

Summer project: Updating an Android app

Regina Hansson, 2019

Introduction

During this summer I worked on a project at Senion, a company whose indoor positioning product is relying on sensors and sensor fusion. They use an app to install and configure their products, but the Android version had become outdated and did not reach the iOS app's level regarding usability and functionality. It was my task to update the graphical as well as logical user interface.

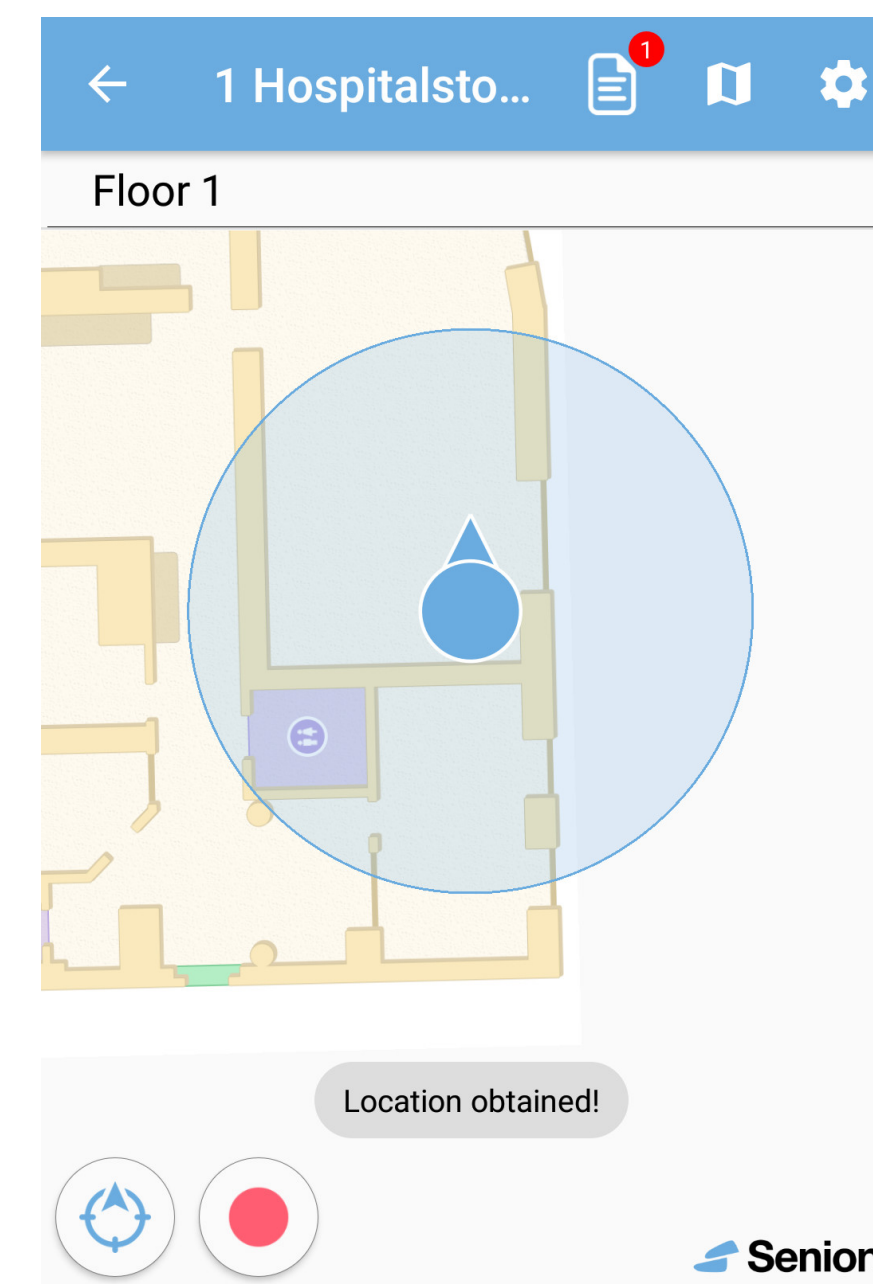


Figure 1: Using the app for positioning

Problems encountered

My first task - to update the graphical user interface of the main menu in the app - turned out not being trivial at all. The structure of the code did not permit the user flow that was desired, so I had to read and understand a lot of code *not* related to the graphics in order to even get started.

I also tried to add a redo functionality in one of the app views. When manually testing my initial implementation of the redo options, everything seemed to work fine. Though, redoing more complex actions did not work at all, and after a lot of reading in the existing code I had to realise this was a too intricate problem for me to solve. The large inheritance depth in the structure and untrivial command patterns would take too long to understand and interact with - which was why the redo functionality wasn't implemented in the first place.

When testing the app (e.g. as in figure 1) after making my updates, I couldn't get the sensors to work properly. After verification from my co-workers, it turned out it probably wasn't the app that misbehaved - it was me. I learned that the interactions between the sensors and humans were sensitive in ways I had not thought of, and it opened my eyes to the complexity of doing work depending on sensor information.

Of course I also encountered problems with git (the version handling tool). Some of my work disappeared due to clumsy handling, and when I tried to merge my work with the existing code I had to realise that it demanded a great deal of cautiousness.

Results

The project succeeded despite the problems mentioned.

I managed to restructure the main menu interface (as seen in figure 2) and its user flow. Furthermore I implemented an awaited toggle functionality and a performance-enhancing download-on-demand, replacing the old automatic download. I also did some of the graphical design.

In collaboration with my teammates the work I had done was tested on several levels, verified and released.

On a more personal level I have learned much about the system and product chain. I have begun understanding how to contribute in a large project and how to collaborate as a team. Of course I've gained a lot of technical knowledge, even regarding things I did not work on myself, thanks to e.g. discussions in the office. Last but not least, I've come to realise how much more than automatic control is needed to build a successful product...

Conclusions

To conclude, these are some insights I bring from the summer:

- Collaboration with teammates is essential
- In a small team your tasks can be very varying
- Skills development is a continuous, always ongoing process
- You really do learn from mistakes
- Testing takes time
- Version handling doesn't have to be difficult

Acknowledgements

Special thanks to Senion and my coach Mats Selander.

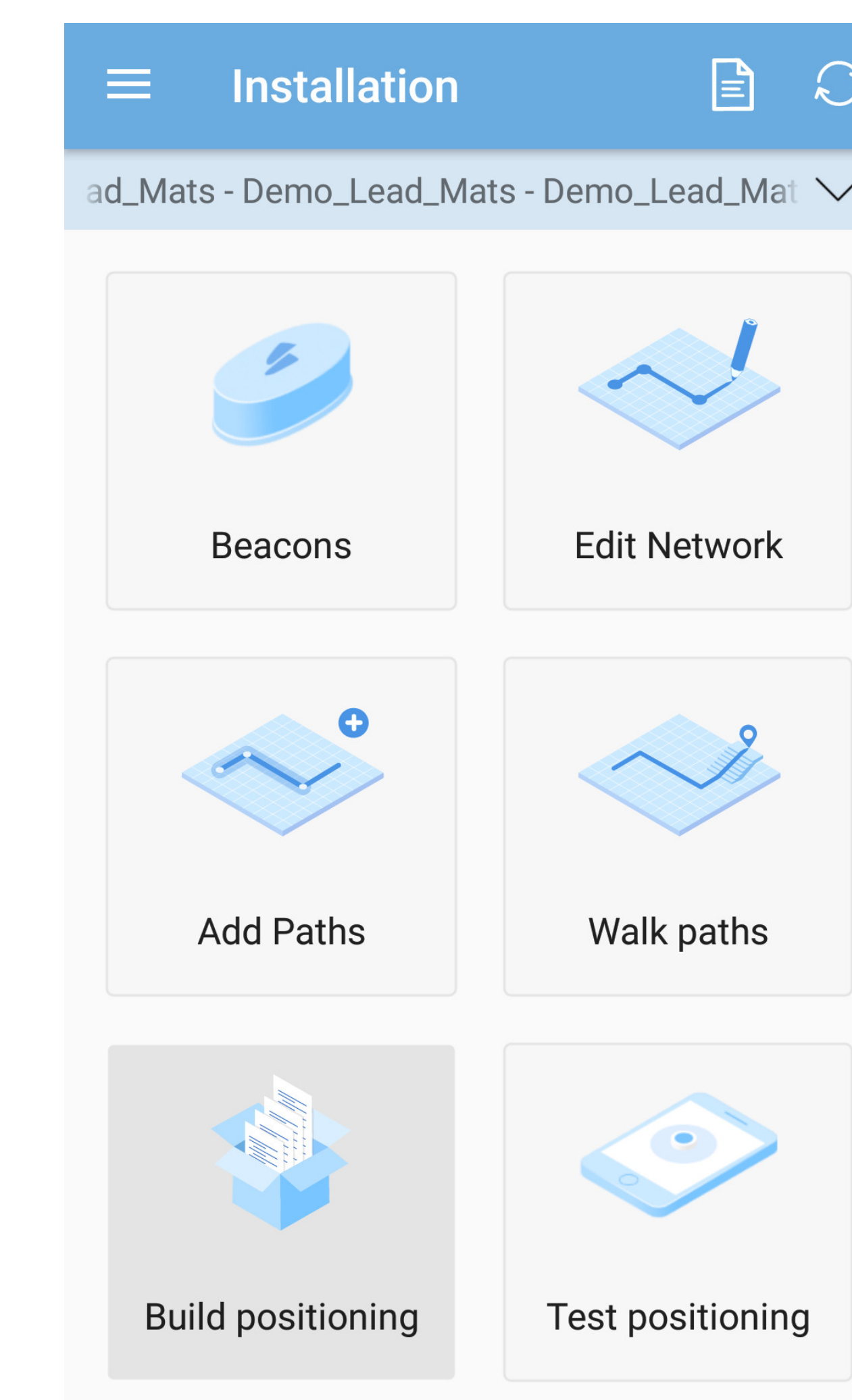


Figure 2: The resulting main menu