Milestones - Yes we’ve done everything

Changes to the system from now to the final release?

We’d need to redo the server (look at below point) and we’d need to think about how to scale the custom line sensor - use underground tape or cheap tape.

Are multiple robots needed and is there added complexity for this?

Yes we are expecting museums to want multiple robots per museum. The obstacle avoidance feature is very good because then all we’d just need is to decide which robot obstacle avoids and which one stands still and they won’t interfere with each other. We’d probably need to rework the server though. Firstly it’d be better if we used our own server and not a cheap web hosting service and we’d need to rework the logic because the server was originally intended to only provide communication between 1 robot and 1 app and with multiple robots and multiple apps we’d need to redo how our server works.

Limitations.

Area is small, can’t track location reliably so the obstacle avoidance is limited to immobile objects in certain areas.

Other applications.

Anywhere with a large indoor environment where people regularly go and don’t know the layout and speak multiple languages - hotels, airports, trains, shopping malls, expos and fairs.

Modularity of the System

Yes a lot of the features could be reused. If you wanted text-to-speech or speech-to-text in the app you could reuse the code a custom line and ultrasound sensor reuse Michals code, obstacle avoidance, reuse the code etc. Most features that have been implemented could be reused

FSM framework xD

Deviations from the start

We implemented everything we said we would. Added multi phones and custom. At the start we overestimated the power of the ev3 brick but we got around it.

Investor Demo

We’re gonna focus more on marketing of the robot and use cases.

How was the experience?

A lot more work than we thought, ev3 power caused a lot of headaches.

Most happy with the system.

I think how the app turned out, the speech to text I’m really happy with, also the custom sensor