CS 461 / 462 / 463

Client Requirements Document Outline

**Bluetooth HW Control Environment**

**Team Name:** APE Industries *“we don’t monkey around”*

**Team Members:** Sean Penney ([penneys@onid.oregonstate.edu](mailto:penenys@onid.oregonstate.edu)), Aaron Egger ([eggera@onid.oregonstate.edu](mailto:eggera@onid.oregonstate.edu)), Paul Atkinson ([atkinspa@onid.oregonstate.edu](mailto:atkinspa@onid.oregonstate.edu))

**Client/Sponsor/Mentor:** Don Heer, School of EECS , 541-737-2978, heer@eecs.oregonstate.edu

**Introduction to the problem:**

The client wants a toolbox that will allow people with limited knowledge to control bluetooth enabled hardware. This app is mainly geared towards people with electronic knowledge, so they can interface with hardware without having to program it themselves. There isn’t a base for this project, so we will be implementing it “from scratch”.

**Project Description:**

We will be designing an app that can control bluetooth devices. This app will have “modules” that control the devices it is connected to. So for example, the module for a bluetooth enabled lamp would have an on and off switch, which would be used to turn the light on and off. There would be a few modules installed by default, but adding support for every bluetooth device that exists would be impossible. So we will have a website set up where people can download external modules made by other people. So the final project will be a success if it is able to connect and control a set of bluetooth device and has an online portion where people can download external modules.

**Requirements:**

**Android GUI stuff**

1. Modules will support setting a timer on the bluetooth device.
2. Modules will support setting the bluetooth device to a certain percentage “i.e. a lamp with a dimming feature.”
3. Gui capable of adding modules.
4. Gui capable of removing old modules.
5. We will use snap to grid functionally for the drag and drop interface.
6. The modules will have a disconnect button so it doesn’t unnecessarily drain the battery.
7. The app will be have layouts for both phones and tablets.

**Website Stuff**

1. A user will be able to upload code for new modules to the website.
2. A user will be able to download new modules to be added to the app via the website.
3. Website will use git as the backend.
4. Database where new modules can be added to the source.
5. Modules stored in the database will have specific tags defining what the module does.
6. Website page where modules can be reviewed.

**Android Bluetooth stuff**

1. App capable of connecting to at least two bluetooth devices at a time.
2. App will be capable of showing incoming data from bluetooth device.
3. App will be capable of sending data to the bluetooth device.
4. App and bluetooth control hardware will be capable of using bluetooth low energy.
5. The app will have the ability to query nearby bluetooth devices and be able to connect to said devices (as opposed to the user going and connecting to devices through the phone settings).
6. App is capable of supporting profiles for different bluetooth devices.

**Other**

1. App will be implemented with Java for android devices.
2. The app will be on the google play store.
3. There will be ECE hardware in addition to other test devices (bluetooth keyboard, bluetooth gamepad, and an arduino with a BLE shield) that the app will natively be able to control.
4. The arduino board will be programmed to support the GATT profile.
5. There will be skeleton code available as a framework for those who want to write new modules (and then upload those modules to the website).

**Versions:**

**Version 0.0.0.0**

Will be setting up the project on android studio and compiling the empty project on

a device.

**Version 0.0.0.1**

Will be able to connect to a bluetooth classic device and display a message when

connected

**Version 0.0.0.2**

Will be able to connect to a bluetooth low energy device and display a message

when connected

**Version 0.0.0.3**

Will be able to receive input from the bluetooth keyboard and display.

**Version 0.0.0.4**

Will be able to receive input from BLE arduino and display it.

**Version 0.0.0.5**

Will plan bluetooth profiles to use with ECE device.

**Version 0.0.0.6**

Will have module capable of sending and receiving info from the bluetooth

device

**Version 0.0.0.7**

Will create module framework

**Version 0.0.0.8**

Create module for Arduino BLE, gaming controller, and keyboard

**Version 0.0.0.9**

Create github repo for modules

**Version 0.0.1.0**

Preliminary website ui and backend.

**Version 0.0.1.1**

Integrate website ui and backend

**Specific tasks to be undertaken:**

**Setup project**

* Install Android Studio.
* Write code to communicate with bluetooth classic device.
* Write code to communicate with bluetooth low energy device.
* Setup Arduino with bluetooth low energy shield to communicate with Android device.

**Setup module framework**

* Create skeleton code that other users can code using.
* Use the skeleton code to create our own modules to test the framework/skeleton code.

**Setup website**

* Set up a Github repo that contains the skeleton code and instructions to import module.
* Create UI for the website.
* Create the backend of the website that uses Github.

**Risk Assessment:**

**Problem one:** setting up the plugin framework.

**Solution one:** If we are unable to integrate enough plugins into our app, we will have to include all modules as part of the latest .apk and users will sort through what modules they want on their phones.

**Problem two:** BLE is not good enough for our app.

**Solution two:** If bluetooth low energy does not provide adequate control over hardware we will use Bluetooth classic.

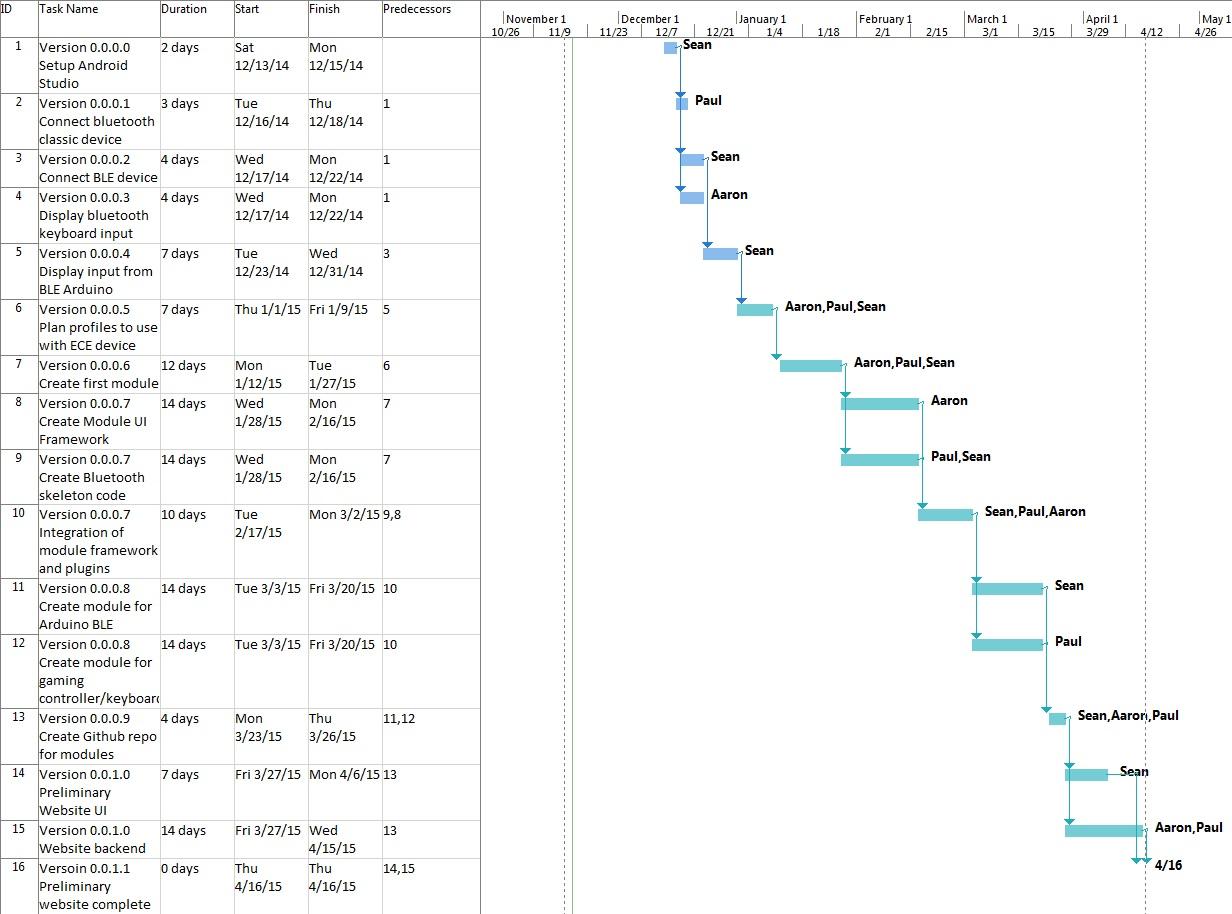
**Testing:**

We will use test driven development while integrating new features and improvements.

We will test our app on a Moto X and a Nexus 7 as well as a GenyMotion emulator.

We will use the uiautomater library to create user interface tests.

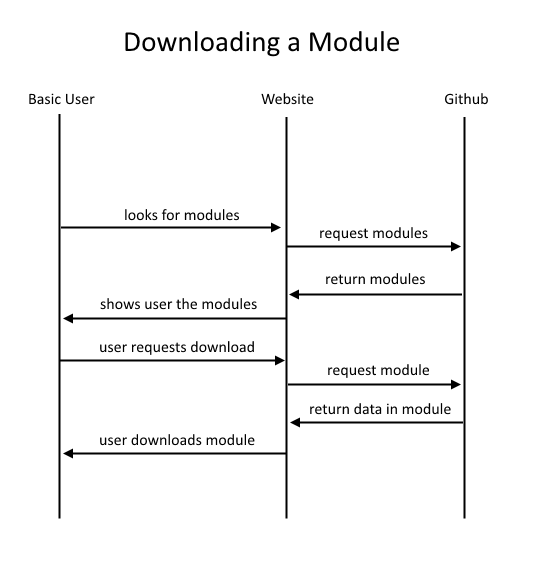
**Preliminary Timetable:**

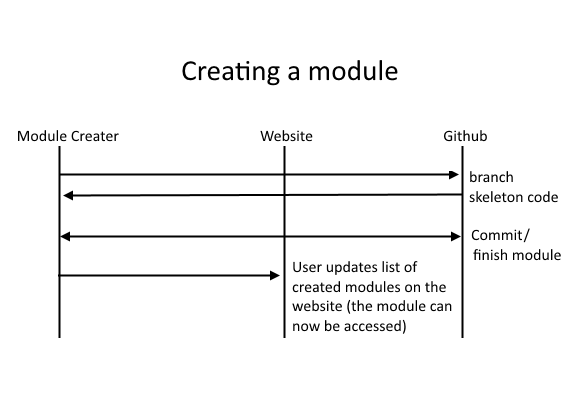


**Integration Plan:**

We plan on having a Github repo that we will all commit to. We will also make sure that everything we write will all be integrated together at least every two weeks (see the Gantt chart for more detail).

**Dataflow sequence diagram:**





**User interface requirements:**

Users can drag and drop modules that control different devices.

There will be a display of data coming back from the bluetooth device

**References:**

https://developer.android.com/guide/topics/connectivity/bluetooth-le.html

**Glossary:**

A bluetooth profile is a specification describing the communication between bluetooth devices.

Generic Attribute Profile (GATT) is a general specification for sending and receiving pieces of data. All low energy profiles are based on GATT.

Github is a website that people can commit to to upload and store their code. The advantage of using Github is you have everything backed up and there is a base code that everyone modifies.

**Signatures:**

**Team Members**

Paul Atkinson: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sean Penney: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aaron Egger: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Sponser**

Don Heer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_