Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Spring 2021

The Brews
Bluetooth Hydrometer

Jesus Adrian Guerra
Jorge Avila
Calvin Mata
Doungpakanh Keomaxay-Hampf

Revision History

Revision	Date	Author(s)	Description
0.1	4.1.2021	JAG	document creation
0.2	4.11.2021	JAG,DPKH,JA,CRM	complete draft
0.3	5.01.2021	DPKH	V2
0.4	8.04.2021	JAG	Final draft with updated logo

Contents

List of Figures

List of Tables

1 Introduction

The Bluetooth hydrometer is a system that allows remote monitoring of temperature and specific gravity of beer during the fermentation process. Home brewers, the intended audience, will be able to check both temperature and specific gravity(SG) at any point during the brewing process by using their smart phones running an app or a web server on a Raspberry Pi. The Bluetooth hydrometer can be sanitized and placed into a fermentation vessel. This would allow a home brewer to take the specific gravity and temperature of their brew without having to open their fermenter to take a sample which could potentially expose their beer to contamination. The device would send the temperature and SG data to the brewer's phone running an app or a web server. By using this data the home brewer would then be able to monitor the fermentation process enabling them to know when they should change the temperature of their brew. Key requirements of the Bluetooth hydrometer include being able to measure the temperature and specific gravity of the brew throughout the entire brewing process, plotting specific gravity and temperature measurements, needing a website on a Raspberry PI or a phone app to poll the hydrometer to receive and display the current brew temperature or specific gravity, needing a website on a Raspberry PI or a phone app to log historical temperature data or specific gravity, being able to poll the hydrometer to receive the current temperature or specific gravity of the brew, and displaying the current specific gravity or temperature data.

2 SYSTEM OVERVIEW

The Architectural Design Specification of the Bluetooth Hydrometer project will have 3 layers. These layers include the Hydrometer, Data Acquisition, and Data Display. The hydrometer will be submerged in a brew tank and will communicate with the other layers using Bluetooth.

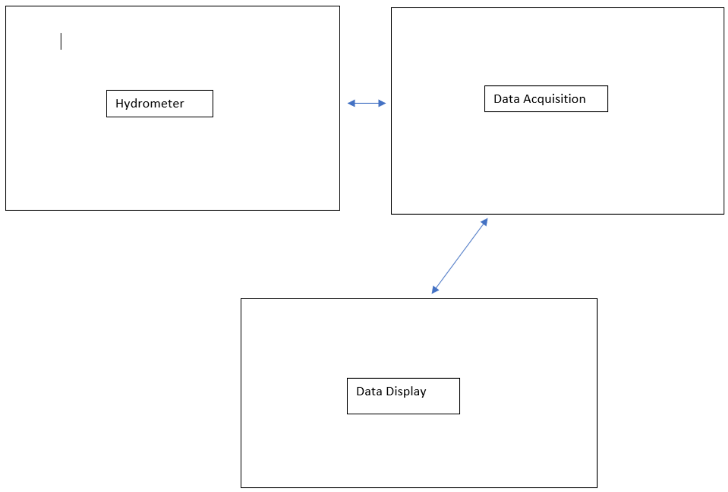

Figure 1: A simple architectural layer diagram

2.1 HYDROMETER DESCRIPTION

The Bluetooth Hydrometer is a device that receives specific gravity and temperature data from inside a brewing keg during the fermentation process. The IMU will be used to measure the specific gravity data to be sent to the microcontroller. The temperature sensor will send temperature data to the microcontroller and the on board BLE will be used for wireless communication between the hydrometer and the Raspberry Pi 4 on board Bluetooth module. The Hydrometer will be powered by a battery that will be inside a watertight container also with the hydrometer. The hydrometer will send the specific gravity and temperature data to a Raspberry Pi by using the on board Bluetooth. The subsystems within the hydrometer layer are named after the individual components that the hydrometer is composed of.

2.2 DATA ACQUISITION DESCRIPTION

The Data Acquisition subsystem is the layer that will define the communication between the Bluetooth Hydrometer and the app. The computer's Bluetooth module will receive data from the hydrometer. The computer will control the flow of data and execute calculations on data.The database will hold a history of data to be accessed at will. The computer will be plugged into a power source and be responsible for computing with incoming data. The subsystems in the Data Acquisition layer are called the communications sub layer for the Bluetooth module, the computation module sub layer for the computer, and History Logging/ Logger sub layer for the database.

2.3 DATA DISPLAY DESCRIPTION

The Data Display is the layer that will define how things are displayed from the Logger/Database back onto the web page using web services and tools such as API's that will quickly show data to the user. This layer is the most abstract to the user that aids in hiding the back end nuisance.

3 Subsystem Definitions & Data Flow

This section graphically represents all the logical subsystems that compose each of our layers. Below is a data flow diagram that shows the interactions/interfaces between all our subsystems. The section after will describe each subsection with greater detail.

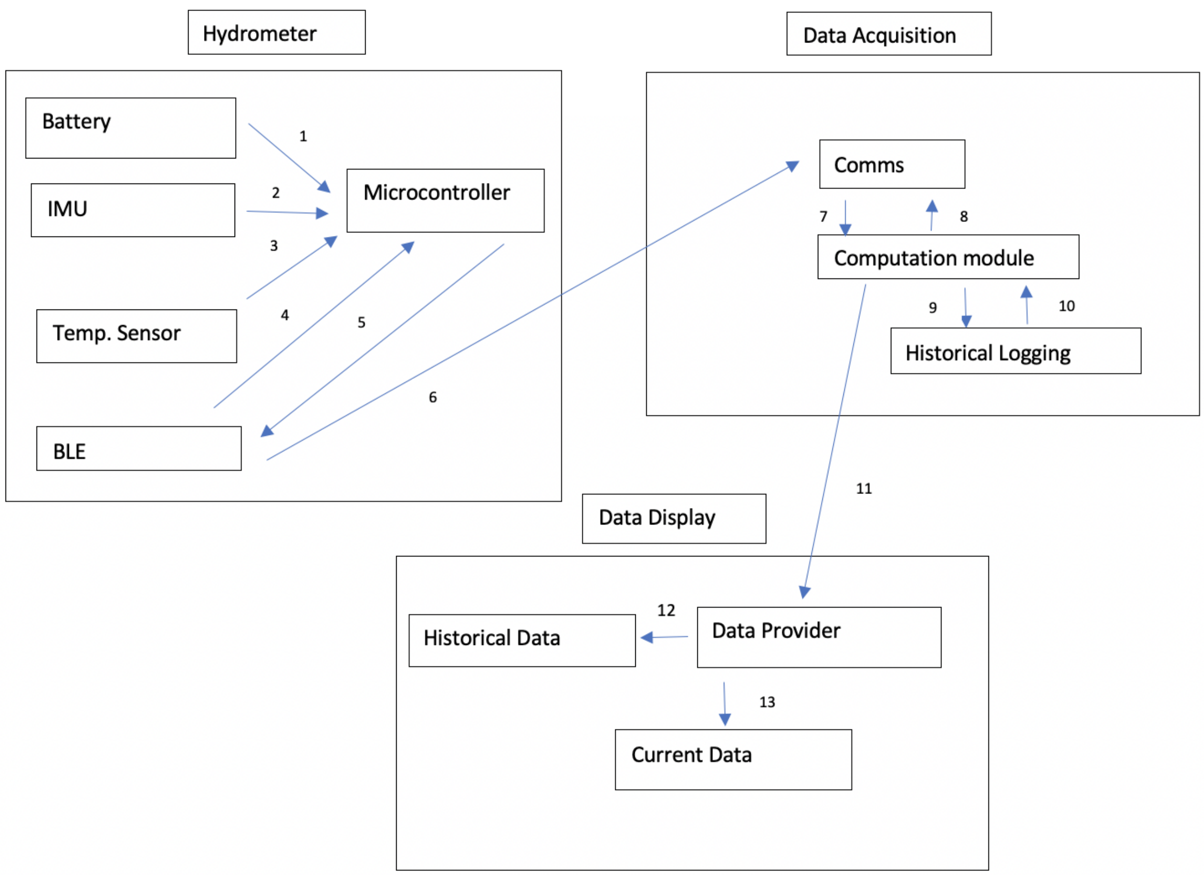

Figure 2: A simple data flow diagram

4 Hydrometer Layer Subsystems

The hydrometer is the main feature of our project and gathers specific gravity as well as temperature data from inside a brewing apparatus during the fermentation process. The device will send this data to the outside by using Bluetooth communication. The components that make up the device include a battery, microcontroller, IMU, temperature sensor, as well as the on board BLE(Bluetooth Low Energy) module. All these components are contained within a waterproof container.

4.1 Battery

The battery is the power source for the hardware that makes up the hydrometer. The only output is the current sent to the microcontroller in order to power it.

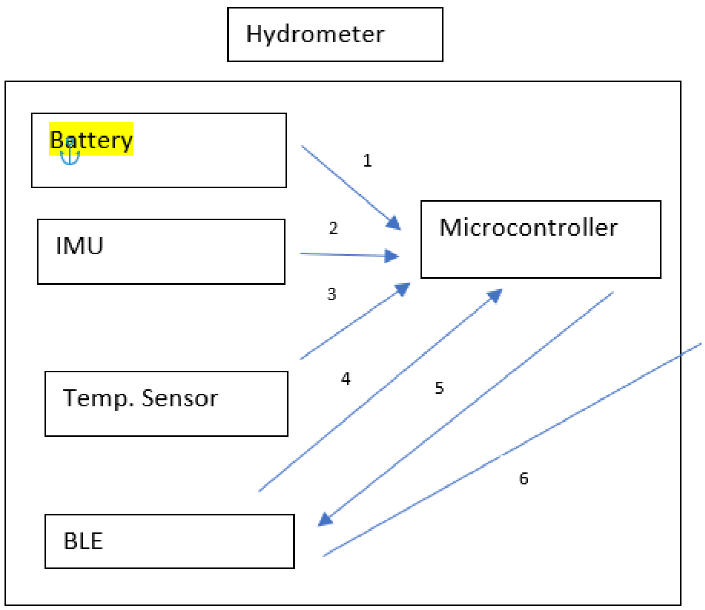

Figure 3: Battery Subsystem

4.1.1 Assumptions

It is assumed that the battery will be small enough to fit within the confines of the container while also being able to power the hardware for the duration of the fermentation process. The fermentation process can last anywhere from a few days to a few weeks.

4.1.2 Responsibilities

The battery has the sole responsibility of powering our hardware within the container.

4.1.3 Battery Interfaces

Table 2: Battery interfaces

ID	Description	Inputs	Outputs
#1	Battery	N/A	Current

4.2 IMU

The Inertial Measurement Unit(IMU) is what measures the tilt of the hydrometer while inside the brew container during the fermentation process. It outputs voltage levels that can be read by the microcontroller through the use of analog to digital converters.

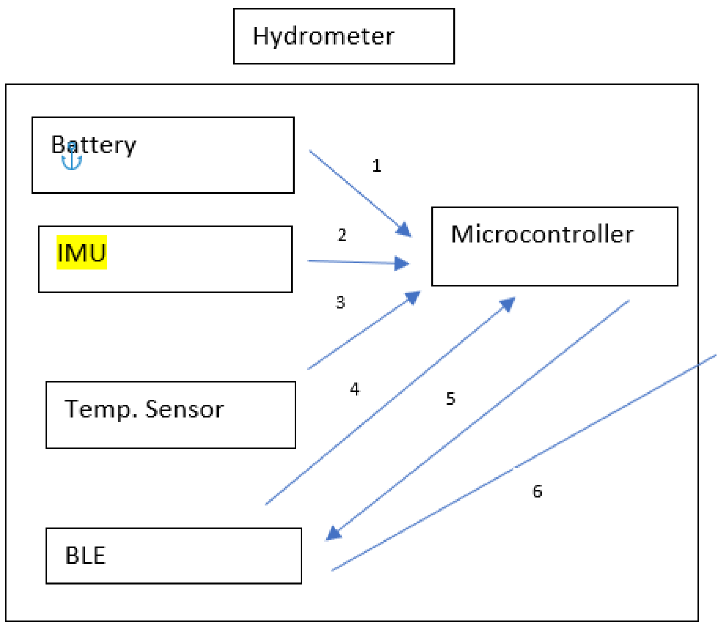

Figure 4: IMU subsystem

4.2.1 Assumptions

It is assumed that the IMU will be compatible with the microcontroller we choose. It is also assumed that the readings we get from the IMU are accurate and not affected by drift.

4.2.2 Responsibilities

The only responsibility the IMU has is to provide accurate measurements of tilt of the hydrometer during fermentation.

4.2.3 IMU Interfaces

Table 3: IMU interfaces

ID	Description	Inputs	Outputs
#2	IMU	N/A	Tilt Measurements

4.3 Temperature Sensor

The temperature sensor is what measures the temperature of the brew by outputting analog voltage.

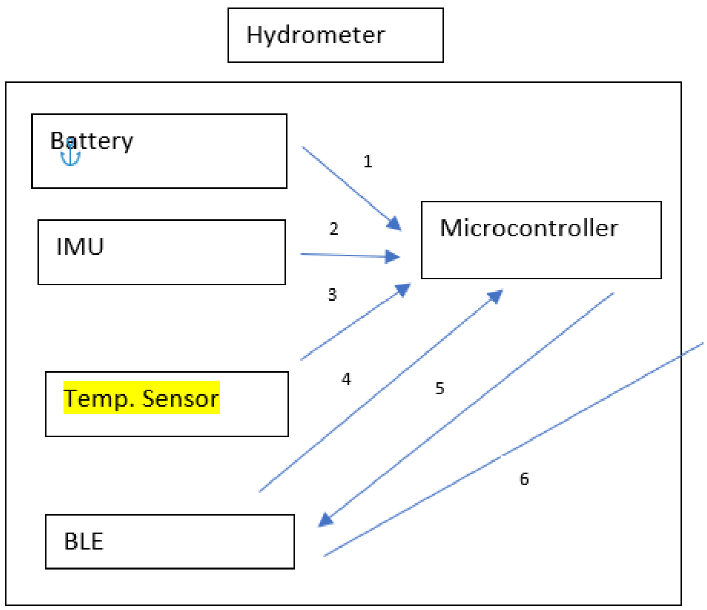

Figure 5: Temperature Sensor subsystem

4.3.1 Assumptions

It is assumed that the temperature sensor will be compatible with the microcontroller we choose and that the output voltages it gives will be fairly accurate. It is also assumed that the temperature sensor will be able to measure a wide variety of temperatures.

4.3.2 Responsibilities

The temperature sensor is responsible for outputting analog voltage that will be used to measure temperature.

4.3.3 Temp Sensor Interfaces Interfaces

Table 4: Temp Sensor interfaces

ID	Description	Inputs	Outputs
#3	Description of the interface/bus	N/A	Temp Measurements

4.4 Bluetooth Low Energy

The on board BLE is the component that requests and receives the specific gravity and temperature data from the microcontroller. It then sends this data to the comms module in the next layer.

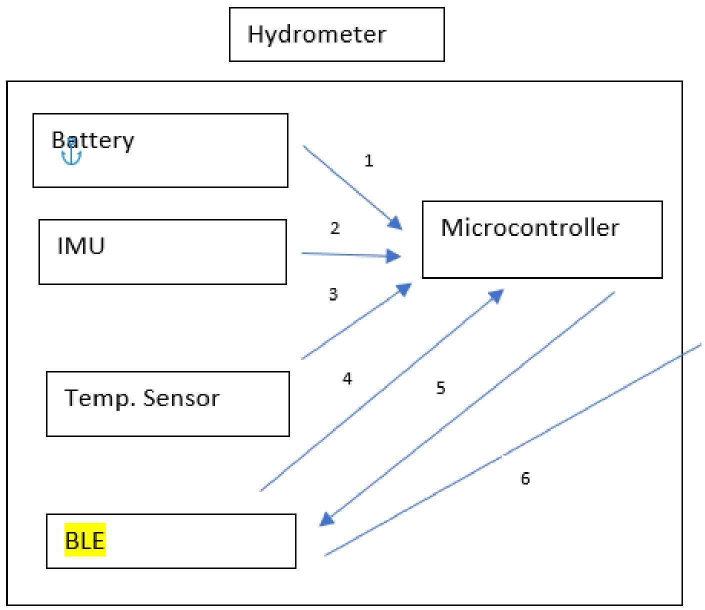

Figure 6: BLE subsystem

4.4.1 Assumptions

It is assumed that the on board Bluetooth module will work as well as the alternative independent BLE.

4.4.2 Responsibilities

The on board Bluetooth is one of the most important components because without it we would not be able to send the specific gravity and temperature data to devices outside the brewing unit. It is the component responsible for requesting that data from the microcontroller and sending it to the comms module.

4.4.3 BLE Interfaces

Table 5: BLE interfaces

ID	Description	Inputs	Outputs
#4	BLE to Microcontroller Transmitter	N/A	Data Request
#5	BLE Receiver	Input Data	Output Data
#6	BLE to Comms Transmitter	N/A	Output Data

4.5 Microcontroller

The microcontroller we choose will be powered by current provided by the battery and will receive data from the IMU, temperature sensor, and BLE module. This data will be processed and sent back to the BLE module for processing in the next layer.

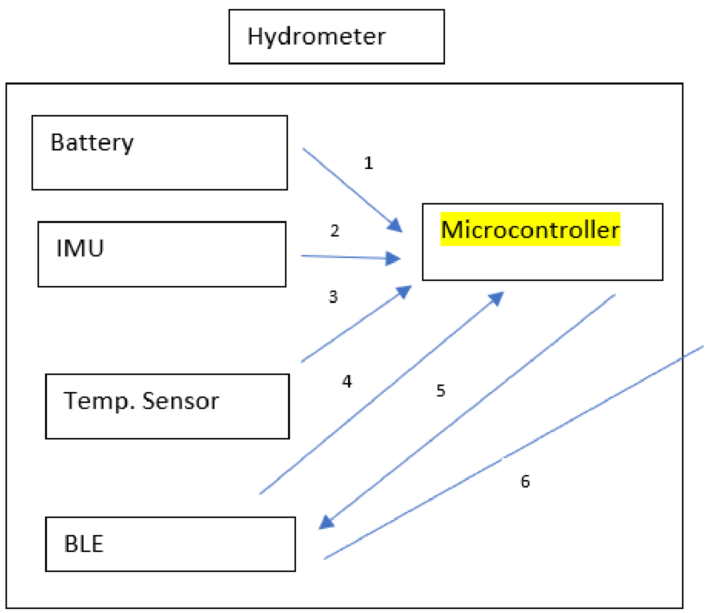

Figure 7: Microcontroller subsystem

4.5.1 Assumptions

It is assumed that the microcontroller we use will function together with all the other components and that it will be of a small enough size for use in our container.

4.5.2 Responsibilities

The Microcontroller will be responsible for receiving voltage data from the temperature sensor and the IMU. It must be powered by a battery and it will also be responsible for receiving and transmitting data to and from the BLE module.

4.5.3 Microcontroller Interfaces

Table 6: Microcontroller interfaces

ID	Description	Inputs	Outputs
#1	Battery Terminal	Battery Current	N/A
#2	IMU ADC	Analog Voltage	N/A
#3	Temp Sensor ADC	Analog Voltage	N/A
#4	Microcontroller UART Receiver	BLE Data	N/A
#5	Microcontroller UART Transmitter	N/A	Output Data

5 Data Acquisition Subsystems

The data acquisition subsystem will include a communications, computation module, and history logging module. This layer is how the data will be transferred from the Hydrometer and processed with the correct calculations. The communications module will receive data from the Hydrometer layer of the system. This data will be stored in a historical logger for later distribution to the user. The computation module will be communicating with both of these modules and communicating with the Data display layer.

5.1 Communications subsystem

The communications subsystem is how the hydrometer will transfer data to the computation module and history logger.

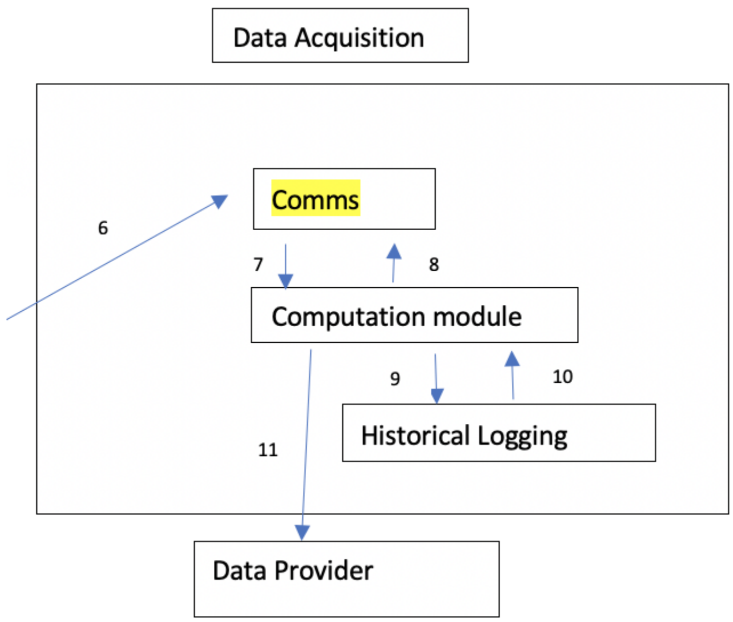

Figure 8: Communication subsystem

5.1.1 Assumptions

We are assuming the communication sub layer does not have to request data from the hydrometer and will be updated based on a hard coded timer.

5.1.2 Responsibilities

The communications module is in charge of receiving data from the hydrometer and transferring this data to the computer.

5.1.3 Communications Interfaces

Table 7: Communication interfaces

ID	Description	Inputs	Outputs
#6	Receiving data from Hydrometer	temperature input angle input	N/A
#7,#8	Computation module requesting data/ Communications sharing data	request from computer	data requested

5.2 Computer Subsystem

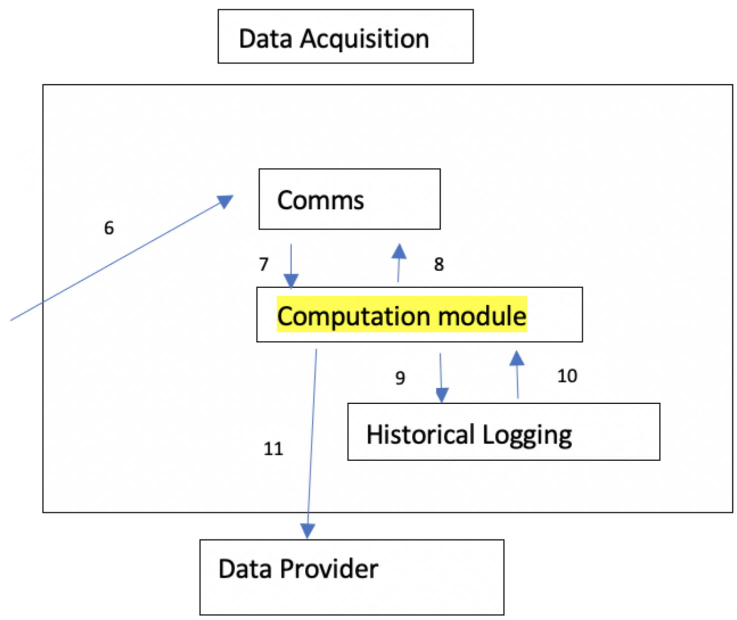

Figure 9: Computer subsystem

5.2.1 Assumptions

We are assuming that the computation module sub layer will be able to request data from the communications module, update the history logger, and request stored data from the history logger.

5.2.2 Responsibilities

The computer will take this data and process it/ do calculations with acquired data. It has a bidirectional communication with the history logger module. The computation module will also communicate with the Data Display layer to update the app with the data the computation module requested from the history logger.

5.2.3 Computer Interfaces

Table 8: Computer interfaces

ID	Description	Inputs	Outputs
#9,#10	Updating History Logger/ requesting stored data in History Logger	requested data	updated data
#11	Updating user app	N/A	updated data

5.3 History Logger Subsystem

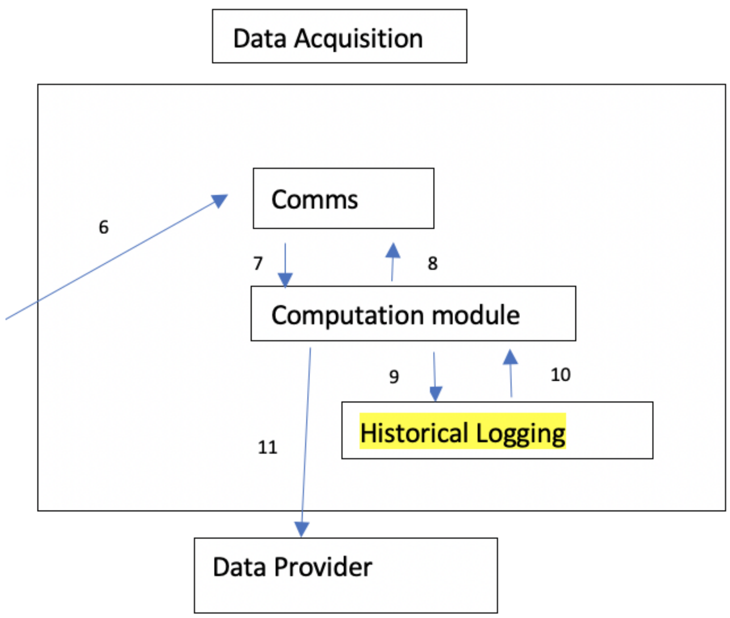

Figure 10: History Logger subsystem

5.3.1 Assumptions

We are assuming that the History logger can receive updates from the computation module and properly store data.

5.3.2 Responsibilities

The history logger module needs bidirectional communication because the it will need to be updated by the computation module and will need to request information from the logger.

5.3.3 History Logger Interfaces

Table 9: History Logger interfaces

ID	Description	Inputs	Outputs
#9,#10	Receives data from Computer sub layer/ shares stored data requested from Computer sub layer	updated data	requested data

6 Data Display Subsystems

The data display subsystem is the subsystem that receives data from computation module to be used by the data provider. The data provider will distribute the data to be displayed to the user. This subsystem is made up of the data provider, current data, as well as historical data components. Specific gravity and temperature data is taken from the data provider to be displayed as either historical or current data.

6.1 Subsystem 1 - Data Provider

This section is illustrating more in depth on how the Data Provider is constructed. This diagram shows how we inquire data from the Data Acquisition layer, then brought over to the Data Display layer to leverage on behalf of the user. We want to display that data to the user in a form of a web service to the data provider.

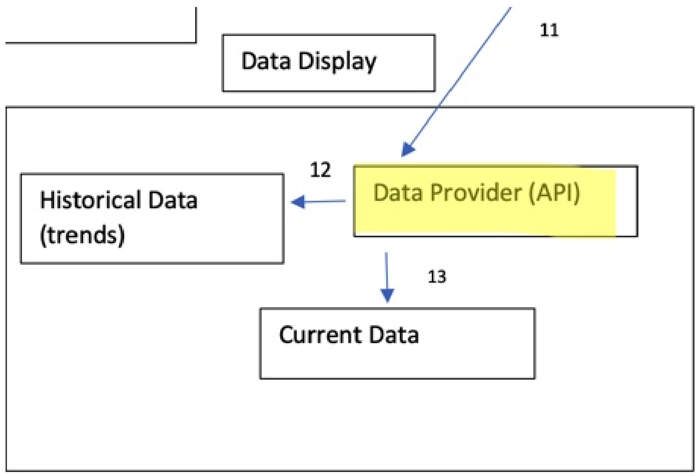

Figure 11: Data Provider Diagram

6.1.1 Assumptions

We assume that we are working with a template framework called Angular, using AngularJS to construct a quick web service, and working with basic API using RESTFul methods to render onto the screen.

6.1.2 Responsibilities

Data Provider: The Data Provider is the API in a sense, that it will go on and retrieve information from the computation module, otherwise also known as a type of database that will be held in place to grab information when needed. The type of database will most likely be fire-base, since it is most compatible with android and google services. This will then be taking advantage of either the historical or current data sections.

6.1.3 Data Provider Interfaces

Table 10: Data Provider interfaces

ID	Description	Inputs	Outputs
#11	Data Provider	Historical Logger/ Database	web page 1

6.2 Subsystem 2 - Current Data

This section illustrates how Current data will be displayed on it's own web page to the user viewing the data. Current data is the data that is currently being displayed continuously through a module set up on a web page that is brought in through data provider. An example of this is through the fermentation process, the specific gravity of the liquid at the exact moment.

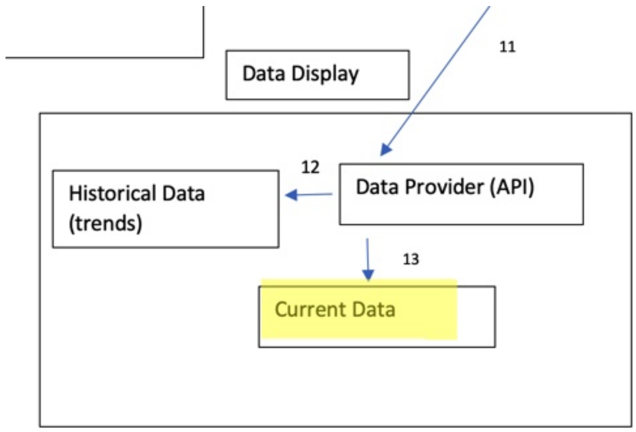

Figure 12: Current Data Diagram

6.2.1 Assumptions

We assume that we will display test points as well to the user to make sure the data they are reading is accurate.

6.2.2 Responsibilities

Current Data: Current Data is it's own web page on the website service that will display data at the very moment the user is requesting continuous data.

6.2.3 Current Data Interfaces

Table 11: Current Data interfaces

ID	Description	Inputs	Outputs
#13	Current Data	Data Provider	N/A

6.3 Subsystem 3 - Historical Data

In this section, the Historical Data layer is used for all trending data that the IMU sends back to user on a web page for their consumption of data analysis.

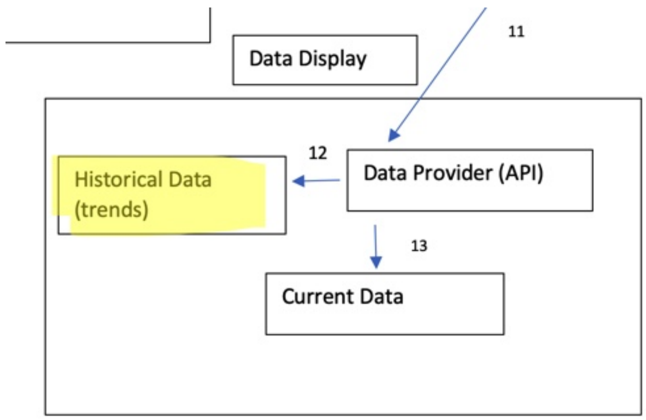

Figure 13: Historical Data Diagram

6.3.1 Assumptions

We assume that we will hold all data, whether it be discrete or continuous onto the database for user's benefit.

6.3.2 Responsibilities

Historical Data: The responsibilities include, effective graphs of discrete and/or continuous data points be displayed for the user. An example may be having the specific gravity of the liquid during the fermentation process. This gives the user an idea of the state of the brew for the past 3 days (trends of an accumulation of data).

6.3.3 Historical Data Interfaces

Table 12: Historical Data Interfaces

ID	Description	Inputs	Outputs
#12	Historical Data	Data Provider	N/A

REFERENCES