Development Plan: SE 4G06, TRON 4TB6

Team 26, STRONE
Jordan Bierbrier
Azriel Gingoyon
Taranjit Lotey
Udeep Shah
Abraham Taha

April 5, 2023

Revision History

Date	Version	Notes
9/26/2022	1.0	Initial commit
9/26/2022	1.1	Updated Workflow Plan Section
9/26/2022	1.2	Updated Major Milestones
9/26/2022	1.3	Updated Technology Section
4/4/2023	1.4	Updated formatting and revised POC content

Contents

List of Figures

List of Tables

Symbols, Abbreviations and Acronyms

symbol	description
Age groups	(15-30, 31-50, 51-75, 75+)

1 Team Meeting Plan

The team will meet in person on Wednesdays at 4:30pm on a weekly basis. These meetings will typically be done in person at Thode Library or in the capstone project room. However, additional meetings or daily checkups will be scheduled on a per-need basis depending on the status of the project and upcoming deliverables. Additional meetings may either be held in person or virtually through the use of the team's Discord server.

2 Team Communication Plan

The team will have two primary sources of communication which include a Discord server as well as a Facebook messenger group chat. The Discord server is used for hosting virtual meetings as well as sharing files and useful links. Communication will also occur through commit messages and issues when collaborating on deliverables through GitHub. The purpose of using GitHub will also be for team collaboration as well as changing history and reverting opportunities.

3 Team Member Roles

Although we are split up into different roles, which are based on the workload and our skills, we will work closely with one another and review each other's work thoroughly. It is important to communicate with other team members as all of our modules will be linked together. Additionally, roles can change depending on the amount of work and time required.

Jordan Bierbrier

Embedded Systems Developer Lead - The role of this position is to create the software for the microcontroller.

Azriel Gingoyon

Hardware Developer - The focus of this position is to handle and design all the hardware components for the project.

Taranjit Lotey

Application Developer - The role of this position is to assist in the creation of the application for the end user.

Udeep Shah

Embedded Systems Developer - The focus of this position is to create the software for the microcontroller.

Abraham Taha

Application Developer Lead - The focus of this position is to create the application for the end user.

4 Workflow Plan

- Git will be used for collaborating, sharing, tracking and saving our work (code and documents). We will include a main branch, which will consist of the most updated working code. The main branch will connect sub-branches of various hardware components, ideally, each hardware component will have its own sub-branch which will be connected via code in the development branch.

- Branches will be used when editing or adding code and documents to the repository. Branches will follow a naming convention for consistency and coherency. They will be created based on issues, for example, IssueNumber-Description-Branch. As a result, this allows easy branch linking. Additionally, comments or questions found within an issue can easily be traced to a particular branch.

- When a group member wants to merge a branch, a pull request must be created. Pull requests will consist of a comprehensive description of the incoming changes, and additional comments, such as tests conducted.

- We will also be using a pre-product branch giving the capability to test the functionality of referencing branches within each other before making a pull request into main.

- There will be a continuous integration (CI) test to see if the incoming changes are acceptable, by passing a linter check, for example. Furthermore, reviewers will be added to assess the changes and leave any comments which will be documented to review and store thought processes at that current stage of development. Once the CI test passes and the reviewers are satisfied with the changes, the reviewers can accept the changes and the branch can be merged into main.

- We will track issues within GitHub using the Kanban board, which will reference the issue to a designated branch during the pull request. We can tag issues by priority (low, medium, high), by milestone (hardware, application, firmware, etc) and labels. Issues can be assigned to designated group members, and have reviewers to check up on status and work.

5 Proof of Concept Demonstration Plan

The major areas where the project could face issues include:

- **Sound recognition** - We might not be able to isolate or classify signals accurately. This is especially hard when there is a lot of background noise. This is the hardest part of our project where we are likely to face issues or make certain compromises on. Our POC will focus mainly on achieving or testing the feasibility of different sound recognition methods.

- **Integration of hardware onto a small footprint** - This is the biggest risk with our project. We might not be able to fit all the components i.e the microcontroller, motor, microphone and battery in the small footprint that we desire. Though given the limited resources and our team's tight budget, this will be lightly considered. The reason why this is hard to tackle is that the smaller we go with our hardware, especially microcontroller and battery, the more cautious we have to be with power consumption and resource allocation.

- **Bluetooth communication** - There is a risk that we might not be able to implement the Bluetooth communication, especially with the amount of data we are sending.

- **User Application** - There is a risk that the application developed will not be cohesive with the functionality of the device.

The risks associated with our project may change as our development is still ongoing. Our proof of concept will try to individually implement a working example of the above features. The main goal of the POC will be to extract features from an audio file. The other part will be to develop a classification model based on the extracted features of a sample set of audio files. We will also test out the sending and receiving of data using Bluetooth. Also, we will be creating a preliminary user application to showcase future features and expected useability.

6   Technology

- **Programming Languages** - C will be used for the embedded software on the microcontroller for the signal processing. A potential drawback of using C is that the libraries used for classification can be quite low level, making it difficult to integrate within our code. Although, by using C, we can assure that our code will be more efficient since it is programmed at a lower level. Additionally, Java will be used to create the application for the end user. It will be easy for our team members to review the C and Java code, since their syntax is quite similar.

- **Integrated Development Environment (IDE)** - Visual Studio Code will be used for software development, and primarily for the application. A benefit of using Visual Studio Code is that it includes several extensions, such as linters and git tracking. ~~To add, Eclipse will be used for embedded programming.~~ Also, Android studio will be used for development of the android application. A potential issue will be using different IDEs for software, although they are two separate modules.

- **Linters** - Linters will be used to keep the code format consistent throughout our various modules. For C, we will use the Clang Format linter. For Java, we will use the Checkstyle linter.

- **CI** - ~~CI will be included within our project. We will apply the linter during the CI to see if there are any issues with the format or obvious errors. Additionally, the C code can be compiled and built during the~~

~~CI. The latter CI step may be redundant, as you can build the C code locally, although this will be determined after it is tested.~~

- **Hardware** - Hardware that will be used include a microcontroller, analog-to-digital converter, microphone, motor, and a battery. The risks with the hardware components are finding small enough components that meet our goals. Additionally, we need to find a microcontroller with high enough processing power.

- **Testing** - Unit tests will be conducted on the modules that are added to the project. Unit tests will be done in both C and in Java.

7 Coding Standard

- Leave concise and appropriate comments.

- Name variables in Camel case.

- Implement exception handling wherever possible.

- Do not use the same identifier for multiple purposes.

- Use proper indentation.

- Conform similar pieces of code into a function.

- Test the code whenever possible.

- Avoid lengthy lines of code.

8 Project Scheduling

The project will mainly be scheduled according to deadlines where a lot of effort will be made to ensure that members will have enough time to complete tasks. In terms of project scheduling software, we will be using Google Calendar. The calendar will be populated with all of the project deadlines, as found in the course outline, and shared with the team members. When decomposing large tasks into smaller ones, the team will try to meet up in-person or virtually to first analyze the task, discuss our thoughts/findings,

and then have a vote to try to account for everyone's opinions as much as possible. During the analyzes, we will create a list of the overall goal of the task and subgoals that can be summed up to achieve the overall goal. For deciding who does what, the team will try to meet up in-person or virtually to discuss firstly who wants to do what while also keeping in mind the skills/expertise/knowledge of each individual. After taking all of this into consideration, we will discuss, have a vote, and allocate the tasks where splitting of tasks can be considered if need be. Furthermore, different sizes of the tasks will also be taken into consideration and final decisions on task allocation will also be dependent on equal shares of the work being completed by each team member. In terms of our people resources, we believe that the most effective and realistic way to use our people is to assign them tasks that best suit their skills while also maintaining motivation by allowing team members to try tasks that can expand their skill sets.

Major Milestones

- Researching required hardware components for the device.

  - This encompasses finding components such as a vibration motor that fits under certain size constraints, a microcontroller that can support a sound recognition implementation, a bluetooth module for communication with software and a microphone for taking signal inputs.

- Purchasing the hardware components.

  - This is important for having the necessary time to get a working prototype of the device and get an understanding of our hardware.

- Creating a communication path between the hardware and the software.

  - Essential for determining which user chosen inputs our device will create outputs for.

- Creating a PoC that provides a working model of the core requirements.

  - This will demonstrate that the core requirements of our project are attainable before creating a finalized version of the device.

6

- Implementing a graphical user interface that can be used to configure the hardware by individual users.

    - This will give the users of the product a path to customize the device to their preferred use case without the need for a technical background.