

Notes meeting Customer 23-2-2018

People present:

Lars (customer), Frank (second customer), (both referred to as customer below) Andrei, Zino, Marc

Summary:

- Customer wants voice control capability of the app
- Customer wants a nice iOS look and feel design of the app.
- Functionality is not important first, but installing devices is primary

Extended version:

Last year the customer worked on a Hestia Server

(<https://github.com/RUGSoftEng/2017-Hestia-Server>) and Hestia Android app (<https://github.com/RUGSoftEng/2017-Hestia-Client>). (They stopped develop it after the course ended.) This is on GitHub with extended documentation on how it works and can be set up.

The idea is to connect a peripheral to the server and the iOS device is also connected to the server to control the peripheral. The server can run on a Raspberry Pi, but you should also be able to test it on a PC (see documentation). The customer had it running on its MacBook during the meeting.

The system works with plugins for peripherals to control the functionality. We do not have to do something with this, but the web development team could build a marketplace for plugins. (IKEA lighting)

The customer would like to see voice control (Siri) ability in the app. Also he wants a nice native iOS look and feel of the app. The customer finds this really important.

The app should be able to install a device. (The plugins contains different kinds of Philips Hue lights and a mock lock (see documentation)).

The customer is not really happy with the current Android app design, so that should not be mimicked.

The app can connect to the server using a RESTful API. (The server is build with Flask, but this is not really essential for the iOS team.) This API can be called to install plugins on the app. So you asks the server to search for a device, all the information is stored on the server.

The old Android app can be downloaded from GitHub and tested on Android phones (maybe on PC with an emulator (it was developed with Android Studio)).

If the app is going to be developed in a cross-platform framework (Xamarin), the customer thinks it is nice if there is a shared code base for the back-end of the app to support

cross-platform functionality. Then separate iOS (and Android) layers can be added for the User interface.

The functionality is less important to the customer for now, but installing devices has priority regarding functionality.

The customer would like it if we use TravisCI (continuous integration testing). He also suggested CodeCov. The customer wants clean code. (See also the recommended book)

The customer also suggested using a project board (Jira, trello, github project board) and would also like to be added to that.

In the current setup, the Server is a local server. In the future it would be a nice extension if the iOS client could connect to a webserver that connects to the local server. (Then you can safely turn off lights from outside the house). (This is also something that should be discussed with the web team.)

The customer thinks it is difficult to secure the connections, but he asks us to at least have look at it and communicate if high security is possible or not.

Zino came up with the idea of third party authentication (Facebook/ Google).

We decided that we will plan a new meeting with the customer in about 2 ½ weeks. (Yet to be arranged). The customer would appreciate a separate slack channel to communicate with him.

Side note: We discussed that time learning about development frameworks can also be logged in the time-tracking documents.

The customers likes the architecture document to be concise and visual.

The customer would like to see different people at meetings with him, it does not matter how many people show up.

As a useful extension, we could define more activators for devices. Now only two activators are defined ((switch (bool) and slider (float))). This has to be implemented in the Server code via the customer.

The customer suggested Adobe XD for design, but Andrei also knows a tool (he has to look up the name).