

Menstrual Product Bathroom Tracker

By

Sarah Lau (lau29)

Erin Rothenbaum (eroth8)

Anna Wilkowski (annaw7)

Project Proposal for ECE445, Senior Design, Spring 2026

TA: Eric Tang

18 September 2026

Project No. 3

Table of Contents

Table of Contents.....	2
1. Introduction.....	3
1.1 Problem.....	3
1.2 Solution.....	3
1.3 Visual Aid.....	4
1.4 High-Level Requirements.....	4
2. Design.....	5
2.1 Block Diagram.....	5
2.2 Subsystem Overview.....	5
2.2.1 Sensing Subsystem.....	5
2.2.2 MCU Subsystem.....	6
2.2.3 Power Subsystem.....	6
2.2.4 Backend Subsystem.....	6
2.2.5 User Subsystem.....	6
2.3 Subsystem Requirements.....	6
2.3.1 Sensing Subsystem.....	6
2.3.2 MCU Subsystem.....	7
2.3.3 Power Subsystem.....	7
2.3.4 Backend Subsystem.....	7
2.3.5 User Subsystem.....	7
2.4 Tolerance Analysis.....	7
3. Ethics, Safety, and Societal Impact.....	7
3.1 Ethics and Safety.....	7
3.1.1 IEEE Code of Ethics I.1: Safety, Health, and Welfare of the Public.....	7
3.1.2 IEEE Code of Ethics II.1: Fair Treatment of Individuals.....	8
3.2 Societal Impact and Accessibility.....	8
3.2.1 Overall Societal Impact.....	8
3.2.2 ADA Dispenser Compliance.....	8
References.....	9

1. Introduction

1.1 Problem

Finding menstrual products on a college campus can be unexpectedly difficult. At UIUC, some bathrooms may have menstrual products available while others may be empty or not stocked at all. When someone unexpectedly needs a product, they may have to check multiple bathrooms or ask staff where products are located. This can be inconvenient, time-consuming, and especially frustrating when they are in a hurry.

There is currently no centralized way for students to determine which campus bathrooms have menstrual products available and how much stock remains.

1.2 Solution

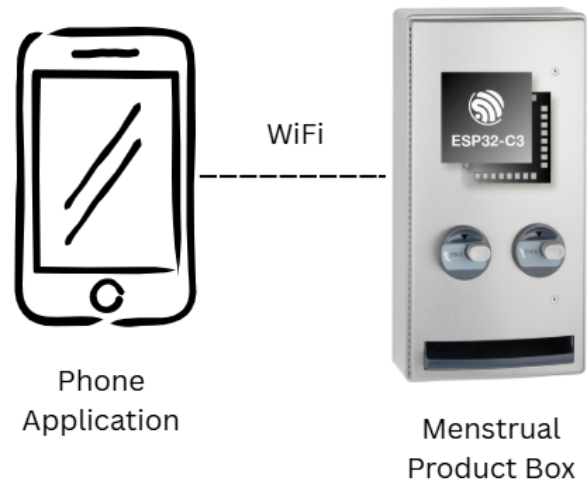
We propose an IoT-based system that monitors menstrual-product availability in bathrooms across UIUC and makes this information accessible through an app.

In boxes made specifically for the project, time-of-flight sensors would be installed on the inside of the lid; these sensors would bounce an IR light signal off the top of the period product stack, and use the time it takes for the signal to return to calculate the distance to the top of the stack from the lid. We can use the inverse of that, i.e. the distance from the top of stack to the bottom of the box (total height - distance from top of stack to lid), to measure the total height of the stack and divide by the individual height of a pad container, confirming the amount of products available. This sensor would periodically transmit its measurements to a centralized server.

The mobile application would aggregate this information and display nearby bathrooms along with their estimated product availability. Users could quickly identify the closest bathroom with products rather than searching multiple locations.

The system could also provide useful information to campus facilities staff. When a bathroom's supply falls below a predefined threshold, the system could automatically flag the location for restocking.

1.3 Visual Aid

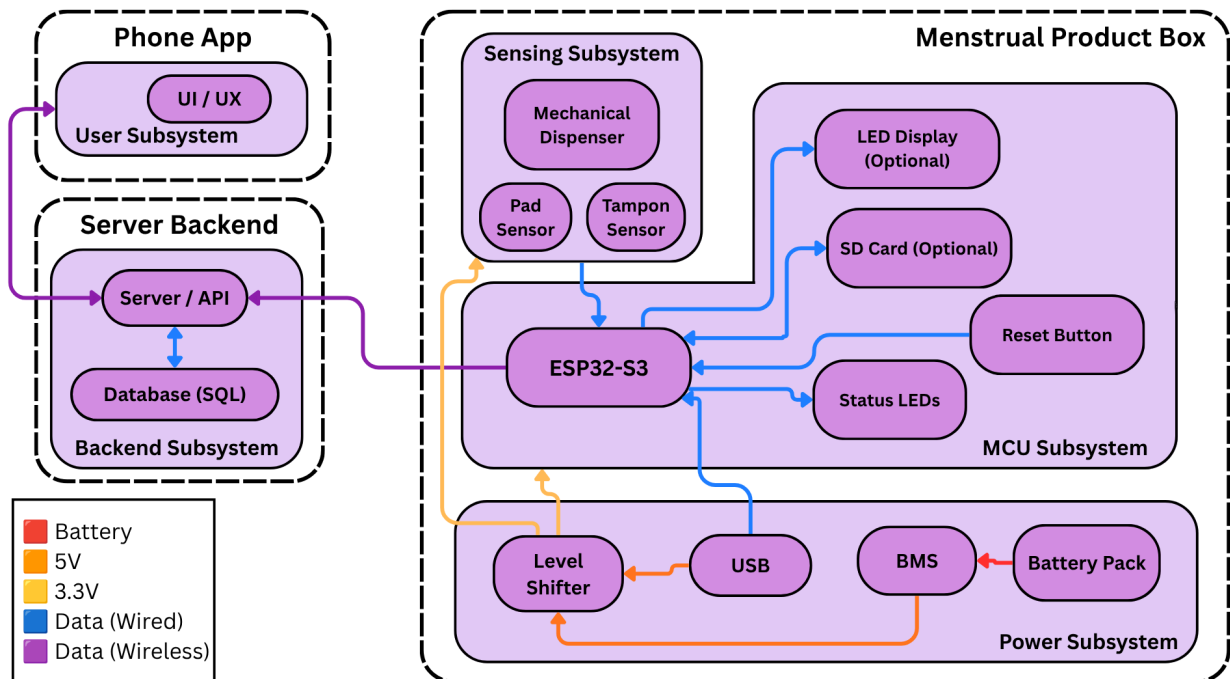


1.4 High-Level Requirements

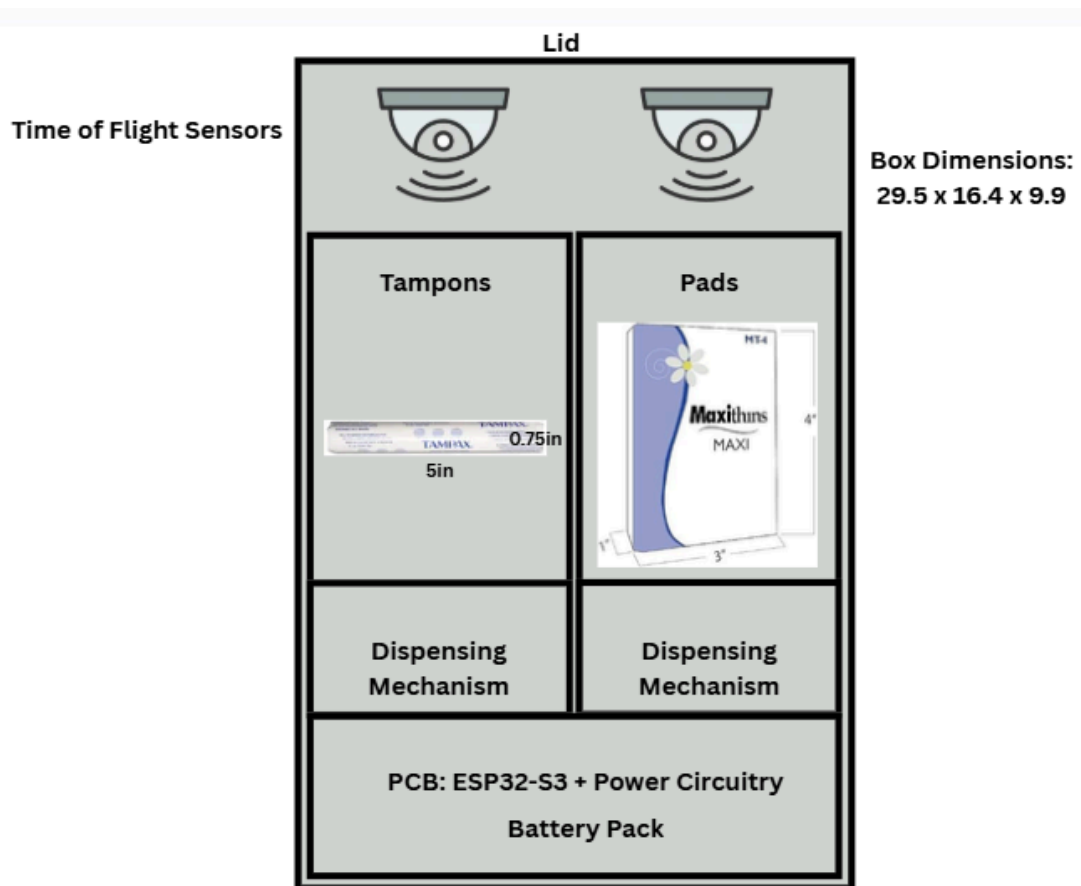
- The final product must display the number of products on the application within 90% accuracy.
- The final product must update inventory data on the phone application within 10 seconds of restock or dispensing.
- The mobile application must determine the user's location within 20-30 m and display the nearest available dispenser.

2. Design

2.1 Block Diagram



2.2 Physical Design



The physical design consists of a 3D-printed menstrual product dispenser containing separate compartments for pads and tampons. A time-of-flight (ToF) sensor is mounted to the inside of the lid directly above each product stack. Each sensor measures the distance between the lid and the top of its corresponding stack, allowing the system to estimate the number of products remaining. Pads and tampons are stacked vertically and are dispensed individually using separate mechanical dispensing mechanisms. The PCB, containing the ESP32-S3 and supporting circuitry, and the battery pack are housed in a separate portion of the enclosure so that users do not have direct access to the electronics. The dimensions of the product compartments are selected to accommodate the pads and tampons while maintaining the ToF sensors within their required sensing range.

2.3 Subsystem Overview

2.3.1 Sensing Subsystem

The sensing subsystem is used to find the height of the stack of menstrual products so that the device can count the number of menstrual pads and tampons. Two separate time-of-flight (ToF) sensors will be attached to the lid of the dispenser, one per stack of menstrual products. The mechanical dispenser is similar to the typical menstrual product dispensers used in university bathrooms. The pads and tampons will be stacked one on top of the other, as per a typical dispenser, to obtain the correct distance from the lid. The products will be able to dispense simply with a click of a button.

2.3.2 MCU Subsystem

The MCU subsystem is used to process the data from the sensing subsystem to be sent to the backend subsystem. Using an ESP32-S3, data will be sent to the server over WiFi. Any GPIOs, such as the status LEDs and the reset button, will be managed by the microcontroller. The data from the ToF sensors is sent over I2C to the MCU.

Optionally, the MCU will be connected to an LED display to indicate the number of available products in the bathroom. An SD card may be utilized for additional memory to store average user data.

2.2.3 Power Subsystem

The power subsystem is used to provide power to the necessary components of other subsystems. A battery pack and BMS is used to provide 5V to the board. The level shifter converts 5V to 3.3V. The USB connector will be used to provide power and update the microcontroller firmware.

2.3.4 Backend Subsystem

The backend subsystem will provide communication between the ESP32-S3 dispenser units and the mobile application. A REST API will be implemented using Python and FastAPI to receive inventory measurements from each dispenser and provide current inventory information to the mobile application. A PostgreSQL database will store dispenser locations, product counts, and timestamps of the most recent measurements.

2.3.5 User Subsystem

The user system consists of the mobile phone app designed in React which displays to the user the number of products available in the dispenser remotely. The app organizes the data for ease of use. The GPS data of the user's phone will be used to determine the nearest dispenser.

2.4 Subsystem Requirements and Verifications

2.4.1 Sensing Subsystem

Requirement	Verification Process
Each time-of-flight sensor must measure a distance of at most 2ft (60.96 cm) with at least 95% accuracy.	<u>Method</u> Using a tampon or a pad as a target, the ToF sensor will be set to measure the target at variable distances. We will measure the distances by hand using a ruler to measure error. The ToF sensor and target will be in the dark since it will be inside of a box when in use. <u>Results</u> Using the built in C function, the ranging data will be sent over I²C. Using PlatformIO, we can obtain the value of these measurements. <u>Presentation</u> All measurement data will be presented in a table. We will have multiple trials per distance, with the

	longest being 2ft. The error will be averaged from each trial.
The box must dispense one menstrual product when the corresponding mechanical button is pressed.	<u>Method</u> Check if one product is dispensed for three cases: 1. Only one product is left 2. 5 products are left 3. Box has been refilled Repeat for 5 trials each. Test for tampons and pads. <u>Results</u> Count the number of products dispensed. <u>Presentation</u> If it dispenses one product at a time, this aspect is true.
Each time-of-flight sensor must update distance within 200ms of restocking or dispensing.	<u>Method</u> Track ToF sensor data after three cases: 1. Dispensing one menstrual product 2. Dispensing several menstrual products 3. Refilling the dispenser We will have multiple trials for each case and each type of menstrual product. <u>Results</u> Using built in C functions, the ranging data will be sent over I²C. Using PlatformIO, we can measure the timing in between the measurements. <u>Presentation</u> Timing will be presented in a table for each case. We will have multiple trials and take the average time for each case.

2.4.2 MCU Subsystem

Requirement	Verification Process
-------------	----------------------

Must process time-of-flight sensor data and send it over Wi-Fi within 5 seconds.	<u>Method</u> We will use much of the same method as with ToF sensor to validate timing data. Additionally, we have included one User-LED for programming on the MCU, as well as a Wifi-LED, these can be used, in part, to depict whether or not different parts of the subsystem are functioning correctly in a visible way. <u>Results</u> Using built in C functions, data should be uploaded from the MCU via USB flash, and sent to our computers where we can investigate the results using PlatformIO. Additionally, we can watch the programmable LEDs for visual cues regarding MCU functionality and have an immediate reaction time. <u>Presentation</u> Timing will be presented in a table for each case. We will have multiple trials and take the average time for each case. LEDs will blink visibly.
Must maintain Wi-Fi connection with the server continuously.	<u>Method</u> We have included one User-LED for programming on the MCU, as well as a Wifi-LED. These can be used, in part, to depict whether or not different parts of the subsystem are functioning correctly in a visible way. <u>Results</u> If there is no or poor WiFi connection, the WiFi LED will not have a good signal and will not light up the way it needs. Thus we can detect this and manually fix the WiFi connection.

2.4.3 Power Subsystem

Requirement	Verification Process
Must provide 5V and 3.3V +/- 0.2V to the system.	<u>Method</u> There are test points on the PCB where 3.3V and 5V can be measured using a multimeter. One lead is connected to the test point and the other is connected to ground. <u>Results</u> The results can be taken from reading the voltage on the multimeter for 3.3V and 5V. <u>Presentation</u> A table will display the voltage and the maximum offset.
Must supply at least 500 mA continuously.	<u>Method</u> After different intervals of time (i.e. 1 hour, 3 hours, 24 hours) measure the output current from the battery. <u>Results</u> Can measure the current output of the battery with a multimeter. <u>Presentation</u> A table will display the current and the minimum current measured.

2.4.4 Backend Subsystem

Requirement	Verification Process
Must store the most recent inventory measurement for each dispenser.	<u>Method</u> Send known inventory measurements from a dispenser to the backend for three cases ... 1. Dispensing one menstrual product

	2. Dispensing several menstrual products 3. Refilling the dispenser After each transmission query the PostgreSQL database and compare the stored product count with the measurement sent by the dispenser. <u>Results</u> Record the transmitted inventory count and the corresponding value stored in the database for each test case. <u>Presentation</u> Present the transmitted and stored inventory counts in a table.
Must update inventory data in the database within 3 seconds of receiving a measurement over Wi-Fi from the dispenser.	<u>Method</u> Record a timestamp when the FastAPI server receives an inventory measurement and another timestamp after the corresponding PostgreSQL database entry has been updated. Calculate the elapsed time between these two timestamps. Repeat the test for ... 1. Dispensing one product 2. Dispensing several products 3. Refilling the dispenser With multiple trials for each case. <u>Results</u> Calculate the database update time for each trial. <u>Presentation</u> Present the update times and average update time in a table.
Must provide dispenser inventory data to the mobile application through REST API.	<u>Method</u> Send a GET request to the FastAPI inventory endpoint for several dispensers with known values stored in the PostgreSQL database. Compare the dispenser ID, product count, location, and most

	recent measurement timestamp returned by the API with the corresponding database entries. <u>Results</u> Record the expected and returned values returned by the REST API. <u>Presentation</u> Present the expected and returned values in a table.
Must associate each dispenser with its corresponding bathroom location.	<u>Method</u> Add multiple dispensers with known bathroom locations to the database. Query each dispenser through the REst API and compare the returned bathroom location with the location assigned to that dispenser in the database. <u>Results</u> Record the dispenser ID, expected bathroom location, and location returned by the API. <u>Presentation</u> Present the results in a table.

2.4.5 User Subsystem

Requirement	Verification Process
Must receive updates from the backend subsystem over Wi-Fi continuously.	<u>Method</u> Open the app and record the displayed inventory count for a test dispenser. Change the dispenser's inventory value in the backend and allow the application to retrieve the updated data. Repeat this test for ... 1. Dispensing one product 2. Dispensing several products 3. Refilling the dispenser

	<u>Results</u> Record the inventory value stored in the backend and the value displayed by the application after each update. <u>Presentation</u> Present the backend and application values in a table.
Must accurately display data retrieved from the backend subsystem within 90% accuracy.	<u>Method</u> Prepare known inventory counts for pads and tampons in the physical menstrual product box and retrieve the report values from the backend using the mobile application. Calculate the display accuracy for each trial as: Accuracy (%) = $(1 - \text{Displayed Count} - \text{Actual Count} / \text{Actual Count}) \times 100\%$ Repeat the test using multiple inventory levels for both pads and tampons. <u>Results</u> Calculate the display accuracy for each tested inventory level. <u>Presentation</u> Present the actual inventory, displayed inventory, and calculated accuracy in a table.

2.5 Tolerance Analysis

Ranging Accuracy: Table 13 of the ToF datasheet [1] gives an accuracy rating for white and gray targets. Assuming a distance of 70cm, a gray target (which has a reflectance level of 17%) will have a guaranteed minimum accuracy of about 7% with a 33ms timing budget. If we assume that this accuracy holds at shorter distances (i.e. the noise grows with distance, so that the noisiest signal is when the sensor is looking into

an empty container), then it is possible that at a height of 30 cm that one reading could have a higher percentage of error.

Error_noise = (0.07 * 30cm) = 21mm which is a height of about 2mm; realistically, this could cause the sensor to detect +- 1 layer of period products in the box.

Given that the ToF sensor takes N readings, however, this accuracy decreases:

$$N = 36: \sigma_{noise} \approx (0.07 * 300mm) / \sqrt{36} = 3.5mm$$

$$N = 64: \sigma_{noise} \approx (0.07 * 300mm) / \sqrt{64} = 2.6mm$$

$$N = 144: \sigma_{noise} \approx (0.07 * 300mm) / \sqrt{144} = 1.75mm$$

Error in scaling: There is a 10cm offset calibration mentioned in the datasheet in tandem with a 3% distance-dependent error. So, for a full stack height where the distance from the ToF sensor is 100mm:

$$\varepsilon_{error} = 0.03 * 100mm + 10cm = 13mm$$

So an error distance of about +- 13mm.

3. Cost and Schedule

3.1 Cost Analysis

Graduate student average salary: \$90,115 (\$43/hour) from Student Success Survey [2]

2.5 X 8 hours/week X 12 weeks = 240 hours

Item	Quantity	Price	Total
Labor Costs			
Labor: Sarah Lau	240 hours	\$43/hour	\$10320
Labor: Erin Rothenbaum	240 hours	\$43/hour	\$10320
Labor: Anna Wilkowski	240 hours	\$43/hour	\$10320
3D Printing Cost			
SUNLU PLA	1	\$15.28	\$15.28

Parts Cost

Description	Manufacturer	Part No.	Quantity	Cost
Capacitors (10nF, 100nF)	Yageo Kemet Gp.	C0805C104K5RA CTU, C0805C103K5RA CTU, C0805C104K5RA CTU	10, 5, 10	\$1.75
Capacitors (4.7uF)	Samsung	CL21A475KBQN NNE	10	\$1.69
Diodes	Lite-On Electronics, INC	LTST-C171GKT	10	\$1.04
USB C Receptacle	GCT	USB4085-GF-A	5	\$3.86

1x04 Connectors	Sullins	PPTC041LFBN-RC	10	\$3.54
Resistors	Yageo, Stackpole Electronics, Vishay	RC0805FR-07330RL, RMCF0805FT10K0, CRCW0805499RFKEA, RC0805JR-075K1L, RMCF0805JT1K10, RC0805FR-071K5L	15, 30, 5, 10, 5, 25	\$3.05
Buttons	Aratas	B3S-1000P	10	\$7.29
Linear Regulator	Diodes Inc	AP2112K-3.3	5	\$0.95
ESP32 MCU WROOM Module	Espressif	ESP32-S3-WROOM-1	5	\$28.30
ToF Sensor	STMicroelectronics	VL53L0CXV0DH1	2	\$9.26
TVS Diode	Würth Elektronik	WE-TVS-82400102	5	\$4.30

Total cost: \$31,040

Total cost of parts: \$80.31

3.2 Schedule

Orange events are on the class calendar!

Due Date	Task	Member
October 2nd	Design Document	Everyone

October 5th	Design Review / Peer Review	Everyone
October 5th	Research / Begin Dispenser CAD Design	Anna / Sarah
October 5th	Research / Begin ESP32-S3 Code for breadboard demo	Anna / Sarah
October 5th	Buy 1st Draft PCB Parts	Erin
October 8th	2nd Round PCB Order	Erin
October 12th	Test 3D print for menstrual box	Anna / Sarah
October 12th	Buy 2nd Round PCB Parts	Erin
October 12th	Start PCB Testing	Erin / Sarah
October 12th - 14th	Breadboard Demo	Everyone
October 15th	3rd Round PCB Order	Erin
October 19th	Buy 3rd Round PCB Parts	Erin
October 19th	Begin backend development	Anna
October 19th	Final PCB updates	Erin / Sarah
October 26th	4th Round PCB Order	Erin
November 2nd-4th	Progress Demo	Everyone
November 2nd	Buy 4th Round PCB Parts	Erin
November 2nd	Menstrual Box Dispenser Print Finished	Anna / Sarah
November 2nd	Finish Backend / start react code	Anna

November 16th	Testing and Debugging	Everyone
November 19th	Mock Demo	Everyone
November 19th - 20th	Mock Presentation	Everyone
November 26th	Final debugging / Fall Break	Everyone
November 30th - December 2nd	Final Demo	Everyone
December 3rd - 4th	Final Presentation	Everyone
December 7th	Best Projects Judging and Awards Ceremony	Everyone
December 9th	Final Papers	Everyone
December 10th	Lab Checkout	Everyone
December 10th	Lab Notebook	Everyone

4. Ethics, Safety, and Societal Impact

4.1 Ethics and Safety

4.1.1 IEEE Code of Ethics I.I: Safety, Health, and Welfare of the Public

In our project, it is paramount that the health and safety of our user base is prioritized [1]. This is a project that is specifically targeted towards individuals in need of health resources. Since this is a product that would be utilized in a public restroom, it is important that user data is kept private and that no individuals are harmed due to the use of our project. All information specific to a user, such as GPS information, will be properly disclosed and optional. No unnecessary or unauthorized information about our user base will be utilized for our design.

For safety, the battery management system is used to monitor the battery such that there is no risk of endangering the health of the user. All important hardware will be encased by the dispenser to prevent tampering and/or damage.

4.1.2 IEEE Code of Ethics II: Fair Treatment of Individuals

The menstrual product bathroom tracker is created to improve access to menstrual products, thus it is paramount that no individuals are discriminated against and are treated fairly when utilizing our project. We do not tolerate the harassment of others or the exclusion of individuals based on their gender, gender identity, disability, or other personal characteristics. The system will be designed to provide equal access to menstrual product availability information for all users.

4.2 Societal Impact and Accessibility

4.2.1 Overall Societal Impact

The menstrual product bathroom tracker is meant to improve the quality of life of its user base by providing an efficient way to locate available menstrual products. The

current dispensers at the university are useful since they are free to use and disability friendly. The project is meant to improve on what is already offered by increasing accessibility and ensuring that menstrual products are available at all times.

4.2.2 ADA Dispenser Compliance

Ideally, our dispenser would be ADA compliant such that it is accessible to all individuals in need of access. All menstrual product dispensers located at the university are already ADA compliant, so it is integral that our product is put to the same standards.

According to 205 and 309 (Operable Parts) [2], a dispenser should be operable with one hand and not require tight grasping, pinching, or twisting of the wrist. The location of the dispenser must also comply with 308.2.1 (Unobstructed Forward Reach) such that the highest accessible reach is 48" maximum from the ground and the lowest accessible reach is 15" minimum from the ground.

4.3 Lab Safety

As said previously, all electronics will be housed in a 3D printed menstrual product box. The device will be powered by batteries and the voltage will be regulated by the battery monitoring system to ensure product longevity and user safety. Users will not have any direct access to the electronics.

References

- [1] "VL53L0CXV9DH/1 Datasheet," ST Microelectronics, 2018. Accessed: Sep. 18, 2026. [Online]. Available: <https://www.st.com/resource/en/datasheet/vl53l0x.pdf>
- [2] University of Illinois Urbana-Champaign, "All Campus Undergraduate 2023-2024 Report," Illini Success. Accessed: Oct. 02, 2026. [Online]. Available: <https://illinisuccess.illinois.edu/23-24-annual-report>
- [3] IEEE, "IEEE Code of Ethics | IEEE," [ieee.org](https://www.ieee.org). Accessed: Sep. 18, 2026. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8>
- [4] "U.S. Access Board - Chapter 6: Toilet Rooms," [Access-board.gov](https://www.access-board.gov). Accessed: Sep. 18, 2026. [Online]. Available: <https://www.access-board.gov/ada/guides/chapter-6-toilet-rooms>