Sustainable Hi-Traffic Toilet Redesign

Authors: Brian Bakerman, James Gualtieri and David Murray: Electrical and Computer Engineering,
Carnegie Mellon University

***Abstract*—The Sustainable-Hi Traffic Toilet Redesign employs improved sensing technologies to decrease water usage and improve maintenance of restrooms. The SHTTR system employs wireless sensors to monitor room occupancy and toilet paper usage to determine how the toilet was used, as well as trigger the toilet to be able flush accordingly. This system provides an improved model on the self-flushing toilet, with proper dual flush significantly saving on water usage. The system monitoring and triggering the wireless sensors also acts to supply analytical data for improved maintenance scheduling and supply monitoring to improve upkeep of public restrooms.**

***Index Terms*—RF, Water Efficiency, Toilet, Waste Reduction**

I. INTRODUCTION

THE Sustainable Hi-Traffic Toilet Redesign, or SHTTR System, worked to improve the modern toilet design. Before the project, we identified two major problems. First, that automatic flushing toilets make no effort to differentiate between a high-flush use and a low-flush use of the toilet. This represents a major gap in worldwide water usage capabilities and waste reduction. Second, it can be understood that different individuals use different hygiene habits in their regular bathroom usage. This can lead to current implementations of the automatic flushing toilet prematurely flushing, as the design only factors in people who stay seated during the duration of their use. Finally, as the population grows, there are more and more bathrooms in use at any given time, and as such the upkeep and maintenance of these is a non-arbitrary concern. In order to solve all of these problems, we reimagine the standard use case situation. To solve the first major problem, the SHTTR System only flushes once a user has exited the bathroom stall. Our solution is one that takes into account how much toilet paper was used during a bathroom use and only flushes after it is clear that the toilet is done being used. Finally, the information contained in the sensor data and toilet usage is logged over the internet to a server which updates a maintenance application in real time, allowing for better analytics of bathroom usage to aid in the maintenance problem. Our goal was to make it so that our project consistently is able to know exactly how much toilet paper was taken on every use and every time flush after the toilet is done being used.

II. DESIGN REQUIREMENTS

In order to satisfy the accuracy necessary to create the SHTTR system, it necessary to constrain the solution. This requires that the toilet paper usage sensor is accurate to realistic toilet usage, the wireless monitoring system is polling at proper intervals, and that door latch sensor provides accurate feedback in bathroom occupancy.

For the toilet paper usage sensor, the intended accuracy of the design is within 5 sheets of paper. As the majority of toilet clogs come from the paper usage, it is important that the system can accurately determine the amount of toilet paper used to the extent that a high volume or low volume flush is triggered. In order to obtain this accuracy, the plan is to use a load cell to measure downwards force applied to the roll of toilet paper. This works in two ways, both to measure the time that a user is pulling downwards to retrieve sheets of paper as well as a rudimentary scale. Using the load cell as a scale, the precision is based on the following equation:

$$Resolution = \frac{Load\ Cell\ Max\ Weight\ (g)}{ADC\ Resolution * ADC\ Gain * Load\ Cell\ Sensitivity\ (mV/V)}$$

$$Resolution = \frac{500g}{2^{16\ bits} * 0.128 * 0.7mV/V} = \pm 0.085g$$

Though it may initially seem counterintuitive to use the Max Weight to calculate the resolution, this is the correct calculation. This is due to the way that load cells function digitally. Essentially, they elicit a resistive response based on weight, and this is split up into a discrete number of “chunks” by the ADC. Thus, the min weight that can be sensed (resolution) is determined by the max weight, as a higher max weight with the same number of “chunks” will yield a larger resolution.

In general, the real-world ADC has a resolution of 24 bits, but however, because of some general noise, it is generally given that only 16 of these bits should be taken as completely accurate for the measurement. This resolution of the scale is well within the defined parameters of +/- 5 sheet accuracy of the toilet paper usage sensor.

The intended accuracy occupancy of the system is reliant on how often data is collected in order to make decisions based on the current state. The goal of the occupancy sensing is to determine within 5 seconds whether or not a single toilet stall is in use. Both the door latch sensor and the toilet flushing trigger are not time cycle dependent and can be read from remotely as the rate of the system polling. The toilet paper usage sensor, once activated, should be updated at a rate of 60 Hz then sending the results to the monitoring system. In order to complete this monitoring reliability, the system should read from the proper tags every 80ms as well as writing to tags when necessary in 80ms. This will allow for accurate occupancy sensing while not wasting power by constantly polling as often as possible.

In addition to these requirements, we also need the system to be reliable for long term use. With some basic calculations with a basic but naïve power usage, the intended battery powered portion of the sensor could operate for 46.92 hours until the battery dies. However, by using a battery free wireless sensor as a wakeup relay, the sensor should not be powered on while the room is not occupied, so there will be no power wasted. Assuming that a single stall is occupied for around 1 hour a day, then the battery life of the toilet paper usage sensor extends to over a month.

Finally, the monitoring system must be able to relay important information in real time to the necessary parties. Using an online database, maintenance schedules will be able to see details such as the amount of toilet paper remaining in specific rolls as well as the amount of times a toilet or bathroom was used.

III. ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

The final architecture for the SHTTR System Consists of several key parts. For a detailed block diagram of the system, please see page [].

The first and most central part in the control hub. In the initial design, the wireless functionality of the hub of the system was to be a ThingMagic M6e-Nano EPC-C1G2 RFID Reader. This reader, in theory, both allows for the powering of the battery free wireless sensors and relays as well as being able to read and write values to and from the same sensors. The reader was to be controlled by a Raspberry Pi 3B+. This would give the system the ability to communicate with a separate server over the internet to post log data from the sensors. The computer will be responsible for logging the information from the reader to a server, as well as processing the information and making the necessary decisions to give commands for the reader's write commands. The M6e-Nano Reader was chosen because of its capability to perform the necessary requirements of the design while still being in the feasible budget range of the class. However, after purchasing and working with the reader, we soon realized we would not be able to use the reader to effectively reach the communication requirements of the project. As such, an alternate solution to wireless communication came in the form of using extreme low power radio frequency communication modules. The NF24L01 chip was chosen for its cost effectiveness and power consumption. The hub is still controlled from a Raspberry Pi, and an Arduino, communication over USB serial. The communication between devices exists in a standard 2.4-2.5 GHz radio band. The chip can receive data on up to 5 channels at once as well as write to another channel. This allows for the hub to read from both a toilet paper sensor and a door latch sensor as well as write to the toilet paper sensor.

Next in the system architecture, there is a separate cell phone application that acts as a way for a human readable monitoring system. In the hub, the Raspberry Pi, reading the information from the Arduino, populates a server with information from the sensors. The application polls from that information, and then updates the application in real time. This will give maintenance workers and schedulers access to the information read from the sensors to inform for proper decision making. This will allow for user alerts when toilet paper is low, as well as a logging information for how often any individual toilet is used.

Next, the Door Latch Sensor is architecturally rather simple. The position of the physical locking of the door latch will be detected with a simple switch, as these are the only two states that the door may be in, necessary to the operation of the SHTTR system. This was originally going to communicate to the RFID reader by a Farsens ROCKY100 wireless sensor. This would have allowed the reader to process the information and determine the state of the system. Instead, in the final system architecture, the communication channel is accomplished using another NF24L01 RF chip, with control being accomplished using an Arduino Nano.

The Flush Valve is similarly relatively simple from an electronic design standpoint. In the initial design, a ROCKY100 sensor would wait for a state to be written from the RFID reader. In the final design, an NF24L01 RF chip is listening for a command for a flush to be sent from the central hub. This then activates a simple LED to represent either a high-volume or low-volume flush from a dual flush valve, which could easily extend to an electronic button press of a multiple volume flushing toilet.

Finally, in the system, there is the toilet paper usage sensor. The initial plan for the sensor was to have it powered on and off remotely from a Farsens Titan Relay. This would have allowed for extremely low power usage of the sensor as it would only be powered on when the bathroom was in use. As the RFID reader was abandoned from the system, as such was this feature in the system architecture. The system is controlled by an Atmega328p microcontroller powered by auxiliary power in the form of a battery. The TAL221 load cell is connected to the microcontroller via the HX711 load cell amplifier which allows the microcontroller to interpret the signal being received from the load cell and process information accordingly. In addition, there is a rotational sensor in the form of a button that is to be triggered with every rotation of the roll. In the original design, through a GPIO interface from the microcontroller, another Farsens ROCKY100 would have communicated the state of the toilet paper sensor to the central RFID reader. Instead, the microcontroller controls another of the NF24L01 RF chips and transmits the readings of to the central hub.

IV. DESIGN TRADE STUDIES

When selecting components for our project, there were two major pieces we had to decide on: Load Cells, and the RFID Reader to use as the basis for the system. Each of these required unique considerations, and the results of these trade studies are described below.

A. Load Cell Selection

When deciding on load cells, we had to consider two major factors: maximum capacity and resolution when coupled with our HX711 load cell amplifier. The options show above are the cells that we initially considered. Our main goal was to have as precise of a resolution as possible. However, we had to rule out all load cells with a capacity under 500g, as they would not be able to handle the weight of the toilet paper. After looking at these options and calculating theoretical resolution when paired with the HX711, it became clear that the TAL221-500 was the best load cell for our design. It combines adequate capacity with ±0.085g resolution, enough to measure with sub 5-sheet accuracy.

TABLE I. LOAD CELL COMPARISONS

Model	***Capacity (g)***	***Sensitivity (mV/V)***	***Resolution (g)***
TAL221-100	100	0.6	0.020
RB-Phi-203	100	0.6	0.020
TAL221-300	300	0.7	0.051
TAL221-500	500	0.7	0.085
Seed 114990100	2000	2	0.119
TAL221-1000	1000	1	0.119
TAL220-3000	3000	1	0.358
TAS606	5000	1.5	0.397

B. RFID Reader Selections

Selecting the RFID Reader was initially a critical part of the design of this project. The RFID Reader sits at the heart of the SHTTR system and working with the UHF spectrum using EPC-C1G2 necessitated looking into high-powered commercial readers. Being on a relatively tight budget, we were fairly constrained, though that didn't stop us from looking at the options on the market. In the end, we decided on the ThingMagic M6e-Nano, mainly due to the fact that its $224 price fit well within our budget, compared to the next cheapest reader at $695. This is not to say that it doesn't fit our use case. For this iteration of the SHTTR system, the range afforded by the M6e-Nano's 27dBm maximum transmit power provides more than adequate range to cover a single stall.

Another major benefit is compatibility with the ThingMagic MercuryAPI, the standard API for interfacing with EPC-C1G2 readers over USB/Serial/Ethernet. If SHTTR were to be expanded to a full bathroom install, either the ThingMagic M6e, or IMPINJ Speedway R420 would be the reader of choice. The full M6e board would allow for four antennas, and with the maximum transmit power of 31.5dBm, would have allowed for four circles of 16ft overage, enough for most single-gender restrooms. For an even more robust installation, the IMPINJ Speedway R420 could be used for its expandability, using up to 32 antennas to cover an entire suite of restrooms, such as a men's and women's room in close proximity.

Additionally, its ethernet connectivity would allow it to be controlled over the network, removing the need for a local Raspberry Pi or similar microcomputer for command management and data logging. Although not used in the final product of the final system design did not use the RFID reader, the effort in selection was non arbitrary.

TABLE II. RFID READER COMPARISONS

Model Name	***Maximum Transmit Power (dBm)***	***Antenna Support***	***Price***	***Interface***
ThingMagic M6e-Nano	27	1	$224	FTDI/Serial
ThingMagic USB Pro	30	1	$695	USB
ThingMagic M6e	31.5	4	$795	UART/USB
ThingMagic Sargas	30	2	$1,190	Ethernet/USB
ThingMagic M6	31.5	4	$2,090	Ethernet/USB/Wifi
IMPINJ Speedway R420	32.5	4 (32 expandable)	$2,365	RS-232/Ethernet/USB

C. RF Module Selection

As the RFID system was no longer feasible, we had to pivot to another method of wireless communication between signals. This led us to decide on low frequency radio communication. Looking into the various options for RF communication, it became clear that the NRF24L01+ was the optimal chip for SHTTR. The major considerations for this part were power consumption and price.

From a power consumption perspective, we needed something with low cost, as we would need six of whatever module we selected. This put the NRF24L01+ firmly in the lead, as it is almost ten times cheaper than the next-cheapest wireless module.

Moving on to power consumption; we needed a low-power device. Most of the sensors, save for the pi itself, would be battery-powered in a production version of the system. So, we wanted to keep the power draw as low as possible. This also puts the NRF24L01+ on top, as it only draws 12mA of current at max read power, compared to 40mA for the next lowest-power module.

To address the shortcoming of the NRF24L01+, we did not need a long-range device, as bathrooms are fairly small, and the max range of 1000 ft from the NRF24L01+ is more than sufficient. This also gave us wiggle room to cut the transmit power and save additional power.

A nice benefit that didn't matter to the initial design is that the NRF24L01+ ended up having the highest max data transmission rates of al of the modules. We didn't need high data rates, but this breathing room was appreciated in the design process.

Overall, pretty much any of these devices could have accomplished the same goals we sought out for, but we chose the best one accordingly.

TABLE III. RF MODULE COMPARISONS

Model Name	***Max Range***	***Power Consumption***	***Data Rate***	***Price/ea***
NRF24l01+	1000 ft	12mA @ 3.3V	RF 2Mbps, Serial 8Mbps	$1.20
RFM69HCW	1600 ft	130mA @ 3.3V	RF 300Kbps, Serial 1Mbps	$11.95
XBee3	4000 ft	40mA @ 3.3V	RF 250Kbps, Serial 1Mbps	$17.95
XBee3 Pro	2 mi	135mA @ 3.3V	RF 250Kbps, Serial 1Mbps	$28.95
XBee Series 1	300 ft	50mA @ 3.3V	RF 250Kbps, Serial 1Mbps	$24.95

D. Cloud Provider Selection

The below tools were all thought about when considering which database to use in our project. The primary consideration was whether the service was offered as cloud-first: setting up a backend is an unnecessary task when so many alternatives exist that are feature rich and robust. Firebase and MongoDB immediately stood out as possibilities. Both provide immediate consistency and concurrency.

However, the power of MongoDB lies mostly in having distributed nodes that can reliably backup the system in the case of a downtime event. The need for this level of robustness was questionable for our application. Firebase is considerably more feature-rich than MongoDB: extending the functionality of our app to include authentication, server-side functions, and more would be quite easy to do. Thus, Firebase was the obvious choice for its ease-of-setup, general robustness, and extensibility. Given the simplicity of the nature of our project, Firebase also gives the ability of us to expand in the future if necessary.

TABLE IV. CLOUD PROVIDER COMPARISONS

Server	***Consistency***	***Concurrency***	***Cloud-first***	***Other server-side features***
Firebase	immediate	yes	yes	yes
MySQL	immediate	yes	no	no
SQLite	immediate	yes	no	no
MongoDB	immediate	yes	yes	no

V. SYSTEM DESCRIPTION

A. Toilet Paper Sensor

The Toilet Paper Sensor involves processing the rotation of and force on the roll for the microcontroller to process and determine the number of sheets of toilet paper used and what flush volume to use.

For the rotational portion of the roll, in order to communicate specific rotation, the system was originally to be using an infrared LED as well as an infrared receiving diode to process a simple physical signal. Inside the cardboard tube of a roll, a reflective surface in the form of a small square of foil is placed in specific position along the side of the roll. The IR LED is constantly sending the exact same magnitude, and the IR receiving diode is waiting to receive a specific amount of the signal back. When the foil is directly under the IR LED, the diode will receive a much large value than usual received when the diode is reflecting against the bare cardboard that is around the rest of the inner circumference of the roll. As the toilet paper is in an enclosed space, the light pollution for the IR sensor is at a relative minimum compared to being present in an open room.

In the final system, the rotational sensing is accomplished using a mechanical switch and a piece of acrylic plastic that acts as a tooth-like gear, striking the switch with each rotation. This eliminates any of the concerns of a more complicated digital system using the IR LEDs.

The load cell exists to act as a force sensor. This accomplishes two goals of the design. Firstly, it acts as a relatively precise digital scale (resolution determined via trade study). This information is necessary to process the amount of toilet paper left on the roll, which in turn helps the system determine how many sheets are used per a rotation of the roll. The load cell also includes a secondary functionality. When a user pulls downwards on the roll to obtain sheets of toilet paper, then a downwards force is applied to the roll. By measuring the amount of time, the force is applied, we can have a secondary sensor in the system to improve upon accuracy. This is useful as a fallback in case the real-world resolution of the load cell system is not sufficient for ~5 sheet accuracy.

The load cell must be connected to a load cell amplifier. The basic functionality of a load cell is communicating force into a variable resistance. The HX711 amplifier runs a voltage through the load cell (a Wheatstone bridge) and is able to amplify that voltage to the microcontroller. The microcontroller then determines the amount of force that correlates with the voltage excited from the load cell based on a calibration factor.

The microcontroller, an Atmega328p, is responsible for controlling the TP sensor and relaying the information to the Farsens ROCKY100 in order to facilitate the originally planned RFID communication with the main reader unit. Instead, the microcontroller corresponds with the RF module to give the data to the hub. The microcontroller takes in the before and after weight of the toilet paper roll, the force applied to it while pulling, as well as the rotation logged and uses that information to decide whether the flush should be Hi or Lo.

Power is a concern in this system, as battery replacements cause inconvenient maintenance overhead. In the initial plan, we had decided to integrate a battery-free RFID triggered relay, the Farsens Titan into the TP sensor to allow it to be powered off when the stall is not occupied. This relay connects a battery to the Atmega328p and, based on our initial calculation a low-power Atmega328p could run on a 1000mAh battery for 76.92 operating hours. Estimating bathroom usage of a single stall of the UC first floor, this would be 20 visits per day, with a 4 min per-visit average time, that equates to roughly 2 months of battery life. This is linearly related to battery capacity, so fitting the stall with a 2000mAh battery would yield a ~4-month battery life, requiring changes only 3 times per year. However, as RFID was removed from the system, the relay system was removed as well.

The final system is simply powered from a wall plug for a standard 9V power supply. It should be trivial to route a power cable to the location of this sensor and eliminating the battery from this component significantly simplifies the maintenance and upkeep of the system as a whole.

B. RFID Reader System

The following was the intended system design of the RFID Reader. The chosen RFID reader uses the ISO standard EPC-C1G2 set of commands. This is the standard for almost all commercial UHF RFID readers, and also the standard that the Farsens tags communicate on. While there are many commercially available readers that are compatible with EPC-C1G2, many of these are far outside the realm of possibility for this project based on cost alone. The M6e-Nano does not have an impressively extensive range in its communication abilities using the built-in antenna, nor support for multiple external antennas. However, it will be capable of providing the intended functionality of the design for a theoretical single-stall installation, with the possibility to attach a single external antenna is need be for range extension purposes. If the design were to be adapted to a full-restroom use case, an EPC-C1G2 reader with support for higher transmit power and multiple antennas would be used, allowing the system to be controlled from a single reader unit with multiple TP sensors, Door Latches, and Flush Valves, as readers on this specification support up to 150 individual EPC tags.

The commands will be given to the reader over FTDI via a serial-to-USB module connected to the USB of the Raspberry Pi 3B+. The reader is able to read a specific tag within 7ms of the given command (150 tags/sec), and able to very quickly relay this information to the Pi. The Raspberry Pi is then responsible for parsing all incoming data from the reader. On board, it is responsible for determining the next state of the SHTTR system. The computer gives the necessary read and write commands to the reader for continued operation as well as logging the relevant data to the Firebase server.

C. Central Control Hub

The final design employs the same Raspberry Pi and Microcontroller to communicate with the rest of the devices, but using an Arduino based "central hub" connected to the PI via usb, rather than an RFID reader. The hub's main function is to be the central processing and control for the System. Using the NRF24l01+ Chip, the hub receives the incoming sensor information door latch. Once the latch is determined that the user is in an occupied state, then the hub is put into "occupied" mode. This begins to keep track of the amount of toilet paper being used over the course of the session. Every time a wiping session" is concluded in the TP sensor, the data is received by the hub, which adds it to a "toilet paper tally". This tally is used to determine if the toilet should be flushed before the user exits the stall in order to prevent premature clogging.

If the user does not reach the "clogging cutoff", then once the hub received a signal that the door latch is open again, it concludes "occupied" mode. This then takes the "toilet paper tally" and makes a decision whether it should be a hi or lo-flush based on the amount of toilet paper used. The hub the sends this information over another channel to the flush valve, which executes the correct flush.

D. Communication Server

The server provides a common interface for the SHTTR system to communicate information. It is responsible for both receiving the current state of the supplies and bathroom usage as well as integrating the ability to relay maintenance analytics of this information. It is integrated using the Firebase Realtime Database framework.

The Raspberry Pi will be responsible for communicating with the server over Wi-Fi interface. The main functionality of the server is to be a constant log for the information being received from the RFID reader. The stall occupancy data, as well as flush frequency/volume, and toilet paper remaining will be logged to the Firebase server for access by the app.

The server also is connected to the companion SHTTR application. The SHTTR app provides an easily readable, formatted version of the information that can act to inform maintenance workers of the proper actions to take in response to the current state of the bathroom.

E. Door Latch Status Sensor

Mechanically, the door latch sensor is relying on a standard contact switch. The switch is in an "off" position when the latch on the bathroom stall door is not locked. When the latch is engaged, the switch makes contact and sends and "on" signal. The latch was originally connected to a ROCKY100 Farsens RFID tag. The information of the state of the latch was to be read by the ROCKY100 and transmitted to the base station when the read command is sent, roughly twice every second to ensure quick detection of occupancy and boot-up of the TP sensor via the Titan Relay.

In the final system, the door latch is an Arduino Nano-based switch sensor that uses the NRF24L01+ to communicate with the central hub. A contact switch is still used to provide door state information.

F. Flush Valve

The flush valve is another relatively straightforward component of the overall sensor system. This component consisted of a ROCKY100 adapted to send a signal to a commercial dual-flush valve for either a Hi or Lo volume flush.

In the final system, a very similar Arduino Nano and NRF24L01+ system is used to represent the flush valve. This then send the hi or lo-flush output to two LEDs, blue for hi and yellow for lo. Due to budget constraints, a real valve will not be used, instead 2 LEDs will illustrate either a Hi or Lo flush.

G. SHTTR App

Using iOS frameworks, the application uses the built-in communication protocol to access information over any connected network (Wi-Fi, cellular data, etc.) Because the server is using the Firebase framework, it only updates the information on the phone as information in the database changes. This way, less information and processing power are waster in constantly polling the entire database. It can also allow for multiple access points of the same information, so in essence, an unlimited number of phones could access the information. The phone is able receive data changing in real time as the server is updated.

VI. PROJECT MANAGEMENT

A. Schedule

Please refer to the Figure 2 at the end of the document for the image of the updated schedule. In general, the schedule has changed throughout some time as some of the general design of the SHTTR system has gone through some revisions.

B. Team Member Responsibilities

Brian was primarily responsible for the SHTTR Application development as well as the usage of the server.

James was primarily responsible for the development and usage of the software on the Raspberry Pi responsible for giving commands to the RFID reader. He also dealt with integrating the "hub" sensor's output to log data to the Firebase server.

David was in charge of the hardware development. This included developing the firmware for the Arduino control units and sensors, as well as designing and prototyping the PCBs, and fabricating the load cell assembly and modified TP.

C. Budget

Refer to table IV at the end of the document for the expense document for this project. Much of our costs were inelastic, as Farsens is the only company making battery free wireless RFID sensors commercially. The rest of the costs were determined via our Design Trade studies.

D. Risk Management

From the design phase, there was some initial risk management in the initial project design. It is important to be ambitious enough with the design to engineer something original and interesting but not too ambitious such that the project is unfeasible. This continued into the realization of the final design as the project was refined. This involved avoiding spending too much time on unnecessary portions to the idea of the project as a system, such as abandoning goals to improve occupancy sensing, which is essentially a solved problem.

Throughout this, there were several phases of design decisions made in an effort to minimize risk. First, instead of trying to create our own battery free wireless sensors, we have decided to use commercially available sensors. This limits the risks that would be inherently present when building complicated systems.

Second, we thought about the skills within our team before proceeding with our project. David largely has experience with high-level design of hardware systems. David was largely the architect of the overall system and understand the commercial and non-commercial components we needed for the project. Further, he does great hands-on work such as fitting sensors. James is a great low-level coder and is largely doing much of the interfacing with our RFIDs using a Raspberry Pi. Brian is handling much of the data logging, as he has good networking databasing, and networking experience. While challenging, each part of the project seemed well within the purview of at least one of our group members.

Third, we chose to research each proposed technology to make sure that it fits our requirements. For example, with our TP load sensor, we made sure that the granularity of the weighing system actually fits our needs and isn't too inaccurate. Another example is with the RFID writing: we are generally worried about the undefined behavior of writing to a switch sensor tag. As such, we are actively investigating alternative solutions for writing before an implementation has unexpected behavior or, worse, renders the tags unusable. This foresight, however, was a risk that did not pan out in our favor. We were unable to use all of the purchased RFID technologies for the extent we had prior intended. This led to a large waste in project funding that could have allowed for more work elsewhere.

Fourth, we have been active about speaking to experts in the particular areas that worried us most. Speaking with Professor Rowe a handful of times with regards to things like occupancy helped us clarify the most effective way to approach that problem and saved us time, money, and headache from picking an inferior solution.

Fifth, we made sure to pick solutions in our project that are cost-effective. We understand that our budget is $600 and picked technologies that both did the job and would not push our budget to the limit. We are doing quite well with this administrative aspect which can haunt a team down the line if handled incorrectly.

VII. RELATED WORK

On the market, there are a lot of automatically flushing toilet valves. The majority of them have a cost tradeoff with accuracy in occupancy sensing. However, none of these products are capable of employing variable volume flushing automatically.

There are many projects out there, however, that are focused on similar goals of waste water conservation and study of toilet usage. Reclaimed water has been an area of study that has grown a lot in the past few years. That is, reclaiming water previously used in sewage into usable and in some cases drinkable water. There are also a number of studies looking into solutions for human waste removal without the use of water.

But overall, there are very few projects that employ a similar idea or spirt as what. is happening in the SHTTR System.

VIII. SUMMARY

Overall, the project should be considered a success. However, there were a few areas in which we did not reach the metrics we sought for.

The system has a full connectivity and relatively inexistence latency in communication between devices.

One of the major shortcomings of the initial design requirements compared to the final product was the failure of the accuracy of the toilet paper sensor. Although as a scale it was able to achieve the desired resolution, the force of constantly pulling on the toilet paper caused the sensor to drift and not reset properly. As such, most of the measurements were determined based on rotational readings. In our public demo, the toilet paper sensor performed as expected in every trial. In a personal study of rotational data, we found that rotation to be accurate in all but 1 of 50 trials, giving us an accuracy of 98%. Throughout all of our efforts in testing the RF reliability, we found that the range of the chip performed as listed, and we only had issues with packet loss when attempting to send transmissions between rooms with brick walls.

Finally, we attempted a basic survey of personal interest in such a system if fully implemented in public bathrooms. We simply asked every who came to try our project at the demonstration if they would be prefer it to the current systems in place. Only 28% of those preferred the SHTTR system always, 8% preferred existing systems, and 64% did not have a preference.

Overall, however, we found the project to reach most of its intended goals, and as such should be considered an overall success.

A. Future work

As far as personal contributions, the team does not plan on focusing on this project in any capacity in the future. As none of us are very related or attached to the toilet industry, and we are all relatively attached to software, the project would require leads much more focused on circuit design and specialist that have a better understanding of the RFID technologies we initially planned to implement.

The system itself, however, is ripe for expansion in the future. The current design of the RF module can allow for up to two toilet systems for every hub RF, using four of the five available channels to receive data, and up to four hubs per one Raspberry Pi computer controlling the systems, using each of the four USB ports on the device. In theory, using abstract control commands and a USB hub, the extensibility of hubs only increases.

The current application has rather limited capabilities, as it stands as a simple real time server display. Extending this, more detailed analytical data could be logged and displayed if a full system of bathrooms was ever to be implemented.

B. Lessons Learned

We learned several important lessons in the course of this project. Firstly, we realized the importance of working with mentors. In the beginning of this project planning, we had reached out to a professor that was a relative specialist in the field we were working. While his initial help gave us many of the original ideas that shaped the design, he was unable to provide us with the necessary support to execute many of them, mainly the struggles we encountered in working with RFID.

Related, we learned the importance of the ability to pivot entire functions of your project. We were forced to switch our functionality from the RFID communication to simple RF communication. This represented a huge shift in our goals for the project and plan of executing the final system. As such, flexibility is important, but this also taught us that perhaps investigating a backup plan in advance could have saved us some stress along the way.

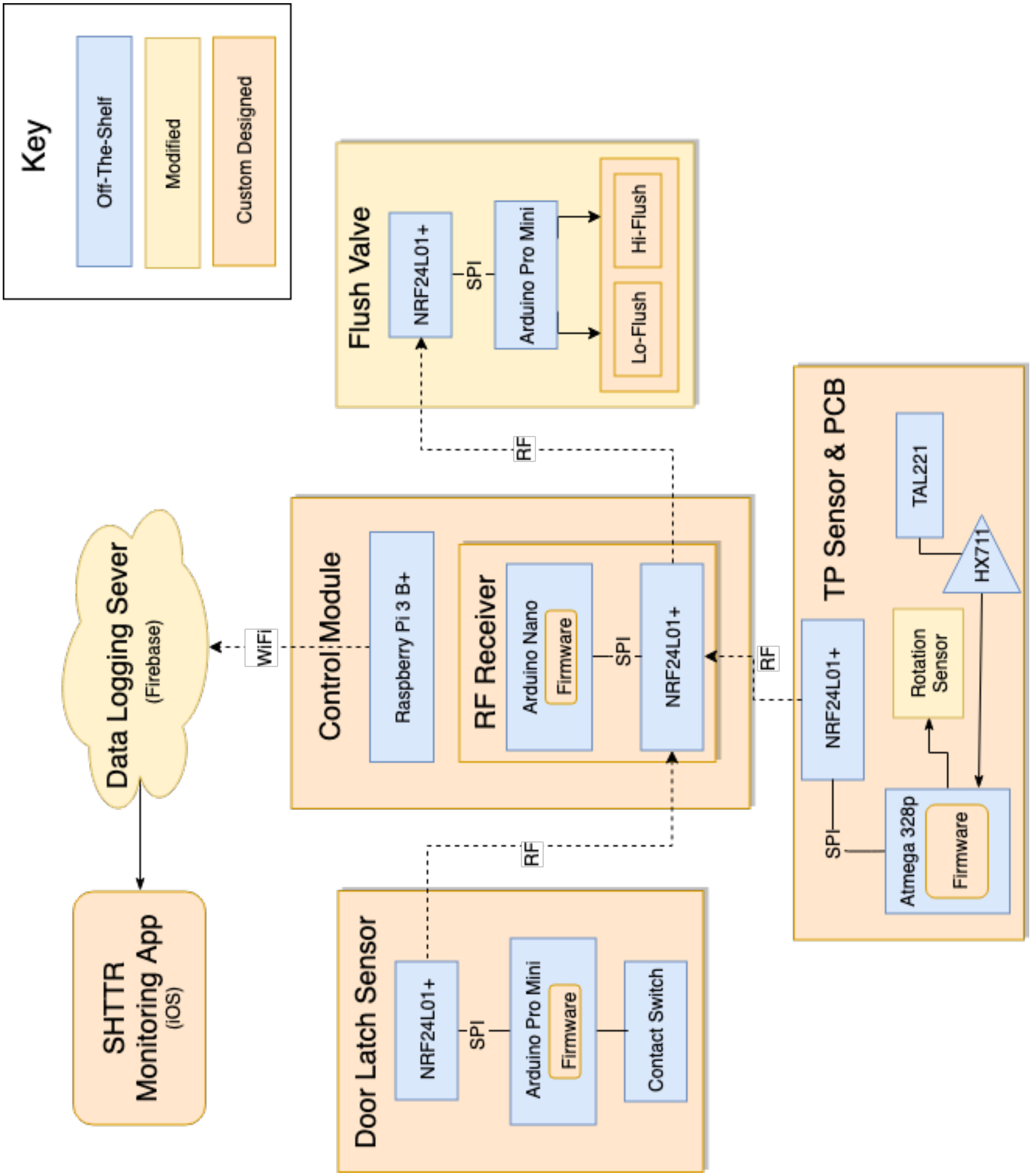

Fig. 1. Block Diagram

Fig. 2. Schedule Gantt Chartt

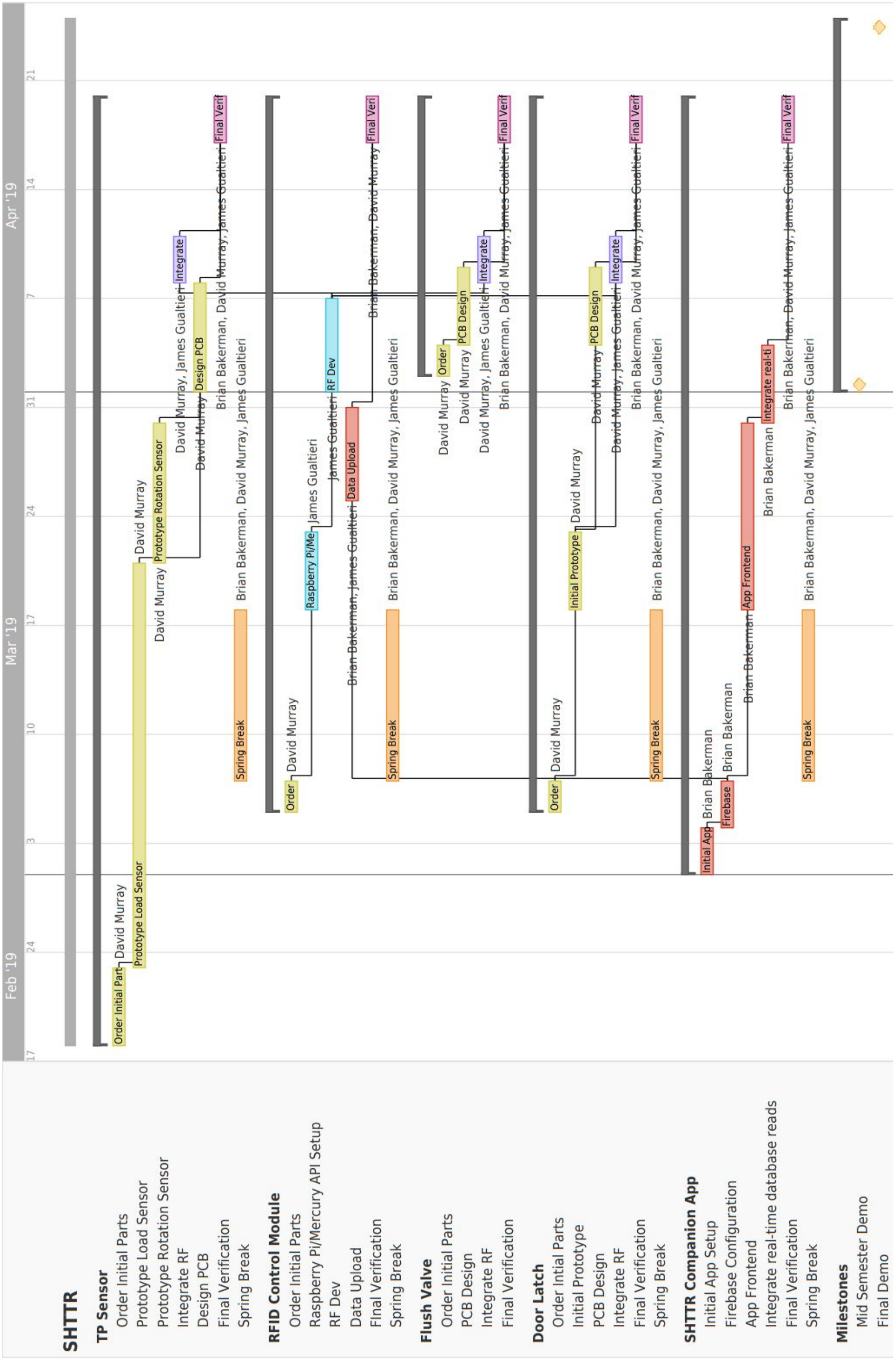

TABLE V. BUDGET

Part	Used	Purpose	Supplier	Qty	Price/ea	Cost
500g Load Cell	Y	Load Cell	SparkFun	2	$9.95	$19.90
HX711	Y	Load Cell Amp	SparkFun	1	$9.95	$9.95
Arduino Pro Mini 5V	Y	Microcontroller	SparkFun	1	$9.95	$9.95
FTDI Breakout	Y	Programming Arduino	SparkFun	1	$14.95	$14.95
Breadboard PSU	N	Powering Arduino	SparkFun	1	$11.95	$11.95
Shipping 1		Shipping	SparkFun	1	$13.28	$13.28
RFID Reader	N	Read/Write tags	SparkFun	1	$224.00	$224.95
Serial to USB	N	communicate with reader	SparkFun	1	$7.95	$7.95
RFID Tags	N	test tags	SparkFun	1	$1.95	$1.95
Shipping 2		Shipping	SparkFun	1	$13.28	$13.28
EVAL01-X5-R	N	Battery-Free Switch Sensor	Farsens	2	€40.00	$89.53
Shipping 3		Shipping	Farsens	1	€20.00	$22.38
Assorted M3 Bolts/Spacers	Y	Mounting	Amazon	1	$11.99	$11.99
Raspberry Pi 3B+	Y	Reader Control	Amazon	1	$38.10	$38.10
Micro SD Card 32gb	Y	raspberry pi os	Amazon	1	$6.99	$6.99
5x NRF24L01	Y	rf communication	Amazon	1	$11.98	$11.98
3 Arduino Nanos	Y	sensor control	Amazon	1	13.99	$13.99
Custom PCBs	Y	Build	JLCPCB	15	$2.00	$30.00
RLC/Diodes/Regulators	Y	Power Regulation	Mouser	1	$29.95	$29.95
Total						**$583.03**