ISSN 2581 – 3781

AJCSE
ASIAN JOURNAL OF COMPUTER SCIENCE ENGINEERING

Available Online at www.ajcse.info
Asian Journal of Computer Science Engineering 2018;3(3):47-54

RESEARCH ARTICLE

An Android Based Intelligent Anti-theft and Tracking System for Automobiles

Amit Mishra[1], Joseph Elijah[2], Jamilu Yahaya Maipan-uku[3], Aisha Awal[2]

[1]Department of Computer Science, Baze University, Abuja, Nigeria, [2]Department of Mathematics/Computer Science, Ibrahim Badamasi Babangida University, Lapai, Nigeria, [3]Department of Maths/Computer Science, Faculty of natural science, Ibrahim Badamasi Babangida University, Lapai, Nigeria

Received: 31-04-2018, Revised: 30-05-2018, Accepted: 25-06-2018

ABSTRACT

Theft of automobiles has been on the increase in recent time. The purpose of this research work is to develop an Android-based anti-theft and tracking system for automobiles in real-time, it includes feature necessary for securing automobiles against threat. The system will be relying on the Android operating system; an open source mobile platform. This is because Android provides a rich set of services for mobile developers which will be invaluable for this project, chiefly among them is the location-based service and short messaging service functionality. Software development life cycle methodology is used with object-oriented approach for the development of the system.

Key words: Android, Anti-Theft, GSM, GPS Tracking, LBS

INTRODUCTION

In Nigeria, one of the most prized possessions is a car. This is even more true in urban areas where of most city work-force who reside in satellite town which are far flung from their workplaces, transportation of such population becomes paramount, in such scenarios as one described, owning a car is a necessity.

However, ownership of such a highly valuable and expensive property is becoming a dilemma, as vehicle is becoming susceptible to car theft on a daily basis. Most car owners purchase an automatic lock system, which is most easily hacked by such criminals. Furthermore, the security agencies in Nigeria are not properly equipped to tackle the menace of car theft, leading to unfavorable news in the battle against car theft. Therefore, the weight of securing vehicles lies heavily on the owner. A successful implementation of this proposed system will provide a cheap, friendly, and efficient technology for vehicle monitoring and location in the event of theft. This will simplify the tedious task of endless search for stolen vehicles by easily locating such vehicles, which may then serve as vital intelligence to security agencies which are responsible for handling such a task.

Address for correspondence:
Amit Mishra,
E-mail: i.amitmishra@gmail.com

Car theft has become an increasing occurrence in the society as the volume of car increases. The stealing of one's car may have several impacts which may range from physical, economical to psychological.

According to the Nigerian police force, between January and July of 2012, 600 cars were stolen in Lagos State alone, and about 400 were recovered. In 2007 and 2008 about 907 and 669 cases of car theft were recorded with 321 and 282 cars recovered in 2007 and 2008, respectively.[1] It should be noted on the one hand that this is reported cases only; some missing vehicles are not reported to the law enforcement agencies. On the other hand, this report covers only Lagos state Nigeria.

This research work seeks to proffer a software system solution which will reduce the risk of an automobile, which uses the solution, from getting stolen, and in the case of such car being stolen, effort for recovery will be enhanced by the system.

Background of the study

The project is to develop an anti-theft and tracking system for automobiles using Android mobile device. An Android mobile operating system (OS) device will be concealed in the vehicle intended to be tracked. This device will be responsible for the tracking of the vehicle which hosts it. Android is one of the most popular mobile phone OS.

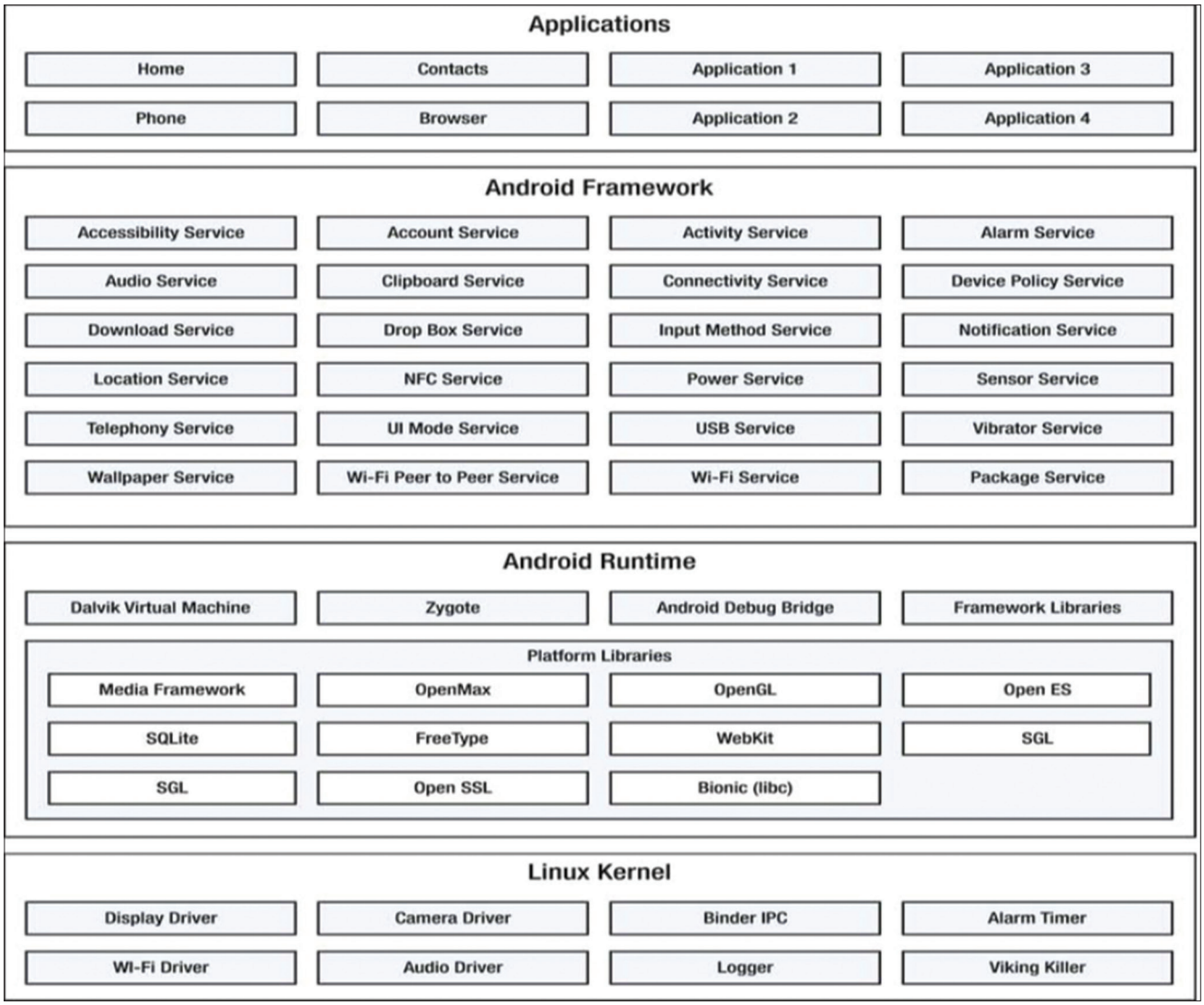

Figure 1: Android architecture

Android Inc. was acquired by Google in 2005 thereby having legal right to the OS.

In general, location data can be represented in one of two ways; in spatial terms or as text descriptions. A spatial location is represented in the latitude-longitude-altitude coordinate system. Latitude is 0–90° north or south of the equator, while longitude is 0–180° east or west of the prime meridian, a line that passes through Greenwich, England. Altitude is a representation of height above sea level in meters. Text description, on the other hand, is the representation of location as a street address, including city and country.[2,3]

The Android mobile OS

The Android OS is the name of the Linux based OS owned by Google and supported by the Open Handset Alliance. Android is used as an OS for devices such as cell phones, tablets, and notebooks. Google bought the original developer of the software, Android Inc., in 2005. Android's mobile OS is based on a modified version of the Linux kernel. Android's kernel (core of the OS) was derived from Linux but has been modified by Google developers. Android is also open source, which means developers can customize the OS for different phones and applications. This is why different phones may have different looking graphical interfaces and features even though they are running the same OS.[4]

Android platform architecture

Android is more of a complete software stack for mobile devices than an OS. It is a combination of tools and technologies that are carefully optimized for mobile needs.

As shown in Figure 1 each layer in the architecture provides different services to the layer just above it.[5,6]

RESEARCH METHODOLOGY

Requirement elicitation and analysis

The system requirement elicitation focuses on the definition of the problem to be tackled by the system for the system to be said to have solved a problem. The technique used for eliciting the requirement in this project is a combination of observation, brainstorming, and research; it should be noted that other requirement elicitation exists, such as questionnaire, interview, and prototyping.

Functional requirement

As shown in Figure 2, the location module discusses about the location of the device as shown below:

- Last known location
- The system will have a functionality that returns the last location saved by the device before it went offline.
- This will enable the device to always return a location when requested, even if the location is not "fresh," i.e., current.
- Current location
- Functionality of the system that returns the current location of the device, this is only possible if the is online.
- The current location can be from a GPS satellite or a mobile base station.
- The location is then updated at time-interval to keep it fresh.
- Check location accuracy
- This system function checks the level of accuracy of a new location with respect to the old location, if the old location is more accurate, the new one is discarded

As shown in Figure 3, the short messaging service (SMS) module does the following things

- Listen for incoming SMS
- The system function to listen to all SMS received by the mobile device.
- Check SMS for command
- The system function that parse every received message for application-specific commands. This functionality is offered to enable user send command using any mobile phone's SMS functionality.
- Respond to command through SMS.

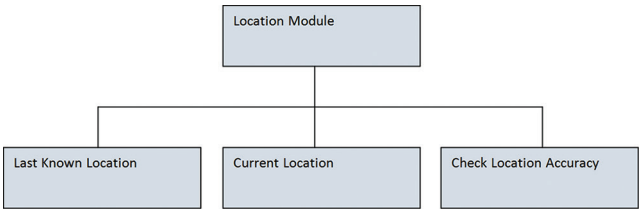

Figure 2: Location module

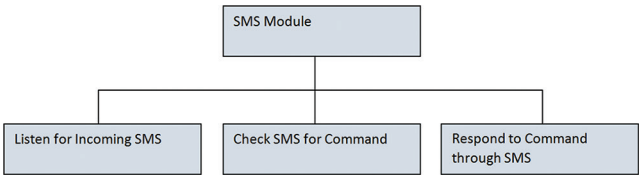

Figure 3: Short messaging service module

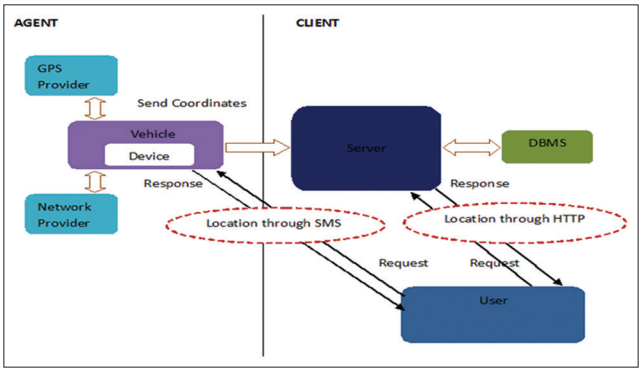

Figure 4: System architecture diagram

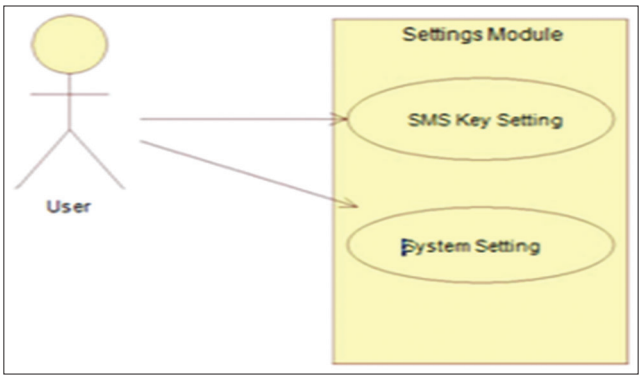

Figure 5: Use case for the settings module

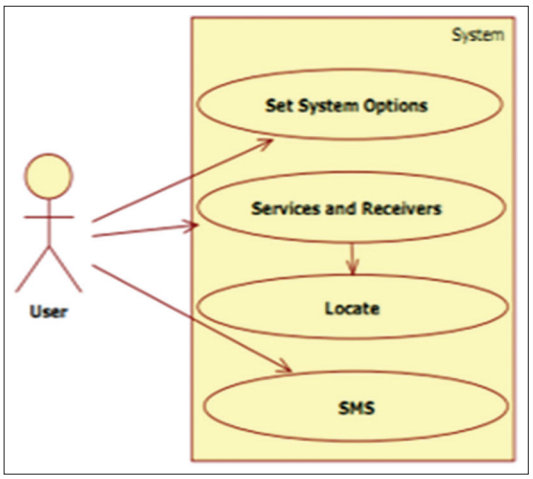

Figure 6: Use case diagram for services and receivers

System architecture diagram

As shown in Figure 4, the system architecture design discusses about the device will be communicating about with network user and the server.

System design

The design phase of a software system to construct a solution to the problem as defined in the requirement elicitation process. Design phase establishes an overall architecture by partitioning the software into components, the relationships and dependencies between such components are then established.

Use case diagram for the proposed tracking software

Overview use case diagram

The use case diagram shown in Figure 5 talks about the kind of setting users will be able to operate with it means how they can use the setting for the SMS and the system itself.

Figure 6 presented diagram number 6 is a use case diagram of services and the receivers. The diagram talks about how the user will be communicating with the mentioned functions of the systems.

Table 1 presented below discusses about the features of the SMS Key settings functions.

Use case for the services and receivers module

The above diagram presented in Figure 7 talks about the services and the receivers once again but this time how the functions will be communicating with the Android OS.

Table 2 presented below is showing the use case description of the system service. As mentioned it discusses about the actions taken by the actor and the response given by the system.

Use case for the location module

Figure 8 is an elaborated view of the combination of actions and the responses of the short messaging service module and the services and receiver module, respectively.

Use case description

Table 3 elaborates the actions of the actors and the response from the system while creating the location.

Table 4 discusses the actions of the actors and the response from the system while updating the location of the device.

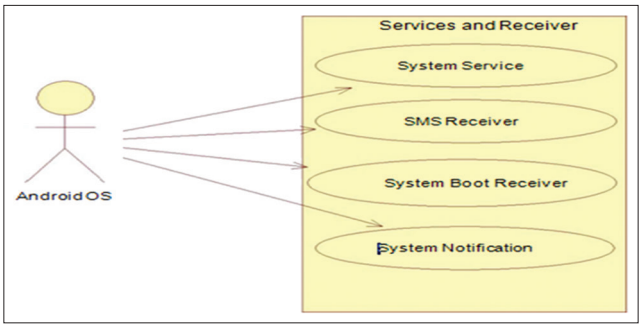

Figure 7: Use case diagram for services and receivers

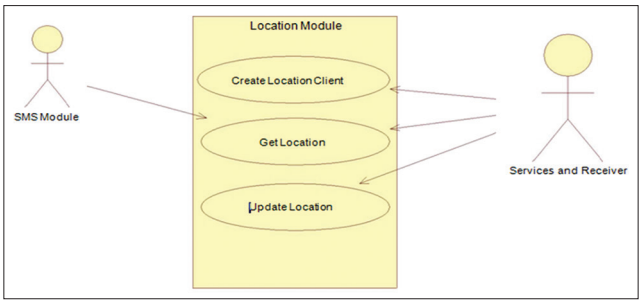

Figure 8: Use case diagram for location module

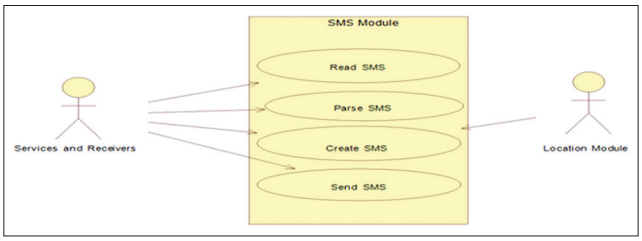

Figure 9: Use case diagram for short messaging service module

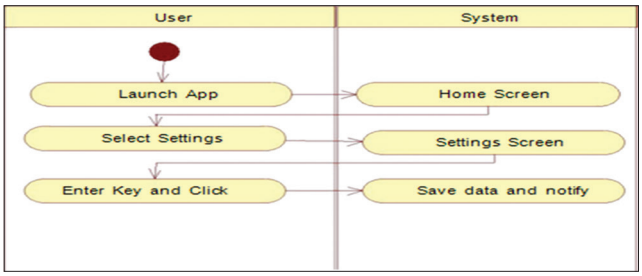

Figure 10: Activity diagram for settings module

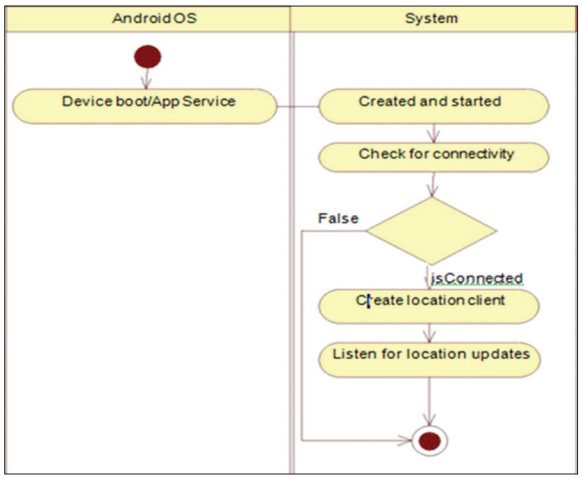

Figure 11: Activity diagram for system Service

Table 1: Use case description

Use case name: SMS key settings	
Brief description: This use case allows the user to enter a secret key which will be used at the beginning of a SMS command to the app	
Actor: User	
Main flow:	
Actor action	**System response**
Launch application	Display home screen
Select settings	Displays settings screen
Enter secret key into system and click "Save"	System save data and display notification

SMS: Short messaging service

Table 2: Use case description

Use case name: System service	
Brief description: This use case is described an android service which always runs in the system background. This service is used to create location, retrieve location, and send SMS to the user as a response to a command	
Actor: Android system (device boot and application launch)	
Main flow:	
Actor action	**System response**
Android device boot/application launch	Service is created and started
	Service checks for network connectivity
	Network connectivity is available
	Service checks for Google Play Service on device
	Service creates location client and get last known location
	Service listens for location updates
Alternative flow:	
	A-3 step 4: Network connectivity is not available Service stops and listen for network change
	A-5 step 6: Google Play Service not installed on device Services use the alternative
Use case name: SMS receiver	
Brief description: This use case allows the system to listen to incoming SMS and to be able to read the SMS and the sender ID	
Actor: Android system (SMS broadcast)	
Main flow:	
Actor action	**System response**
	System registers for SMS broadcast receiver
	Receiver class get called once SMS is received by the device
	Receiver class checks if SMS contains secret key as the first word
	SMS contains secret key and service is called to initiated SMS module
Brief description: This use case allows the system to start the service which creates location client immediately the device boots	
Pre-condition: App must be installed on the device and not the SD card	
Actor: Android system (on system boot)	
Main flow:	
Actor action	**System response**
	System registers for on boot broadcast receiver
	Receiver class get called once device boots
	Receiver class starts the service

SMS: Short messaging service

Use case for the SMS module

Figure 9 presented a use case diagram of the function of the SMS module. As shown in the diagram, it shows the pattern of communication of the between location modules and services and receivers module.

Table 5 presented the elaborate view of the read SMS function. It talks about the actions of the actor and the response of the system.

Table 6 presented the elaborate view of the parse SMS function. It talks about the actions of the

Table 3: Create location client

Use case name: Create location client	
Brief description: This use case creates an object which allows the system to get location	
Pre-condition: Device must have Google play service installed	
Actor: System service module	
Main flow:	
Actor action	**System response**
Start app service	Instantiate location request object
	Create location request object
	Set request interval and priority
Use case name: Get location	
Brief description: This use case fetches the last known location of the device	
Pre-condition: Location client has been successfully created	
Actor: System service module	
Main flow:	
Actor action	**System response**
Start app service	Get location object reference
	Get last known location of device

SMS: Short messaging service

Table 4: Update location

Use case name: Update location	
Brief description: This use case updates the location at specified intervals or distance covered	
Pre-condition: Location client has been successfully created	
Actor: System service module	
Main flow:	
Actor action	**System response**
Start app service	Check if location update is requested
	Update location client at intervals

SMS: Short messaging service

Table 5: Read SMS

Use case name: Read SMS	
Brief description: This use case reads the SMS retrieved by the SMS receiver use case	
Pre-condition: A SMS which begins with secret key is retrieved by the SMS receiver use case	
Actor: System service module	
Main flow:	
Actor action	**System response**
Start app service	If SMS begins with secret key
	Read the received SMS into variable

SMS: Short messaging service

actor and the response of the system.

Table 7 presented the elaborate view of the create SMS function. It talks about the actions of the actor and the response of the system.

Activity diagram for the proposed tracking software

The activity diagram is a unified modeling language (UML) diagram which describes the workflow of a system. Activity diagrams are dynamic in nature as it the sequence of activities performed in a system and the expected response. Below are the

Table 6: Parse SMS

Use case name: Parse SMS	
Brief descriTption: This use case parses the SMS read into word groups to recognize commands	
Pre-condition: SMS must have been read	
Actor: System service module	
Main flow:	
Actor action	**System response**
Start app service	Parse read SMS into groups
	Recognize and compare each group with system command to identify the command in the SMS

SMS: Short messaging service

Table 7: Create SMS

Use case name: Create SMS	
Brief description: This use case creates or composes SMS as response to user's command sent through SMS	
Pre-condition: SMS must have been parsed	
Actor: System service module	
Main flow:	
Actor action	**System response**
Start app service	Gather necessary data to be transmitted as SMS
	Retrieve SMS address from the retrieved SMS
	Format SMS into appropriate form
Use case name: Send SMS	
Brief description: This use case sends SMS to specified address	
Pre-condition: SMS must have been read	
Actor: System service module	
Main flow:	
Actor action	**System response**
Start app service	Retrieve composed SMS
	Retrieve address to send SMS
	Send SMS to retrieved address

SMS: Short messaging service

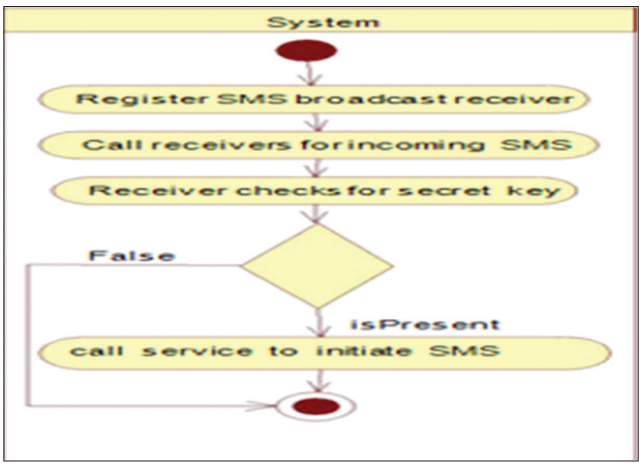

Figure 12: Activity diagram for short messaging service Receiver

activity diagrams of the proposed system, which is a breakdown of the use case diagram.

Activity diagram for the settings module

SMS key settings

In Figure 10, activity diagram of the SMS, key setting function has been shown which is showing how and what will be the action of the user and the response of the system.

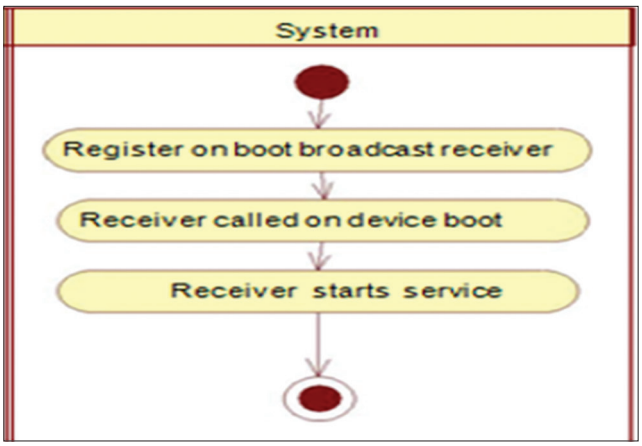

Figure 13: Activity diagram for bootloader

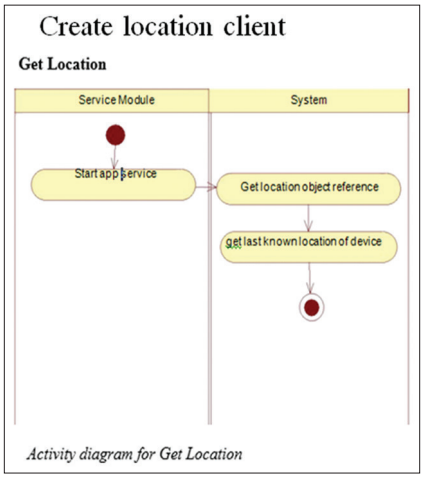

Figure 14: Activity diagram for create location client

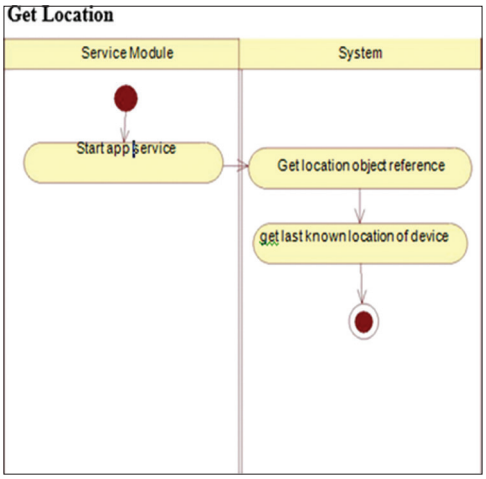

Figure 15: Activity diagram forget location client

Activity diagram for the services and receivers module System service

Figures 11 and 12 are the activity diagrams of the system and the SMS receiver module. In Figure 11, the action of the OS and the response of the system has been depicted whereas in Figure 12a using a flowchart SMS receiving option has been discussed.

System boot receiver

Figure 13 as presented the activity diagram for the system boot receiver which registers on boot broadcast and finally starts receiving services.

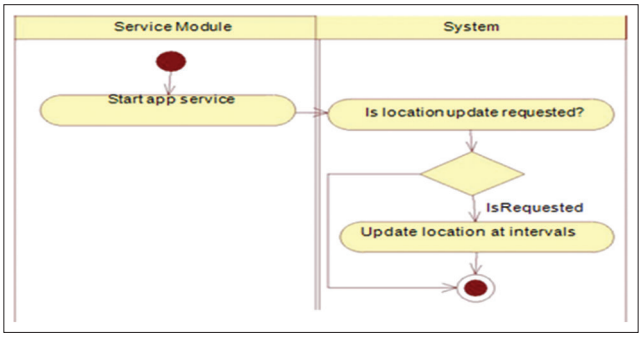

Figure 16: Activity diagram for update Location

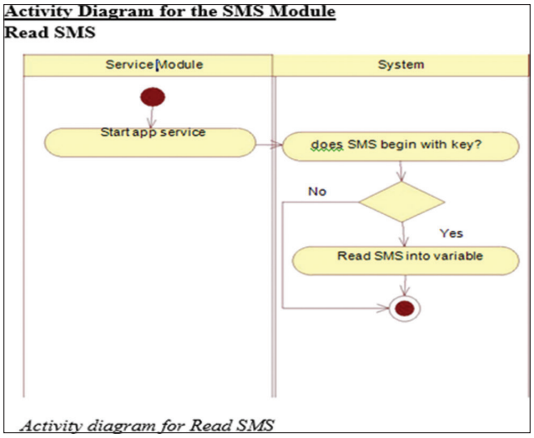

Figure 17: Activity diagram for read short messaging service

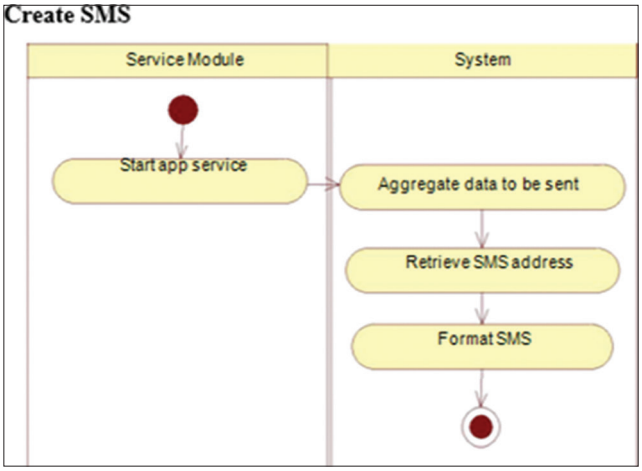

Figure 18: Activity diagram create short messaging service

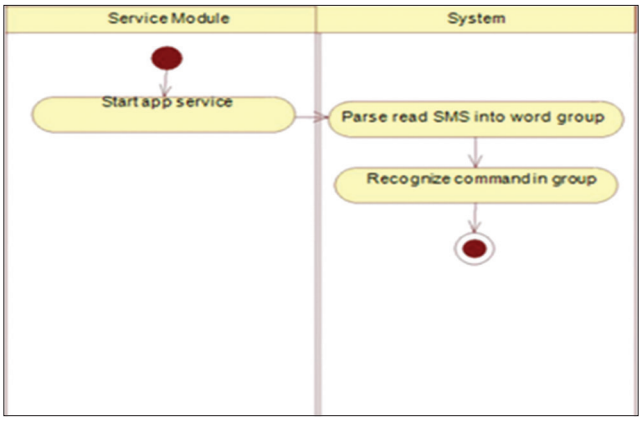

Figure 19: Activity diagram to parse short messaging service

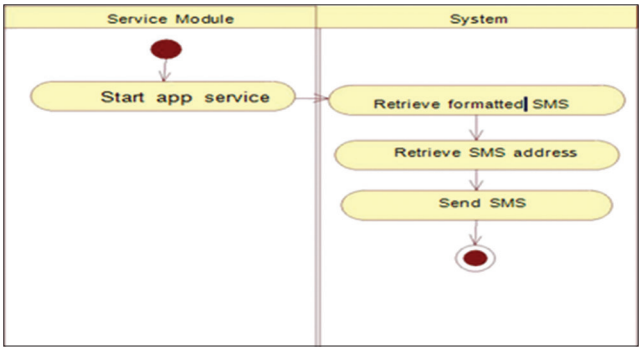

Figure 20: Activity diagram for send short messaging service

Activity diagram for the location module

Create location client.

Update location

Figures 14-16 presented the location module activity diagram. Figures 14-16 are showing the activities of creating location, get location, and finally update location activities.

SMS module activity diagrams

Figures 17-20 are the activity diagrams for the SMS module. Figure 17-20 are showing the activities performed by the read SMS, create SMS, parse SMS, and send SMS, respectively.

CONCLUSION

In this research work, a set out to develop a cost-effective solution to tackle the recurrent problem of carjacking and its attendant apprehension by utilizing Android mobile device for tracking purpose. The project highlighted the challenge of owning a car in a developing country such as Nigeria and the uneasy feeling the theft of such valuable possession usually has on the owner. This project, therefore, using information system standard of software development life cycle (SDLC), Android Station, Java and N)-SQL was used to develop this project.

This project, therefore, using information system standard of software development life cycle (SDLC) and object-oriented analysis and design, set out to develop a simple system which when installed on a mobile device, and possibly concealed in a vehicle, could report the location of such device and invariably the vehicle. Following the SDLC stages, the UML diagram; use case diagram, activity diagram and sequence diagram of each component and the entire system was designed.

Eclipse was the development environment of choice for the implementation stage of this project, as it provides a reach set of free plug-ins for the development of the Android application, the most important plug-in which is the Android developer toolkit (ADT) for Eclipse. The application developed was tested using one of the tools in the ADT called the Dalvik Debug Monitor Server.

REFERENCES

1. This Day Live (August, 2012): This Day Newspaper. Available from: http//www.thisdaylive.comarticlesin-lagos-600-cars-snatched-in-seven-months121453. [Last accessed on 2013 Aug 03].
2. Yao D, Yu C, Dey AK, Koehler C, Min G, Yang LT, et al. Energy efficient indoor tracking on smartphones. Future Gener Comput Syst 2014;39:44-54.
3. Location Based Services (LBS) by Margaret Rouse. Available from: http://www.searchnetworking.techtarget.com/definition/location-based-service-LBS. [Last retrieved on 2014 Oct 23].
4. Bahl P, Padmananabhan VN. RADAR: An in-Building RF-Based user Location and Tracking System. In: Proceedings 19th Annual Joint Conference of the IEEE Computer and Communication Society (INFOCOM '00), March 2000; 2000.
5. Application Package Interface (API): Webopedia. Available from: http://www.webopedia.com/TERM/A/API.html. [Last retrieved on 2014 Oct 25].
6. Kristian Y, Armanto H, Frans M. Utilizing GPS and SMS for tracking and security lock application on android based phone. Proc Soc Behav Sci 2012;57:299-305.