Aaron Egger

Paul Atkinson

Sean Penney

Team 24, [Bluetooth HW Control Environment](http://eecs.oregonstate.edu/capstone/capstone.cgi?project=121)

Tech Review

The goal of this project is to be able to control different objects in a house from a phone without having prior programing knowledge. The phone should act like a universal remote for bluetooth devices. This application should be able to interface with *any* bluetooth enabled device: whether or not the device is sending or receiving data. If the hardware that I am trying to control already has bluetooth functionality I want to simply be able to connect and control the hardware. If the hardware does not have bluetooth functionality I want to be able to simply hook up “the ECE module” to easily control and interface with the hardware. Through the drag and drop interface I want to be able to change the functionality of my application. For example, I have a bluetooth enabled lamp: I would like to drag and drop an on/off switch to control the lamp. If this lamp has a dimming function I would like to drag and drop a module to control that functionality.

# Possible Technologies

**Integrated Development Environment:**

**Android Studio:**

This is the IDE that Google uses to develop applications. Looking online, it is by far the most recommended IDE. The downside of using Android Studio is it is still in beta, so there may be some bugs associated with it. This will be the IDE that we will use for this project.

**Eclipse with ADT plugin:**

Eclipse is the old hat in the android development environment. It has many quirks and kinks that make using it an experience most unpleasant. However, Eclipse might be considered the more stable contender. Ultimately all three of us have used eclipse before and have found it to be frustrating so we would like to consider using Android Studio and if we have problems with it then we can fall back onto Eclipse.

**Netbeans:**

It is an IDE that many people prefer over Eclipse.

**Bluetooth communication:**

**Bluetooth BLE:**

The low energy variant of bluetooth is prefered because it is able to power on and

off at will. However, the new tech might not be fully stabilized in the android OS.

**Bluetooth Classic (SPP profile):**

Bluetooth classic is the better choice in more power intensive work. Also it might be

more stable considering it has been out for so much longer.

**Bluetooth Classic (other profile):**

There are more modern profiles, however, we would need to develop our own

bluetooth profile. According to the android developer's website most developers choose to use the build in Android Socket (SPP).

**Backend of Website :**

**GIT:**

Github is the community standard for source control. Since we want users to be

able to easily share their new code with other users using something that everyone

already has makes the most sense. People already use Github for everything.

**SVN:**

Users not being able to easily add and edit with github is kind of a deal breaker with

this project.

**Archived Bzipped Tarballs:**

This would be primitive.

**Our selections:**

**IDE:** Android Studio

**Bluetooth Communications:** Bluetooth Classic (SPP profile)

**Website backend:** Git