BILKENT UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CS 399
SUMMER TRAINING REPORT

Nursena Kurubaş
21602965

Performed at

STM SAVUNMA TEKNOLOJİLERİ MÜHENDİSLİK VE TİCARET A.Ş.

10.08.2020 - 08.09.2020

Table of Contents

1 Introduction

I have done my summer training at STM's Command Control Information Systems department as a software engineer intern. I chose this company because I was interested in defense industries, and I thought that since STM is a well-known national company with several international projects, their success can contribute to my success as well. I also have heard from a STM employee that I know for several years that STM values their interns, and gives them a chance to take responsibilities at real projects, let them contribute to teamwork and help them improve their skills.

I have been given the task to do a wide research about the technologies that can be used at a subproject of the FocusFlite project -which has two parts: GS (Ground Segment) and AS (Air Segment)-, and do some designing and start developing if I happen to have any remaining time. This sub-project is considered to be a future extension or a seperate application that offers some new and useful features that are currently not existing at the FocusFlite GS project.

At this report, I mentioned the company that I did my training in, the details of the work I have done, the ways that I did my work, and some issues that I had came across during doing my training at Siemens, like how was the team work proceeded and what sources did I get use of along the training etc. Finally, I concluded the report by discussing how my training experience overall was.

I need to point out that I was not able to go to the company at the last days of my internship because of some health issues, so I could not collect and save screenshots from my demo program and I could not reach the reports I have written throughout the internship. Thus, the figures in this report are just illustrations of my project, and I briefly mentioned about the content of the reports without attaching them here.

2 Company Information

In this section of my report, I gave some information about the company that I have done the summer training.

2.1 About the company

STM is a company that provides other technical solutions for the defence industry, supplies hardware and software productions, offers technical and logistic support, system engineering, project management along with other services. It was established in 1991 by the decree of the Defence Industry Executive Committee to provide support to Turkish Armed Forces (TAF) and Undersecretariat for Defence Industries (SSM) [1].

2.2 About your department

I worked at the Command Control Information & IT Systems Department that is capable of implementing cross-platform IT solutions and informative programs at strategic and tactical levels, along with providing decision support and communication solutions. This department has a dozen projects that are released to the market, and actively being used in the field. The 8th floor of the main office of the company is reserved for this department [2].

2.3 About the hardware and software systems

While working at the office, employees are not allowed to use their personal computers for work, and thus the company provides a secured computer for each employee. Only these computers are able to connect the intranet of the office. Employees are able to request different or more powerful hardware products from the company, and as long as they have the right reasons, related departments take action and their requests are fulfilled. Employees use a chip card for entering the building, that only opens their own office doors and some floors that are open to common usage (like cafeteria). Since this company also produces hardware, the prototypes of the products and needed hardware materials are also existing.

As software systems, all computers have Windows 10 operating systems, and they use Outlook and intercom for communication between different offices and employees. To install a software program they wish, employees need to open a request for a related department to check the reliability of the software, and only after they approve the request, the software can be installed. They claim that this process is thought to be very important for security reasons. Also there are softwares installed to employees computers to control the actions and prevent them to perform actions that are unrelated to their work.

2.4 About your supervisor

My supervisor is Dr. Tahsin Alp Yanar, the platform support systems group leader at our department. He graduated from Hacettepe University, Computer Science and Engineering Department in 1999. Then, he did his master and doctorate at Middle East Technical University on Geodetic and Geographic Information Technologies [3].

3 Work Done

In this section, I described the project I worked on, and explained the work I have done to contribute to the project in detail, week by week.

The Project

I have been informed that I will work on a proof-of-concept project by doing detailed research and developing a demo. The project that I will work on will be an independent side project related to FocusFlite Mission Support System, which is a complete product of STM that is released to market. FocusFlite is an “Airborne Digital Moving Map System” that is designed to help the pilots during the mission. This program is integrated into the dashboard of the cockpit of vehicles [4].

On the other hand, the project that I will work on is significant in terms of tracking the airborne vehicles via a map dynamically and communicating with them (such as giving commands to them and receiving messages etc.) from land. Thus temporarily, this project is called Flight-Tracker. This project is thought to be a cross-platform project that should be able to work on different OS such as Android, iOS, Windows, Linux and embedded systems. This requirement is also one of the main differences between FocusFlite and Flight-Tracker, since even the FocusFlite Ground Segment project is designed only for embedded systems. Another requirement is to have one code-base for all operating systems and the reason behind this additional requirement is to maintain the project easily when we need to change or add some feature.

Week 1

I have started by doing research for the best technologies for the implementation of Flight-Tracker project concept. As a result of my research, I presented a list of technologies for the purpose of our project with their advantages and disadvantages along with their features and specialties. I took some informal notes as a reminder for my presentation but since the notes are very sloppy, I decided not to share them here. But I will mention the technologies I have found.

At first, I searched “Code-Free Multi-Platform App Creators” for our Client Side. These technologies are quite useful for easy and fast user interface development by designing the application through an interface -mostly website-, it is especially useful for those who are not familiar with coding. Some good examples for these programs are The App Builder, Appery, Appy Pie and Good Barber. I have searched these 4 in detail. At the end, I decided that these technologies will not be suitable for our purposes because using this kind of a platform with no-coding will cause us to have additional limitations, since we will only be provided the features that the creators allow us. I thought that if we do the coding ourselves, we will be a lot more flexible in terms of UI components, adding new features, and using the technologies we choose to create more efficient applications. So I continued doing my research.

Second stop of my research was progressive web applications. I was familiar with this method since I have used a similar technology at the senior design project. Progressive web applications are working and being implemented just like a regular web-page. So they can run on any device with a browser, this is a great plus for this project since it should be working on many different devices. The withdrawal from the website idea was mobile devices because this program is required to be looking like a native application. That is where progressive web-app concepts step in, by writing just a few lines of codes, the website will be able to be initialized by clicking the icon from main menu, and runs as if it is just a regular application -without browser components like link bar, back button etc.- The installation is also very simple, for example, in iOS, the website can be installed to the phone just by clicking the “Add to Home Page” button. However, this technique creates some inconvenience, such as when it comes to routing. If there is any routing somewhere in the app, the browser tools starts to show itself again, and it destroys the “native looking app” concept. So, although it is a good option, we decided not to use it if we can find a more convenient technology.

Then, I started searching for cross-platform development technologies. Although this is almost a new technology, there are dozens of tools for implementation. My first stop was the ones I know or heard from before: Flutter, Xamarin, React Native. While searching for their details and capabilities, I came across many other cross-platform technologies that might be useful, and searched for them as well, such as Electron, Adobe PhoneGap, Ionic. These all had their pros and cons, and very popular application products. However, I needed to cross each of them out for one or more of the following reasons (from most important to least):

- Although 60-80% of the program works on the same code-base, some required and additional features require native implementation for each different system.
- Do not work on every platform that we need. Some tools are only for mobile, some for desktop.
- For map technologies, we needed to consider that these needs might be extended in the future, so map integration needs to have many features and be flexible for development (changing and manipulating the map according to our needs). Because the map is the leading part of our program. Some did not have web integration and were using web-view to display a map, which was undesired since it is highly possible to cause problems in terms of performance. And some of them only had its own specialized map plugin, which will only allow us to use its own existing features and therefore, limitate us.
- Hard to understand and solve errors since there are not many resources and users.
- Not stable since it is a new technology.
- Has performance issues (runs slowly or creates lags in UI).
- Lack of UI components.

Finally, I came across the Qt technology. Qt has QML (Qt Modelling Language) that is used with JavaScript for UI implementation, and by doing some digging, I found this language to be perfectly compatible with our needs [5]. It works on every platform we require, has rich UI components, has map plugin that allows the manipulations that we need, works on one and only code base for any system, efficient and fast to learn and code. I also found a technology called Felgo, which is a SDK that specialized for Qt framework. It was claiming that it makes coding with QML a lot faster with its specialized components and live server [6]. Waiting for a couple of minutes at each time we need to test a UI feature was a significant waste of time, so I thought Felgo will indeed help us to develop the project faster.

In the end, for the client side, JavaScript and QML with Felgo SDK is chosen for the demo program. I installed the Qt Creator environment with Felgo SDK, which took a couple of days because of the company procedures. In the meantime, I started to learn QML basics by reading the documentation and examining the example applications. I started the development of the demo program on Thursday.

Week 2

I did some trials to get familiar with QML a little more by learning the components, their properties, functionalities and capabilities to observe the conventions and to see what I can and cannot implement by using them. By adding the MapBox plugin to the screen and putting some items on map to represent the vehicles, I have created the base UI of our program. I also added an extra future, that is, after clicking the circle icon that represents a vehicle, the map zooms in to that location, and a white box slides in to the screen that includes the latitude, longitude, altitude data. Clicking the cross icon, brings the application to initial form. I dealt with several errors during development, and solved them by searching the internet for the people that encountered similar problems.

Later, I have done some research for the server side. I was considering building a REST API for client-server communication since I was very familiar with this technology and also, it was convenient to use for our case because we were going to fetch the data from the server for every one or two seconds, thus we were going to be able to get any kind of data coming from server in a second, and by using REST API, we could send any message anytime we desire.

Since we were using QML already, I thought we could use Qt for the server side, and thus use C++ language and libraries. However, after some digging, I realized that developing REST API with C++ is hardly feasible and not convenient. After getting this knowledge, I chose using Node.js framework for server implementation because it was easy to code and running fast, and as a plus, we still were going to be able to integrate any C++ code to this server. Then, I installed the required tools and started developing the server side. I created some fake data by using MongoDB, and tested the server endpoints by using the Postman testing tool.

Week 3

I have started the week by carrying out a progress presentation for my team. I presented what I have done until now, what kind of problems I have encountered during development, how is this technology working for our purpose etc. We discussed what additional features we can do in the future, if we have any time left after completing the base of the project.

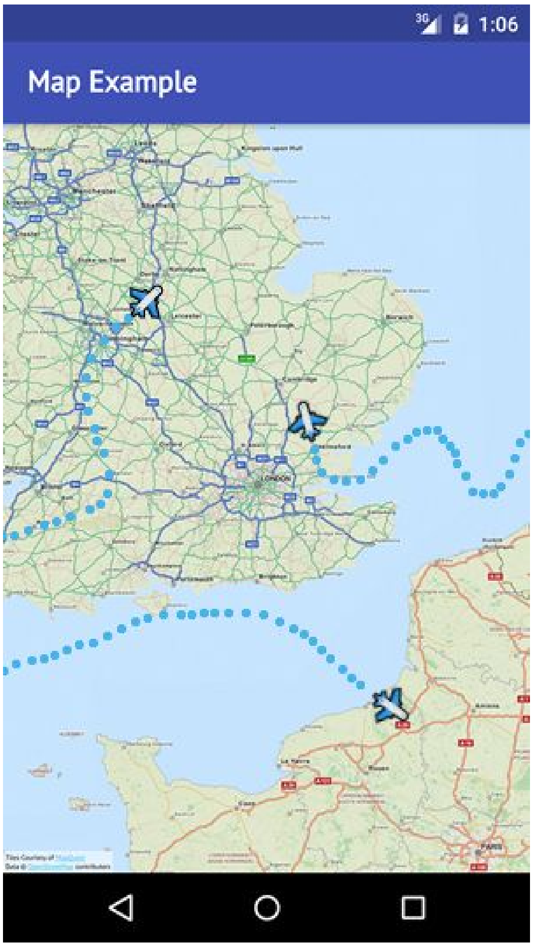


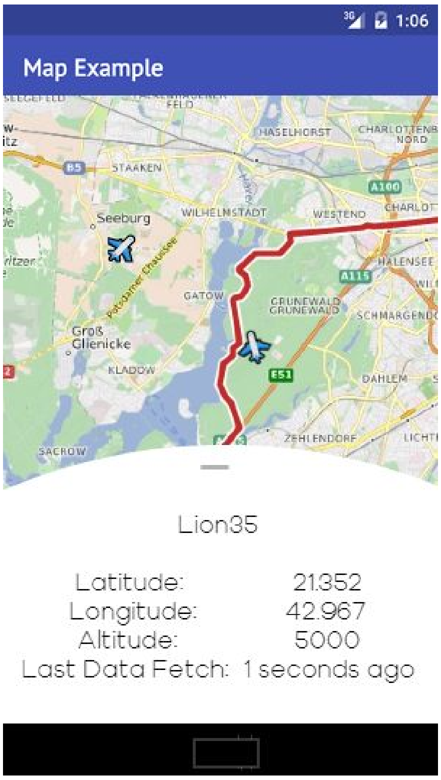
Figure 1. Flight Tracker Vehicle Trails

We created a to-do list for next weeks as follows:

- Refactor backend code.
- Display rotation using a plane icon.
- Show the mark that vehicles leave behind.
- Develop one way (client to server) messaging feature. (User should be able to send messages to vehicles. In the backend, logging is enough.)
- When the data of a vehicle stops coming in, show on the map its latest location and for how long we cannot receive data from that vehicle.
- Error Fix: Zooming in/out changes positions of the vehicle icons. Fix that problem.
- Calculate the risk of crash by using velocity information.
- Send alarms to the vehicle as the risk of crash increases.
- Try to use plain QML instead of Felgo. Write down any findings.

For starters, we decided to take a step further, and show dynamic data to represent moving vehicles. At this point, we integrated a previous code written in C++ of my teammate İsmail SAVAŞKAN that continuously creates changing location data for several vehicles. So, we refactored the backend code and got rid of the MongoDB dependency. I spent a couple of days fixing the UI to display and destroy components dynamically.

On Friday, I have managed to get the other four tasks that are following the first one done, so the project had many additional features than promised. I worked on the error written above and realized that it is related to Qt QML's MapQuickItem component. I could not find a way to change its default behavior. I looked up alternatives for putting an item on the map like icon buttons, but the other elements did not provide good practices for our purpose.



Lion35

Latitude: 21.352
Longitude: 42.967
Altitude: 5000
Last Data Fetch: 1 seconds ago

Figure 2. Flight Tracker Vehicle Details

Week 4

I started to investigate the pros and cons of replacing Felgo components with plain Qt QML components, and using plain Qt Creator without Felgo SDK. I have written a 1-2 page report about my findings and claims on this issue which briefly suggests that converting client side from Felgo to plain QML is possible, however it will probably take more time than I have for the rest of my internship, since it almost requires to start over the project and plain QML is harder to implement than the Felgo features.

It is decided to work on productization instead of adding more additional features, such as crash-detection. Also, I received a new to-do list as follows:

Research on:

- How to package the project as a mobile application?
- How can we make the current project work as a website?
- To what degree we are able to use our current C++ codes at the front-end of the project?
- How can we change the map base? Can we use OpenStreetMap instead of MapBox and take the map from another URL than www.openstreetmap.org?

I have made my research and reflected on these issues, and wrote down a report that answers each of these questions. I have managed to change the map-base from MapBox to OpenStreetMap. On the other hand, what I found on the productization research is as follows:

I could not try deployment since a lot of tools should be downloaded, and we need to open a request for each of them to start the procedure. And as I proceed, some more tools may be needed as a result of the trial-and-error method, and at this moment, constantly opening requests may cause unnecessary work for the IT employers who deal with the installation process.

However Felgo has packaging documentations with figures and videos for each system, and they are very detailed and explanatory, so I gave reference to them in my report for this week's research.

After I have done with the research, I started to reflect on the crash risk calculation algorithm. I came up with an algorithm that calculates the 30-second possible route of each vehicle considering their velocity by calculating the location change at past 2 seconds, and assuming the current rotation will be kept at following 30 seconds. Then it checks if any of the routes has the same coordinate at same seconds. If they do, it means there is a crash risk, and it sends an alarm to the user. Of course, this algorithm was not complete since the route prediction can be improved by using higher technologies, and there can be less tolerance while comparing how close the routes are etc. I implemented this algorithm at the server side, but could not find time to test it and implement the client side of it, that is, alarm receiving.

I have done a final presentation to explain all the work I have done, and my internship is over.

4 Performance and Outcomes

In this section, I explained some specialties of my work and different outcomes of my training.

4.1 Solving Complex Engineering Problems

The first problem I encountered was the lack of documentation. Since I was having hands-on experience on a language and technology that I had never experienced before, my work was highly dependent on the information given on documentation. Therefore, any missing features and explanations in the documentation caused me to have a difficult time while trying to understand how I can implement what I have in mind or how I can solve certain problems. At this point, I tried to get help from Google like I did while doing a project or homework at the university as our professors suggested to us, and some of the time I got use of the answers given to the other developers who had the same questions. But in many cases, I observed that there were not enough people who used that technology to have the same problem with me, or to solve this problem we had. In this case, I tried to approach the problem from different angles and find different solutions and did many other trials. If that did not work out either, I have consulted one of my teammates to get use of their experiences.

4.2 Ethical and Professional Responsibilities

STM has very strict rules about work ethics. They gave us handbooks about work ethics in orientation which was explaining which behaviors can be disturbing, and what consequences will people encounter if they insist on unrespectful behaviors, what should we do if we experience such behavior personally or if we witness that it happens to someone else. Moreover, the managers and directors were all very interested in us. They were always there to answer our questions, giving advice for our academic and professional life and being very helpful for anytime we need, although it is a large company and there are many employers who may consume their precious time, their doors were always open to help.

4.3 Making Informed Judgments

My judgements throughout this internship was mostly on choosing the technologies that will be used for this project. For example, I could choose to work with plain Qt QML but I chose to use it with Felgo SDK. I mentioned some of my technical reasons before but I chose it for these reasons as well: It was a new entrepreneurship project which I found to be very useful and clever, and I believed that they deserved to be supported for their work. I also considered the benefits of my company. This technology seemed to be quite useful for them once they bought it, and Felgo also has a special pricing for companies by creating a package according to their needs which I believe is very efficient, economically. This package is meticulously prepared, which can include courses, 7/24 technical support, help in projects etc. according to company's choice.

4.4 Acquiring New Knowledge Using Appropriate Learning Strategies

As I mentioned earlier, my internship was mostly passed by doing research about new technologies. I get use of sources on the internet, the explanations of experienced people and Q&A websites to find alternatives and choose the most convenient technology for our project. Then, I used the documentation of the technologies to learn how to use them, and again I used Q&A websites to solve the errors I have encountered. Q&A websites were very useful in terms of observing the claims and solutions of experienced developers. Documentations are also quite useful especially if they are detailed since they are written by the experts of that technology, and they are meant to explain every feature, property and functionality of the related technology.

4.5 Applying New Knowledge When Needed

By doing online research and reading the guidelines of experienced people, I learned how to build a REST API server from scratch using Node.js framework. I found that Postman is a qualified application to test the api, so I used it as well. I created a MongoDB database for my project. And by reading the documentation of both Felgo and Qt, I developed a user interface for my project using QML.

5 Conclusions

Looking back, it is clear for me that I have gathered a lot of technical knowledge, and I also found the chance to make practice about them. This was my first time to start such a project from scratch on my own, and I have encountered a lot of unexpected real-life problems, and it helped me improve myself on problem-solving and working under stress. I improved my cross-platform application development skills and I also learned using many new technologies and tools.

I experienced a lot about office life, such as participating in meetings and conferences. I had a chance to observe how things work at a defense company. I worked with a team of highly-skilled engineers, so I was able to get use of their knowledge and experiences. And I can say that this training improved my work-related social skills as well. As a conclusion, I can say that this training was very improving and effective for me in all aspects.

References

[1] “STM | Company Profile”. https://www.stm.com.tr/en/about-us/company-profile. [Accessed: Aug 13, 2020].

[2] “STM | Command Control Information & IT Systems”. https://www.stm.com.tr/en/business-areas/technology/competencies/command-control-information-systems. [Accessed: Aug 13, 2020].

[3] “Tahsin Alp Yanar | LinkedIn”. https://www.linkedin.com/in/tahsinalpyanar/. [Accessed: Aug 13, 2020].

[4] “FocusFlite”.
https://www.stm.com.tr/en/our-solutions/command-and-control/focusflite-en. [Accessed: Oct 11, 2020].

[4] “Qt QML 5.15.1”. https://doc.qt.io/qt-5/qtqml-index.html#qml-and-qml-types. [Accessed: Oct 11, 2020].

[4] “Felgo Documentation | Felgo Documentation”. https://felgo.com/doc/index/. [Accessed: Oct 11, 2020].

Self-Checklist for Your Report

Please check the items here before submitting your report. This signed checklist should be the final page of your report.

- Did you provide detailed information about the work you did?
- Is supervisor information included?
- Did you use the Report Template to prepare your report, so that it has a cover page, the 8 major sections and 9 subsections specified in the Table of Contents, and uses the required section names?
- Did you follow the style guidelines?
- Does your report look professionally written?
- Does your report include all necessary References, and proper citations to them in the body?
- Did you remove all explanations from the Report Template, which are marked with yellow color? Did you modify all text marked with green according to your case?

Signature: Nursena Kurubaş