Parkbuddy – Find My Car Android Mobile Application

Flora Stephanie Francis[1], Nurfauza Jali[2], Ahmad Hadinata Fauzi[3], and Suriati Khartini Jali[4]
Faculty of Computer Science & Information Technology,
Universiti Malaysia Sarawak, 94300, Kota Samarahan, Sarawak, Malaysia.
renlenox.np@gmail.com

***Abstract*—As a result of technological progress, smartphones become an excellent choice for the user to make their life easier. This paper discussed how to locate parked vehicle using mobile application. The situation of forgetting where last vehicle location was parked and trying to remember it can become problem to some people especially to those who are having deterioration of memory such as a short term memory and dementia. This mobile application helps these users to locate their vehicle by utilising the global positioning system (GPS) system. This paper presents and critically analyses the system developed to solve the problem.**

***Index Terms*—Smartphone; Android; Global Positioning System; GPS; Parking; Vehicle; Mobile Application; Vehicle Location; IoT.**

I. Introduction

One of the trivial issues faced by some individuals is spending the time to search for their vehicle while trying to remember the last location of their vehicle. Mobile devices such as smartphones had become an excellent choice for the user to make their life easier such as to handle the simple daily task. By taking advantage of this opportunity, a mobile application is developed to enable the user to locate back their vehicle. This application can help the user to locate their vehicle by utilising the global positioning system (GPS) system. It also helps to pinpoint the current location of the vehicle and then mark the location on a map for the vehicle whereabouts. One of the reasons this application is used to solve the problems because mobile devices are portable. The user interface for the application is more user-friendly compared to the typical software systems. Hence, no hardware such as keyboard or mouse needed for the user to interact with the application. Nowadays, users can find many vehicles related applications that can help them to perform almost all of daily tasks and plenty of mobile applications can be used by the users on the go [1].

The main aim of this paper is to present the design and analyse of the mobile application that can pinpoint the location of the vehicle being parked. It is also able to implement a full-fledged Android mobile application which can pinpoint the vehicle location using global positioning system (GPS).

This paper organised as follows, Section I introduces the background study and overview of the system, while Section II explains on the methodology used to build the system. Section III discusses the system and user requirements followed by the system design along with the user interface design. In Section IV-VII, the design, system implementation and user interface are explained. Section VIII summarises the testing and results. Contribution, conclusion and future works are drawn up in Section IX and Section X, respectively.

II. Literature Review

A. Vehicle locator application

The vehicle locator applications come in handy and very suitable for perpetual vehicle seekers. The mobile devices must have global positioning system (GPS), a location sensor and orientation sensor for it to function properly. Three of the components are important for a locator application to provide navigating instruction for the user. The mobile application will retrieve current geographic location on maps from mobile devices GPS and remember that location as the parking spot of the vehicle. The user is given two options to track vehicle location whether using map or compass. Locating vehicle in a big or multilevel parking building can be tricky, but it can be countered by providing an extra functionality to ease the searching session. One of the ways is to providing vehicle details such as vehicle type, plate number, time of last seen or taking a picture and the most important component is to locate the position of the vehicle and provide return route to the location.

B. Evaluation of the existing system

There are three types of existing vehicle locator application that has been evaluate and compare then based on the result of comparison, a proposed application can be design and developed. The three applications that had been review are Car Locator by Developer_Ako [2], Car Finder by Jarvis Lin [3] and Parked Car Locator by Christian Martell [4]. Based on the evaluation table (refer Table 1), proposed application is inspired by implemented some of the strong point and counter the weakness of each application.

Table 1
Comparison of features between existing application and the proposed application

	Parked Car Locator	Car Locator	Car Finder	ParkBuddy – Find My Car
Save current location	√	√	√	√
Track location using map	√	√	√	√
Track location using compass	X	X	X	√
Add vehicle record	X	X	√	√
Capture vehicle image	X	X	√	√
Share location and record of vehicle	X	X	X	√
Set reminder to notify user	X	X	X	√
Simple and user friendly interface	X	√	√	√
Prompt user to active GPS	√	√	√	√

Nowadays, there are lots of existing mobile applications that can be an aided tools to perform almost most of daily task such as locating vehicle [5]. The following is the drawbacks of the existing mobile applications: (1) Unclear navigation; (2) No sharing ability; (3) Tracking options other than map; (4) Vehicle record section is not provided; and (5) Reminder or timer to notify user.

III. Software Development Model

The system development model is proven to be an excellent approach to develop interactive applications with a more graphical user interface [6]. This method is a form of guesswork and estimation which resembles prototyping model but suitable for a project with a few outline requirements, more open-ended and less formal than other software development model. The exploratory development model is one of a system development model (SDM) used to develop a system or product. It consists of planning and testing different designs from initial to final version until the system meets the user’s requirement and satisfaction.

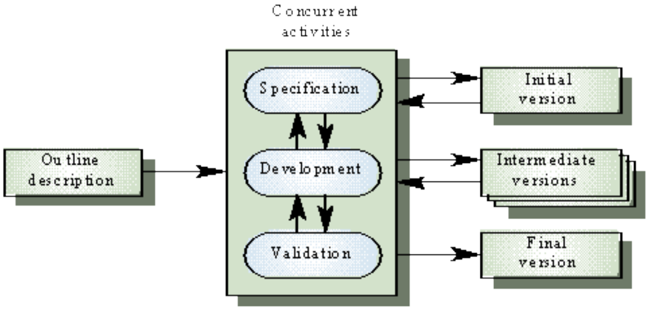


Figure 1: Exploratory development model [7]

A. Requirement gathering and analysis

This is the first step in this project which is to create a list of outline system requirement and gathered information needed to build the application. This outline requirement determines the functionality of the application which can help the user to face their main problem. The requirement gathered is needed before building the application for a better understanding on how to develop the application and fulfil the main idea. All the requirement analysis and specification documentation are done in this step before proceeding to develop the application. The project specification documentation consists of the project objectives, background study, problem statements, project requirement, application design and ParkBuddy application that is developed after in-depth study on several existing applications.

B. System design and development

The system design and development phase is to design and build the system based on the requirements specification documentation. The system design helps in describing and specifying the hardware, software, system requirement and its architecture. Then, the system is built accordingly to the system design and requirement documentation. This phase is one of the longest phases in development model and related to step 2 in exploratory software development model which is the application design and development.

As mentioned before, this application is designed and developed according to the requirements gathered in the first step of this software development model. All the Unified Modeling Laguage (UML) [8] models of this project are designed to show the flow of the application. The features and functionalities of the application are the most basic model based on the main idea and need to fulfil the outline requirements of the project. This application is the first version of the application that needs to be built and developed according to the user interface. This user interface design is the rough sketch of the functionality and user interface of the application. The programming language used to develop the system is Java in a Block-based programming language by using MIT Apps Inventor 2 as the software tools.

C. System implementation and testing

In this phase, the system is tested after the designed and developed phases. Besides, the system is tested to make sure it meets the requirements and working properly. Hence, this phase is the longest phase in this development model because the iteration process of developing and testing phases until it fulfils user’s and system’s requirements. After that, the application is finally can be deployed when the functionality of application fulfils the user and specification requirements. The users decide the final product. This phase is related to step 3 until step 6 in exploratory software development model which is the application testing and modification.

In this phase, the current application which is the first version of the application is tested by the users. After the first version is tested, the users will evaluate the application based on its features and functionality. Later, any changes in requirement and design are decided by the users or no changes are made if the current application is satisfying enough for the user. Thereafter, the second version of the application is developed to counter all the weakness and bugs existed in the first version as well as to improve the application. This addition is done after the first testing and evaluation. Then, the previous version (*n*th version) of the ParkBuddy application is tested and evaluated again by the users. The new version is tested and evaluated to decide whether any new improvement needed onto the latest version of the application. This step keeps on repeating until the users satisfied with the features and functionality of the application. The testing strategy and planning are also conducted in this phase.

In the implementation step, the users will evaluate the latest version of the application. If the latest version meets user's requirements and satisfaction, the version will be finalised. The finalised version must fulfil system requirement proposed by the user and related to the main idea of the project which able to help users faced their problem. Finally, the latest version presented to the users and implemented.

IV. Requirement Analysis and Design

The system requirement analysis and design of the ParkBuddy application are discussed in detailed. This chapter enlists all the system requirement specification and behaviour description of the system for this project development. In order to derive all the necessary system requirement, a clear understanding of the ParkBuddy application needs to be done.

A. *User requirement*

The user's requirements of this project are gathered through observation based on the current scenarios of the problem. Through observation, there are several requirements that can be gathered which include how the ParkBuddy application should perform to help the user face their problem (functionality) and the design of ParkBuddy application (user interface). The ParkBuddy application requirement includes the ability to pinpoint and track the exact location of the vehicle being parked by utilising the global positioning system (GPS) and mobile devices sensor component.

B. *Functional requirements*

A functional requirement is a description of specific behaviour of the function in the system. The functional requirements include the output of specific function when a certain condition is met. The functional requirement include able to:

- detect and save the current location.
- track back location saved earlier.
- show the latest location and guide the user to saved location when the user presses the "Compass" button.
- notify the user when the GPS is not active and prompt user to active GPS manually, and the user can set time reminder to notify them about their vehicle.
- take a picture and save it as a record for user reference.
- share the vehicle location and vehicle details saved earlier is recorded.

C. *Non-functional requirements*

A non-functional requirement is a description on how the system should behave and its limitation. The non-functional requirement includes as shown in Table 2.

D. *Hardware and software requirements*

Hardware and software requirements are the limitation and compatibility of the system in a different kind of environment. The requirements include as shown in Table 3.

E. *ParkBuddy Requirement Modeling*

The use case diagram of UML is used to show the interaction between the user and system's features (refer Figure 2). Based on the use case diagram, the user is able to add a new record of vehicle location and manipulate the record by editing, updating or deleting existing record. The user is also able to save and share the vehicle's location to another user. Besides, the user will be able to track back the current location of the vehicle.

Table 2
Non-Functional Requirements

Non-functional requirement	Descriptions
Performance	The application must be able to respond within 10 seconds after the user enables GPS before starting the application and able to run smoothly.
Reliability	The application is able to communicate and compatible with other devices functionality such as camera and social networking service. The graphical user interface must be user-friendly and understandable.
Availability	The application is able to get the location from devices GPS and retrieve the details to display on the screen.
Maintainability	The application maintenance must be practical (not time-consuming) and economical.
Portability	The application is an Android application that compatible with all handheld devices with Android operating system and global positioning system (GPS).

Table 3
List of Hardware and Software

Hardware	Software
Laptop with Intel CORE i5 Processor, mobile devices (smartphones) with location sensor component embedded in it and 4GB of SD card.	• Windows 7 and higher • Mozilla Firefox 51.0 and higher • Google Chrome 31.0 and higher • Android Operating System 6.0.1 (Marshmallow) and higher • MIT App Inventor 2 • MIT AI2 Companion Apps • Google Maps API V3 and higher

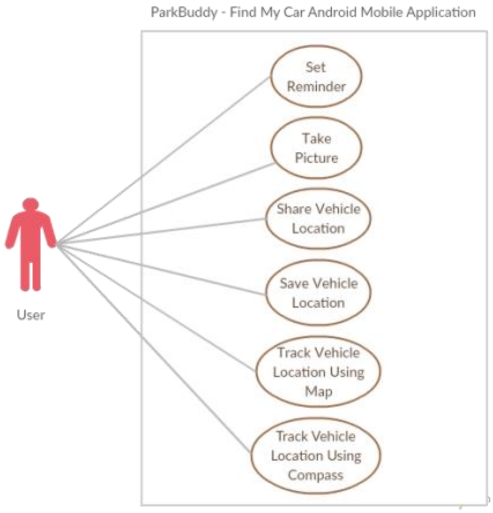

ParkBuddy - Find My Car Android Mobile Application
Set Reminder
Take Picture
Share Vehicle Location
Save Vehicle Location
Track Vehicle Location Using Map
Track Vehicle Location Using Compass
User

Figure 2: Use case diagram for ParkBuddy application

The activity diagram is used to illustrate the flow of the system activities. This activity diagram is used to describe the functionality of the system. Figure 3 depicts the flow of the system and specifies to determine what and how the user can perform the use case.

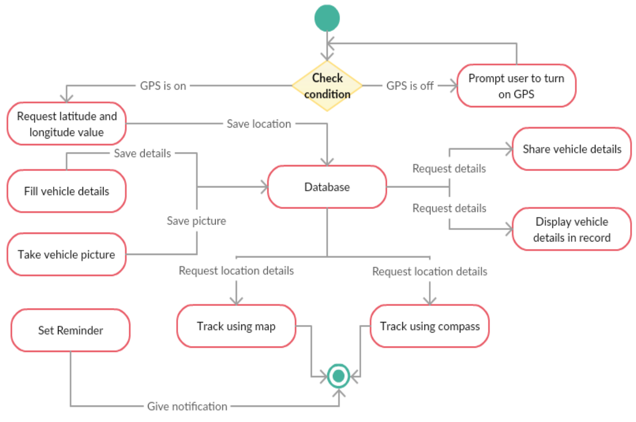


Figure 3: Activity Diagram for ParkBuddy application

V. Application Design

The design of the application is briefly explained in this section. Figure 4 illustrates the collection of ParkBuddy features. These features are inspired and built based on the evaluation of the existing application. Upon review, most of the existing application is not user-friendly and interactive. Application with the user-friendly interface will increase its usability because of the user is able to communicate and navigate with the application [9]. Hence, ParkBuddy application has a simple and clean yet offers an interactive user interface design.

The application can share the location of the vehicle with other users which can be useful in an emergency. For example, owner of the vehicle is unable to retrieve the car because of the emergency case, so he or she will be able to share vehicle location with the other person (i.e. family, spouse) that has a spare key.

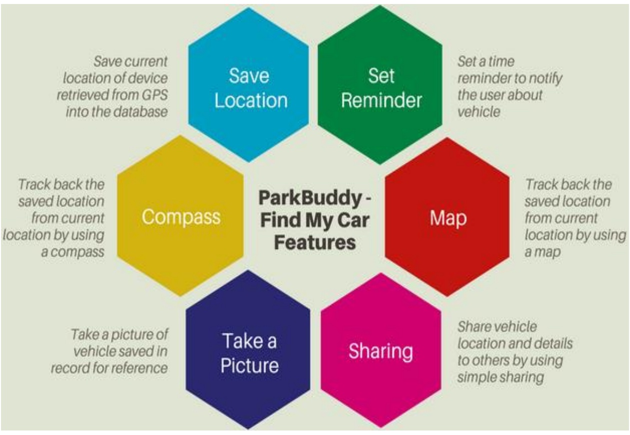

Save current location of device retrieved from GPS into the database
Save Location
Set Reminder
Set a time reminder to notify the user about vehicle
Track back the saved location from current location by using a compass
Compass
ParkBuddy - Find My Car Features
Map
Track back the saved location from current location by using a map
Take a picture of vehicle saved in record for reference
Take a Picture
Sharing
Share vehicle location and details to others by using simple sharing

Figure 4: Features of ParkBuddy application

Furthermore, the user also able to set reminder related to vehicle or parking. The application will send a notification to user’s devices (i.e. smartphones, tablet) to pick when the time that user set is reached. Besides that, other feature offers by the application is to send a notification to the user if the GPS is not activated, so it will prompt the user to activate the GPS manually. The user can also take a picture of their vehicle and it will auto-saved into the record for future reference. The pictures will auto-update when the user takes a new picture. The user can add vehicle details that include vehicle type and plate number to be saved into record together with the picture. This application will give the user power to choose on how to track the vehicle location whether using a map or orientation compass. The devices will collect coordinate value through GPS and location sensor before saving it into the database. The location of saved location will be shown in map and user able to track back location. For orientation compass, the coordinate value will be retrieved and after that, the bearing between two locations will be calculated using formula [10]. The formula shown is used to calculate initial bearing (azimuth) from the start point to the end point. The saved location earlier will be the end point, and current location will be the start point. This compass will guide the user to the end point of location.

VI. System Implementation

The system development is the phases involve when developing and implementing the application while system flow shows the architecture flow of the application and its logic. Figure 5 shows the development process of this project.

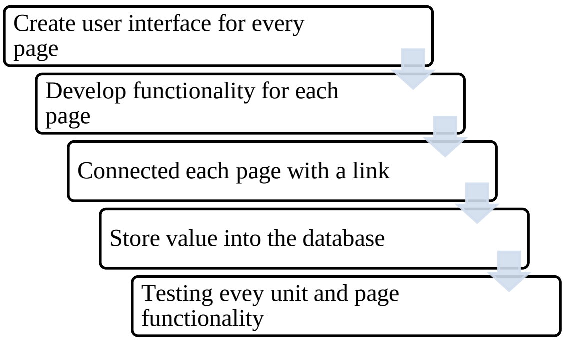
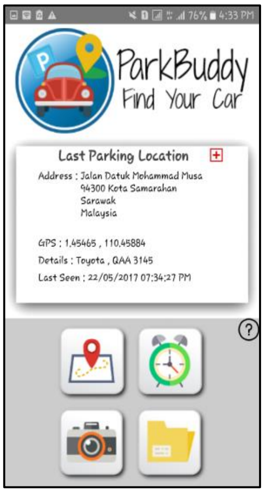

Create user interface for every page
Develop functionality for each page
Connected each page with a link
Store value into the database
Testing evey unit and page functionality

Figure 5: Application development phase

VII. User Interface

The user interface of the application allows the user to use it and control the application operation according to their needs.

The interface of this application is simple and neat in order to properly show all the available functions. This application consists of five pages in total. They are *home*, *track location*, *compass*, *record* and *set reminder pages*. Figure 6 depicts the main interface and the track location of the vehicle via map feature of the application.



Last Parking Location

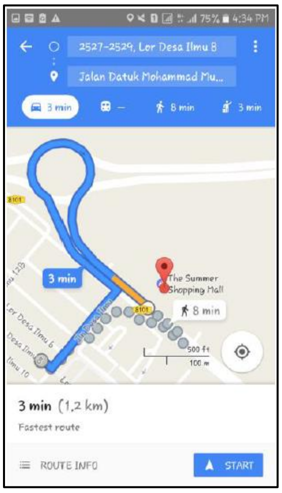
Address : Jalan Datuk Mohammad Musa
94300 Kota Samarahan
Sarawak
Malaysia

GPS : 1.45465 , 110.45884

Details : Toyota , QAA 3145

Last Seen : 22/05/2017 07:34:27 PM

(a)



(b)

Figure 6: (a) Main screen and (b) map tracking feature of the ParkBuddy application

VIII. Testing

The result of the testing performed on the usability, functional requirement, non-functional requirement and field testing. The functional testing is performed to test every function in this application from retrieving GPS coordinate, set time reminder, save record, add record, taking a picture and perform location tracking. The non-functional requirement testing is performed based on few aspects which is the performance, reliability, availability, maintainability and portability. The field testing is performed to validate the usability and effectiveness of application in real time situation. Every function available in the application is used during the field test and confirmed the functionality through this test.

A set of questionnaire is distributed to perform the usability testing that cover all sort of vehicle driver from every age, working background, gender and country. The focus type of respondent for this usability testing is drivers that has tendency to forget their vehicle location with the total of 30 people. Based from the result, most of the respondents agree that the flow of the application is easy to understand and learn. This proved that the application user interface is user friendly. While based from result of the next question, majority of the respondents think that this mobile application able to help them tracking back their vehicle easily. Another half of respondents agree that the features provided by the mobile application is suitable in helping them tracking back vehicle location, for example the vehicle record section. While the rest minority agrees that the user interface for this mobile application is clean and easy on the eyes or more features should be implemented in the future or a seamless and beautiful user interface should be created. The overall response from the respondents is very positive because majority of the respondent rate the mobile application as good while the rest rate it as an average mobile application. In conclusion, the mobile application that has been developed fulfills its objectives and gave a good impression to its users.

IX. Contribution

The ParkBuddy mobile application gives the developer some high commercial values when developing this prototype such as low development cost for the equipment and can be mass produced. This application needs no additional cost as it requires no hardware because most mobile electronic device is already embedded with different kind of sensors such as location and orientation sensors. In addition, this application can be mass produced as it does not require extra cost to maintain or upgrade server as it only used small database and storage can be saved inside the devices. Hence, this application is a cost-effective solution for maximising the value of commercial value and does not exceed budget limits.

X. Conclusion and Future Works

The ParkBuddy – Find My Car mobile application has been developed by utilising Global Positioning System (GPS), location sensor and orientation sensor embedded in mobile devices to help user to track back their vehicle location. The features include saving current location of vehicle by using GPS system, tracking back the saved vehicle location from record created by using map or compass, adding vehicle details record and taking vehicle picture for future reference.

However, ParkBuddy application has a limitation just like other applications that include the user interface, functionality and sensors accuracy. This type of limitations can be enhanced with new technologies and development techniques in the future as new technology is mushrooming. There are more enhancement can be made on the mobile application, one of the future enhancement is to increase the accuracy of the sensor to retrieve location. This solution can be fulfilled by integrating Wi-Fi system as another solution to retrieve location besides using Global Positioning System (GPS). The Wi-Fi system can be used indoor tracking if GPS signally is weak. Another future enhancement that can be implemented is by adding more functionality such as parking rates calculator. This new feature can help notify the user how much they need to pay for parking fees in the mall and how many parking coupons should they display.

Acknowledgment

We would like to acknowledge and thank Faculty of Computer Science and Information Technology, UNIMAS Innovation and Universiti Malaysia Sarawak.

References

[1] S. Edelstein, *'Dude, where's my car? Check out the best Android car apps'*. [online] Digital Trends, 2016. Available at: http://www.digitaltrends.com/mobile/best-android-car-apps/amp/ [Accessed: 20 Sept. 2016].

[2] Develpor_Ako. Car Locator (Version 1.2) [Mobile Application Software], 2015. Retrieved from https://play.google.com/store/apps/details?id=developerako.com.carlocator

[3] J. Lin, Car Finder (Version 1.1.3) [Mobile Application Software], 2016. Retrieved from https://play.google.com/store/apps/details?id=developerako.com.carlocator

[4] C. Martell, Parked Car Locator (Version 1.0) [Mobile Application Software], 2013. Retrieved from https://play.google.com/store/apps/details?id=appinventor.ai_martell_christian.ParkedCarLocator

[5] Y. B. Thosar and V. P. More, "Location Based Task Reminder System Using Android Mobile," *International Journal of Scientific Research Engineering & Technology (IJSRET)*, vol. 4, no. 3, pp. 282-286, 2015.

[6] J. Sametinger, A. Stritzinger, and J. Kepler, "Exploratory software development with class libraries," In *Proc. 7th Joint Conference of the Austrian Computer Society and the John von Neumann Society for*

Computing Sciences, pp. 24-31, 1992.

[7] R. Abdul, *'Software Development Cycle for Data Management System - Rob Abdul Digital Expert'*, [online]. Rob Abdul Digital Expert, 2015. Available at: http://www.robabdul.com/softwaredevelopment/software-development-cycle-for-data-management-system/. [Accessed: **1**9 Nov. 2016].

[8] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley Publ. Comp, 1999.

[9] Exousia.tech, *Importance of User Interface in App Development*. [online], 2016. Available at: https://www.exousia.tech/importance-of-user-interface-in-app-development/ [Accessed: 19 Nov. 2016].

[10] C. Veness, 'Movable Type Scripts - Calculate distance, bearing and more between Latitude/Longitude point'. [online], Movable Type Scripts, n.d.. Available at: http://www.movable-type.co.uk/scripts/latlong.html [Accessed: 10 Dec. 2016].