Imperial College London

Summer Placement Report

Internship at Fleetonomy

Lei Wang - 01214043

Submitted: September 16, 2018

1 Description

The summer placement took place in Tel Aviv, Israel. I worked for a company called Fleetonomy for ten weeks. The internship was mainly about web application development, using Python, HTML, JavaScript and CSS.

Fleetonomy is a company that provides solutions about demand prediction, vehicle assignment, and fleet redestribution to fleet owners, auto manufacturers and mobility service providers to help enhance efficiency, generating more insightsand create new revenue streams [3]. The company mainly focuses on automating and predicting the coordination of car fleet, which is similar to Uber but provides extra functionality of making predictions using artificial intelligence. The working hours usually started between 9 and 10 in the morning and ended between 7 and 8 in the evening, which was quite normal for software companies in Israel. The people I worked with were quite very enthaustic about their jobs that they often pushed codes to the repository very late at night or very early in the morning. It was a privilege to work with such a group of diligent and industrious people.

At the beginning of the work, the supervisor explained to me in details about the structure of the system they have been working on for years. I started by doing relatively simple tasks and moved on to more difficult ones later. The tasks were quite comprehensive, covering a wide range of scenarios. Due to the company's policy, no codes will be listed or discussed in the report. The internship was similar to lab sessions: required some pre-known knowledge and lots of testing and debugging.

It was part of the company's doctrine to review each other's work before putting into any production environment, which I benefited a lot from. Usually my tasks were about adding extra functions to previous work which I needed to go through first. I could not have done this without my colleagues' help. They helped get started with the tools they used in software development like Git, for version control [4], CircleCI, for continuous software development [1], Slack, for team communication [5], etc. Those tools and concepts were very advanced ideas since previously, it was only me alone who was working on an assignment or any personal project. I learnt about effective communication with my coworkers on how to convey ideas and advice accurately without taking much too much time.

Besides work, I had been a tourist during my time in Israel and those places I had been to including Goren Park, Jerusalem, Bethlehem, Masada, En Gedi and the Dead Sea. My colleagues adviced me to go to those places and I hoped to learn about their culture. It was really thrilling to have a break from the heavy work and see all those wonderful places.

2 Tasks

2.1 Vehicle Status Monitoring

The company uses Datadog [2] for monitoring the health status of the system. To use datadog, the program must send real-time data to the datadog's server. Then datadog will then check data according to rules set by the operator. Under this task's context, datadog will trigger an alarm when a reading received from client exceeds a certain limit. My job was to make the program commonicate with datadog by making API calls to the datadog's server. The reading was about how many vehicles in the system that had not been moving for certain amount of time. If at least 20 percent of the total number of vehicles were not moving, the alarm would be triggered by datadog. The operators would then be informed through Slack integration, and via email. The tasks familarized myself with the structure of the whole system and the connection among different parts.

2.2 Display System Error Logs via Slack

Slack is a communication software broadly used in team work projects that supports third party integrations to perform autonomous notifications from other platforms. Datadog, as mentioned previously, has this integration of pushing notifications when an alarm is triggered. The integration shows a link in the Slack channel to the Datadog's web interface, where the operator can view the detailed status of the monitor, while it does not show enough information through the integration directly. The main system sends logs to datadog on real time. When an error is found in the logs by the datadog service, datadog does not send the error logs to Slack due to the minimum frequency that Datadog scans data that may trigger the alarm. The operator needs to go through a series of web pages to finally reach the logs. Also, deveopers want to keep logs for future reference while Datadog only keeps the logs for a few days before logs are deleted.

The solution is to overwrite the Python default logger handler class and write code that interfaces with Slack. Instead of detecting logs indicate errors by datadog, the system detects errors in logs by itself. The logs will also be sent to Amazon Cloudwatch other than Datadog. The logs sent to Amazon Cloudwatch will be stored in Amazon S3 bucket for future reference. After logs that indicate errors being detected by the system, a code snippet will be generated and the system will make a http call to the API provided by Slack called file-uploading. Slack is configured prior to use so that the file-uploading is allowed and the code snippet received will be shown in the correct commonication channel on Slack.

In order to use Amazon Cloudwatch or Slack file-uploading via API calls, access tokens are needed. The service providers, for instance, Amazon Web Services and Slack, will charge the user according to the number of API calls or how much storage those calls take up. The access token is the way to identify the user and hide user's identity over the Internet. As a result, web service providers usually will not allow users to make API calls to their servers without quoting access tokens.

2.3 Add Prebook Function to web GUI

Beginning from this task, the placement shifted from only about backend development using Python towards web frontend development using HTML and JavaScript. The software Fleetonomy is developing includes server applications, web interfaces for system administration, Android and iOS applications for the end users, and testing software for developers, which also a web application. This task was about making changes on the testing software to add more functions.

My job in this task was to add components on the web interface to make API calls to make reservations, which was done by adding drop down menus and buttons then binding JavaScript functions to those components.

2.4 Unit Tests for Prebook Function

Every line of code the developers have written must go through a series of tests before the codes are actually merged into the master branch. The task focused on the testing of the backend code. The unit tests written were for making prebook API calls using sentences in Python such as Assume. The backend is already implemented the ability to detect and throw errors. The unit tests for the prebook API include making calls using the right parameters, and making calls using parameters that should cause errors. The errors should be correctly identified by the program by comparing the error output with the error that is assumed to be thrown out.

The task was not difficult to tackle after my supervisor walked me through the mechanism of using the prebook API. I discussed with my supervisor about all the scenarios that could happen in production environment and then wrote the unit test code.

2.5 Fix Test Application Bug

In the test application, it requires the operator to login with a valid credential before proceeding further. When the operator logs in, inside the memory of allocated to the browser, the username and the access token of the operator will be stored into two new fields that reside inside the allocated memory. When the either of the two fields is corrupted, for instance, the operator deliberately opens the browser's console and issues such operation, the web interface will automatically logout the operator without directing to the login page. To fix the bug, simply clear the local storage of the browser and reload the page to simulate the login page where no credentials are stored in lcoal memory.

2.6 Map Right Click Menu

This task was about frontend development. The test application allows the operator to set the start location and the destination of the route. The already implemented feature was to set the locations via typing in the text boxes on the webpage but not click and set directly in the map displayed on the web page. My supervisor asked me to add a right click menu to the map display with three commands: set the starting location, set the destination and set the current location. When the right click menu

was done, add an animation that shows the movement of the car per second. To simplify the task, the route was set to be the straight line between two points and equally divided into 100 segments. The map display is actually done through an API provided by Google Maps. I know if anyone who uses Google Maps through any modern browser can easiy right click on the web page and do all sorts of things. But unfortunately, no API provided by Google Maps that provides the right click menu. The solution was simply draw shapes on the Google map in-page canvas as the right click menu, style the menu, then bind JavaScript functions to the items in the menu to accomplish the tasks. The car movement animation was done by adding markers to the Google Maps in-page canvas and reposition those markers every time a timeout event was triggered to update the frame.

2.7 Diagnostics Application

This was my last task before I wrote it from ground-up. The appilcation was web-based and the map display was done through Mapbox instead of Google Maps. The purpose was to search locations of vehicles during an adjustable time interval, display and animate the results on the in-page map display. The developers at Fleetonomy already implemented the part for transfering and storing vehicle locations using Amazon Firehose. The task required using Amazon Athena to set up tables according to the data stored in Amazon S3 bucket, and make queries. Amazon Athena can be thought of a serverless query service that makes charges based on the amount of files that are scanned. The result of the query is stored in Amazon S3 bucket.

The flow of the application is divided into several stages. First, collect parameters for the query from the web page and send to backend. The backend issues queries to Athena based on the passed-in parameters. Second is about Amazon Athena processes the query and puts results in Amazon S3 bucket. The third step is that the webpage issues requests to the backend to get data from Amazon S3. The backend passes on the request and get data back from S3 and send back to the web page. The webpage will then finally display the positions as paths of the vehicle. Animation of vehicle movement is done by updating the locations of the markers.

The code I wrote can run queries about multiple vehicles at the same time and display different paths on the map. I used arrays to store the information so that data belong to different vehicle do not get mixed up. The application can be put into real production environment as an administrative tool.

3 Observation and Conclusion

The placement at Fleetonomy taught me a lot about the cycle of software development when working with teams. To have the code merged into the master branch in the code repository for production environment, the code must go through a series automated tests. The code repository hosting service the team used is called Gerrit. Gerrit is more suitable to show the differences and leave comments for different versions of the same branch, which is good for code reviewing, compared with traditional code repository hosting service such as GitHub and bitbucket. After the code is uploaded to a branch that is parallel with the master branch, an automated code testing system called CircleCI will take the code from Gerrit, build a virtual environment and run the unit tests for the code. It is the first step of verification. After the code passes the automated tests, another two developers in the team will check the code and leave comments. The person who writes the code needs to reply to the comments and change code if necessarily. The final step of verification is quality assurance. The company have emplyees outside Israel who do the quality check without knowing any conversation or comments related to the original to ensure unbiased views. If all goes well, the code will be merged into the master branch and put into the production environment.

I had this conversation with my boss, discussing what makes a good programmer. He did not say anything about diploma but experience, quality and speed. A resume filled with experiences always interests him more than a resume decorated with diplomats. He also encouraged me to try different things and he mentioned that a good programmer is one who always learns new things to ensure one is not left behind.

References

[1] Inc. Circle Internet Services. Circleci website. accessed in September 2018. http://circleci.com/.

[2] Datadog. Datadog website. accessed in September 2018. https://www.datadoghq.com/.

[3] Fleetonomy. Fleetonomy. accessed in September 2018. https://www.fleetonomy.io/.

[4] a member of Software Freedom Conservancy Git. Git, fast version control. accessed in September 2018. https://git-scm.com/.

[5] Slack. Slack website. accessed in September 2018. https://slack.com/.