**Software Engineering Project**

Group Learning Journal

* **Team:**

Seventeen

* **Member:**

15211145 Wei Lan, 15211282 Zeng Chen, 16203136 Chang Wen-Ting

* **Total number of pages:**

11

* **Date:**

Sunday 23th February, 2017

Content

[1 Group Work 1](#_Toc480577094)

[**1.1** **Percentage of Group Submission** 1](#_Toc480577095)

[**1.2** **Skills Practiced** 2](#_Toc480577096)

[**1.3** **Learning Done** 3](#_Toc480577097)

[**(1)** **Technical skills** 3](#_Toc480577098)

[**(2)** **Team work** 4](#_Toc480577099)

[**1.4** **Goals** 5](#_Toc480577100)

[2 Problem encounter 5](#_Toc480577101)

[3 Technical 6](#_Toc480577102)

[**3.1** **Project Architecture** 6](#_Toc480577103)

[**3.2** **Finished material** 6](#_Toc480577104)

1. **Group Work**
   1. **Percentage of Group Submission**

This section briefly describes the tasks distribution and work done during the project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint | Sub-task | Main Owner | Vice owner | Finished % |
| One | Create RDS instance | Chen | Chang | 100% |
| Built database environment | Chen | Chang | 100% |
| Link to database | Chang | Lan | 100% |
| Scrap data from JCDecaux API | Chang | Lan | 100% |
| Constantly run scrapping data process | Lan | Chen | 100% |
| Data maintain | Lan | Chen | 100% |
| Data backup | Lan | Chang | 100% |
| Trello and meeting note | Lan | Chen | 100% |
| Two | Python code for flask app | Chen | Chang, Lan | 100% |
| Python code for query database – station static data | Chang | Chen, Lan | 100% |
| Python code for query database – station dynamic data | Chang | Chen, Lan | 100% |
| Python code for query database – summarized data | Chang | Chen, Lan | 100% |
| Bootstrap template for webpage | Lan | Chen, Chang | 100% |
| Javascript/jquery for show google map | Lan | Chen, Chang | 100% |
| Javascript/jquery for add marker on map | Chen | Lan, Chang | 100% |
| Javascript/jquery for click marker and show station information | Chen | Lan, Chang | 100% |
| Javascript/jquery for click marker and show station information | Chang | Lan, Chen | 100% |
| Trello and meeting note | Lan | Chen | 100% |
| Three | Javascript/jquery for click marker and show statistical chart | Lan | Chang, Chen | 100% |
| Javascript/jquery for show weather information | Chen | Chang, Lan | 100% |
| Data recovery | Chang | Lan, Chen | 100% |
| Flask app run on EC2 instance | Chang | Lan, Chen | 100% |
| Trello and meeting note | Lan | Chang, Chen | 100% |

* 1. **Skills Practiced**
* Amazon Web Service : how to operate cloud service
* Python: utilize python packages (pymysql, flask-sqlalchemy, sqlalchemy, numpy, pandas)
* Linux: use command line to operate EC2 instance
* MySQL: use mysql and MySQLWorkbrench to operate database
* Google map API: use javascript/jquery to access google map API
* Google developers chart: use javascript/jquery to built interactive chart
* Open weather API: use javascript/jquery to query weather forecast/history data
* JSON: query and analysis json file
* Flask: micro web framework to build simple web application
* Jinja2: used to create HTML markup formats that are returned to the client side via an HTTP response.
* JQuery: to simplify JavaScript to of the client-side scripting of HTML
* Javascript: to manipulate data and webpage at client side
* Bootstrapt: framework for faster and easier web development
* Github: version control, commit history, revision record and trace, multiple development.
  1. **Learning Done**

1. **Technical skills**

In first semester, we had learned basic python, javascript and PHP to manipulate webpage and request json data from API or database. In this project, we require to learn quickly on new technique. Under limitation of time and resources, it’s important to collect and comprehend new information then make the decision of entire structure of the project.

In sprint one we learn:

* How to use Amazon web service and use linux command line to make operation on EC2, such as python package management, network connection checking.
* How to use python package to create database, create table, insert data and checking data.
* How to use python built in function pickle to backup data
* How to use python to scrap data from API
* How to use setup.py to pack egg package and install as a custom package
* How to use ‘screen’ command to make program constantly run on EC2 instance.
* How to utilize python traceback and exception to capture error.

In sprint two we learn:

* How to use flask to build web application framework
* How to use jinja2 to capture data throw from flask through router
* How to use sqlalchemy or pymysql to query database
* How to use mysql to query database
* How to use bootstrap template to build webpage
* How to use javascript/jquery to manipulate html

In sprint three we learn:

* How to use google developer API to create interactive chart
* How to use jquery to request open weather API
* How to use EC2 as server for publicly access

1. **Team work**

In these three sprints we try different kind of ways of cooperation mode. In first sprint, it’s more like as the waterfall process model, we split our task into three steps. Each step can start only the previous step is perfect and finished. However, at the end of the sprint we learn that it’s less efficient because we constantly need go back to the previous stage to do some revision. As the beginner of software developer to implement this model will increase the develop time instead. Therefore, in second sprint, we adopt the model is more like spiral model in the first half of sprint and agile process model in the last half of the sprint. That is, we take more time to do some research on whole project structure, which architecture is more efficient and more extendable. Also, since lots of technique is new in this sprint, we spent more time on learning before start. In the second half of sprint, we learn by utilizing user story to select features. At the beginning, it’s easily to idealize the outcome and try to implement all features require or think of. But due to the time and resource restriction, we need to learn how to priorities these features, which is necessary and which can be postponed. Moreover, in this sprint we learn that it is useful and efficient to use test driven development during the coding process. In first sprint we test the code after the whole function is finished, which is a big function that combine all sub-functions together. This make it hard to test by the developer even hard to test by others. Therefore, in this sprint, we cross testing other members’ code by manual and python nose packages every time a small function is finished. This proof to be more efficient than the mode we use in first sprint. In the last sprint, we took some time to tune the whole features we selected at the beginning, which should be removed and which need to be added. During these sprints we basically work together in meeting room, therefore, one thing we forgot to do is to take a good use of github. At the end we found out this make it hard to do the version control and the trace of commit history. When we want to roll back to certain version we prefer, we take longer to redo some code if we forgot to backup the previous version. This is the critical issue we want to improve in the final sprint.

* 1. **Goals**

This assign simulates the real market software development project. We summarize few goals we plan to done in the future:

* Practicing to scrap data from online using API, store into Online.
* Learning Screen how to run in the background.
* Practicing link to SQL to require the specified data
* Learning more skills about JavaScript, Jquery or PHP how to request the data from database to show on web application.
* Practicing Google charts.
* Learning more knowledge about flask.
* Flask is a powerful web template, Learning more flask packages .
* Practice on EC2 run the web application
* Practicing using scrum management to manage the project
* Practicing using the Trello or GoogleDocs to manage the all meeting notes

1. **Problem encounter**

This section by listing down the features we try to implement in each sprint, the problem encountered and how we solve it or discard it.

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Feature | Problem | Solution |
| One | Create RDS and use MySQLWorkbrench/ python link to it | Cannot successfully link to it. | Reason: didn’t open to public access while create RDS. Revised and solved |
|  | Periodically scrap static station information and replace data in database. | No problem but probably unnecessary. | Put to lower priority. Do have code to implement it but didn’t run it during the project. |
|  | Constantly scrap real time data and insert to database | No problem with the code to implement it but due to the rapidly usage of AWS it could reach the limit very soon. | Discard the idea, only run for one week. |
|  |  |  |  |

1. **Technical**
   1. **Project Architecture**
   2. **Finished material**