The Chinese University of Hong Kong
Academic Honesty Declaration Statement

Submission Details

Student Name	CHOW, Tsz Kui (1155093841)		
Year and Term	2019-2020 Term 2		
Course	IERG-4999-DJ01 Final Year Project II		
Assignment Marker	Professor CHEUNG Man Hon		
Submitted File Name	無標題文件.pdf		
Submission Type	Individual		
Assignment Number	2	**Due Date (provided by student)**	2020-05-18
Submission Reference Number	2660934	**Submission Time**	2020-05-18 10:51:30

Agreement and Declaration on Student's Work Submitted to VeriGuide

VeriGuide is intended to help the University to assure that works submitted by students as part of course requirement are original, and that students receive the proper recognition and grades for doing so. The student, in submitting his/her work ("this Work") to VeriGuide, warrants that he/she is the lawful owner of the copyright of this Work. The student hereby grants a worldwide irrevocable non-exclusive perpetual licence in respect of the copyright in this Work to the University. The University will use this Work for the following purposes.

(a) Checking that this Work is original
The University needs to establish with reasonable confidence that this Work is original, before this Work can be marked or graded. For this purpose, VeriGuide will produce comparison reports showing any apparent similarities between this Work and other works, in order to provide data for teachers to decide, in the context of the particular subjects, course and assignment. However, any such reports that show the author's identity will only be made available to teachers, administrators and relevant committees in the University with a legitimate responsibility for marking, grading, examining, degree and other awards, quality assurance, and where necessary, for student discipline.

(b) Anonymous archive for reference in checking that future works submitted by other students of the University are original
The University will store this Work anonymously in an archive, to serve as one of the bases for comparison with future works submitted by other students of the University, in order to establish that the latter are original. For this purpose, every effort will be made to ensure this Work will be stored in a manner that would not reveal the author's identity, and that in exhibiting any comparison with other work, only relevant sentences/ parts of this Work with apparent similarities will be cited. In order to help the University to achieve anonymity, this Work submitted should not contain any reference to the student's name or identity except in designated places on the front page of this Work (which will allow this information to be removed before archival).

(c) Research and statistical reports
The University will also use the material for research on the methodology of textual comparisons and evaluations, on teaching and learning, and for the compilation of statistical reports. For this purpose, only the anonymously archived material will be used, so that student identity is not revealed.

I confirm that the above submission details are correct. I am submitting the assignment for:

[X] an individual project.

I have read the above and in submitting this Work fully agree to all the terms. I declare that: (i) the assignment here submitted is original except for source material explicitly acknowledged; (ii) the piece of work, or a part of the piece of work has not been submitted for more than one purpose (e.g. to satisfy the requirements in two different courses) without declaration; and (iii) the submitted soft copy with details listed in the <Submission Details> is identical to the hard copy(ies), if any, which has(have) been / is(are) going to be submitted. I also acknowledge that I am aware of the University's policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the University website http://www.cuhk.edu.hk/policy/academichonesty/.

I also understand that assignments without a properly signed declaration by the student concerned will not be graded by the teacher(s).

	18/5/2020
Signature (CHOW, Tsz Kui, 1155093841)	Date

Instruction for Submitting Hard Copy / Soft Copy of the Assignment

This signed declaration statement should be attached to the hard copy assignment or submission to the course teacher, according to the instructions as stipulated by the course teacher. If you are required to submit your assignment in soft copy only, please print out a copy of this signed declaration statement and hand it in separately to your course teacher.

IERG 4999D Final Year Project II

Real-time recommendation system based on open data

BY
CHOW, Tsz Kui
1155093841

A FINAL YEAR PROJECT REPORT
SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIRMENTS
FOR THE DEGREE OF BACHELOR OF INFORMATION ENGINEERING DEPARTMENT OF INFORMATION ENGINEERING
THE CHINESE UNIVERSITY OF HONG KONG

May, 2020

Abstract

Dynamic traffic conditions lead to the challenge for drivers to select a suitable parking area that avoids time and charges lost due to this manually. In this study, we are trying to explore a new mechanism and users' interface, with the use of open data sources from the government, to improve the performance on finding suitable parking area that can reduce charges and time from traveling and parking. A system, which consists of a cross-platform client and an API server are created to prove the above concepts.

Tasks performed in Term 2 by me

In the project of term 2, I am responsible for the design and configurations on the client-side. This including the web view and mobile view. Work is doing the setup and configure on the website page of the system to provide web access on the system to the user through PC and a cross-platform mobile application to provide mobile access on the system to the user through IOS and Android.

Introduction

Background

This final year project is focusing on traffic issues. Traffic conditions are dynamic in the modern world. In everywhere, the conditions change time by time. This leads to the challenge for drivers to select a suitable parking area that avoids high road traveling time and expensive parking and traveling charges manually. Hence, this produced a large cost in time and charges, caused a large cost in drivers and industry.

For solving the above problem, with the development of technologies, some solutions are created by the third party and available in the market now. For example, “Parking loudly (停車大聲公)” [1], which is mobile application, is available on the market. This application is to try to help drivers to pick the parking area that is suitable for them.

From the current solutions, we found that some gap are existed to achieve the correct selection to a suitable parking area. There are two challenges for those solutions:

1) Users’ interface – only static user interface is provided in the existing solutions. The existing solutions are only provided a static user interface to users. In this static user interface, the users can merely pick their target parking area, with the static data provided by the operators of parking areas, and without the forecasting traveling time and expected a fee for traveling and parking. Besides, the selection of suitable parking areas is select manually. The static user interface is difficult for users to make a correct decision on the selection procedure of suitable parking areas due to the static user interface lacks essential dynamic information. *2) Mechanism* – The mechanism of selection to the parking area is static. In those existing solutions, the selection of a suitable parking area is manual. By using this mechanism, the consideration of real- time traffic status and possible charges are not contained in the decision of the suitable parking area. This means that the mechanism of selection to the parking area is static. The static mechanism brings difficulty in determining the parking area that is most suitable to save time and fees.

Annual Open Data Plans

This plan is an open data scheme driven by the HKSAR government. This plan is proposed by the Chief executive Carrie Lam’s “Policy Address” in 2018, to provide government materials to boost technology research, innovation, and smart city development. This scheme is started in 2019, about 650 datasets from different department of the HKSAR government were already released to the public, including transportation. [2] In this project, we will use a real-time dataset, *Parking Vacancy Data*, that provided by this plan, as the source of our data to build up our system.

Google Map Platform

Google Map Platform is a library of APIs and SDKs that provided by Google to let developers able to use the function of Google map in their project in different platform. Including place autocomplete, map display and route suggestions. In this project, we will use the APIs and SDKs in this platform in our system, to provide real-time traffic status, route suggestions and the real-time estimated time of the journey to collect the real-time dynamic traffic status and make use in the mechanism of our system.

Node.js

Node.js is a tool that provides the environment to allow JavaScript code to execute on the server-side. In this project, we will use Node.js to build up the server to process generating the results of the suitable parking area.

Flutter

Flutter is an open-source SDK developed by Google. This SDK is used in the development of the application on different platforms, such as Android and IOS, in only one set of Dart code. This can reduce the problem of development applications on different platforms, including time and work. In this project, we will use Flutter to develop the mobile side client application to allow users to access our system.

Methodology

Existing solutions are having difficulties in users' interfaces and mechanisms in finding a suitable parking area to save time and fees. In this study, we are going to explore a new mechanism and users' interface, with the use of open data sources from the government, to improve the performance on finding suitable parking areas that can reduce charges and time from traveling and parking. As a result, a cross-platform application, including web and mobile devices, will be the solution to problems. In the following sections, the details process in each step of the solution will be presented.

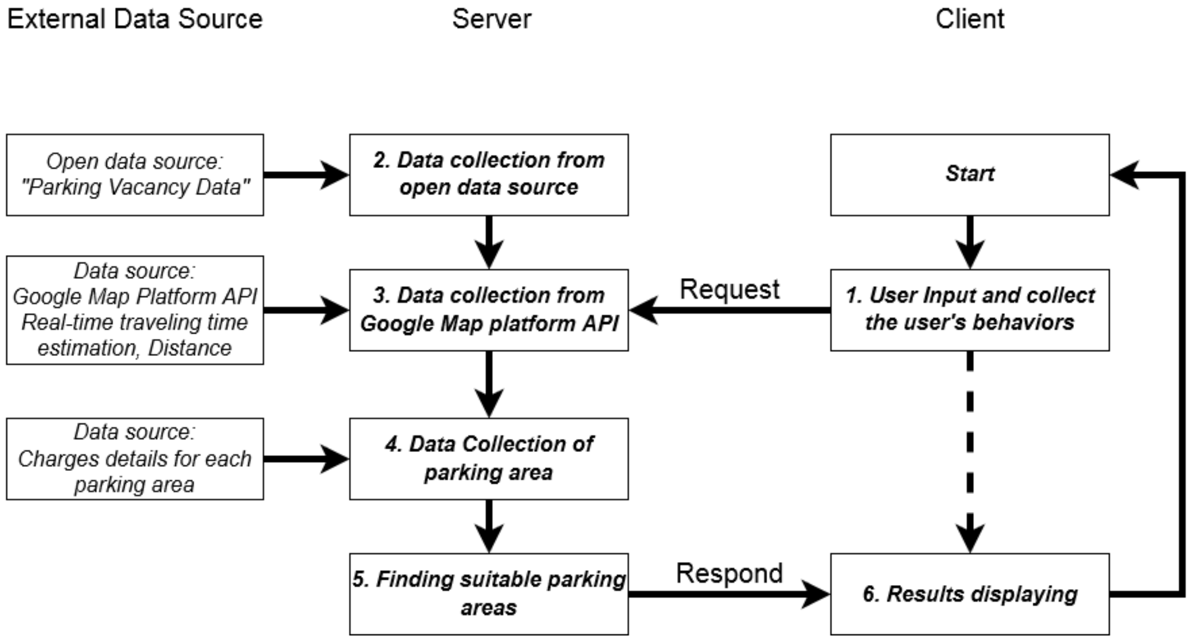


Figure 1: Solution workflow

1) Data Collection from users

In order to avoid manual procedures, the data collection will be one of the two input parts by users. In this part, we will require to collect 4 issues – starting location, destination parking time, and user' behaviors.

For the starting location and destination, it will support by Google Map platform API. Users will input two places respectively, Google API will automatically locate the place, and convert to the value of latitude and longitude for the next step.

For parking time, the user will be asked to input the parking time hourly. The parking time will use in finding the parking fee in the later stage.

For the user' behaviors part, we are going to collect them from users individually. There are 3 things we will collect. First is car type, we will divide the users' vehicle into 5 types. They are "private car", "motorcycle", "LGVs", "HGVs" and "Others" respectively. Second is oil consumption, which allows users to input the oil consumption detail of their car in a unit of L/km. The third is the searching distance, which allowed users to set a certain number of kilometers as the radius of the area that searching the parking area. Those 3 users' behaviors are store in a database with users' details that registered in the system, and those details stored in the database will be taken out and used in this stage to generate the results that fit the users' preference in the later stage.

2) Data collection from the open data source

The open data source can provide real-time data for the parking area. The parking area data will be given by an HKSAR government API, called "Parking Vacancy Data" [3]. This API will return the information of the parking area in Hong Kong, including the opening status, vacancy, price, and location of the parking area. In the server side, it will keep on checking the status of the parking areas in every 2 minutes to ensure the status is the up to date and reduced the time that needed to request of the newest parking status when we start to finding parking area. As mentioned in the searching distance concept and those users' preference settings in the previous section, a list of the parking areas will be listed out, which are meet to the users' preference set in the previous step.

3) Data collection from Google Map platform API

In this stage, we will get the real-time traveling time estimations and distance from Google Map platform API. For each parking area in the list, our application will receive two types of data from Google Map platform API: *1) Real-time traveling time estimations*. This estimation data consists of the "driving time" and "transit time", which are representing the estimated driving time from starting point to the parking area and the estimated time of using mass transport (i.e. bus, subway, train, tram and minibus) and walking from the parking area to the destination respectively in seconds. *2) Distance*. The data is static and it is representing the distance of driving and transiting mentioned in 1) respectively in kilometers.

4) Data Collection of parking area

"Parking Vacancy Data" provided information on the parking area in Hong Kong, but most of the provided items are not include the parking fee details. Therefore, we will get those details by a local JSON-format file in this stage. This file is created by collecting all the parking fee details of parking areas provided by *"Parking Vacancy Data"*, including the fee and charge policies.

5) Finding suitable parking areas

After the data collection procedure, we will analysis those collected data, and finding three parking areas, which are the cheapest in expense, the shortest in traveling time and the best in time and fee respectively. In the calculation, the total time and expense will be found by those formulas below.

Let "driving time" and "transit time" as a and b respectively (in seconds). Let "distance" as c (in km). Let the parking fees and the oil price per kilometers as \$d and \$e respectively. Total time = a + b …(1) Total expense = d + ce …(2)

After calculation, the results will compare with each other, and then return all the required three results. But for the best in time and fee mode, another formula will be used, it is shown in follow.

$$\boxed{\text{Best mode value} = (1) + (2) * 20 \ldots (3)}$$

6) Results displaying

After all the results are generated, the results will send back to the client and display on the users' interface. The user can choose the most suitable parking area from the results of three modes. After selection, the driving route from the starting point to the selected parking area and the transit route from the selected parking area to the destination will be shown on the users' interface.

In practice, the mechanism of finding results will run under a Node.js server, and multi-platform input and display client. The mechanism will build in JavaScript. Meanwhile, for the mobile part of the multi-platform client, Flutter will be used to finish the construction work. For the database that storing the users' details and preferences, we will use MySQL, which is an open-source management software of the database.

Results

The result of the whole project view will be presented in this section, in order to provide the details of each part mentioned in the methodology part. The presentation will divide into web view and mobile application view in here. For both view, you can access it by using this account.

Email Address: alexchow2014@gmail.com Password: sk105

1) Web view

In the web view, we registered a domain for providing the PC access to our system. The platform is at "rtpts.coms.hk". No security and safety issues are found on the domain and platform that we used here, although this is not a usual domain (coms).

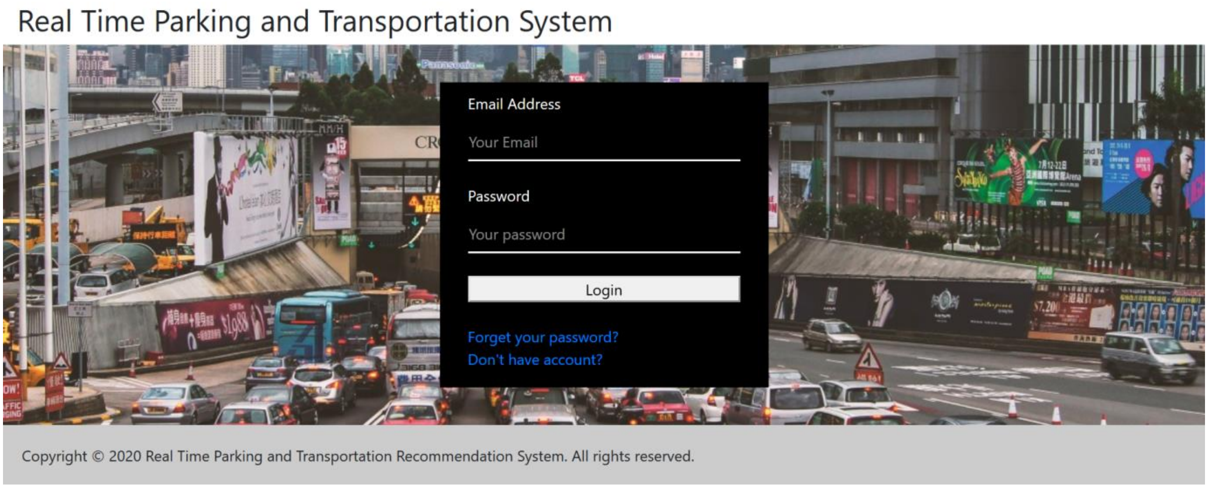


Figure 2.1 Login page of system

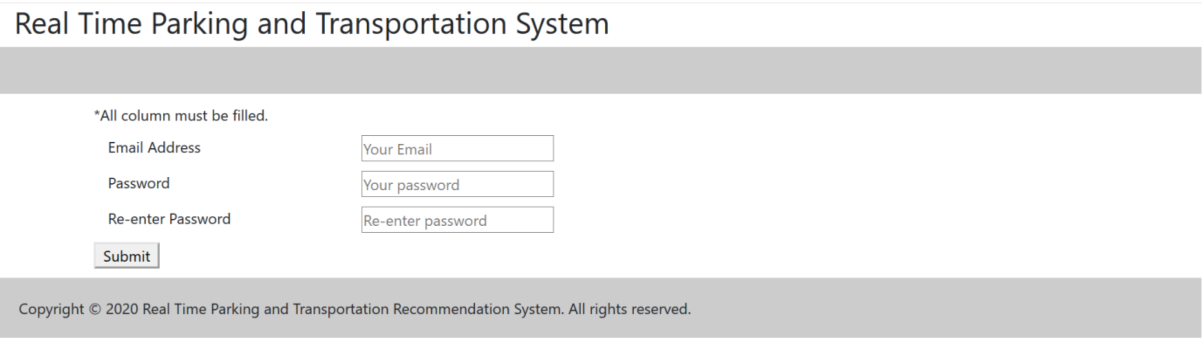


Figure 2.2 Registration page of system on the right

In figure 2.1 and 2.2, we are showing the account login and registration feature. To avoid the congestion of the system server, ensure the security of the system, and collecting users' preferences for further operation mentioned in the methodology part, we forced the entire user to register their account and login into the system before doing anything.

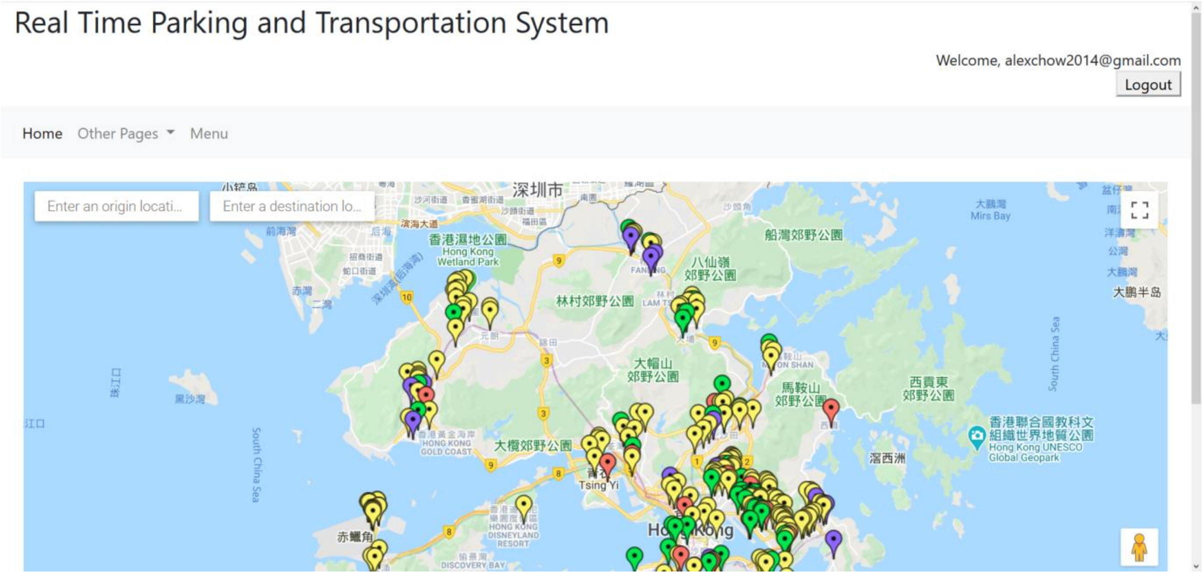


Figure 3 Main page of the system

Figure 3 is showing the main page of the system. This main page consists of a few features. First is the input box on the starting point and destination, this can start the searching in the parking area. Second is the marker on the Google map screen, those markers are presenting the location of parking areas provided by the open-source data, and current availability details of those parking areas. User can click on the marker on the Google map to get more details of the parking areas, including current availability details, the location of the parking area, and the contact details of the parking area.

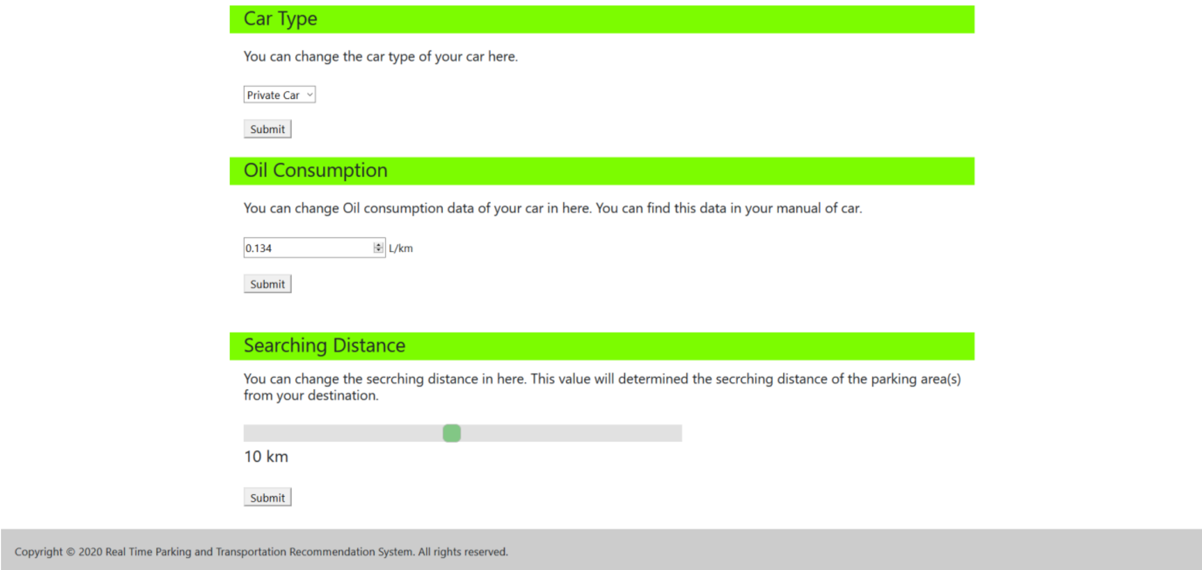


Figure 4 Menu page of the system

Figure 4 is presenting the menu page of the system, which allows the user to edit their searching preference, including car types, oil consumption, and searching distance. This page will first show the current details that access from the database that stores all the details. Users can edit the value and choices of each preference item and press submit to update their record on the database.

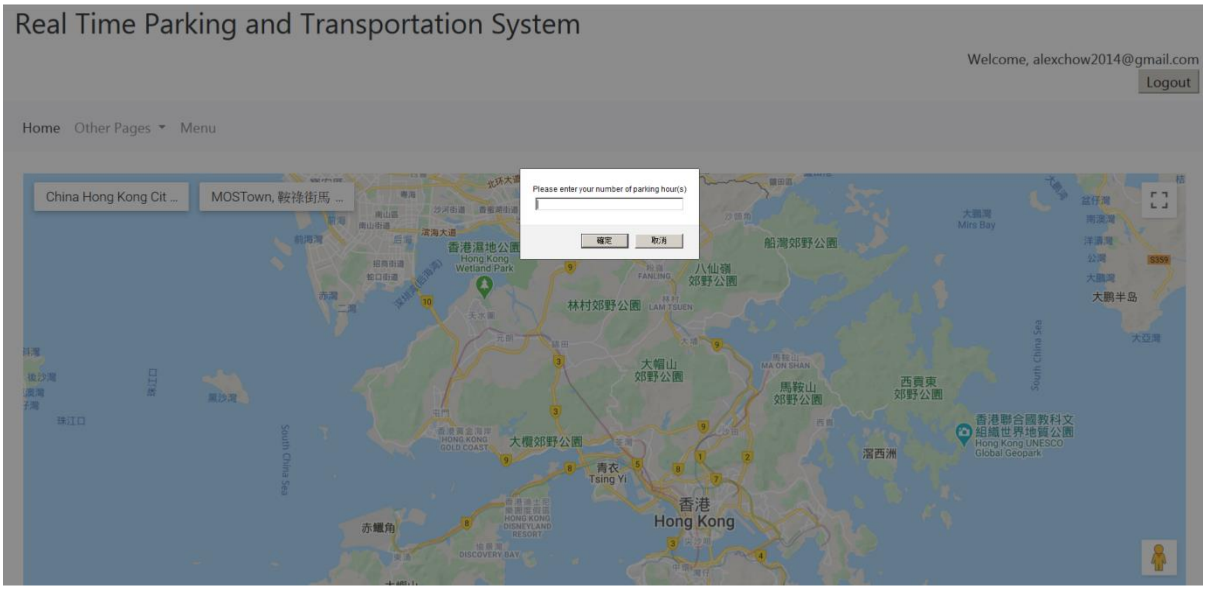


Figure 5 Input box for parking time in web view

Figure 5 shows the input box for parking time. This will show after the user entered the starting point and destination in Figure 3. Users can input the expected parking time on this page hourly.

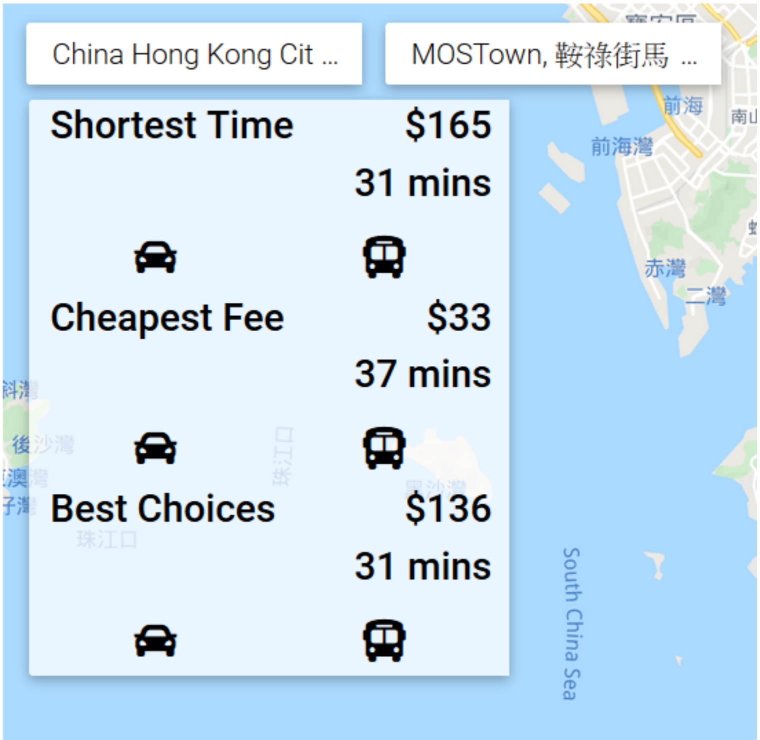


Figure 6 Result showing in web view

Figure 6 is presenting the result after the result is generated by the server and returns to the client. This shows 3 choices, which are the cheapest in expense, the shortest in traveling time, and the best in time and fee mentioned in the methodology part, with their corresponding estimation on time and fee respectively. Users can choose to adopt the most suitable parking area from the results of three modes.

The results of each choice are including driving route and transit route, user can choose to show the driving or transit route on the map for each choice.

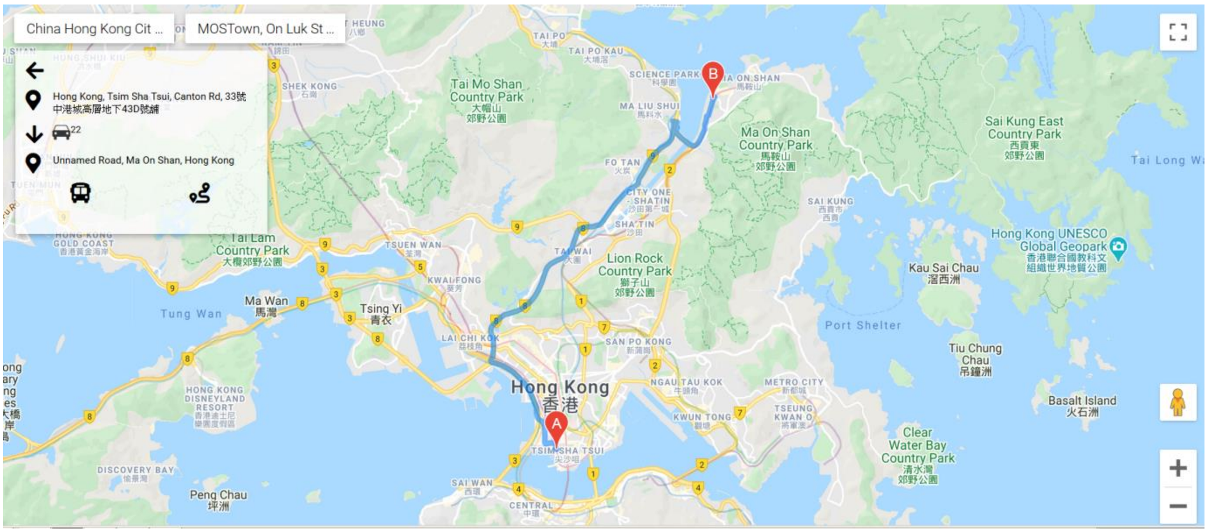


Figure 7.1 Showing driving route in web view

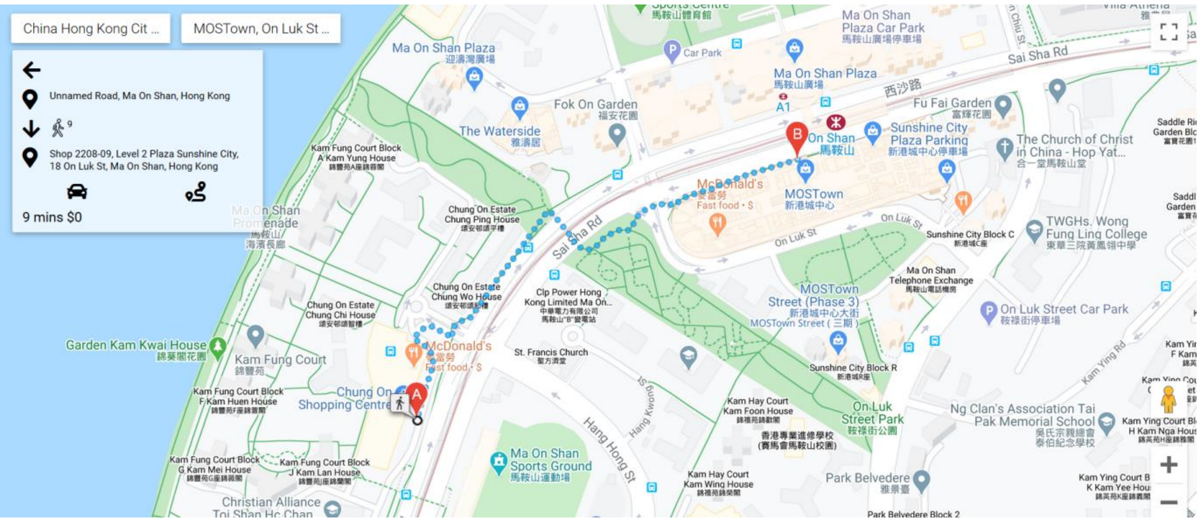


Figure 7.2 Showing transit route in web view

Figure 7.1 and figure 7.2 are presenting the route of the choices. Figure 7.1 and figure 7.2 are presenting the driving route and transit route of the selected choices respectively. The screen of presenting shows the suggested route and estimated time for both driving and transit (include walking and using public transport) respectively. The view of showing the transit route will also provide a transit fee to the user.

In both of the view that presenting driving route and transit route, they are having the button on using Google navigation tools to navigate user go to the destination, and a button to switch the view between presenting driving route and transit route.

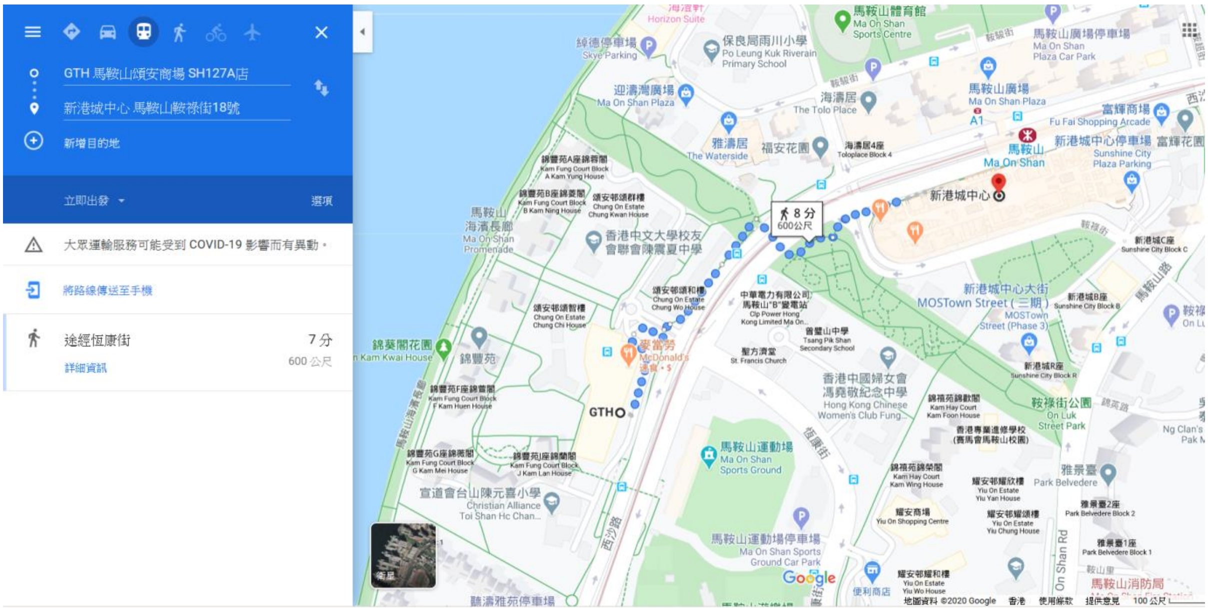


Figure 8 Google navigation in web view by clicking the Google navigation button on the page that showing transit route

2) *Mobile view*

In the mobile view, we created a mobile application to provide mobile side access to our system. This mobile application, as stated in the previous part, it is an application that using Flutter, and available on IOS and Android platform devices. Basic on the consideration of the user friendly on mobile, the mobile view has a different design with the web view.

The Android and IOS mobile application file is on the following:

Android .apk file https://drive.google.com/file/d/1cMzMVWCXSRnzTNW05PIeYeMA2osm7lwm/view?usp=sharing IOS .ipa file https://drive.google.com/file/d/1_5mVMQGq4sMrPLpjQckhEZIilP33dqQ3/view?usp=sharing

Details of the mobile view will show as follows.

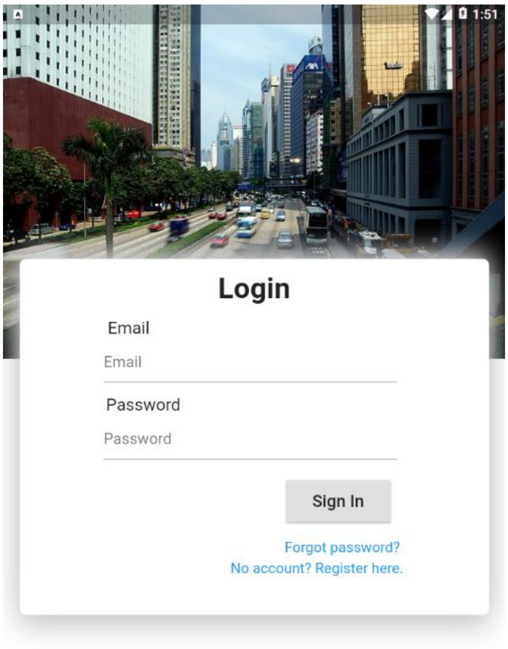
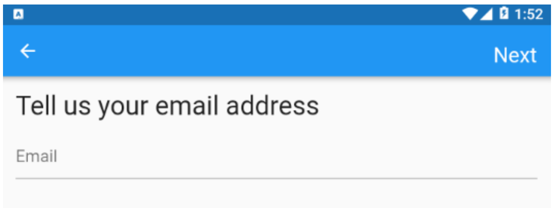


Figure 8.1 Login page of system in mobile view



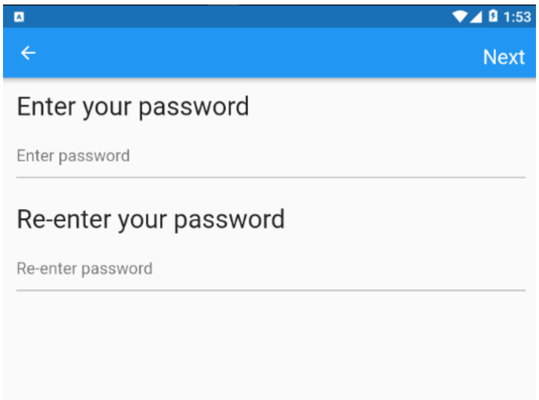


Figure 8.2.1 and 8.2.2 Registration page of system in mobile view

The login page and registration page of the mobile phone are shown in Figure 8.1, Figure 8.2.1 and Figure 8.2.2 respectively. The progress and functions of these two pages are the same as in the web view. However, the layout is created according to the requirements of the mobile device.

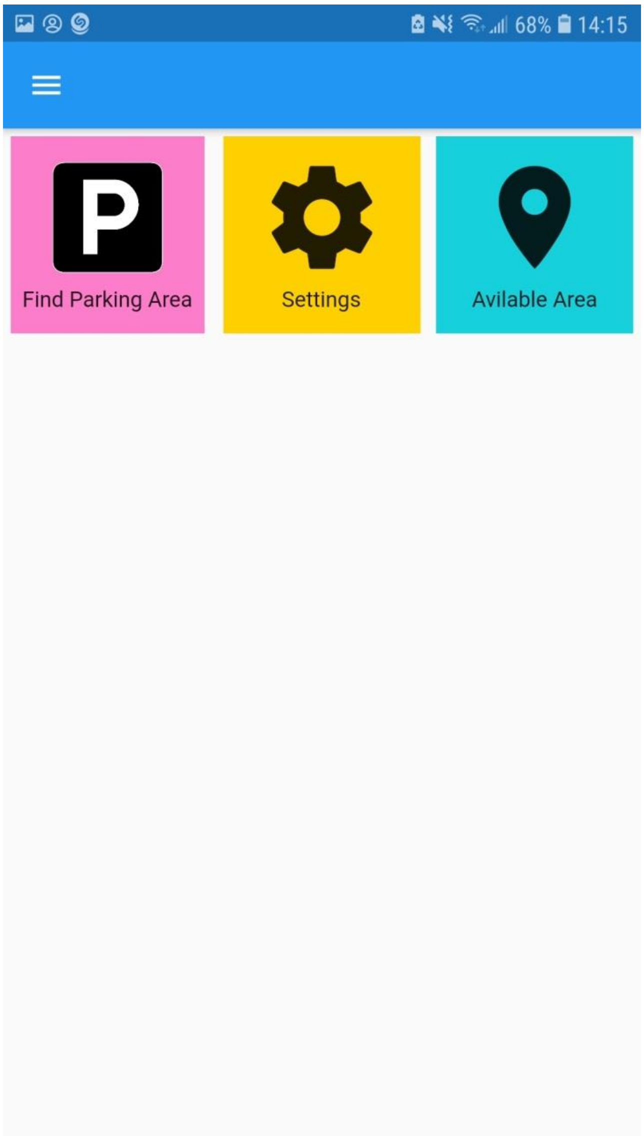


Figure 9 Main page in mobile view

Figure 9 shows the main page of the mobile application. This is the page that allows user to select their required service, such as find parking areas, settings on preference, and checking on the availability of parking areas. By clicking each of the icons on the screen, the system will enter the corresponding page.

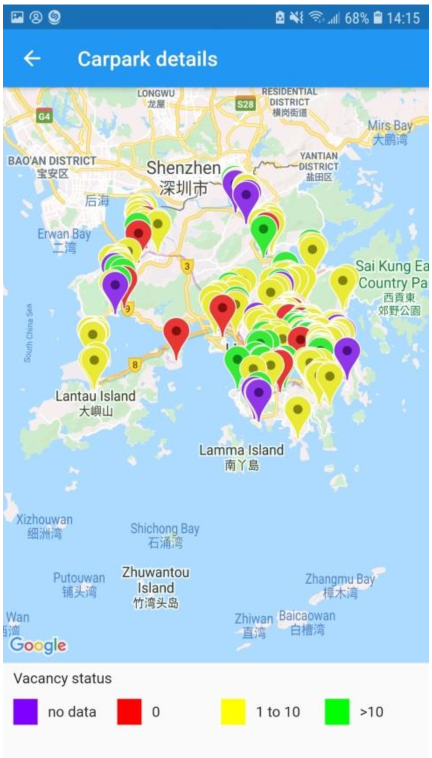


Figure 10 Availability of parking areas in mobile view

Figure 10 is showing the availability of parking areas. That is the same as in the web view, markers on the Google Map show the location of parking areas provided by the open-source data and current availability details of those parking areas.

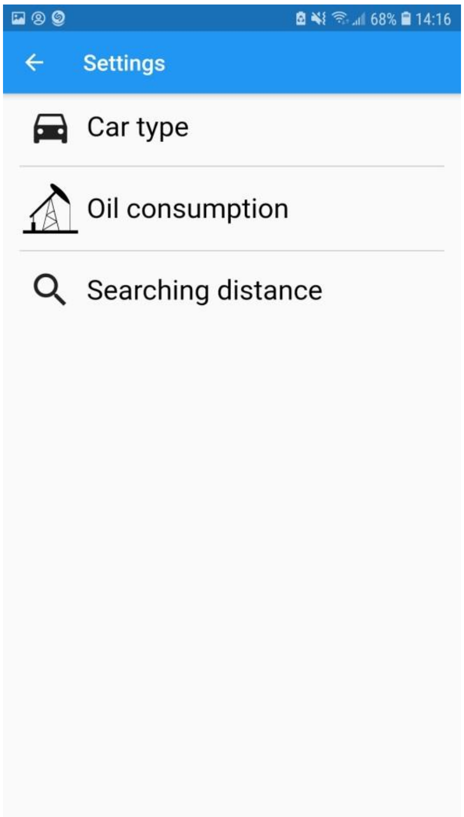


Figure 11.1 Menu page in mobile view

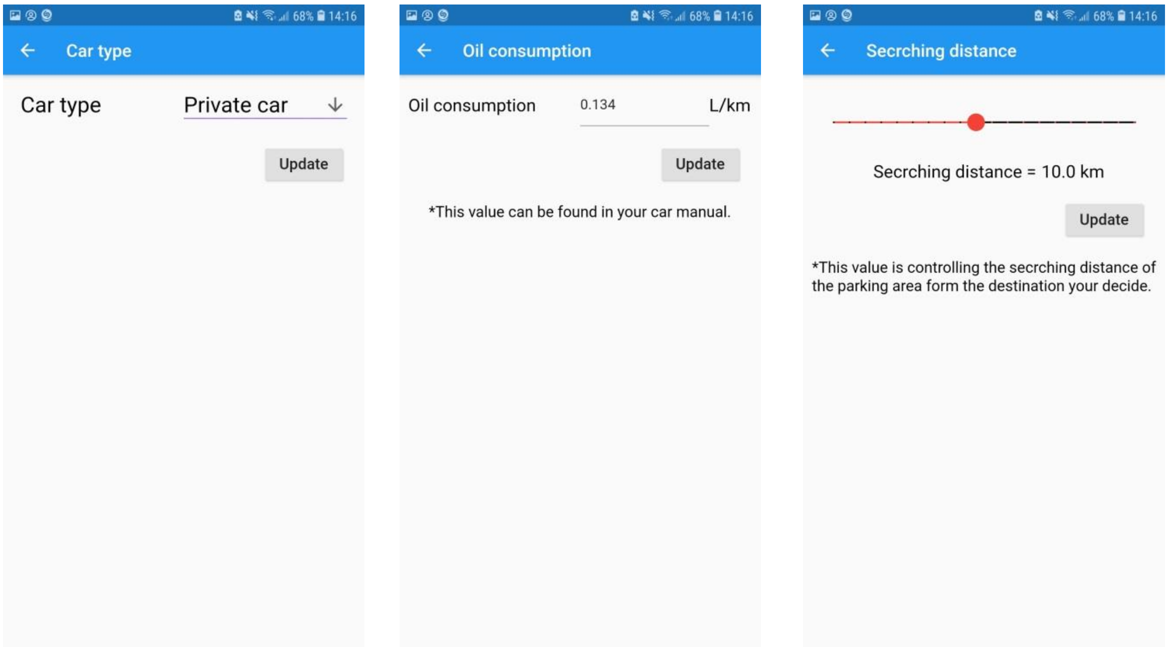


Figure 11.2 Configuration page of each of the items of users' preference in mobile view

Figure 11.1 and figure 11.2 are the series of pictures that presenting the menu page in the mobile view. In the mobile view, I put those 3 items of users' preference configuration into 3 different and separated pages, which are showing in Figure 11.2. Figure 11.1 is the menu page that provides access to those pages in Figure 11.2. The configuration process is as same as in the web view.

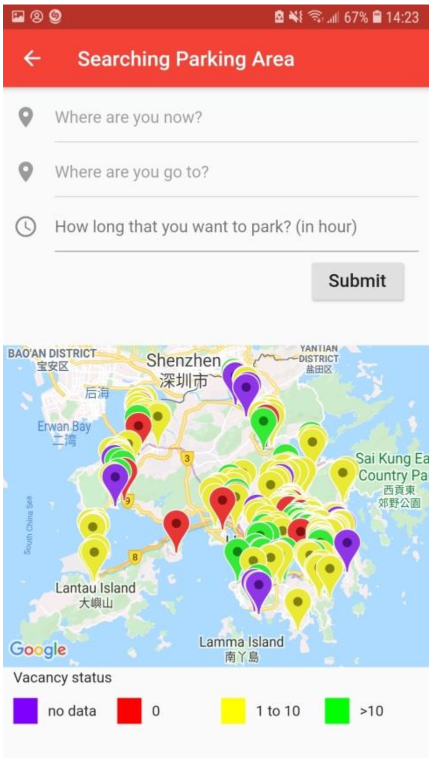


Figure 12 Input page in mobile view

Figure 12 is the input page, which has the same function in those on the main page of the web view. But in the mobile view, the expected parking time will be entered by the user with the starting point and destination together on this page.

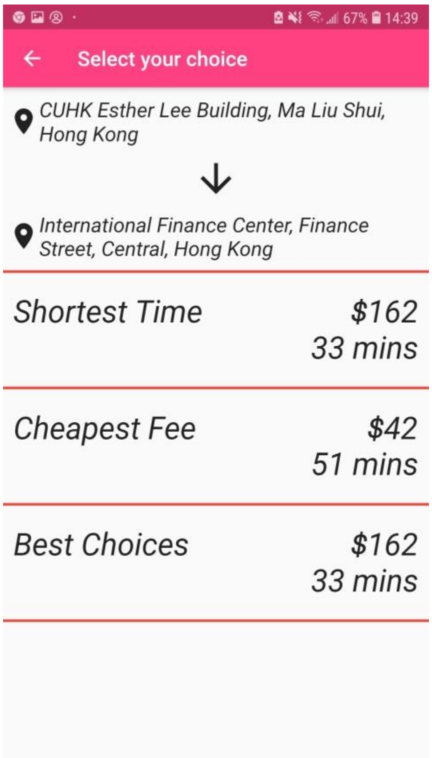


Figure 13 Result showing in mobile view

Figure 13 is presenting the result after the result is generated by the server and returns to the client. The features of this page are the same as in the web view. Users can select one of the most suitable parking areas from the results of three modes by clicking the relevant spaces.

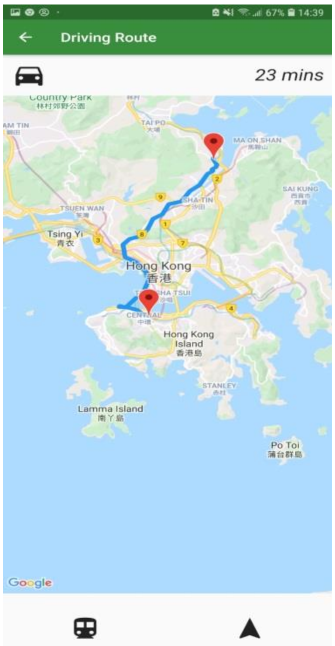


Figure 14.1 Showing driving route in mobile view

Figure 14.2 Showing transit route in mobile view

Figure 14.1 and figure 14.2 are presenting the driving route and transit route of the selected choices respectively. The details are the same as in the web view. Same as web view also, un both of the view that presenting driving route and transit route, the button of using Google navigation tools to navigate user go to the destination, and a button to switch the view between presenting driving route and transit route is presented.

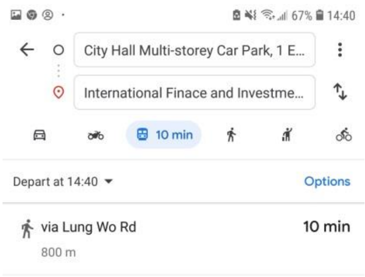


Figure 15 Google navigation in mobile view by clicking the Google navigation button on the page that showing transit route

Experiment

In the following part, in order to test our system, 15 datasets are used. Those data in the dataset are the essential input details, including *Starting Point, Destination, Choice* and *Parking Time*. Meanwhile, the searching distance and oil consumption are fixed as 10km and 0.134L/km respectively.

Dataset	Starting Point	Destination	Parking Time (in hour)
1	China Hong Kong City (22.3005315N, 114.1678015E)	Kam Tai Shopping Centre (22.4087594N, 114.2217721E)	10
2	Choi Po Court (22.5024902N, 114.1241018E)	MOKO (22.3231937N, 114.1723194E)	5
3	MOKO (22.3231937N, 114.1723194E)	Whampoa Garden (22.3037907999999N, 114.1892659E)	8
4	W Hotel Hong Kong (22.3049382N, 114.160669E)	Whampoa Garden (22.3037907999999N, 114.1892659E)	1
5	South Horizons (22.243373N, 114.1476512E)	Whampoa Garden (22.3037907999999N, 114.1892659E)	2
6	V city (22.3953878N, 113.9739331E)	Whampoa Garden (22.3037907999999N, 114.1892659E)	2
7	MOKO (22.3231937N, 114.1723194E)	MOStown (22.4241668N, 114.2314834E)	47
8	MOKO (22.3231937N, 114.1723194E)	IFC mall (22.285482N, 114.157879E)	36
9	CUHK Esther Lee Buliding (22.4139135N, 114.2086403E)	Amoy Plaza (22.3244614N, 114.2163021E)	48
10	Yuen Long Theatre (22.4415016N, 114.0229416E)	Hong Kong Maritime Museum (22.2866442N, 114.1620552E)	10
11	Yuen Long Theatre (22.4415016N, 114.0229416E)	Hong Kong Maritime Museum (22.2866442N, 114.1620552E)	5
12	Yuen Long Theatre (22.4415016N, 114.0229416E)	Hong Kong Maritime Museum (22.2866442N, 114.1620552E)	1

Table 1: Details of dataset used in experiment

In order to get the actual result and avoid some abnormal deviation due to error or network situation, when in the testing of all datasets, 3 trials will be done and calculate the average time of these 3 trials as the results.

The chart below shows the duration of getting the results and show on the user interface after we sent out the request on using the system in both web view and mobile view.

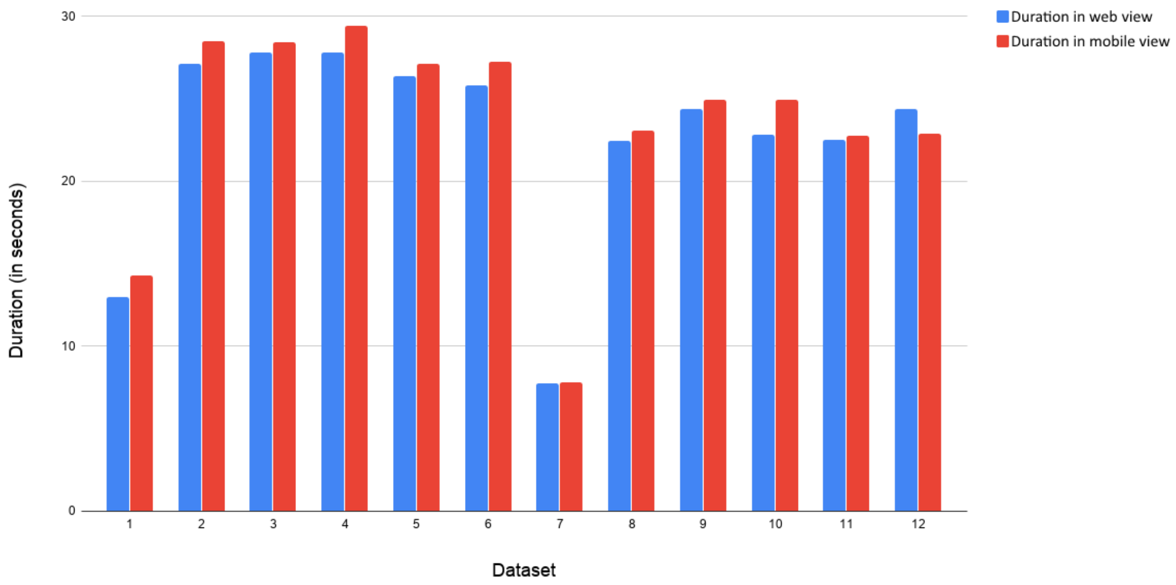


Figure 16 Duration of getting the results after request sent in both web view and mobile view, under the condition of searching distance is 10km, oil consumption is 0.134L/km.

From the above chart, all of the cases that we tested can receive the result by the system within 30 seconds, but a shorter time is required for the other cases. The mean duration in web view and mobile view are 22.685 and 23.455 seconds, with standard deviation are 5.916 and 6.112 seconds respectively.

According to a study of tolerable waiting time, it is shown that the waiting time for users is far away from the limit of stay attention to the progress. [4] In most of the case are required more than 20 seconds to wait. The reason for the long waiting time is that those cases need to consider more parking areas. And the destinations of those cases are the area that dense in the number of parking areas in the searching distance. Therefore, longer waiting time is required to concern and compare with all parking areas in the searching distance for generates the result.

In order to reduce the waiting time, we adjust the user preference of a testing user, which the email address is alexchow2014@gmail.com, and test with the adjusted searching distance, which reduced to 5km.

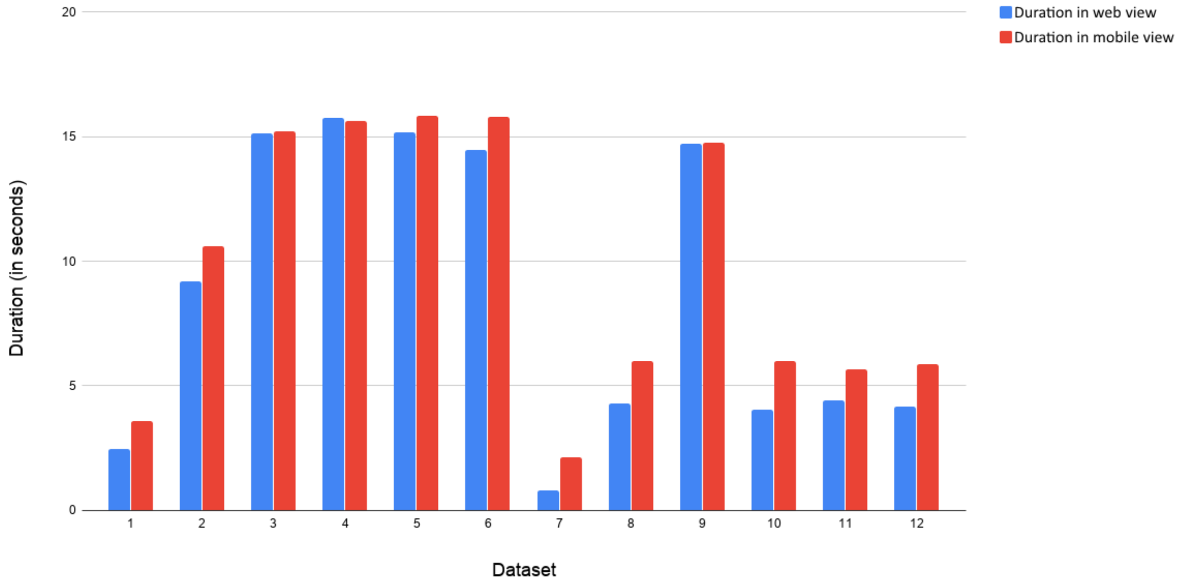


Figure 17 Duration of getting the results after request sent in both web view and mobile view, under the condition of searching distance is 5km, oil consumption is 0.134L/km

From the above chart, most of the cases that we tested can receive the result by the system within 15 seconds in a smaller searching area. The mean duration in web view and mobile view are 8.718 and 9.756 seconds, with standard deviation are 5.665 and 5.169 seconds respectively. According to the study of tolerable waiting time mentioned above, it is shown that the waiting time for users is near the limit of stay attention to the progress [4]. From the results, we can observe that the necessity of adjusting user preference by the user, because this can provide the flexibility to the user on the choices of the parking area. However, the cost of the larger searching distance to provide more choices of the parking area will be the longer reaction time, which may be non-reasonable to let users stay attentive.

The result shows that the reaction time of the system is reasonable for most of the users in the case with the suitable searching distance. The results imply that the mechanism of the system is complex, but it is still efficient to get the result. Therefore, we can conclude that the time performance of the system is good at this stage. However, the time performance of the system is not as good as the prototype. The reason is that the prototype is directly computed by JavaScript, the reaction time is much faster than using a server as it does not have time consumption on the communication time between client and server.

Conclusion and Discussion

In the above discussion, the importance of providing a dynamic solution to finding parking areas for the user has shown on the above part. Meanwhile, after the building work and experiment on our system, we can conclude that as follows.

First is a dynamic solution system on finding a parking area is a fully possible solution that can be applied to all users. Finding the source of parking areas, real-time vacancy situation, charge details, and real traveling time estimation to support the whole project to develop and realize are required in the actual application case. In the process of the realization, it is clear that enough sources of information that enables this progress are existed. Such as Annual Open Data Plans by HKSAR government and Google map platforms, they are providing the all necessary information such as the location of parking areas, traveling time from starting point to the parking area and to destination and real-time vacancy situation for those parking areas. Therefore, in the mechanism, we can state that the dynamic solution system on finding a parking area is a fully possible solution to helping to solve the problem of finding a suitable parking area to save time and fees.

Second, the system has a user interface that users no need to choose the parking area that suitable to them manually. As shown in the above, users are required to input a few details only, like starting point, destination, and parking time, with some pre-configured users' preferences, then it can find a suitable parking area and generate the results to users. No manual selection and other settings have required the user to do. Moreover, the user interface of this system provides real-time dynamic information on the vacancy situation. From the above situation, we can see the user interface is dynamic and able to reduce the chance of manual selection on the parking area thus can help the user to make a correct decision on the selection procedure of suitable parking areas.

From the mechanism and UI, this system shows that it is a fully possible solution, and providing a dynamic mechanism and UI to help the user to find a suitable parking area to save time and fees.

However, some problems are still occurred in the system and need to be solved before publishing to market.

First, limited number of parking areas is provided from the external source. In this time, the provided parking areas from Annual Open Data Plans by the HKSAR government are only a few hundred, which is only a small part of the total parking areas in Hong Kong. Meanwhile, there are mainly related to government property and public housing estates, lacking the private source of the parking area. Also, apart from the Annual Open Data Plans, there is no other open-source of the parking area is available on the market. If the service needs to be publishing to market, this will require more open-source of the parking areas or collection and collaboration from private operators.

Second, some essential details are not available on the API and require updates by manpower. During the development, some of the essential details, like parking fees and its policy, are not available on the API and required to collect them during the development by using manpower, including searching and visiting. In this situation, updating the information is performed in a not effective way if this product is published to market, as this will require extra time to maintain and update the information of the parking area. Therefore, it is required to consider building a management platform on the source of parking areas, including the price and fee policies. Meanwhile, the management platform should allow the update details from the public to speed up the update procedures.

All in all, this project on open-source data has already shown that open source data can be used in a lot of aspects. And the product of this project shows the potential of it on improving the performance on finding suitable parking areas, and able to reduce charges and time from traveling and parking of users. Also, this can have the potential for further development. Such as more parking area sources from operators, improvements on the server to achieve further shorter waiting duration.

Reference

[1] Play.google.com. (2019). [online] Available at: https://play.google.com/store/apps/details?id=com.alfred.parkinglot&hl=zh_HK [Accessed 26 Oct. 2019].

[2] Info.gov.hk. (2019). *Over 650 datasets to be released in first year of government open data plans (with photo/video)*. [online] Available at: https://www.info.gov.hk/gia/general/201901/03/P2019010300255.htm?fontSize=1 [Accessed 31 Dec. 2019].

[3] Data.gov.hk. (2019). Parking Vacancy Data (One-Stop Version) | DATA.GOV.HK. [online] Available at: https://data.gov.hk/en-data/dataset/hk-ogcio-st_div_04- carpark-info-vacancy [Accessed 27 Oct. 2019].

[4] Nah, F. , “A study on tolerable waiting time: how long are Web users willing to wait?”. Lincoln, Nebraska, U.S.A., Rep. 2004. Available at: https://www.researchgate.net/publication/220893869_A_Study_on_Tolerable_Waiting_Time_How_Long_Are_Web_Users_Willing_to_Wait.

Appendix – weekly log book

Week 1:
→ Conference: progress follows up
→ Development: development on the mobile client

Week 2:
→ Conference: with instructor, reporting work, progress follow up, showing the interim state of the whole platform
→ Development: development on the mobile client

Week 3 - 10
→ Development: development on the security settings of the mobile application client and website client

Week 11 - 18
→ Development: development on user preference part
→ Document: Preparing thesis

Week 19 - 20
→ Experiment on the system
→ Document: Preparing thesis and PPT