# Senior Project Proposal

https://pabstaaron.github.io/AutoCoffeeMaker/

Aaron Pabst, Ben Nagel, Nathan Donaldson, Elliot Carr-Lee

***Abstract*—Imagine having a machine that would allow you to make the perfect cup of coffee right in your own home or office, without ever having to trek to the coffee shop. A machine that allows you to specify exactly how you like your coffee with exacting detail and optional scheduling at which point it will flawlessly produce that beverage for you. A machine that can recommend new beverages based on your and other users past preferences.**

**There are plenty of machines on the market that claim to be fully automatic, but none are capable of going all the way from raw ingredients to a finished beverage without any intervention from the user. With these machines, the user still has to manually froth milk, dispense flavoring syrup, and mix the beverage themselves; the machine only handles the grinding, tamping, and brewing tasks. These machines also provide a very limited scope of control to the user, giving them only a few options for dictating how they would like their coffee to be produced. Finally, most of these machines have very limited user interfaces and few have an option for remote control from a more user friendly device, such as a smart phone or tablet.**

**Our product differs by focusing on automation, personalization, and user-friendliness above all else. Our machine will house all ingredients internally, receive instructions wirelessly, and will have an automatic cleaning mechanism. The minimization of human interaction allows us to have complete control of the brewing and mixing process, allowing us to make a consistent cup of coffee. This also allows us to operate the machine wirelessly, allowing for streamlined UI capabilities and scheduled events. Autonomous control allows you to monitor the machines use, personal consumption habits, and ingredient consumption, while also giving the user the ability to tweak individual settings to make a personalized and reproducible cup of coffee. The machine itself will consist of a series of boilers, chillers, and pumps as well as a specially designed chamber for automatically frothing milk. These components will be driven from custom heater and chiller control circuitry as well as an embedded Linux controller. The physical device will be backed by some remote user-interface and a database for storing ingredient information and user data.**

## I. Introduction

There are many espresso machines available for commercial and consumer applications. The most frequent of which an average consumer will encounter is the manual espresso machine used by their local coffee shop. These machines require a substantial amount of training and practice to wield effectively. The operator must master the skills of grinding the coffee, tamping grounds, and physically pulling the espresso; tasks that are out of reach for the average consumer to perform on their own. What's more, the operator must manually froth milk and dispense an appropriate amount of flavoring syrup for more complicated beverages, such as lattes.

However, there are an increasing number of espresso machines on the market that automate some of this process. Most of these machines grind beans, tamp the grounds, and pull the espresso with little intervention from the user. The user, however, must still manually froth milk (a difficult task to master), deploy flavoring syrup, and mix the beverage on their own.

There is not presently an *eleagantly* implemented solution for a fully automated espresso machine that is easy and fast for the average consumer to use. This is largely due to the fact that there are certain mechanisms and control systems that would have to be created for such a device that are non-trivial to design and implement.

The remainder of this document discusses the functionality of various modern coffee machines. We then compare our implementation of a fully automated, web connected espresso machine. The discussion itself will include the scope of the project, the design approach, some background, and time estimates.

### *A. How Espresso Machines Work*

There are many different types of coffee makers available. Each of these systems is unique in its own way and each has its pro's and con's. All coffee makers, however, have one thing in common: they all must push hot water through ground coffee beans in some way. In the case of espresso, hot water is heated to a near boiling temperature and pressurized in some way. This hot water is then forced through densely packed coffee grounds (known as a "puck") in order to produce a thick, creamy coffee [3].

There are several different ways in which the water may be pressurized. One of the most common ways this is accomplished is to simply let the water pressurize as it heats in a sealed container and turns into steam. Once a suitable temperature is reached, the container is unsealed and the pressurized water passes through the puck. This approach is used in most low-end espresso machines as it requires few mechanical components and is generally inexpensive. It also tends to produce lower quality espresso as the water pressure is difficult to regulate and drops as the brewing cycle progresses.

Higher end machines used in most commercial applications use an electric pump to force the water through the grounds, allowing for tighter control of the water pressure as well as faster brewing times.

An optional, but important, component of an espresso machine is the frothing wand. The frothing wand is a hollow shaft of aluminum that is used to direct high pressure steam into milk (or a milk-like product), the effect of which is incorporating air into the milk that makes it light and foamy (or frothy, as the name suggests). Most modern espresso machines have a built-in frothing mechanism. On lower end machines, the frothing wand connects to the same boiler that produces the brewing water. This is undesirable due to the fact that frothing water needs to be heated to much higher temperatures than brewing water in order to produce the necessary high pressure steam. Machines that only have a single boiler therefore need to introduce a long delay between the brewing and frothing cycle while the boiler switches from one task to another.

Higher-end machines will generally introduce a second boiler for producing frothing steam. This boiler will operate at a much higher temperature than the brewing boiler in order to produce the necessary steam pressure.

Another important task that plays a large role in producing high quality espresso is the grinding of the beans. While this task seems simple, there are many factors introduced in grinding that can have a large impact in the flavor and consistency of the final product.

Coffee for espresso is usually ground to a very fine powder so that it can be densely packed into a puck. This is subject to personal preferance, however, and some may prefer the flavor of a coarser grind. In many modern espresso machines, the grinder is integrated into the machine. Many baristas, however, prefer to use a seperate grinder to give them more control over the process.

Most grinders used in commercial and upscale domestic applications are burr grinders. Burr grinders have two vertically aligned steel plates with teeth (burrs) on them. A hopper containing coffee beans sits above these two plates and allows coffee beans to fall into the grinder. The plates are roatated manually or via an electric motor to grind the beans, which are pushed through the teeth and fall into a collection vessel once they are split into fine enough pieces. The space between the plates determines the final fineness of the grind.

## II. Project Overview

This project will culminate in a device that can produce espresso drinks that consist of varying degrees of espresso, frothed milk, and flavoring syrup. The device will be capable of varying parameters that effect the brewing and frothing processes. The user will be able to define how hot the brewing water should be, the pressure with which it is pushed through the grounds, how fine of a grind is used, the ratio of water to coffee grounds, how much steam is used to froth milk, and how much froth is produced in the milk. The machine will be fully self-contained and hold its own water, coffee beans, milk, milk alternative, and flavoring syrup (milk will be held in a refridgerated tank). The machine will additionally be capable of a degree of self-cleaning and will be capable of flushing the non-refigerated portion of the frothing system with detegergent to prevent bacteria build up.

The machine will be internet connected and ingredient information and collected user data will be stored and processed in an SQL database. The database will provide the machine with information on how to use certain ingredients (for example, it may provide optimal brew settings for a certain brand of coffee) and recommend beverages and settings to the user.

The machine will be operated by some remote user interface running on an Android device to allow for a more streamlined user experience.

### A. Physical Machine Overview

The physical espresso machine will consist of a brewing mechanism, a burr grinder, a tamping mechanism, a frother, several syrup dispensers, and storage tanks for water, milk and a non-dairy equivalent, and detergent.

*1) Burr Grinder:* The grinder will be based around flat (rather than conical) burr plates. The burr plates will first be designed in a 3D CAD system and then be fabricated in one of two ways, depending on cost and time constraints.

The plates may be sent to a computer aided-machining service to be milled out of stainless steel.

The 3D design may alternatively be used to 3D print a negative impression (mold) of the plates. This mold will be used to slip cast ceramic (boron carbide, preferably) burrs. These slip cast plates will then need to be hardened in a kiln [2]. A similar process is used in the production of ballistic body armor. Ceramic burrs have many advantages over steel burrs.

The top-most of the burrs will be stationary (with respect to rotary motion) while the lower burr is connected to a sufficiently powerful dc moter. The vertical distance between the two plates will be adjustable via a stepper motor coupled to a threaded shaft that will raise or lower the upper plate.

*2) Tamping and Brewing Mechanism:* The brewing mechanism will consist of the following major pieces: a boiler for heating brew water, a peristaltic pump for pushing the water through the grounds, and a valve for allowing water into the boiler from the storage tank. The boiler will be a cylindrical structure built out of stainless steel. This boiler will have two threads for connecting it to the water source and to the pump. Another connection point will be present for attaching a thermocouple.

*3) Automatic Frother:* The frother will be implemented as a chamber sitting directly above the dispensing end of the device. Milk will be pumped into this chamber from the refrigeration tank. Inside this chamber there will be a telescoping pipe attached to a servo acting as the frothing wand. This pipe will be able to autonomously move into and out of the milk and will be connected to a boiler producing high pressure steam. In this manner, the machine will be able to froth exactly as a human barista would.

The wand will be designed in a 3D CAD program and then 3D printed to verify functionality, at which point the device will be sent to a computer-aided machining service to be fabricated in stainless steel or aluminum.

The boiler will need to be capable of reaching much higher temperatures than the one used for producing brew water and will need to be capable of withstanding a large amount pressure. For this reason, a safety blow valve will be incorporated into the steam boiler.

### B. Mobile User Interface Overview

The UI will be implemented as an Android application written in Java. Upon opening the application, it will ask the user to pair their Android device to the coffee machine given it's serial number. Afterwards It will take the user to a login screen which will allow the user to register or not. Registering will allow the user to have a few extra options given from a database integration. These options will allow the user to search recent beverage choices, favorites, recommendations, schedules, and an online database. Each menu will include a similar list layout with searchable drinks. Upon clicking one of the listed items, it will show all of the settings and have descriptions based on what the user (whether it is yourself or another user) has written. It will then have the option of favoriting, brewing, and scheduling the drink right in that given screen. In the scheduling menu, drink schedules may be canceled or changed. The benefit of registering will allow a user to login on any application and have all of the features described above. Given the choice of not creating a login, a user will still have to pair their phone to the device through the database, but wont have to deal with any extra UI that comes with a registered user. A non- registered user will be allowed to brew coffee the way they want at that instant, and that will be the most basic functionality. Registering will be as simple as creating a login name and password with an email verification to prevent non-human users from registering. After authentication, you will be sent to a screen with multiple options, buttons that will take you to menus such as: scheduling, beverage creation, recent beverages, recommended beverages, favorited beverages, or a searchable database. If the user is not registered, and the device is paired, it will take them to the beverage creation menu. For registered users, the schedule, recent, recommended, and favorited menus will be available and of the same list format. For the beverage creation menu, there will be advanced and basic settings options, allowing the user to be very precise, or general in their brew. Once all settings are selected, if registered, a user may have the decision to leave information on the drink created and post it in a searchable database and/or their recents/favorites. Information may be a title, description, etc. If the user is not registered, it will not prompt them for anything and brew their drink.

### C. Data Aggregation and Processing Overview

The application will store user registrations, favorites, and other data in a standard SQL database. This database may be hosted separately from the machine, and could potentially be on a cloud service provider. The server will be encapsulated in an application server. The application server provides a container that can manage the database and its communication with the user application. The REST protocol will be used to communicate with the application server. Code running inside the application server will translate JSON or XML objects sent by the user into Java objects. Mappers will be used to dynamically translate these objects into SQL statements to update the database. The database will posses a users, beverages, and favorite's table. The users table will store core user data: name, email, salted password hashes, etc. The beverages table will store all of the settings you would need to instruct the coffee maker to produce an exact drink. Each beverage and user table row has a unique UUID. Each favorites list row will be connected to a single user via the UUID. Finally, each row of the favorites list will store all of the UUIDs that are associated with that list. That way, when a user sends a REST protocol request to get all of the favorites list of a particular user, the mapper can generate a SELECT request to grab all the relevant lists from the favorites table and then use the uuids stored within those rows of the favorites table to grab the drinks associated with each list. All the aggregated data is then returned to the user via REST protocol.

### D. Flask

Flask is a web framework that provides tools to allow you to build a web application. Flask is a micro-framework, so it requires no outside dependencies or external libraries. This means that Flask is lightweight. Flask is designed for creating web applications, with a database backend that operates through the browser. We will be using it to open a port on the local wifi network that will act as a REST server. We will use this server to send commands through the Raspberry Pi to the microcontroller. Thus, Flask is a middleman from the android application and the components inside the coffee machine.

## III. Project Tasks

### A. Physical Machine

- Design frothing mechanism
- Design tamping mechanism
- Burr design completed
- Design temperature regulation circuity
- Design mainboard based around the Raspberry Pi Compute Module
- Design grinding mechanism
- Fabricate frothing wand, tamping mechanism, and burr plate mold
- Fabricate ceramic burr plates
- Construct boilers
- Construct refrigeration system
- Assemble grinder
- Assemble brewing system
- Assemble frother

### B. Mobile User Interface

- Device pairing main screen
- Login screen which allows optional user login/registration or not.
- Main menu screen, allowing decision of brew creation, recommendations, recents, favorites, and a search menu.
- Beverage creation screen has an advanced/basic toggle button and sliders, value inputs, and other selections

- Recommendations, scheduling, recents, favorites, and search sharing same UI layout (minor difference in scheduling), will query different data structures.
- Database integration with mobile and coffee machine
- Communication with web server
- Communication between application, webserver, database and coffee machine.

### C. Machine Side Software

- Flask application setup
- Flask can receive commands from Android application
- Mainboard communicating with regulators and monitors
- Mainboard controls motors, servos, and steppers
- Mainboard can adjust grinder fineness and start/stop grinding mechanism
- Mainboard can initiate tamping process
- Mainboard can initiate brewing process
- Mainboard can initiate frothing process
- Mainboard can initiate cleaning process

### D. Database/Web Server

- decide which application server, database mapper, and programming language to use for web server.
- Setup application server that is capable of receiving and sending REST requests.
- Establish API for communicating with android application (decide on communication format, i.e. xml or json)
- Convert REST data into language specific objects.
- Develop a set of mappers that can convert language specific objects to SQL inserts and vice versa.
- Set up SQL Database and populate schema.

## IV. Project Materials

- Raspberry Pi compute module
- Stainless steel stock
- Cartridge heaters
- Custom PCBs
- Various IC's and discretes
- Copper pipe
- Silicone Tubing
- Custom machined parts
- Boron carbide powder
- PLA plastic
- SLA Resin
- Peltier elements
- Moldmax 30 silicone
- Servos
- Stepper motors
- Peristaltic pumps
- Thermocouples
- Liquid pressure sensors
- DC motors

## V. Testing Approach

### A. Machine Side Software Testing

Testing against the Flask microframework part of our project will be heavily reliant on 'mocking' outputs and inputs of recieved calls. Flask as a microframework does not need to be tested, however our devices reaction to GPIO eventst needs to be tested. This can be occumplished using the python unittest.mock framework [1] where we can create tests that will 'mock' returned data to be able to flush out multiple tests without relying on any network or GPIO setup. We will also have to test a buffer system to make sure 'real' requests are getting recieved and queued and multiple requests of the same requests are discarded.

### B. Physical Machine Testing

*1) Mechanical Testing:* Considering some of the physical components will involve heat, pressure, and sharp spinning burrs, careful testing will need to be preformed on each aspect of the physical machine in order to ensure the safety of the final product.

Starting with the water boilers, these vessels will be responsible for containing water under pressure and will need to be sufficently strong to withstand this pressure. If the boilers are not strong enough or if the temperature inside them is not well monitored and controled, explosive decompression could occur. In order to assure that the vessels are up to specifications, they will be filled with the largest possible quantity of water and heated to a temperature that far exceeds the maximum operating point of the machine. This pressuraization will occur in an environment away from passerbys and will be monitored very carefully for signs of failure. The pressure will be held for ten minutes and then slowly released via a remotely operated solenoid valve.

*2) Electrical Testing:* All heat and motor controlers will be verified through independent testing outside of the main machine.

### C. Mobile User Interface Testing

Mobile Interface testing will be done one task at a time. Creating the layout and listeners will be the first goal. By doing this it will ensure a skeleton to work with. Each screen will be done one at a time, moving in the flow as a user would. Afterward another pass will be taken, adding data. Starting with the login screen, communication between database, webserver, and the device will be established and pairing/login data will be tested and verified on every side. We then move to the basic function of non registered user: creating a drink. If the mechanical portions are not ready yet, then data collection will be done next on all of the other menus: scheduling, favorites, recommendations, recents, and online searching. During all of these integrations, testing with the actual coffee machine will take place, ensuring communication is working on all ends.

## VI. Project Demonstration

At the project demonstation, the machine will be setup with an Android tablet and loaded with ingredients. Visitors will be invited to use the tablet to design a drink watch the machine in action. Various spare internal components will be set out to aide in explaining how the machine works.

### A. *Mobile Demonstration*

We will first demonstrate functionality with non-registered user implementation (the basics on mobile) and how the mobile device can communicate with the machine properly to make a cup of coffee that is asked for. Then we will demonstrate how data collection offers different options to users. Grabbing information from the database and laying it out in multiple menu's will give users the ease of quickly making what they need or searching for something new. Using these different menu's, we will show that the data integrity holds and creates the drink as it should.

## REFERENCES

[1] Python Software Foundation. 26.5 unittest.mock - mock object library.
[2] https://www.thespruce.com/how-to-slipcast-ceramics 4154220. Sara d'souza.
[3] John Smith. Wikipedia - espresso machine.