

CSE 123A - Winter 2026

Group 8

Problem statement

Living with roommates can sometimes mean your filtered water device is empty when you need water. This can be very frustrating and leave you needing to drink tap water while other roommates get to drink cold refrigerated and filtered water.

Need Statement

We want a way to know if the container is out of water when we need water.

- Avoid having no water left in addition to everybody using the water filter, not knowing there is no water left.
- Need a way to know the current water level of the pitcher so that it isn't empty without everyone knowing it's empty.
- We need to know if the container is empty, in order to not run out of clean fresh water.

Goal

- Keeping water level known at all times, while informing everybody in the household if the water level runs too low.
- Create a system to inform users if the container is empty.

If the container is empty, inform users.

Personas and Users

This is intended for shared households that rely on a shared filtration water container.

- College Roommates

-
- John Doe, 20 years old, Student at UCSC. John lives in a dorm with two other roommates, and they all rely on a shared water filter for drinking water. Occasionally, John comes out to get water only to find that the water is empty. This creates frustration and delays as John has to choose between missing the bus to refill water or getting to class on time while being thirsty.

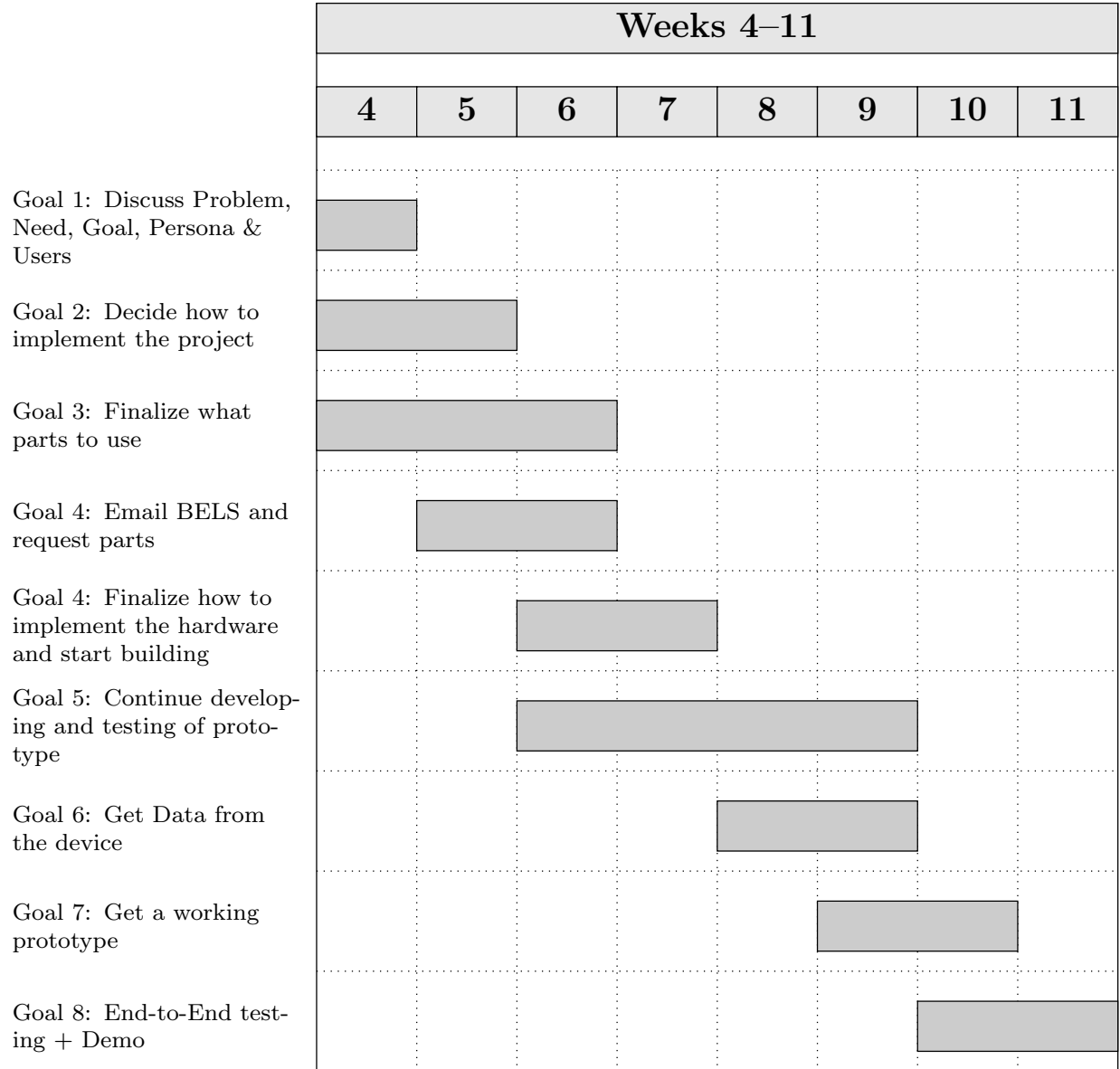
- Family

- Jane Smith, 39 years old, working mother. Jane lives with her husband and two children. Jane gets home after a long, exhausting day of work and is feeling thirsty from the drive home. She goes to the shared water filter to find that it is completely empty. Jane feels frustrated with her stay-at-home husband and two children for not refilling the water filter. This causes her to lash out and yell at them until they all cry, creating a hostile environment in the household.

- Office

- Joe Micheal, 30 years old, works with Excel spreadsheets. He has very limited time to refill his mug, and his boss, George, never refills the water filter. As a result, Joe's efficiency has substantially decreased at his desk.

Timeline



Decision Table

Items to decide on:

- Microcontroller
 - Ultrasonic: Pointed down from the lid onto a floating object
 - Load Cell: Place the container on the load cell to measure the weight and calculate the water level

-
- Laser: Shoots a laser into the water from the lid, and the distance is sent back to the sensor
 - Camera: Setup to watch the water filter and measure the water level based on computer vision
 - Non-Contact Sensor: Patch that sits on the side, when water level goes below sensor it can alert the device
- Server
 - HTTP
 - AWS
 - Vercel
 - Challenges
 - Signal in the fridge
 - Size limit
 - Contact Vs No Contact
 - Wifi connectivity
 - Power source (battery, USB, etc.)

Table 1: Sensor Decision Table

| Criteria | Weight | Design A | Design B | Design C |
|--------------------|--------|----------|----------|----------|
| Cost | 0.30 | | | |
| Performance | 0.40 | | | |
| Power | 0.20 | | | |
| Complexity | 0.10 | | | |
| Total Score | | | | |

Web Code

The goal of this section is to define the web code we will be using to display information from the microcontroller. This will eventually migrate from a web implementation to a phone app, but in the initial stage it exists as a web display.

Initial Design

In its most simple form it must be able to display static data on a web page. We have created a .html file with this basic information, shown in the following figure.

From here there are a few changes that can be made. These changes follow the line of next steps to test this code and our designs.

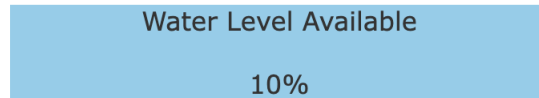


Figure 1: Output of Initial HTML Code

- Dynamically updating data with a controller (Proof of concept testing w/ simple text)
- Sending and interpreting sensor data
- Graphic interface changes

In order to test these things we will create a local server interface using one of our computers that hosts this data. Then we will use a microcontroller to send HTTP PUT requests. Finally, using the received data to change what gets displayed.

End Point of this Design

The plan is to host this using Vercel. The microcontroller is able to send data to Vercel that can be interpreted and displayed properly using Javascript. We are in the process of setting up Vercel to get this system working as intended. Once it is set up it is easy to migrate this initial testing code over to that platform and continue development from there.