists of multiple components/subsystems that each utilize distinct scientific, mathematical, or engineering principles.

The project design needs to include several different components that operate in sync. These components include a modern front end that conforms to the provided standard that is capable of running on web devices and a touch screen display, a Flask API running on a Raspberry Pi Server, a card reader to scan NetIDs, and a Database to store the information for the project. Each of these components must be able to interact with each other.

2. The problem scope contains multiple challenging requirements that match or exceed current solutions or industry standards.

   The project will be implemented using the industry standard tools. We will use a Javascript/HTML/CSS front end and a Flask API for the backend. The database will be implemented with a MariaDB database running on the Pi. All of these components are industry standard.

4.2 Design Exploration

4.2.1 Design Decisions

1. Host server on the Pi vs separate server
2. What type of Database to use
3. How to most reliably interact with the hardware

4.2.2 Ideation

For hosting the server on the Pi vs on a separate server we identified several options

1. Use ETG provided server to host only the database and host the API on the Pi
2. Use ETG provided server to host database and API, connected with hardware via separate API on the Pi
3. Run both the database and the API on the Pi
4. Run the database and the API on the Pi with a backup database on an ETG server
5. Host the database and the API on the cloud and connect with the hardware via a separate API on the Pi

4.2.3 Decision-Making and Trade-Off

We decided to go with option 4 after talking through the trade offs of each decision with our client. The issue with options 1, 2, and 5 are that in the event of a network outage, the locker would be unusable. The advantage with the database being separate from the hardware is that in the event of the Pi crashing, there is no chance of losing the database data. To increase reliability in the event of an outage, we came up with the idea of running both on the Pi. The downside to this, option 3, is the chance of the hardware failing. This led us to option 4, adding a backup database to ensure the reliability of the system at all times, no matter what.

4.3 Proposed Design

So far, we have implemented and tested the solenoid and raspberry pi connection and have the basic ISU HTML/CSS template set up for the website. We are currently working on getting the back and front end to communicate. We also have a working Flask API that is able to operate the solenoid remotely when the end point is reached.

4.3.1 Design Visual and Description

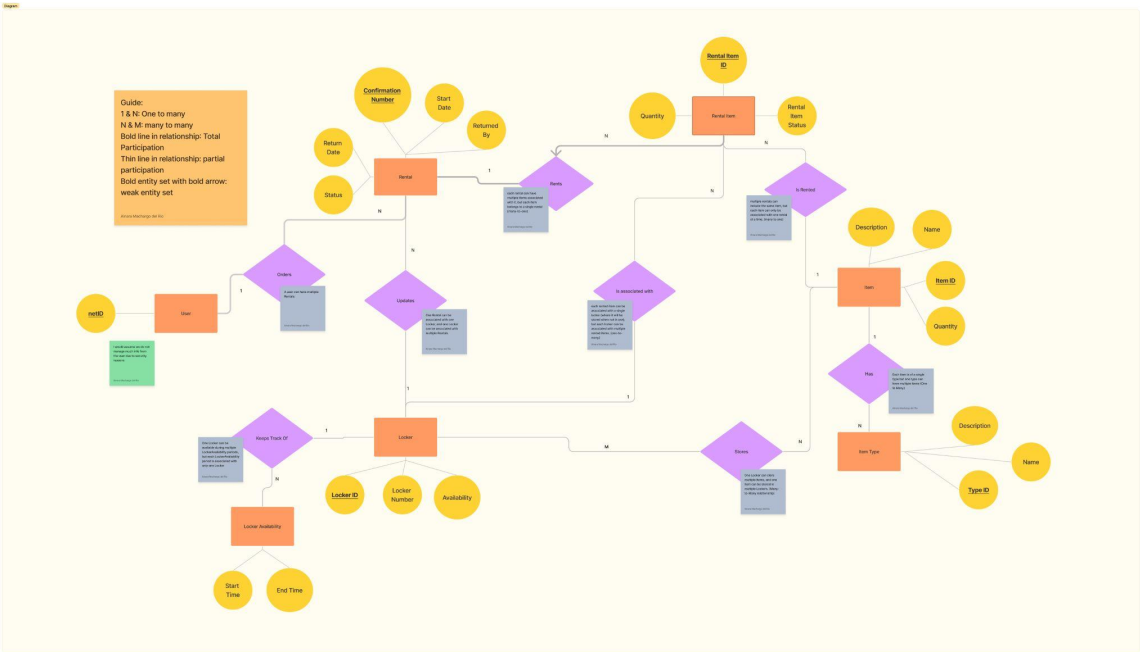

Figure 3: ER Diagram for the database

- User - Stores users' net ID
- Rental - Keeps track of details regarding a rental a user makes, such as the rental start and end date and status.
- Item - Stores information about items that can be in the locker and how many of them are available.
- Rental Item - Keeps track of a specific item that is being rented.
- Item type - Stores details about the type and category an item can be.
- Locker - Contains information about lockers and their availability.
- Locker availability - Stores specific information about the locker's availability such as the start and end date.

4.3.2 Functionality

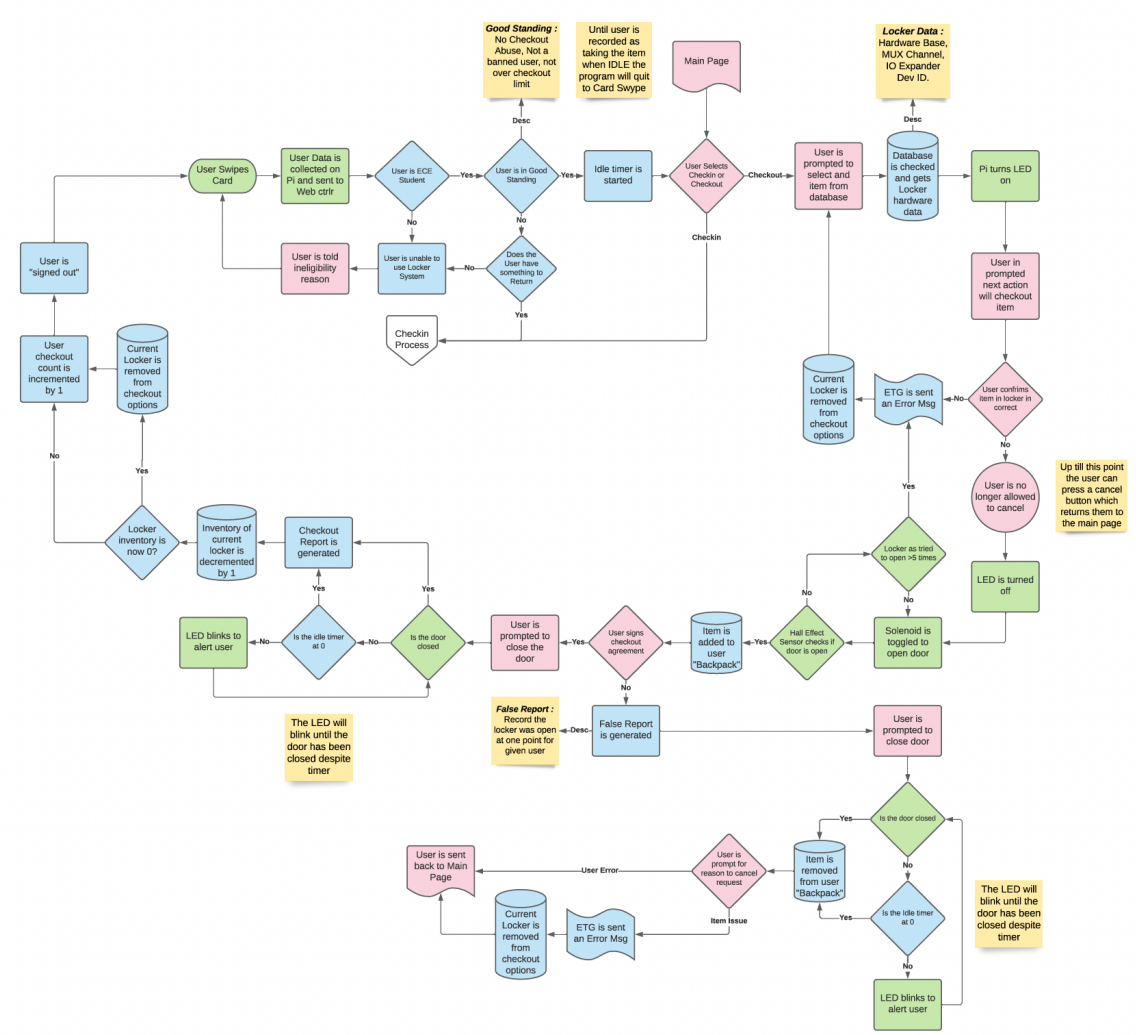

Figure 4: Function Diagram

4.3.3 Areas of Concern and Development

The biggest concern is what types of disasters or system failures do we need to plan for in the project. This affects how we will approach the high level design.

Currently, for the issue of system failure and backups, we plan to have a server to store backups that will be updated daily. We have already consulted with our client and advisor and they agree that this would be an adequate solution for our problem.

4.4 Technology Considerations

Strengths:

1. Raspberry Pi is a cost-effective and compact solution for running the system.
2. The touchscreen GUI provides an intuitive interface for users to check-out and check-in equipment.
3. The SQL server provides a reliable and efficient database management system for inventory tracking.
4. Flask is a lightweight and easy-to-use framework for building the website for equipment reservations and admin purposes.

Weaknesses:

1. The Raspberry Pi may not be able to handle a large number of concurrent users, which could lead to slow performance or system crashes.
2. The touchscreen may not be as durable as a physical keyboard or mouse, and may require more frequent maintenance.
3. Using a separate website for equipment reservations may add complexity to the system and could potentially cause confusion for users.

Tradeoffs:

1. Using Raspberry Pi and Flask allows for a more affordable and lightweight solution, but sacrifices some performance and scalability compared to more powerful hardware and frameworks.
2. Choosing SQL server for the database management system provides reliability and efficiency, but may require more setup and configuration compared to cloud-based solutions.

Possible solutions and design alternatives:

1. To address the durability of the touchscreen, a protective case or screen cover could be used to prevent damage from wear and tear.

4.5 Design Analysis

We are currently working on the proposed design made in section 3.3 . We are on track to finish our project early next semester. Flask has been a great way to integrate the frontend and backend seamlessly without much overhead from the system. We already have the touchscreen up for testing, we are able to open and close lockers remotely, and we have already created a SQL database that just needs to have some information populated in it. However, we have added more milestones as requested from our client.

We have had to modify our original design by adding more milestones as requested by our client. We have started creating a simple GUI using Python and GUIZero that will be displayed on the touch screen that will allow students access to their reservations, check out equipment, and return equipment with ease. We also have a new milestone that is creating a website that allows ETG staff access to add and remove equipment from the database and to see information inside of the

database (such as what equipment is currently checked out, what equipment is overdue or due, and who has what equipment) for ease of use of the database and its contents.

4.6 Design Plan

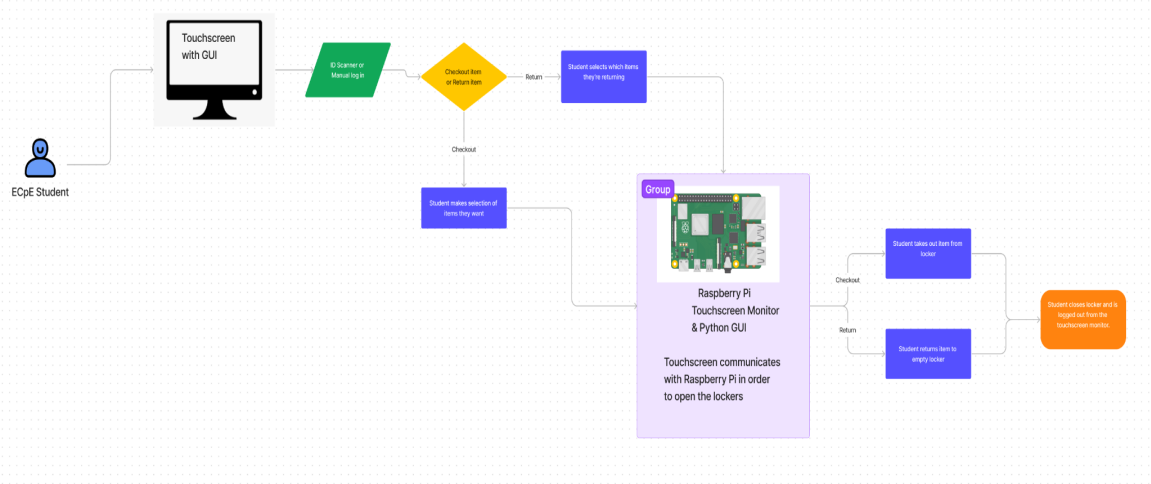

Figure 3: System Diagram

The diagram above shows a user would interact with the touchscreen and lockers in order to checkout or return items. The student will interact with the touchscreen, which will have a GUI that asks for the student to login with their Student ID card or to manually enter their NetID information. The student will then have the option to checkout items or return any items they had previously checked out. The touchscreen will then communicate with the Raspberry Pi, which will give the command to deactivate the solenoid inside of the locker, so that the locker can open. After retrieving or returning the selected items, the student will then close the locker and will be automatically logged out of the touchscreen so that another student can use it.

5 Testing

5.1 Unit Testing

The team will be testing if data is being recorded and collected from our database, to ensure the data in the MySQL database is correct. We will test the backend connection to the solenoid, checking whether or not it is opening and closing as it should, and test the GUI for the website as well. The team will test individual components of the GUI, such as buttons, labels, and text fields.

To do this we will be creating unit tests on Python that will create “dummy data” and will be used as input towards our functions. Python contains a library called unittest that will be a helpful tool for our Unit Tests. These tests will check the performance of our code and will help us identify any errors the team might’ve missed.

5.2 Interface Testing

Since the team will be creating a web application with a separate GUI for the touch screen, we will be testing the interface for both. As mentioned above, we will be conducting unit testing to the GUI on individual components like buttons and texts. The team will also be performing end-to-end testing and functional testing, which means that the entire GUI will be tested by simulating user behavior and testing the GUI in different scenarios.

To help with these tasks, the team will be using a library called PyAutoGUI, which simulates user inputs like mouse clicks and keyboard presses. In addition, we will be using the library Selenium, commonly used for web application testing and GUI testing. These libraries will be helpful tools when testing our project's interface on a computer and on the touch screen.

5.3 Integration Testing

To test that the API is properly integrated with the hardware and database, the team will use the PyTest suite and Postman. This will ensure that the expected data is being written to the correct place at the correct time. These tools will allow us to make sure that the application's backend is properly functioning with the rest of the project.

5.4 System Testing

We plan on our overall system testing to be conducted with a combination of unit tests, interface tests, and regression tests that have been laid out in sections 4.1, 4.2, and 4.3. All of these tests will be run on each new build of the project before it is deployed. These tests will cover the complete functionality of the project and include all of the components that make up the system.

5.5 Regression Testing

We will be testing each addition to ensure these stay within the project. Since we will be dividing the project load into small tasks, each team member can test their code and, at the same time, test the project once their code is merged.

Our critical features are:

- Lockers open and close on user's demand. (This will be tested in the Senior lab where the lockers are located).
- Users can use the reservation website. (This can be tested by running the website and ensuring one can reserve an item).
- Users can use the touchpad to get an item. (This will be tested in the Senior lab where the lockers are located).
- Admin can use admin website. (This can be tested by running the website and ensuring one can view correct rental info, and add and remove items from the lockers).

5.6 Acceptance Testing

Our main goal is for the project to be done at the end of this semester. So we can use the summertime to test our design and ensure every requirement is met. These will allow students to give feedback, and we can work on it over the fall.

5.7 Security Testing (if applicable)

5.8 Results

At the end of our tests, we will have a comprehensive overview of what aspects of the system are not functioning as intended. Any errors in the code will be caught by the unit tests we will be implementing, and any errors in the functionality will be found by the interface and integration tests. A successful test suite will ensure that the code is performing as expected and that the system is functioning as expected.

6 Implementation

Part of our design currently is to have full functionality by the end of semester one. Next semester should be testing and making small improvements based on testing and the student feedback. The improvements will probably be focused on usability and accessibility.

7 Professionalism

This discussion is with respect to the paper titled "Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

Work Competence	Tenet numbers 3, 5, and 7 in the IEEE Code addresses this. Work competence can be demonstrated by accurately estimating deadlines and costs, by understanding how to apply the tools that are used, and by fair and honest feedback to coworkers and by accepting feedback on our own work. All of these are a crucial part of work competence. The IEEE code is much more brief and less explicit in a lot of areas than the NSPE is when it comes to competence.
Financial Responsibility	Tenets 2, 3, and 4 address this area.

	Conflicts of interest are always a problem in our line of work. Our responsibility is to our employer. Inaccurately estimating costs is an example of something covered by tenant 3 that falls under financial responsibility. Tenet 4 is about bribery. Accepting brides is never a good idea. The IEEE code has a lot more that is focused around financial responsibility than the NSPE code does.
Communication Honesty	Tenet 3 of the IEEE code addresses this directly. Tenet 3 is about honestly representing data and not misleading others. This is the only tenet that directly addresses the importance of honesty in Engineering. The NSPE code has an entire paragraph dedicated to this and it's much more in depth than the single sentence that the IEEE code has. Tenet 8 also addresses this by treating everyone fairly and with respect.
Health, Safety, and Well-being	Tenet 1 of the IEEE code addresses this almost word for word. The IEEE code says "health, safety, and welfare". This is the area that IEEE addresses in almost the exact same way as table 1. This is because of how important this tenet is. Almost all of the NSPE code indirectly addresses this tenet. It is the most important responsibility of any engineer.
Property Ownership	Tenets 3 and 8 of the IEEE code address this. Engineers should be respectful of others' thoughts and ideas. They should also respect the data and information of the company. This is probably the area that the IEEE code addresses the least. The NSPE code is much more in depth about the respect engineers should have for others.
Sustainability	Tenets 1 and 5 address this area. Sustainability is something that falls under the heath, safety, and welfare of others. Harming the environment indirectly harms others. Tenet 5 addresses knowing the consequences of any actions. Knowing any potential sustainability impacts from a system is a great example of how tenet 5 is

	important. The NSPE code does not address sustainability apart from the health and safety concerns.
Social Responsibility	Tenets 1, 5, 8, 9, and 10 address this area. Tenets 1 and 5 are important for the same reasons as sustainability. Tenet 8 is important because engineers have a responsibility to treat others with respect. Tenet 9 is relevant because of our responsibility to the health and safety of others. Tenet 10 also fits here, since our responsibility to others also involves teaching and helping them. This is an area that both IEEE and NSPE cover very well. This may be due to how Social Responsibility and the health and safety of others are so tightly related.

7.2 Project Specific Professional Responsibility Areas

a. **Work Competency:** This is a critical area that applies to our project. We need to be able to complete the tasks that have been assigned to use as a part of the project. In doing so, we need to be following the timelines that have been set for use and performing to the best of our abilities. Our performance so far has been a medium. We have completed our high level design and have prepared to demo code in the near future but there is always room for improvement in this category.
b. **Financial Responsibility:** This is an area that doesn't apply to our project. Technically it still does, but in practice, we have no funding allocated to our team and do not expect to spend anything in order to complete the project. All of the materials have been given to us and we do not require any paid software. This area isn't really applicable to our project, but our performance is very high in this category since we haven't spent any money and don't plan on doing so.
c. **Communication Honesty:** This area is also important to our project. We need to be honest and direct with our client about where we are on the project and how likely we are to hit our deadlines. This has been a strong point for our group. Our performance so far has been high. We have been communicating with both our client and the ETG staff that we need to work with in order to complete the project.
d. **Health, Safety and Well-being:** This area does not apply to our project. While the health and safety of others comes first in engineering, everything we are doing on this project is software based. There is no potential to harm others. While we will keep this area in mind at all times, it does not apply to our project. In the context of our project, our performance has been high, because there is no potential to harm others in any fashion, ever.
e. **Property Ownership:** This area applies to our project. It is important for us to respect the ideas, property, and information of our client. From the hardware that we are working with to the interactions we have with the ETG staff, we need to always focus on respecting our client. This can take the form of listening to their ideas and not dismissing them, even if we disagree and it always means that we will take care of the hardware we work with. Our

performance in this area has been high. We have been respectful in meetings and been careful to take care of the hardware.

f. **Sustainability:** This area does not apply to our project. Since the hardware we will be using has already been created and the rest of the project is software based, there is little potential for waste. The only way we can impact sustainability while working on the project is to keep our code efficient to reduce the potential power draw on the pi. Since this impact would be almost unnoticeable, there is no way for us to impact the environment on this project. As such, our performance in this area is high. This is because we will have no negative environmental impact over the course of creating this project.

g. **Social Responsibility:** This area does apply to our project. This is because the purpose of this project is to help other students. It is our responsibility to help them find a way to checkout equipment when ETG is closed. This can help improve students' lives by allowing them more flexibility in planning when they will work on their homework and by allowing them to work on projects after ETG closes at 5pm. Our performance in this area is low, because while we are having a positive impact on students, there is no greater contribution to society outside of the ECPE department.

7.3 Most Applicable Professional Responsibility Area

Work Competency: Work competency is an important area to both our project and our team. Our main purpose for this project is to give access to students to request materials/tools out of normal staffed ETG hours. We need to create a user-friendly interface that interacts with the lockers promptly and safely. Although we have just started working on the project, we have created a high-level design and coded components for the front end. We still need to perfect a few ideas for the back end, such as database changes, etc.

One way to keep us on track in this category is to work closely with our advisor and client to ensure we meet the specific requirements and deadlines. By doing so, we can adjust deadlines as needed and ensure that we meet the expectations that our client has on this project. Meeting frequently with our client also ensures that we will be consistently making progress on the project.

Another way to keep on track in this category is to work together when possible. This will decrease the likelihood of making a mistake and improve everyones' knowledge of the overall system. This is important because everyone on the project should know how the system works and how to fix it if something goes wrong. If everyone knows how the system works, this will improve the overall work competency of the group and thereby improve the outcome of the project.

8 Closing Material

8.1 Discussion

We feel that the design we have presented to our client satisfies the requirements for the project. Our design will enable students to use the software without any training, while being maintainable for ETG staff in the future. The prototype we have developed will hopefully be tested over the summer and we can take the feedback and improve the design in the fall. Currently, users can use the touch screen to check out and return specific items. We hope to include more database integration before the end of the semester.

8.2 Conclusion

This semester, we have created a design for the system and began implementing it.Currently, we have a web frontend to reserve items, a GUI for the touch screen, a Flask server, and a MariaDB database set up and running. Our goal is to have these functional enough to test over the summer. In the fall, we will work on improving the system and adding more functionality such as email confirmation and backing up the database. In the future, we will need to change the system architecture to accommodate the email system and the backup database. Overall, significant progress has been made this semester and the project is on pace to be completed next semester.

8.3 Team Contract

Team Members:

1) Ainara Machargo del Rio
2) Camille Cramer
3) Jon Gonzalez
4) Laura Mejia
5) Benjamin Johnson

Team Procedures

Day, time, and location (face-to-face or virtual) for regular team meetings:

Tuesdays at 4:00pm in Coover Hall.

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-

mail, phone, app, face-to-face):

iMessage for team updates, reminders, issues, and scheduling, Discord for virtual meetings and sending important links/resources, emails for communication with advisors and ETG staff.

3. Decision-making policy (e.g., consensus, majority vote):

Consensus.

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be

shared/archived):

Laura will be in charge of keeping track of meeting minutes, but anyone can contribute to it if they see fit. These will be written on a google spreadsheet and shall be shared on our group's google drive folder, inside a minutes folder.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

a. **Attendance:** Everyone is expected to be present at the meeting on Tuesdays unless they have an unforeseen circumstance. If they are not present they should read the meeting minute on the google drive and ask questions to any team member if there are any.

b. **Punctuality**: Everyone is supposed to arrive at the meeting at the same time since it starts immediately after the 491 lecture ends, so there should be no punctuality issues. However, if someone is late for any reason, it should be fine as long as they make it for the majority of the meeting and do not make a habit of arriving late to meetings. Better late than never.

c. **Participation:** Each team member is expected to contribute to all meetings and activities. It is important that everyone is involved in the design process and makes contributions to the discussions.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

Team members are expected to work on every assignment before its deadline. iMessages will be sent to ensure that everybody does their part in a timely manner and with the due date in mind.

3. Expected level of communication with other team members:

a. There should be regular conversations a couple of times a week via iMessage. Everyone should read these messages whenever they are able to and demonstrate signs of life on the groupchat.

b. If any team member is stuck on their part of the project, they should reach out in the group chat or to any team member personally that could help them out instead of keeping the doubt to themselves. This is so that team members can help each other and to avoid staying in a single task for too long.

c. If any team member is experiencing a personal problem that would interfere with their performance in the project, they should mention that they are having problems on the iMessage chat as soon as possible so that the rest of the group can prepare and plan accordingly.

4. Expected level of commitment to team decisions and tasks:

a. Team Decisions: If there are any decisions to be made, someone should mention the issue via iMessage and everyone should text on the chat if they agree or disagree with the decision. For more complicated issues that cannot be solved via iMessage, we shall schedule a Discord meeting at a time where everyone is available or discuss the issue in our regular weekly meetings, whichever can be achieved first.
b. Tasks: On our project GitLab, we will create issues for each specific task, and each person should have at least one of these issues assigned to themself. There will be a couple of these issues picked out in a specific timeframe (2 or 3 weeks possibly) so that we can have a group sprint. Everyone should try their best to resolve their issues before the sprint, and bring up any concerns or questions in our weekly meetings or on the groupchat to avoid people being stuck on their issues. Whenever the issues are resolved, they should be pushed to the GitLab. If anyone is done with their issues they can pick some from the backlog if there are any.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction,

individual component design, testing, etc.):

- Leadership roles will be decided after the high level design is completed. Until then, each team member will contribute to that design.

2. Strategies for supporting and guiding the work of all team members:

- **Regular communication:** Open and honest communication is the foundation of a successful team. Sharing ideas, points of view, information, and expertise helps to keep everyone informed and in the loop.
- **Prioritize work:** By prioritizing your own work, you're making sure that someone else can prioritize theirs. It's also a useful way of keeping productive and making decisions. Since we will be using GitLab issues, we can have a tag: "goals for the week". So when we are looking for more tasks to get done, we can focus on getting those with that tag on first.
- **Set reasonable goals:** Having too much pressure to hit targets can leave people feeling burned out and frustrated. It's, therefore, important to have reasonable goals that the team as a whole can work towards. Similarly, each member should take responsibility for setting their personal goals. We can set these goals during our weekly meetings.

3. Strategies for recognizing the contributions of all team members:

Team members who go above and beyond get a sunshine sticker that week. We will allow time during our weekly meetings to shout out anyone who is doing a good job that week. Tem members are also encouraged to comment positive things on other's resolved issues on gitlab for good work.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the

team.

- **Different backgrounds:** Each of us have different backgrounds and experiences. This gives us all different perspectives on how to approach a problem.
- **Different majors:** Our team is composed of 2 Software Engineers, 2 Cybersecurity Engineers, and a Computer Engineer. This gives us a wide range of knowledge that we can use to solve a variety of problems.

2. Strategies for encouraging and support contributions and ideas from all team members:

- **Give Everyone a Chance to Speak:** During our weekly meetings, everyone will have a chance to speak and explain their ideas. The rest of the team should be attentive and actively listening. So then, they can give feedback and constructive criticism.
- **Show empathy and respect for each other:** We all have the same goal, which is to finish our senior project design. We know our classes can be time-consuming and stressful, so showing some empathy and respect for each other will create a safe environment where we can share our ideas.

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will

a team member inform the team that the team environment is obstructing their

opportunity or ability to contribute?)

- **Sharing issues**: We will take time at each meeting to share any issues that are preventing members from continuing their work.
- **Communicating with the team:** Any team member will always be welcome to share any concerns they have with the team at any time. This can be done using the iMessage chat or via our team's discord server.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

- **Positive team environment:** We will be working together for 2 semesters. It is important to have a friendly and positive atmosphere
- **Working prototype:** After meeting with our advisor, we are targeting having a working prototype ready by the end of the semester so that it can be tested over the summer.

2. Strategies for planning and assigning individual and team work:

- **Set a strong agenda for weekly meetings:** Team members should bring ideas, issues, or updates to share with the rest of the team during our weekly meetings. That way, we can take advantage of the limited time that we have to meet, and everyone can be on the same page about what they are or will be working on
- **Assess team members' skills and workload:** It is important that we match tasks according to skill set and experience, as well as availability.
- **Break down the project into smaller tasks:** Once we have discussed and set the workload for this semester and have assessed the skill set for each member, then we can break those goals into smaller chunks. This will help us set reasonable weekly goals.

3. Strategies for keeping on task:

Consequences for Not Adhering to Team Contract

- **Set clear expectations and checkpoints:** During our weekly meetings, the team should set weekly goals. Share their progress, issues, or questions so we can all be on the same page about how we are doing regarding workload. We can also use weekly meetings to demo what each one has been working on or finished working on.
- **Participate in meetings:** Team members should be ready and attentive when in meetings. If a team member is to be absent, then he/she should look at the meeting notes or send a message on the group chat to get caught up.
- **If stuck, ask for help:** To maintain a workflow, team members need to be transparent and share when they are having problems completing a task so we can figure out a plan to overcome this roadblock.

1. How will you handle infractions of any of the obligations of this team contract?

- If a pattern of infractions becomes apparent, the team will speak to the member who is responsible and let them know how they can improve their behavior.

2. What will your team do if the infractions continue?

- If infractions continue after meeting with the team, the team will hold a second meeting with the member.
- If infractions continue after the second meeting, the team will reach out to the instructors to discuss how to proceed.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the

consequences as stated in this contract.

1) _____Laura Mejia_____ DATE ___02/16/22_____
2) ____ Jon Gonzalez_____ DATE___ 02/19/22_____
3) _____Ben Johnson_____ DATE ___02/19/22_____
4) _____Ainara Machargo del Rio _____ DATE ___02/19/22_____
5) _____Camille Cramer_____ DATE ___02/19/22_____