The aim for our project was to create an application that records a voice command using the built in voice recognition in android, and then store that command in a database. The command that have just been stored will then be return to you as text on the screen. We use led-lights to simulate something that could happen when you give the application an input like turning on your TV and so on.

Here are our research questions that we had for this project.

**Research questions**

 How can we make voice recognition a more reliable input?

 How can voice recognition be used to improve the quality of life?

Arduino

The arduino is connected to the Bluetooth via 2 cables rx and tx for transmitting and receiving data. And uses the ports 2-6 for the led-lights that we use to simulate something real.

The Arduino receives an ID from the Bluetooth module and with that ID the Arduino board knows which led to light up, this code is written in arduino playground.

The expected results for this project were to make a program that would be able to record your voice and do something with it. Like a mentioned earlier simulate something in the real world and for now that is lighting led-lights and writing the command you just said as a text that you can read to see if it understood what you said.

We have been working according to XP for the most past of this project although we have not been that Strict when it came to writing story cards. Pair programming were used a lot, and we used black box white box testing for when developing the application.

Github was our revision control system of choice.

The problems we had going through with this project were learning all the new software programs like android and arduino playground.

The implementation of SQlite was also a hand full.

But the most demanding part of our project was the Bluetooth connection between the arduino and the android application. It took us more time than I like to admit to make it work.

The problems we had were solved by implementing code from tutorials and taking bits of code from already completed examples. Also pair-programming helped solve these problems because two heads are better than one.

On a side note our average velocity were 0.8 with 259 active work hours.