DRINC

Dynamically Refreshing Interplexing Number of Cordials

Brandon Arnold

Owen Ledvina

Hoang Phan

Kyle Timins

# REVISION HISTORY

First Revision: 2013-11-01

# INTRODUCTION

## PURPOSE OF THIS DOCUMENT

The purpose of this system architecture document is in partial fulfillment of the DRINC project.

## ARCHITECTURAL ASSUMPTIONS

-The machine will not collapse on its own.

-The bottles must not be able to be moved without removing any safety restraints.

**SUMMARY OF REQUIREMENTS**

**HARDWARE**

-Frame must securely hold 9 liters of mixers, glass enclosures as well as the back end control systems and drink transport track.

-Power Supply must be capable of powering the back end control system, drink transport track, valves, and flow meters must be included.

-A microcontroller with enough I/O pins to monitor a flow meter and valve for every mixer, as well as the servos controlling the drink transport track and a data connection to the front end system must be used to control the back end control systems

-The drink transport track must be able to safely, securely, and accurately move a cup to any point on a square grid.

-The transport track requires at least two servos, which must be strong enough to move a full pint of mixed drink reliably.

-Each mixer must have a valve that can be quickly turned on and off via the back end control system.

**SOFTWARE**

-The website must be able to log in a user based off of authentication held in the backend database.

-The website must be able to log a user off of the account.

-The website must present the user with the main menu upon logging in

-The website must have the following options:

-Create a Custom Drink

-Select a Drink

-Most Drank

-The Android app must be able to log in a user based off of authentication held in the backend database.

-The Android app must be able to log a user off of the account.

-The Android app must present the user with the main menu upon logging in

-The Android app must have the following options:

-Create a Custom Drink

-Select a Drink

-Most Drank

The goal of this project is to create a machine that will accurately mix drinks selected by the user from a preset list of mixers.

**SYSTEM ARCHITECTURAL DECOMPOSITION**

**PHYSICAL/MECHANICAL ARCHITECTURE**

Back end requirements:

The Raspberry Pi

-Processor: Broadcom 700 MHz

-RAM: 256MB

-Graphics: VideoCore IV

-OS: Debian Linux

Front end requirements:

The Nexus 7 Android

-Processor: ARM Cortex-A9 Nvideia Tegra 3 T30L 1.2 GHz quad-core

-RAM: 1 GB

-Graphics: 7in Touch screen

-OS: Android Jelly Bean

Hardware design overview

Touch screen talks to backend server on the raspberry pi. This formats the proper protocol commands that are sent to the arduino. The arduino then processes these commands and sends the proper data instructions to the proper pins that will control the solenoids.

Frontend

The frontend is an Android device which will run the frontend software described later.

The frontend will connect to the backend with a custom protocol that will communicate with the front end over a USB cable.

Backend

The backend is a raspberry pi running the backend software and and an arduino device that will communicate with the valves/flow meters directly.

Machine

-Frame

The frame of the machine will accommodate all other components. It will suspend 9 bottles 750mL in volume above a track consisting of a cup holder moving horizontally along two rails assisted by a pair of servos. The frame will also house an Arduino Mega, a Raspberry Pi, and a standard ATX power supply, as well as a mounting point for the frontend device.

-Electronics

The backend will communicate with an Arduino Mega microcontroller. The microcontroller will in turn control a valve attached to each available mixer. Additionally, the microcontroller will control a pair of servos to move the cup holder to any of the available mixers.

-Arduino Mega:

-16MHz core clock

-124KB available flash

-49 digital I/O pins

-15 digital I/O pins capable of PWM

-16 analog I/O pins

-Valves:

-Aquatech AQT15SP Solenoid Valve

-12V operating voltage

-¾” diameter

-Servos:

-43R Robot Rotation Servo

-5V operating voltage

-Direction and speed controlled

-60 RPM

-Power Supply

A standard 400W ATX power supply will adequately power all electronic components. The backend can bring the PSU out of standby and shut it back down when the machine is not in use.

**SYSTEM ARCHITECTURE**

**SOFTWARE ARCHITECTURAL DECOMPOSITION**

This section will describe, in detail, the composition, function, and interaction of the DRINC’s software subsystems and their components. These subsystems include the website, Android application, Apache web hosting software, and PostgreSQL database software.

**SOFTWARE ARCHITECTURE**

**Website:**

The HTML/PHP website will run over the Apache server, and integrate with the PostgreSQL database. The website will incorporate CSS for styling and visual appearance, and Javascript to enhance the appearance and functionality of the website.

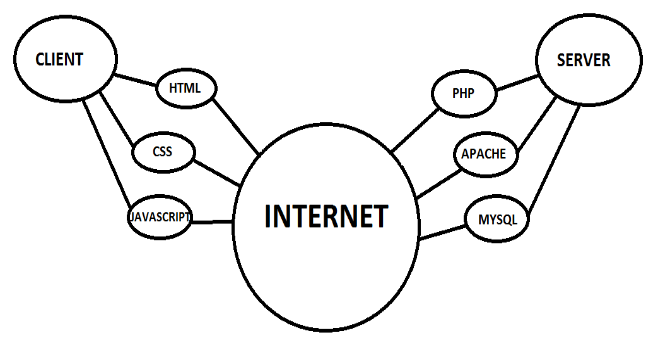
**A basic mockup of the login page for the DRINC website.**

**Android App:**

The Android application will receive the HTML website from the Apache server, and format it for proper dimension sizing on a mobile device.

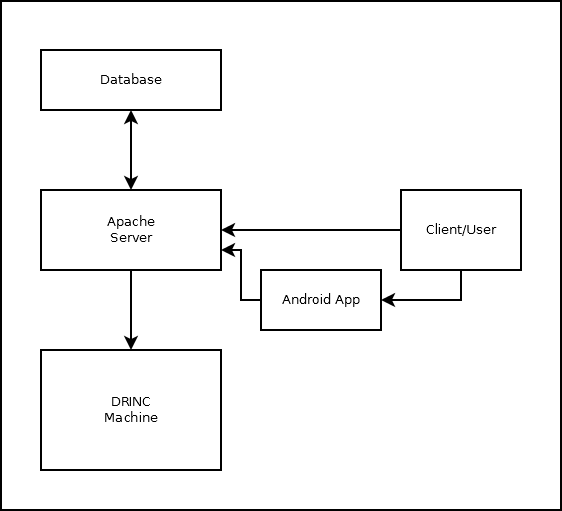
**Apache Server:**

The Apache server will run server side PHP code, and send HTML code over port 80 to the client. It will also connect to the machine to dispense the drinks.



**PostgreSQL:**

The PostgreSQL database will store login credentials, and a list of drinks saved for each user. All private or sensitive information stored in the database will be salted and hashed before being put into the server.



**CONCLUSION**

This document is the System and Software Architecture Document for the DRINC project. The document provides architectural descriptions of the systems and processes involved in automating the process of creating a mixed drink. This document is meant to be used by the software development team in implementing the system functionality in code.

**REFERENCES**

http://www.postgresql.org/

http://httpd.apache.org/

https://projects.gnome.org/dia/