Mobile-based Virtual Traffic Lights

25 August 2020

Paper: Mobile System Development

Student Name: Yanan Li

Student ID: 19058601

Content

1. Research Topic, Scope and Role

1.1 Research Topic

Traffic lights are an essential factor that influences the efficiency of traffic. Since the first traffic light being installed in London, traffic lights has improved the safety of road users and decreased the time consumed on the roads significantly (Seasider, 2020). However, with the development of social economies, there are increasingly vehicles running on the roads and traffic congestion has become a problem that cannot be neglected (Liu et al., 2017). According to Hartanti, Aziza, and Siswipraptini (2019), one of the reasons leading to this problem is that the traffic light system is time-fixed and it can not adjust time intervals automatically when the density of traffic flow changes. Consequently, it is common to see traffic congestion in crowded intersections. Furthermore, during the congestion, vehicles are operating in low-speed mode, which could result in the waste of energy and environmental pollution (Armah, Yawson, & Pappoe, 2010).

In order to mitigate traffic congestion, Virtual Traffic Light (VTL) is proposed by Ferreira, Fernandes, Conceição, Viriyasitavat and Tonguz (2010). VTL is a system when the vehicle is heading to the intersection area, the vehicle uses Dedicated Short-Range Communication (DSRC), Long-term Evolution Vehicle (LTE-V) or 5G communication to determine which car is the lead car, and the passing priority of the vehicle in which direction is determined by the lead car. When the lead vehicle passes an intersection, the distance between the following vehicle and the lead vehicle is calculated. Then it is used to determine whether to pass the priority to the following vehicle behind the lead vehicle. Otherwise, the lead vehicle will be the vehicle in the other orthogonal lane. Through this method, the traffic order of each intersection is determined in turn, so as to improve the traffic efficiency, shorten the waiting time for red lights, and reduce traffic congestion.

1.2 Scope

In this project, an Android application will be developed. The working flow of this application is based on the VTL approach. When individuals drive vehicles crossing intersections and they have installed this application on their mobile phones, the information of the vehicles, such as position, speed and velocity, will be detected and sent to the cloud server; then these vehicles will be instructed to pass the intersections without conflicts.

As shown in Table 1, there are two modules in this project.

Table 1. Project Modules

Number	Module	Sub-Module
1	Application	Vehicle Data Collection
		Traffic Light Display
		Communication Fail Module
2	Backend Services	Vehicle Data Process Module
		Command Distribution Module

The following work will be conducted:

1. Figure out the function of the application, and make a digital prototype of the application.
2. Design the VTL algorithm and implement it.
3. Design the application architecture of the application.
4. Design the web services architecture of the application.
5. Develop and implement the application and the backend services.

1.3 Role

In this project, there will be several roles I play. They are project manager, designer, architect, developer and tester.

Project Manager: A project manager collects the requirements, and manage the overall progress of the project. Besides, the communication between different roles are also coordinated by a project manager.

Designer: A designer designs the user interfaces of the application.

Architect: An architect designs the architecture of the application, for example, deciding what kind of design patterns should be used, such as Model-View-Controller (MVC) or Model-View-View Model-Model (MVVM).

Developer: A developer implements the application according to the architecture designed by an architect.

Tester: A tester conducts test on the product developed by developers according to the written test plan.

Due to the limited time, the role that I mainly play in this project is a developer. In other words, I will focus more on the development side of the project. In this project, the most important part is the implementation of the VTL algorithm, and this task is the job that is needed to be undertaken by developers.

2. Development Process

Software development life cycle (SDLC) is a process that consists of different stages during the development of a system or software (Ruparelia, 2010). Although there have been a lot of development processes proposed and employed in industry in order to achieve various objectives, the most prevalent three will be discussed in this section, which are Waterfall Process, Agile Scrum Process and Incremental Process. Then, the process that is employed in this research project will be illustrated, and the timeline of this project will be described as a Gantt chart.

2.1 Waterfall Process

The waterfall process is a model that contains a series of linear stages of constructing a software. As Munassar and Govardhan (2010) stated, the waterfall model has been adopted in a lot of industrial organisations. According to Royce (1987), there are seven stages involved in the waterfall process. These stages are system requirements, software requirements, analysis, program design, coding, testing and operations, which could be seen in Table 2. The beginning of each stage relies on the outcomes of previous stage, and with the progress of each stage, the software is illustrated in more details.

One of the characteristics of the waterfall model is that there are numerous documents produced along with the cycle of the project (Petersen, Wohlin, & Baca, 2009). The number of produced documents in the waterfall process is greater than four, which can be seen in Table 2. The quality of each stage is also evaluated within stages. Take coding stage as an example; there is unit testing and integration testing; by conducting these tests, the quality of codes could be improved significantly.

Table 2 Waterfall Process and Documents

Stage	Document
System requirement	Requirement document
Software requirement	
Analysis	Design specification
Program design	
Coding	
Testing	Test cases and results
Operation	User manual

However, there are disadvantages along with the waterfall model. One of them is that there is no feedback and iteration between previous and later stages. If there are drawbacks in one stage, and the project is proceeding in a later stage, there is no way of fixing this situation. Due to this disadvantage, the waterfall process only works well when the project is small enough or the requirements of a project are quite unambiguous (Kramer, 2018).

2.2 Scrum Process

Agile is a development model that break a project into many smaller projects (Ruparelia, 2010). It employs iterative and incremental methods to develop a project. In addition, in this model, stakeholders are involved in the whole process of the project, such as testing and prototyping. This model is created to improve the flexibility of a project when the requirements change frequently. There are a lot of development models proposed with the principles of agile, such as extreme programming (XP) and test driven development (TDD).

Scrum is also an implementation of agile model. In this process, a large project is divided into small inter-connected and runnable features. During this process, the software or system is available continuously.

Table 3 Sprint Process, Outputs and Roles

Stage	Output	Role
Requirements gathering	Product backlog	Product owner (PO)
Sprint planning meeting	Sprint backlog	Scrum team
Daily scrum meeting	Burn down chart	Scrum team
Sprint review meeting	N/A	PO, scrum team
Sprint retrospective meeting	N/A	PO, scrum team

As shown in Table 3, there are four stages involved in scrum process. The first stage is requirements gathering. The PO is involved in this stage, and the output of this stage is a document called product backlog. After having the product backlog, the Scrum team will have a meeting called sprint planning meeting; during this meeting, some features from product backlog are selected and included in a sprint. These selected features are called sprint backlog. Then each member of the scrum team chooses detailed tasks from the sprint backlog. During the development period, a 15 minutes meeting called daily Scrum meeting is held daily, each member should report to other members what they did previous day, what they plan to do current day, and what the problems they have met. Besides, after finishing this meeting, the burn down chart should be updated accordingly. When Scrum team finishes all the tasks in sprint backlog, a sprint comes to the end, there are also two meetings to be held, which are sprint review meeting and sprint retrospective meeting. These two meetings are the summary of a sprint. These stages and mechanisms in Scrum could assure the delivery of the most appropriate product releases.

2.3 Incremental Process

In Incremental Process, a software or a system is divided into many small sub-systems and the developers treat each module as an incremental component to analyse, design, code and test. Compared with the waterfall model, developers do not need to submit the entire software product to users at the final stage; alternatively, they can submit them in iteratively.

Incremental Process starts from a set of given requirements and implements development activities by constructing a series of executable intermediate versions. The first version includes some requirements, the next version includes more requirements. The process comes to the end until the system is completed.

There are three general stages within the Incremental Process. The first one is the stage before developing. In this stage, it is still necessary to conduct requirements analysis and determine the system's demand framework based on incremental components, and to determine the composition of the components. The design of the architecture of the software system is also finished in this stage. The second stage is developing. In this stage, we can develop incremental artifacts. Meanwhile, it is necessary to refine the requirements of the component, and then perform design, code testing and function validation. The last stage is integration. After completing the development of an incremental artifact, it is time for us to integrate the component into the system, and verify the validity of the system, and then continue the development of the next incremental artifact.

2.4 Development Process Employed

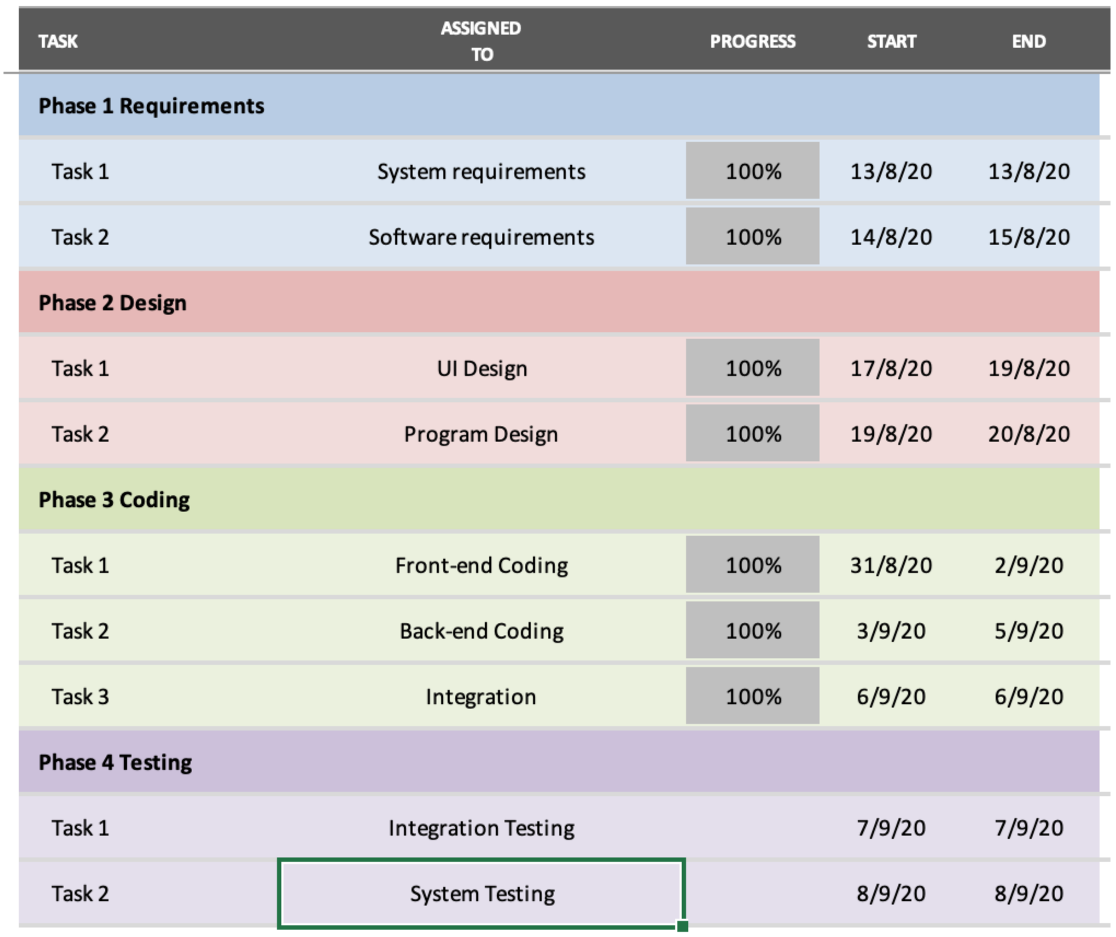
The development process that will be employed in this mobile system is the Waterfall Process.

Waterfall Process is appropriate for projects that have clear requirements and have few requirement changes along with the progress (Kramer, 2018; Shukla & Saxena, 2013). In contrast, the other two processes introduced above, which are Scrum Process and Incremental Process, have better performance when the requirements are in the state of changing. In considering that the requirements in this project are not complicated and the requirements have little possibility of changing, the Waterfall Process is the best option for this project.

With the Waterfall Process, there will be five stages involved in this project, and they are system requirements, software requirements, analysis, program design, coding, testing, operation or maintenance. In addition, there will be four documents and specifications produced along with the project.

2.5 Timeline

The timeline of this project is described in Figure 1.



TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1 Requirements				
Task 1	System requirements	100%	13/8/20	13/8/20
Task 2	Software requirements	100%	14/8/20	15/8/20
Phase 2 Design				
Task 1	UI Design	100%	17/8/20	19/8/20
Task 2	Program Design	100%	19/8/20	20/8/20
Phase 3 Coding				
Task 1	Front-end Coding	100%	31/8/20	2/9/20
Task 2	Back-end Coding	100%	3/9/20	5/9/20
Task 3	Integration	100%	6/9/20	6/9/20
Phase 4 Testing				
Task 1	Integration Testing		7/9/20	7/9/20
Task 2	System Testing		8/9/20	8/9/20

Figure 1 Project Gantt Chart

3. Technology

In this section, I will discuss three technologies that are related to develop mobile-based virtual traffic, which are React Native, Uni-app and Flutter. Then, The reason why Uni-app is chose to develop this project is explained.

3.1 React Native

React Native is a framework that can be used to develop cross-platform mobile applications (Prajaut, 2020). It is based on the JavaScript framework React, which is developed by Facebook. In the early stage, React Native could only be used to develop iOS applications; however, with its popularity, it is upgraded to support the development of Android applications (Hansson & Vidhall, 2016). This framework could give developers the ability of using JavaScript to write native applications. It is consistent with React in design principles and builds user interfaces through declarative components mechanism.

There are many reasons why developers using React Native. The first one is 'learn once, write anywhere'. It means that developers only need to write the code once, they could release the applications that could run on iOS platform and Android platform. This could help companies save human resources and financial resources greatly. Another advantage is that the performance of React Native is better than hybrid approach. The views used in React Native are identical with native components, which is different from the components rendered when using hybrid development methods. The third reason is that the learning curve is effortless. For a developer who has experience in using JavaScript, HTML and CSS, he or she will only need a little time to learn this framework.

However, there are risks with using React Native. The most serious problem is the debugging problem when using React Native (Eisenman, 2015). He explained that an extra layer is introduced into the project, and this layer is between React Native and the mobile platform leading to increasing the complexity of debugging.

3.2 Uni-app

Uni-app is a front-end framework. It is based on the JavaScript framework Vue.js. By using this framework, developers can develop only one codebase for various mobile platforms, such as iOS, Android, HTML5 etc. As the company DCloud (2020) declares, there are more than 5 million developers adopting Uni-app as their development framework. Besides, there are more than 100,000 applications developed by employing Uni-app that are serving at least 1 billion monthly active mobile users. The most important aspect is that the official provides developers with thousands of plugins to accelerate the development process.

There are also drawbacks when using Uni-app. The first one is that Uni-app has been released for a relatively short time, and there are many places where it is not perfect and there are many pitfalls (Ren, 2019). Another drawback is mentioned by Xiao (2019), which is the inconvenience of using animation with Uni-app.

3.3 Flutter

Flutter is the development framework introduced by Google. Google invented this framework for Fuchsia Operating System (Fuchsia OS). In order to be compatible with low efficiency Internet of Things (IoT) devices, Fuchsia OS is designed with the feature low power consuming (Grønli, Biørn-Hansen, & Majchrzak, 2019). As a result, the performance of Flutter is better than other cross-platform frameworks (Wu, 2018). As an user interface library, Flutter only needs to render the interface. Unlike HTML5, Flutter interface library does not even have basic functions, such as video, positioning, etc. It is a pure typesetting engine that draws common interface components such as text, buttons, and images. Although the performance of flutter is the best, the flexibility of Flutter is limited. This is because the method of writing layout in Flutter is restricted, only with restrictions, the speed of rendering pages could be improved (DCloud_heavensoft, 2020). The cost of improving performance in this way is that the code nesting in Flutter makes developers crash when laying out complex interfaces.

3.4 Technology Employed

The technology that I will use in this project is Uni-app. Firstly, Flutter is so complicated that a JavaScript developer need to spend a lot of time on learning this this technology (Jayalath, 2020). For this reason, Flutter is not chose as the developing technology. Secondly, developers need to write two user interfaces for iOS and Android when they use React Native or Flutter (DCloud_heavensoft, 2020). For instance, in Flutter, if you want to render a button on iOS platform, the component ‘CupertinoButton’ is needed, however, the button component for Android platform is ‘RaisedButton’, this mechanism do need the developer to write two sets of code. This is also the reason why React Native has only proclaimed ‘learn once, write anywhere’ instead of ‘write once, run anywhere’. Finally, Uni-app framework does not have the drawbacks that I mentioned above.

4. Design of Deliverables

In this section, the initial structures of this mobile system is illustrated.

4.1 Logical Decomposition

The purpose of this system is to collect vehicles’ speed, moving direction and position, then give vehicle owners instructions about passing the interaction or stopping at the intersection.

There are two types of users within this system, which are vehicle owners and pedestrians. The environment where the system is employed is unified. The only environment is at an intersection.

4.1.1 Users

Vehicle owners are the first type of user. When the owner drives his or her vehicle and approaches an intersection, he or she follows the instruction delivered by the application installed on their mobile phones.

Pedestrians are the second type of user. When pedestrians approach interactions, they will also follow instructions from the VTL application.

The use cases of this two kinds of users are illustrated in Figure 2.

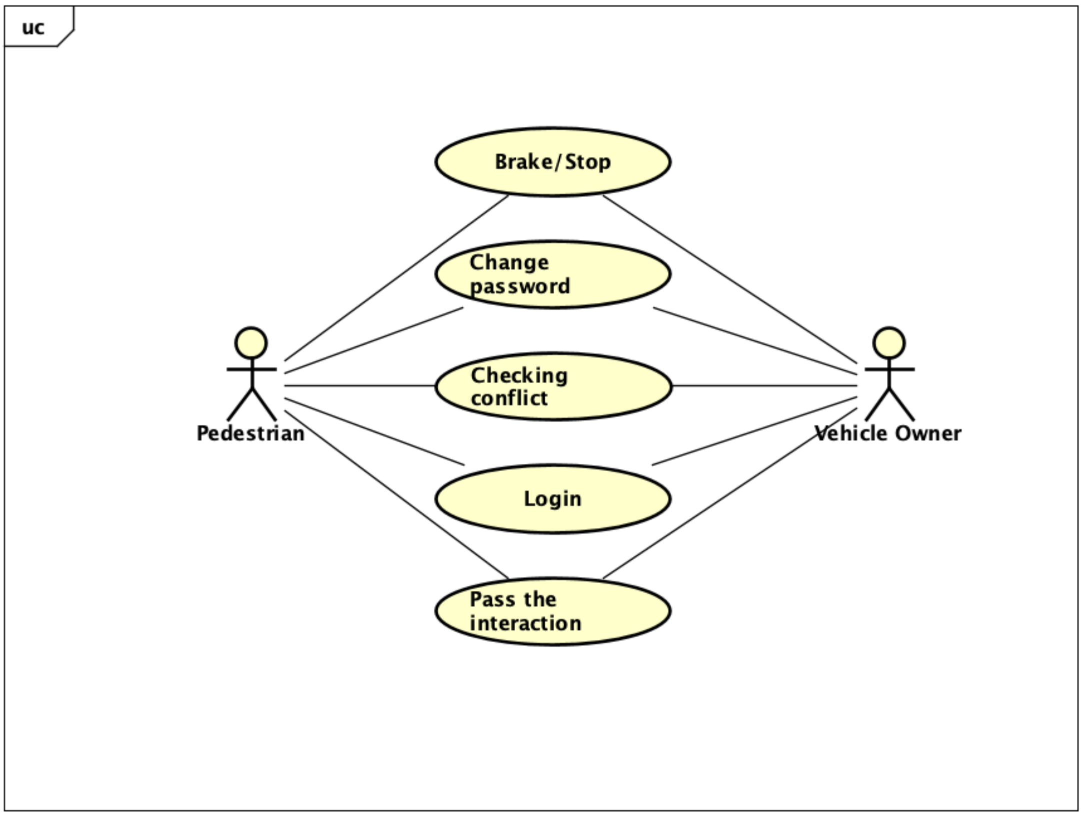


Figure 2 Use Cases

4.1.2 Domain Classes

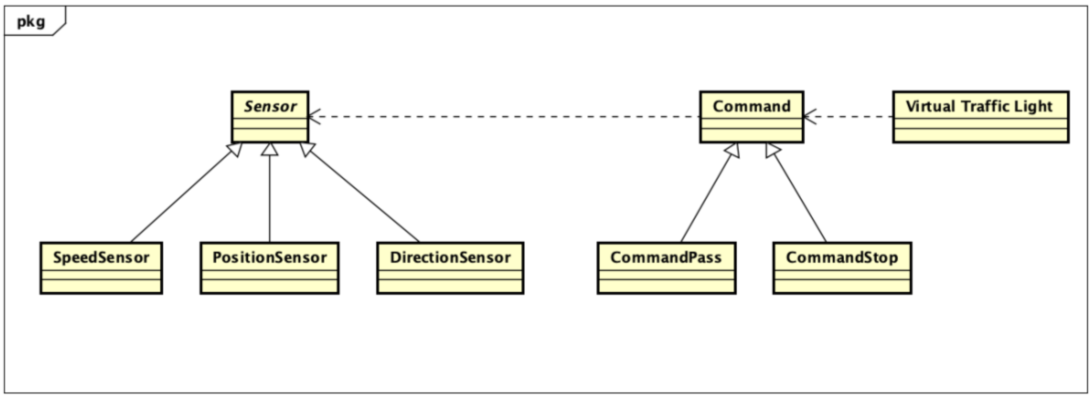


Figure 3 Class Diagram

As shown in Figure 3, there are three main domains in this system.

Firstly, sensor is the general concept of all the sensors that are used in this project. They are utilised to collect vehicle environmental data. Among these sensors, the speed sensor is used to collect the speed of the vehicle, the position sensor is used to collect the position of the vehicle, and the direction sensor is used to collect the direction.

Secondly, command is the general concept of the instructions. Specifically, the pass command is used to notify the vehicle to pass the intersection, and the stop command is used to notify the vehicle to stop at the intersection.

The last one, virtual traffic light, is used to give drivers indication of passing or stopping.

4.1.3 Sequence Diagram

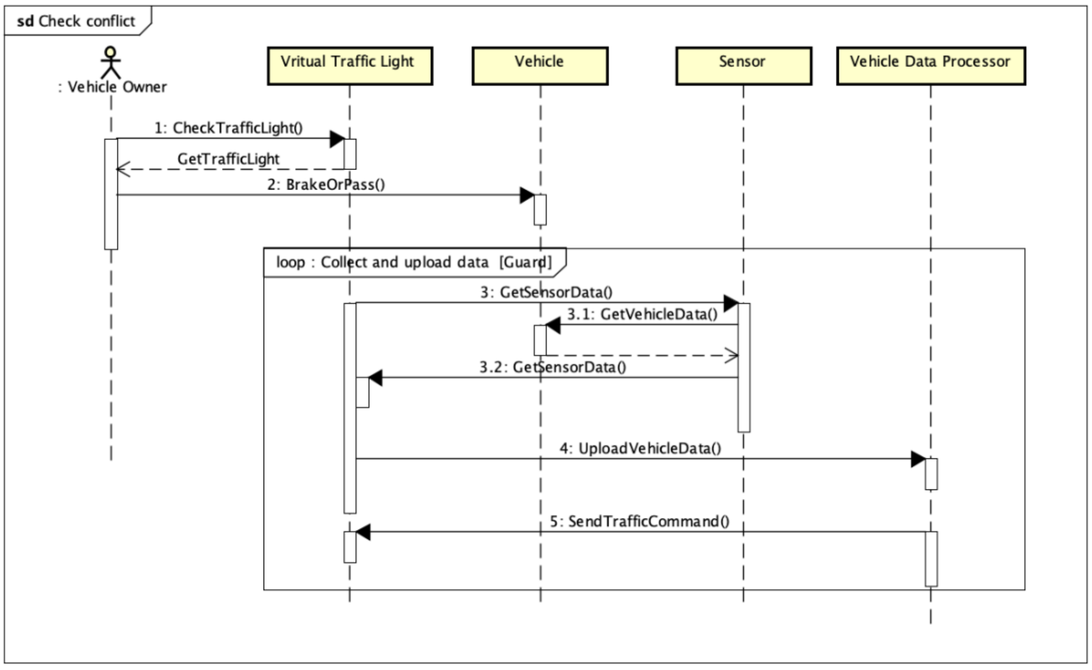


Figure 4 Check Conflict at Intersections

4.1.4 State Diagram

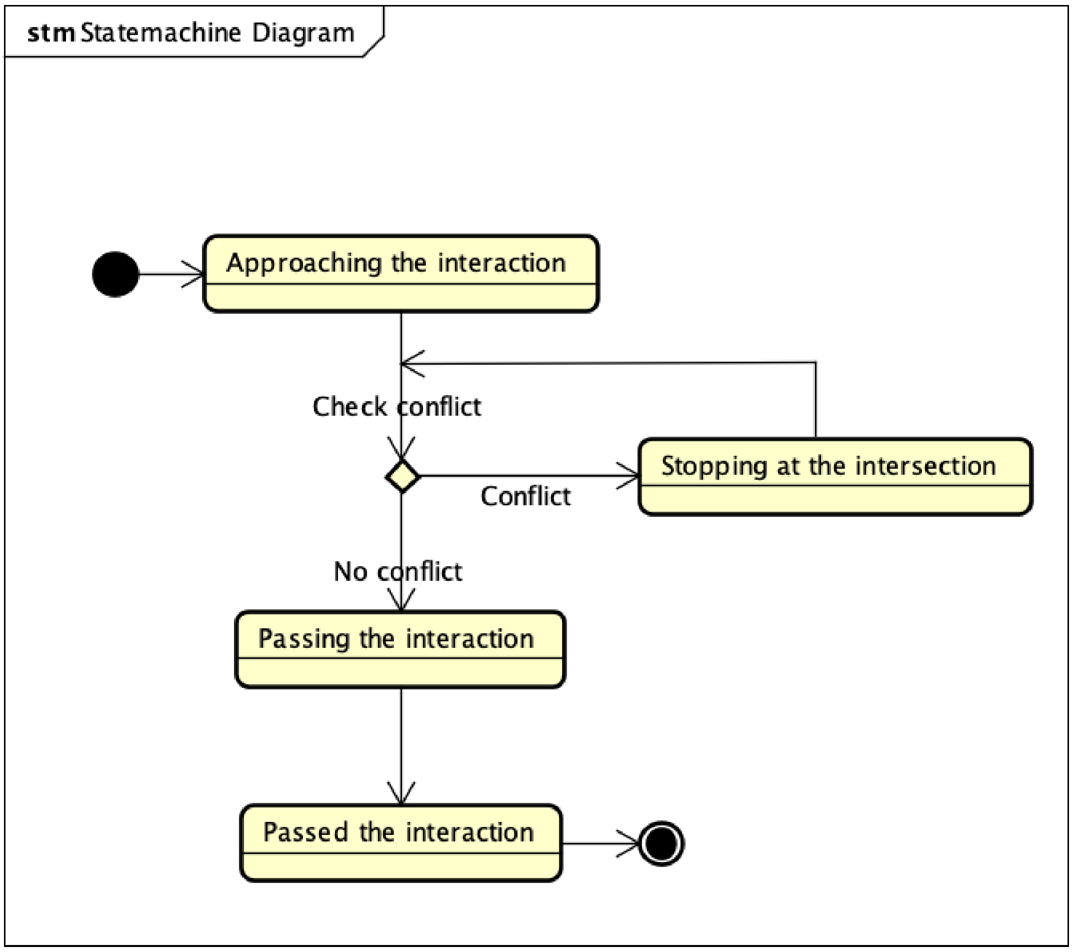


Figure 5 State Machine Diagram

4.2 Runtime Characteristics

4.2.1 People At the Centre of Mobile Application Development (PACMAD) model

According to Harrison, Flood and Duce (2013), usability is an essential factor that influences the success of an mobile application. They proposed a usability evaluation model called People At the Centre of Mobile Application Development (PACMAD). With this model, they expanded the existing usability model to the mobile application context. There are seven components in PACMAD model, which are effectiveness, efficiency, satisfaction, memorability, errors, cognitive load and learnability. Although PACMAD model could be used as the mobile application usability model, its main drawback is the lack of detailed guidelines and evaluation metrics.

The PACMAD usability model of the checking conflict task in this project is presented in Table 4.

Table 4 Check Conflict Task Usability Model

	User(Vehicle Owner)	Task (Check conflict)	Context (Approaching an intersection)
Effectiveness	The vehicle owner could get the correct instructions at the intersection.		
Efficiency	The vehicle owner could get the correct instructions at the intersection within two seconds.		
Satisfaction	The vehicle owner is satisfied with using the application to get instructions at the intersection.		
Memorability	The vehicle owner can remember how to get instructions using the application.		
Errors	There are no errors occurring while the vehicle owner checks conflict.		
Cognitive load	There is almost no cognitive processing needed to check conflict.		
Learnability	All vehicle owners can learn how to check conflict within this application.		

Besides, there will be several design patterns used in the project, such as MVC, MVVM, singleton pattern and factory pattern. These patterns can improve the performance of the system.

4.2.2 Navigation Graph

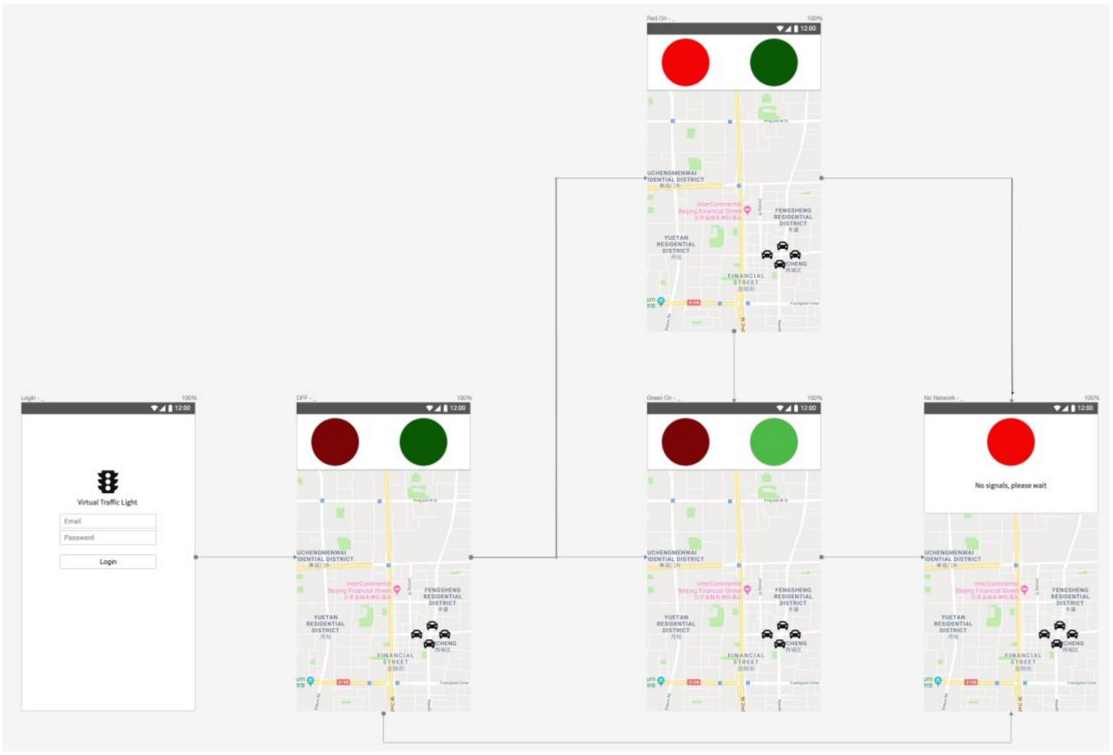


Figure 6 Navigational Graph

The navigation graph is shown in Figure 6. When a user opens the application the first time, he or she enters the login page. After signing up, the user enters the traffic light off page. If the user is not approaching an intersection, the traffic light will keep being off. When the user is approaching an intersection and there are not collisions detected, the user will enter the green light on page. If there are collisions detected, the user will enter the red light on page. In the worst scenario, which is the signal of the mobile phone being lost, the no signal page will be displayed.

4.2.3 User Interface Prototype

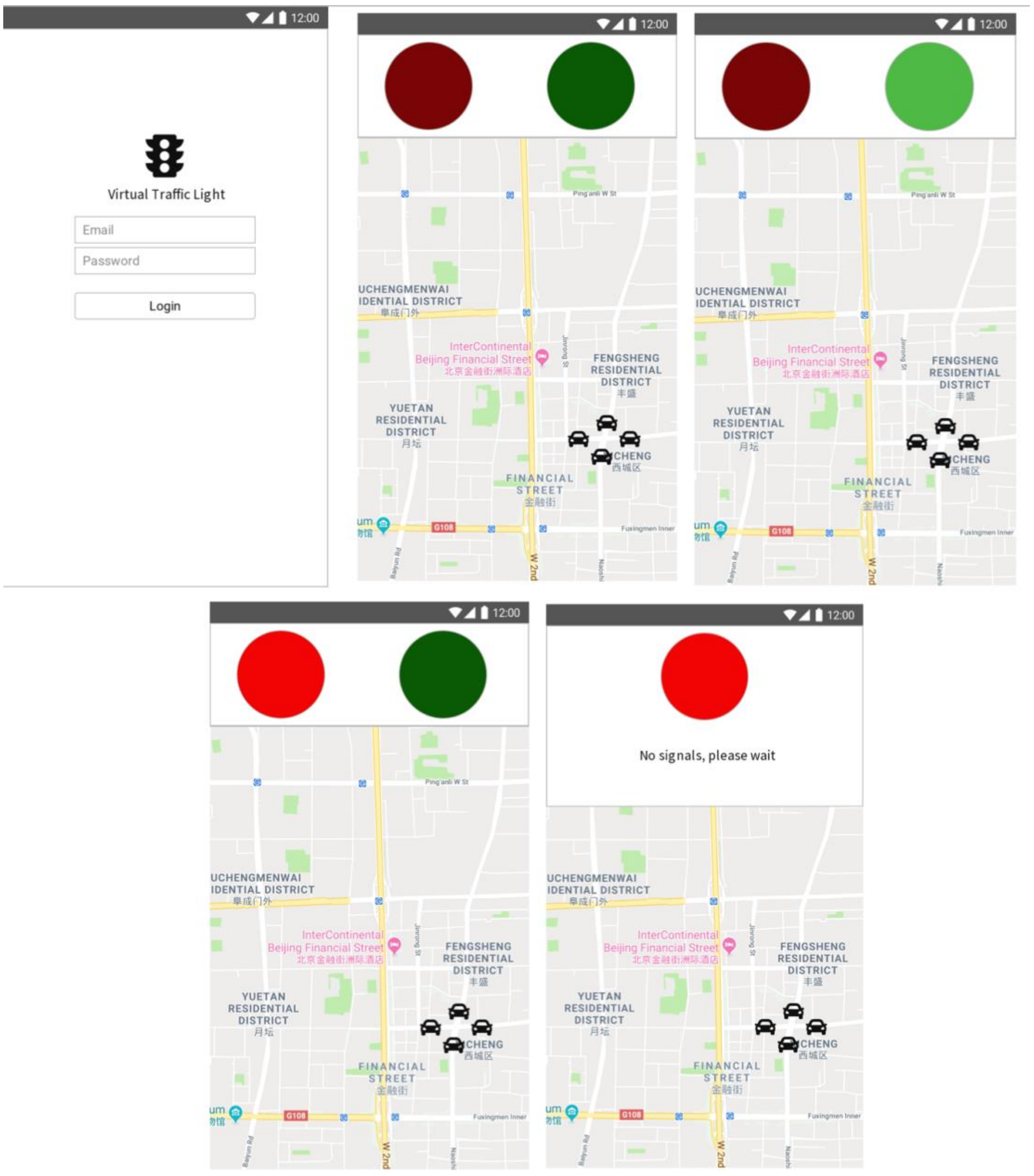


Figure 7 User Interface Prototype

As shown in Figure 7, there are seven pages in the application, and they are login page, traffic light off page, green light on page, red light on page and no signal page respectivelly. The login page is used for users to sign in. The traffic light off page is displayed after the user signing in. The green light on page is used to indicate that the user can cross the intersection. The red light on page is used to indicate that the user cannot cross the intersection. The no signal page is used to tell the user that the signal of the mobile phone is lost, and the user has to stop at the intersection.

4.3 Physical Characteristics

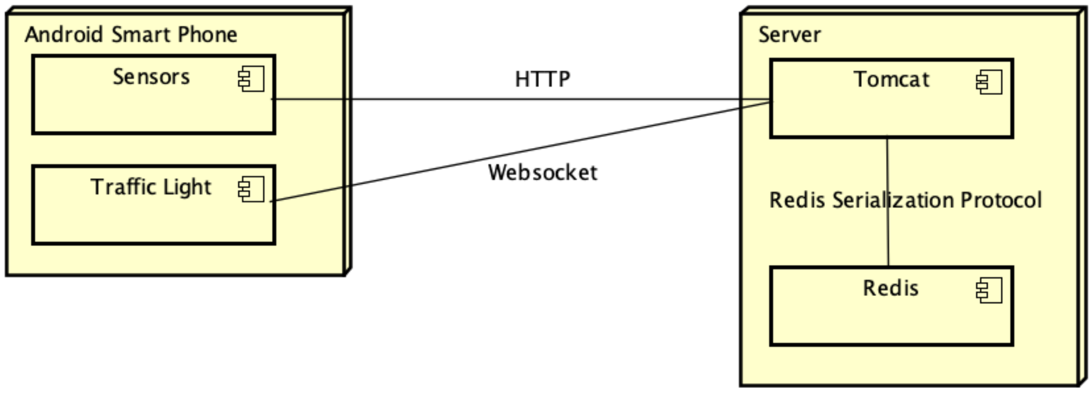


Figure 8 Deployment Diagram

As shown in Figure 8, the VTL application is installed in the Android Smart Phone, the vehicle data is collected from sensors within the Android Smart Phone, then the data is sent to the server through HTTP. Then the server stores data in Redis database. The Server broadcasts instructions to all vehicle clients through Websocket.

5. Evaluation Criteria

The evaluation criteria are listed in Table 5, including both basic functional evaluation criteria and evaluation criteria in abnormal conditions. For mobile applications, it is often not enough to successfully complete all business function tests. When a mobile application is installed and used by a large number of users, it will expose many problems that were not anticipated before.

Table 5 Evaluation Criteria

Task	Initial state	Conditions	Result
Login	The application is not opened and signed in.	Comfortable environment with access to the Internet	The user can login with the right username and password.

Logout	The user has signed in.	Good access to the Internet	The user can logout successfully and the login page shows up.
Cross the interaction	The vehicle is approaching the intersection and there are no other vehicles in the vertical direction.	Good access to the Internet	The vehicle crossed the intersection with the light being green.
Cross the interaction	The vehicle stops at the intersection and the traffic light is red.	Good access to the Internet	The traffic light turns green and the vehicle cross the intersection.
Stop at the intersection	The vehicle is approaching the intersection.	Good access to the Internet	The traffic light turns red and the vehicle stops at the intersection.
Stop at the intersection	The vehicle is approaching the intersection	Poor access to the Internet	The traffic light turns red and give indication about poor Internet access.
Switch the application to background, then switch it back	The application is running normally	Good access to the Internet	The application is still running normally.
Turn off location permission of the application	The application runs normally.	Good access to the Internet	An unauthorized location permission message is shown on the screen.

References

Armah, F. A., Yawson, D. O., & Pappoe, A. A. (2010). A systems dynamics approach to explore traffic congestion and air pollution link in the city of Accra, Ghana. *Sustainability, 2*(1), 252-265. https://doi.org/10.3390/su2010252

DCloud, (2020). *What is Uni-app*. Retrieved August 25, 2020, from https://uniapp.dcloud.io/README

DCloud_heavensoft, (2020). *The comparison of Flutter, React Native and Uni-app*. Retrieved September 3, 2020, from https://ask.dcloud.net.cn/article/36083

Eisenman, B. (2015). *Learning react native: Building native mobile apps with JavaScript*. CA: O'Reilly Media, Inc..

Ferreira, M., Fernandes, R., Conceição, H., Viriyasitavat, W., & Tonguz, O. K. (2010). Self-organized traffic control. *In Proceedings of the seventh ACM international workshop on VehiculAr InterNETworking*, 85-90. https://doi.org/10.1145/1860058.1860077

Grønli, T. M., Biørn-Hansen, A., & Majchrzak, T. A. (2019). Software development for mobile computing, the internet of things and wearable devices: Inspecting the past to understand the future. *In Proceedings of the 52nd Hawaii International Conference on System Sciences.*

Hansson, N., & Vidhall, T. (2016). Effects on performance and usability for cross-platform application development using react native.

Harrison, R., Flood, D., & Duce, D. (2013). Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Science, 1*(1), 1. https://doi.org/10.1186/2194-0827-1-1

Hartanti, D., Aziza, R. N., & Siswipraptini, P. C. (2019). Optimization of smart traffic lights to prevent traffic congestion using fuzzy logic. *TELKOMNIKA Telecommunication Computing Electronics and Control, 17*(1), 320-327. https://doi.org/10.12928/TELKOMNIKA.v17i1.10129

Jayalath, P. (2020). *Flutter VS React Native*. Retrieved September 3, 2020, from https://medium.com/@pramodayajayalath/flutter-vs-react-native-eab8692b7fdd

Kramer, M. (2018). Best practices in systems development lifecycle: An analyses based on the waterfall model. *Review of Business & Finance Studies, 9*(1), 77-84.

Liu, J., Wan, J., Jia, D., Zeng, B., Li, D., Hsu, C. H., & Chen, H. (2017). High-efficiency urban traffic management in context-aware computing and 5G communication. *IEEE Communications Magazine, 55*(1), 34-40.

Munassar, N. M. A., & Govardhan, A. (2010). A comparison between five models of software engineering. *International Journal of Computer Science Issues (IJCSI), 7*(5), 94.

Petersen, K., Wohlin, C., & Baca, D. (2009). The waterfall model in large-scale development. *In International Conference on Product-Focused Software Process Improvement,* 386-400.

Prajaut. (2020). *React Native.* Retrieved August 25, 2020, from https://en.wikipedia.org/wiki/React_Native

Ren, J. (2019). *Advantages and Disadvantages of Uni-app Development*. Retrieved September 3, 2020, from https://www.jianshu.com/p/7125360f3d2a

Royce, W. W. (1987). Managing the development of large software systems: concepts and techniques. *In Proceedings of the 9th international conference on Software Engineering,* 328-338.

Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes, 35*(3), 8–13. https://doi.org/10.1145/1764810.1764814

Seasider. (2020). *Traffic light.* Retrieved August 25, 2020, from https://en.wikipedia.org/wiki/Traffic_light#History

Shukla, A. K., & Saxena, A. (2013). Which Model is best for the Software Project?" A Comparative Analysis of Software Engineering Models". *International journal of Computer applications, 76*(11).

Wu, W. (2018). React Native vs Flutter, Cross-platforms mobile application frameworks.

Xiao, C. (2019). *How is the Uni-app experience? Uni-app development experience and analysis of advantages and disadvantages*. Retrieved September 3, 2020, from https://translate.google.com/translate?hl=en&sl=zh-CN&u=https://blog.csdn.net/qq_35616850/article/details/94439675&prev=search&pto=aue

Zahra, F., Hussain, A., & Mohd, H. (2017). Usability evaluation of mobile applications; where do we stand?. *In AIP Conference Proceedings, 1891*(1). https://doi.org/10.1063/1.5005389