ARDHI UNIVERSITY

LOCATION BASED SERVICES TO IMPROVE CUSTOMERS' NAVIGATION

A Case Study of Kariakoo Area

MOSHY, GABRIEL F

BSc Geoinformatics
Dissertation
Ardhi University, Dar es Salaam
July, 2023

LOCATION BASED SERVICES TO IMPROVE CUSTOMERS' NAVIGATION

A Case Study of Kariakoo Area

MOSHY, GABRIEL F

A dissertation submitted in the department of Geospatial Science and Technology in partial fulfillment of the requirements for the award of Bachelor of Science degree in Geoinformatics at Ardhi University.

CERTIFICATION AND COPYRIGHT

The undersigned hereby declare that, they have supervised and proof read the dissertation and recommend for acceptance by The Ardhi University a dissertation document entitled "**Location-Based Services to Improve Customers' Navigation**; **A Case Study of Kariakoo Ilala Municipality**" In fulfilment of the requirements for the Bachelor of Science degree in Geoinformatics.

……	……..
Dr. Anastazia D. Msusa	Mr. Iriael Mlay
(Supervisor)	(Supervisor)
Date ……	Date ……

DECLARATION AND COPYRIGHT

I**, Moshy, Gabriel F,** declare that the contents of this dissertation are the results of my own findings, obtained through studies and investigations. To the best of my knowledge, it has not been presented to any other university as a thesis for an award of Diploma, Degree or similar professional award.

……..…….

MOSHY, GABRIEL F

22866/T.2018

ACKNOWLEDGMENTS

First and foremost, I give all glory and honor to God, who has been my guiding light and source of strength throughout this journey. Without His grace, wisdom, and unwavering presence, I would not have been able to complete this dissertation.

I am deeply grateful to my supervisor, Dr. Anastazia D. Msusa and Mr. Mlay, for their unwavering support, invaluable guidance, and expert advice. Their commitment to academic excellence, patience, and encouragement have played a vital role in shaping the direction and quality of this research.

I would like to acknowledge the support and assistance provided by the staff and resources at Ardhi University. Their extensive collection of academic literature and research materials has been instrumental in broadening my understanding of the research topic and supporting the literature review.

I am indebted to my friends and family for their unwavering support, encouragement, and patience throughout this journey. Their belief in my abilities and their constant encouragement have provided the motivation and strength to overcome challenges and persevere during the demanding phases of this dissertation.

Finally, I would like to express my heartfelt appreciation to all the individuals who have supported me in ways both seen and unseen. Your words of encouragement, understanding, and motivation have been instrumental in keeping me focused and determined to complete this dissertation.

To everyone mentioned above and to those who have played a part in this dissertation, I extend my deepest gratitude. This work would not have been possible without your invaluable contributions, guidance, and support.

Thank you all sincerely.

DEDICATION

To my beloved parents, Francis Moshi and Hildagalda Francis, your boundless love, guidance, and sacrifices have been the cornerstone of my journey. You've nurtured my dreams with unwavering support, and I am forever grateful for the values you've instilled in me. To my dear brothers, Chrispian Moshi, Venance Moshi, Bernad Moshi, Piusi Moshi, and Gregory Moshi, you've been more than just siblings; you've been my companions, confidants, and pillars of strength. Our shared laughter and shared challenges have shaped me in profound ways.

To my esteemed guardian, Mr. Emmanuel Francis Mrema, your mentorship has been a guiding light in my journey. Your wisdom, patience, and belief in my potential have opened doors I never thought possible. Your role in my life goes beyond words; you are a true exemplar of dedication and encouragement.

This dedication is a tribute to the unbreakable bonds that tie us together. My achievements are a testament to the love and support I've received from my parents, brothers, and Mr. Emmanuel Francis Mrema. With profound gratitude, I offer this dedication to each of you, for you have enriched my life beyond measure.

ABSTRACT

Over the years, customers at Kariakoo have experienced difficulties while navigating to an appropriate location. This has been the case because there is no information about services (both spatial and non-spatial information) and there is a need for development of mobile applications that provide information about nearby services. The problem addressed in this study revolves around the challenges faced by customers like confusion, wastage of time and resources to the customers. On the other hand, poor search results like search engines and websites may not return the most relevant results to enable customers to attain services at a minimum cost. This study focused on leveraging location-based services (LBS) to enhance customer navigation through a dedicated NearBy services application. Therefore, there is a need to develop an actual value-added map product in a mobile application form for the customers when searching for services at Kariakoo.

The methodology encompassed identifying user requirements, development, debugging and deploying, and finally getting the expected Location-based mobile phone platform which allows customers to receive services and direct them geographically using their smartphones.

In conclusion, this study highlights the development of a location-based services application aimed at revolutionizing customer navigation through nearby service offerings. By harnessing cutting-edge geolocation technologies and personalized recommendations, the application has the potential to significantly improve users' navigation experiences while opening new avenues for businesses to engage with their target audiences.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

API	Application Programming Interface
GIS	Geographic Information System
GPS	Global Positioning System
HTTP	Hyper Text Transfer Protocol
IDE	Integrated Development Environment
JDK	Java Development Kit
JSON	JavaScript Object Notation
LbRS	Location-based Recommender System
LBS	Location Based Services
NB Services	NearBy Services
OSM	OpenStreetMap
POI	Point Of Interest
SDKs	Software Development Kits
UI	User Interface
URL	Uniform Resource Locator
XML	eXtensible Markup Language

CHAPTER ONE
INTRODUCTION

1.1 Overview

The research's topic will be introduced in this chapter. This chapter will cover the problem's historical context, its statement, its objectives and research questions, its significance and a description of the study area.

1.2 Background of the Study

A location-based service is a mobile computing application that is linked to the user's current location (LBS). Customers can get the most recent information about their neighborhood and are given the most recent developments. (Gupta & Rao, 2017.)

The location-based service (LBS) is altered by the user's current location and is retrieved from a spatial database that is remotely placed on the LBS server. (Rajalakshmi & Goyal, 2018).

The Location-based Recommender System (LbRS) is another recommender system where users can request recommendations for a particular location (Logesh et al., 2019).

Users can locate and assess their points of interest using LbRS when presented with a number of possibilities (POI). Additionally, it uses data mining techniques to generate recommendations based on historical user behaviors and characteristics. (Nazzawi et al., 2018).

With simply an internet connection and a web browser like google maps, you may view or explore different searching of places, but by using a Web-GIS, which is an interactive map and supporting database can carry out fundamental operations including zooming, panning, identifying, and measuring of distances. In-depth data analysis is also made possible by this, including the development of activity clusters or "hot spots" and the localization of a dataset's geographic center. These days, Web-GIS frequently includes of database and mapping servers that can collaborate to deliver consumers with incredibly scalable and effective maps and analysis. (Kerski & Baker, 2019).

However, typical GIS tools and features need to be reexamined and recreated in order to increase usability, particularly in terms of learnability, flexibility, and robustness (Kulawiak et al., 2019).

A Geographic Information System (GIS) is a system of computer hardware, software, data, and people that enables users to enter, manipulate, analyze, and present data and information about a specific area on the Earth's surface to support decision-making. (Obaidat et al., 2018).

In order to satisfy modern needs and handle the enormous amount of information that is transmitted from around the world, GIS offers various options to improve all types of services. Additionally, the use of smartphones with GPS capabilities is fast increasing, which accelerates the development of location-based services. (Kim et al., 2019)

Over the years, customers have experienced confusion when navigating the streets in the Kariakoo neighborhood in search of services. In addition, people frequently run across a number of challenges like confusion, wastage of time and resources when figuring out where they are exactly. Web-GIS is increasingly being used in this setting to share information (Ershad et al., 2020).

1.3 Statement of the Problem

Over the years, customers at Kariakoo have experienced difficulties while navigating to an appropriate location. This has been the case because there is no information about services (both spatial and non-spatial information) that lead to confusion, wastage of time and resources to the customers. On the other hand, poor search results like search engines and websites may not return the most relevant results to enable customers to attain services at a minimum cost. Therefore, there is a need to develop an actual value-added map product in a mobile application form for the customers when searching for services at Kariakoo.

1.4 Research Objectives

1.4.1 Main objective

To develop a location-based services to improve customers' navigation at Kariakoo.

1.4.2 Specific objectives

This main objective can be achieved by defining the following specific objectives:

i. To design and develop a GIS mobile application.
ii. To use a web-based GIS platform such as Open Layers to visualize the services available in the user's vicinity.
iii. To test the NB Services application.
iv. To deploy the NB Services application for Kariakoo area.

1.5 Research Questions

In order to achieve the above objectives, the following questions should be answered;

i. What software would be appropriate in designing and developing a GIS mobile application?
ii. How can web GIS on a mobile platform be used to provide better access to location-based services?
iii. How will be the NB Services application be tested?
iv. How will be the NB Services application be deployed?

1.6 Beneficiaries of the Study

i. Businesses: Businesses who are offering location based services can benefit from the study by gaining insights on their customers' usage patterns and preferences and improving their services accordingly.
ii. Developers: Developers can use the study as a reference for developing applications that use web GIS on mobile platforms.
iii. Government agencies: Government agencies can use the study to develop navigation and search applications that make use of web GIS and mobile platforms.
iv. Academic researchers: Academic researchers can use the study to gain insight into the usage of web GIS and mobile platforms and to further study the potential of location based services.

1.7 Limitation and Scope of the Research

NB Services application will run on android system only. It will require GPS embedded phones and internet connection when developing the application using Web GIS technology and when fetching the background map of Kariakoo area and also when. Also the application should provide users doing some queries and interacting with satellite images, OpenStreetMap and other geospatial data, as well as route and navigation capabilities only.

1.8 Significance of the Study

The study seeks to demonstrate the potential of Web GIS on a mobile phone device to provide a more intuitive and efficient way to access location-based services. The study will focus on the development of an Android-based application that allows users to easily locate and access services based on their current location. It will also evaluate the performance of such an application, as well as the user experience. The results of this study could be used to inform the design and implementation of future location-based services.

1.9 Description of Study Area

1.9.1 Geography

Study area will be Ilala city council comprise Kariakoo ward. This region lies between latitudes of 6°49’18” S to 6°48’54” S and Longitudes of 39°16’10” E to 39°16’46” E. Kariakoo is found at the elevation of 15.72 Meters above sea level. The selected region serves customers. The region also has the highest number of customer’s connections translating to highest revenue collection for the authority. Figure 1.1 illustrates the location of the study area.

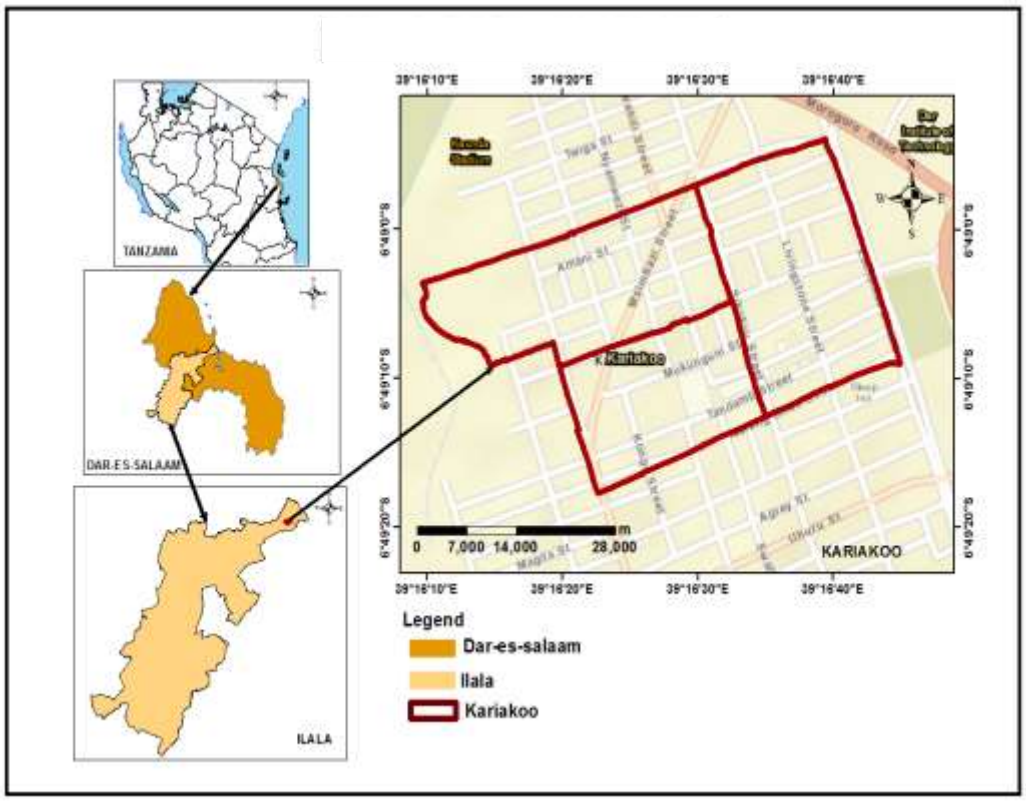


Figure 1.1: A map showing Kariakoo area

1.9.2 Demography

Dar es Salaam is the largest city in Tanzania and serves as its economic and administrative capital. Administratively the city is divided into five districts, namely: Ilala, Kinondoni, Ubungo, Kigamboni and Temeke. However, based on the 2012 census data, here is an overview of the demography of Dar es Salaam city in Tanzania.

Population:

According to the 2012 census, the population of Dar es Salaam was approximately 4.4 million people. However, please note that this figure is from 2012, and the population has likely increased significantly since then due to population growth and urbanization.

Ethnicity and Language:

Dar es Salaam is a diverse city with a mix of ethnic groups. The major ethnic groups in the city include the indigenous Bantu-speaking groups such as the Zaramo, Makonde, and Machinga. There is also a significant population of people from other ethnic backgrounds, including individuals of Arab, Indian, and European descent. Swahili is the most widely spoken language in the city, while English is commonly used for business and administrative purposes.

Economy:

Dar es Salaam serves as the economic center of Tanzania. It is home to various industries, including manufacturing, finance, telecommunications, and transportation. The city's port, the Port of Dar es Salaam, is one of the busiest in East Africa, playing a vital role in regional trade.

Migration:

Migration from rural areas to Dar es Salaam has been a significant factor in the city's population growth. Many people move to the city in search of employment opportunities and better living conditions. This influx of migrants has contributed to the city's cultural diversity.

1.9.3 Business services available in Kariakoo area

Kariakoo is a vibrant and bustling commercial area located in the heart of Dar es Salaam, Tanzania. It is known as one of the city's busiest and most popular trading hubs, offering a wide range of business services. Based on a nearby services application, here are some common business services you can find in the Kariakoo area:

Retail Stores:

Kariakoo is known for its vibrant market, which offers a variety of retail stores selling a wide range of products. You can find shops selling clothing, accessories, household items, electronics, and more.

Wholesale Markets:

Kariakoo is also a hub for wholesale trade, with several wholesale markets in the area. These markets cater to retailers and bulk buyers, offering a wide range of products at competitive prices.

Banks and Financial Institutions:

In the Kariakoo area, you can find branches of various banks and financial institutions. These establishments provide banking services, including account management, ATM services, currency exchange, and financial advice.

Mobile Phone and Electronics Services:

Given the popularity of mobile phones and electronics, it is common to find repair shops, service centers, and accessory stores in the Kariakoo area. These businesses offer phone repairs, software updates, accessories, and other related services.

Repair and Maintenance Services:

Numerous small businesses in Kariakoo specialize in repairing and maintaining various items such as car services, electronics, mobile phones, shoes, and garments. These services are often available at affordable prices.

Restaurants and Cafes:

The Kariakoo area is home to numerous restaurants and cafes offering a variety of cuisines. You can find local Tanzanian dishes, as well as international options. These eateries serve as meeting points for business discussions or a place to grab a quick bite during a busy day.

Business Consultancy:

Some consulting firms and business service providers are located in the Kariakoo area. These firms offer a range of services, including business advisory, market research, legal assistance, and accounting services, to support entrepreneurs and established businesses.

Hospitality and Accommodation:

Kariakoo is located near the city center and attracts tourists and business travelers. As a result, you can find hotels, guesthouses, and lodges that offer accommodation services in the area.

1.10 Dissertation Organization

This dissertation consists of five chapters, explaining in detail all the methods, principles, procedures and the results obtained in developing develop a location-based service using Web GIS on a mobile platform that enables users to locate and navigate to services they are looking for at Kariakoo.

Chapter 1 explains the background of the study, which gave rise to the problem. The objectives, research questions, significance and beneficiaries of the research, together with the description of the study area. Chapter 2 presents a brief literature review. It explains all literatures and studies done in relation to developing a location-based GIS system. Chapter 3 covers all the methods and techniques involved in the study. Chapter 4 provides the analysis and discussion of the results. Chapter 5 contains the conclusion and recommendations for the study.

CHAPTER TWO
LITERATURE REVIEW

2.1 Overview

This section will discuss many studies related to Advances in GIS, Trends in GIS, Mobile GIS, location-based systems, and location based recommender systems and their application. Also this chapter provides a review of literatures related to Web-GIS and Location based services. How other researchers think about the advantage of having navigation application system, will be discussed also in this chapter.

2.2 Advances in GIS

Geographic Information Systems (GIS) have been rapidly evolving over the past few decades, with new trends emerging in response to technological advancements and changes in user needs. Some of the current trends in GIS include the use of cloud computing, the integration of artificial intelligence and machine learning, the development of open-source GIS software, and the increased use of mobile devices for data collection and analysis. GIS has been advancing from Desktop GIS, Server GIS, Web GIS and Distributed GIS. (Shannon et al., 2021).

Advances in GIS technology have transformed the way nearby services are accessed and utilized. Over the years, there have been significant developments in various types of GIS, including desktop GIS, server GIS, web GIS, and distributed GIS. Let's explore these advances in detail:

i. Desktop GIS (Stand-alone GIS)

Desktop GIS refers to the traditional software installed on personal computers. Advancements in desktop GIS have led to improved functionality, enhanced processing power, and more intuitive user interfaces. For example, the integration of advanced 3D visualization capabilities has allowed users to explore nearby services in a more immersive manner (Esri, 2021). The introduction of sophisticated spatial analysis techniques, such as network analysis and geostatistics, has facilitated more comprehensive analyses of nearby services (Longley P. A., 2015). Additionally, the increased interoperability of desktop GIS software with other software and data formats has enabled users to incorporate diverse data sources for a more comprehensive understanding of nearby services (Kraak, et al., 2019).

ii. Server GIS (Client-server GIS)

Server GIS involves the use of centralized servers to store, manage, and distribute spatial data and services. Advances in server GIS have focused on scalability, performance, and accessibility. For instance, the adoption of cloud-based infrastructure has provided scalable storage and processing capabilities for spatial data, making nearby service data more readily available (Zhang et al., 2017). The introduction of web services, such as Web Map Services (WMS) and Web Feature Services (WFS), has allowed users to access and consume nearby service data in real-time (Cao et al., 2018). Geoprocessing services offered by server GIS platforms have reduced the computational load on client devices, improving overall performance for analyzing nearby services (Bishr & Kuhn, 2018).

iii. Web GIS

Web GIS involves the integration of GIS functionality into web-based applications, accessible through standard web browsers. Advances in web GIS have brought about transformative features. For example, interactive mapping capabilities have empowered users to visualize and interact with nearby service information through dynamic map interfaces (Rogerson, 2019). Mobile-friendly interfaces have made accessing nearby service data more convenient on smartphones and tablets (Huang et al., 2020). Collaborative mapping features have facilitated crowd-sourced data collection and updating of nearby service information, fostering community engagement (Goodchild & Li, 2012).

Web GIS, also known as Web Geographic Information System, is a framework that combines the power of geographical information systems (GIS) with web technologies to provide interactive and dynamic mapping applications accessible through web browsers. In the context of a nearby places application, Web GIS allows users to visualize and explore points of interest in close proximity to their location through an intuitive web interface.

iv. Distributed GIS

Distributed GIS involves the distribution of spatial data and processing across multiple devices or locations. Advances in distributed GIS have focused on improving data synchronization, scalability, and collaborative analysis. The utilization of peer-to-peer networks has enhanced data sharing and exchange between users for nearby services (Nebert, et al., 2018). Decentralized data processing techniques have improved processing efficiency and enabled more complex analyses for nearby services (Huang, et al., 2019). Real-time data

synchronization mechanisms have allowed distributed GIS systems to maintain consistency and accuracy across distributed datasets (Zhang et al., 2019).

2.3 Trends in GIS

The trends in GIS for nearby services applications include increased integration with mobile devices, enhanced real-time data processing, and the incorporation of augmented reality (AR) capabilities.

2.3.1 Integration with Mobile devices

According to a report by Research and Markets, the use of mobile GIS applications has become prevalent, allowing users to access location-based services on their smartphones or tablets (Research & Market., 2021). These applications leverage the GPS capabilities of mobile devices to provide users with information about nearby services such as restaurants, gas stations, and points of interest.

2.3.2 Real-Time Data Processing

Furthermore, advancements in real-time data processing have enabled GIS applications to provide up-to-date information on nearby services. With the increasing availability of data streams from various sources, such as social media, sensors, and public databases, GIS applications can now analyze and process this information in real time to offer accurate and timely insights into nearby services (Salas et al., 2019).

2.3.3 Incorporation of Augmented Reality

Another emerging trend in GIS for nearby services applications is the integration of augmented reality. AR technology enables users to overlay digital information, such as directions, ratings, and reviews, onto the real-world environment through their mobile devices or specialized AR glasses. This integration enhances the user experience by providing a more immersive and interactive way to interact with nearby services (Huang et al., 2020).

These trends in GIS for nearby services applications demonstrate the increasing sophistication and utility of geospatial technologies in providing personalized and context-aware information to users, improving their overall navigation and decision-making experiences.

2.4 Mobile GIS

Mobile GIS refers to the use of Geographic Information Systems (GIS) on mobile devices such as smartphones and tablets. It enables users to access, collect, analyze, and share geospatial data

in real-time, anywhere and anytime. According to Li and Li, mobile GIS has become an important tool in many fields, including emergency response, environmental monitoring, transportation, and urban planning, among others (Zhang, 2017).

Mobile GIS applications typically use location-based services and GPS technology to provide real-time location tracking and navigation services. Users can view and interact with various types of geospatial data, including maps, satellite imagery, and other location-based information (Li, 2018).

Mobile GIS also supports data collection through the use of mobile devices' sensors, such as cameras and microphones, and enables data sharing and collaboration through cloud-based storage and sharing services. Mobile GIS offers several advantages over traditional desktop GIS, including increased mobility, real-time data collection and analysis, and improved collaboration and communication among team members. It also provides a cost-effective solution for data collection and analysis, as it eliminates the need for expensive hardware and software (Zhang, 2017).

2.5 GIS for Nearby Services

A Geographic Information System (GIS) is a technology that captures, manages, analyzes, and presents geographic data in a visual and spatial context. In the case of a nearby places application, GIS can be utilized to store and process information about points of interest, such as restaurants, hotels, and other relevant locations, along with their spatial coordinates (Longley et al., 2015).

By leveraging GIS, developers can create applications that allow users to search for nearby places based on their current location or specified parameters. GIS provides the ability to perform spatial queries and calculations, enabling the identification of nearby points of interest within a certain radius or proximity.

2.5.1 Application of GIS within Nearby Services Application

The application of GIS within a nearby places application offers numerous benefits for spatial analysis and visualization of points of interest. By incorporating GIS, the application can leverage geospatial data to enhance location-based services and provide valuable insights to users (Longley et al., 2015).

With the use of GIS, the nearby places application can efficiently process and analyze spatial data, enabling functionalities such as proximity-based searches, routing, and spatial clustering. GIS allows for the calculation of distances between locations, identification of nearby points of interest within a specified radius, and generation of optimized routes for navigation.

Furthermore, GIS enables the visualization of nearby places on interactive maps, providing users with a visual representation of their surroundings. This visual context enhances the user experience by allowing users to explore nearby places spatially and make informed decisions based on their geographic relationships.

2.5.2 Users of Nearby Services Application

Users of a nearby services application in Kariakoo, Dar es Salaam City, who search for certain services are typically individuals or businesses residing or operating in the area. Kariakoo is a bustling commercial district known for its vibrant markets, shops, and diverse range of services. The nearby services app caters to the needs of these users by providing them with a convenient platform to search for specific services in their vicinity.

Local Residents:

Kariakoo is a densely populated area with a mix of residential and commercial properties. Local residents often use the app to find essential services like grocery stores, pharmacies, healthcare facilities, salons, and restaurants. They may also search for services such as home repair, cleaning, or transportation options.

Small Business Owners:

Kariakoo is home to numerous small businesses, including retailers, wholesalers, and service providers. These business owners rely on the nearby services application to connect with customers who are looking for their specific products or services. They may include clothing stores, electronics shops, stationary suppliers, tailors, mechanics, and more.

Tourists and Visitors:

Kariakoo attracts a significant number of tourists and visitors due to its vibrant markets and cultural heritage. Travelers in search of unique experiences, local cuisine, or souvenirs may utilize the nearby services application to discover restaurants, hotels, tour guides, art galleries, or handicraft stores in the area.

Students:

The presence of educational institutions, such as universities and colleges, draws a substantial student population to Kariakoo. Students often use the application to find nearby libraries, study

centers, bookstores, printing services, internet cafes, or places to eat and socialize.

Professionals and Office Workers:

Kariakoo is also a hub for various professionals and office workers who require services tailored to their needs. They may seek out business centers, co-working spaces, conference facilities, printing and copying services, courier services, or nearby restaurants for business meetings and lunches.

Service Seekers:

The application would also cater to individuals actively seeking specific services, such as car dealers and auto repairs, electricians, hairdressers, or healthcare providers. They would use the application to find and connect with nearby professionals who can fulfill their requirements.

Overall, users of the nearby services application in Kariakoo, Dar es Salaam City, come from diverse backgrounds and have specific requirements depending on their residential or business-related needs. The application serves as a valuable tool for connecting these users with the services they seek in the bustling commercial district of Kariakoo.

2.6 Location-Based System

A location-based system is a technology that utilizes geographical information to provide services or information based on the user's current location. These systems often employ GPS (Global Positioning System) or other positioning technologies to determine the user's coordinates and deliver relevant content or functionality.

According to a study by M. Hazas, et al. titled "Location-Based Systems: A Review and Future Research Directions," location-based systems have become increasingly popular due to the widespread adoption of smartphones (Hazas et al., 2013). These systems can offer a wide range of services, such as navigation, geotagging, personalized recommendations, location-based advertising, and social networking features.

2.7 Location-Based Recommender System

A location-based recommender system is a technology that utilizes geographical data and user preferences to provide personalized recommendations based on the user's current location. These systems combine location information, such as GPS coordinates, with user profiling and historical data to suggest relevant items or services in the vicinity. By considering the user's location as a contextual factor, location-based recommender systems can offer

recommendations tailored to the user's immediate needs and preferences. For example, mobile applications like Yelp or TripAdvisor employ location-based recommender systems to suggest nearby restaurants, attractions, or accommodations based on the user's current location (Zhang Z. , 2018).

2.8 Application Programming Interface (API)

An API, or Application Programming Interface, is a set of rules, protocols, and tools that governs how different software applications interact and communicate with each other. It defines the methods, data formats, and conventions that should be followed for seamless integration and data exchange between software components.

According to the Oxford Dictionary, an API is defined as "a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service" (Oxford Dictionary, n.d.). APIs provide a standardized interface for developers to interact with software systems, allowing them to access and utilize the functionalities, data, or services provided by those systems.

An API for a nearby places application, such as the Google Places API, is a programming interface that provides developers with the ability to access and retrieve information about points of interest in close proximity to a given location. This API enables developers to incorporate features into their applications that display nearby places such as restaurants, hotels, shopping centers, and other points of interest based on specific search criteria and user preferences.

2.8.1 Google Places API

The Google Places API, a service provided by Google, is particularly useful for developers creating nearby places applications. This API enables developers to access a wealth of information about various points of interest in close proximity to a specified location. By integrating the Google Places API into their applications, developers can provide users with features that display nearby places, including restaurants, hotels, shopping centers, and other relevant points of interest. This functionality enhances the user experience by helping them discover and explore their surroundings.

The API enables developers to create applications that leverage location-based data, making it easier for users to find and interact with places of interest. Developers can use the API to

implement features like displaying nearby restaurants, searching for specific types of businesses, retrieving photos and reviews for places, and even autocomplete suggestions for place names in search boxes.

By integrating the Google Places API into the applications, developers can enhance the user experience by providing accurate and up-to-date information about places worldwide. Whether it's building a travel application, a restaurant finder, or a mapping application, the Google Places API offers a comprehensive set of tools and data to create powerful location-based experiences.

2.9 Navigation System to be used in Kariakoo Area

2.9.1 Google Earth

Google Earth is a free web-based application that allows users to explore the world in 3D and satellite imagery. It was first released by Google in 2005 and has since become a popular tool for viewing and interacting with geospatial data. According to Han et al., Google Earth provides a powerful platform for visualization and analysis of complex geospatial data, as it enables users to easily navigate and interact with data in a realistic 3D environment (Han et al., 2017). Google Earth offers several features that allow users to view and analyze geospatial data, including the ability to search for specific locations, view satellite imagery, and add layers of data such as road networks, weather patterns, and demographic information. Users can also create and share custom maps and routes, and view historical imagery to analyze changes over time.

In addition to its use by individuals, Google Earth has also been adopted by a variety of organizations, including governments, non-profit organizations, and businesses. For example, it has been used in disaster response and emergency management efforts, as well as in urban planning and environmental monitoring.

2.9.2 Open Street Map

OpenStreetMap (OSM) is a free and open-source web-based mapping platform that allows users to create and edit maps using crowdsourced data. It was launched in 2004 and has since grown to become a widely used alternative to commercial mapping platforms. According to Haklay et al., OSM has become a valuable resource for a variety of applications, including humanitarian aid, disaster response, urban planning, and environmental monitoring (Haklay et al., 2010).

One of the key features of OSM is its user-generated content model, which allows anyone to contribute to the platform by adding, editing, or updating map data. This approach has resulted in a highly detailed and up-to-date map of the world, which can be accessed and used by anyone free of charge.

In addition to its use as a mapping platform, OSM has also been used in a variety of research applications. For example, researchers have used OSM data to analyze patterns of urbanization, to study transportation networks, and to monitor land-use changes over time.

Despite its many advantages, OSM does have some limitations, such as incomplete coverage in some regions and potential quality issues with the user-generated data. However, ongoing efforts by the OSM community to improve data quality and coverage are helping to address these issues.

2.10 Android Studio and Java Language

2.10.1 Android Studio

Android Studio is an integrated development environment (IDE) specifically designed for building Android applications. It was developed by Google and was first released in 2013. Android Studio is built on top of the popular IntelliJ IDEA IDE and provides a wide range of tools and features for developing, testing, and deploying Android applications. According to Google, Android Studio is the official IDE for Android application development (Google, 2021).

Android Studio uses Java as its primary programming language for Android app development. Java is used to write the source code for Android applications, which can then be compiled and run on Android devices. Java provides a wide range of features and tools for Android application development, including object-oriented programming, memory management, and multithreading.

2.10.2 Java Language

Java is a popular programming language used for developing a wide range of applications, including Android applications. Java was first released in 1995 and has since become one of the most widely used programming languages in the world. Java is known for its platform independence, which means that Java applications can run on any platform that supports the Java Virtual Machine (JVM).

One of the key advantages of using Java for Android application development is the large and active developer community. There are a wide range of resources available for Java developers, including libraries, frameworks, and online communities. This makes it easier for developers to find solutions to problems and to learn new techniques and best practices.

2.10.2 Hash Map

A hash map, also known as a hash table, is a data structure that enables efficient storage and retrieval of key-value pairs by using a hashing function. It employs an array to store the elements, where each element is assigned a unique index based on its key's hash code.

According to Cormen et al. (2009), a hash map provides constant-time average-case complexity for operations such as insertion, deletion, and retrieval, making it a highly efficient data structure. The hashing function plays a crucial role in this efficiency by mapping keys to unique indices, allowing direct access to the corresponding values in the array.

When inserting a key-value pair into a hash map, the key is first hashed to determine its index in the underlying array. If two keys hash to the same index, a collision occurs. Various collision resolution techniques, such as chaining or open addressing, can be used to handle collisions effectively.

One key advantage of a hash map is its ability to provide fast access to values based on their associated keys. Retrieval of a value from a hash map involves calculating the hash code of the key, mapping it to an index in the array, and directly accessing the corresponding element, resulting in constant-time complexity on average.

Additionally, hash maps are widely used in many applications and programming languages due to their versatility and efficiency in storing and retrieving data.

2.11 Operating Systems

An operating system (OS) is a software component that manages computer hardware and provides essential services for running applications. It acts as an intermediary between the user and the computer hardware, enabling the execution of tasks and ensuring proper resource allocation (Silberschatz et al., 2020).

Different operating systems exist for various computing devices, including personal computers, smartphones, servers, and embedded systems. Examples of widely used operating systems

include Microsoft Windows, iOS, and Android.

A mobile operating system is a software platform specifically designed to power mobile devices such as smartphones and tablets. It provides the necessary infrastructure and services for running applications, managing hardware resources, and facilitating communication and connectivity (Korhonen et al., 2019).

2.11.1 Android Operating Systems

One of the most widely recognized mobile operating systems is Android, developed by Google. Android offers a flexible and customizable platform that supports a broad range of devices from different manufacturers (Korhonen et al., 2019). It provides a rich set of features and APIs, enabling developers to create diverse applications for various purposes.

2.11.2 MAC Operating Systems

Another prominent mobile operating system is iOS, developed by Apple for its iPhone, iPad, and iPod Touch devices. iOS offers a seamless and intuitive user experience, tightly integrating with other Apple services and devices (Korhonen et al., 2019). It emphasizes security, privacy, and a curated application ecosystem.

2.11.3 Window Operating Systems

Windows 10 Mobile, developed by Microsoft, is another example of a mobile operating system, although it is no longer actively supported. It provided a unified experience across different devices, allowing for smooth integration with the Windows ecosystem (Korhonen et al., 2019). These mobile operating systems, among others, offer functionalities such as application management, multitasking, notifications, wireless connectivity, and access to a wide range of applications through application stores (Korhonen et al., 2019). They also include features like voice assistants, location services, and advanced security measures to enhance user experience and protect sensitive data.

2.12 NB Services Application

NB Services application is a mobile application designed to help users find various services and businesses in their immediate vicinity. The application leverages location-based services and utilizes a combination of mapping technology, GPS data, and user preferences to provide relevant information about nearby service providers.

The main purpose of a NB Services application is to assist users in quickly locating and

accessing the services they need, such as restaurants, hotels, cafes, retail stores, banks, pharmacies, healthcare facilities, gas stations, and more. The application typically offers a find functionality that allows users to search their desired service or business category and filters the results based on their location.

One of the key features of a nearby services application is its ability to display search results on a map interface. Users can view the locations of nearby businesses as markers on the map and easily identify their proximity to their current position. Additionally, the application often provides detailed information about each business, including contact details, opening hours, photos, and sometimes even menus or product catalogs.

2.12.1 Advantages of NB Services over Google Maps

One advantage of NB Services application over Google Maps is its ability to provide specific and targeted information about nearby services, catering to users' immediate needs. While Google Maps offers a wide range of information about businesses and services, a dedicated NB services application can provide more focused and relevant results based on user preferences and requirements.

For example, NB Services application can allow users to filter search results based on specific criteria such as distance, ratings, reviews, and availability. This level of customization helps users quickly find the most suitable service providers in their vicinity, saving them time and effort.

According to a study by Mooney et al. (2019), dedicated nearby services applications were found to have a higher level of accuracy and comprehensiveness in providing relevant service information compared to general mapping applications like Google Maps.

Furthermore, NB Services applications often have additional features and functionalities that enhance the user experience. These can include in-application bookings, real-time updates on service availability, user reviews and recommendations, and personalized notifications. Such features contribute to a more seamless and tailored experience for users seeking specific services.

In summary, NB Services application offers advantages over Google Maps by providing more focused and targeted information, higher accuracy in service listings, and additional features that enhance the overall user experience.

CHAPTER THREE
METHODOLOGY

In this chapter, methods implemented, techniques applied and data inputs used throughout this study are described. This chapter describes overall methods ranging from identifying user requirements, development, debugging and deploying, and finally getting the expected Location-based mobile phone platform which allows customers to receive services and direct them geographically using their smartphones. This study involved development of Location-based system that combines street address data and spatial data. The whole process from identifying user requirements, development, debugging and deploying of the system step by step is summarized in figure 3.1

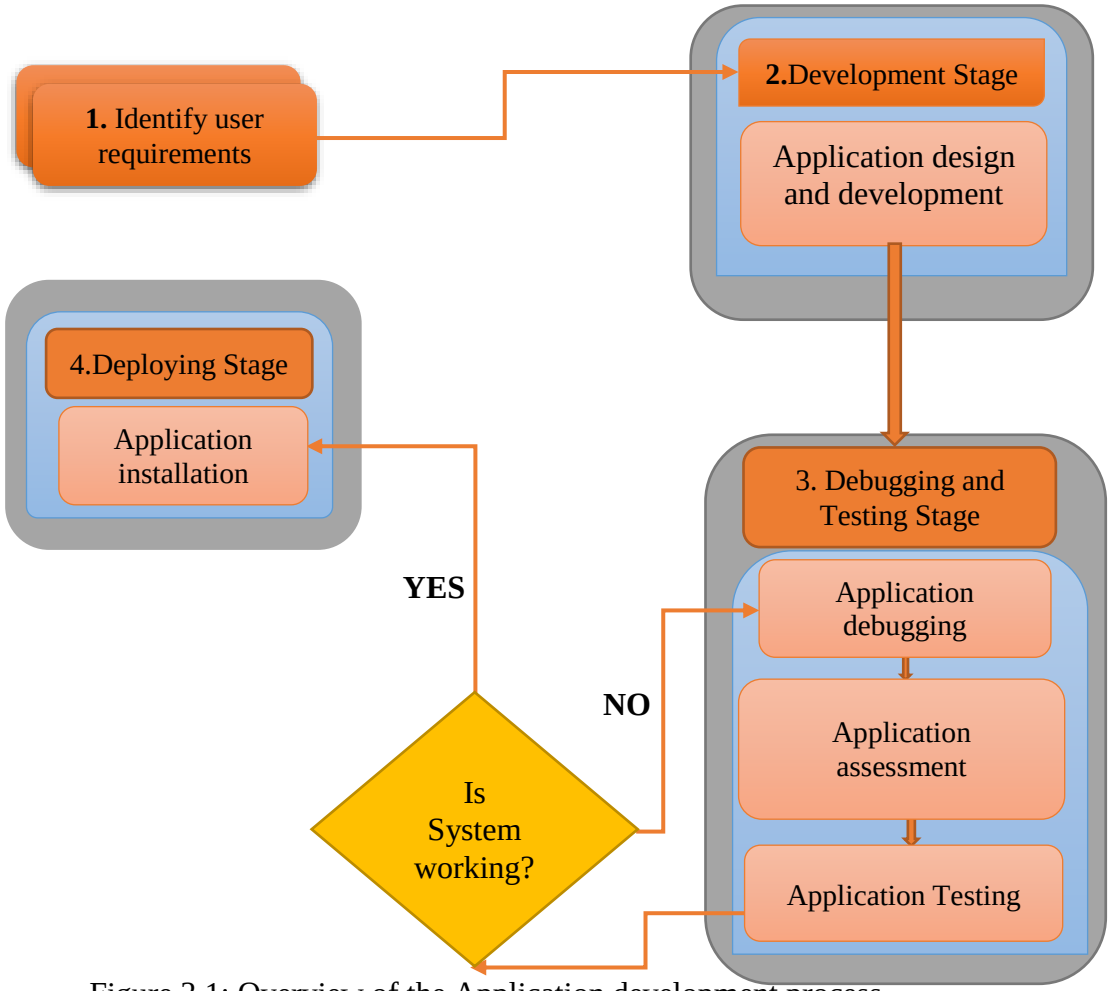


Figure 3.1: Overview of the Application development process.

3.1 Identifying User Requirements

3.1.1 Data Availability and Quality

Data used for developing the system was collected from site work and other sources, including

Google Maps, GPS sensor and phone signals. Summary of data is portrayed in table 3.1 below.

Table 3.1: The type of data used in the research and their source

SN	DATA	SOURCE	USE
1	Geographic Data	Google Places API	Geographic data is essential for determining the user's current location and nearby places. It includes coordinates (latitude and longitude) of various points of interest.
2	Place Data	Google Places API	Place Data is used to retrieve details about nearby places, such as name, address, contact information, opening hours, ratings, etc.
3	Networking Data	Place APIs	Networking Data is used to fetch data from remote APIs, such as place details, reviews, ratings, or any additional information required.
4	Map Data	Google Maps API	Map Data is used to display maps, markers, and routes for nearby places.
5	Reviews and Ratings Data	Google Places API	Reviews and ratings provide user-generated feedback and help users make informed decisions about nearby places.

3.1.2 Materials

In this research different materials were used, this included various software and hardware as shown in the table below.

Table 3.2: The material used in the research

SN	SOFTWARE/ TOOLS	FUNCTION
1	Development Machine	A computer or laptop with sufficient hardware specifications to run Android Studio and perform development tasks efficiently.
2	Android Studio	The primary Integrated Development Environment (IDE) for Android application development. Android Studio provides a range of tools and features to write, debug, and test Android applications.
3	Android SDK	The Android Software Development Kit (SDK) provides the necessary tools, libraries, and APIs for developing Android applications. It includes Android platform components, build tools, system images for emulators, and more.
4	Java Development Kit (JDK):	The Java Development Kit is required to develop Android applications using Java as the programming language. It includes the Java compiler, runtime environment, and other tools necessary for Java development.
5	Android Device or Emulator:	An Android device or emulator to test and debug the application. An actual Android device allows to test the application on real hardware, while an emulator provides a virtual device for testing purposes.
6	Google Maps API Key	To access the Google Maps SDK and use the Google Places API, you need to obtain an API key. The API key authenticates your application and allows it to access Google Maps services.
7	Collaboration and Version Control Tools	Collaboration tools (e.g., Git, GitHub, Bitbucket) and version control systems enable efficient teamwork, code sharing, and tracking changes during development. They facilitate collaboration with other developers and help manage codebase versions.
8	Internet Connectivity	A stable internet connection is required to access online resources, communicate with APIs, and fetch place data. It's crucial for testing and utilizing online services like the Google Places API.

3.1.3 NB Services Application Design:

The development of NB Services Application was based on the Five-tier client-server architecture. This architecture logically partitions major functions into five tiers. These five tiers are:

i. Presentation tier - which is responsible for the user interface (UI) and user interaction.
ii. Business Logic Tier - handles the application's core functionality and logic.
iii. Data Access Tier - responsible for retrieving and persisting data.
iv. Network Communication Tier - handles the communication between the application and remote services
v. Utility Tier - provide common functionalities and services used across different tiers.

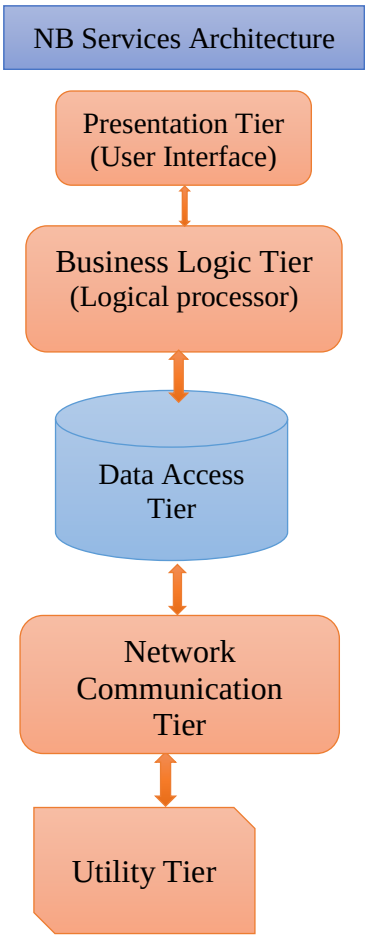


Figure 3.2: NB Services Architecture

Presentation Tier

Presentation Tier refers to the layer that is responsible for presenting data to the user through a user interface (UI). When developing a navigation application, the presentation tier is an important component as it is responsible for displaying the location data and providing the user

with an intuitive and easy-to-use interface. The presentation tier of a navigation application typically includes a map interface, which displays the location data to the user. The map interface can be created using various technologies, such as Google Maps API, OpenLayers, or Leaflet. The UI also includes buttons, dropdown menus, or other graphical elements that allow the user to interact with the application (Banerjee & Banerjee, 2019). The mobile android application was developed as a Graphical User Interface (GUI) environment using java language.

Business Logic Tier:

The Business Logic Tier, also known as the domain layer, is responsible for implementing the core functionality and logic of an application. It encompasses components and classes that handle data processing, manipulation, and business rules specific to the application's domain.

The business logic tier encapsulates the rules and operations that define how data is transformed and manipulated within the application. This tier often includes classes for data validation, calculations, decision-making, and other domain-specific operations (Fowler, 2003).

In the context of Android development, the implementation of the Business Logic Tier can vary depending on the app's requirements and architectural choices. It typically involves designing classes, interfaces, and services that encapsulate the business logic and interact with the data access and presentation tiers (Phillips, et al., 2019).

Data Access tier:

The Data Access Tier, also known as the persistence layer, is responsible for retrieving and persisting data in an application. It includes components and classes that interact with data sources, such as databases or external APIs (Gousios, Spinellis, & Zaidman, 2015).

One popular approach for data access in Android development is using the Room Persistence Library, which provides an abstraction layer over SQLite and simplifies data access operations. Room allows developers to define entities, DAOs (Data Access Objects), and a database class.(Android Developers, n.d.).

Network Communication Tier:

The Network Communication Tier is responsible for handling the communication between an application and remote services, such as APIs or web servers. It encompasses components and classes that facilitate making HTTP requests, handling network responses, and managing

authentication and authorization (Cheng, 2020).

One popular library for network communication in Android development is Retrofit. Retrofit simplifies the process of making network requests by providing a high-level API and supporting various serialization formats, such as JSON or XML.

Utility Tier:

The utility tier includes various utility classes and helper components that provide common functionalities and services used across different tiers. It may include classes for handling permissions, location services, date/time calculations, image loading, logging, or other utility operations needed in the application.

3.2 Development stage

3.2.1 Application design and Development.

"NB Services Application" was developed in Android Studio, an IDE from Microsoft that is used to develop a wide range of computer and mobile devices programs for android platforms. The Application's source code was written in java language, which is one of the built in computer programming languages within android studio.

This stage included creating a new Project in Android Studio (namely NB Services) by adding a new Windows Form followed by customizing the graphical user interface, creating essential directories, writing codes for all background processes, debugging and finally deploying the App. Within the Project files, The Application was viewed in many forms including Design View – where all the visual customization was done, and Code Editor View – where all the code writing and editing was done.

Example: Creating a new project NB Services code

```
package com.example.nbservices;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

3.2.2 Creating User Interface.

User interface (UI) design is a critical aspect of developing a navigation application, as it directly impacts the user's experience and satisfaction with the application. A well-designed UI can make the application easier to use, more intuitive, and more enjoyable for the user. According to Nielsen, clear and concise labeling of controls and functions, consistent use of colors, icons, and typography, providing feedback to users, and ensuring accessibility for all users are important considerations when designing the UI for a navigation application (Nielsen, 1993).

An Android Form created was given a name "NB Services". Then a Find Button Control and Spinner Type Control was added onto Home Windows (NB Services Application). The main function of a Find Button Control in a NB Services application is to allow users to search for specific places or services based on their preferences or requirements. It provides a convenient and efficient way for users to find desired locations within the nearby area.

Also the main function of a Spinner Type Control in a NB Services application is to provide users with a selection menu for choosing a specific category, type, or filter criteria related to the services they want to explore. The spinner allows users to easily navigate and select from a predefined list of options.

Here are the key functions of a Spinner Type Control in a NB Services application:

Category Selection:

The spinner can offer a list of categories such as restaurants, hardware's, car dealers, shopping mall, or other service types. Users can select a specific category to filter the services and view only the ones that match the selected category.

User Interaction:

The spinner enables users to interact with the NB Services application and customize their search experience. By selecting different options from the spinner, users can explore different categories of services and filters to find the desired services more efficiently.

Example: Find Button Control and Spinner Type Control

```
//Initialize Find Button Control and Spinner Type Control variable
Spinner spType;
Button btFind;
```

3.2.3 Background Processes.

In a NB Services application, the background processes involve various tasks, including fetching satellite images, defining coordinate systems, and creating layers for displaying data. These processes contribute to providing users with accurate location information and visually appealing maps. Here's a breakdown of these background processes:

3.2.4. Fetching satellite images (maps) from external servers.

Satellite Image Fetching: The application may utilize external map providers or satellite imagery services, such as Google Maps or OpenStreetMap, to fetch satellite images of the desired location. This process involves making requests to the respective APIs and retrieving the appropriate map tiles or imagery data.

```
public void onMapReady(GoogleMap googleMap) {
    //When map is ready
    map = googleMap;
    // Enable the user's location button
    googleMap.setMyLocationEnabled(true);
    //Zoom current location on map
    LatLng currentLatLng = new LatLng(location.getLatitude(), location.getLongitude());
    googleMap.addMarker(new MarkerOptions().position(currentLatLng).title("Current Location"));
    map.animateCamera(CameraUpdateFactory.newLatLngZoom(
            new LatLng(currentLat,currentLong), v: 12

    ));
}
```

3.2.5 Defining Coordinate Systems.

The application establishes a coordinate system to accurately represent locations on the map. Commonly used coordinate systems include latitude and longitude. The chosen coordinate system ensures consistency and accuracy in mapping and geolocation operations. All latitudes and longitudes were defined in "***double***" format andrendered by a string "LatLng(lat,lng)**".**

```
//Get latitude
double lat = Double.parseDouble(hashMapList.get("lat"));
//Get longitude
double lng = Double.parseDouble(hashMapList.get("lng"));
//Get name
String name = hashMapList.get("name");
//Concat latitude and longitude
LatLng latLng = new LatLng(lat,lng);
```

3.2.6 Connecting hashMap to URL.

In a NB Services application, a hashMap can be used to connect a URL (Uniform Resource Locator) to specific data or information related to services. Here's how the hashMap can be connected with URL:

Data Storage:

The hashMap serves as a data structure to store key-value pairs. In this case, the key can be a unique identifier or category related to the nearby service (e.g., "restaurants," "car dealers," "hardware"), and the value can be the corresponding URL that provides more information about that specific category.

```
private String downloadUrl(String string) throws IOException {
    //Initialize url
    URL url = new URL(string);
```

URL Retrieval: When a user interacts with the NB Services application and selects a particular category, the associated URL can be retrieved from the hashMap using the corresponding key. The URL is then used to redirect the user to the desired web page or API endpoint for further information.

```
//Initialize url
String url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json" + //Url
"?location=" + currentLat + "," + currentLong + //Location latitude and longitude
"&radius=50000" + //Nearby radius
"&types=" + placeTypeList[i] + //place type
"&sensor=true" + //Sensor
"&key=" + "AIzaSyBYn-QYcTconOJz38pHf_iDlSIcLl5kOyk";//Google map key

//Execute place task method to download json data
new PlaceTask().execute(url);
```

By connecting URLs to specific categories or types of services through a hashMap, the application can provide seamless navigation and access to additional details or resources related to the selected service category.

3.2.7 Creating Layers

In Spinner Type Control and Android Studio, Layers are created to overlay various types of data on the map. For example, a layer can be created to display the locations of nearby services, such as restaurants, hotels, or shopping malls. Another layer can represent points of interest, landmarks, or user-generated data. Each layer is associated with specific attributes and styling options to enhance the visual representation of the data.

```
//Initialize the Layers to show array of place types
final String[] placeTypeList = {"atm","bank","restaurant","bar","clothing_store","shoe_store","shopping_mall",
        "pharmacy","beauty_salon","electronics_store","car_dealer","car_repair","hardware_store"};
//Initialize array of place name
String[] placeNameList = {"ATM","BANK","RESTAURANTS","BAR & PUB","CLOTHING & FASHION","SHOE STORE","SHOPPING MALL",
        "PHARMACY & STORES","COSMETICS & BEAUTY SALON","ELECTRONICS","CAR DEALERS","SPARE PARTS & CAR SERVICES","HARDWARES"};

//Set adapter on spinner
spType.setAdapter(new ArrayAdapter<>( context: MainActivity.this
        ,android.R.layout.simple_spinner_dropdown_item,placeNameList));
```

3.2.8 Selecting a specific Point onClick

In NB Services application development, selecting a specific point onClick refers to the action taken when a user clicks or taps on a particular location or point of interest on the map or within a list of NB Services. Here's how this interaction is typically handled:

User Interaction: When the user interacts with the application by clicking or tapping on a

specific point or marker on the map or within a list, the application registers the onClick event associated with that action.

Point Identification: The application determines which point or marker was selected based on the user's interaction. This can be done by identifying the coordinates or unique identifier associated with the clicked point or by matching it with the corresponding data in the application's internal database or data source.

Information Display: Once the selected point is identified, the NB Services application displays additional information or details related to that point. This can include information such as the name, address and any other relevant information about the selected service.

3.2.9 Find/select user location

To find or select the user's location without listing points in a NB Services application, you can utilize the "private void getCurrentLocation ()" method. Here's an overview of how this method can be implemented:

Request Location Permissions:

Before retrieving the user's location, the application requests the necessary permissions to access the device's location information. This involves checking if the required permissions are granted and requesting them if not.

Initialize Location Services:

Once the location permissions are granted, the application initializes the location services API or framework provided by the platform, such as Google Play Services or Android Location Services.

Retrieve the User's Location:

Within the "getCurrentLocation ()" method, the application requests a single location update from the location services API. The location services API communicates with the device's GPS sensor and other available location providers to obtain the user's coordinates, including latitude and longitude.

Handle Location Result:

Once the location update is received, the application extracts the user's coordinates from the result. These coordinates represent the user's current location.

Use the User's Location:

The retrieved location can be used in various ways within the HB Services application. For example, the application can display the user's location on a map, center the map view around the user's location, or perform distance calculations to nearby services.

```
private void getCurrentLocation() {
    //Initialize task location
    Task<Location> task = fusedLocationProviderClient.getLastLocation();
    task.addOnSuccessListener((OnSuccessListener) (location) → {
            //When success
            if (location != null){
                //When location is not equal to null
                //Get current latitude
                currentLat = location.getLatitude();
                //Get current longitude
                currentLong = location.getLongitude();
                //Sync map
                supportMapFragment.getMapAsync(new OnMapReadyCallback() {
                    @Override
                    public void onMapReady(GoogleMap googleMap) {
                        //When map is ready
                        map = googleMap;
                        // Enable the user's location button
                        googleMap.setMyLocationEnabled(true);
                        //Zoom current location on map
                        LatLng currentLatLng = new LatLng(location.getLatitude(), location.getLongitude());
                        googleMap.addMarker(new MarkerOptions().position(currentLatLng).title("Current Location"));
                        map.animateCamera(CameraUpdateFactory.newLatLngZoom(
                                new LatLng(currentLat,currentLong), v: 12

                        ));
                    }
                });
```

3.2.10 Finding nearest point from a user's location.

Finding the nearest point from the user's location in a NB Services application involves identifying the point of interest or service location that is closest to the user's current coordinates. Here's an explanation of how this can be achieved:

Obtaining User's Location:

The application retrieves the user's current location using location services such as GPS, network-based location, or a combination of both.

Collecting Point Data:

The application needs access to a dataset or database that contains the coordinates (latitude and longitude) of various service locations or points of interest. This dataset can be obtained from

external sources, APIs, or user-generated content.

Presenting the Nearest Point:

The application displays the information of the nearest point to the user. This information can include the name, address, or any other data associated with the selected point.
Additionally, the application can provide further functionalities such as directions, reviews, or the ability to contact the nearest point.

By implementing this approach, the NB Services application provides users with the ability to find the nearest service location or point of interest based on their current location. This feature enhances the user experience by offering personalized results that are relevant and convenient.

3.3 Debugging and Testing Stage

The goal of the debugging and testing stage in the development of a NB Services application is to ensure the application functions correctly, meets the specified requirements, and provides a reliable and satisfactory user experience.

3.3.1 Application debugging.

App debugging in Android Studio for a NB Services application involves using the built-in debugging tools and features provided by the development environment to identify and fix issues. Debugging within Android Studio provides developers with powerful tools and features to identify, diagnose, and fix issues in the nearby places services app. By leveraging breakpoints, stepping through code, inspecting variables, using logcat, and making code modifications, developers can ensure that the app functions correctly, delivers accurate nearby places information, and provides a smooth user experience.

3.3.2 Application assessment.

By conducting a comprehensive application assessment within Android Studio, it is possible to identify areas of improvement, address potential issues, and ensure that the NB Services application delivers a high-quality and user-friendly experience. Regular assessments and iterations based on feedback and testing help refine the application and meet the needs of its users effectively.

3.3.2 Application testing.

By conducting a comprehensive range of tests within Android Studio, it is possible to ensure that the NB Services application functions correctly, provides accurate information, and

delivers a smooth and user-friendly experience. Testing helps identify and fix issues, validate the app's functionality, and ensure it meets user expectations.

3.4 Deploying Stage

The deployment stage in Android Studio involves preparing your nearby places services application for distribution and making it available to users. Here's an overview of the deployment process:

3.4.1 Application Installation

During the deploying stage of NB Services application in Android Studio, application installation refers to the process of installing the application on devices for testing or distribution purposes. Here's an overview of the application installation process:

Install on Test Devices: Before distributing the application, it's important to test it on various devices to ensure it works correctly and displays properly. Testing can be done on physical devices or emulators. Connect the target device to your development machine via USB and enable USB debugging in the device settings. In Android Studio, select the target device from the device dropdown menu. Click on the "Run" or "Debug" button to build and install the application on the selected device. Android Studio will handle the installation process automatically.

Install on Physical Devices for Distribution: When it’s ready to distribute nearby places services application, it can be installed manually on physical devices to the users.

CHAPTER FOUR

RESULTS AND ANALYSIS

4.1 Results And Analysis

In this chapter, the results and analysis are obtained through the implemented research methodology and discussion are presented according to the intended objective of this research which was to develop a "Location-based services to improve customers 'navigation in Kariakoo area".

4.1.1 NearBy Services Data Retrieval

The objective of this study was to retrieve nearby services data in the Kariakoo area of Dar es Salaam city using URLs, first a service or API that provides location-based information for that region should be identified. Once a suitable data source is chosen, it is responsible to consult the service's documentation to understand the URL structure and parameters required for retrieving nearby services in Kariakoo. The parameters may include the latitude and longitude coordinates of the area or specific search terms related to the services you are interested in, such as restaurants, banks, or shops. By constructing the URL with the appropriate parameters, an HTTP request to the service and receive a response containing the nearby services data for the Kariakoo area can be made.

Upon receiving the response, it is required to parse the data to extract the relevant information. The response could be in a standard format like JSON or XML, which can be processed using programming languages or libraries to extract details such as service names, addresses, contact information, or ratings. Further processing of the retrieved data could involve filtering the results based on specific criteria, sorting them by distance or rating, or presenting them on a map interface for visual representation. It is crucial to refer to the documentation of the chosen service or API to ensure accurate implementation and to account for any authentication requirements or rate limits that may be in place.

4.1.2 NB Services Application

The other objective of this study was to design, develop and finally to implements a NB Services app with a spin-type layer and a find button is to provide users with a convenient and intuitive way to discover and locate various services or amenities in their vicinity. The spin-type layer is designed to offer a visually engaging and interactive user experience. Users can rotate the spin layer to browse through different categories or types of services, such as

restaurants, hotels, pharmacies, or shopping centers. This feature allows users to quickly explore various options without the need for manual search input.

Once users have selected a specific service category, they can use the find button to trigger the search and retrieve nearby services within that category. By clicking the find button, the app initiates a data retrieval process using the user's current location as a reference point. The application then queries the database or makes API requests to fetch relevant data on nearby services that match the selected category. The retrieved data is then displayed to the user, either as a list or on a map, presenting them with the available options in their vicinity.

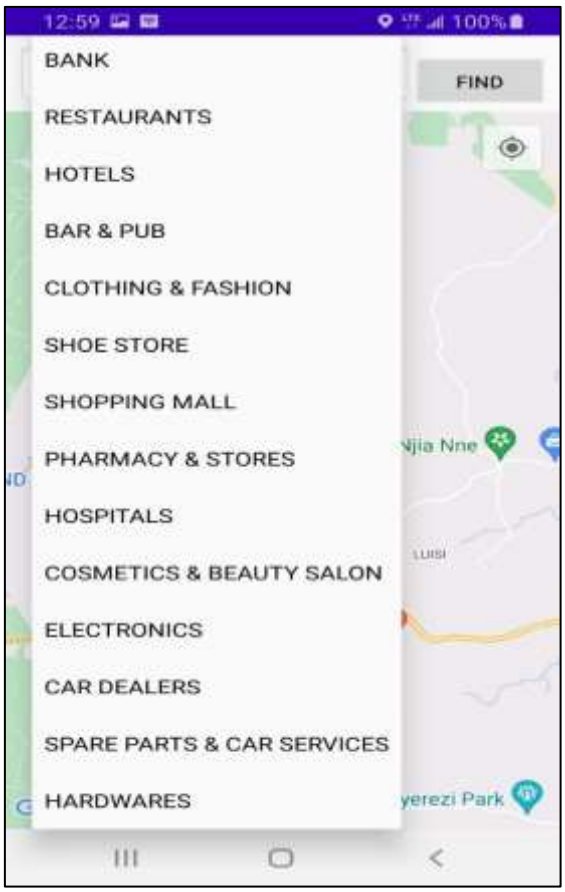


Figure 4.1: NB Services Application showing different categories of services

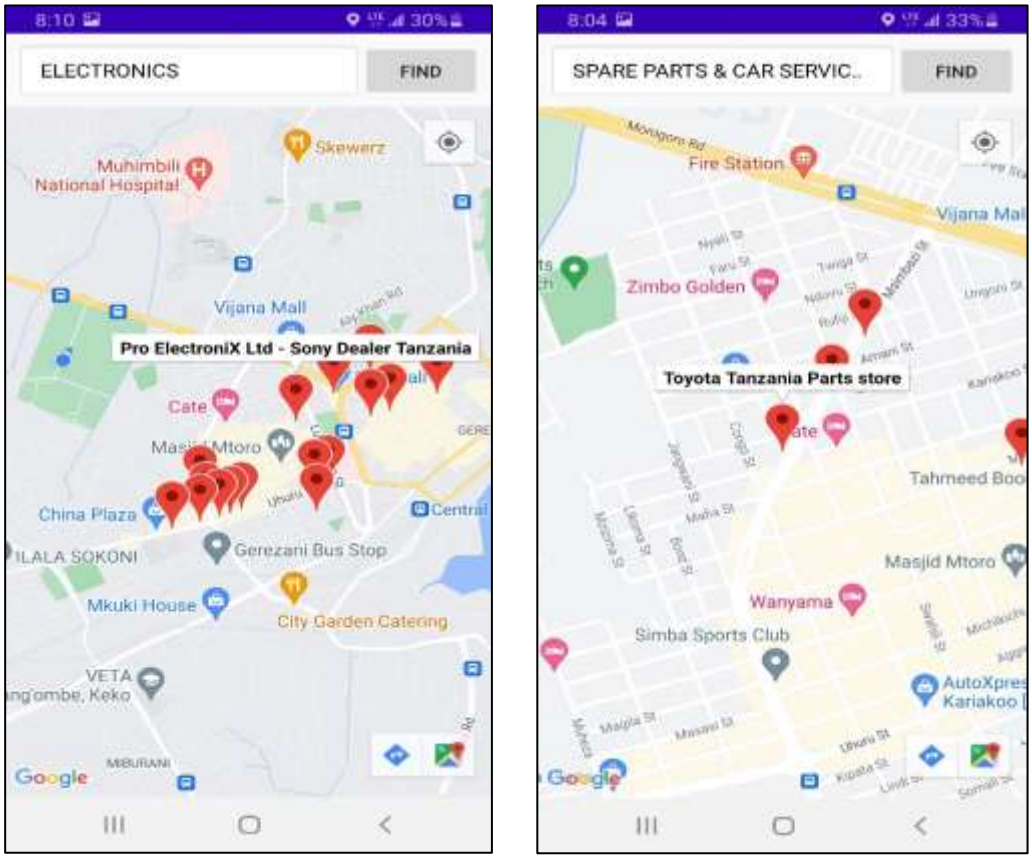
4.1.3 Proximity Analysis

Proximity analysis on nearby services in the Kariakoo area of Dar es Salaam city involves examining the distance and spatial distribution of services in relation to the user's location or a specified search area. By analyzing the proximity of nearby services, users can identify options that are conveniently located, saving time and effort. This analysis helps in understanding the accessibility of services in the Kariakoo area, identifying the nearest options for users, and allowing for the customization of search results based on desired distance parameters. It enables

users to make informed decisions by considering the proximity of services in relation to their current location or specific areas of interest within the Kariakoo neighborhood.

4.1.4 Category Selection

Category selection on nearby services in the Kariakoo area of Dar es Salaam city involves the process of choosing specific service categories from a predefined list or interface. This selection allows users to narrow down their search and focus on the types of services they are interested in finding. By presenting users with a variety of categories such as electronics, clothing & fashion, shopping mall, or spare parts & car services, the category selection feature enables users to customize their search based on their preferences and needs within the Kariakoo area. It enhances the user experience by providing a more targeted and relevant set of nearby service options, making it easier for users to find the specific types of services they require in their vicinity.

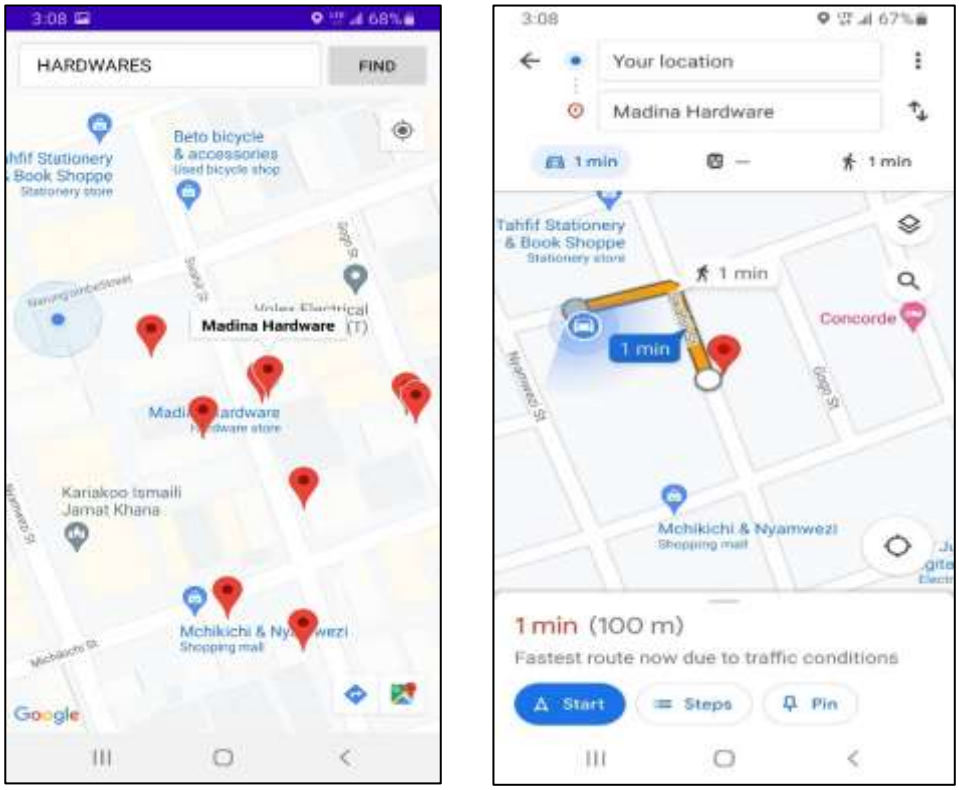


(a) (b)

Figure 4.2: NB Services showing (a) Electronics nearby services (b) Spare parts & Car services nearby services

4.1.5 Service Category Selection

Service category selection on nearby services in the Kariakoo area of Dar es Salaam city involves the process of choosing specific service categories from a provided list or interface. Users are presented with a range of categories such as electronics, clothing & fashion, shopping mall, or spare parts & car services, and more. This category selection feature allows users to refine their search and focus on the types of services they are interested in finding within the Kariakoo area. By selecting a specific category, users can customize their search experience, ensuring that the displayed results are tailored to their preferences and needs. The service category selection empowers users to easily explore and discover nearby services that align with their requirements, making it convenient for them to find the specific types of services they desire in the Kariakoo area of Dar es Salaam city.



(a) (b)

Figure 4.3: Service Category showing (a) Madina Hardware (d) Fastest route to Madina Hardware

4.1.6 Determining User Location

The application utilizes GPS (Global Positioning System) technology or a "Show Location" button to determine the user's current location. When the user opens the application, it accesses the device's GPS capabilities to retrieve precise geographic coordinates, enabling the

application to pinpoint the user's exact location. These coordinates are then used to provide accurate and location-specific services or recommendations tailored to the user's immediate vicinity. Alternatively, the application may offer a "Show Location" button that, when clicked, retrieves the device's GPS information and displays the user's current location on a map within the application. By leveraging GPS or providing a user-initiated location display, the application ensures that users can easily determine their current location and access nearby services with enhanced convenience and accuracy.

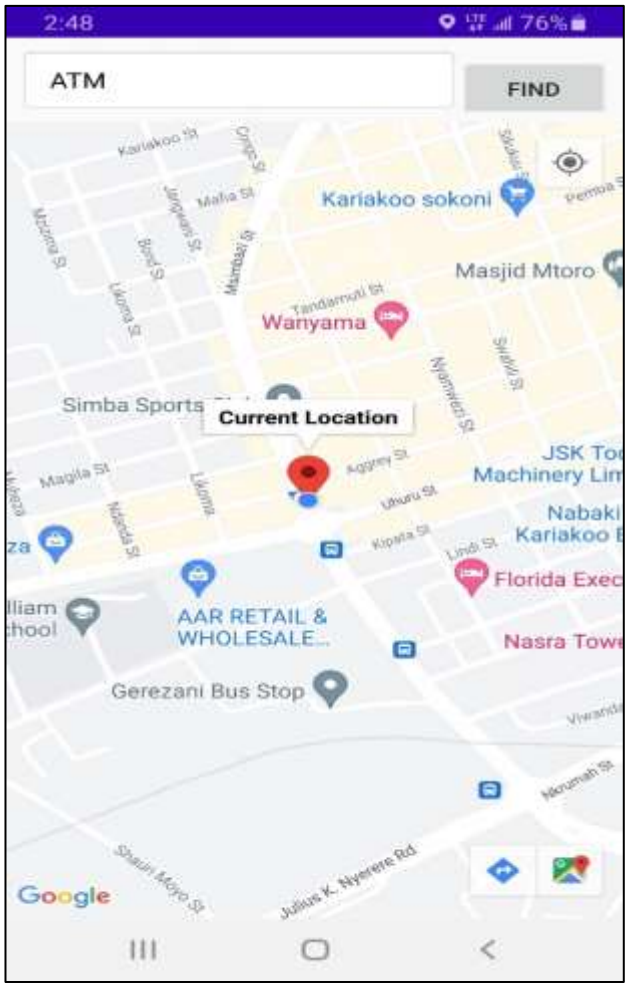


Figure 4.4: NB Services Application showing user current location by accesses the device's GPS.

CHAPTER FIVE
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The purpose of this study was to develop and implements a mobile phone platform using Web-GIS which was succeeded. By leveraging location-based technology and real-time data, applications have become invaluable tools for individuals seeking various services, such as restaurants, shops, gas stations, healthcare facilities, and more. They have simplified the process of finding nearby businesses and establishments, eliminating the need for tedious research and guesswork.

This study has played a crucial role in supporting local businesses and fostering economic growth. By connecting users with nearby services establishments, these applications help drive foot traffic and increase visibility for small businesses that may otherwise struggle to reach their target audience. Additionally, nearby services applications have improved the safety and security of users by offering real-time alerts, such as emergency services, road closures. This proactive approach to information dissemination ensures that individuals stay informed and can navigate their surroundings more effectively.

5.2 Recommendations

One of limitations of the NB Services Application is that it only works on Android smartphones; it is crucial to address the platform limitation by expanding the application's compatibility to include iOS and Windows devices. Developing versions of the application for these platforms would significantly increase its user base and accessibility, allowing a wider range of users to benefit from its features and services. This can be achieved by utilizing cross-platform development frameworks such as React Native or Xamarin, which enable developers to write code once and deploy it across multiple platforms. By making the application available on iOS and Windows, it can reach a broader audience and maximize its impact.

REFERENCES

Android Developers, (. (n.d.). *Room Persistence Library. Retrieved from*. Retrieved from https://developer.android.com/training/data-storage/room.

Banerjee, S., & Banerjee, S. (2019). *Development of a location-based social networking application using GPS and Google Maps API. In 2019 2nd International Conference on Computing, Communication, Control and Automation (ICCUBEA) (pp. 1-6). IEEE.*

Bishr, Y., & Kuhn, W. (2018). *Server GIS. In International Encyclopedia of Geography: People, the Earth, Environment and Technology (pp. 1-10). John Wiley & Sons, Ltd.*

Cao, X., Mao, B., Kwan, M., & Huang, F. (2018). *Design, implementation, and evaluation of the Web Map Services (WMS) performance benchmark. Transactions in GIS, 22(3), 665-683.*

Cheng, Z. (2020). *Professional Android Development: The Roadmap to Successful Mobile App Design and Development. Wiley.*

Deng:Li, L, Li, W., M, & Wu, H. (2019). *A study on real-time data synchronization strategy in distributed GIS. ISPRS International Journal of Geo-Information, 8(12), 536.*

Ershad, E. Ali, and Components. (2020). *"Geographic Information System (GIS): Definition, Definition, Development, Applications,"Development, Applications,.*

Esri, 2. (2021). *Desktop GIS. Retrieved from https://www.esri.com/en-us/what-is-gis/overview#desktop-gis.*

Fowler, M. (2003). *Patterns of Enterprise Application Architecture. Addison-Wesley Professional.*

Goodchild, M., & Li, L. (2012). *Assuring the quality of volunteered geographic information. Spatial statistics, 1, 110-120.*

Gousios, G., Spinellis, D., & Zaidman, A. (2015). *Software Analytics: So What? IEEE Software, 32(5), 14-16.*

Gupta & Rao. (2017.). "An exploration to location based service and its privacy preserving techniques: a survey,". *Wireless Personal Communications, vol. 96, no. 2,*, pp. 1973-2007,.

Haklay, M., Basiouka, S., Antoniou, V., & Ather, A. (2010). How many volunteers does it take to map an area well? The validity of Linus' law to volunteered geographic information. *The Cartographic Journal*, 47(4), 315-322.

Han, J., Kim, M., & Kim, J. (2017). The use of Google Earth for teaching GIS: A review of the

literature. *Journal of Geography in Higher Education*, 41(1), 1-18.

Hazas, M., Scott, J., & Krumm, J. (2013). Location-Based Systems: A Review and Future Research Directions. ACM Computing Surveys,. *Article 10*, 45(1).

Huang, H., Meng, L., & Zang, L. (2020). *Web GIS and mobile GIS for public participation in environmental protection: A review. ISPRS International Journal of Geo-Information, 9(8), 476.*

Huang, X., Cheng, S., Ma, W., Wang, J., Zhang, X., & Liu, X. (2020). *Design and implementation of augmented reality mobile GIS based on Android platform. Future Generation Computer Systems, 107, 963-972.*

Huang, X., Yan, X., & Li, Q. (2019). *An optimized framework of distributed GIS for near real-time forest fire prediction. Future Generation Computer Systems, 91, 255-264.*

Kerski and T. R. Baker. (2019). "Infusing educational practice with Web GIS," in Geospatial technologies in geography education:. *Springer,*, pp. 3-19.

Kim, G. Yang, H. Jung, S. H. Lee, and J. J. Ahn. (2019.). "An intelligent product recommendation model to reflect the recent purchasing patterns of customers,". *Mobile Networks Applications, vol. 24, no. 1,*, pp. 163-170.

Korhonen, H., Li, Y., & Aho, E. (2019). *Mobile Operating Systems: A Comparative Analysis. IEEE Access, 7, 55057-55081*.

Kraak, M. J., & Ormeling, F. (2019). *Cartography: Visualization of Spatial Data. CRC Press.*

Kulawiak, A. Dawidowicz, and M. E. Pacholczyk. (2019.). "Analysis of server-side and client-side Web-GIS data processing methods on the example of JTS and JSTS using open data from OSM and geoportal,". *Computers & Geosciences, vol. 129,*, pp. 26-37,.

Li, &. L. (2018). *Mobile GIS. In International Encyclopedia of Geography: People, the Earth, Environment and Technology (pp. 1-7). Wiley-Blackwell.*

Logesh, S. V. (2019). "Efficient user profiling based intelligent travel recommender system for individual and group of users,". *Mobile Networks Applications, vol. 24, no. 3,*, pp. 1018-1033,.

Longley, P. A. (2015). *Geographic Information Systems and Science. .* Hoboken, NJ: Wiley.

Longley, P., Goodchild, M., Maguire, D., & Rhind, D. (2015). *Geographic Information Systems and Science. Hoboken, NJ: Wiley.*

Nazzawi, Al & T, S. (2018). "Toward privacy protection for location based recommender systems: a survey of the state-of-the-art,". *in 2018 1st International Conference on Computer Applications & Information Security (ICCAIS),*, pp. 1-7: IEEE.

Nebert, D., Atkinson, P., Curran, P., Foody, G., & Rosenqvist, A. (2018). *Distributed GIS. In International Encyclopedia of Human Geography (pp. 27-33). Elsevier.*

Obaidat, K. A, G., & A.-M, B. (2018). *"Integration of geographic information system (GIS) and PAVER system toward efficient pavement maintenance management system (PMMS)," Jordan Journal of Civil Engineering, vol. 12, no. 3.*

Oxford Dictionary. (n.d.). *(n.d.). API. Retrieved from*. Retrieved from https://www.oxfordlearnersdictionaries.com/definition/english/api.

Phillips, B., Stewart, C., & Marsicano, K. (2019). *Android Programming: The Big Nerd Ranch Guide. Big Nerd Ranch.*

Rajalakshmi & M, Goyal. (2018,). "Location-Based Services: Current State of The Art and Future Prospects," Singapore,. *Springer Singapore.*, pp. 625-632:.

Ravi. (2019.). "Hybrid location-based recommender system for mobility and travel planning,". *Mobile Networks Applications, vol. 24, no. 4,*, pp. 1226-1239,.

Research & Market. (2021). *Mobile GIS Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021 - 2026). Retrieved from.*

Rogerson, P. (2019). *Geographic Information Systems (GIS): Trends and Technologies. Routledge.*

Salas, E., Kray, C., Hollenstein, A., & Hertweck, M. (2019). *Stream-processing for real-time analysis of Volunteered Geographic Information (VGI) in Mobile GIS applications. International Journal of Digital Earth, 12(1), 17-35.*

Shannon, K & Philip, H;. (2021). *Web GIS: Architectural patterns and practices. LA:ESRI.*

Silberschatz, A., Galvin, P., & Gagne, G. (2020). *Operating System Concepts, 10th Edition. Wiley.*

Zhang, L. Z., Li, L., & Yuan, M. (2017). *A cloud-based architecture for next-generation GIS. Computers, Environment and Urban Systems, 63, 74-85.*

Zhang, Z. (2018). *Location-based Recommender Systems. In Recommender Systems: The Textbook (pp. 319-335). Springer.*

APPENDICES

APPENDIX 1

MAIN ACTIVITY

```
package com.example.nbservices;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;

import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;

import com.google.android.gms.common.util.Strings;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;

import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.HashMap;
import java.util.List;
```

```
public class MainActivity extends AppCompatActivity implements OnMapReadyCallback {
   private GoogleMap googleMap;
   private SupportMapFragment mapFragment;
   //Initialize Find Button Control and Spinner Type Control variable
   Spinner spType;
   Button btFind;
   SupportMapFragment supportMapFragment;
   GoogleMap map;
   FusedLocationProviderClient fusedLocationProviderClient;
   double currentLat = 0, currentLong = 0;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      //Assign variable
      spType = findViewById(R.id.sp_type);
      btFind = findViewById(R.id.bt_find);
      supportMapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.google_map);
      //Initialize the Layers to show array of place types
      final String[] placeTypeList =
{"atm","bank","restaurant","bar","clothing_store","shoe_store","shopping_mall",

"pharmacy","hospital","beauty_salon","electronics_store","car_dealer","car_repair",
"hardware_store"};
      //Initialize array of place name
      String[] placeNameList = {"ATM","BANK","RESTAURANTS","BAR &
PUB","CLOTHING & FASHION","SHOE STORE","SHOPPING MALL",
            "PHARMACY & STORES","HOSPITALS","COSMETICS & BEAUTY
SALON","ELECTRONICS","CAR DEALERS","SPARE PARTS & CAR
SERVICES","HARDWARES"};
      //Set adapter on spinner
      spType.setAdapter(new ArrayAdapter<>(MainActivity.this
            ,android.R.layout.simple_spinner_dropdown_item,placeNameList));
      //Initialize fused location provider client
      fusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);
      //Check permission
      if (ActivityCompat.checkSelfPermission(MainActivity.this ,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED){
         //When permission granted
```

```
            //Call method
            getCurrentLocation();
        }else {
            //When permission denied
            //Request permission
            ActivityCompat.requestPermissions(MainActivity.this
                    ,new String[]{Manifest.permission.ACCESS_FINE_LOCATION},44);
        }
        btFind.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //Get selected position of spinner
                int i = spType.getSelectedItemPosition();
                //Initialize url
                String url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json" +
//Url
                "?location=" + currentLat + "," + currentLong + //Location latitude and
longitude
                "&radius=50000" + //Nearby radius
                "&types=" + placeTypeList[i] + //place type
                "&sensor=true" + //Sensor
                "&key=" + getResources().getString(R.string.google_map_key);//Google map key

                //Execute place task method to download json data
                new PlaceTask().execute(url);
            }
        });
    }
    private void getCurrentLocation() {
        //Initialize task location
        Task<Location> task = fusedLocationProviderClient.getLastLocation();
        task.addOnSuccessListener(new OnSuccessListener<Location>() {
            @Override
            public void onSuccess(final Location location) {
                //When success
                if (location != null){
                    //When location is not equal to null
                    //Get current latitude
                    currentLat = location.getLatitude();
                    //Get current longitude
                    currentLong = location.getLongitude();
                    //Sync map
                    supportMapFragment.getMapAsync(new OnMapReadyCallback() {
```

```
                @Override
                public void onMapReady(GoogleMap googleMap) {
                    //When map is ready
                    map = googleMap;
                    // Enable the user's location button
                    googleMap.setMyLocationEnabled(true);
                    //Zoom current location on map
                    LatLng currentLatLng = new LatLng(location.getLatitude(),
location.getLongitude());
                    googleMap.addMarker(new
MarkerOptions().position(currentLatLng).title("Current Location"));
                    map.animateCamera(CameraUpdateFactory.newLatLngZoom(
                            new LatLng(currentLat,currentLong), 12
        }
    });
  }

  @Override
  public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
    if (requestCode == 44){
      if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED){
        //When permission granted
        //Call method
        getCurrentLocation();
      }
    }
  }
  @Override
  public void onMapReady(GoogleMap googleMap) {
  }
  private class PlaceTask extends AsyncTask<String, Integer, String> {
    @Override
    protected String doInBackground(String... strings) {
      String data = null;
      try {
        //Initialize data
        data = downloadUrl(strings[0]);
      } catch (IOException e) {
        e.printStackTrace();
      }
      return data;
```

```
        }
        @Override
        protected void onPostExecute(String s) {
            //Execute parser task
            new ParserTask().execute(s);
        }
    }
    private String downloadUrl(String string) throws IOException {
        //Initialize url
        URL url = new URL(string);
        //Initialize connection
        HttpURLConnection connection = (HttpURLConnection) url.openConnection();
        //Connect connection
        connection.connect();
        //Initialize input stream
        InputStream stream = connection.getInputStream();
        //Initialize buffer reader
        BufferedReader reader = new BufferedReader(new InputStreamReader(stream));
        //Initialize string builder
        StringBuilder builder = new StringBuilder();
        //Initialize string variable
        String line = "";
        //use while loop
        while ( (line = reader.readLine()) !=null ){
            //Append line
            builder.append(line);
        }
        //Get append data
        String data = builder.toString();
        //Close reader
        reader.close();
        //Return data
        return data;
    }
    private class ParserTask extends AsyncTask<String, Integer, List<HashMap<String,
String>>> {
        @Override
        protected List<HashMap<String, String>> doInBackground(String... strings) {
            //Create json parser class
            JsonParser jsonParser = new JsonParser();
            //Initialize hash map list
            List<HashMap<String,String>> mapList = null;
            JSONObject object = null;
```

```
            try {
                //Initialize json object
                object = new JSONObject(strings[0]);
                //Parse json object
                mapList = jsonParser.parseResult(object);
            } catch (JSONException e) {
                e.printStackTrace();
            }
            //Return map list
            return mapList;
        }
        @Override
        protected void onPostExecute(List<HashMap<String, String>> hashMaps) {
            //Clear map
            map.clear();
            //Use for loop
            for (int i=0; i<hashMaps.size(); i++){
                //Initialize hash map
                HashMap<String,String> hashMapList = hashMaps.get(i);
                //Get latitude
                double lat = Double.parseDouble(hashMapList.get("lat"));
                //Get longitude
                double lng = Double.parseDouble(hashMapList.get("lng"));
                //Get name
                String name = hashMapList.get("name");
                //Concat latitude and longitude
                LatLng latLng = new LatLng(lat,lng);
                //Initialize marker options
                MarkerOptions options = new MarkerOptions();
                //Set position
                options.position(latLng);
                //Set title
                options.title(name);
                //Add marker on map
                map.addMarker(options);
            }
        }
    }
}
```

APPENDIX 2

JSONPARSER.JAVA

```
package com.example.nbservices;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class JsonParser {
   private HashMap<String, String> parseJsonObject(JSONObject object){
      //Initialize Hash map
      HashMap<String, String> dataList = new HashMap<>();
      try {
         //Get name from object
         String name = object.getString("name");
         //Get latitude from object
         String latitude = object.getJSONObject("geometry")
               .getJSONObject("location").getString("lat");
         //Get longitude from object
         String longitude = object.getJSONObject("geometry")
               .getJSONObject("location").getString("lng");
         //Put all values in hash map
         dataList.put("name",name);
         dataList.put("lat",latitude);
         dataList.put("lng",longitude);
      } catch (JSONException e) {
         e.printStackTrace();
      }
      //Return hash map
      return dataList;
   }
   private List<HashMap<String,String>> parseJsonArray(JSONArray jsonArray){
      //Initialize hash map list
      List<HashMap<String,String>> dataList = new ArrayList<>();
      for (int i=0; i<jsonArray.length(); i++){
         try {
            //Initialize hash map
            HashMap<String,String> data = parseJsonObject((JSONObject) jsonArray.get(i));
```

```
            //Add data in hash map list
            dataList.add(data);
          } catch (JSONException e) {
            e.printStackTrace();
          }
        }
        //Return hash map list
        return dataList;
    }
    public List<HashMap<String,String>> parseResult(JSONObject object){
        //Initialize json array
        JSONArray jsonArray = null;
        try {
            //Get result array
            jsonArray = object.getJSONArray("results");
        } catch (JSONException e) {
            e.printStackTrace();
        }
        //Return array
        return parseJsonArray(jsonArray);
    }
}
```

APPENDIX 3

BUILD.GRADLE

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 29
    buildToolsVersion "29.0.3"

    defaultConfig {
        applicationId "com.example.nbservices"
        minSdkVersion 23
        targetSdkVersion 29
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
'proguard-rules.pro'
        }
    }

}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    implementation 'androidx.appcompat:appcompat:1.0.2'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
    implementation 'com.google.android.gms:play-services-maps:17.0.0'
    implementation 'com.google.android.gms:play-services-location:17.0.0'
}
```

APPENDIX 4

ANDROIDMANIFEST.XML

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.nbservices">

   <uses-permission android:name="android.permission.INTERNET"/>
   <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/>
   <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />

   <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:roundIcon="@mipmap/ic_launcher_round"
      android:supportsRtl="true"
      android:theme="@style/AppTheme">

      <meta-data
         android:name="com.google.android.geo.API_KEY"
         android:value="@string/google_map_key"/>

      <activity android:name=".MainActivity">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
   </application>

</manifest>
```

APPENDIX 5

ACTIVITY_MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   tools:context=".MainActivity">

   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="horizontal"
      android:padding="8dp">

      <Spinner
         android:layout_width="0dp"
         android:layout_height="match_parent"
         android:layout_weight="1"
         android:id="@+id/sp_type"
         android:background="@android:drawable/editbox_background"/>

      <Button
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:id="@+id/bt_find"
         android:text="Find"/>

   </LinearLayout>

   <fragment
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:id="@+id/google_map"
      android:name="com.google.android.gms.maps.SupportMapFragment"/>


</LinearLayout>
```