EE/CprE/SE 491 WEEKLY REPORT 07

Mar 21 – Apr 2

Group number: 3

Project title: Small Equipment Locker

Client &/Advisor: Matthew Post

Team Members/Role: Laura Mejía, Ben Johnson, Camille Cramer, Ainara Machargo del Rio, and Jon González

(All the above information should be there in each weekly report. The format/color scheme etc need not be the same. However, please remove everything that is in a bracket from your final submission. These are just part of the template and need not be a part of the report.)

o **Weekly Summary**

The team met up with Mr. Post and presented the connection between the solenoid and the Raspberry Pi. The demo consisted of connecting one of our team member's computer to the Raspberry Pi and then sending commands so that the solenoid would deactivate(open the latch) and activate(closing the latch). During our meeting, ECPE's Mr. Jason Boyd was present and gave us great feedback in regards to how we should make our server and make a backup in case the hardware fails. The team then sent an email to Jacob Grundmeier, asking if he approved the idea of the team developing the website using Flask instead of PHP. Jacob replied that he had no problem with the team using the Flask library. In addition, the backend team met last week in order to discuss and create the database that will be used for the website. They managed to create a first prototype that will be worked on and updated as we move forward with the project.

o **Past week accomplishments** *(Please describe/summarize as to what was done, by whom, when and, collectively as a group. This should be about a paragraph or two in length. Bulleted points are acceptable as well. Please keep only your technical details related to your project. Figures, schematics, flow diagrams, pseudocode, and project related results are acceptable, but please ensure that they are legible (clear enough to read) and to provide an explanation. If researching a topic, please add a few details about what was learned and how it is relevant to the project. If two or more people worked on a single task, be sure to distinguish how each member contributed to the task. Specific details relating to the assistance provided to other members may be included here. Do not include classwork, such as individual reflection assignments, and group meetings as part of your duties.)*

- Successfully demoed the Raspeberry Pi working alongside the solenoid
- Managed to build a MySQL database that will hold the data of our users and items

Pending issues *(If applicable: Were there any unexpected complications? Please elaborate.)*
- Add the necessary code to the repository to achieve a round trip
- Test that the Server is successfully communicating with the Raspberry Pi and the team is able to open and close the solenoid from the website.

o **Individual contributions** *(Creating this section is optional, but it is **Required to include the "Hours Worked for the Week" and their "Total Cumulative Hours" for the project for each member somewhere relevant in your report. Your individual weekly hours should be at a minimum of 6-8 hours for this course. So please manage your time well. Also, ensure that individual contributions support your claim to the weekly hours. Be honest with the reports.)***

NAME	Individual Contributions *(Quick list of contributions. This should be short.)*	Hours this week	HOURS cumulative
Jon González	● Worked on the team's website ● Added code for roundtrip	4	22
Ben Johnson	● Set up Pi given to us by client ● Tested existing functionality of API endpoints	2	21
Camille Cramer	● Researched hosting database on Pi ● Worked on SQL for the backend	3	22
Ainara Machargo del Rio	● Did some backend implementation in SQL	2	24
Laura Mejia	● Worked on the team's website.	2	20

o **Plans for the upcoming week** *(Please describe duties for the upcoming week for each member. What is(are) the task(s)?Who will contribute to it? Be as concise as possible.)*
- Our goal is to have a front-end and back-end that can communicate with each other and send a request to open and close the locker. This means the backend team needs to be able to receive a request and open/close the locker doors remotely and that the frontend team needs to be able to hit a button and call the API endpoints to send that request. We also want to make the database inside of the Raspberry Pi.