COMP826

Mobile System Development

Student Name: Duo Tong

Student ID: 19075095

Lecturer: Roopak Sinha

Chandan Sharma

8th September 2020

The application of virtual traffic lights on android-based mobile system

1. Introduction

1.1 Topic: Mobile-based Virtual traffic lights

Traffic congestion has become an increasingly daunting issue, which affects the daily life of individuals around the globe. However, Virtual Traffic Lights (VTL), which is a promising self-organizing scheme of traffic control, has been designed to address this problem. This project proposes a prototype of the application of VTL on android-based smartphones. The VTL application is used to replace the infrastructure of traffic signals while can support vehicles to pass intersections more safely and faster based on the traffic regulations [12].

1.2 Background and objectives

Recently, both road safety and traffic congestion play an important role in the prominent concerns in society. As stated in [1], it is expected that traffic collisions would be the third cause of mortality by 2020, and the majority of them advents at intersections. Hence it is essential to find measures to alleviate the negative effects of environment and health. It is a cost-effective, promising but beyond the expected solution to alter the manner on the road where traffic is regulated by the utilization of intelligent transportation systems (ITS) in which the communication technologies are implemented to escalate traffic [11].

As an indispensable segment of ITS, VTL can enormously decrease the percentage of collisions by automatically assigning the priority of passing through junctions to vehicles. Besides, VTL is the key to enhance the traffic flow and significantly decline emissions from cars, which can promote the quality of life, efficiency as well as preserving the environment [11]. Due to the elimination of conventional traffic signals, VTL can vitally reduce the expenditure on installation and maintenance, as well [10].

Nonetheless, one of the premises of VTL is the complete implementation of the technology of vehicle to vehicle communication (V2V), which also named as Dedicated Short Range Communications (DSRC) technology. In this project, I aim to apply the concept of VTL as an application on the Android mobile system, which assumes that the hardware components in terms of GPS, map, wifi,

microcontroller, and display are already available.

1.3 Scope

Because of the limited time on development, I shrink the scope of the project that is listed as follows:

- With the premise of the achievement of DSRC and automatic driving, the slight severe network issues should not interrupt the leader election.
- The VTL application is based on the Android mobile system.
- Architectural design should indicate the future extensions of the product, which can result in higher compatibility and the ability of communication.
- Achieve communication among vehicles through relative hardware, including sensors, GPS, and wifi.
- Display self-location and the locations of neighbor cars on a map.
- Consistent with the principle of VTL, involving conflict detection, leader consensus, and election, the broadcast of VTL message and leader handover.
- Data warehousing and interface should be implemented as high security.

1.4 Roles

Within the development of a product, there are five roles that are outlined below. Nonetheless, due to the limitation of labors (one person acts as all roles), I only highlight three of them associated with system architect, UI designer, and developer.

- **System architect** is responsible for the comprehension of the needs from customers, the creation of system requirements, and the creation of a reasonable system, which can be defined as the separation of the system into components, interfaces as well as technologies and resources utilized during the period of design and implementation.
- **UI designer** is the person who aims at the generation of pleasurable, usable, and satisfying user interfaces. UI/UX design is extremely significant since it is the direct interaction with customers, which is the key to attract and maintain customers.
- Database administrator concentrates on the design, test, and maintenance of data.
- Developer focuses on the achievement of functions taking the performance into consideration.
- **Tester** whose major responsibility is to discover the problems and ensure the quality of the application before launching into the market from the perspective of functionality, usability, and

performance. Although testing plays an important role in software development, it will be assigned less labor due to the restriction of time.

2. Development processes

To accomplish the objectives, there are a number of development processes have been explored and widely utilized in the industry. Whereas I only emphasize three types of project management involving waterfall, spiral and scrum methodology.

2.1 Waterfall

The waterfall development process also named a linear sequence model, is one of the most stereotypical and conventional ways of software construction, especially for complicated and large projects. Its development lifecycle can be separated into seven linear stages in a sequence: conception, requirements, system design, manufacture, testing and maintenance [5]. In spite of the prevalent utilization in complex projects, the waterfall model has several inherent shortages such as inflexibility and high cost. More specifically, due to the rigid structure and long up-front planing time, it is inappropriate to apply the waterfall model in those projects confronted with the continuously updating requirements [5]. In addition, when requirements and system design involve numerous resources, then the following alterations can be pricey.

Thus, the waterfall model is suitable for those projects whose goals, requirements and technology stack are impossible to significantly vary during the process of development.

2.2 Spiral

The spiral method is a variant with diverse refinements of the waterfall process, which incorporates the characteristic of risk-driven, as well [3]. On the basis of the distinct risk patterns of projects, the spiral model instructs a team to apply the most appropriate method in terms of the waterfall, incremental model. It can combine most existing models as a special case so that it best fits a certain project. The process of the spiral can be usually divided into four steps containing planning, risk assessment, development and validation, and evaluation and planning for the next loop [3].

The main objective of the spiral is to decline the risk. As a result, it is the optimal choice when the critical project requests a high level of documentation as well as validation. In addition, it takes great advantages if the requirements are not clear and explicit. Nevertheless, in practice, this process is hardly applied because of unexpected outlay and time. Instead of implementation, it is mostly acted as

an instance of the way to comprehend the iterative development approach [3].

2.3 Agile and scrum

Because of the increasingly dynamic business requirements, organizations require the ability to develop, release and study from products in rapid and frequent cycles. Hence it is necessary for development processes to regularly deliver versions, include users and compile and prioritize their feedback, which motivates the occurrence of agile patterns [13].

Agile whose most popular methodology is scrum is a dynamic and iterative development approach. In other words, the agility of development refers to the inclusive and unexpected flexibility as well as leanness, which contributes to the capability of adapting and embracing changes and perceive the values of customers [14]. Opposed to the strict structure of the waterfall, scrum allows cross-functional teams have enough time to construct and release products for feedback, which means it has the ability to accommodate alteration and refinement of requirements in stages of the development process.

With additional parts of continuous delivery and feedback, scrum encourages developers to construct software by utilizing the concept of sprint [13]. During the period of development, the features and requirements can be furthered negotiated and discussed in sprints, as well. It also highlights and promotes cooperation among relative teams and the interaction between developers and customers [13]. The inclusion of feedback cycles inspires developers to respond to feedback and notify the customers about alterations within the updated version, which is beneficial to maintain users in the long term. Moreover, the participation of customers leads to a more satisfying product.

Through the nature of documentation, frequent iterations, early testing and the participation of customers, scrum can effectively address volatile and updating requirements [5]. As a consequence, it is really useful for continuously updating products. However, it is a thoughtless choice of budgets and timelines are extreme in the team since the attribute of flexibility can easily remodel original schedule, budgets or even engender inconsistency with current architecture [14].

2.4 The adopted development process

Although there are a number of types of development processes available, the option should be consistent with the actual situation of projects, which makes it easy to manage and keep the correct direction. I am inclined to opt scrum to construct the software due to the uncertainty of requirements

and its characteristic of flexibility. Additionally, it is more simple and easier to analyze the requirement at first, then built a minimum value product, and finally achieve the incremental functions according to their priority, which is in accordance with the feature of agile. The details of the VTL development process is as followings:

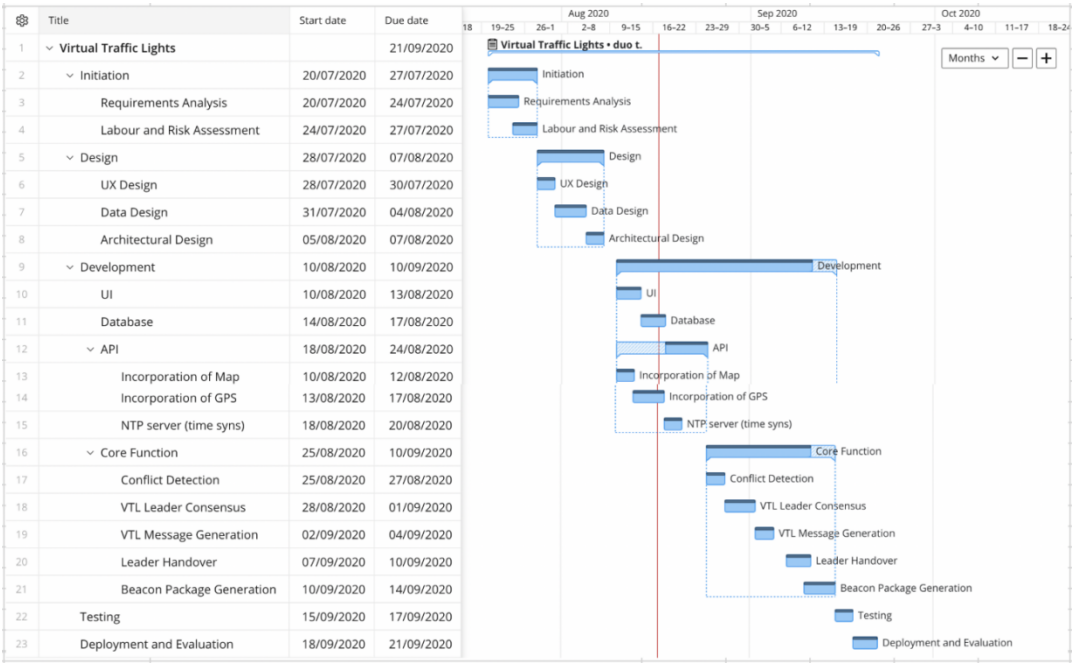


Figure2.4 Gantt chart for VTL development process

3. Related technologies

In terms of the development of applications, the first stage for developers is to opt for the appropriate platform, which can be categorized as three main classes including cross-platform, hybrid, and native.

Cross-platform can be considered as a WORA (Write Once, Run Anywhere) medium which means the same code can be deployed in multiple operating systems [7]. Therefore, it can significantly decrease the outlay of investment, maintenance and developing time but enlarge the proportion of users. Moreover, cross-platform applications only involve mobile elements [7], which means its rendering engine and UI is native. Its typical tools are React Native, Xamarin and Flutter.

Hybrid is a WORA platform, as well. Nonetheless, distinguishing from cross-platform applications, hybrid applications integrate web components into mobile elements. That is, the code base is invented through the utilization of standard web technologies such as HTML, CSS and JavaScript, but wrapped into a native container called WebView [2]. As a consequence, cross-platform applications exceed hybrid ones in performance. Its most prevalent tools are Ionic, PhoneGap and Corona.

Native is developed to be applied in a single programming language for a specific operating system, either IOS or Android [7], which engenders prolongation of developing time and increase of cost on development and maintenance. Reversely, amongst these three platforms, native applications take great advantages of performance due to the minimum of latency as well as faster CPU render time [7]. Additionally, native applications can fully access to the certain hardware of the latest device and software technologies.

Since the VTL application focuses on higher performance, I exclude the hybrid platform at first. Based on this, I prefer to select the optimal among Flutter, React Native and native, which are illustrated as below.

3.1 Flutter

Flutter is one of the most popular cross-platform mobile SDK, which was published by Google at the end of 2016 [15]. Instead of using web views or depending on OEM widgets of devices, Flutter renders view elements by directly utilizing its own rendering engine, which causes the possibility of resembling performance with native applications [15].

The overview of the Flutter system can be categorized as three parts, including framework, engine and embedder [8]. In the Dart framework, the most significant concepts are the multiple categories that are arranged in layers to alleviate complexity. Due to the design of layers in Flutter, each layer can be programmed relying on the complication of tasks. The engine is regarded as the medium to render the interaction of UI in the framework, while embedder is used to deal with the difference among platforms to ensure its compatibility. Another critical definition is that everything is a widget and a complicated widget consists of existing widgets, which are interface elements applied to generate user interface of apps.

3.2 React native

Another prevalent native cross-platform framework for building mobile applications is react native, which is an open-source created by Facebook [8]. Rather than presenting the content in the components of browsers, it implements business logic whereas convert UI-relevant code into commands related to native UI elements through the utilization of a JavaScript engine [4]. The fundamental structure of react native can be divided into four segments, involving react, JavaScript, bridge and natives [2]. More specifically, react native apps are developed by using JSX, which is the

integration of JavaScript and XML-esque markup. Next, under the hood, the native rendering APIs are invoked by its bridge in Object Swift and Java for IOS and Android, respectively [2]. Hence the react native applications render by adopting real UI mobile elements, which results in a similar appearance and attributes to other mobile apps. Besides, owing to the exposure to JavaScript interfaces for the platform of APIs, react native applications can invoke the platform features such as camera and GPS [4].

3.3 Native platform - Android SDK

Android SDK using Java is proposed to develop mobile applications by Google. It comprises of the platform, mediums, sample code, tutorials, debugger, libraries as well as documentation, which are necessary for the development of application based on Android mobile system [6]. Similar to the IOS, Android applies the WebKit browser engine, with touch screen, advanced graphics display and Internet access. It emphasizes the search function and has a better interface than iPhone [6]. It can also be said to be a single platform integrated into all web applications. Furthermore, the Android platform provides us with a lightweight storage class named SharedPreferences, especially suitable for saving software configuration parameters such as some default greetings, usernames and passwords [8]. Moreover, in contrast to other alternatives in terms of flutter and react native, it is the easiest approach to the functionality of devices such as camera, microphone and location [8]. Apart from the above benefits, the UI/UX in native applications is perfect, which makes users feel more comfortable.

The most attractive part is the openness and free service of the Android mobile phone system, which means Android is a platform that is completely open to third-party software so that developers have greater freedom in developing programs for them [6]. It breaks through the shackles of iPhone and other fixed software that can only be added to a few fixed software. At the same time, it is compatible with Windows Mobile and Symbian. Depending on the manufacturer, the Android operating system is provided to developers for free, which can save nearly 30% of the cost [6].

3.4 The final option - Android SDK

As mentioned above, in terms of performance and UI/UX, native applications are superior to flutter and react native platforms [8]. Additionally, the size of the native application is smaller than react native ones, which also takes advantage of overall performance.

Virtual traffic light applications require both high performance and low latency to guarantee

synchronization with the real situation as much as possible. Not only the display of the map but the communication with neighbor vehicles does desire outstanding performance. If not, the proportion of collisions could be climbed because of inaccurate information. UI/UX also plays a vital role in the development, since it will provide users a comfortable view experience, especially for the presentation of a real-time map.

As a consequence, regarding these two important factors, I eventually decide to opt for the native platform to implement the VTL application. In addition, considering the outlay (costing $99 to enroll for IOS developers) and the flexibility of platforms [9], in this project, I am not concentrated on deploying it on IOS but Android-based smartphones through Android SDK.

4. System design

4.1 Logical decomposition

To achieve the final objectives, this project exploits the approach called Model View ViewModel to design the architecture, which is revealed in Figure 4.1.1. Activity displays the layout and visual elements. Repository serves as the bridge between activity and web service or model, while the model is regarded as the domain object with a DB helper to interact with the database.

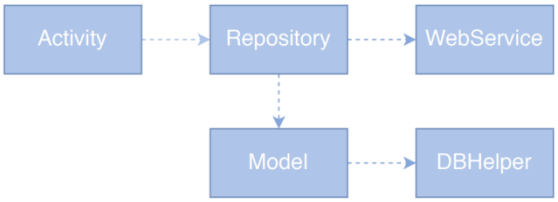


Figure4.1.1 MVVM architecture pattern

As illustrated in figure4.1.2, in the system, it is indispensable to invoke GPS, map and time synchronization. Also, the VTL module includes beacon packet generation, conflict detection, VTL leader consensus, VTL information creation and leadership handover.

a) Beacon packet generation and conflict detection

When approaching a junction, according to its own location and the neighbor locations acquired from beacon package generation, the vehicle estimates whether there are conflicts at the upcoming intersection. More specifically, based on the speed and location of the car that is near an intersection, the system will compute the time when will it reach the crossing, and then check if there is any other vehicle that will reach in the meantime. If so, a conflict is identified.

b) Leader election and consensus

If a collision is recognized, the closest vehicle to the junction on the same road will act as a cluster leader. Next, all cluster leaders have to consent to select a VTL leader among them for the crossing. Upon the VTL leader is detected, it broadcasts the leadership to cluster leaders that will respond with acknowledgment once acquiring the message. Acting as temporary traffic lights, the opted leader takes charge of the generation and announcement of traffic signal information. Others serve as slaves who comply with the traffic light information announced by the VTL leader. Noticeably, while leading traffic, the elected leader presents as red light, which means it must stop at the junction.

c) VTL message generation

As long as a VTL leader is selected, it decides the duration of the right of way for each approaching direction, which can be configured according to some parameters in terms of the quantity of vehicles, the degree of congestion. The leader assigns the information of traffic signals to other vehicles. However, when it recognizes that no more vehicles attempt to cross the junction on the road with green lights, the current situation is broken, then the green light will be distributed to the next road.

d) Leader handover

When the green signal is presented in the leader's lane, to maintain the normal function of virtual traffic lights, a new VTL leader has to be selected based on two principles: the current leader transfers the leadership to one of the automobiles stopped at the junction before red lights; otherwise, the new leader election is executed again.

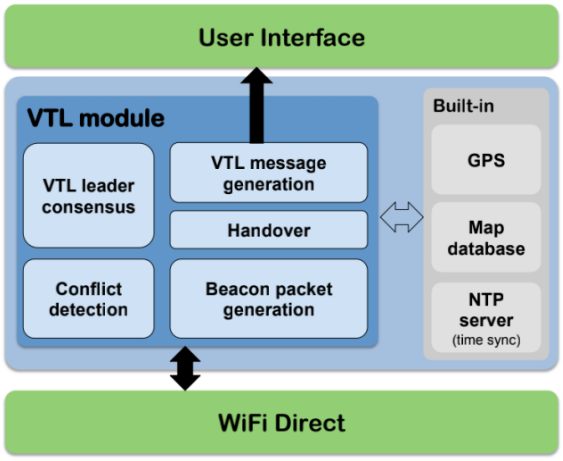


Figure4.1.2 The architecture of VTL

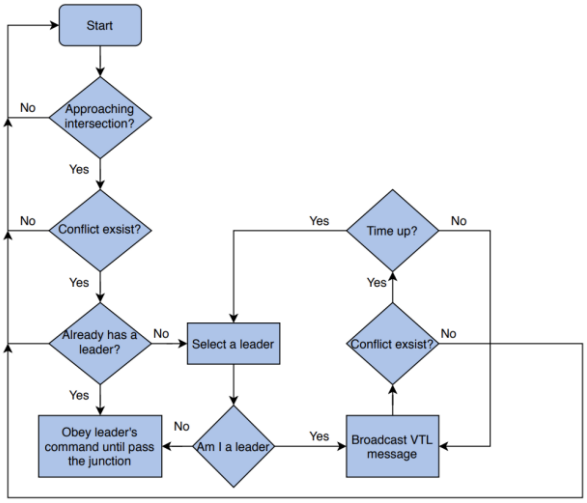


Figure4.1.3 Flowchart of VTL algorithm

Domain classes

BeanPacketGeneration is utilized to acquire the location, lane and role of vehicles.

ConflictDetection is the class to identify the existence of collision according to the distance to the intersection and the reaching time.

LeaderElection is used to select a leader and leadership handover when leading time is over.

VTLMessageGeneration functions to produce VTL message and supervisor of leading time and conflict detection.

Map displays the location and neighbour environment.

GPS can be used to offer the concise location of vehicles.

Sensor can acquire the speed, which is indispensable for conflict detection.

NTP server ensures the time synchronization.

Figure4.1.4 UML diagram

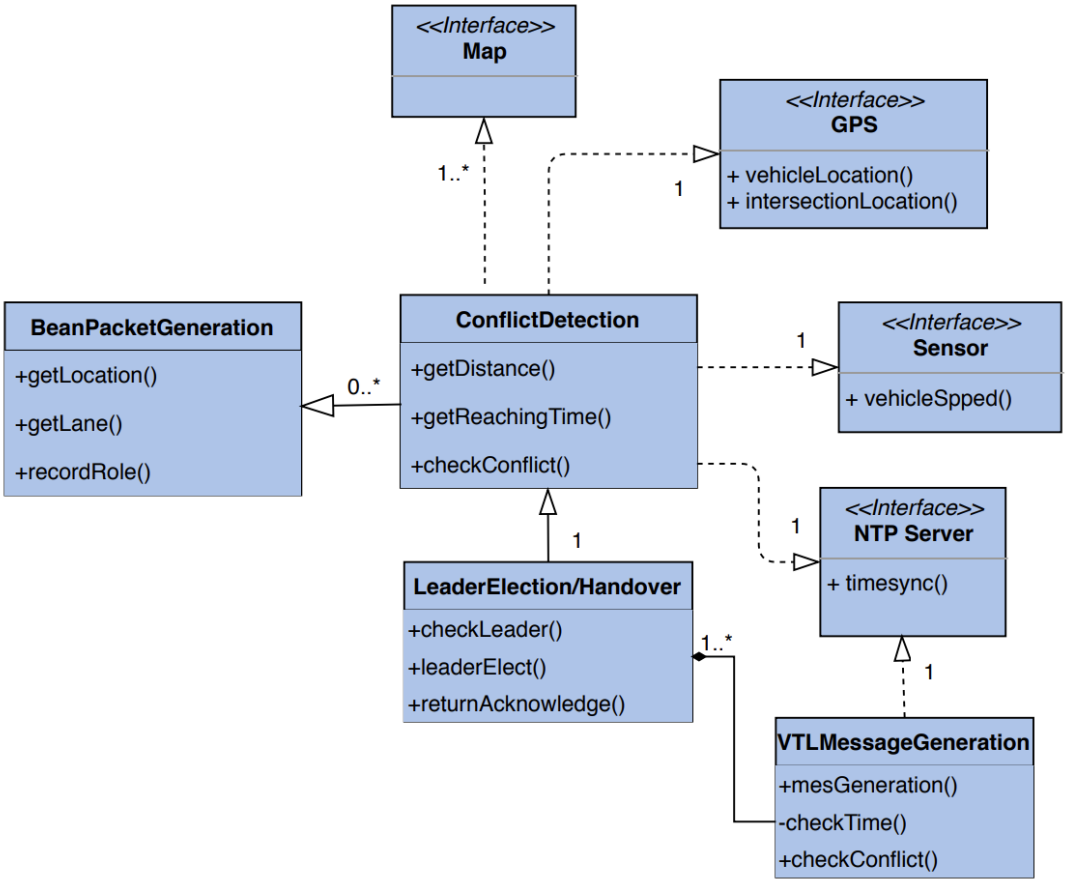
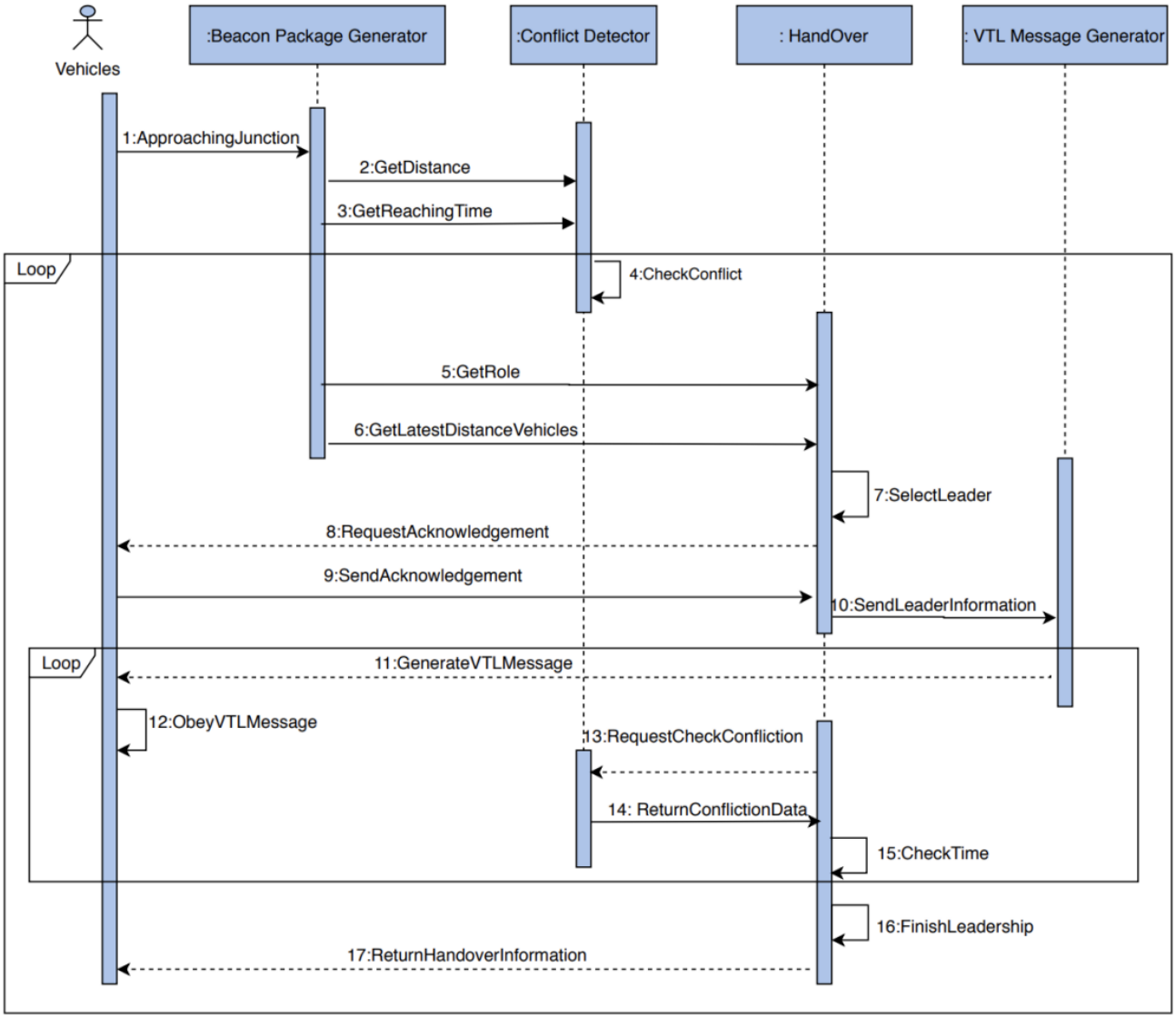


Figure4.1.5 Sequence diagram



4.2 Runtime characteristics

Usability describes how well users can use a product to accomplish defined tasks effectively, efficiently and satisfactorily. According to [10], Nielsen defined usability as five attributes including efficiency, satisfaction, ease to use and remember as well as errors. In this project, the presentation of UI is brief, explicit and comprehensive. Its details are exhibited in Figure 4.2.1 (navigation graph) and Figure 4.2.3 (UI prototype), and the definition of usability model (PACMAD) is shown in Figure 4.2.2.

Performance refers to the ability to achieve its goals in terms of response time and throughput. In general, the architecture, code (especially the logic) and the amount of function massively affect the performance of the application. For VTL app, I will deploy the native platform (AndroidSDK) whose

performance is outperforming. I will also keep GUI simple and optimize code (such as loops optimization, API utilization and code review) to enhance its performance. Eventually, I would apply static tests to analyze and evaluate the performance by using Raxis.

Figure 4.2.1 Navigation graph

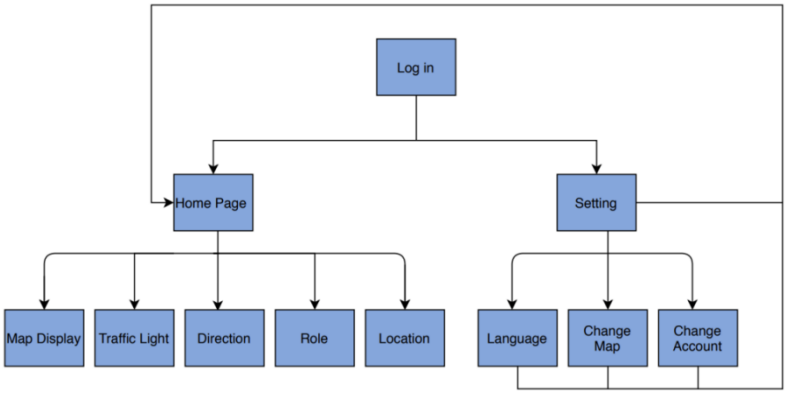


Figure 4.2.2 PACMAD

	User	Task	Context
Effectiveness	1. When users are approaching intersection, they can get the outcome of conflict and leader election. 2. After leader election, users can get traffic signals information from the VTL leader to pass or stop 3. The VTL leader must be in the red light status until finishing its leadership or no more conflicts. 4. When the leader finish its responsibility, the next leader should be elected if there is any conflict at the junction.		
Efficiency	1. When users are approaching intersection, they can get the correct traffic signals instructions instantly. 2. The leadership (handover) must not be interruptted when there is conflict.		
Satisfaction	High performance of acquiring traffic information.		
Learnbility	It is easy to use for users because the UI is simple, which only involoves two pages.		

Figure 4.2.3 User interface prototype

Figure 4.2.3.1 Log in page Figure 4.2.3.2 Register page

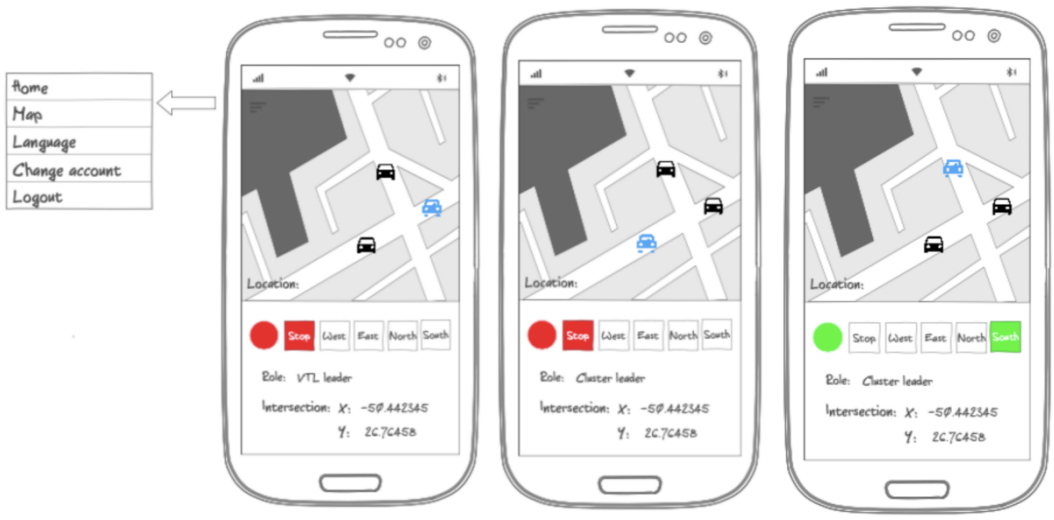


Figure 4.2.3.3 Home page - display traffic lights information

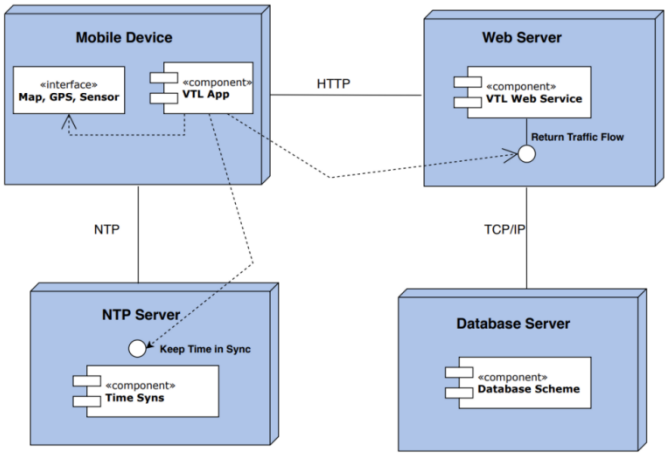
4.3 Physical characteristics

Sensor capability can be considered as the ability for devices to support the required sensors and GPS, which are used to obtain the location and sense the speed of vehicles, respectively.

Time synchronization is also regarded as a necessary section since asynchronized time between vehicles in the same region could cause the appearance of traffic collisions.

Stable and high-speed network is a significant physical feature, as well. Unstable and relatively poor network engenders the latency of communication among VTL leaders and other vehicles, which could result in the occurrence of traffic accidents.

Figure 4.3.1 Deployment diagram



Mobile device is the most significant node in deployment. It plays an indispensable role in the interaction between users and the system. More specifically, it contains VTL applications as well as three interfaces such as map, GPS and sensor. Additionally, the app connects to the web server (via HTTP) and NTP server (through NTP) to acquire the traffic flow and keep time in synchronization, respectively.

Web server is the segment of the system offering the traffic information for the application.

Database server involves the database scheme, which functions as a storage service for the VTL application.

NTP server is the medium to achieve time synchronization for all users.

5. Evaluation

5.1 Register

Initial state: no account
Result: the system requires to register to use the application

Condition: application and stable network available.

5.2 Login

Initial state: having account
Result: entering email and password to log in
Condition: application, account and stable network available.

5.3 Junction identification

Initial state: approaching an intersection.
Result: the system can effectively and correctly detect intersections forward.
Condition: GPS, sensor and stable network available.

5.4 Conflict detection

Initial state: approaching an intersection.
Result: the system can effectively and correctly detect conflicts.
Condition: GPS, time synchronization, sensor and high speed and stable network available.

5.5 No leader election

Initial state: approaching an intersection but no conflicts.
Result: vehicles pass through the conjunction directly
Condition: GPS, time synchronization, sensor and high speed and stable network available.

5.6 Cluster leader election

Initial state: no leader but have conflicts.
Result: the closest vehicles in each road is elected as the cluster leaders.
Condition: GPS, time synchronization, sensor and high speed and stable network available.

5.4 VTL leader consensus

Initial state: cluster leaders have been elected.
Result: the VTL leader is elected through the consensus amongst cluster leaders.
Condition: GPS, time synchronization, sensor and high speed and stable network available.

5.7 The status of VTL leaders

Initial state: acting as the VTL leader
Result: The VTL leader performs as the red light status (stopping beside the road).
Condition: GPS, time synchronization, sensor and high speed and stable network available.

5.8 VTL information generation

Initial state: vehicles are waiting for passing at the junction and the VTL leader has been elected.

Result: waiting vehicles get the traffic light information to decide to move ahead or stop.
Condition: high speed and stable network available.

5.9 VTL information generation

Initial state: approaching intersection and collision detected
Result: vehicles should be at red light status
Condition: GPS, time synchronization, sensor and poor network available.

5.10 Leadership handover

Initial state: leading time is almost over for the current VTL leader while still have conflicts.
Result: the new leader is selected to continue to control the traffic.
Condition: GPS, time synchronization, sensor and high speed and stable network available.

5.11 Process over

Initial state: no conflicts
Result: vehicles continue moving without VTL instruction.
Condition: GPS, time synchronization, sensor and high speed and stable network available.

5.12 The compatibility of different size screens

Initial state: no
Result: the display on each page is correct and comprehensive
Condition: smart phones with different size screen available.

5.13 The compatibility of different version of Android

Initial state: no
Result: the display and functions on each page are correct and comprehensive
Condition: smart phones/tablets with different version of Android operating system available.

References

[1] A. Bazzi, A. Zanella, B. M. Masini, and G. Pasolini, “A distributed algorithm for virtual traffic lights with IEEE 802.11 p,” In 2014 European Conferenceon Networks and Communications, pp. 1–5, 2014.

[2] B. Eisenman, “Learning react native: Building native mobile apps with JavaScript,” O’Reilly Media, 2015.

[3] B. W. Boehm, “A spiral model of software development and enhancement,” Computer, vol. 5, pp.61–72, 1988.

[4] C. Rieger, and T. A. Majchrzak, “Towards the definitive evaluation framework for cross-platform app development approaches,” Journal of Systems and Software, vol. 153, pp. 175–199, 2019.

[5] F. Ji, and T. Sedano, “Comparing extreme programming and Waterfall project results,” In 2011 24th IEEE-CS Conference on Software EngineeringEducation and Training, pp. 482–486, 2011.

[6] J. Steele, and N. To, “The Android developer’s cookbook: building applications with the Android SDK,” Pearson Education, 2010.

[7] K. Shah, H. Sinha, and P. Mishra, “Analysis of cross-platform mobile app development tools,” In 2019 IEEE 5th International Conference for Convergence in Technology, pp. 1–7, 2019.

[8] L. Dagne, “Flutter for cross-platform App and SDK development,” 2019.

[9] M. Nakamurakare, W. Viriyasitavat, and O. K. Tonguz, “A prototype of virtual traffic lights on android-based smartphones,” In 2013 IEEE InternationalConference on Sensing, Communications and Networking, pp. 236–238, 2013.

[10] R. Harrison, D. Flood, and D. Duce, “Usability of mobile applications: literature review and rationale for a new usability model,” Journal of InteractionScience, vol. 1, 2013.

[11] R. Sinha, P. S. Roop, and P. Ranjitkar, “Virtual Traffic Lights+ A Robust, Practical, and Functionally Safe Intelligent Transportation System,” Trans-portation research record, vol. 1, pp. 73–80, 2013.

[12] R. Zhang, F. Schmutz, K. Gerard, A. Pomini, L. Basseto, S. B. Hassen, and O. Tonguz, “Virtual traffic lights: System design and implementation,”In 2018 IEEE 88th Vehicular Technology Conference, pp. 1–5, 2018.

[13] S. Klepper, S. Krusche, S. Peters, B. Bruegge, and L. Alperowitz, “Introducing continuous delivery of mobile apps in a corporate environment: A case study,” In 2015 IEEE/ACM 2nd International Workshop on Rapid Continuous Software Engineering, pp. 5–11, 2015.

[14] T. Dingsyr, S. Nerur, V. Balijepally, and N. B. Moe, “A decade of agile methodologies: Towards explaining agile software development,” 2012.

[15] W. Wu, “React Native vs Flutter, Cross-platforms mobile application frameworks,” 2018.