

4TB6: Development Plan

Stonecap Solutions - Smart Serve

Max Turek *turekm*

Ryan Were *werer*

Sam Nusselder *nusselds*

Peter Minbashian *minbashp*

David Bednar *bednad1*

09/26/2022

Contents

1 Team Meeting Plan 3

2 Team Communication Plan 3

3 Team Member Roles 3

4 Workflow Plan 4

4.1 Branches 4

4.2 Issues 4

4.3 Labels 4

5 Proof of Concept Demonstration Plan 4

6 Technology 5

7 Coding Standard 5

7.1 JavaScript 5

7.2 Python 5

8 Project Scheduling 5

8.1 Scheduling 5

8.2 Milestones 6

8.3 Tasks 6

Table 1: Revision History

Date	Developer(s)	Change
09/26/22	David Bednar	Initial Draft
09/26/22	Peter Minbashian	Initial Draft
04/05/23	David Bednar	Revision 1.0

This document outlines our plans for the development of Smart Serve. This includes plans for team meetings and communication, the roles of team members, a workflow plan, a plan for the proof of concept demo, technology that will be used, coding standards, and project scheduling.

1 Team Meeting Plan

A weekly team meeting will take place at 1:30 pm on Wednesdays either in person or online. The leader of the meeting will rotate each week, and will follow the weekly agenda. All team members have access to the agenda in google docs and can add topics that need to be discussed. The meeting will last at least an hour but may go overtime if all members are available and if the meeting needs more time. The leader of the meeting will assess the agenda and time constraints, and set time limits so that each topic gets adequate time. At the end of each meeting, all members should have action items that need to be completed.

2 Team Communication Plan

The team will use weekly team meetings to discuss all important topics for the week and will add extra meetings when more time is needed. The team will use a private discord server as a group text channel to bring up ideas and topics, share useful information, and discuss work items. Discord will also be used for group voice chats, screen sharing, and one on one discussions.

No team member can fall in love with another team member. It is forbidden!

3 Team Member Roles

The team is currently split into hardware and software members because the project utilizes both components. These roles are not firm, software members will work on hardware components at certain parts of the project and vice versa. Members will be flexible and ready to change roles as needed.

Member	Role	Responsibility
David	Software	Development, Testing
Peter	Software	Development, Testing
Max	Hardware	Design, Development, Testing
Sam	Hardware	Design, Development, Testing
Ryan	Hardware	Design, Development, Testing

4 Workflow Plan

4.1 Branches

GitHub will be the tool used for sharing and submitting code within the group. With the group being split into two sub-teams (Hardware and Software) two branches will be made off of the Main Branch, these will be the sub-team branches. The two teams will then make branches off those two branches to develop any code, appropriately titling these branches based on the spirit of code (i.e. feature, bug fixes, etc.), making Pull Requests (PRs) to the sub-team branches, and tagging all members of the sub-teams in these PRs. Once any changes are ready to be made to the Main Branch, a PR will be made again but waiting for the approval of at least two group members.

4.2 Issues

The Issues feature on GitHub will be used throughout the project to report any bugs. Issues that are detected, will be given to the person responsible for developing said code.

4.3 Labels

Labels will be used to communicate any concerns with code written by other team members. Labels should follow the default naming convention:

<https://docs.github.com/en/issues/using-labels-and-milestones-to-track-work/managing-labels>

Additionally, all labels must have a detailed description regarding the spirit of the label.

5 Proof of Concept Demonstration Plan

There have been three main risks identified for the success of the project at various stages. The first of which is portability. The project will most likely be the size of a coffee table and will be hard to transport. We will be able to overcome this risk by making the project modular and easy to assemble and dismantle. The next risk is using electronic components around large amounts of fluid. This risk will be mitigated by securing all wires in protective coating and separating the electronic components from the fluids in the design. A testing risk is using real drinks in testing. Real alcohol and soda can be expensive,

sticky, hard to clean, and unnecessary. We will overcome this by testing mainly with water and food coloring.

6 Technology

- **Languages:** JavaScript/Typescript, Python, HTML, CSS
- **IDE:** VSCode
- **Unit Testing:** Selenium (JavaScript) and Pytest (Python)
- **Testing:** Measuring and monitoring the output of fluids as well as User and Stress testing the Web Application.
- **Continuous Integration:** CI will not be used because of the complexity of the system and potential conflicts with hardware integration.
- **Document Generation:** Commenting and naming conventions for JS/TSX, and python code.
- **JS Libraries:** ReactJS, Mantine, express, Websockets, ReactApexChart, axios
- **Python Libraries:** RPi (rasberry pi), threading, json
- **Hardware:** Perisaltic Pumps and peripherals, Raspberry Pi and peripherals, Batteries, Wood, Wiring
- **Build Tools:** Carpentering Tools, Soldering Tools

7 Coding Standard

7.1 JavaScript

For JavaScript, the Google JavaScript Style guide will be standard for the project. **Linters:** standardjs

7.2 Python

For Python, the PEP8 guide will be standard for the project. **Linters:** flake8

8 Project Scheduling

8.1 Scheduling

Working hours throughout the week will be decided upon at each weekly meeting. A total of 9 hours will be scheduled for each week where members must be online and working on the project. Meeting progress will be tracked on discord.

8.2 Milestones

Major General Milestones will adhere to the following schedule:

Requirements Document Revision 0: **October 5**
Hazard Analysis 0: **October 19**
VV Plan Revision 0: **November 2**
Proof of Concept Demonstration: **November 14-25**
Design Document Revision 0: **January 18**
Revision 0 Demonstration: **February 6-17**
VV Report Revision 0: **March 8**
Final Demonstration (Revision 1): **March 20 31**
Final Documentation (Revision 1): **April 5**
Expo Demonstration (Revision 1): **April 11**

All specific milestones tailored towards the design and implementation of the project will be finalized when the project design is agreed upon. Milestones will be documented in LaTeX format on overleaf, which will be a live document. These documents will be finalized and uploaded to the Github repository.

8.3 Tasks

Tasks will be identified at each weekly meeting. Tasks will be divided up evenly amongst the sub-teams and each member has the week to complete it, having the completed task pushed to GitHub before the upcoming meeting.