Parking Management System : A Case Study of Kampala-City

Kitutu K. Robert [1], Mumbere Vulere Fidel [2], Ingurat Emmy Simon [3]

[1]Department of Networks, College of Computing and Information Sciences, Makerere University
[2]Department of Data Engineering, International Business Science and Technology University (ISBAT)
[3]Department of Networks, College of Computing and Information Sciences, Makerere University

DOI: 10.29322/IJSRP.14.05.2024.p14936
10.29322/IJSRP.14.05.2023.p14936

Paper Received Date: 21st April 2024
Paper Acceptance Date: 24th May 2024
Paper Publication Date: 31st May 2024

***Abstract*—** *Urbanization is increasing at an alarming rate with 50% of the world population expected to be living in urban areas by 2025 and more transformation is expected in development countries[9]. This rapid urbanization in turn will affect sustainable urban traffic management. This huge increase in urban population has also resulted in rapid car ownership which causes traffic problems hence the need for proper traffic management solutions. Transport development is very crucial in cities for the movement of goods and people and there is therefore need for proper traffic management for the efficient running of the economy.*
Individuals and groups find themselves wasting a lot of time looking for where to park their cars around Kampala. In the process, they incur unnecessary expenses like fines for parking illegally along pavements; they park their cars in insecure places and sometimes find themselves in the hands of con-men. These con-men take advantage of their ignorance (especially visitors driving around the city for the first time) and end up overcharging them. The project will be an android application that will provide direction to various parking lots nearest to the car owners around Kampala, the location of the car parks and the safety/security provided as well as the nearest petrol stations to the car parks. This will save on the time and expenses incurred by individuals and groups in accessing car parking facilities as well as reduce on congestion in the city. This proposal will provide an effective and easy solution to Kampala Capital City Authority (KCCA) as far as collecting tax from the car owners and managing congestion within the city is concerned.

Index Terms- *Administrator, Application, Car park owner, GPS, GPS-Navigator, Stakeholder, User, Web-portal*

I. INTRODUCTION

Transport development is very crucial in cities for the movement of goods and people and there is therefore need for proper traffic management for the efficient running of the economy. Traffic management has a lot to do with the maneuvering of automobiles without interfering with each other and the other road users.
During the working hours, most car owners will be hooked up with work and hence there is need for the provision of more parking spaces to avoid a situation whereby parked cars will encroach in the road as this will interfere with the moving traffic. Urban traffic causes problems like congestion, air pollution and noise pollution. Traffic congestion is like a disease which if not treated will bring death to the heart of the city [10]. Land value in the city centers will fall because of the lack of parking facilities. City delay is associated with poor traffic management measures and people have a tendency of running away from such cities as more productive time is lost because of traffic congestion [10] and this also brings negative effects on people's emotional state.
Transport planning in developing countries is very crucial since there is an increase in road carnages, delays in movement and traffic congestions. [11] There is a direct link between the levels of development and sustainability with transport planning with developing countries heavily affected due to inappropriate infrastructure to cater for the huge increase in traffic.
The huge increase in car ownership also results in increase in traffic problems affecting both the developed and the developing countries as the traffic infrastructure is failing to cope up with the ever-increasing demand [12]. This proposal tackles the issue of traffic management in relation to rapid urbanization as these two are inseparable as they complement each other.

The process of accessing parking lots in Kampala has been a hectic, stressful and tiresome procedure. This is due to the lack of an organized setting dedicated to offering the service to vehicle owners. Over the years, there has been a significant increase in social amenities, recreation places, business ventures and tourist attraction centers. All these sectors have a great demand as far as car parking services and accessing parking lots are concerned.
However, these facilities are very scarce and hard to access. As a result, people get frustrated and end up wasting time in search of these parking facilities.

Traffic management cannot be separated from parking. Most cities are faced with the parking problems because of the huge increase in car ownership. The problems result from failure to allocate more parking spaces or the misuse of the parking facilities by motorists. Parking lots can be provided by business owners for customers and employees. Usually these parking lots are labeled to avoid scrambling for the parking space. Since the motorization in the 1930s, parking has been a problem for urban planners, transport engineers and politicians [13]. Some city councils tried to solve the problem of parking by introducing parking discs, but however this actually increased the problems. Parking space need to be provided close to the motorist's place of work to avoid walking a longer distance which will prompt driving hence the increase in traffic in the Central Business District (CBD). Private parking where some parking space is owned by individuals or companies is used in some countries. This has been attributed to destroying the living space of cities since there will be huge demand of such space and some parking lots will be underutilized. Parking can be managed in terms of organizing the origins and destinations of the trips so that even movement will be programmed in terms of routes to be followed.

Kampala Capital City Authority (KCCA) plans to introduce effective and efficient control measures to regulate parking lot management within the city by streamlining motor vehicle parking and traffic. This has currently been achieved within the center of the city where parking controls are currently operational [14].
In a bid to improve the parking management program, KCCA appoints around 200 Inspectors who patrol the sectors where the system is implemented. It is expected that the number of inspectors shall reach around 550 during the year 2014 [14]. This is evidence that KCCA is striving to manage parking and that the proposed system is still lacking.
Currently, on street parking is at a premium in Kampala. It should be removed since it occupies a large road capacity which would be utilized by pedestrians and moving traffic. Street parking should be replaced by spaces in multi-storey facilities which occupy less area per parking space [15]. KCCA makes sure that its Parking Inspectors (PIs) are always available at all parking sectors where the Parking Management Program is active. The main objective of the inspectors is to give advice and consultancy to parking users to insure proper and legal usage of parking spaces. Locating the parking spaces is a big challenge to vehicle owners. This shows that KCCA is spending a lot of money to pay all this parking inspectors.
In case parking users fail to follow the guidelines of parking usage, appropriate measures are taken by its parking inspectors. They include issuing Parking Violation Tickets and arranging for car towing is required.
It is however a very challenging task for drivers to access the parking areas. This is because of the lack of knowledge of the designated areas for parking. As a result, vehicle owners resort to parking on pavements and along the roads. This causes accidents as pedestrians are forced to get off the pavements and walk on the road [16]**.** Parking on the pavements causes congestion in the city. This congestion attracts wrong doers like thieves and blocks traffic flow.

This research paper discusses the design and development of a mobile parking lot management system that will locate and manage parking lots in the city of Kampala, maintain a listing of the parking lots within the city, and determine if the parking lot is full or not. This system will also notify the drivers of the available parking spaces in different parking lots.

II. LITERATURE REVIEW

This section provides an analysis of existing literature on this subject with the objective of revealing contributions, weaknesses and gaps. This section also reviews the existing mobile applications that have already been developed and how the Car Park Android Application is different from them. It also provides a clear picture as to why the proposed Car Park Android Application is necessary: how it helps in the management of traffic congestion, parking lots and how it helps private car owners/ drivers to locate and access parking lots.

A. Car Parking in Kampala

In the city of Kampala, we have many car parks. These include large, medium and small sized car parks. Most car parks provide a wide range of services and charging models to fulfill customers' needs. The most common services are monthly, daily and hourly parking reservation service. Monthly and long term parking services are provided via prior registration. Daily and hourly services are provided in a drop-in/walk-in and first come first serve (FCFS) manner. An example is Watoto Church parking lot located along Kampala Road that provides the mentioned services. Currently there is neither car web-based park searching systems existing nor online car park reservation system. The following are some related topics that have been studied, including related technologies and related applications.

B. Review of other existing systems

The mobile application market has several applications that are designed around the use of cars and the issue of parking inside a city. As cities become more and more crowded and the number of cars used is continuously increasing, the issue of car parking has become a daily concern for people that use a car. These types of applications are designed to help drivers park their car inside cities and are usually developed to work inside certain countries and cities.

C. Car Park Finder UK

Car Park Finder UK is a free Android application created from Paul Clarke. This application is designed to help the user find nearby car parks in the counties of England Wales and Scotland [3].
It works by loading the GPS position of the user and then displays a list with the car parks within a certain perimeter. The user can also change the view to see the Car Parks on the Map (Figure 1.0).

Figure 1.0: Car Park Finder UK [3]

Figure 1.1: alternative search [3]

Car Park aims to help the user navigate and find car parks in an unknown city, town or village. It is designed for people on day trips and holidays, or business people who have meetings in locations they aren't familiar with [3]. The map is interactive and directions are also available to help the user find the chosen car park.
An alternative search is/can be performed by entering a place or a postcode to find car parks instead of using the user's current location (Figure 1.1).

D. Collaborative Car Park Applications

In the case of a car park application, being a collaborative application means that the users have access to the car park database synchronously and that they can collaboratively contribute in different ways to that database (E.g. by creating or removing car parks).
AA parking [7] is an android application created by the Automobile Association [5] that is using the parking database from Parkopedia [6]. Parkopedia is a parking information provider worldwide that provides information on over 25 million parking spaces in 28 countries. Parkopedia is provides a collaborative Web-based application that allows users to search for parking anywhere in the world and contribute to the database by adding a new space. It relies on the individual user to provide accurate data and help populate the database with more parking places. Finally, the user is provided with the ability to book a parking space in advance if the online service is available on that particular car park.

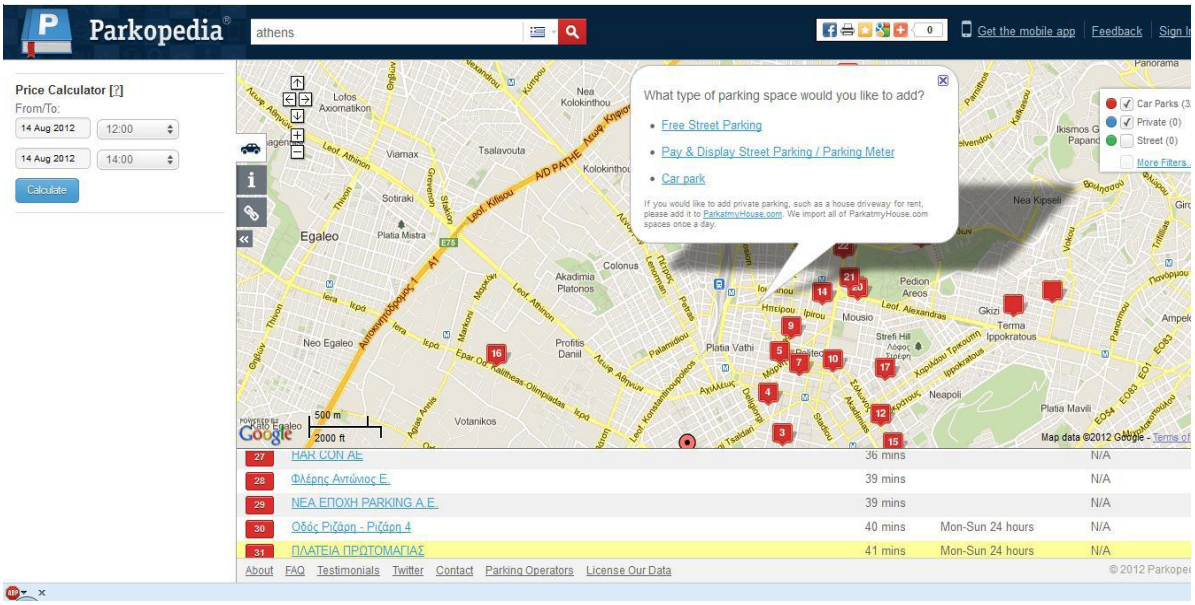


Figure 1.2 Parkopedia

The database of Parkopedia is constantly being populated with new car parks all around the world in a surprisingly high rate.
Parkdroid [4] for example, except from displaying the nearby parking spaces, it has a feature where the user can save the car location he parked his car in order to be able to find it later. It can be also used as a car locator that provides the user with walking directions to the place the car is parked. Finally, it has a feature of setting a parking meter alarm in order to act as a reminder. Another parking application that is a little different is BestParking [2]. This application is a tool that enables the user to find car parks and compare prices. It supports 43 cities and 79 airports in the U.S.A [2]. It is used as a search engine for parking rates and it frequently conducts manual updates to ensure accuracy.

III. METHODOLOGY

This section comprises of the main methods, tools and techniques used to develop the system and its descriptions. The methodology provided is to be followed so as to provide a solid realistic system structure. Methodology includes tools, instruments, approaches, processes and techniques that to be used in data collection, analysis, design, logical flow, implementation, testing and validation of the system.

A. Feasibility Study

This section discusses methods and procedures of collecting data as a necessary requirement for system development so as to have a rooted understanding of the users information need and identify user and system requirements henceforth. The research paper considers interviews as the research methods.
Data that is collected includes the following:

(i) the challenges faced by Kampala Capital City Authority (KCCA) in managing congestion of motor vehicles in the city of Kampala
(ii) the challenges faced by parking lot owners in managing the car parking lots and
(iii) the challenges faced by drivers (car owners) in accessing the parking lots, locating the parking lots and identifying petrol stations located nearby the parking lots.

B. Interviews

Interviews are used to collect the requirements. This is done by interviewing a number of car park owners to find out how they manage their car parks. KCCA officials are also interviewed to find out how they manage congestion and car parking within the city of Kampala. In addition drivers are also interviewed to find out how they access and locate the parking lots.
Interviews are used to get a more thorough investigation into the project scope and problem statement. This helps to identify the functional requirements and non-functional requirements from the stakeholders.
Interviews deepen the understanding of the problem and explain some of the statistical data obtained in the literature review section. Interviews present opinions from stakeholders on how the problem of car parking could be solved.
All this is achieved by structuring the questions that allow interviewees express how they think and feel about the problem and the presented solutions. Interview questions should be open ended.

C. Requirements Gathering

This section consists of four phases that contain methods, tools and techniques used to collect/gather user and system requirements. They help to achieve the objectives and determine what the system should achieve**.** The phases include interviews, requirements analysis, requirements modeling and requirements validation as elaborated below:

(i) Interviews

Interviews is the common and popular method used by the requirement engineers to elicit
System requirements and comprehend objectives of the system through verbal conversation with the stakeholders [18]. Interviews could be structured or closed (i.e., in the form of predefined questions), semi-structured (i.e., a blend of predefined and unplanned questions) and unstructured or open (i.e., an informal interview that does not involve predefined questions). The first two approaches largely aim towards acquiring quantitative data, whereas the later approach attributes to understand user expectations through open discussions with the stakeholders and acquire qualitative data [17]. In this paper, the later approach of interviews is used to collect qualitative data.

(ii) Requirements analysis

This involves technical staff working with users to find out about the application domain, the services that the system should provide and the system's operational constraints. The following elicitation techniques are used to analyze the requirements collected above;

❖ *Group Work:*

This technique is used to elicit the requirements of the system by inviting different stakeholders in a group meeting. This technique is effective to elicit requirements and resolving conflicts among the stakeholders by discussing all aspect of requirements with proper suggestions by the group members in a cooperative environment. However, it requires a lot of effort to conduct such meeting as it is always difficult to get hold of all the stakeholders at the same time

❖ *Brainstorming:*

Brainstorming is used to generate numerous ideas in a shorter time span regardless of focusing on a specific issue through informal discussions among the participant of different stakeholder groups. It is mostly used in innovative type
of projects where participants share their ideas on the basis of their experience and personal research about the project. The key disadvantage of brainstorming is that it cannot be effectively used to resolve major issues [17, 19].

(iii) *Requirements modeling*

Linear Requirements Engineering Process Model, envisaged by Linda Macaulay, bears lesser complexity and is primarily meant for administering projects. This model entails five activities in sequences which are: concept, problem analysis, feasibility and choice of options, analysis and modeling, and requirement documentation [17]. This model shall be used to attain final requirements and satisfy all stakeholders.

(iv) *Requirements validation*

This is concerned with demonstrating that the requirements define the system that the users really want. Requirements error costs are high so validation is very important i.e.
fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error. Therefore the system will be reviewed basing on the final requirements to ensure that it provides the functions which best support users' needs.

IV. SYSTEM DESIGN

A. Model-View-Controller (MVC)

Presented in 1978 [21], Model-View-Controller (MVC) is the oldest design pattern and has been successfully applied for many systems since its creation [22], [23]. The goal of this model is to separate business logic from presentation logic. The business logic modifications should not affect the presentation logic and vice versa [21].
MVC consists of three main components: Model, View and Controller. The Model represents a data to be displayed on the screen. More generally, Model is a Domain model that contains the business logic, data to be manipulated and data access objects. The View is a visual component on the screen, such as a button. The Controller handles events from user actions and communicates with the Model. The View and the Controller depend on the Model, but the Model is completely independent. The design pattern states that all Views should have a single Controller, but one Controller can be shared by several Views.

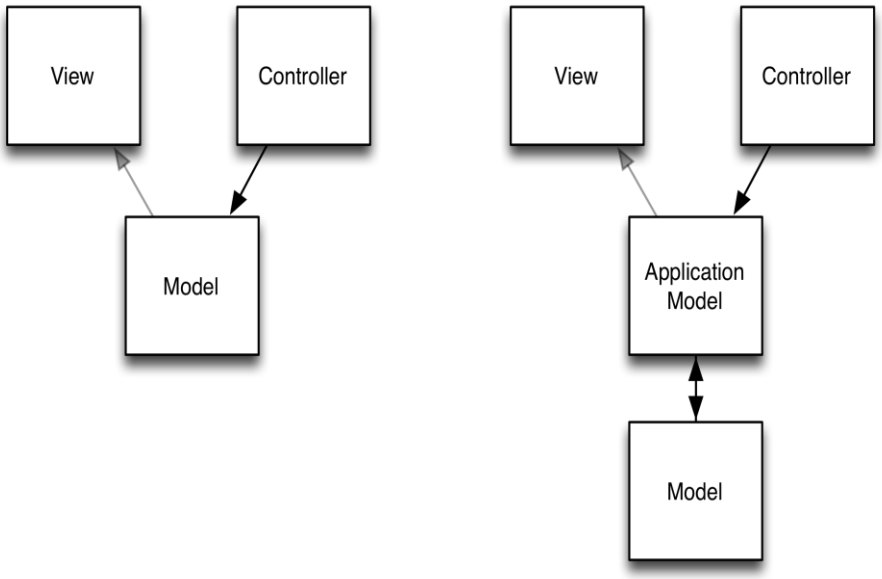


Figure 2. Classic MVC and Application Model MVC

In all variants, Controller handles events and communicates directly with a Model that is indicated by a black arrow. On the Classic MVC the Model processes data and notifies the View.
The View handles messages from the Model and updates the screen using the data received from the Model. This behavior is implemented using the Observer pattern (faint arrow in Figure 2). Conversely, the communication between the Model and the View in Passive Model MVC is done exclusively via the Controller. The Model notifies Controller which then notifies View and finally the View makes changes on the screen [24].
The Application Model MVC is an improved Classic MVC with an additional component. The Application Model component was added for the presentation logic (e.g. change the screen colour if the value is greater than 4) that was often added to View or Controller previously and makes a bridge between the Model and the View-Controller couples.

B. Android Domain Model

The Model of Android Passive MVC is a Domain Model containing business methods, web service call methods, database access objects, reusable methods and data model objects.
A Domain Model architecture should include components usually used in Android applications, such as Database manager, Web services manager and Business logic. Those components should be independent, as the architecture should be adaptable. Reusable components should be also separated. The basic model architecture is shown in Figure 5.

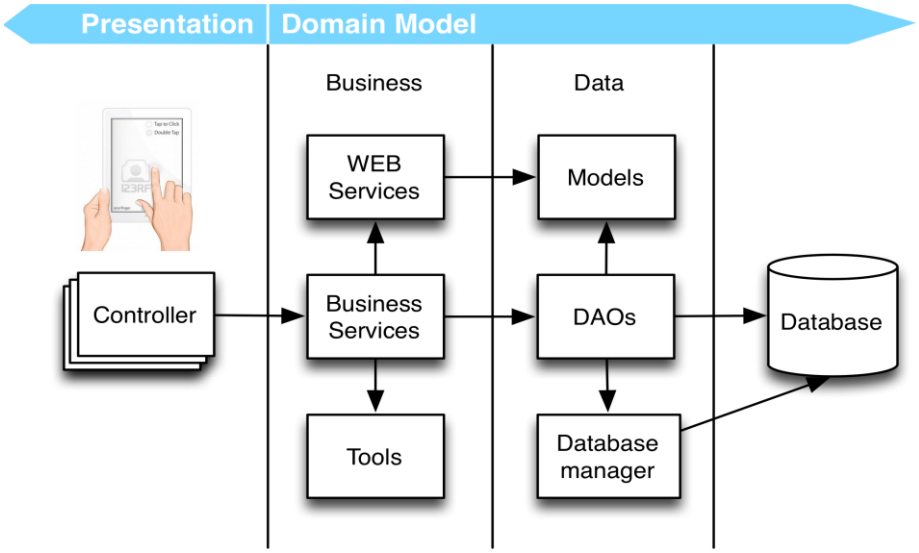


Figure 5. Domain Model Architecture

The architecture of Domain Model proposed in this document is inspired by 3-tier architecture that separates the presentation, the business and the data access layers [25].
The business layer of this model regroups objects and methods that use web services, business services and reusable tools. Business services contain business logic. If an application works via Internet as well as locally, all necessary verifications are done in Business services, which calls corresponding methods. The communication between a presentation and a domain model layer are made via Business services.
The data layer contains Models, Data Access Objects (DAO) and Database Manager. DAO and Model are the implementation of the Data Access Object pattern. Model contains data being persisted in the database or retrieved by web services calls. Model is a simple Plain Old Java Object (POJO) that contains only variables and their getter and setter methods. Data is manipulated and transferred through the application using those lightweight objects that are often called Data Transfer Object (DTO).
Persistence methods are organized in DAOs. DAO contains methods that enable the data in a database to be saved, deleted, updated and retrieved. Even if Android proposes an abstraction on the data access level with Content Provider, DAO simplifies the code of the application. The DAO design pattern creates a weak coupling between components and use a lightweight Model object instead of an Android cursor object in the application. DAO can also be used for the data stored in XML or text files. Good practice is to make DAO accessible via interfaces. It allows DAO modification (for example the change of SQLite to XML storage) without any change in Business services, which increases maintainability.

Database manager is in charge of the database creation. Database manager exists only if SQLite database is used by the application.
It stores the name of the database, and of its tables and methods to be able to create, drop, open and close the database.

This architecture regroups logically similar methods together, increases cohesion. High cohesion facilitates the maintainability of the software.
The final code of the application could be organized in packages by architectural components: Activities, Views, Controllers, Business Services, Tools, Web Services, Model, Data Access Objects and Database.
It gives the clear structure of an application and limits the package number. Additional packages could be created for interfaces, parsers (e.g. XML, JSON) and constants.

C. The Architecture of the System

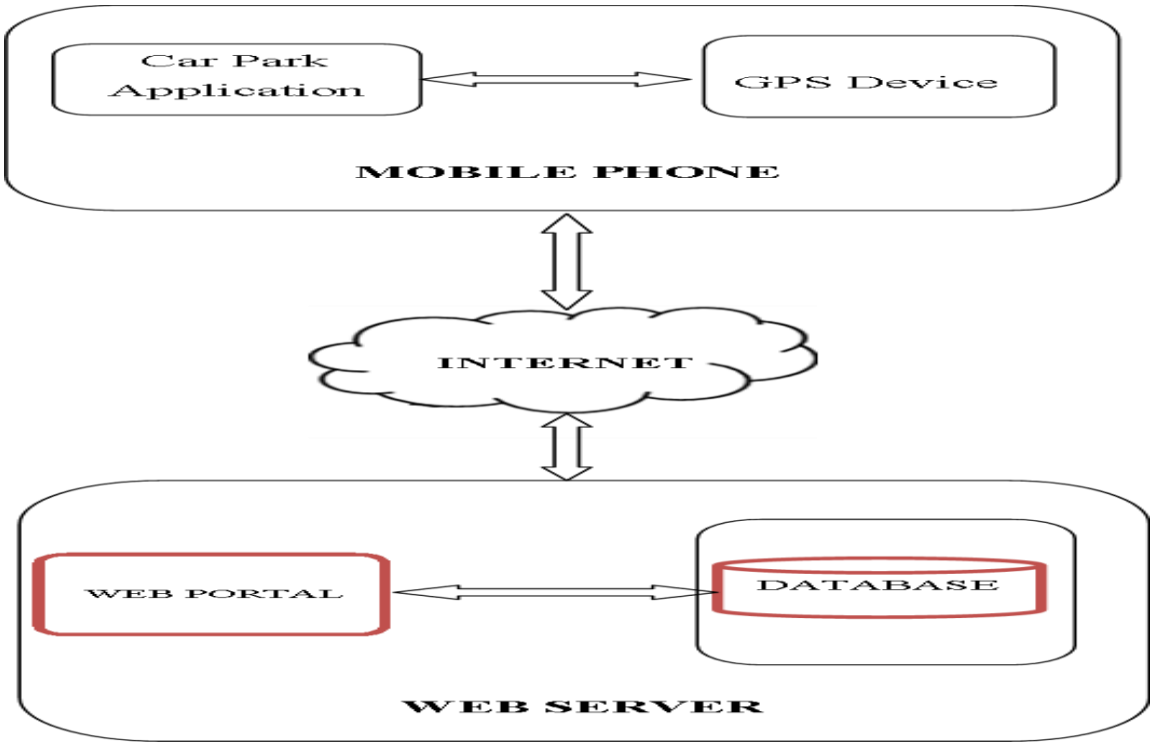


Figure 6- System architecture

This system will consist of two parts: one mobile application and one web portal. The mobile application will be used to find parking lots and view information about them while the web portal will be used for managing the information about the parking lots and the system as a whole.

The mobile application will need to communicate to a GPS application within the mobile phone, which in turn communicates with a physical GPS device to find the location of the user, see Figure 1. The GPS will provide the mobile application with locations of both the user and the parking lots and the distance between them, but it will also provide maps and the functionality to display the application's data on the map. The functionality provided by the GPS will be embedded into the application in order for the user to be able to use the functions in the application in a seamlessly manner.

Since this is a data-centric product it will need somewhere to store the data. For that, a database is used.

Both the mobile application and web portal will communicate with the database, however in slightly different ways.

The mobile application will only use the database to get data while the web portal will also add and modify data. All of the database communication will go over the Internet.

The mobile application has some restrictions about the resource allocation. To avoid problems with overloading the operating system the application will only be allowed to use 20 megabytes of memory while running the application. The maximum amount of hard drive space will be also 20 megabytes.

V.. IMPLEMENTATION

The development tools used to implement the system include a smart mobile phone, web portal and a personal computer.

The hardware and software operating environment in which our application has been developed and used are:

❖ **Software Requirements**

For developing the application the following are the Software Requirements:

- ✓ Operating System: Windows 8
- ✓ Language: Android SDK, Java

- ✓ Database: MySQL
- ✓ Tools: Eclipse IDE, Android Plug-in for Eclipse
- ✓ Technologies used: Java, MySQL, PHP, XML.
- ✓ Debugger: Android Dalvik Debug Monitor service
- ✓ Server: XAMPP Server

For running the application the following are the Software Requirements:

- ✓ Operating System: Android 2.3 or higher versions
- ✓ Network: Wi-Fi Internet or cellular Network

❖ **Hardware Requirements**

For developing the application the following are the Hardware Requirements:

- ✓ Processor: Pentium IV or higher
- ✓ RAM: 4GB
- ✓ Space on disk: minimum 1GB

For running the application:

- ✓ Device: Android version 2.3 and higher
- ✓ Minimum space to execute: 1GB

A. Functionality

Performance requirements, such as data through put, reliability, timing and user interface features have been considered.

User Interface Features such as screen resolution, display and colors. The key to success with any application is to get the UI and graphics right. This is a challenge because there are a lot of different resolutions, a lot of ways to present content and, naturally, a lot of different hardware.

With new high-end devices gaining popularity among users, the Android eco-system seems to be quickly headed towards high-resolution displays. High-resolution screens obviously make for a better user experience, but to take advantage of this, there is a need for us to update this application regularly and test for these UI features.

B. Limitations and safety

The system requires an active Internet connection. If there is no Internet connection, comments sent from users cannot be received, users registration details cannot be received or submitted into the database and also users cannot access the map view of the car parks.

When the system is in use, error messages will pop out to prevent incorrect input or use of the system.

The administrators shall enter/add all eligible car park locations. This is to ensure that only eligible car parks are entered into the system.

The administrators shall verify all car park managers. This is to ensure that only registered car park managers are entered into the system.

The administrators shall assign user levels/sessions to all users. This is to ensure that a user accesses the interface specified to them by the administrator therefore enhancing safety and security of information/data stored in the system.

VI.. RESULTS

This section describes and specifies the implemented system completely:

A. Design Input

Input 1: User Registration Details

Initially, the user has to register his details with the application for the first time. This is a one-time registration. The user has to enter details like first name, second name, username, password and Email. All this data is stored on the server.

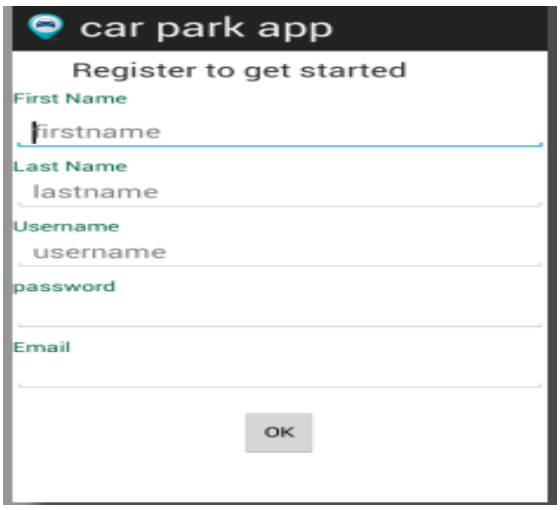


Figure1.1. User Registration

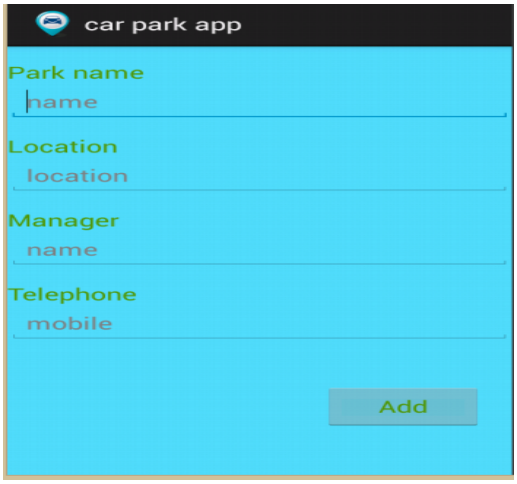
Login :

Once the user has registered, he/she can use his username and password to login in future. This will authenticate the user.

Figure1.2. User login

Input 2: Car Park Details

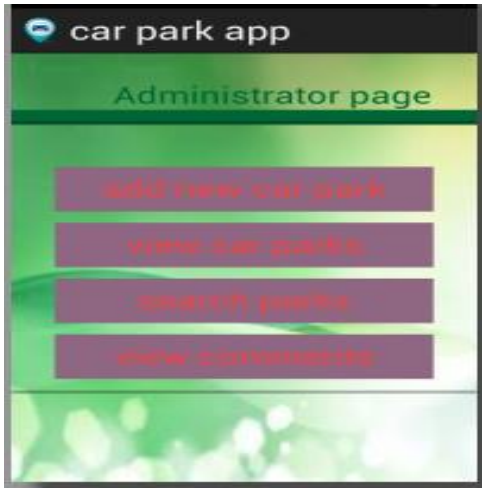
Only the administrator is allowed to add a new car park into the system. The car park details are received from the administrator's Add car park page. When they are submitted, the system stores the car park details in the database on the server. The car park details include Park name, location, Manager and telephone as shown below:



B. Design Output

The output that the computer will produce include the following:

Administrator page
Depending on the Login detail submitted at the time of Login, the different users are redirected to different pages. This is the administrator page that is displayed to the administrator upon login



VII.. CONCLUSION AND RECOMMENDATIONS

A. Conclusion

This research study is our second attempt in developing a mobile application as a group project which gave us a basic understanding of development and challenges of mobile application development. The main aim of the research study is to provide an easy to use application for searching and locating the car parks in the city of Kampala. The application has been implemented and tested on real devices.

B. Future Work

The application can be improved in many ways and can be extended to support more devices like the tablets and iOS devices. Following are some of the possible extensions:

- The application can be extended to provide a more advanced search by taking the zip code as the input.
- For a selected distance range, all the car parks in the distance range can be displayed irrespective of the city.
- There can also be ways to directly navigate the user to the car park's site or provide a way to call the car park manager when clicked on the phone number.

ACKNOWLEDGMENT

We give thanks and praise to the Almighty God for blessing us with His grace, strength and determination to bring this research project to a successful culmination.
Our thanks go out to all those who helped through their comments, feedback, edits or suggestions.
We would like to thank the team of dedicated lecturers and the whole establishment of the College of Computing and Information Sciences of Makerere University. The guidance and counsel they have always shared with us will carry a long way in our personal and academic research development.
We would like to expres our obligations to our project supervisor Dr. Ruth Mbabazi Mutebi (Senior Lecturer at the Department of Networks | College of Computing and Information Sciences)for her valuable inspiration and constructive suggestions that were so helpful to us throughout the research project.
Special thanks go to our beloved parents and families for the moral and financial support they have accorded us throughout this research study project.

REFERENCES

[1] Paving way for private parking space in Kampala. [Online]. Available: https://www.monitor.co.ug/artsculture/Reviews/Paving-way-for-private-parking-space-in-kampala-/-/-691232/1404864/-/fmocpl/-/index.html
[2] BestParking.[Online].Available: https://play.google.com/store/apps/details?id=com.bestparking&feature=related_apps
[3] Car Park Finder UK. [Online]. Available: https://play.google.com/store/apps
[4] Parkdroid.[Online].Available: https://play.google.com/store/apps/details?id=com.parkdroid
[5] Automobile Association. [Online]. Available: http://www.theaa.com/
[6] Parkopedia.[Online].Available: http://en.parkopedia.co.uk/
[7] AA Parking. [Online]. Available:https://play.google.com/store/apps
[8] "Android developers", Google Inc. Retrieved January 20, 2012. [Online]. Available: http://developer.android.com/index.html
[9] World Bank, (2003) *Urban Development: Facts and Figures.* [Online]. Available: www.Worldbank.org/urban/facts.html, worldbank.org/…/uk_traffic_manag_cracknell.pdf
[10] Brierley, J. (1962), *Parking of Motor Vehicles*, Applied Science Publishers, London, England.
[11] Willson R. (2001). *Assessing Communicative Rationality as a Transportation Planning Paradigm.* Transportation 28, pp 1-31
[12] Vuchic, V. R. (2005). *Transportation for Livable Cities*. New Brunswick, NJ: Center for Urban Policy Research.
[13] Newman, P and Jeffrey K, (1999), *Sustainability and Cities: Overcoming Automobile Dependence*, Island Press, and Washington DC, USA.
[14] KCCA Ministerial Policy Statement for FY 2012/2013, March 5 2012.
[15] Kampala City Car Parks project, unpublished.
[16] Uganda's Economy. [Online]. Available: www.ugandaeconomy.com/invest.../kampala-capital-city-authority-kcca
[17] Arif.S., Khan. Q. & Gahyyur. S.A.K., (2009-2010). Requirement Engineering Processes, Tools/Technologies, & Methodologies, International Journal of Reviews in Computing (IJRIC), ISSN: 2076-3328, Vol.2
[18] Gunda S. G., (2008). Requirements engineering: elicitation techniques. Trollhättan
[19] Zowghi. D., & Coulin. C., (2005). Requirements Elicitation: A Survey of Techniques, Approaches, and Tools, Engineering and Managing Software Requirements, Aurum, A & Wohlin, C (Eds.), Springer, USA

[20] Davey, B. & Cope, C., 2008. Requirements Elicitation – What's Missing? Issues in Informing Science and Information Technology, 5(1), p.53-57.

[21] G. Krasner and S. Pope, "A description of the model-view-controller user interface paradigm in the smalltalk-80 system," Journal of object oriented programming, vol. 1, pp. 26–49, 1988.

[22] P. Sauter, G. V¨ogler, G. Specht, and T. Flor, "A Model-View-Controller extension for pervasive multi-client user interfaces," Personal and Ubiquitous Computing, vol. 9, no. 2, pp. 100–107, Mar. 2005.

[23] M. Veit and S. Herrmann, "Model-view-controller and object teams: a perfect match of paradigms," in AOSD '03: Proceedings of the 2nd international conference on Aspect-oriented software development. ACM Press, Mar. 2003, pp. 140–149.

[24] S. Burbeck. (1997 Mar.) Applications Programming in Smalltalk-80TM: How to use Model-View-Controller MVC. [Online]. Available: http://st-www.cs.illinois.edu/users/smarch/st- docs/mvc.html [retrieved: March 2013]

[25] P. D. Sheriff, Fundamentals of N-Tier Architecture. PDSA Inc., May 2006.

AUTHORS

First Author – KITUTU K. ROBERT,
B.Sc. Software Engineering, Department of Networks, College of Computing and Information Sciences, MAKERERE UNIVERSITY.
E-mail: robertkitutu793@gmail.com

Second Author – MUMBERE VULERE FIDEL,
M.Sc. Information Technology, Department of Data Engineering, International Business Science and Technology University (ISBAT).
E-mail: mumfidel@gmail.com

Third Author – INGURAT EMMY SIMON, B.Sc. Software Engineering, Department of Networks, College of Computing and Information Sciences, MAKERERE UNIVERSITY .
E-mail address:eingurat504@gmail.com