Daily Scrum Log

Software Engineering Assignment 2

Jörg Striebel

21205773



Software Eng (COMP30830)

UCD School of Computer Science

02/02/2022

Personal Notes

# Learning journal:

## 2022/02/07

Doing research on what tools are available and suitable for our Scrum process management. Although the tool “scrumwise” has been ranked on several scrum websites as one of the best management tools, we decided to give “goodday.work” a shot since this one offers a free version. We decided to create a shared drive on Google Drive to store all documentation.

## 2022/02/08

First lab: Trying to get started with using GitHub, but since none of us has had any prior experience with this tool, it took some time to familiarize ourselves with the basic functionality of this fairly comprehensive tool.

## 2022/02/09

Second lab: We had our first meeting with our product manager Mandy. Although we hadn’t much to report yet, it was good to hear about potential pitfalls along the way such as the need of dropping all security walls on AWS to access the RDS from different locations and different developers - issues with changing IP addresses.

## 2022/02/10

Figuring out how to set up security rules so that the RDS in AWS can be accessed from different locations, say on campus the IP addresses tend to change on a daily basis, whereas at home it tends to remind the same.

## 2022/02/11

Familiarising myself with the Scrum fundamentals: how to write appropriate user stories; what is the difference between epics and user stories; how to create actual tasks out of user stories (e.g. create database to store weather information, etc.). Writing first user stories into the product backlog.

## 2022/02/12

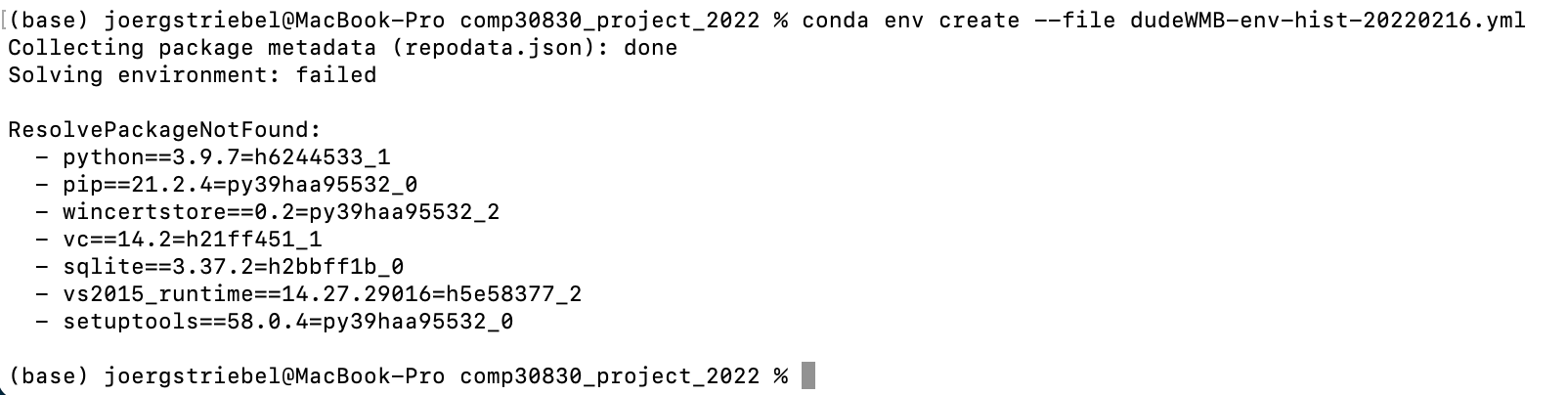
Creating the first draft version of the Software Requirements Specification (SRS). Setting up the basic structure of the SRS.

## 2022/02/14

Enhancing the SRS document by chapter “product perspective” and “product functions”. Additionally, creating a wireframe and incorporating it into the SRS.

## 2022/02/15

Facing issues with importing the virtual anaconda environment on my MacBook. Tom has created and exported a “.yml export file” via anaconda on his Windows machine. Trying to import this environment specification file on my MacBook has failed. Error message “ResovePackageNotFound”, as shown in the screenshot below. We’ve also tried “pip” to export/import virtual environments across different platforms but we didn’t succeed. Addressing this issue to one of the TA’s in the next lab.



## 2022/02/16

Still trying to resolve this issue with the pip and conda packages for the virtual Anaconda environments.

## 2022/02/17

During the lab today, we were discussing how to organise the data within the RDS. Particularly, whether it would make sense to separate the static data from the dynamic data. Both are received as one JSON message via API. We also discussed how the weather data could correlate with the dynamic data, e.g. using timestamps, etc.

After the lab, our TA, Karl Roe, was so kind helping me to sort out my issue with importing/exporting the Anaconda virtual environment and getting it running across Windows/MacOs platforms. Unfortunately, we weren’t able to import the virtual environment via yml or pip files, so we came to the conclusion not to investigate this issue any further since we had already spent over 8 hours on trying to get it up and running. The workaround will be creating my own independent development environment on my MacBook.

## 2022/02/18

The decision has been made; from now on I’ll be keeping track separately of my virtual Anaconda environment. Starting to set up my own environment. Also, I’ve started planning how to implement changes in the dataloader.py

## 2022/02/21

Integrating station-ID into the data\_loader.py; is the primary key of the station table and is created by utilising the auto increment option. This Station-ID is used to uniquely identify a station within the database, since the station number which is returned by the JCDecaux is only unique in combination with the contractName.

## 2022/02/22

Additional function “stationExists(..) in data\_loader.py implemented. This function checks if a station is already listed in the station table. It returns True if stations already exist. I couldn’t figure out who to extract number of existing stations from the “connection.execute(....) → workaround: iterating over results and incrementing counter variables. Not the neatest approach but it works.

## 2022/02/23

Gathering ideas to draft the first wireframe. Also, looking for a suitable free software that lets you easily create wireframes and offers templates; Visual-Paradigm seems to be the best and freely available software

**2022/02/24**

Creating the first wireframe draft in Visual-Paradigm. At first glance the tool has a lot to offer but usability isn’t as good as it could be… might just take a while to get used to it.

**2022/02/25**

Picking up where we’ve left off; still creating the first wireframe draft in Visual-Paradigm. The wireframe starts taking on shape…

**2022/02/28**

Looking at ways to implement a first HTML/CSS layout that is compatible with Flask and that can run on an EC2 instance.

**2022/03/01**

It’s been a while since implementing some HTML/CSS, but it’s coming back ;) Splitting up the HTML in a base and then inheriting from this base for all templates seems to be a brilliant idea - works fine!

**2022/03/02**

Now it’s time to put some elements and the Google maps on the page. I’ve faced an issue with the Google API key but after fiddling around for a good while I could fix it.

**2022/03/03**

Further enhancing the index.html using flexboxes to arrange all the elements nicely on the webpage. Also, implementing some JavaScript code to create station selection dynamically.

## 

# Personal review:

## 

# Retrospective of the project: