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

# Detailed Design Specification
# CSE 4317: Senior Design II
# Summer 2021


## Team Friendship
## Back Burner Brew

Luke Brown
Marcos Juarez Casillas
Ju Young Isa Jung
Sujan Dumaru
Sunghwa Cho
Matthew Francis Schultz

## Revision History

| Revision | Date | Author(s) | Description |
|---|---|---|---|
| 0.1 | 6.14.2021 | JYJ | document creation |
| 0.2 | 6.28.2021 | ALL | complete draft |
| 1.0 | 6.28.2021 | ALL | official release |

# CONTENTS

## List of Figures



## List of Tables

# 1 Introduction

The "Back Burner Brew" is built with the sole purpose of brewing large batch of beer in the home environment. This product provides home brewers with a low-cost electric home brewing system that allow them to have precise control over the brewing process. The brewing process can be automated with the help of sensors and micro-controllers like the ESP32 which is then hosted to a local website or an app interface.

The user should expect to input desired commands, controls, and specific settings such as temperature and length of time by a website that is easily acccessible through a computer or phone. The user can expect that whichever temperature they set for their desired

# 2 System Overview

The Brew System layer is the core of the Back Burner Brew. The goal of the project is to automate the brewing process, and although the system itself is very simple, the steps required to brew a proper beer is time consuming and requires a large amount of attention from the brewer. The user interface layer will ask the brewer to enter their desired temperature and time for each step of the brewing process. The analog and digital components will send data to the digital components. This data will contain the temperatures of the Hot Liqour Tank (HLT) and the mash tun, once the data is receieved by the digital components, the Web server layer will store that data into a cloud database and that data will be used to determine when a heating element or pump will turn on.

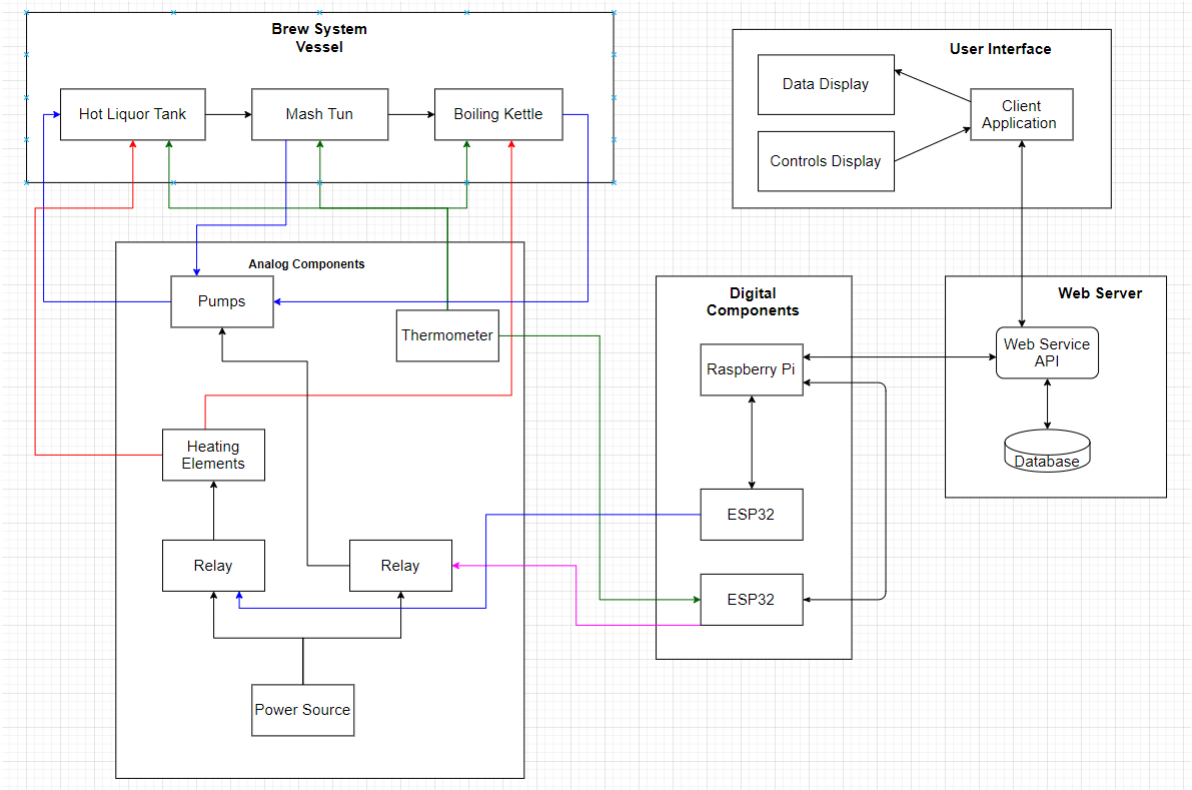


Figure 1: System Overview

# 3 Brew System Vessel Layer Subsystems

This section is completely depended upon the hardware. It is composed of only hardware all combined together to form a suitable environment for brewing. All the hardware must be connected properly with the system to ensure quality beverages.

## 3.1 Layer Hardware

The most important hardware required for this layer are three vessels/kettles, probably of 5 gallon each. Each vessel is connected to the 330 GPH Low Suction Electric pumps with suction hose to pump the liquid out of the vessels and DS18B20 Thermometer Temperature Sensor Probe in order to keep track of the temperature in each vessel.

## 3.2 Layer Operating System

There is no operating system involved in this layer. However, the temperature data and the pump of this layer is controlled by the Arduino Uno micro controller which is in different layer.

## 3.3 Layer Software Dependencies

Since this is completely hardware dependent layer, there is no any software dependencies.

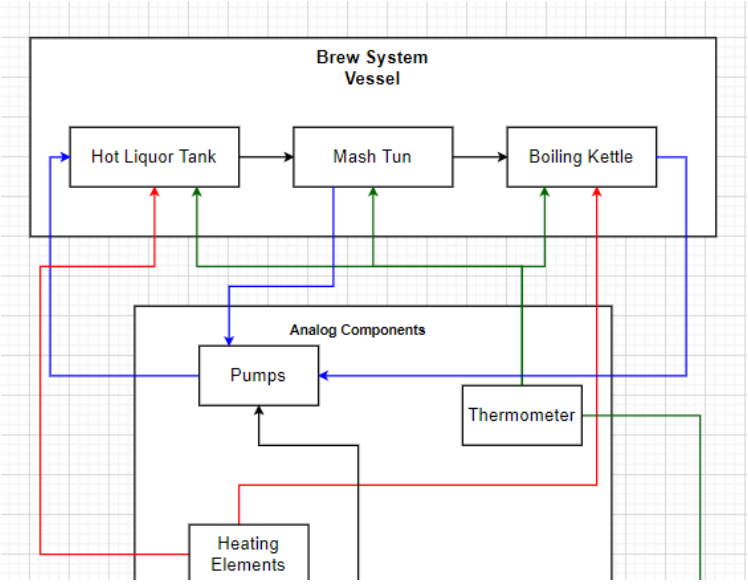


Figure 2: Brew System Vessel Layer Representation

## 3.4 Hot Liquor Tank Subsystem (HLT)

The main purpose of Hot Liquor Tank subsystem is to monitor and heat the water to the desired temperature. This subsystem is simply a vessel hardware used for boiling purpose.

### 3.4.1 Subsystem Hardware

Apart from the hardware mentioned above, this subsystem have one more hardware involved which is a Kettle Heating Element.

### 3.4.2 Subsystem Operating System

No Operating System is involved in this subsection.

### 3.4.3 Subsystem Software Dependencies

The Heating Element is depended on the input provided by the user on the basis of temperature as the water is only heated to that temperature. So, this subsystem require input from the user/brewer to properly function.

### 3.4.4 Subsystem Programming Languages

In order to provide the input, the user is using Arduino programming language for Arduino Uno.

### 3.4.5 Subsystem Data Structures

No specific data structures are being used in this subsystem. It is plain number being passed in and out of this subsystem.

### 3.4.6 Subsystem Data Processing

The thermometer temperature sensor probe reads the data and simply passes it to micro controller and the micro controller uses that data in order to raise the temperature of water by turning on or off the heating element.

## 3.5 Mash Tun Subsystem

The main purpose of Mash Tun subsystem is to mix the hot water with grains to produce wort. This subsystem is simply a vessel hardware to separate mash and wort.

### 3.5.1 Subsystem Hardware

There are no separate hardware used in this subsystem.

### 3.5.2 Subsystem Operating System

No Operating System is involved in this subsection.

### 3.5.3 Subsystem Software Dependencies

The hardware used in this subsystem are not software dependencies.

### 3.5.4 Subsystem Programming Languages

No programming languages are used.

### 3.5.5 Subsystem Data Structures

No specific data structures are being used in this subsystem. there is just a float value being passed out of this subsystem.

### 3.5.6 Subsystem Data Processing

The thermometer temperature sensor probe reads the data and simply passes it to micro controller.

## 3.6 Boiling Kettle Subsystem

The main purpose of Boiling Kettle subsystem is to boil wort for the required amount of time. This subsystem is simply the final vessel hardware used to produce beverage.

### 3.6.1 Subsystem Hardware

Apart from the hardware mentioned above, this subsystem have more hardware involved like a Kettle Heating Element and a NY Brew Supply copper wort chiller.

### 3.6.2 Subsystem Operating System

No Operating System is involved in this subsection.

### 3.6.3 Subsystem Software Dependencies

The software dependencies is similar to Hot Liquor Tank Subsystem. The Heating Element is depended on the input provided by the user on the basis of temperature as the wort is only heated to that temperature.

### 3.6.4 Subsystem Programming Languages

In order to provide the input, the user is using Arduino programming language for Arduino Uno.

### 3.6.5 Subsystem Data Structures

No specific data structures are being used in this subsystem. It is plain number being passed in and out of this subsystem.

### 3.6.6 Subsystem Data Processing

The thermometer temperature sensor probe reads the data and simply passes it to micro controller and the micro controller uses that data in order to raise the temperature of water by turning on or off the heating element. After the desired time is reached, the wort is sent to chiller which then cool downs to form beverages.

# 4 Y Layer Subsystems

In this section, the layer is described in terms of the hardware and software design. Specific implementation details, such as hardware components, programming languages, software dependencies, operating systems, etc. should be discussed. Any unnecessary items can be ommitted (for example, a pure software module without any specific hardware should not include a hardware subsection). The organization, titles, and content of the sections below can be modified as necessary for the project.

## 4.1 Layer Hardware

A description of any involved hardware components for the layer. For example, if each subsystem is a software process running on an embedded computer, discuss the specifics of that device here. Do not list a hardware component that only exists at the subsystem level (include it in the following sections).

## 4.2 Layer Operating System

A description of any operating systems required by the layer.

## 4.3 Layer Software Dependencies

A description of any software dependencies (libraries, frameworks, etc) required by the layer.

## 4.4 Subsystem 1

Descibe at a high level the purpose and basic design of this subsystem. Is it a piece of hardware, a class, a web service, or something else? Note that each of the subsystem items below are meant to be specific to that subystem and not a repeat of anything discussed above for the overall layer.

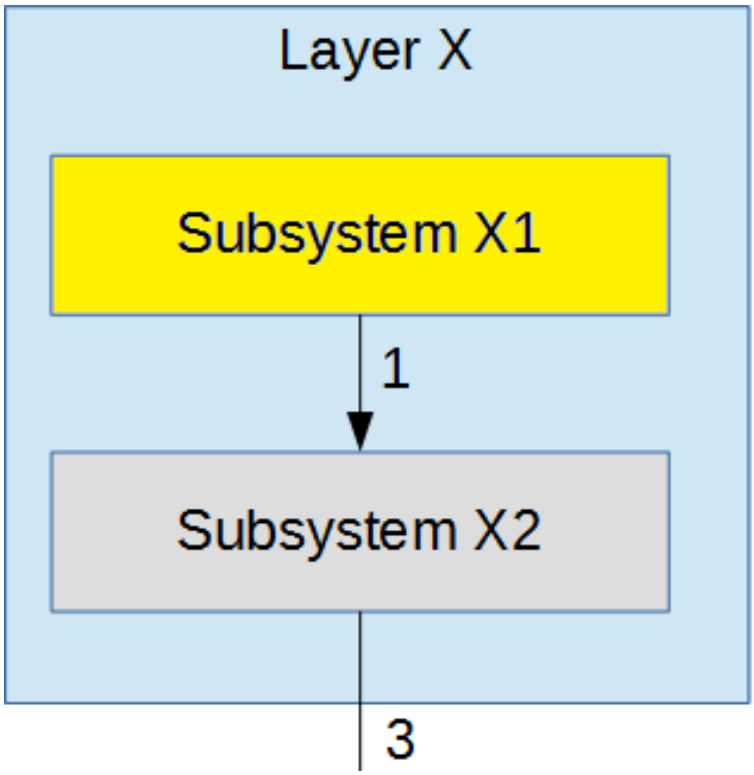


Figure 3: Example subsystem description diagram

#### 4.4.1 Subsystem Hardware

A description of any involved hardware components for the subsystem.

#### 4.4.2 Subsystem Operating System

A description of any operating systems required by the subsystem.

#### 4.4.3 Subsystem Software Dependencies

A description of any software dependencies (libraries, frameworks, design software for mechanical parts or circuits, etc) required by the subsystem.

#### 4.4.4 Subsystem Programming Languages

A description of any programming languages used by the subsystem.

#### 4.4.5 Subsystem Data Structures

A description of any classes or other data structures that are worth discussing for the subsystem. For example, data being transmitted from a microcontroller to a PC via USB should be first be assembled into packets. What is the structure of the packets?

#### 4.4.6 Subsystem Data Processing

A description of any algorithms or processing strategies that are worth discussing for the subsystem. If you are implementing a well-known algorithm, list it. If it is something unique to this project, discuss it in greater detail.

# 5 Web Service Layer Subsystems

Since web service API and Database are hosted on Cloud service, no hardware is required. This section will discuss API tool platforms and databases.

## 5.1 Web Service API

API is an application interface that integrates two applications or systems to support the transfer of data to each other. Mulesoft is a platform where provides many convenient functions to build and deploy API. Using Mulesoft make it easier to connect MySQL.

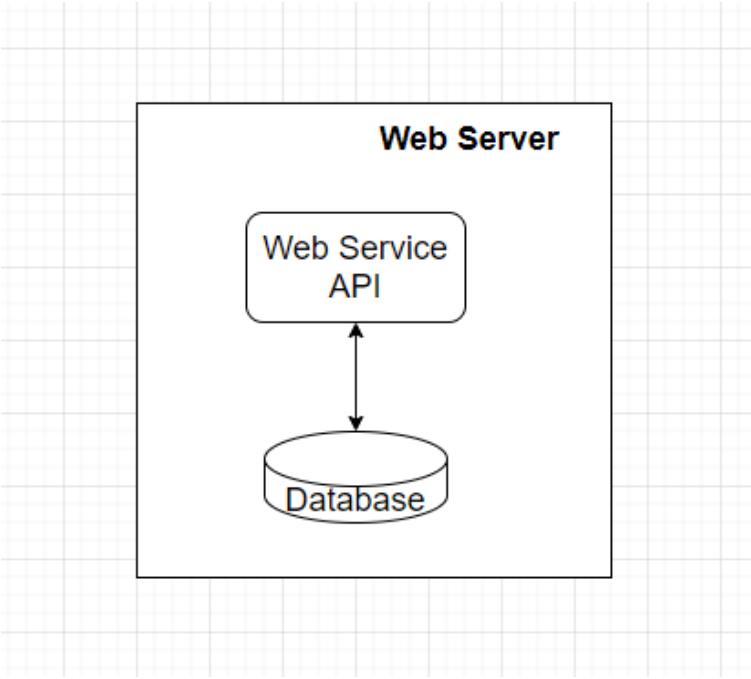


Figure 4: Example subsystem description diagram

### 5.1.1 Web Service API Software Dependencies

MuleSoft Exchange provides many dependencies and other APIs that are needed to build our API through a connector that prebuilds connectivity to an endpoint.

### 5.1.2 Web Service API Programming Languages

DataWeave is the MuleSoft expression language for accessing and transforming data that travels through a Mule app.

### 5.1.3 Web Service API Data Structures

MuleSoft Design Center has a function called API specification which helps to define incoming and output data format and creates automated documentation for the APIs. Through API specification, the general structure of API will be provided.

### 5.1.4 Web Service API Data Processing

API will listen to HTTP requests and based on a certain path, it will retrieve data from the database and return it to the user.

## 5.2 Database

The online database allows you to store sensor readings from the ESP32 and read from anywhere in the world by accessing your server domain.

### 5.2.1 Database Programming Languages

A query is used to communicate with the database.

### 5.2.2 Database Data Structures

There are some required tables that the database has to have such as temperature for each component and recipe.

### 5.2.3 Database Data Processing

The database will Run the query for requests and return the data to API.

# 6 User Interface Layer Subsystems

## 6.1 User Interface

The website used to control the system will be compatible with major web browsers and will be connected to a database and with the brewing system. Website is verified to work on Chrome for Windows 10, version 91.0.4472.124, and Chrome for MacOS Big Sur, version 91.0.4472.124

## 6.2 Layer Operating System

Website available on Windows 10 version 21H1 and MacOS Big Sur version 11.4

## 6.3 Layer Software Dependencies

The user interface components will depend on Mulesoft's Mule software, version 4.3.0

## 6.4 Website

The website provided to the user will have multiple responsabilities. First, it will display real-time information relevant to the user about the current brew. The website will also display a controls section for the user to interact with the brewing system. The information send to and from the website will be sent to the servers and then to the brewing system utilizing the above mentioned Mulesoft software.

### 6.4.1 Subsystem Operating System

No specific operating system required.

### 6.4.2 Subsystem Software Dependencies

The website and UI will utilize the React JS library

### 6.4.3 Subsystem Programming Languages

JavaScript will be used for creating the website.

# 7 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

## References