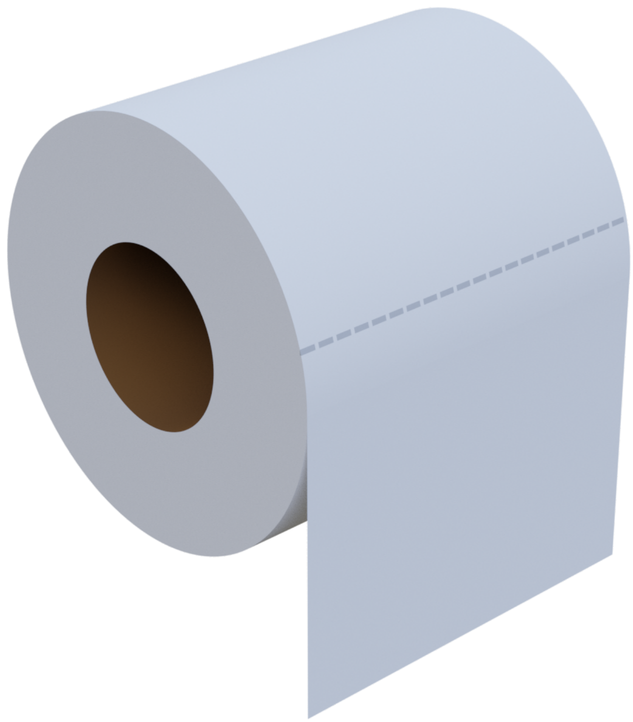
Toilet Paper As-A-Service (TPaaS) Project Design Proposal



"Making the world a better place, one roll at a time"

by

Bryce S. Corbitt
Matthew R. Hagan
Nicholas H. Hollander
Brendan M. Train

Date
April 7, 2020

Contents

Problem Definition

The sudden and rapid spread of COVID-19 has had many unforeseen impacts on our society. Many people are now working from their homes, avoiding public gatherings, and minimizing their number of weekly shopping trips. As a result of this, bulk purchases of various essential items have become increasingly common, causing shortages of said items, which in turn scare people into rushing out to purchase resulting in a vicious and self-sustaining cycle.

Of all common household items, none has been stuck by this fate harder than the humble roll of toilet paper. Fear has swept the world as this once common item has become scarce. In Australia, two women were charged with affray after assaulting another patron over limited supplies [1]. Elsewhere in the world, opportunists are attempting to score a quick buck by acquiring large quantities of bathroom tissue and reselling it online [2].

Given the scarcity of Toilet Paper as a critical resource, now more than ever it is desirable to have control over the quantity of toilet paper being consumed by members of a household. The average American uses approximately 23.6 rolls of toilet paper per year [3], or approximately one roll every other week. With an average of 2.63 people per household [4], that equates to approximately 62.1 rolls of paper per year, or 1.2 rolls per week.

This problem of controlling access to bathroom tissue has been recognized since long before the advent of the so-called ”Internet of Things”. Devices for controlling access to toilet paper and other rolled paper products have existed since as early as 1965 [5] with varying degrees of complexity, with some featuring suites of sensors and electronic controls [6]. Despite existing technology, no device manufacturer has yet connected this technology to the internet, or created a system by which tissue usage is strictly rationed.

Proposed Solution

Our proposed solution to the issue of restricting and controlling access to ones precious toilet paper reserves is an internet connected roll dispenser system. Within the system, a store of up to 6 rolls of toilet paper can be kept at the ready, yet securely contained within the body of the unit. When access to a new roll is desired, the user may utilize a cell phone to request access to the dispenser, which after validating the request, instructs the dispenser to release a single roll of paper, as illustrated in Figure 1.

In order to facilitate the release of the rolls, a system of electromechanical actuators can be utilized to ensure that access to the roll input hatch is only granted to an authorized user, and the ability to retrieve a new roll from the bottom of the device is only possible through authentication and verification via a control application.

Due to the importance of being able to effectively monitor, manage, and control the device even when not in the immediate proximity of the unit, some method of remotely accessing the device must be employed. There are many methods available for long distance remote control however none are as effective as taking advantage of mankind's greatest piece of technological infrastructure; the internet. Connecting the device to the internet not only allows for easy and effective remote management of the device, but also allows for users to easily connect to and request paper from the device.

The final component of a solution is a software application the user can interact with in order to request and obtain access to the toilet paper. Such an application can manifest as either a web server which serves a live interactive application to the user, or as a standalone application installed onto the users device which sends authentication and control signals directly to the toilet paper roll dispenser for verification and execution.

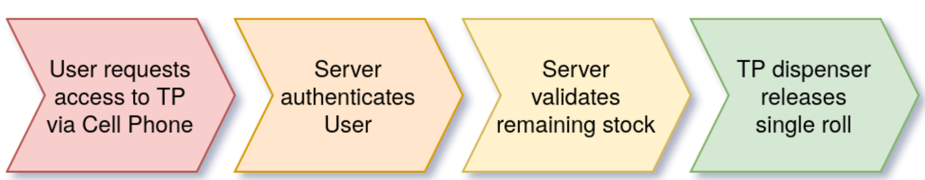


Figure 1: Sample Request And Release Flow

Implementation, Approach, and Tasks

Tasks in the project implementation will be classified from three different sections: Assembly, Electronics, and Software. Assembly tasks pertain to the construction of the physical dispenser. Electronics tasks involve the management, circuitry, and driving behind electronic components in the system. Software tasks include the processing of information, network communication, and interfacing between the system and the user.

Assembly

The primary structure of the dispenser will be a tall rectangular box to store up to six rolls of toilet paper on top of each other. Market Basket branded toilet paper rolls will be used in testing, each with a measured diameter and height of four inches. Using these dimensions for the prototype, the anticipated enclosure will not exceed 5x5x30 inches in size. This enclosure may be made from wood, plastic, or any other material that is sturdy and capable for mounting electrical components. The top and bottom of the enclosure will be open for loading and dispensing rolls. A hinge will be mounted on top of the enclosure to create a latch that may be opened to load rolls. Holes on the back of the enclosure will also be present to mount electrical components (see Figure 2).

Electronics

Due to the timing of the project, availability of purchasing electronic components is limited. Thus, the design for the dispenser (as shown in Figure 2) is comprised only of electronic components currently in possession of the team.

An infrared proximity sensor will be placed on the inside of the lid facing down at the interior of the enclosure. This sensor will be used to determine the capacity of rolls at a given time. The sensor itself has a dead zone of 10 centimeters, and will need to be accounted for during the enclosure's construction. Additionally, a lining of metal wire will be placed around the lid to detect if the lid is open at any given time. Sensing if the lid is open is crucial, as readings from the proximity sensor will be inaccurate while rolls are being loaded.

The dispensing mechanism will be comprised of four 5V servo motors split into two sets. The bottom set will be used to hold the bottom roll in place until it is dispensed. Upon receiving instruction to dispense, a platform between the servos will open downwards, allowing the roll to fall through the bottom of the enclosure. Extensive testing under various loads will be conducted to ensure that this system is implemented reliably. Due to the slow movement of the servo motor, there is evident concern that the platform will be unable to retract in time, causing multiple rolls to dispense at once. To overcome this, a second pair of servo motors will be placed to construct a platform directly above the first set. This platform will be used to hold the remaining rolls in place while the bottom roll is being dispensed. Once the first set is able to retract, the top platform will open downwards, allowing for the remaining rolls to fall into place. To better manage the servos simultaneously, they will be driven via a SunFounder PWM Servo Driver board. The 5V servos draw a considerable amount of current, and will require their own power supply. This power will be provided using the 5V peripheral pins of a PC power supply.

All electronic components in the dispenser will be controlled by a Raspberry Pi computer. While the Raspberry Pi is capable of controlling electronic devices through its GPIO header pins, it also runs its own Linux based operating system, allowing it to easily process and serve data. Additionally, the Raspberry Pi is commonly used for prototyping IoT devices and applications. In 2015, the International Journal of Computer Networks and Applications published an article on the applicability the Pi has to IoT. The Raspberry Pi was tested on local networks for reliable data transfer and secure authentication while serving web content to devices. Communication between multiple devices locally was proven to be reliable for testing purposes, and was especially convenient to interface with electronic components [7].

Both models of the Raspberry Pi computer in the team's possession have on-board Wi-Fi and are suited to wirelessly serve data to users. Due to the fact that the Raspberry Pi does not have a built-in ADC, an ADC circuit will also be constructed for the Raspberry Pi to properly read analog data from the IR proximity sensor.

Software

Due to its simplicity and widespread support on the Rasbperry Pi, driving code for the dispenser will be written with the Python3 programming language. Code will need to be written to read converted data from the IR sensor. To determine the reliability of the proximity sensor data, code must also be written to check the status of the lid. After sensor data under various loads is recorded and analyzed, recordings will be used to abstract future recordings into physical counts for the remaining rolls in the dispenser. Utilizing the SunFounder PWM driver, existing libraries allow for easy control of the servos that power the dispensing mechanism. Dispensing times under a single roll load will be recorded to determine the length of time necessary for the bottom platform to safely retract. Once this is interval is obtained, it will be used as a delay for the top platform to release. Recordings will also be taken under two roll loads to determine the length of time for the top platform may safely retract.

To communicate with users, the Raspberry Pi will host a minimal webserver. This server will be responsible for sending reliable information with regards to the dispenser's status, as well as process user requests and use driver code to control the dispenser from user requests. To ensure the integrity of server code, a REST API will be defined with endpoints to implement. These endpoints will documented carefully, adhering to conventions followed from the "Rest API Design Rulebook" by Mark Massé. In addition to these conventions, Massé covers various platform architectures to which such APIs are implemented in. The webserver implemented will follow Massé's definition for a client-server architecture, to which the server retains full control of resources and is implemented independent of the client [8]. This model is not only widely used in practice, but will also decrease the coupling of software related tasks. While API endpoints have yet to be formally defined, endpoints to retrieve dispenser capacity and request dispensing are currently planned to be implemented.

To complete the aforementioned client-server architecture of the TPaaS system, a client application will be implemented in the form of a mobile app. The client (as shown in Figure 2), will display the current capacity of rolls in the dispenser. Additionally, a button will be present for the user to press in order to dispense a new roll of toilet paper. This app will be built using React Native, a framework developed by Facebook that transforms JavaScript front-end code into native Android and iOS applications. For the purposes of the prototype, only an Android version of the client will be actively maintained. However, a web implementation of the client code may be ported to provide cross-platform capability.

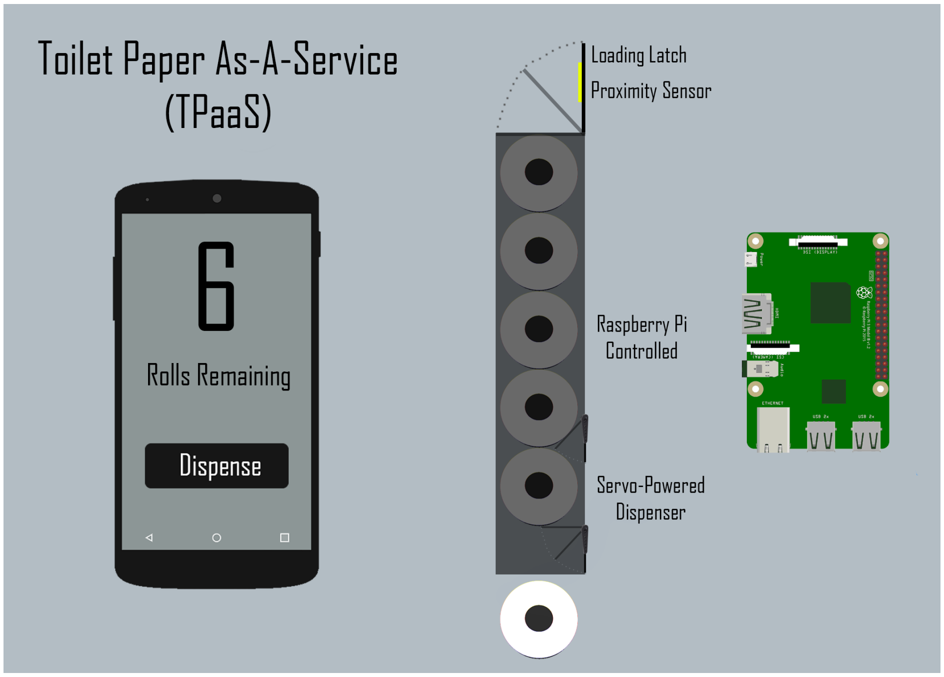


Figure 2: Mockup of TPaaS system and mobile interface.

Evaluation Strategy

To properly evaluate this project's ability to provide toilet paper as a service, we as a team have come up with a few metrics that we would like to use to see how TPaaS performs both qualitatively and quantitatively. These Metrics are outlines as follows.

Reliability

Reliability is an important aspect of this project. Our system needs to be stable enough that it can properly dispense the right amount of toilet paper when demanded without a problem. Beyond that, the reliability and stability of our app and embedded computer architecture need to be solid enough to be able to handle any edge cases that may arise, such as a sensor misread or servo abuse (from a user forcing a servo to move therefore dispensing toilet paper without the app).

To evaluate the reliability of our project, we will be performing a variety of tests within the code to see if the system performs as expected. As our codebase is being written in Python[9], we will be utilizing Tox[10] to test the reliability of our code. Tox is a tool that works in conjunction with PyLint[11] and Flake8[12] to perform unit testing on the code to make sure that it is standardized between developers and will run as expected.

Accuracy

Accuracy is also an important aspect of this project. As the main purpose of this system is to tell the user how many rolls they have in their TP Vault and to dispense a roll when requested, it is important that the system is accurate and tells the user both the correct number of rolls and only dispenses one roll when requested.

as such, we will also be performing real-world edge case testing on the system to ensure that it performs as expected and will work when the unexpected happens. Some examples of ways that we want to test this are by loading partially used Toilet Paper rolls into the system. doing so will ensure that the code based fail-safes are properly inputted into the system.

Scalability

Additionally, Scalability is also an important goal of this project. as such, we want to be able to test how scalable our system is. As such, we will be developing methods to make the system easier to implement for users so that it can be added to other households or better yet commercial spaces such as WPI. To perform this testing, we will be looking into the memory and CPU usage that our programs use on both the mobile device and on the embedded computer. this will allow us to see if our system can work on lower resource systems such as a Particle Photon or a Raspberry Pi Zero.

Ease of Use

Finally, Ease of use is critical to this system. as TPaaS is aimed primarily at residential users, it is important that people of all ages can operate this system. as such, we will be conducting tests upon family members with the household where the prototype exists. this will act as a basic surface level test to see how others respond to using our system. As we cannot responsibly interact with other people outside of our household, we have refrained from including them in our testing infrastructure. If this pandemic does clear up before the end of the term, we will reevaluate accordingly and report back.

Project Logistics, Timeline, and Milestones

The current situation regarding COVID-19 and WPI's decision to move all work off campus, the logistics of developing a project in a group setting have been greatly complicated. To address this, we have implemented an organizational structure which allows us to effectively work as a team despite the physical separation. Primary development, including the construction of the primary prototype, will be undertaken at the residence of Bryce S. Corbitt. The software development process will be a collective process undertaken by all members of the team facilitated by git[1], and hosted on GitHub[2]. Additional prototyping and experimentation for components will be undertaken by Nicholas H. Hollander.

In order to ensure that a fully functional prototype device is ready by the conclusion of the term, we have pre-established a timeline of events outlining our desired progress and areas of focus. This process is outlined in Figure 3, and is broken down as follows.

Week 1: Brainstorming

Before beginning on our journey to implement the worlds next revolutionary IoT device, we first had to come up with an idea. A meeting was held in which we evaluated the current state of the world, and determined areas of focus most suitable for the creation of an internet connected solution. This process yielded several areas of focus related to the current COVID-19 epidemic, including potential systems for enhancing social distancing, controlling and mitigating disease transmission vectors, and management of low-availability critical resources.

[1] https://git-scm.com/
[2] https://github.com/ECE2305-Group-25

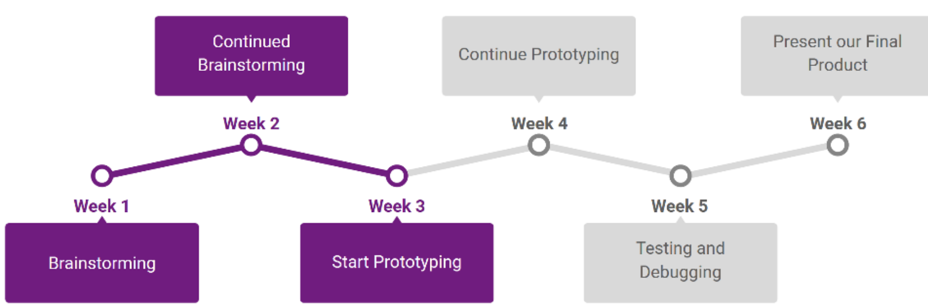


Figure 3: Proposed Design and Development Timeline

Week 2: Continued Brainstorming

Following the selection of our areas of focus, we explored a variety of applied solutions to common issues. As toilet paper shortages were fresh in the news due to wide scale hoardings, we chose to build our solution around the management of this critical resource. Further brainstorming produced the idea of storing rolls of toilet paper in a secure queue to restrict access to only authenticated people via an application installed on or accessed via a users cell phone.

Week 3: Start Prototyping

With our idea fully fleshed out, we are now ready to begin the prototyping phase. Material and component selections are being performed, and an implementable design is being finalized. By the conclusion of this week, the parts and components of our device will be selected, and ready for assembly and wiring.

Week 4: Continue Prototyping

This week will focus on the assembly and completion of the prototype. Due to the limited access to engineering resources, it is probable that a variety of changes will need to be made to the components of the device, which will all be completed during this period.

Week 5: Testing and Debugging

At this point the marriage of software and hardware will take place. Control code, server code, and application code will all be connected to the system, and we will have dispensed our first toilet paper roll successfully.

Week 6: Present our Final Product

At this point Toilet paper As A Service will have been realized, and we will have nothing other than the finest in totalitarian toilet tissue management ready for your judgment.

The other major measurement of our progress and success will be characterized by our achievement of several notable milestones, defined below.

Milestone 1: First roll dispensed (no IoT)

The first major milestone in our project will be the first roll of toilet paper which is successfully released from the secure containment vessel via the built in electromechanical actuation systems. Although at this point there is no server connection, this is still a major achievement.

Milestone 2: Marriage of Technology

The next major milestone will be the marriage of technology, bringing together server code, embedded code, and cellular application code to allow a user's request for a roll of paper to seamlessly travel through the chain of devices down to the dispenser itself. Successful transmission of messages between these points will be our next major achievement.

Milestone 3: Remote release of TP

The next and final major milestone will be the first roll of paper released from the secure containment vessel following an authenticated user request from a mobile device. Representing the completion of our project, this is the most significant of all three milestones.

Bibliography

[1] *'it isn't mad max': Women charged after fight over toilet paper in sydney*, Mar. 2020. [Online]. Available: `https://www.theguardian.com/australia-news/2020/mar/07/it-isnt-mad-max-police-warning-after-shoppers-brawl-over-toilet-paper-in-sydney`.

[2] *Deputies: 8 arrested in california for price-gouging toilet paper, hand sanitizer*, Mar. 2020. [Online]. Available: `https://www.fox10phoenix.com/news/deputies-8-arrested-in-california-for-price-gouging-toilet-paper-hand-sanitizer`.

[3] L. Kaufman, *Mr. whipple left it out: Soft is rough on forests*, Feb. 2009. [Online]. Available: `https://www.nytimes.com/2009/02/26/science/earth/26charmin.html`.

[4] *U.s. census bureau quickfacts: United states*. [Online]. Available: `https://www.census.gov/quickfacts/fact/table/US/HSD310218#HSD310218`.

[5] E. F. Mcgrew, *Electric toilet paper dispenser*, U.S. Patent 3 297 269A, Jan 1967.

[6] K. C. W. Inc., *Electrical roll product dispenser*, U.S. Patent 6 820 785B2, Dec 2001.

[7] J. J. Chea Zhao and S. Loon, "Exploring iot application using raspberry pi," *International Journal of Computer Networks and Applications*, 2015.

[8] M. H. Massé, *REST API Design Rulebook*, S. S. Laurent, Ed. O'Reilly, 2012.

[9] "Welcome to python.org," [Online]. Available: `python.org`.

[10] "Welcome to the tox automation project — tox 3.14.7.dev5 documentation," 2020. [Online]. Available: `https://tox.readthedocs.io/en/latest/`.

[11] "Pylint - code analysis for python | www.pylint.org," [Online]. Available: `https://www.pylint.org/`.

[12] "Flake8: Your tool for style guide enforcement — flake8 3.7.9 documentation," [Online]. Available: `https://flake8.pycqa.org/en/latest/`.