happy flowers

Project Agreement

Sacha Schmid
Rinesch Murugathas

Daniel Kröni

Windisch, 4 October 2016

Table of Contents

1. Basic Conditions

1.1. Involved Persons

Customer	Institute for Mobile und Distributed Systems
Coach	Daniel Kröni
Project Team	Sacha Schmid
	Rinesch Murugathas

1.2. Dates

Project Start	September 19, 2016
Kickoff Meeting	September 23, 2016
Project Presentation	tbd
Project End	January 20, 2017

2. Project description

The flowers at the Institute for Mobile and Distributed Systems are fighting for long dry periods and this project intends to solve that issue by watering the flowers automatically. The application shall be written in Haskell with a low level interface to hardware & database connection and a high level web interface.

The final product should run on a Raspberry Pi 3 and the web interface should show the current moisture of the earth as measured by a sensor. It should also show statistics of the moisture of the plant from the last two days as well as the latest water pump times. There should be also a livestream on the web interface which shows a live image of the plant. The water pump should also be able to be activated through the web interface.

Another requirement is that this application will be used for the fprog lecture to show that one can also solve practical problems with Haskell. Because of that the quality of the work has to be high, meaning the understanding, maintainability and documentation of the code have a high priority.

3. Requirements

3.1. Web

- Show the statistics of the moisture of the flower from the last month as a graph. Showing a historical view of another period is also an option. A zooming mechanic allows viewing sections of data in more detail.
- Show events in the history graph (e.g. waterings, both manual and automatic)
- Show the last manual and automatic water pump time.
- Show a livestream of the flower.
- Offer the ability to log in and log out. The login works with just a password that is stored in a configuration file on the RPi. There is no account management.
- Offer a button to start the water pump manually.
- Only allow starting the pump manually when the user is logged in.
- The web interface will be realised in React and Redux. Use Immutable.js to ensure consistent data structures.
- Offer the ability to name a flower.
- Offer a settings page to change settings like upper and lower limits for moisture and measurement intervals.
- Starting the water pump manually must only allow it to run until stopped manually or until the upper limit of moisture is reached.

3.2. Raspberry Pi (RPi)

- Run the web front end through a web server.

- The web server offers a REST-like API that allows other parties to be informed about possible updates.
- Run a web socket server to enable live communication between the web front end and the RPi.
- Use Stack as the build and dependency management tool for all Haskell code. Alternatively, cabal-sandbox can also be used.
- Document code using Haddock.
- Use modularised code to improve maintainability and ease of understanding.
- Connect to a moisture sensor that measures moisture levels of a plant.
- Store historical data about moisture levels in a database.
- Measure the moisture of the plant on an hourly basis.
- Connect to a camera that can be used for live streaming video. Alternatively, static images could be taken at regular intervals.
- The Haskell code will be compiled on the RPi directly.

3.3. Documentation

- Setup guide covering the setup procedure of the RPi device. This can be written in the form of a README inside the repository.
- Project report describing the resulting product, including its architecture, the interactions between components and justified decisions made by the team.

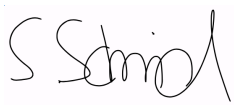
4. Deliverables

- Source Code with documentation
- Project report
- Ready to use RPI

  The source code and project report will be delivered on a USB stick and the project report will be printed as well.

5. Agreement and Signature

Hereby the customer, coach and the project team agree to the terms mentioned in this document and will fulfil their parts accordingly.



Sacha Schmid

Rinesch Murugathas

Daniel Kröni