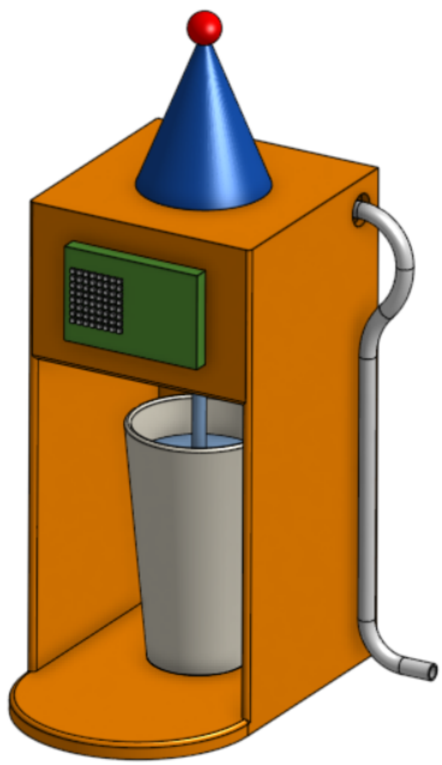
SYSC 3010

Computer Systems Development Project

The Water Buddy



Group L3-G7

Caleb Turcotte, 100929209
Michael Marsland, 101042414
Nicholas Milani, 101075096

TA: Roger Selzler

February 13th, 2022

Table of Contents

1 Introduction

The purpose of this document is to propose a desk based smart hydration system for the SYSC 3010 computer systems development project. The team working on this project consists of three 4th-year Computer Systems Engineering students who share interests in robotics and embedded systems. This proposal will serve to establish a background and motivation for the project idea, as well as an overview of the proposed system's preliminary design and an outline of the team's work plan and schedule.

1.1 Background

Harvard Health states that the average person needs about 4 to 6 cups of water a day to maintain good health [1]. This number can vary based on a multitude of factors including health conditions, body composition, and exercise habits [1]. Environmental factors such as temperature and humidity can also affect the amount of water the human body needs, as warmer, drier environments can cause the body to lose more water due to sweating [2].

Not getting enough water can cause mental and physical fatigue and impact a person's ability to be productive [3]. Reports show that the majority of American adults experience these symptoms of dehydration on a regular basis [3]. From the collected statistics, it is clear that a large portion of the American population have trouble meeting healthy water intake goals in the midst of their busy lives. This could be due to many reasons such as not having easy and convenient access to drinking water while they work, or simply getting distracted by life's challenges and forgetting to fill their water bottle. Standard water coolers are the most common and mainstream solution to this problem, however they are often too bulky to fit conveniently on a desk, and possess little to no smart functionalities to assist in staying hydrated.

1.2 Motivation

Most of the modern ways of collecting water require the user to pour it themselves from a tap or a standard water cooler. With no convenient way to track how much water they are actually consuming each day.

The opportunity exists for a system that can facilitate staying on top of one's hydration levels while also providing easy and convenient access to water during those long, gruelling hours spent at the desk. This will not only promote better overall health, but will also help to eliminate fatigue due to dehydration and boost productivity. This solution specifically targets anyone who is required to spend multiple hours a day at their desk for work or hobbies, however the system can be deployed in the kitchen, living room, or anywhere users spend most of their time.

To take into account the differing needs of every individual, the system can be personalized with body metrics that can be used to calculate the ideal volume of water required. In addition to this, the ambient temperature and humidity of the environment can be taken into account to adjust this level as well. In summary, the solution aims to resolve the vital issue of dehydration and promote a healthy lifestyle by providing convenient access to water, reminding users when it's time to drink, and allowing users to monitor their daily hydration levels and metrics.

1.3 Project Objective

The objective of *The Water Buddy* is to promote a healthy, productive lifestyle by helping users stay on top of their hydration levels. *The Water Buddy* will automatically provide the user with water and remind them to drink based on a dynamic schedule that can be configured and tailored to the user. Finally, *The Water Buddy* will record the user's daily water consumption metrics into a database for easy analysis and comparison.

The Water Buddy will be a lightweight, affordable, and portable station connected to a widespread database that allows for scalability by including as many *Water Buddy* stations as needed for every user of this product. *The Water Buddy* station will draw water it uses from a nearby water jug (the same as those found in common water coolers) but the *Water Buddy* will allow for this just to be positioned on the ground out of the way of the common workstation unlike other desk mounted water coolers.

1.4 Specific Goals

Our main goal is to build a desk-mounted smart water dispenser that tracks a user's water consumption and displays the data on a smartphone application. In order to accomplish this overarching task, there are a number of subgoals that must also be achieved. Firstly, we must build a hardware system capable of autonomously filling a water glass. Secondly, we must program a smartphone application capable of displaying and manipulating data from a database about the user's water intake. Finally, the system will update the user's settings to adjust the behaviour of the water dispensing station. We will also be implementing additional simulated devices to demonstrate the scalability and interactivity of the devices.

2 System Design

2.1 System Overview Diagram

The overall system is designed with a modular approach for scalability. Each Water Buddy is a stand-alone unit that can participate in the system as a whole. At a high level the system contains three main components: the water buddy smartphone application where users can manage their settings and user information, see their data, and associate with their friends, a firebase database to store data and pass messages between devices, and a scalable number of Water Buddys. Each user of the system can own any number of water buddy stations assigned to them but the most common use case will be to have one Water Buddy stationed at your desk. For the sake of the project scope and budget we will only be creating one physical water buddy station and we will be utilizing our other Raspberry Pis as water buddy simulators to demonstrate how the system will function when many users own Water Buddys. Figure 1 below showcases the main nodes of the systems as well as the specific components of the physical water buddy.

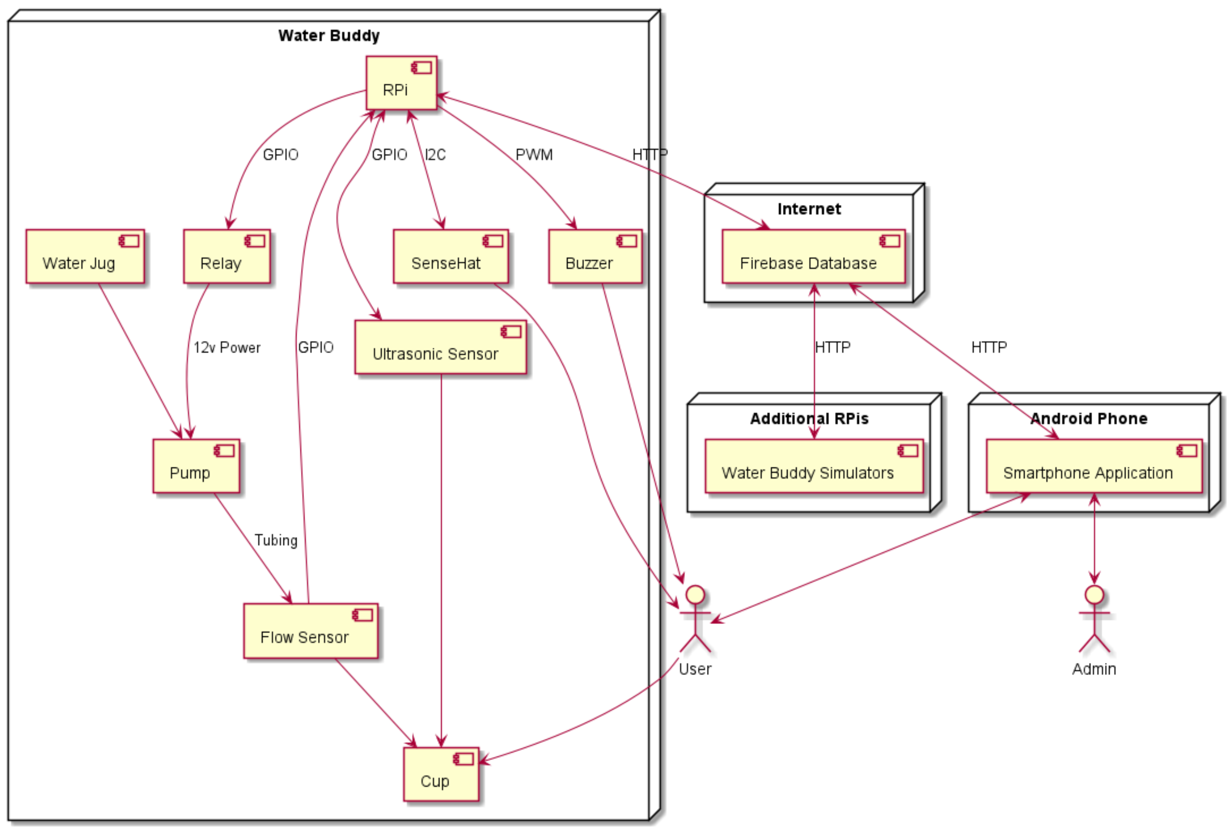


Figure 1: Deployment Diagram of the System

Each Water Buddy consists of a number of hardware components to support its functionality. They will each be built on a Raspberry Pi with a SenseHat for user interaction. The Water Buddy will also have a Piezo buzzer to help get the owners attention when it's time to drink. The unit will have an ultrasonic sensor to detect when a cup is placed in the unit and the relay, pump, and flow sensor will all be required to ensure the cup is filled with the right amount of water. The Water Buddy will draw its water from a store-bought water jug (ss used in common water coolers). This design allows for scalability of the product, where each node can act independently as part of a larger system.

2.1.1 Communication Protocols

The various sensors and actuators will communicate with the RPi through GPIO pins. The ultrasonic sensor will receive a TRIGGER signal from the RPi, and then raise it's ECHO output until the sensor receives the ultrasound echo. The flow rate sensor will continuously output from it's embedded hall effect sensor as a square wave function, which can be used as an input to the RPi's GPIO without conversion. The Piezo buzzer will be controlled through a pulse-width modulation pin on the RPi. Finally, the Sense HAT communicates with the RPi through I^2C.

A Firebase database will be constructed to maintain all data necessary for the system on a cloud-based realtime database. One database will be required for maintaining information about each user's water station. The Pyrebase library will be used in order to interact with this Firebase database from the Water Buddys and the Android Firebase SDK will be used to connect the smartphone application to the database.

2.2 Component Details

As described in 2.1 the system has three main high level components, the firebase database, the smartphone application, and the Water Buddys themselves. The Water Buddys are further broken down into their most important components.

2.2.1 Firebase Database

The first main node in the proposed system is the online database. This database will be used to record all necessary information by the system and will be read and updated by other components. This will include information on each user, their Water Buddy settings, and data on their water intake. The database will also be used to pass messages from the various components of the system, allowing the Water Buddys to take commands from the smartphone application and communicate with other Water Buddys.

2.2.2 Smartphone Application

The second node is the smartphone app. The smartphone application will be used by the owners of Water Buddys for monitoring their own data, receiving push notifications when it's time to drink water, and for interacting with their friends. This app will read data from the firebase database and also contain functionality for admin level data modification as needed.

2.2.3 Water Buddy

The third node in the proposed system is the physical Water Buddy itself. The Water Buddy will be a desk-mounted 3D printed frame which contains all of the necessary hardware. It will be equipped with various sub-components in order to dispense water, communicate with the user, and relay information such as the amount of water dispensed from the water pump and how many glasses of water were used per day to the database. Components used include an ultrasonic sensor, flow sensor, piezo buzzer, sense HAT, and water pump. A wiring diagram of the proposed water buddy system can be seen in Figure 2 below.

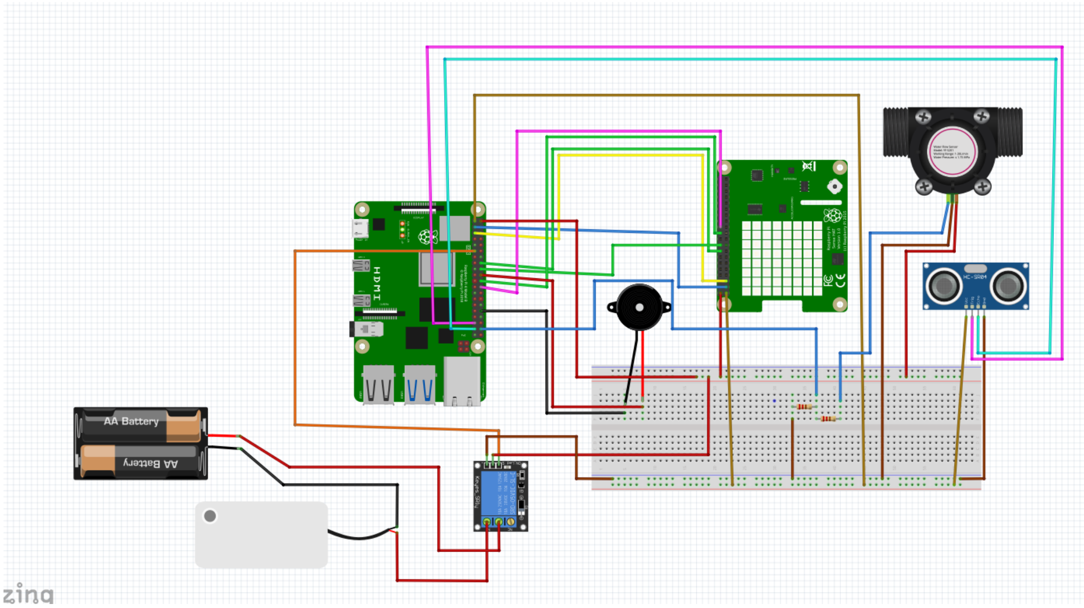


Figure 2: Wiring Diagram of the System

2.2.4 Water Buddy Simulator

The fourth node in the proposed system is the Water Buddy simulator. This simulator will have separate internal nodes that will simulate any missing components in the system in order to demonstrate the scalability of the product. Simulations will include the water being filled by the pump, cup detection from the ultrasonic sensor, and the buzzer notification.

2.3 Use Cases

Described in this section are a few basic ways a user can interact with the system and how the system will respond in these interactions. The use cases below are by no means an exhaustive list but should give a good idea of how the system will behave and how it can be interacted with by a user.

2.3.1 Use Case 1: Filling a Cup

When a user wants to fill their glass of water they will place their empty glass in the unit. The cup sensor (ultrasonic) will detect the cup and notify the RPi. The RPi will then signal the relay to power the pump. The pump will fill the glass, and as this happens, the flow sensor will measure the amount of water being dispensed. When the RPi determines enough water has been dispensed, it will turn off the pump relay and display a full water glass on the SenseHat, accompanied by a jingle from the buzzer. This will alert the user that their cup is full and they will take their water and have a drink! Figure 3 below shows a general sequence diagram of this use case.

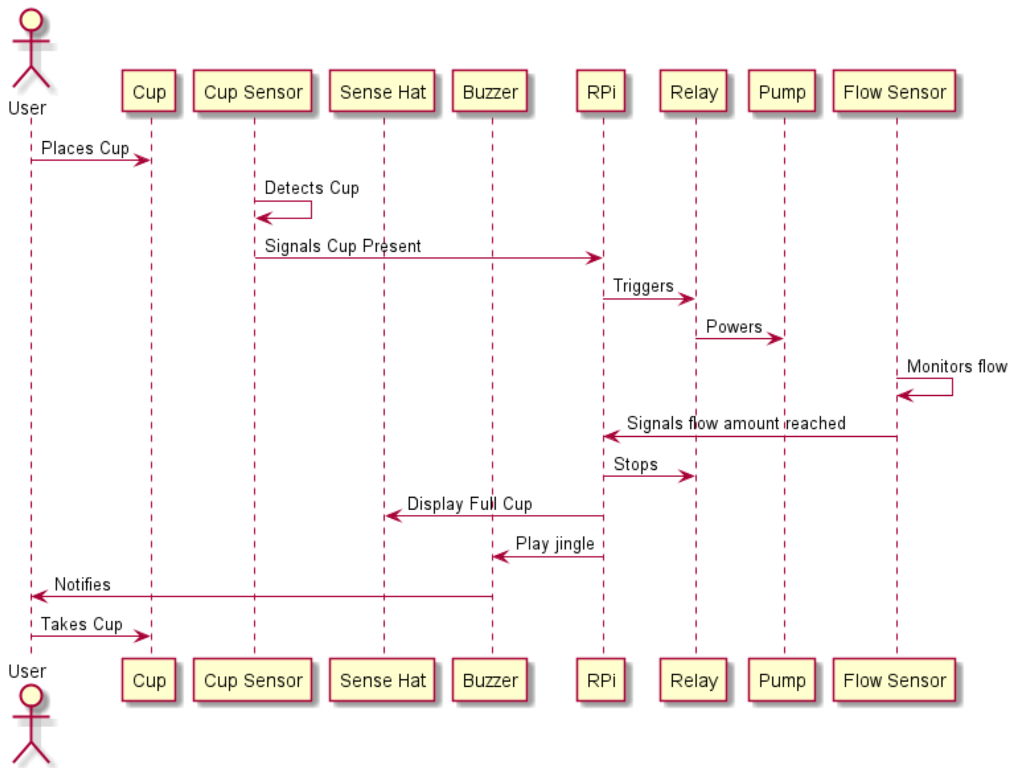


Figure 3: Sequence diagram for Use Case 1: Filling a Cup

2.3.2 Use Case 2: Message passing through Units

After linking units through the smartphone app, each unit can display a notification whenever the other units fill a glass of water. This way you can stay equally hydrated as your friends. When a user fills a glass on one unit it will send notifications to be displayed on the other units so other users know to finish their glass and refill. Figure 4 below shows how the components of the project make this possible.

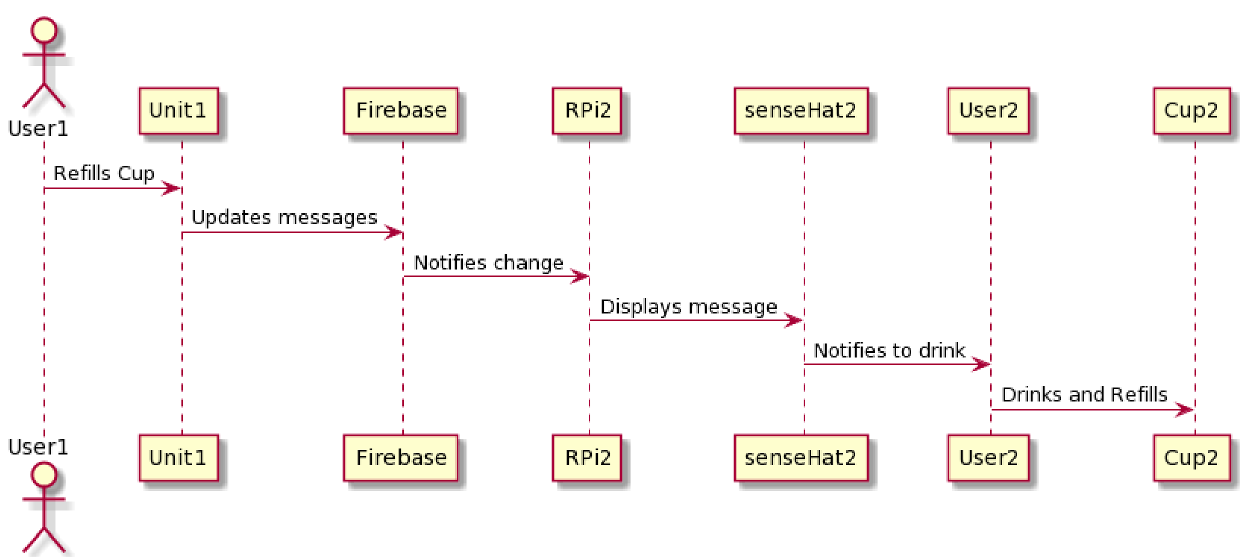


Figure 4: Sequence diagram for Use Case 2: Message passing through Units

2.2.3 Use Case 3: Viewing Data and Adjusting Settings in the Smartphone Application

When a user first gets their Water Buddy they register the Water Buddy with the system through the smartphone application, this allows them to adjust the settings of the water buddy based on their preferences and user information, as well as view their water intake data as recorded by their Water Buddy. The user will log on to the smartphone application, view their data, adjust their settings and the Water Buddy will read this updated data from the database and adjust it's settings accordingly. Figure 5 below shows this interaction in a sequence diagram.

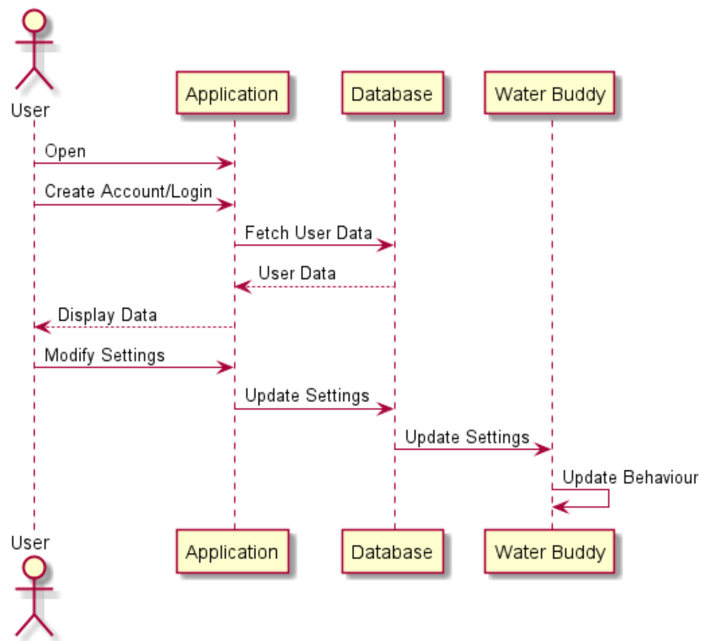


Figure 5: Sequence diagram for Use Case 3: Viewing Data and Adjusting Settings in the Smartphone Application

2.2.4 Use Case 4: Receiving Notifications from Friends

As part of the Water Buddy systems features you can subscribe to receive notifications when your friends drink a glass of water, you can receive these notifications directly on to your Water Buddy or through the smartphone application based on your preferences. The following sequence diagram in Figure 6 shows a friend being notified when their friend refills their glass of water.

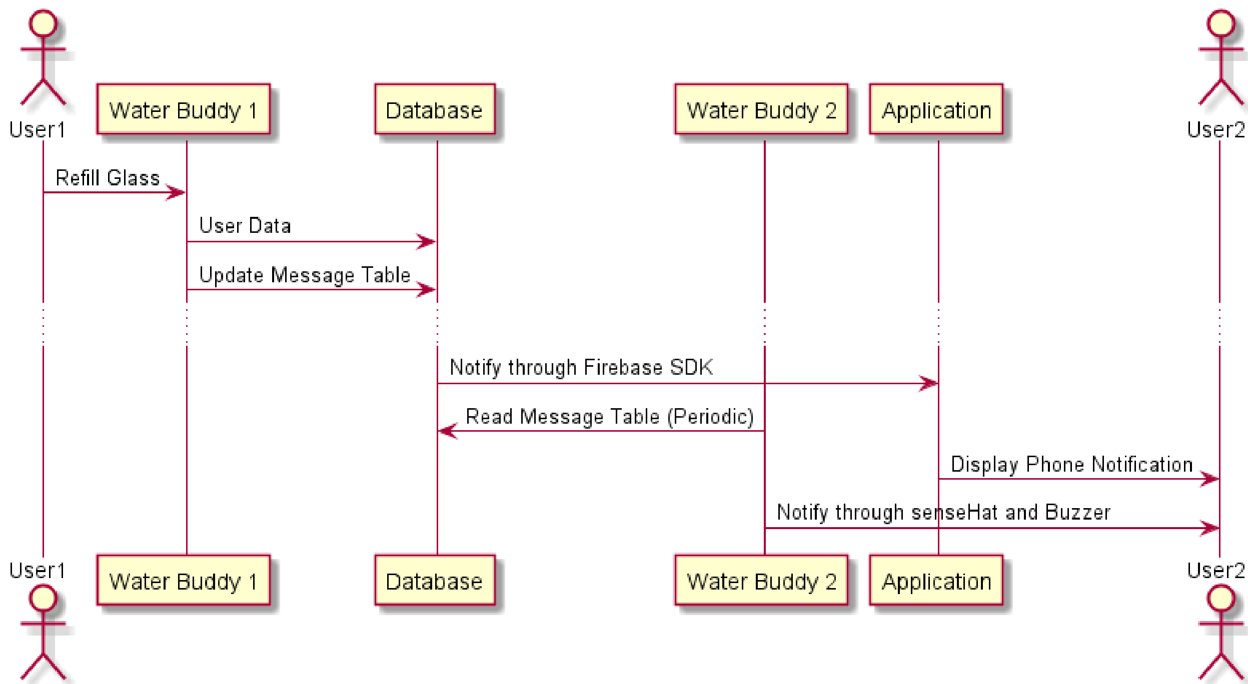


Figure 6: Use Case 4: Receiving Notifications from Friends

3 Work Plan

3.1 The Project Team

Our team is composed of three 4th year Computer Systems Engineering students, each with their own strengths, interests, and experiences to contribute to the project.

Caleb Turcotte has a strong background in embedded software thanks to co-op terms with QNX. He has an interest in low level programming and robotics.

Micahel Marsland has worked on many personal projects using Arduinos and Raspberry Pis. He has developed GUI's using many frameworks as well as applications in Android Studio. He is interested in the hardware and fun features of the project, including building the unit itself and allowing users to communicate with each other.

Nick Milani is interested in electronics and embedded systems. His strengths are focused around electronics interfaces, serial communication protocols such as I^2C, SPI and UART, and embedded C programming. He has had extensive exposure to various hardware components such as microcontrollers, FPGAs, sensors and motors.

3.1.1 Roles and Tasks

We have broken the project down into three distinct components to be developed. The physical Water Buddy, the Smartphone Application, and other Water Buddy Simulators. We have attempted to balance out the workload evenly between all group members and expose them all to new technologies to expand their learning experience while also assigning the components based on our previous experience.

A breakdown of each category of this project as well as the person primarily responsible as well as a secondary for testing is shown in Table 1 below. A detailed breakdown of what is required for each category is listed in the table found in Appendix A of this report.

Category	Primary	Secondary
Administration	All	All
Hardware Development	Nick	Michael
Software (Unit)	Michael	Nick
Construction	Michael	Nick
Software (Simulator)	Michael	Caleb
Smartphone Application	Caleb	Michael

Table 1: Task Breakdown

Caleb will be leading development of the Smartphone application since he has previous experience with Android development. Nick will be in charge of the Water Buddy hardware as he is familiar with wiring and the interfacing of hardware components, and Michael will be managing the Water Buddy software and the concurrent development of the simulators as well as the modeling and printing of the Water Buddy frame since he has experience in software development and 3D modelling.

In general, we will all be working on each component of the project by breaking down each component into a number of smaller tasks, the leader of each component will handle and assign additional tasks as they arise. Appendix A contains a detailed breakdown of all the projected tasks and our current balanced assignment of duties.

3.1.2 Teamwork Strategy

The team will use GitHub as version control to track and report issues with the system throughout development. During project development, issues will be opened on GitHub to indicate problems with the system, while code reviews and pull requests will be used to allow the team to review new additions to the system. All text communication will be done through Slack, while voice meetings outside of lab/class hours will be conducted over Discord, and Lab meetings will be conducted over Zoom during the weekly lab period. Discord meetings will be held outside of class hours when needed, and will be planned through Slack.

3.1.3 What We Will Need to Learn

- App development (Android Studio)
- Computer GUI (Javascript, HTML, CSS, React, tkinter, Electron, or some other gui framework)
- RPi water sensing and control (Hardware programming, GPIO, Relay Control, Electronics building)
- Electronics and circuit design in order to power and control the sensors, buzzer, and water pump
- 3D Modelling (Onshape or other modelling software for printing the unit frame)

3.2 Project Milestones

There are many components to the project that will all be worked on simultaneously. The following milestones are built around each of those components and need not be completed in the order written here. The Gantt Chart in section 3.3 showcases the estimated timeline for the milestones.

3.2.1 Communication through the Firebase Database

Our first milestone will be establishing proper connection to the firebase database from the Water Buddy RPi, the Water Buddy Simulator code, and the Smartphone application. This will require the establishment of a data standard (Schema) for the database and the code needed to utilize the firebase APIs. When we can read and write information and pass messages between all components this milestone will be complete.

3.2.2 Smartphone Application Basic Data

The next major milestone will be continuing development of the smartphone application to display specific information to the user and allow them to enter their personal information as well as adjust the settings for their Water Buddy.

3.2.3 Water Buddy Sensors

The first step in the hardware development will be to connect the Raspberry Pi with the various sensors and write the code loop to poll and communicate to monitor all these sensors first individually and then simultaneously.

3.2.4 Water Buddy Fill System

The next step in the hardware development will be to develop the Water Buddy fill system to respond to cup detection on the ultrasonic sensor, trigger the water pump relay, detect the cup being filled and turn off the relay when the flow sensor has detected that enough water has been added to the glass. This flow should run when triggered by the user for this milestone and later be controlled by the system.

3.2.5 Water Buddy Housing

Once the hardware is connected and working it will be affixed in a 3D printed housing to hold components in place, hide ugly wiring, mount the station on a desk and give the user a distinct place to place their cup and a nice looking addition to their desk. This will complete the hardware set-up of the project.

3.2.6 Water Buddy Software

Along with the Water Buddys hardware development will be the development of the software, this includes controlling the fill system hardware and communicating with the database but also displaying information on the senseHat, alerting the user with the buzzer and overall giving the unit some life. This milestone will be completed when the system runs independently of a user, communicating with the database and controlling it's hardware based on timing loops, settings from the database and user interaction.

3.2.7 Water Buddy Simulators

Along with the development of the software we will be creating simulators to demonstrate the scalability of the project without accruing additional hardware costs. This will mean writing functions to simulate the hardware components both in input from the senseHat joystick and then displaying the state of the simulated hardware on the senseHat display. These simulators will function the same as the regular water buddies but have entirely simulated hardware components.

3.2.8 Smartphone Application Completion

To complete the project the smartphone application will be able to display data from the server based on updates from the Water Buddy unit. The user will be able to manually input glasses of water drank elsewhere, update their personal data, modify their units settings, add and remove friends, and send messages. Admin users will be able to log into the system to modify data as needed and issue community wide events and messages. The user interface for the application will be beautified and the application will be available on each of our smartphone devices.

3.3 Schedule of Activities

The project deliverables, project milestones, and each detailed task (As described in Appendix A) was fit into the projects schedule as a row on the following Gantt Chart shown in Figure 7 and Figure 8. Yellow and purple boxes denote project deliverable and Project milestone due dates and each barren box represents the time that that specific task will be developed.

Figure 7: Project Gantt Chart Part 1

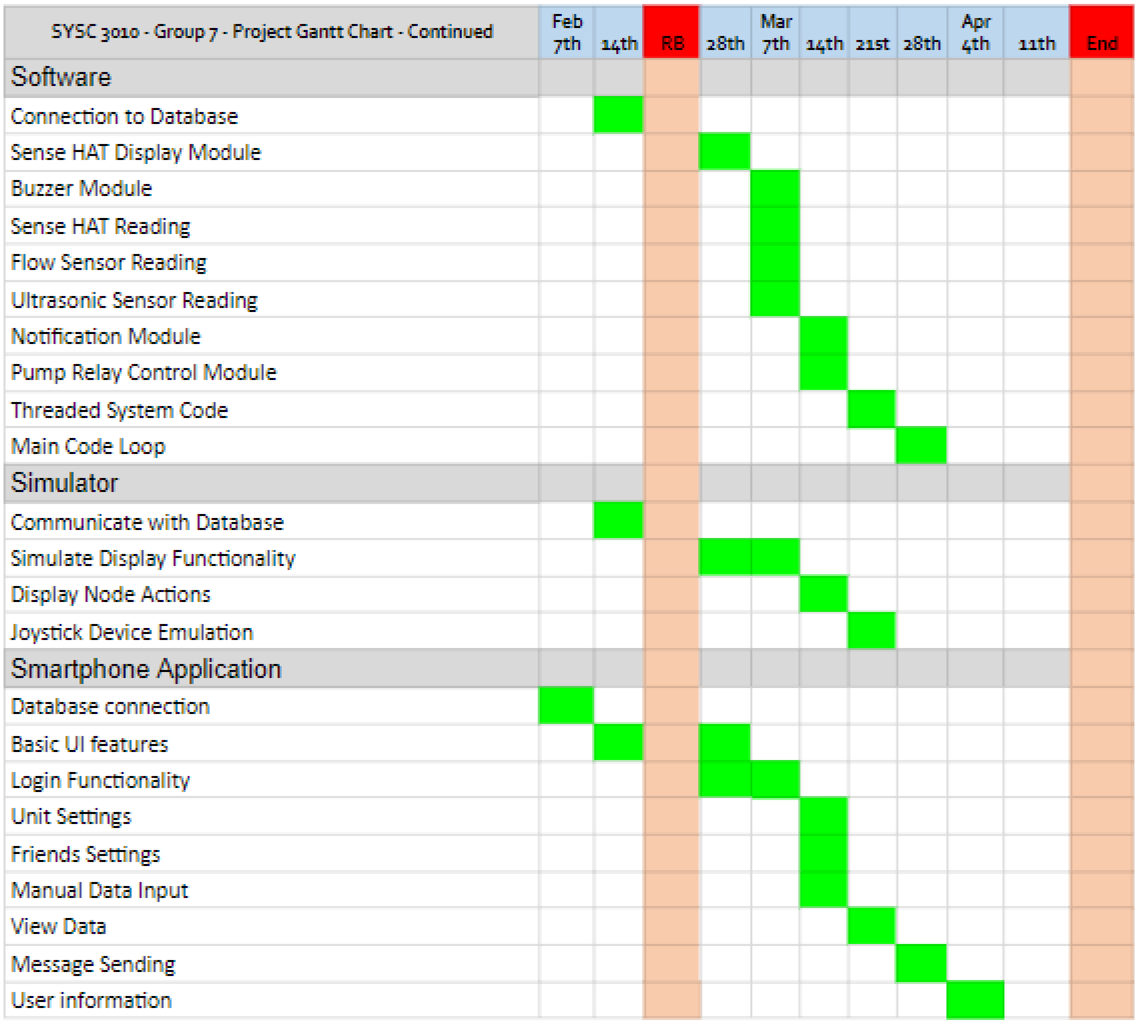


Figure 8: Project Gantt Chart Part 2

4 Project Requirements Checklist

Each requirement for this project is justified in Table 2 below.

Computer Components	Is there a computer (RPi, Arduino, Launchpad, etc.) per student in the group?	RPi for the Unit, Smartphone for the App, Computer for the Admin Panel, RPi's for the simulators
	Is at least one RPi in headless mode?	The unit's RPi will be in headless mode, the simulators may be headless or run on a monitor
Hardware Components	Is there at least one hardware device per student in the group?	Flow Sensor, Relay, Pump, Sense Hat, Pizeo, Ultrasonic Sensor
	Is there an actuator? (i.e., not every hardware device is a sensor)	Relay, Piezo, Sense Hat Display
	Is there a feedback loop? Interaction between input/output devices	Ultrasonic Sensor detects glass, water flows, flow sensor detects limit, water stops
Software Components	Is there a database?	Firebase database
	Does the computer hosting the database have other responsibilities?	The database is hosted in the cloud. Once computer will be able to modify the database and control admin events from a GUI
	Does the database contain at least two tables (or types of information)?	It will contain user information, preferences, data, and pass messages from devices
	Is there a periodic timing loop?	The ultrasonic sensor will always be detecting, (every second or so) and the SenseHat will always be waiting for messages from the server
	Is there some processing or analysis of the IoT data read?	Adjusts timings based on the database settings, displays information from the database on the SenseHat
	Are notifications sent (SMS, email, push notifications)?	Push notifications (or SMS) from the smartphone app, email or sms notifications to the admin from the admin panel.
	Is there at least one bidirectional GUI running on a computer?	The smartphone application will be able to control your unit, the admin panel will be able to control the database, the smartphone application will display information from the server from the unit.

Table 2: Requirements Checklist

5 Additional Hardware Required

Table 3 lists a budget table below. This table lists each part used as well as the DigiKey number associated with that part. A maximum budget of 50 dollars was allocated to this project.

Item	DigiKey Part No.	Cost
Ultrasonic distance sensor	1528-2832-ND	$5.76
Flow Sensor	1597-1518-ND	$8.60
Water pump	1738-1398-ND	$13.41
Water pump relay	1738-1077-ND	$3.56
Water pump power supply	189-QFWB-20-12-US01-ND	$13.44
Power supply Jack Adapter	1528-1386-ND	$2.74
Piezzo buzzer	2104-TP224003-2-ND	$1.10
Breadboard	1528-2236-ND	$1.09
Total		$49.69

Table 3: Additional Hardware

6 References

[1] “How much water should you drink?,” *Harvard Health*, 25-Mar-2020. [Online]. Available: https://www.health.harvard.edu/staying-healthy/how-much-water-should-you-drink. [Accessed: 02-Feb-2022].

[2] “Environmental Issues That Influence Intake of Replacement Beverages,” *National Center for Biotechnology Information,* 1-Jan-1994. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK231133/. [Accessed: 06-Feb-2022]

[3] “Adult Dehydration,” *National Center for Biotechnology Information,* 10-Oct-2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK555956/. [Accessed: 06-Feb-2022]

Appendices

Appendix A: Detailed Task Breakdown

Category	Task	Description
Administration	Proposal Document	Write the proposal
Hardware Development	Determine Parts	Research components needed for system functionality
	Fill out order form	Complete Order Form within budget scope
	Pump and Relay Setup	Configure the pump + relay circuit
	Flow Sensor Wiring	Configure the flow sensor circuit
	Ultrasonic Sensor Wiring	Configure the ultrasonic sensor circuit
	Buzzer Setup	Configure the Piezo buzzer circuit
	System Setup	Integrate all hardware components into the system
Software (Unit)	Connection to Database	Establish connection with Firebase Realtime Database
	Sense HAT Display Module	Develop software to autonomously control the Sense HAT display
	Buzzer Module	Develop software to autonomously control the Piezo Buzzer
	Notification Module	Develop the event-driven notification system
	Sense HAT Reading	Develop software to read temperature and humidity data from the Sense HAT
	Flow Sensor Reading	Develop software to read and manipulate data from the flow sensor
	Ultrasonic Sensor Reading	Develop control loop to trigger and read from the ultrasonic sensor
	Pump Relay Control Module	Develop software to control the relay attached to the water pump
	Threaded System Code	Deploy control modules on concurrent threads
	Main Code Loop	Develop the main system control loop
Construction	3D Model Frame	Create concept model for station frame

	3D Print Frame	Print frame for assembly
	Create bindings and holdings for parts	Add additional features to the housing to hold components in place as needed
	Assemble Hardware into Frame	Place the full hardware set-up into the frame to complete the unit
Software (Simulator)	Simulate Display Functionality	Display on senseHat like a real unit
	Communicate with Database	Establish real time communication to simulate multiple stations acting at once
	Joystick Device Emulation	The joystick can be used to simulate events such as cup placed, etc…
	Display Node Actions	Internal events can be displayed on the senseHat to show how the device's nodes would be working.
Smartphone Application	Database connection	Smartphone Application for viewing user statistics from the database and interacting with data in real time
	Basic UI features	Ability to page between screens
	Unit Settings	Manually set how much water you want to drink
	Friends Settings	Add friends and determine who can send messages to your units, determine if your unit should be notified when others drink
	Manual Data Input	Enter glasses of water drank elsewhere into the database to accurately track your hydration
	View Data	View how much water you've drank today and in previous days
	Message Sending	Be able to send messages to units (Own and enabled Friends
	Login Functionality	Ability to login as a user with authentication
	User information	Manage your user information, so the unit can help determine how much water you need

Table 4: Detailed Task Breakdown