Vietnamese German University
Department of Electrical & Computer Engineering study program

Frankfurt University of Applied Science
Faculty 2: Computer Science and Engineering

VEHICLE SYSTEM MANAGEMENT FOR AUTOMATIC TOLL APPLICATION

BY

DAO MINH HIEU

Matriculation number: 1184776

First Supervisor: Dr. THAI TRUYEN DAI CHAN

Second Supervisor: M.Sc TRAN QUANG NHU

BACHELOR THESIS

Submitted in partial fulfillment of the requirements
for the degree of Bachelor Engineering in study program
Electrical & Computer Engineering,
Vietnamese German University, 2021

Binh Duong, Vietnam, January 2021

Vehicle System Management For Automatic Toll Application

Approved by

Dr. THAI TRUYEN DAI CHAN, First Supervisor

M.Sc TRAN QUANG NHU, Second Supervisor

Thesis Committee

Disclaimer

I hereby declare that this thesis is a product of my work, unless otherwise referenced. I also declare that all opinions, results, conclusions, and recommendations are my own and may not represent the policies or opinions of Vietnamese-German University.

Dao Minh Hieu

Acknowledgement

I would like to express the most profound appreciation to my supervisor, Dr. Thai Truyen Dai Chan, for the continuous support of my study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance has helped me in all the time of researching and writing this thesis.

Abstract

Despite of having many rules and regulations in for transportation traffic, it is still to this day a chaotic environment sometimes. Therefore, finding traffic solution is a work of art in civil engineering world. However, the technology of internet and navigation solution is evolve every day, and this opens many opportunities to increase the quality of current traffic status or even solve problems that people do not even know it existing. It is such a fascinating time we are living where electronics components embedded with latest technology are affordable for hobbyist, and it means more beautiful mind can involve in finding delicate solutions, and current manual toll station is one of the problems that need a better solution. GPS technology and Google database now reach a level that is precisely, smart and fast enough on a portable smartphone to change the traditional toll to a new form by removing all of the obstacle throughout fee paying process at these station and to make it environmental friendly. This project is one of the solution by realizing an on the go payment virtual toll station, replace all the work of manual toll to a smarter automatic one.

Keywords: GPS, automatic, on the go payment

Contents

1 Introduction

1.1 Purpose

This report is submitted for partial fulfillment of the Bachelor Thesis at the Vietnamese – German University. The report demonstrates from general to detail about web application for tracing and checking in food safety and quality

1.2 Audience

The intended audience for this project and report are Dr. Thai Truyen Dai Chan and MSc.Tran Quang Nhu, who will use it as a basis to evaluate and determine a portion of the grades for the Bachelor Thesis.

1.3 Motivation

Unnecessarily construction or human interaction are the reasons pushing technology forward in order to replace them with an intelligent automation system, and traditional manual toll is not different. The availability of IoT solutions are now more and more accessible and secured enough to develop such a system that is money transaction related. The thriving of E-wallet companies in Vietnam in recent year has shifted the purchasing habit of large portion of Vietnamese consumer, hence raising liquidity of these virtual credits. This phenomenon gains trust on the banking system allowing converting real money to virtual or vice versa easier than ever. However, none of road construction companies applied this result to their fee collecting activity. Although, some of them have chosen alternative solution such as using RFID card, but the main purpose of automation is not applied when the driver interaction is still involved during the procedure of paying fee. Therefore, taking advantage of E-Wallet characteristic and GPS technology, this project will enable user to pay road fee securely on the go without any interaction involved, and the solution will turn all physical construction digitally and place them on any road position in Vietnam via few simple configurations on database. This solution could replace the current road toll system or its alternative entirely.

1.4 Objectives

This thesis is created with the objectives of research, design, and implementation of a web application for monitoring and tracing of food safety and quality. The finalization of this project can be used to manage, monitor the quality of food from the beginning of the cultivation stage to the delivery process. A website platform is developed to collect and analyze the data received from IoT devices. Hence, all the analyzed data is displayed as a graph or exported as a table for users to easily access, control, make decisions on the quality of food.

1.5 Outline

- Chapter 2: Stating the problems and arguments for the general direction of the project
- Chapter 3: General description of each system and technology specification
- Chapter 4: Detailed description on every aspect of the project
- Chapter 5: Discussion about the limitation and performance for the system and introduce the security communication scheme using in the system
- Chapter 6: The final conclusion of the thesis report

2 Research background

2.1 Introduction

Road toll is not a modern definition, it existed at least 2700 years ago before any steam engine machine invented, where travelers had to pay to use Susa-Babylon highway under regime of Ashurbanipal [1]. Till this day in Vietnam, a toll is introduced to finance the transport infrastructure, usually national road or highway, but the general process stay the same: stop driving, pay the fee and continue driving. Although there is a solution improving this process by using a toll gate automation [2], this solution only reduce the time and labor cost of manual toll booth, but the gimmick is the same, and the drivers are still be interrupted along their driveway. This project will provide a more efficient and delegate solution by removing all the listed step throughout fee payment activity and make it truly automatic when involving the GPS technology in the process.

2.2 Current toll problem

Current manual toll systems come with many disadvantages in form of cost, efficiency and longevity of the highway in Vietnam.

The cost of a manual toll may vary depending on the construction size, the location, the geographical and the labor cost. The cost of a single national road toll booth on Quoc Lo 1A is around 7.3 billion VND (314,523 USD) [3] connecting from Northern to Southern of Vietnam, and there are 40 tolls placed along this main road for controlling and fining smaller portions of this road. This can add up to thousands of billions VND when including maintenance fee.

Every time a vehicle decelerates to stop at the toll to pay the fee and accelerates to get back on track, the wheels apply a friction to the concrete surface, which can be a static friction or a sliding friction at the contact point. The greater the weight and initial velocity (especially on highway) of the vehicle the greater the friction.

$$F = Mg\mu$$

$F : Friction\ at\ the\ contact\ point\ of\ wheels\ and\ the\ road$

$M : Vehicle\ mass$

$g : Gravitational\ acceleration$

$\mu : coefficient\ of\ friction$

And this force deals a massive amount of damage to the road surface. The reason for building these tolls is to pay back the capital of constructing the road for the contractor but at the same time destroying the road itself. Therefore, this action not only reduces the lifespan of the road surface but also raises the cost of maintenance higher than it should be.

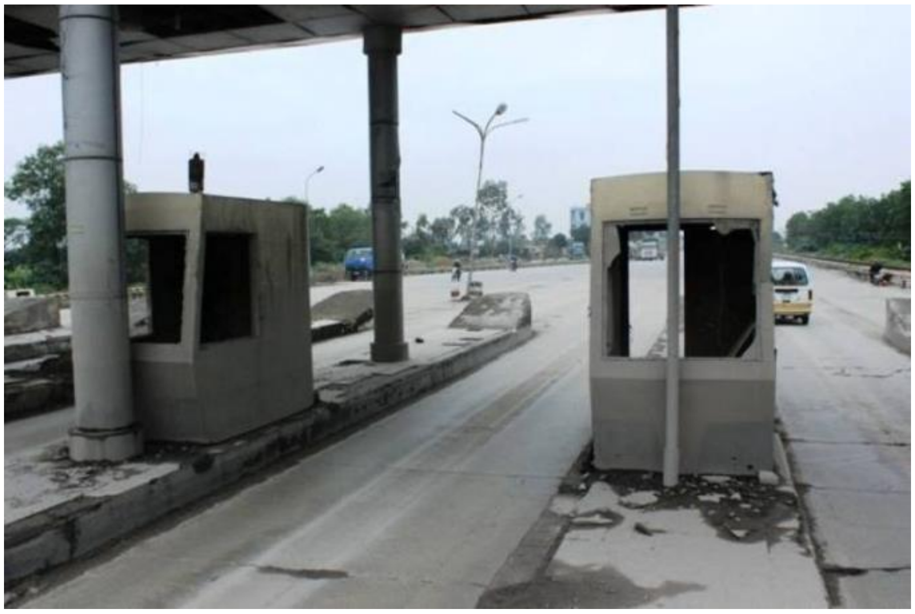


Figure 1 Abandon toll booth with degrade road condition in Ha Nam

The environmental impact will be reduced greatly with this project by saving tons of building materials for a static toll. When these structures are no longer needed since the contractors already earned enough money for the responsible road, they got abandoned in the middle of the highway, blocking the traffic with unnecessary space taken.

Finally, the time it took to stop and pay the fee is around 30 seconds. This time interval will be neglect able for a casual driving speed, but for the highway, where the speed could reach up to 120km/h, the automatic toll will let drivers save up to 1 km of trip distance.

The purpose of this project is to realize an automatic toll system which neglects all the listed downside from manual toll systems and lets users manage their payment in road usage.

3 System architecture and interfaces

3.1 E-wallet application

Electronic wallets are increasing their popularity in recent years in Vietnam. Moreover, due to the spreading rate and serious health impact of CO-vid19 virus, using cash is less of a choice when compared with electronic currency alternatives. Vietnamese people are slowly changing their habit and gaining trust for these transaction services, and this opens many possibilities to have more payment methods linked with these electronic wallets.

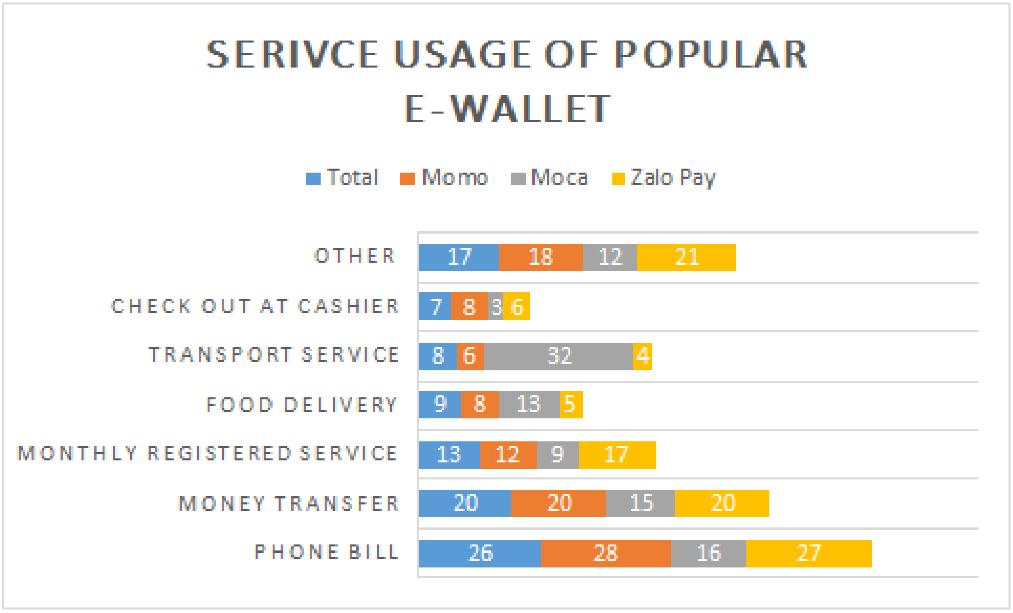


Figure 2 Service usage of popular E-Wallet in Vietnam [4]

For this reason, the payment system for road fee will be enabled through an electronic wallet mobile application linked with their bank account.

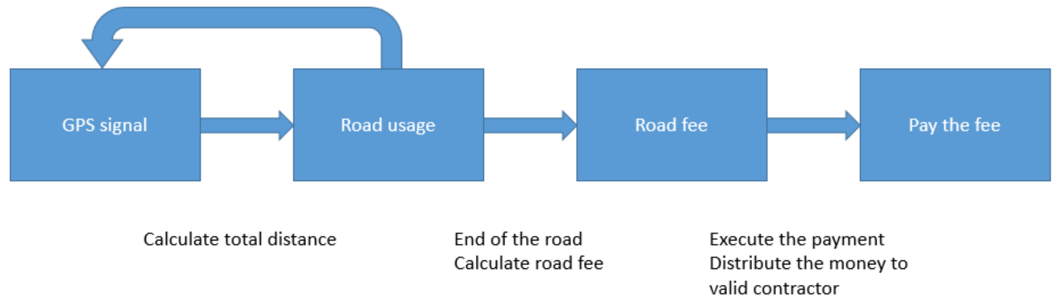


Figure 3 Block diagram of Payment system

The road fee will be based on the calculated road usage with vehicle mass, instead of paying a same fee for every distance on manual tolls. This strategy will save a lot of money for the end user by its integrity aspect and earn the capital back for contractors faster because this system can be implemented on every road and takes no extra fee to build a constant structure. All the processes will run in the background with no supervision needed.

With the described characteristics, this sub-system is designed to replace the roll of road fee collectors at the toll station and placing any station on anywhere at will.

3.2 Central server

For every Iot application, developing a dedicated server for a specific application is mandatory. The central server will handle all user data request from E-Wallet application. It is designed in such a way which can handle multiple device at a same time with minimal data loss. To realize this, this project will decide to use the Socket library with Sqlite3 database manager library to prototype the behavior of central server. By using these basic tools, the project can easily demonstrated the secured data communication due to the vast customization ability of a Python core library. Furthermore, the problem with other multifunctional API is the controllability on the behavior with unnecessarily feature that only take up resources with respect to the native library.

3.3 Vehicle digital registration certificate (VDRC).

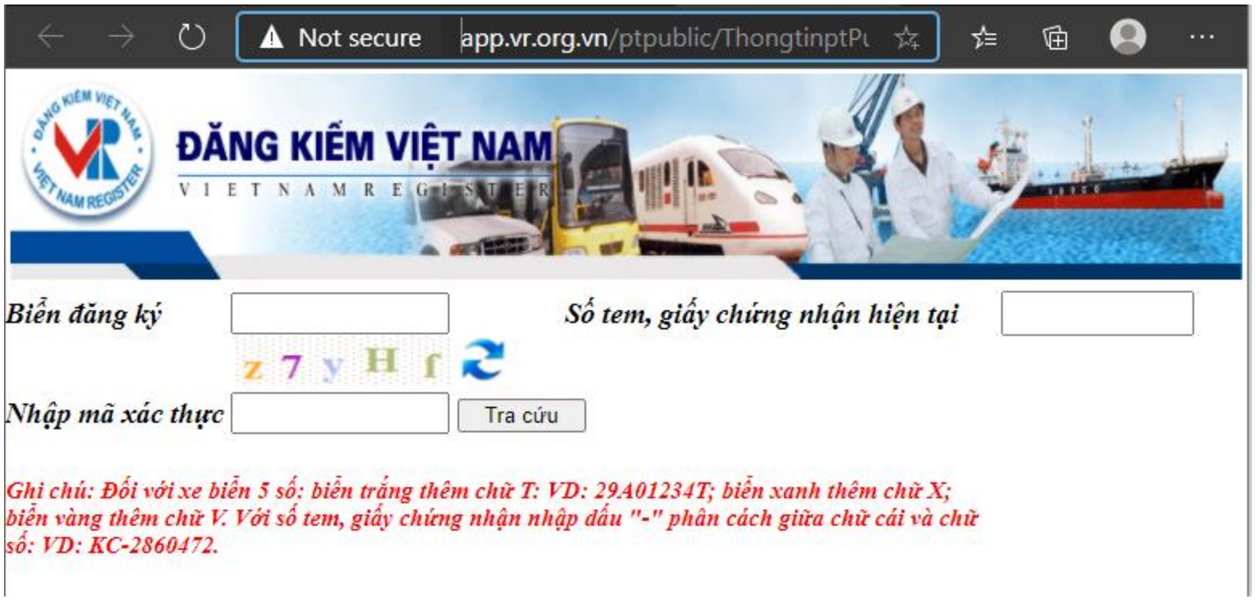
The ideal of registration the vehicle for the automatic toll system taken from a scenario, which you have just brought a new car and registered with the government, in order to legally operating it on the street. If this project is applied in the real life, newly bought car should be required to install a device along with its basic information to serve as a digital vehicle registration certificate (VDRC). Additionally, this device should work on legally operated vehicle to expand the coverage of this system.

While operating a vehicle on the street, different vehicle has different impact. Therefore, the diversification of road fee per kilometer is necessarily, in order to bring equality to drivers. For this requirement, the automatic toll system need to introduce a device which can store vehicle data, especially vehicle mass, mounted permanently on the vehicle itself. Moreover, E-Wallet does not have to be attached to any vehicle as long as the user does not operate that vehicle, just like a normal wallet only belong to its owner, not the car.

The VDRC shall be equipped with Bluetooth communication to interact with E-Wallet. There are 2 main interactions: register vehicle and register driver.

When a VDRC is first mounted on the vehicle, it need to have a procedure to inform itself about the vehicle to later on transmit (detailed description will be on Chapter 4.1.6 and 4.3.1). Therefore, the road fee shall be calculated proportional to the vehicle mass. This is the mandatory reason for the existence of VDRC. Mass is a distinct characteristic belong to the vehicle and not the E-Wallet, and if the system decided to manage this information as a part of E-Wallet, the complexity and amount of duplicated data will exponentially increase. Since such database that manages all legally imported and operated vehicles existed and provided partially publicly by Vietnamese Vehicle Registration Agent [5], VDRC shall take advantage of this availability to update for itself, or more precisely, being updated by the E-Wallet since this would reduce the complexity of VDRC development.

Smartphones in general with embedded Android OS have become unimaginably powerful for such application like gathering and analyzing GPS information or 4G connectivity (or 5G in near future). Therefore, it is pointless and time consuming to reinvent and refurbish these aspects on VDRC for this project just to prototype the ideal of an independent device that can operate on its own on some circumstances. Furthermore, implementing these feature along with the development of E-Wallet application reduced effort, since it is within one centralized development environment.



Biển đăng ký

Số tem, giấy chứng nhận hiện tại

Nhập mã xác thực

Ghi chú: Đối với xe biển 5 số: biển trắng thêm chữ T: VD: 29A01234T; biển xanh thêm chữ X; biển vàng thêm chữ V. Với số tem, giấy chứng nhận nhập dấu "-" phân cách giữa chữ cái và chữ số: VD: KC-2860472.

Figure 4: Vehicle Registration Certification Agent website

3.4 Socket communication

Socket communication, or more specifically, network socket communication using in this project configured for TCP connection. Central server will initiate as server socket accepting client device, which is the E-Wallet. The majority communication activities are start the connection, client sending request to server, server response to client after request processed and connection closed. The request and response message will be divided into 2 parts, header and body using JSON structure in Figure 5.

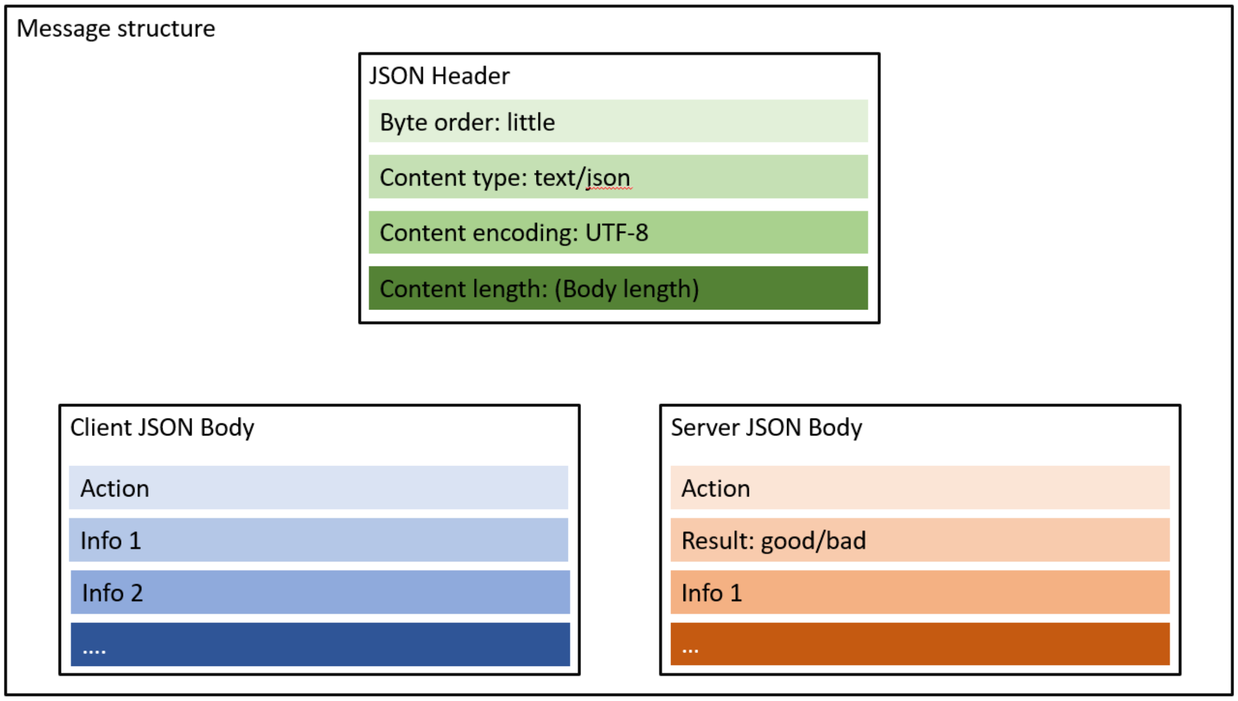


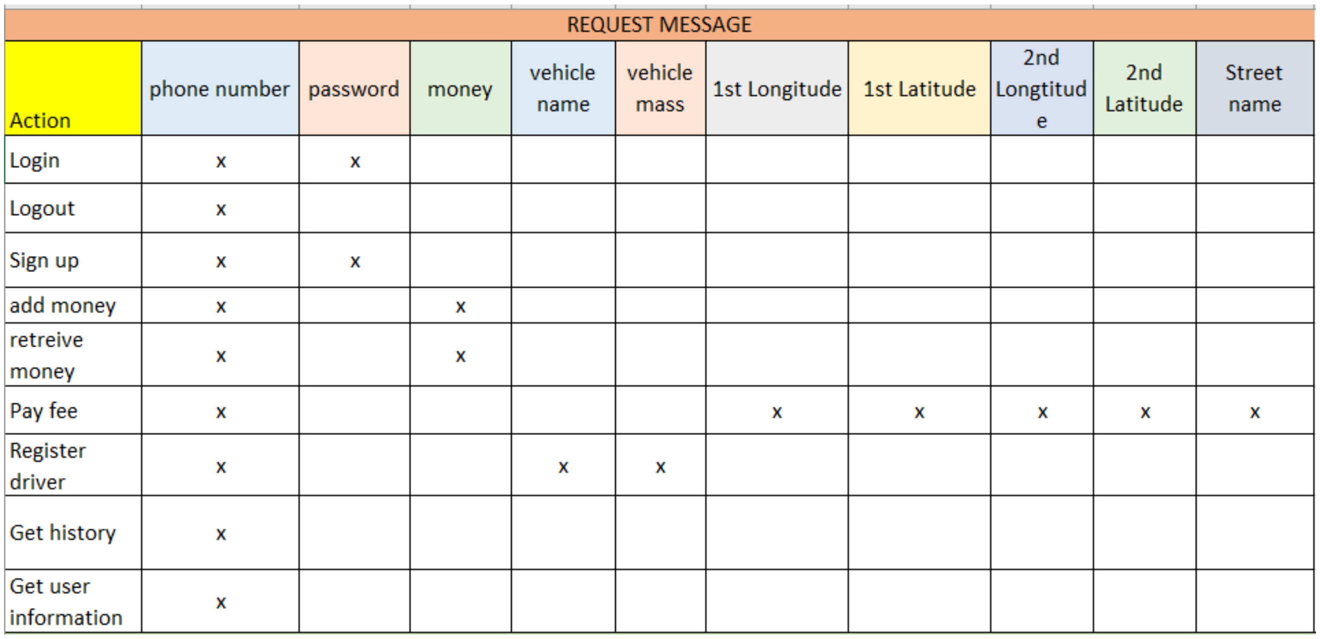
Figure 5: Message format for socket communication

The headers of both request and response message types are the same including Endianness byte order, JSON content type, encoding rule and length of body. Since the E-Wallet application and the Central server is created to work with each other, the information in the Header shall be check with fixed condition except the content length, since this value varied along with the body content, and this checking step will ensure the quality of the communication.

The request body of client device will consist of the Action field, which declared the type of request that the server need to process, along with related information for this request. For example, login request will have Action is login, phone number and password are 2 more information for the server to proceed logging in.

The server side on other hand, the response body will have the Result field to inform the client the status of the request and the Action field of regarded status. The existing of Action field in the response message structure is to ensure the client can distinguish which response of which request, in case there are multiple requests sent from it. Beside from Result and Action field, the response message body also consists information fields matched with the Action.

The request and response information field shall be listed in Figure 6 and 7.



REQUEST MESSAGE										
Action	phone number	password	money	vehicle name	vehicle mass	1st Longitude	1st Latitude	2nd Longtitude	2nd Latitude	Street name
Login	x	x								
Logout	x									
Sign up	x	x								
add money	x		x							
retreive money	x		x							
Pay fee	x					x	x	x	x	x
Register driver	x			x	x					
Get history	x									
Get user information	x									

Figure 6: Request message scenarios

	RESPONSE MESSAGE													
Action	result	phone number	password	budget	vehicle name	vehicle mass	login status	1st Longitude	1st Latitude	2nd Longtitude	2nd Latitude	Street name	money transaction	time
Login	x	x		x	x	x	x							
Logout	x													
Sign up	x													
add money	x													
retreive money	x													
Pay fee	x													
Register driver	x													
Get history	x	x			x	x		x	x	x	x	x	x	x
Get user information	x	x	x	x	x	x	x							

Figure 7: Response message scenarios

3.5 Bluetooth communication

3.5.1 Bluetooth architecture

For this project, the Node MCU ESP32S chip of the VDRC support both Classic Bluetooth and Low Energy Bluetooth technology. The Bluetooth architecture of ESP chip contain 2 stacks: controller stack and host stack. Modules such as PHY, Device Manager, Baseband, Link Manager, Link Controller, and HCI are included in controller stack. On other hand, the host stack support L2CAP, GATT, ATT, GAP, SMP and SDP profiles providing the interface to application layer to operate Bluetooth service. The VDRC uses the default Bluetooth setting for this chip, which is using VHCI (software-implemented virtual HCI interface in Figure 8) to communicate between host stack and controller stack, and the profile used is SPP since it provide many standard communication.

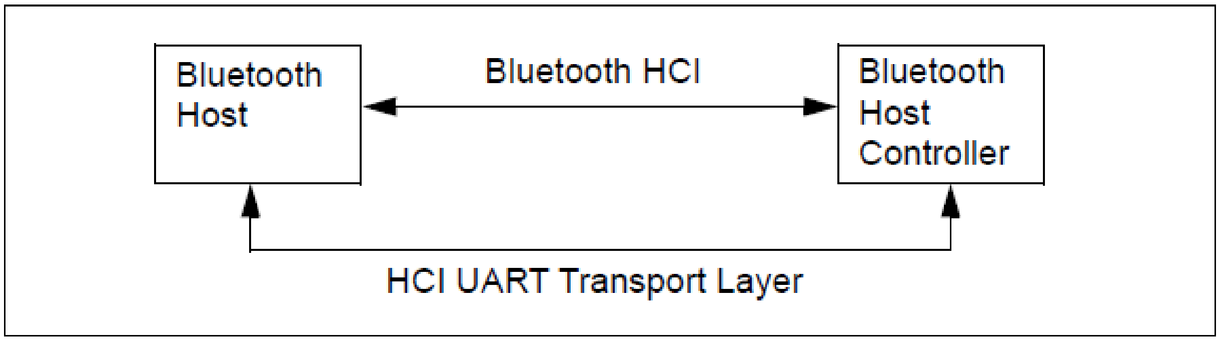


Figure 8: HCI UART transport layer

3.5.2 SPP Profile

Based on ETSI 07.10 and the RFCOMM protol, SPP stands for Serial Port Profile can emulate RS-323 serial message, and it is convinient when using with Serial API from arduino, and RFCOMM protocol is used on mobile device side. The Serial Port Profile set the necessary requirements for Bluetooth devices to act as a wrieless serial cable connections using RFCOMM between two peer devices. The requirements are detailed in terms of services for applications, and by setting the features and procedures that are required for exchanging between Bluetooth devices.

3.5.3 Connection Process

The VDRC will be an Acceptor [6], which initiates itself and form a connection to another device. This done by broadcasting inquiry request at radio frequency of 2.4 Ghz while the mobile device, an Initiator, will listen to this via discovery mode and response if chosen by the user, and two bluetooth devices shall initate the connection.

3.5.4 Bonding and Pairing

When connection is established between the mobile device, they will proceed pairing sequence to creates bonds. This process in Figure 9 will go through 4 phases: Pairing feature exchange, short term key generation, long term key generation, transport specification key deistribution

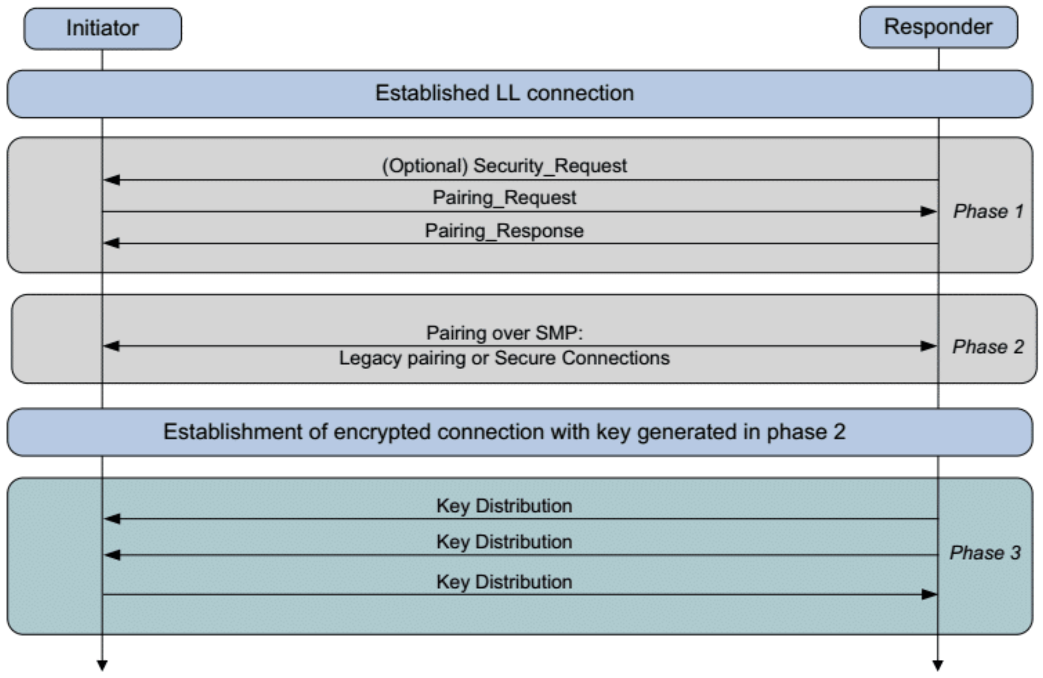
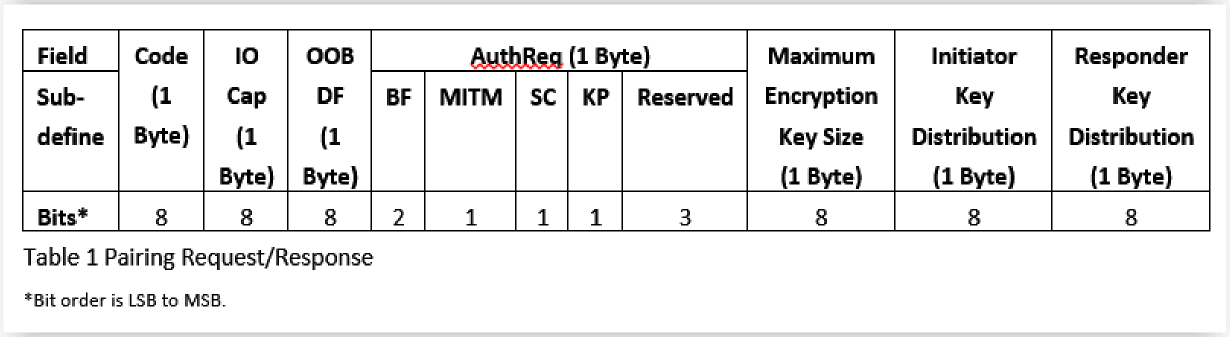

Initiator
Responder
Established LL connection
(Optional) Security_Request
Pairing_Request
Pairing_Response
Phase 1
Pairing over SMP:
Legacy pairing or Secure Connections
Phase 2
Establishment of encrypted connection with key generated in phase 2
Key Distribution
Key Distribution
Key Distribution
Phase 3

Figure 9: Bluetooth pairing sequence

Pairing is the exchange of security features including feature such as Input/output (IO) capabilities, requirements for Man-In-The-Middle protection, etc. The exchange of pairing information between the mobile device and the VDRC is done through the Pairing Request and Pairing Response packet in Figure 10.



Field	Code (1 Byte)	IO Cap (1 Byte)	OOB DF (1 Byte)	AuthReq (1 Byte)					Maximum Encryption Key Size (1 Byte)	Initiator Key Distribution (1 Byte)	Responder Key Distribution (1 Byte)
Sub-define				BF	MITM	SC	KP	Reserved			
Bits*	8	8	8	2	1	1	1	3	8	8	8

Table 1 Pairing Request/Response

*Bit order is LSB to MSB.

Figure 10: Pairing request/response elements break down

3.6 GPS Technology

Global Positioning System (GPS) is based on satellite radio navigation service provided by the US Space Force [7]. By using the information from an on-board atomic clock, coordinates and a current status, the GPS satellite projects a radio signal from medium Earth orbit region (approximately 20,000 km above the ground) to a client device at the speed of light. The GPS embedded device then uses this time information to calculate the distance of itself to at least 4 satellites [8].

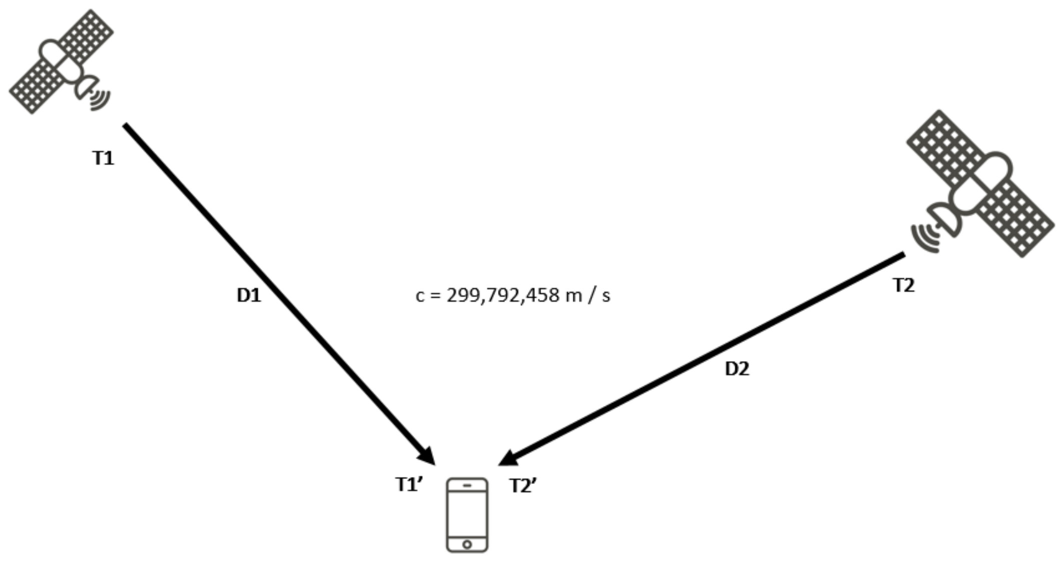


Figure 11 Global positioning method of GPS technology

$$|D| = c * \Delta T = c * (T' - T)$$

$D :$ *Distance from a satellite to the GPS device* (m)

$T :$ *Time instant when the GPS signal leave the satellite* (s)

$T' :$ *Time instant when client device receive the GPS signal* (s)

$c :$ *Speed of light* (m/s)

By knowing the distances to 4 determined points in space, a GPS device coordinate(x, y, z)can be located with the root of following system of equations:

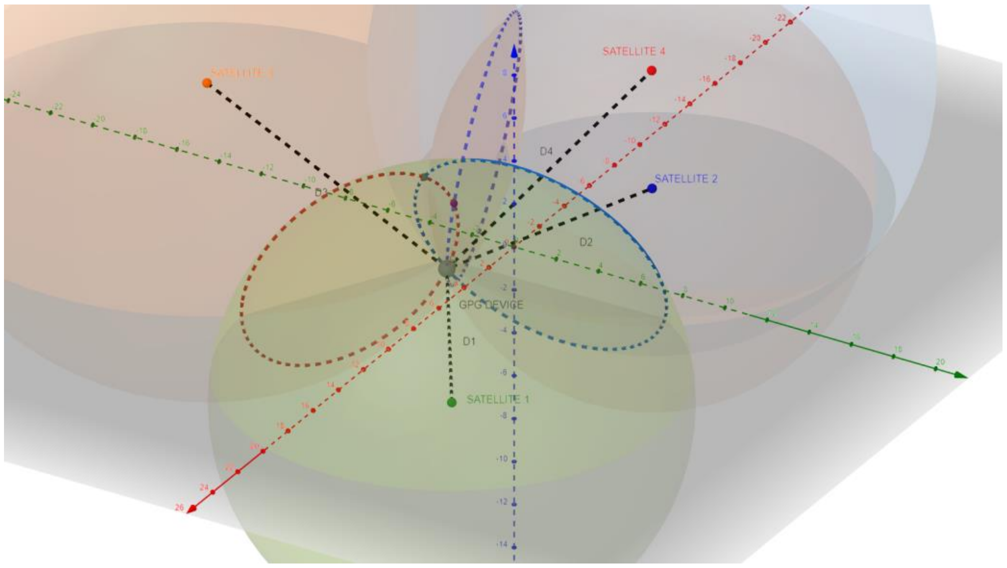


Figure 12 Point determination method from 4 known points in 3 dimension

$$D1 = \sqrt{(x - \boldsymbol{x_1})^2 + (y - \boldsymbol{y_1})^2 + (z - \boldsymbol{z_1}}$$

$$D2 = \sqrt{(x - \boldsymbol{x_2})^2 + (y - \boldsymbol{y_2})^2 + (z - z_2)^2}$$

$$D3 = \sqrt{(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2}$$

$$D4 = \sqrt{(x - x_4)^2 + (y - y_4)^2 + (z - z_4)^2}$$

$Dx: \ Distance\ from\ a\ satellite\ to\ GPS\ device\ (m)$

$x_i, y_i, z_i: Satellite\ coordinate\ (m)$

$x, y, z: GPS\ device\ coordinate\ (m)$

GPS technology is enabled on almost every smartphone nowadays, which is carried everyday by traffic participants. Along with an enormous geography database provided by Google Service, this combination is included in building the GPS tracking and payment subsystem to track the trajectory of the user in real-time in each specific road path. After the user leaves the road, the collected coordinates can be calculated to output a final road usage of the user on that road. The road usage calculation can be realized by following equation:

$$D = \sum_{i=0}^{k} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

$D : Total\ road\ usage\ (m)$

$k : Number\ of\ coordinate\ samples$

x_i, y_i: $Latitude\ and\ longitude\ sample\ at\ time\ instant\ i$(m)

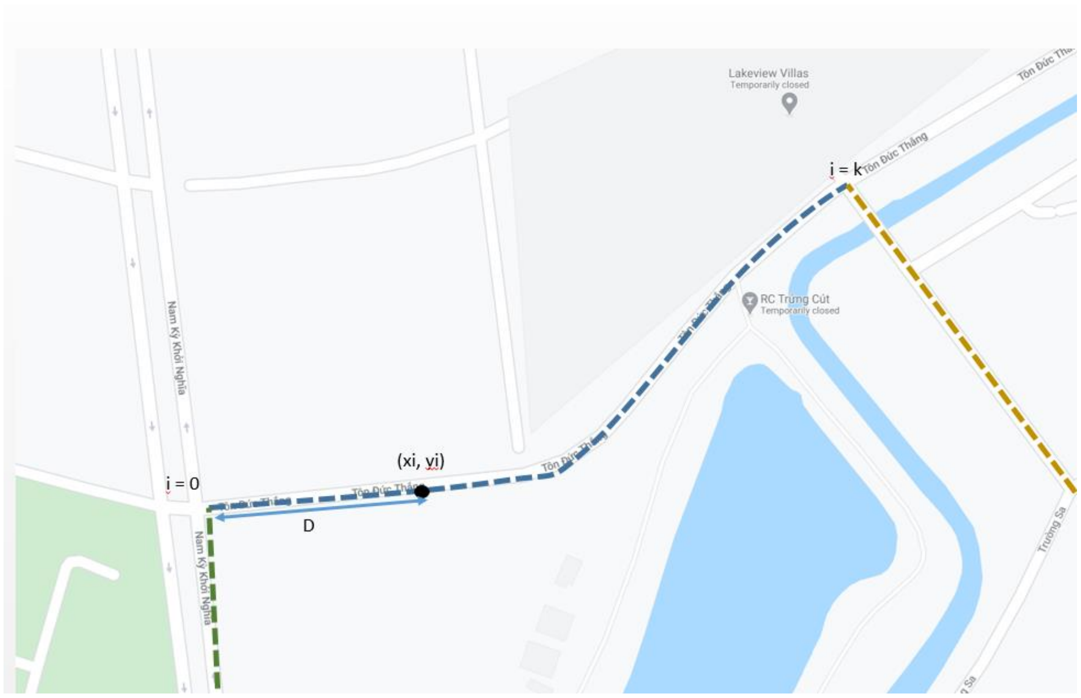


Figure 13 Travel distance calculation strategy

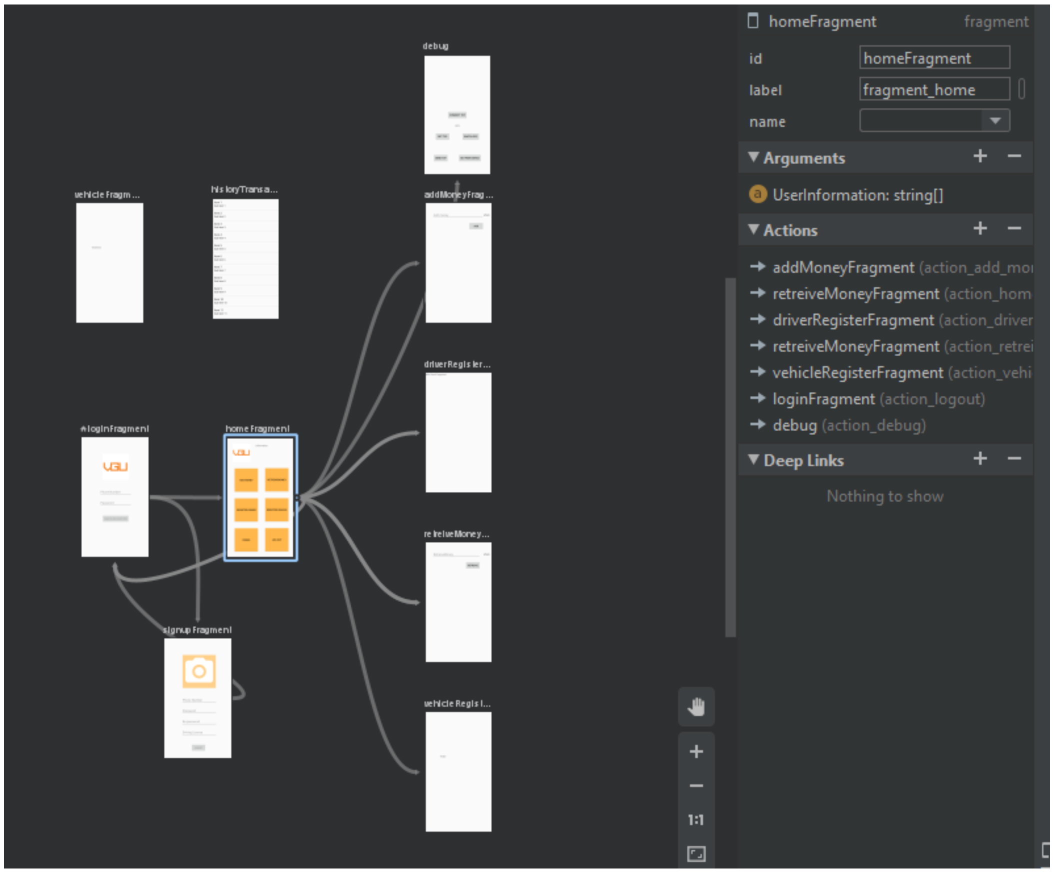
4 System implementation

4.1 E-wallet Android application

4.1.1 Activities and fragments management

An Activity is the main visualize interface that the user interact with most of the time on an android application [9]. Activity class handles entering, pausing, exiting actions for the application while on an activity. When switching from an Activity to next Activity, the system need to save current state of previous activity, destroy current view and generate new view for next Activity [10]. This action consumes unnecessarily energy and memory resources of the mobile device, when some of the UI (User interface) can be re-use as a general theme for entire life cycle of the application. Android library provides an alternative option called Fragment to inherit usable portion of the UI while updating required information. The API makes this possible by introduce modularity and reusability into the hosted Activity [11] via a fragment tag embedded in a XML file, which defines the user interface layout structure of the Activity [12]. The fragment tag can be constrained its position with other elements, which are encapsulated functionalities, respectively in this case.

The usage of Fragments are more reasonable since Android Studio 3.3 arrived [13]. In this version, Google introduced Navigation feature in Figure 14, which is a tool set allows developer to visualize and manage the navigating relationship between Fragments such as path, direction, action, etc., to ensure a consistence and predictable behavior when executing a transition [14]. This tool comes with a graph that illustrates transitions with actions and arguments properties, which are mandatory for the E-wallet application graphical interface management back-bone.



homeFragment fragment

id homeFragment

label fragment_home

name

Arguments

UserInformation: string[]

Actions

addMoneyFragment (action_add_mo

retreiveMoneyFragment (action_hom

driverRegisterFragment (action_driver

retreiveMoneyFragment (action_retrei

vehicleRegisterFragment (action_vehi

loginFragment (action_logout)

debug (action_debug)

Deep Links

Nothing to show

Figure 14: E-Wallet Navigation UI

While only E-Wallet app have one Main activity consisting a navigation bar existing throughout application processes, all other different user procedures shall be contained in Fragments system. This architecture design takes advantage of today mobile device hardware power and decrease potential errors in development process when managing multiple user interfaces and actions between them.

4.1.2 Main Activity

The Main Activity initialize when the app is first start and kept alive until user close the app. On creation of the Activity, Navigation bar shall be introduce to the environment with three main paths to corresponding Fragments: Home, History and Vehicle. Right after that, Login Fragment shall be generated and hide the navigation bar away, because user have to login and connect to central server first before access further into the app. Otherwise, the app would crashed and raised Null pointer exception, since all the data need to be used is initially null before retrieved from central server when logging in success.

4.1.3 Login Fragment

The Login fragment in Figure 15 is an interface allow user to input their phone number and password to access further features of the E-Wallet.

Login is the first Fragment user encounter when Main Activity is created, and this is possible when assigning start destination properties to it [15]. Phone number and password text placeholder is the output of EditText objects creation, these objects then linked to corresponding tag in XML file of Login Layout via findViewById procedure [16], this method applied for all linkage between user interact Java objects and visual interface XML tag.

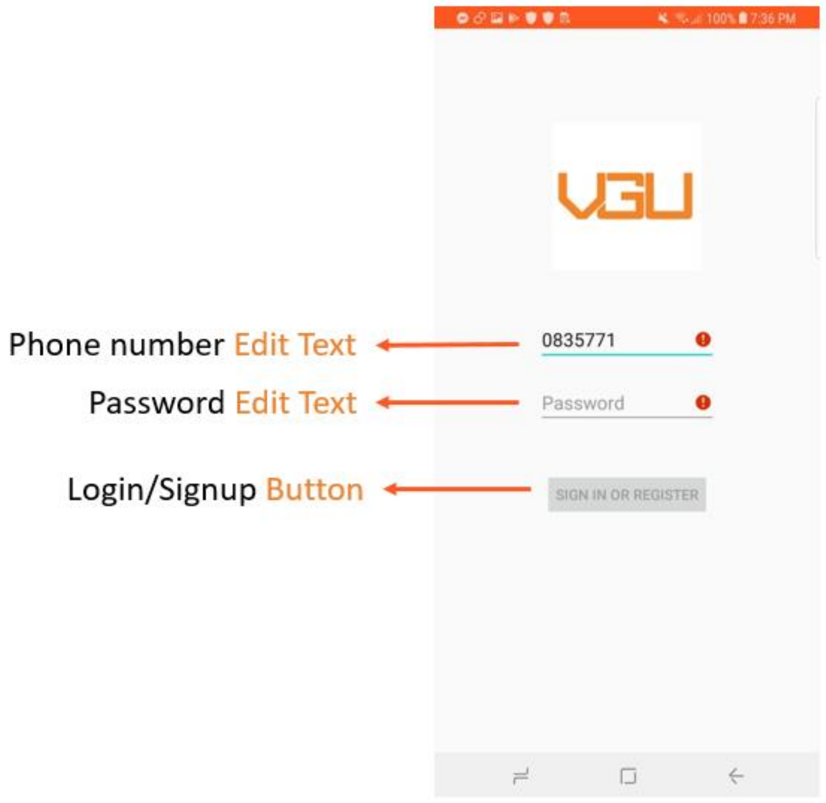


Figure 15: E-Wallet Login Fragment UI

Phone number text field will have a live check if the input is a number and have 10 or 11 digits (standard mobile phone number in Vietnam [17]). Password field will check if input is a string containing alphabet character, number and special character. If these conditions are fulfilled, the login Button object will active to be pressed.

On click handler of login Button [18] will send login information to central server to authorize the user availability. If the authorization success, the application will received a positive feedback along with important user information such as credit card information, E wallet credit and vehicle name to prepare a Bundle [19] to send to Home Fragment.

4.1.4 Home Fragment

The Home Fragment in Figure 16 is an interface allow user to do multiple activities related to:

Budget management

Vehicle/Driver registration

Log out

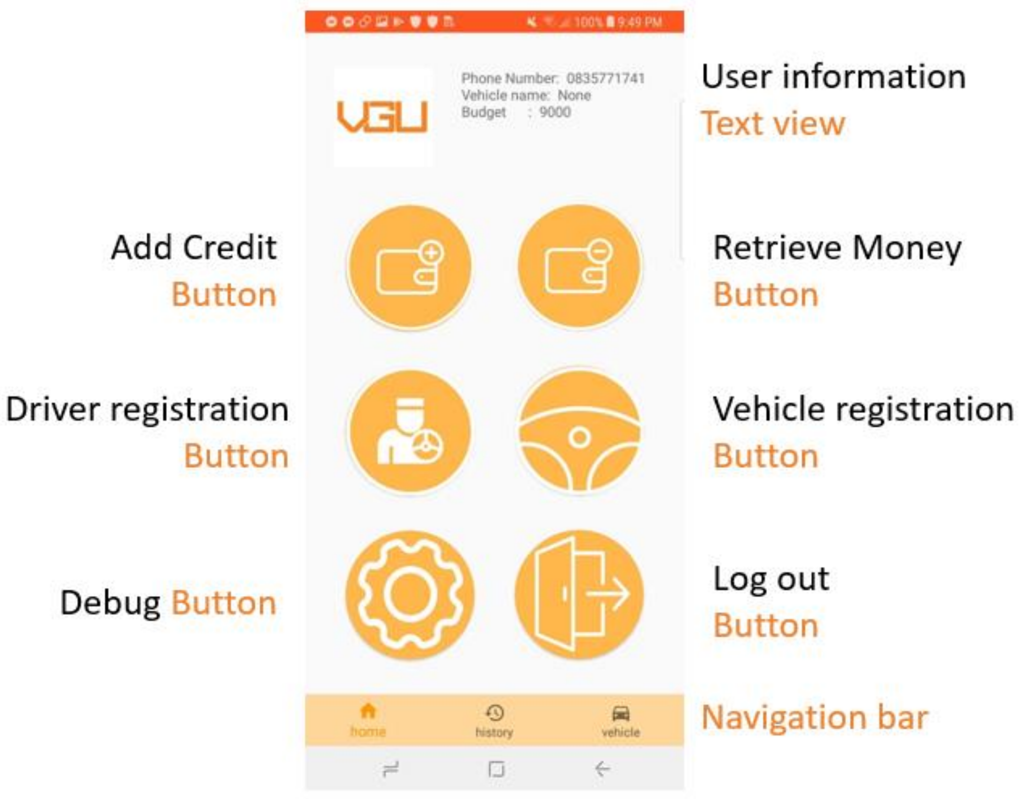
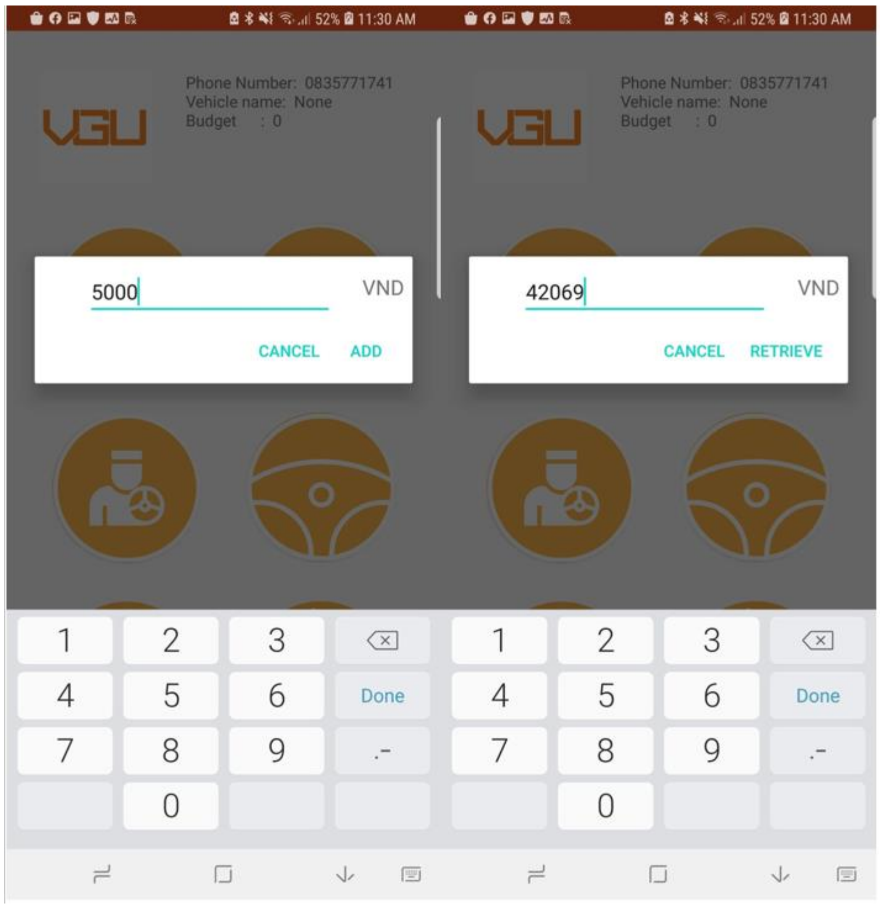


Figure 16: E-Wallet Home Fragment

4.1.5 Add Credit and Retrieve Money

These actions in Figure 17 enable user to change their wallet credit status by adding and retrieving money to linked credit card. After this action finished, the budget data on central server side also update accordingly. Home fragment shall run a loop thread in background to continuously fetching user information every 100ms to update budget value on User Information Text View section.

A Text Watcher [20] shall observed user input when Retrieve Money action is ongoing. This object will enable the Retrieve Button, which allows user to further proceed the action, if the input amount is less than the current credit in the budget. Otherwise, the user have to return to Home Fragment or change the input.


Phone Number: 0835771741
Vehicle name: None
Budget : 0

5000 VND
CANCEL ADD

Phone Number: 0835771741
Vehicle name: None
Budget : 0

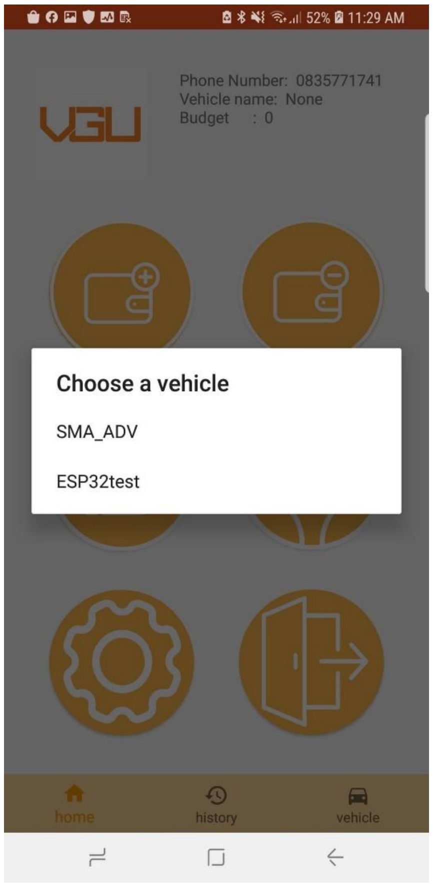
42069 VND
CANCEL RETRIEVE

Figure 17: Add/Retrieve Money dialog

4.1.6 Driver Registration

This action in Figure 18 creates an interface to start exchange information with the VDRC via Bluetooth. At first, the application shall check for Bluetooth availability on the mobile device and ask for access permission if previously not granted yet. If the user does not grant the accessibility permission, the application will return to Home Fragment and wait for other actions. If permission granted by the user, the mobile device shall start searching for a registered VDRC and start exchanging information with it. Further background procedure will be described in *Bluetooth interface handler* section.

After choosing a suitable vehicle, the mobile device shall send a request to the central server to update vehicle data of regard user profile. The vehicle data contains vehicle name and vehicle mass.


Phone Number: 0835771741
Vehicle name: None
Budget : 0

Choose a vehicle

SMA_ADV

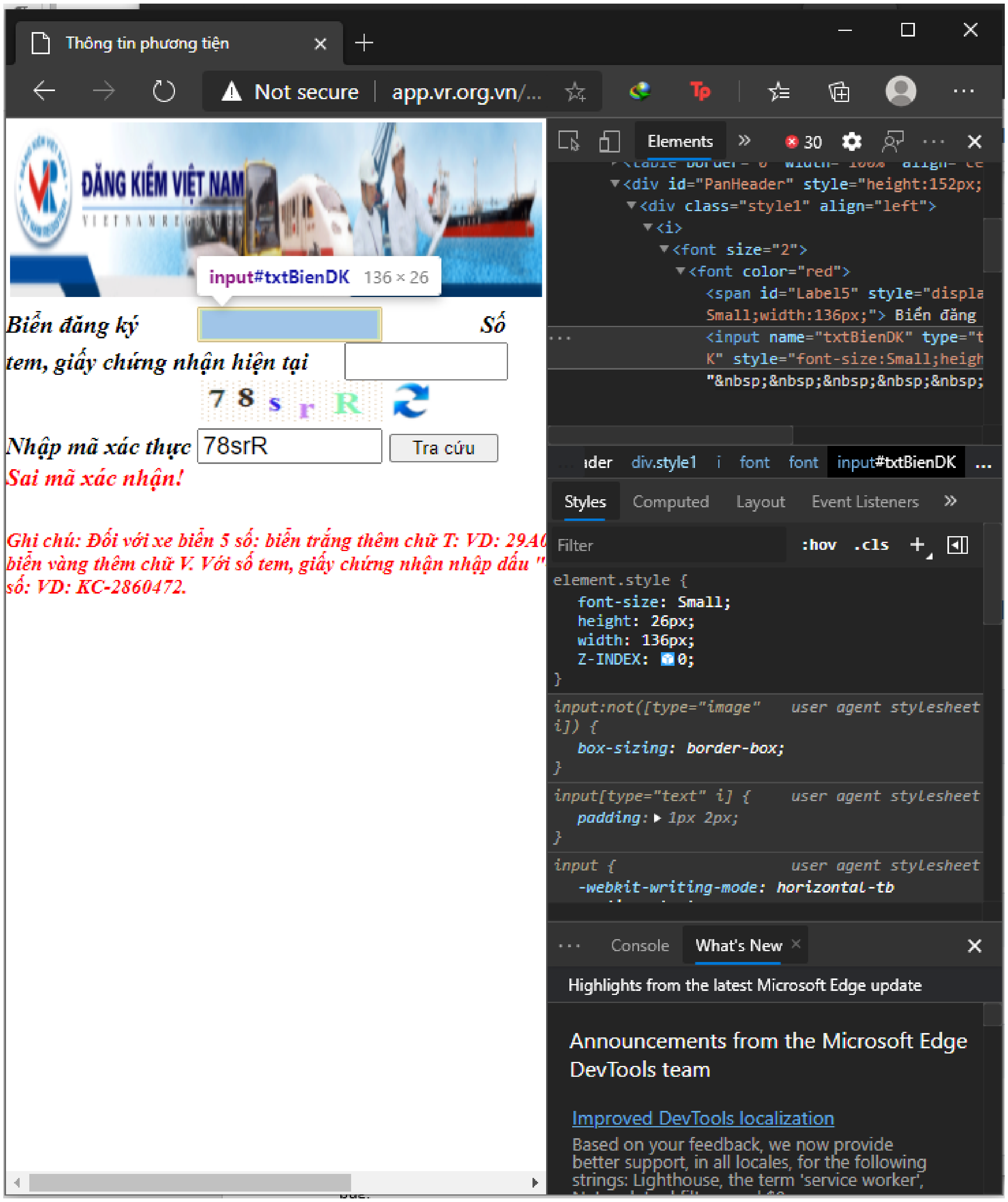
ESP32test

Figure 18: Driver registration dialog

4.1.7 Vehicle Registration

Vehicle registration in Figure 20 will have a similar Bluetooth handler with Driver registration interface including searching for a VDRC, but the main difference is the list of Bluetooth device displaying virgin VDRC. The registration process shall begin after the application the License Plate and Registration Number Edit Text are filled in. The data parsing from the Vehicle Registration Agent website shall be collected after the register button is clicked. Using real vehicle information inputs from the user the E-Wallet can get detailed information from the website via HTML parser and update in Vehicle Information tab.

The HTML parsing task requires a proper inspection of that HTML file. For the Vehicle Registration Certification Agent website, we can use the inspection tool of google chrome in Figure 19 to locate the required HTML tag. In this case, all required the input field is in input tag, and comes with a name: "txtBienDK" is for License Plate, "TxtSoTem" is for Registration Number and "txtCaptcha" is for input captcha. The after filled in all the necessarily information in the application, these information will be forward to the website input placeholder. Then the application will simulate the submit button to enter the vehicle description page. The data obtaining will have the same mechanism and pull to the mobile device local storage. For the captcha, the application will first download the image from the HTML image tag, and run it through an image processor to get the result, since the captcha on this webpage is not hardly distorted, with a few attempts the application will have a good guess. However, the image should be pre-processed first by filter all the dot in the background to reduce the failure attempts.


Thông tin phương tiện

Not secure | app.vr.org.vn/...

Biển đăng ký | Số tem, giấy chứng nhận hiện tại

7 8 s r R

Nhập mã xác thực 78srR | Tra cứu

Sai mã xác nhận!

Ghi chú: Đối với xe biển 5 số: biển trắng thêm chữ T: VD: 29A0 biển vàng thêm chữ V. Với số tem, giấy chứng nhận nhập dấu " số: VD: KC-2860472.

```
▼<div id="PanHeader" style="height:152px;
  ▼<div class="style1" align="left">
    ▼<i>
      ▼<font size="2">
        ▼<font color="red">
            <span id="Label5" style="displa
            Small;width:136px;"> Biển đăng
...         <input name="txtBienDK" type="t
            K" style="font-size:Small;heigh
            "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
```

```
element.style {
   font-size: Small;
   height: 26px;
   width: 136px;
   Z-INDEX: 0;
}
input:not([type="image"    user agent stylesheet
i]) {
   box-sizing: border-box;
}
input[type="text" i] {     user agent stylesheet
   padding: ▶ 1px 2px;
}
input {                    user agent stylesheet
   -webkit-writing-mode: horizontal-tb
```

Highlights from the latest Microsoft Edge update

Announcements from the Microsoft Edge DevTools team

Improved DevTools localization

Based on your feedback, we now provide better support, in all locales, for the following strings: Lighthouse, the term 'service worker',

input#txtBienDK 136 × 26
Elements
30
ıder div.style1 i font font input#txtBienDK
Styles Computed Layout Event Listeners
Filter :hov .cls
Console What's New

Figure 19: Website inspection interface

After all vehicle data is stored locally on a temporally HashMap [21], the information will be packed and sent to 2 destinations: VDRC (via Bluetooth service) and central server (via TCP request). To validate this action, the vehicle name will not appear on the virgin VDRC list in this interface, since it is registered.

When using this feature the connection between E-Wallet and VDRC must be kept alive before doing registration procedure. Therefore, after the user choose a virgin VDRC, the application will also remember the MAC address and name of the device and re-connect afterward preventing unwanted bug.

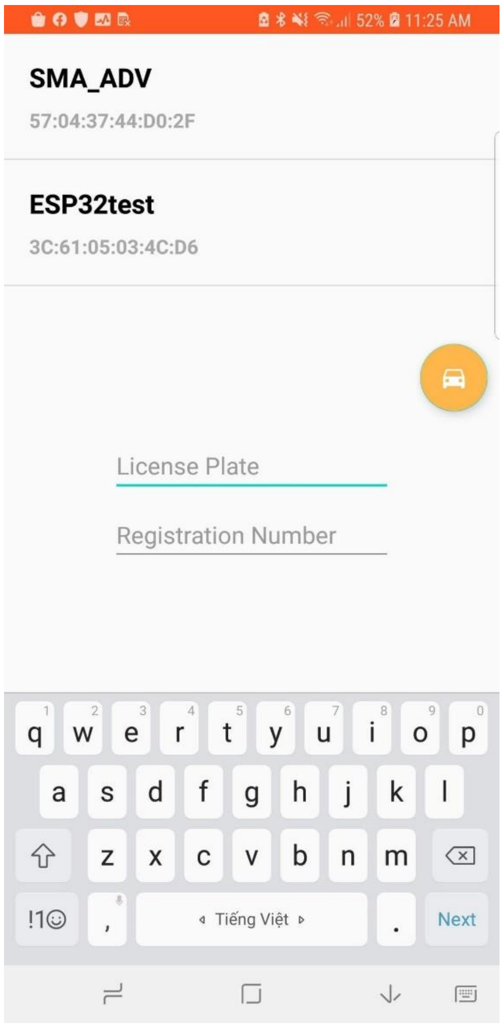

SMA_ADV
57:04:37:44:D0:2F
ESP32test
3C:61:05:03:4C:D6

Figure 20: Vehicle registration fragment

4.1.8 Log out

This action triggers the central server by sending a log out request along with user phone number. After receiving a positive response, application shall transition to Login Fragments, and all previous user data stored on the mobile device shall be set to null.

4.1.9 Client socket interface handler

The mobile device will a client in the TCP socket communication with respect to the central server. The client socket class shall have 3 main procedures: start connection, sending request and listening to response. In order to maintain consistence connection to the central server, these procedures shall run in parallel threads. Start connection thread shall run at the beginning of every Fragment. If the mobile device cannot initiate a connection with central server by throwing an I/O Exception [22] during start connection thread execution, the application will set a 1000ms timer before restart the connection.

Application shall start listening thread right after the connection is established. This thread will run in an infinite while loop only break when an IO Exception is raised. While looping, the application will continuously fetching the data from TCP socket buffer and covert to a string. When this procedure encounters an end of line indication, the application will forward all collected data to further processes.

If the application is in login session, it means the long term key between the mobile device and the central server not existed or erased (more about long term key in *Security data exchange*). Then, when listening thread captures a message from central server, it will redirect this data to JSON analyzer, which is a function derived un-encrypted JSON structured message to extract data field inside the message. If the long term key were stored in mobile device, the listening thread will forward incoming message to the Decryption service, then forward to JSON analyzer.

Before sending any request to central server, the message must be encrypted by a temporally and long term key and sent using sending thread.

When sending or listening, the application shall check for connection status because the read and write I/O stream is initiate at the start connection thread, otherwise, the application will throw I/O or null pointer exception. The reason start connection, sending and listening action must be handled in thread is because the application can still captured new data while sending a request or update secrete key on the go.

4.1.10 GPS tracking service and Maps interface

GPS service is realized by the GPS chip implemented in the mobile device and the data from Google to translate the plain coordinate value to location name.

The GPS service will start when the driver registration complete. At first, the application will check if the device is full filled the requirement of GPS service, which are the availability of GPS interface and the permission to access this service from the user. If all these condition is passed, then the GPS initiation process shall allow the Location callback [23] to be activated. The location callback act as an event handler to process whenever the client location provider object [24] obtained new GPS location for the system. The cycle of GPS fetching service will have 2 time intervals: default interval and fastest interval, defined when configured GPS service. The default interval is proportion to the power consumption of the mobile device, because the narrower the time window, the larger the number of request GPS chip has to handle. For this project, the default interval is choose to be 500ms to balance between the accuracy and application speed. The energy consumption does not need be account on a prototype application, since there is only one device to test on, unlike the marketing E-Wallet, where it has to suit hundred or thousand devices with multiple specification deviation. Although the normal cycle speed is already configured as such, the result is not always as it seems according to the documentation [25]. Therefore, the application need to have fast interval to configure the upper limit in case things get out of hand.

The paying fee request has to inform the central server the street name, distance and vehicle mass with phone number to calculate the fee for the following user account. However, the travel distance will be calculated on server side to maintain the consistency between mobile devices, so the longitude and latitude of the first coordinate when entering a new street or after successfully registration a driver will be in the request message as the replacement of distance. Moreover, these coordinate data will later be pulled from database for history record visualization, so storing these values is also mandatory at central server side (detailed procedure on central server side described on chapter 4.2.2.6). The last coordinate for the payment request will be the last coordinate before the GPS service detected a change in street with the new received GPS. After sending pay fee request, the application will re-send if the response is bad or lost connection or response timeout, which are all considered as negative responses.

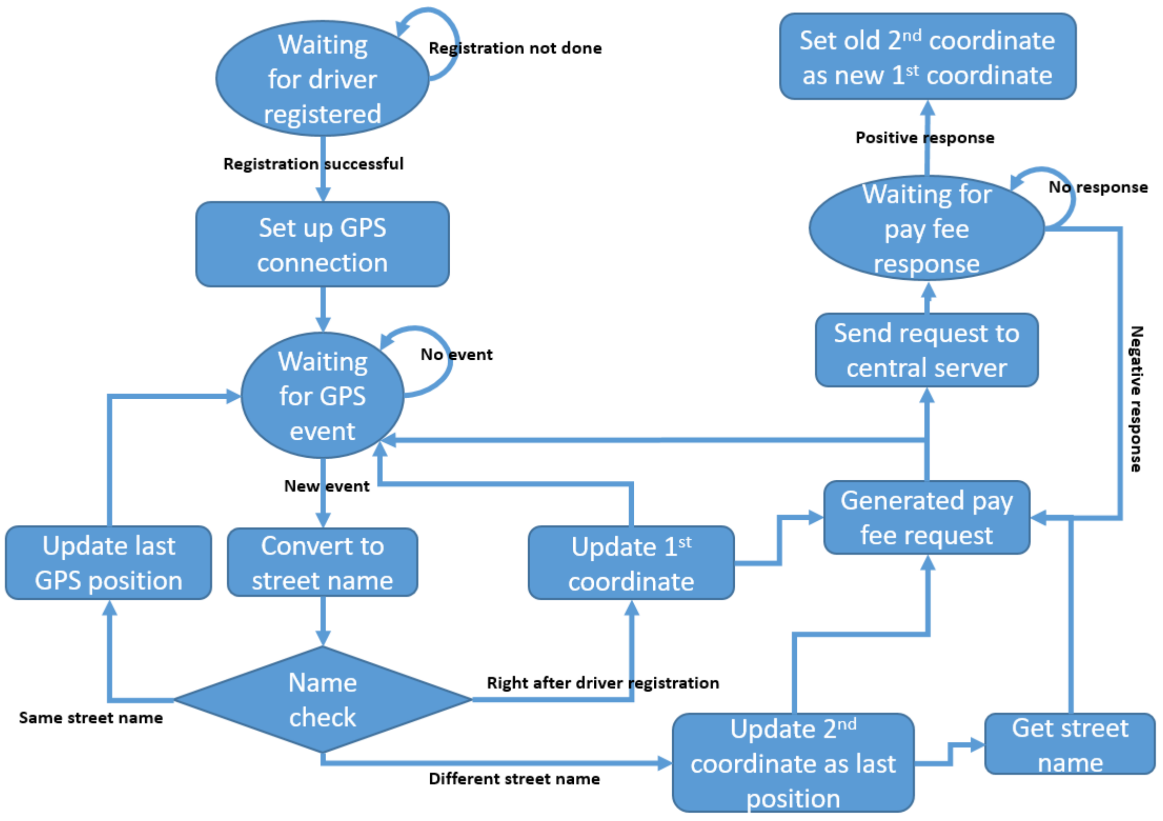


Figure 21: Paying fee process diagram of E-Wallet application

If the response is successfully sent, the coordinate entering new street will be considered as the last updated coordinate just before the pay fee request being constructed. This design will keep the consistence and continuous aspect of the GPS tracking task.

Storing and communicating based on GPS coordinate alone is still sufficient, but for managing and visualizing purpose, the coordinate value have to translate to real street name. So that, the configuration for unit road fee on server side will be more manageable by human. By using Geocoder API [26], the mobile can access the biggest location databased in the world provided Google, and it can obtain street name by calling get address line [27] from Address library.

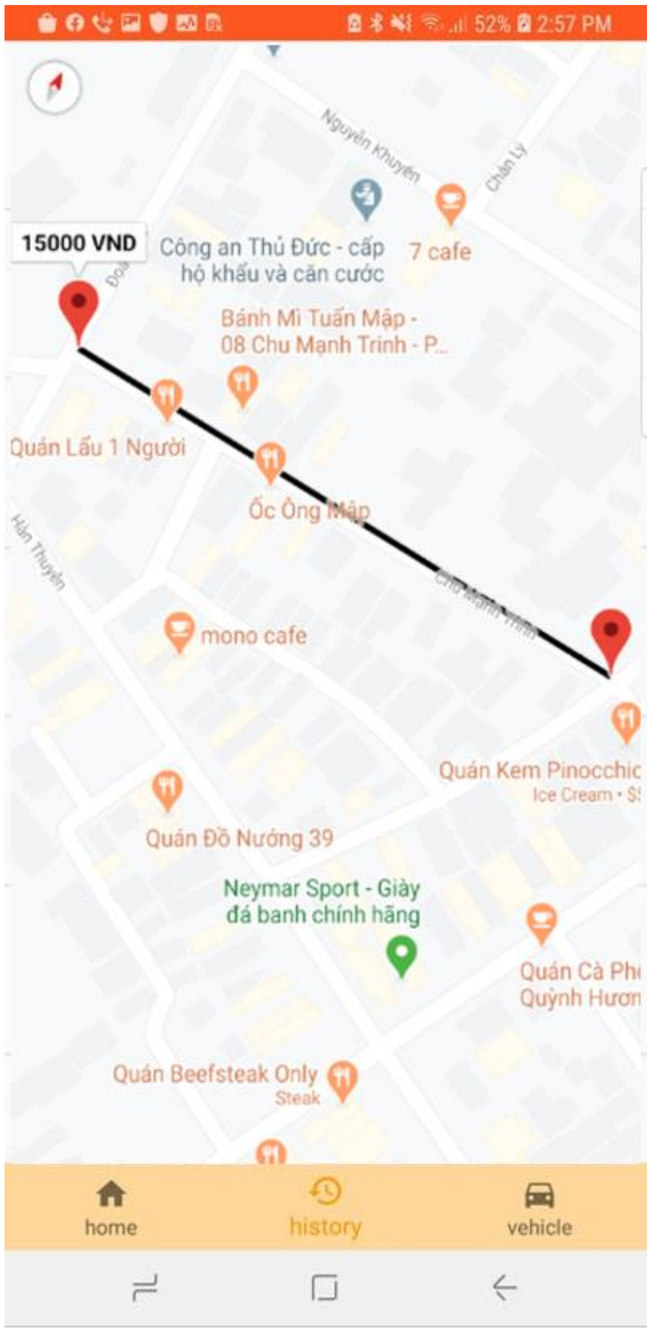


Figure 22: Maps activity

The Maps activity in Figure 22 is mainly inherit GoogleMap [28] and SupporMapFragment [29] classes.

This activity usage to visualize history road fee transaction to help user track their balance integrity.

The activity provides the start and end marker [30] indicating the trip for selected paying record.

4.1.11 Bluetooth service

Bluetooth service on mobile device shall start when application in Vehicle or Driver Registration Fragment. Since Bluetooth is treated as an external device with respect to the CPU on mobile device, or in Android library terminology, a Bluetooth Adapter, the application need to verify the Bluetooth chip whether this mobile device is Bluetooth supported or not [31]. If the hardware requirement is fulfilled, the application shall initiate an Intent [32] and passing the previously introduced Bluetooth adapter to listen to user permission to turn on Bluetooth connectivity in Figure 23. An Intent act like an event handler optimized to run on Android OS instead of un-controlled thread and loop, and this class also provide a Broadcast Receiver [33] in order to captured dynamically Bluetooth messages broadcasted by OS system in publish-subscribe pattern [34]. This handling scheme will be applied on later states of Bluetooth services.

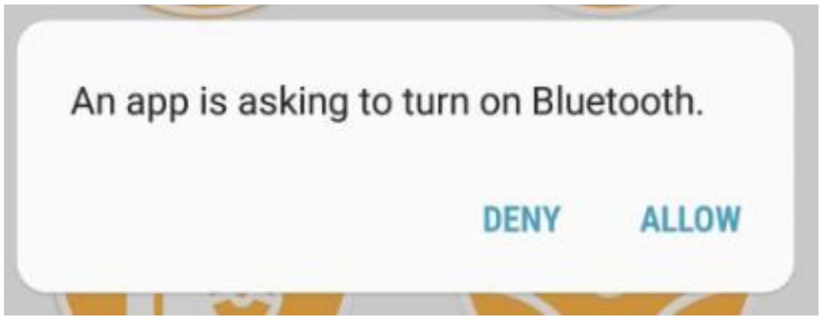

An app is asking to turn on Bluetooth.

Figure 23: The enabling Bluetooth dialog

The mobile device is an Initiator device [35] in this project Bluetooth communication, therefore, the application shall start discovery to retrieve VDRC MAC address. After the VDRC selected by the user, the application shall attempt to initiate the Bluetooth connection as client [36]. To proceed connection initiation, the application create a RF Communication Socket and passes in its UUID to start exchanging with VDRC on a new thread. If there is no I/O Exception occurred, application will cancel discovery mode to prevent slowing down the connection. If the attempt is succeed, the application will finish the thread that running attempt connection and open a Bluetooth socket input and output stream declared. The listening and sending messages handler have a similar design scheme of TCP connection of the application described in chapter *Client socket interface handler.*

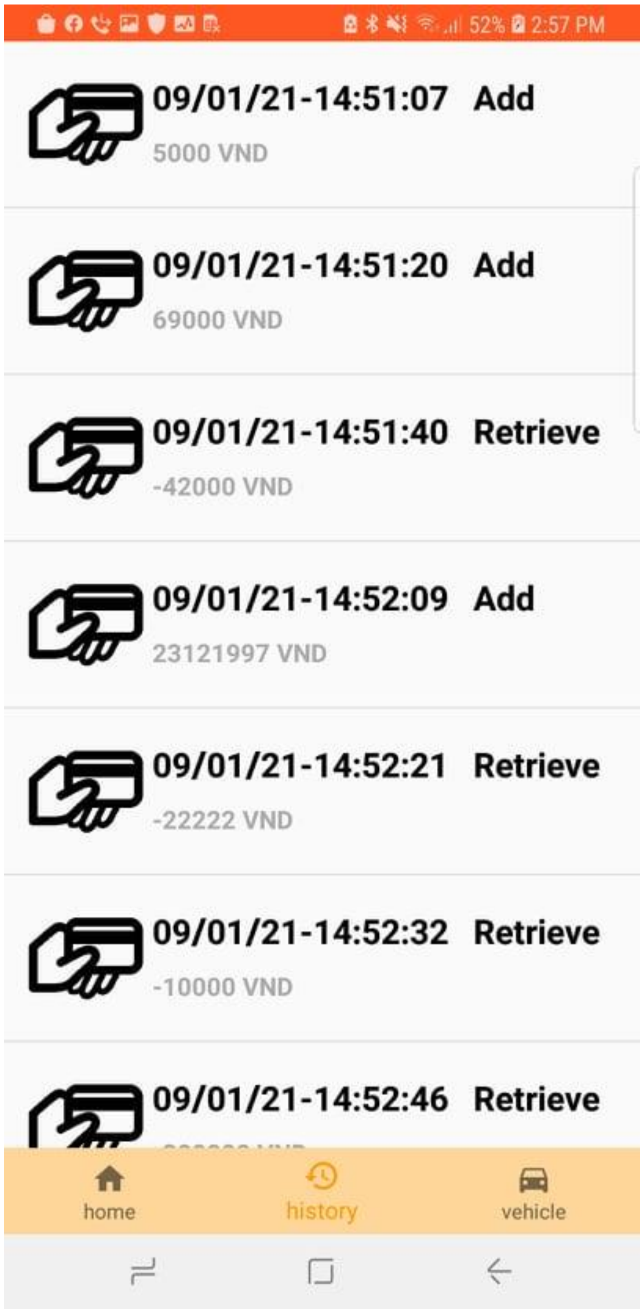
Due to the nature of Bluetooth communication, which is low range and peer to peer, it is not necessarily to develop a secured communication with special authorization process. The data exchange via Bluetooth service is the vehicle information, and this interchange only occur once when register the driver and once when register the vehicle.

4.1.12 Client Encryption and Decryption service

Encryption and Decryption services on mobile device side is handled by Fernet library for Java [37]. In order to encrypt or decrypt any information, the API required a key and a token. The key in this case shall be the temporally key or the long term key depending on the stage of the application. When encrypting, the token is generated using the key and the message package needed to be sent. When decrypting, the token is the decrypted message that the listening thread passing to. Moreover on decryption service, the mobile device need to set the alive time of the token to match will the central server side, in this case, it is 60 second by default. Otherwise, even with a valid key, the TCP communication still not be possible.

4.1.13 History record

This this interface in Figure 24 is activated when user navigate to history tab. The purpose of the fragment is to visualize the transaction history including road fee payment, add and retrieve money action. This feature let the user manage and reviewing their balance.



09/01/21-14:51:07 Add
5000 VND

09/01/21-14:51:20 Add
69000 VND

09/01/21-14:51:40 Retrieve
-42000 VND

09/01/21-14:52:09 Add
23121997 VND

09/01/21-14:52:21 Retrieve
-22222 VND

09/01/21-14:52:32 Retrieve
-10000 VND

09/01/21-14:52:46 Retrieve

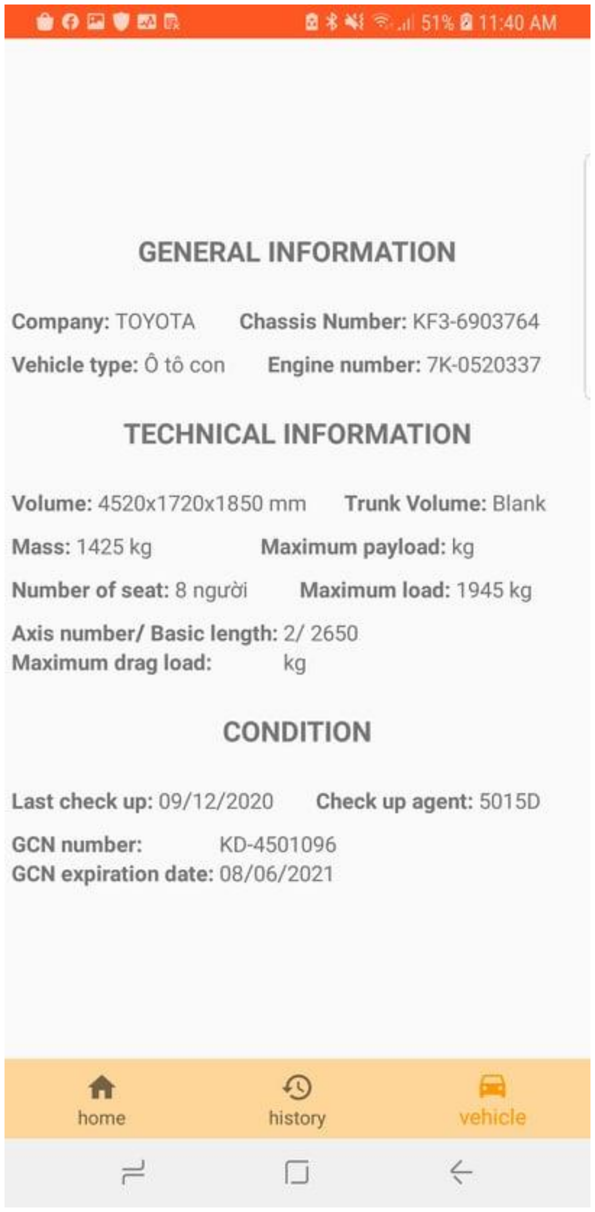
Figure 24: History records fragment

Since all fragment share a same public socket handler initiate at the beginning of parent activity, when History Fragment view is created, it does not need to initiate the TCP connection. Instead, it will generate a blank list view and start filling in later on. To start populate the list view, the mobile device will send a request containing the index of the record that it want to obtain from database, which is 0. This number will increase until to a maximum limit, and this limit value will be the result of screen height and the list view item height in order to fully fill the interface at the beginning.

It is not necessarily to fill up the blank canvas if the number of history record in the database is smaller than the limit value. The fragment also implemented an on scroll listener [38] for the list view object to trigger further request if the user want to investigate deeper in the history. The design prevented from loading unnecessarily resource at the start of this Fragment if the vision limited to only few records at a time.

4.1.14 Vehicle information activity

After register driver or VDRC, there is a tab on navigation bar in Figure 25 to check the information of the vehicle stored locally in the mobile device. This interface can be considered a part of the VDRC, since it can realized the digital part of Vehicle Digital Registration Certificate for visualizing purposes. This tab information will be blank if no vehicle is connected to E-Wallet.



GENERAL INFORMATION

Company: TOYOTA **Chassis Number:** KF3-6903764

Vehicle type: Ô tô con **Engine number:** 7K-0520337

TECHNICAL INFORMATION

Volume: 4520x1720x1850 mm **Trunk Volume:** Blank

Mass: 1425 kg **Maximum payload:** kg

Number of seat: 8 người **Maximum load:** 1945 kg

Axis number/ Basic length: 2/ 2650
Maximum drag load: kg

CONDITION

Last check up: 09/12/2020 **Check up agent:** 5015D

GCN number: KD-4501096
GCN expiration date: 08/06/2021

Figure 25: Vehicle information fragment

4.2 Socket and Sqlite based central server

4.2.1 Server socket interface handler/ TCP connection handler (TCH)

4.2.1.1 Initiate a connection

Large portion of time interval of a connection is to wait for the request from client device. Therefore, the central server is required to set lower the priority of the waiting request procedure than waiting new connection procedure. The forward approach is to set this pending action to a thread and let the main process keep on accepting new connection. However, for this project, the central server uses low level I/O multiplexing trigger via selectors API [39] for a Python server.

4.2.1.2 Read handler

Read handler is the first handler that the TCH execute after a request is received. It will check for the availability for data, decrypting it use previously shared key and decoded it with UTF-8 format. The decoder is provided by Fernet python along with key generator and token generator. All decrypted data shall be processed by JSON reader to take apart and check the consistency of each part in the message. After filtering the action need to be handle, read handler shall send these action to database handler to generate response. At the end of read handler, it will set the state of the message handler to write handler, in order to send response.

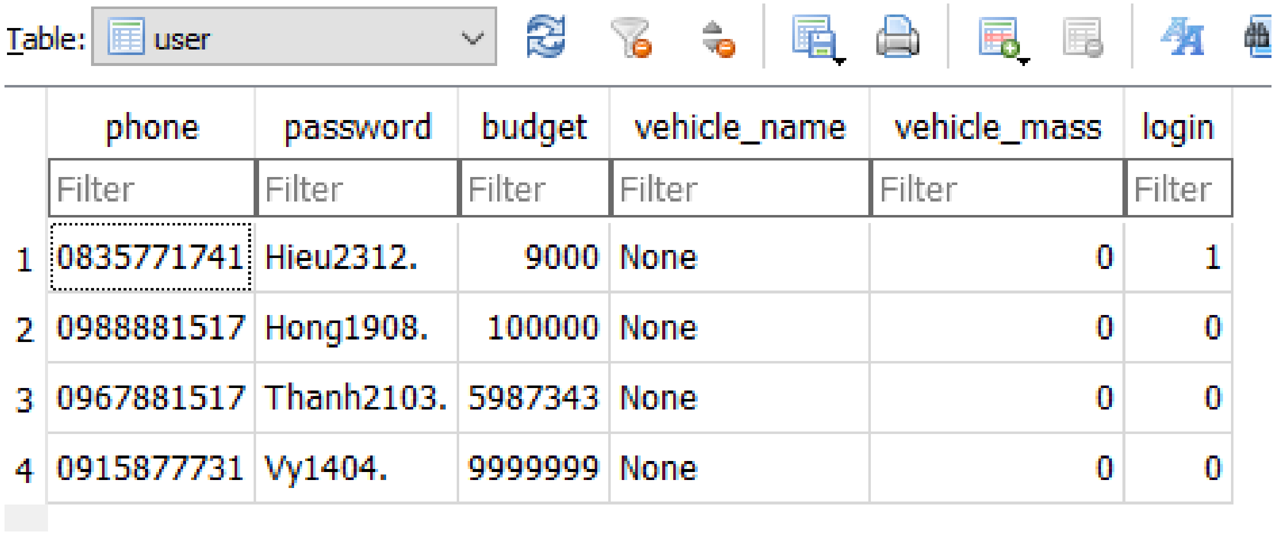
4.2.1.3 Write handler

Write handler usually start after read handler, but in some cases, such as share temporally key, this handler will be initiate first. The purpose of the write handler is encrypting the response message with appropriate key and close the TCP connection afterward.

4.2.2 Sqlite database handler (DBH)

4.2.2.1 Login

SIH will provide to DBH phone number and password from login request. The DBH then check if the phone number exist in the User table or not. If the result is positive, DBH will verify the password comes with that phone number in password column. If the verification is passed, the login status of this phone number shall be set to true, and other related information on the same row will be extract to send a response. At the same time, stored the temporally key into the database for it to become long term key (detailed description will be on Chapter 5.6). Finally set the Result of the response to "good". If phone number does not exist in the database or wrong password, the Result will be set to "bad" and no further action from DBH.



	phone	password	budget	vehicle_name	vehicle_mass	login
	Filter	Filter	Filter	Filter	Filter	Filter
1	0835771741	Hieu2312.	9000	None	0	1
2	0988881517	Hong1908.	100000	None	0	0
3	0967881517	Thanh2103.	5987343	None	0	0
4	0915877731	Vy1404.	9999999	None	0	0

Figure 26: user table in database

4.2.2.2 Log out

Logging out required only phone number to execute. DBH shall set the login status false, vehicle mass to 0 and vehicle name to “None”. This way, the account status will be reset to initial state before login, and the Result will be “good”. However, this action can only be executed if the given phone number is found in the database, otherwise, no action will be done along with “bad” Result.

4.2.2.3 Sign up

Above actions is executed through SELECT data or UPDATE data from database. Signing up an account on other hand, DBH will INSERT INTO the User table new row representing new E-Wallet account. As long as the sign up phone number did not already exist in the database, the Result will be “good”, otherwise, “bad”.

4.2.2.4 Add money

This action required phone number and the amount of money user need to input to E-Wallet. On the User table, the budget column will be increase based on the current value and the additional added regarded the phone number.

Any action related to budget management need to be recorded in History table. Recording procedure must always have a time mark to traceability purpose if the user complain about false information about their budget displayed on the E-Wallet. The time information will captured at the moment DBH start process budget related actions. For adding money, the money amount and existed phone number will be update in History table along with time value, other section will be left “None” or zero.

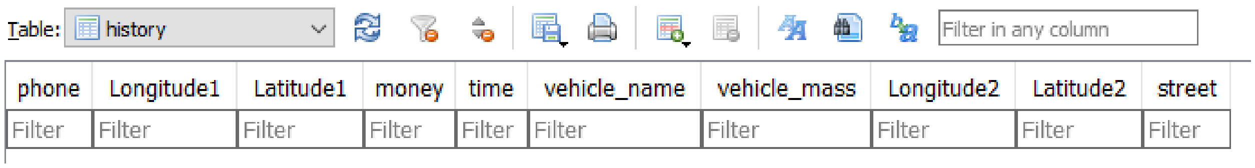


Table: history

Filter in any column

phone	Longitude1	Latitude1	money	time	vehicle_name	vehicle_mass	Longitude2	Latitude2	street
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter

Figure 27: history table in database

4.2.2.5 Retrieve money

Retrieving money request is processed exact the same with adding money but subtracting. However, DBH will give "bad" result if the amount of money user want to retrieve larger than the amount of money currently available. This checking procedure should be done on mobile device side, but for the integrity factor and prevent crashing, this should also be done on central server side independently.

4.2.2.6 Pay fee

Paying fee inherits the verification steps and DBH actions of retrieving money since it has a similar nature from central server perspective. Moreover, this action required start, end GPS coordinate and street name to record to History table. The amount of money need to be subtracted is a multiplication of vehicle mass, street based fee and distant travel:

$$Road\ fee = Vehicle\ mass * distance\ travle * unit\ road\ fee$$

The distance is calculated based on given GPS coordinate with the formula derive in Chapter 3.7. The street based fee or road fee per kilometer data is stored in Street table, and the based value can be search by DBH using given street name in the request. DBH retrieves vehicle mass from User table with respect to the given phone number.

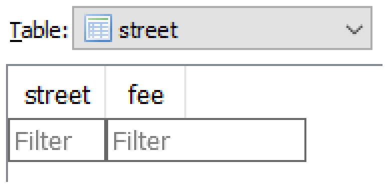


Table: street

street	fee
Filter	Filter

Figure 28: street fee unit table in database

If the user have not registered as a driver, the vehicle mass will be 0, which is invalid, so the response will have “bad” Result and no further action from DBH shall be done to prevent unwanted budget modification to E-Wallet.

4.2.2.7 Register driver

For this request, the central server will need phone number, vehicle mass and vehicle name along with the action to update vehicle status in User table with given phone number.

4.2.2.8 Get history transaction

This function shall handle when a mobile device want to fetch recorded information related to transaction including paying fee, add and retrieve money. This process required phone number to sort out all the transaction record from the database. However, the response generated from DBH only contains 1 item in the list of records, therefore, in the request, the mobile device need to provide an index number in the request for the DBH to choose the right item to pack in the response. If the index number is exceed the amount of items that DBH can find, the response Result shall be “invalid”.

4.2.2.9 Get user information

The response to this action shall contain all characteristic of the user profile in User table regard the phone number provided in the request.

4.2.3 Server Encryption and Decryption service

Due to the privacy and sensitivity of data stored in database of central server, an encrypted communication is highly demanded. Therefore, all exchanged input and output messages must be encrypted with a secured cryptography machine from the current time (detailed description will be on Chapter 5.6).

4.3 VDRC profile and notify application

4.3.1 Virgin registration service

A virgin VDRC is a blank board without any vehicle information stored. The registering process should be done when mount the VDRC on a vehicle and done via a mobile device. During the IDLE mode of VDRC, the device will check the EEP ROM for any data of the vehicle information and the virgin flag. After device boot up, the EEP ROM handler indicate the system that it is virgin and waiting to registration information from a mobile device.

4.3.2 Acceleration sensing handler

The acceleration inputs for moving sensing system taken from GY-511 LSM303DLHC [40] acceleration sensor via SPI communication. Before VDRC registered or paired with a mobile device, the acceleration result will be observed. If the result is above the threshold, VDRC will warning the driver to go through the registration process.

The acceleration checking function will restart after the driver disconnect with VDRC. During the vehicle operation, the Bluetooth connection can be lost and the sensor cannot detect, but the application is previously registered, so the task of indicate the driver about connection lost will belong to the E-Wallet.

4.3.3 Bluetooth interface handler

The ESP32S support both traditional Bluetooth and BLE communication, and for this project the VDRC uses the traditional one. The Bluetooth communication simulated the UART communication so that the Serial read and write handler can take directly from Serial communication Arduino library provided. The difference is that this is not the default port but a virtual without any physical pins output, only for the link layer.

5 System feature results and evaluation

5.1 Signup virgin vehicle

Vehicle registration occurs mostly on mobile devices side, the VDRC only a place holder for such usage. Therefore, the performance of this task based on internet connectivity of the mobile device, since parsing HTML web page does not required a lot of local resource by a few attempts. There is no database that manage these registered vehicle and fully rely on information given by the government website, so the complexity and work load of the central server is reduced, but many loop holes are existed in such careless management.

5.2 Driver alert feature

This driver alert feature performance depending on the quality of acceleration chip. After multiple test run and calibration, this implantation is suitable for prototyping ideal of surround environment awareness feature of the system. If we replace the acceleration inputs with GPS input embedded natively on the VDRC itself, the evaluation of moving vehicle would be much simpler and reliable. However, this alternative inputs would reach its expectation if the GPS module itself has high accuracy as the mobile device, but the electronic module market for hobbyist does not have such device for an affordable price. Therefore, the decision of choosing the GY-511 LSM303DLHC is still appropriate for this project.

5.3 Automatic paying fee feature

Although this is the most important feature of the whole project, the requirement for central server connection is lower than user information fetching for displaying purpose. The design is described in chapter 4.1.10 has no blocking point between sending pay fee request and getting new GPS data, since they operating on 2 different handler with no condition attached.

5.4 Security data exchange

Security system used for TCP connection in this project is designed in such a way that all communication on the central server with any random mobile device is encrypted, and if a network protocol analyzer such as Wireshark [41] try to inspect the communication, it cannot detect and extract any information while the application is running. To realize these requirements, the security data exchange procedure shall have 3 main stages:

Share long term key

Exchange data with long term key

Erase long term key

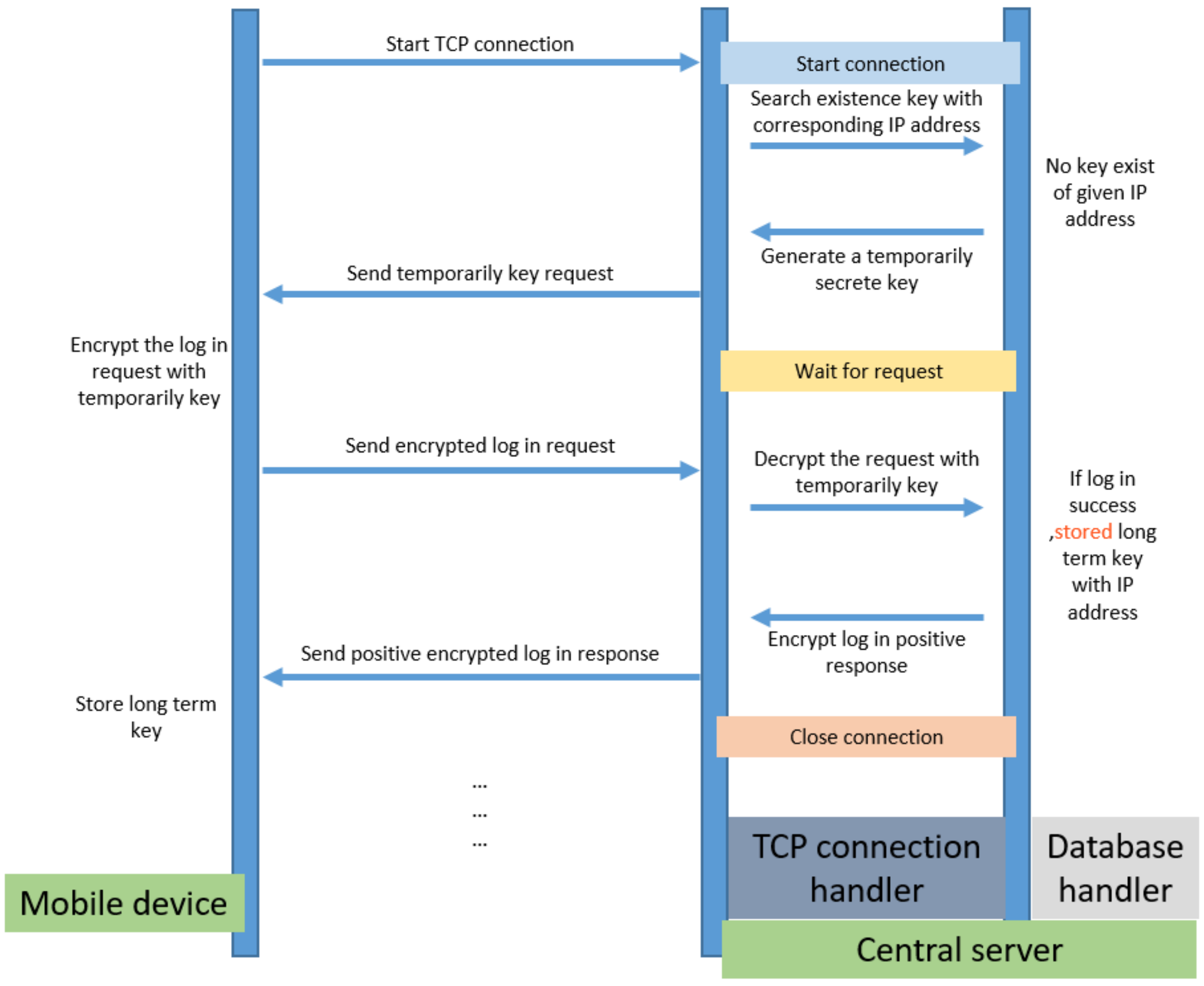

Start TCP connection
Start connection
Search existence key with corresponding IP address
No key exist of given IP address
Generate a temporarily secrete key
Send temporarily key request
Encrypt the log in request with temporarily key
Wait for request
Send encrypted log in request
Decrypt the request with temporarily key
If log in success ,stored long term key with IP address
Encrypt log in positive response
Send positive encrypted log in response
Store long term key
Close connection
…
…
…
TCP connection handler
Database handler
Mobile device
Central server

Figure 29: Start secured TCP communication diagram

The encryption and decryption handler from both application and central server is Fernet. On mobile device side, the Fernet library is provided for Java language. On the central server side, the Fernet library is provided for Python language [42]. A symmetric authenticated cryptography communication like Fernet must have a key, which is a URL-safe based64-encoded 32-byte string, and an encryption mode, which is AES 128-bit [43] in this project .Every request sent from a mobile device to the central server must be encrypted with a temporally key or a long term key. The long term key is the official generated secrete key that stored in database to served further request handling, and the temporally key is the newly generated key at the start of a connection.

When a TCP connection is established, the central server shall use DBH search the IP address of the connection request and search in the ipKey table of the database in Figure 30.

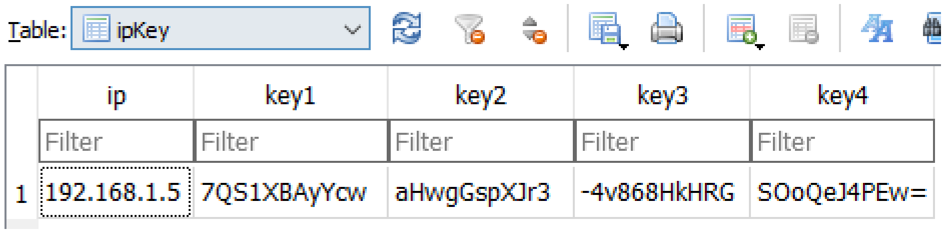


Table: ipKey

	ip	key1	key2	key3	key4
	Filter	Filter	Filter	Filter	Filter
1	192.168.1.5	7QS1XBAyYcw	aHwgGspXJr3	-4v868HkHRG	SOoQeJ4PEw=

Figure 30: ipKey table in database

If the database does not consist the IP address, which means the user have not log in on the mobile device yet, the TCP connection handler (TCH) shall generate a fresh temporally key [44] and send it to the mobile device in JSON message format. On the mobile device side, it will handle this message without attempt to decrypt it and store this temporally key for the login request. The connection between the mobile device and central server shall be kept until the user proceed login action or close after 5 minutes without any response from client side. After login information is ready to be sent as request, the mobile device shall encrypt this information using given temporally key. While the TCP connection is opening, which means the temporally key generated in TCH has not been deleted, the TCH shall decrypt this message and send to DBH to proceed authorization. If the login information is valid, the DBH shall store this temporally key as a long term key along with device’s IP address in the database. After that, the login success message shall be encrypted with new long term key and send to mobile device. When the mobile device decrypted the login success feedback using temporally key, this key also matured to long term key on mobile device side, in conclusion, the key sharing procedure is finished. On a circumstance where the login procedure is not success, no long term key shall be existed on both sides. At the end of this procedure, the TCH will close the TCP connection.

With a long term key created on both side of the communication, the TCH will wait for request after connection start, instead of generating new key. This communication is secured since the key sharing only occurred once and later data exchange is encrypted. However, the long term key will expired if there is no activity from the mobile device for 5 minutes. The TCH will automatically closed the connection, and when mobile device restart the connection, the sharing procedure will start again, but it is an automatically login procedure on application side if the E-Wallet is still active.

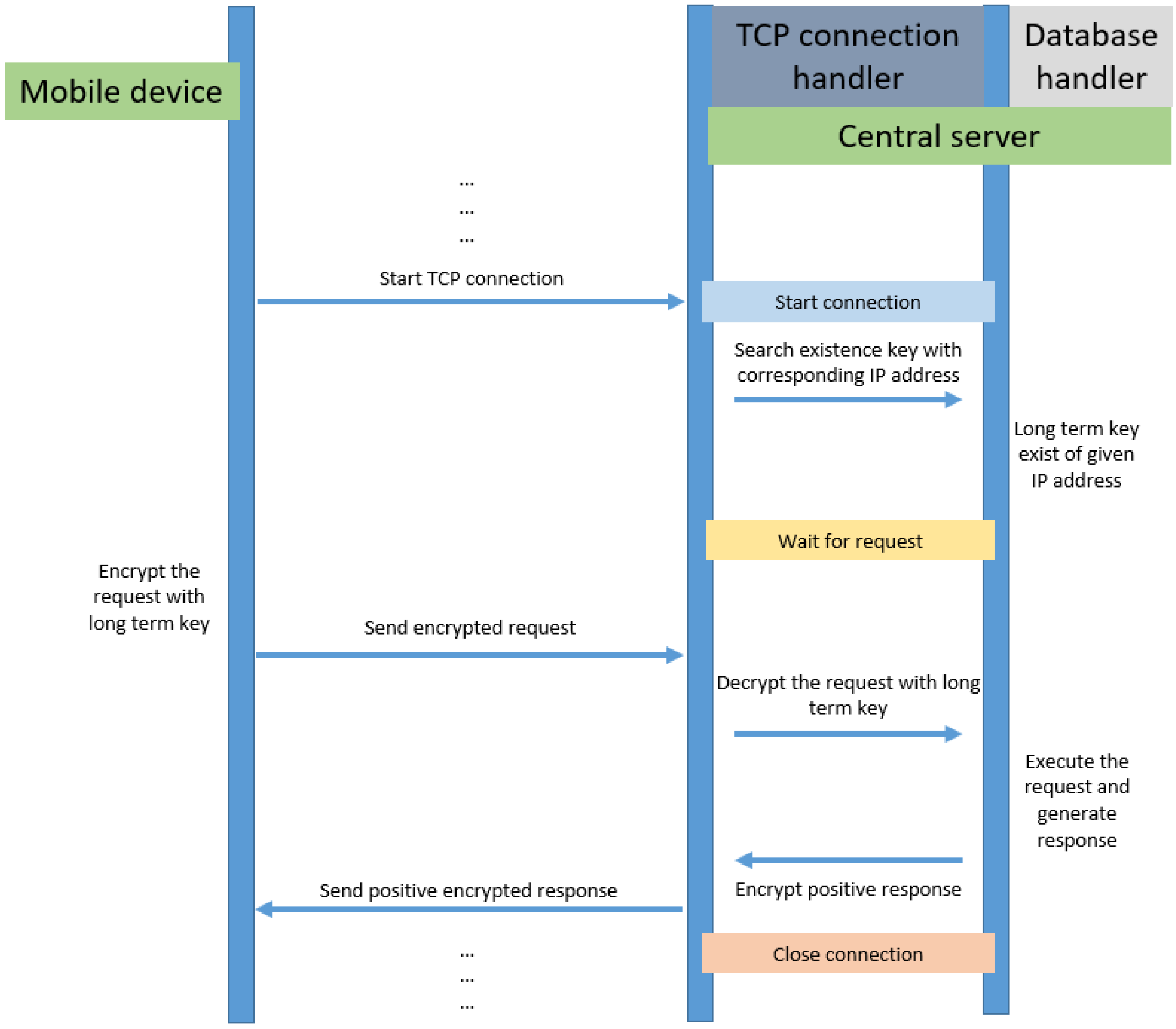

Mobile device
TCP connection handler
Database handler
Central server
...
Start TCP connection
Start connection
Search existence key with corresponding IP address
Long term key exist of given IP address
Wait for request
Encrypt the request with long term key
Send encrypted request
Decrypt the request with long term key
Execute the request and generate response
Encrypt positive response
Send positive encrypted response
Close connection

Figure 31: Secured TCP request/response diagram

When the mobile device send a log out request, the long term key will be erased be DBH from the database, and the user have to login again to restart the whole process.

5.5 Limitation and future improvement

This thesis focuses mainly on the development of E-Wallet mobile application and the external communication of it. Therefore, other aspect of the system might have many holes that need to be patched. The main downside of the VDRC is it can only be a placeholder for the vehicle information as what described in its name, instead of a backup device when there is E-Wallet around.

The limitation of the system design also comes from the negative cases where the driver is not cooperate with the constructed procedure. If they have try to not paying the road fee by not using the E-Wallet, there is nothing to stop them besides a warning from VDRC.

The central server have not been tested on a stress condition to evaluate the performance with multiple devices or under heavy bandwidth. The tested mobile device on other hand is a client, so the same situation will not occur.

The most important improvement for this system is on VDRC functionality. As discussed, it can be a backup device, even with emergency power source, instead of taking directly from the vehicle, to support the driver if negative cases ever occurred to them.

6 Conclusion

Finding a solution for a traffic solution is a challenge task because it involves shaping the habit of the consumer without any adjustment of the law. The automatic toll solution is the solution for contractor saving time any money with respect to the old tradition way. Although it may not seems to support the driver directly at any point of operating the vehicle on street, and especially complexity raises especially when the application has a money transaction and management application like E-Wallet, I believed this could help them as a position of a tax payer since the system increasing the integrity of the whole process. By eliminating unnecessarily, costly construction with heavy environmental impacts, the contactor company can reduce the time they return their profit and lower than chance any shady business occur when the time interval is stretching for such a long time Therefore, having a system focusing on security data protection and responsive can be justified and excepted for the big purpose. The availability of Iot resources and modular component with affordable price, it makes this project possible for a low budget and functional prototype to demonstrate a greater ideal for the society.

References

[1] "Toll road," [Online]. Available: https://en.wikipedia.org/wiki/Toll_road.

[2] N. M. J. K. Rajeshwari Hegde, "Automation of Toll Gate and Vehicle Tracking," in *International Conference on Computer Science and Information Technology*, Singapore, 2008.

[3] H. Cong, "De xuat xay 34 tram thu phi oto vao trung tam Sai Gon," VnExpress, 17 July 2019. [Online]. Available: https://vnexpress.net/de-xuat-xay-34-tram-thu-phi-oto-vao-trung-tam-sai-gon-3953543.html?fbclid=IwAR3UssgzzDKGp6SF7npb7B73ubHII9b6bgZICG-si9COqWi1lxs1ZBW-eZk.

[4] "vietnambiz.vn," 90% thi truong vi dien tu thuoc ve Zalo Pay, Moca va Momo, nhung doi thu den sau van con co hoi lon, 25 March 2020. [Online]. Available: https://vietnambiz.vn/90-thi-truong-vi-dien-tu-viet-thuoc-ve-zalopay-moca-va-momo-nhung-doi-thu-den-sau-van-con-co-hoi-lon-20200325151851209.htm.

[5] "Dang Kiem Viet Nam," Vehicle Registration Agent, [Online]. Available: http://app.vr.org.vn/ptpublic/ThongtinptPublic.aspx.

[6] "Configurations/Roles," [Online]. Available: https://www.amd.e-technik.uni-rostock.de/ma/gol/lectures/wirlec/bluetooth_info/k1_gap.html#Configurations/Roles.

[7] U. government, "GPS.Gov," U.S. government, 2013. [Online]. Available: https://www.gps.gov/multimedia/poster/.

[8] U. Government, "Featured Video: GPS SMC 2.0 Adaptability & Flexibility," GPS.Gov, 2019. [Online]. Available: https://www.gps.gov/multimedia/videos/.

[9] "Activity," Google, [Online]. Available: https://developer.android.com/reference/android/app/Activity.

[10] "Understand the Activity Lifecycle," Google, [Online]. Available: https://developer.android.com/guide/components/activities/activity-lifecycle.

[11] "Fragments," Google, [Online]. Available: https://developer.android.com/guide/fragments.

[12] "Layouts," Google, [Online]. Available: https://developer.android.com/guide/topics/ui/declaring-layout.

[13] "Android 3.3 release note," Google, [Online]. Available: https://developer.android.com/studio/releases#3-3-0.

[14] "Get started with the Navigation component," Google, [Online]. Available: https://developer.android.com/guide/navigation/navigation-getting-started.

[15] "Fixed start destination," Google, [Online]. Available: https://developer.android.com/guide/navigation/navigation-principles#fixed_start_destination.

[16] "findViewById," Google, [Online]. Available: https://developer.android.com/reference/android/view/View#findViewById(int).

[17] "Ma dien thoai Viet Nam," Wikipedia, [Online]. Available: https://vi.wikipedia.org/wiki/M%C3%A3_%C4%91i%E1%BB%87n_tho%E1%BA%A1i_Vi%E1%BB%87t_Nam.

[18] "Button," Google, [Online]. Available: https://developer.android.com/reference/android/widget/Button.

[19] "Bundle," Google, [Online]. Available: https://developer.android.com/reference/android/os/Bundle.

[20] "Text Watcher," Google, [Online]. Available: https://developer.android.com/reference/android/text/TextWatcher.

[21] "Class HashMap," Oracle, [Online]. Available: https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html.

[22] "Class IOException," Oracle, [Online]. Available: https://docs.oracle.com/javase/7/docs/api/java/io/IOException.html.

[23] "LocationCallback," Google, [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/location/LocationCallback.

[24] "FusedLocationProviderClient," Google, [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient.

[25] "setInterval," Google, [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest#setInterval(long).

[26] "Geocoder," Google, [Online]. Available: https://developer.android.com/reference/android/location/Geocoder.

[27] "getAddressLine," Google, [Online]. Available: https://developer.android.com/reference/android/location/Address#getAddressLine(int).

[28] "GoogleMap," Google, [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/maps/GoogleMap.

[29] "SupportMapFragment," Google, [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/maps/SupportMapFragment.

[30] "MarkerOptions," Google, [Online]. Available: https://developers.google.com/android/reference/com/google/android/gms/maps/model/MarkerOptions.

[31] "Set up bluetooth," Google, [Online]. Available: https://developer.android.com/guide/topics/connectivity/bluetooth#SettingUp.

[32] "Intent," Google, [Online]. Available: https://developer.android.com/reference/android/content/Intent.

[33] "Broadcast Receiver," Google, [Online]. Available: https://developer.android.com/reference/android/content/BroadcastReceiver.

[34] "Publish–subscribe pattern," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Publish%E2%80%93subscribe_pattern.

[35] I. Bluetooth SIG, "Configurations and roles," *in Bluetooth Core Spec, 5nd ed., Vol 3, Part G, Chapter 2,* p. 2223.

[36] "Connect as a client," Google, [Online]. Available: https://developer.android.com/guide/topics/connectivity/bluetooth#ConnectAsAClient.

[37] C. Macasaet, "Fernet Java," [Online]. Available: https://github.com/l0s/fernet-java8.

[38] "onScroll," Google, [Online]. Available: https://developer.android.com/reference/android/widget/AbsListView.OnScrollListener#onScroll(android.widget.AbsListView,%20int,%20int,%20int).

[39] "selectors," The Python Software Foundation, [Online]. Available: https://docs.python.org/3/library/selectors.html.

[40] "Cảm biến la bàn số và gia tốc GY-511 LSM303DLHC," IC dayroi, [Online]. Available: https://icdayroi.com/cam-bien-la-ban-so-va-gia-toc-gy-511-lsm303dlhc.

[41] "About Wireshark," Wireshark, [Online]. Available: https://www.wireshark.org/.

[42] "Fernet (symmetric encryption)," [Online]. Available: https://cryptography.io/en/latest/fernet.html#fernet-symmetric-encryption.

[43] "Advanced Encryption Standard," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

[44] "classmethod generate_key()," [Online]. Available: https://cryptography.io/en/latest/fernet.html#cryptography.fernet.Fernet.generate_key.