ADEOSUN NEHEMIAH OLUFEMI - ANALYSIS OF LTE DOWNLINK PERFORMANCE UNDER DIFFERENT MOBILITY CONDITIONS

NEU 2021

ANALYSIS OF LTE DOWNLINK PERFORMANCE UNDER DIFFERENT MOBILITY CONDITIONS

A THESIS SUBMITTED TO THE
INSTITUTE OF GRADUATE STUDIES
OF
NEAR EAST UNIVERSITY

By
ADEOSUN NEHEMIAH OLUFEMI

In Partial Fulfillment of the Requirements for
the Degree of Master of Science
in
Computer Engineering

NICOSIA, 2021

ANALYSIS OF LTE DOWNLINK PERFORMANCE UNDER DIFFERENT MOBILITY CONDITIONS

A THESIS SUBMITTED TO THE

INSTITUTE OF GRADUATE STUDIES

OF

NEAR EAST UNIVERSITY

By
ADEOSUN NEHEMIAH OLUFEMI

In Partial Fulfilment of the Requirements for the
Degree of Master of Science
in
Computer Engineering

NICOSIA, 2021

Nehemiah Olufemi Adeosun: ANALYSIS OF LTE DOWNLINK PERFORMANCE UNDER DIFFERENT MOBILITY CONDITIONS

Approval of Director of Institute of Graduate Studies

Prof. Dr. K. Husnu Can Baseri

We certify this thesis is satisfactory for the award of the degree of Master In Computer Engineering

Examining Committee in Charge:

Assoc. Prof. Dr. Yöney Kırsal EVER — Committee Chairman, Department of Software Engineering, NEU

Yoney

Prof. Dr. Fadi Al-Turjman — Supervisor, Department of Software Engineering, NEU

F. Alturjman

Assist. Prof. Dr. Elbrus İmanov — Department of Computer Engineering, NEU

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name, Surname: Nehemiah Olufemi Adeosun

Signature:

Date:

ACKNOWLEDGMENT

I would like to sincerely thank my supervisor Prof. Dr. Fadi AL-TURJMAN for his understanding, patience, and guidance throughout my graduate studies at Near East University. His supervision was paramount in providing a well-rounded experience in projecting my long-term career goals. He encouraged me to be confident in everything I do. I graciously thank you for all you have done for me Prof. Dr. Fadi AL-TURJMAN.

Lastly, I would like to thank my family for their consistent prayers and love during the time I was not available.

This thesis is wholeheartedly dedicated to my beloved parents, who have been my source of inspiration and gave me strength when I thought of giving up, who continually provide their moral, spiritual, emotional, and financial support. To my brothers, sisters, relatives, mentor, friends, and classmates who shared their words of advice and encouragement to finish this work. And lastly, I dedicated this thesis to the Almighty God, thank you for the guidance, strength, power of mind, protection and skills and for giving me a healthy life to be able to come up with this work.

ABSTRACT

The Internet of Things (IoT) additionally called the Internet of Everything is another innovation worldview imagined as a worldwide technology of either machines or gadgets equipped for cooperating with one another. It accomplishes the objective of intelligent identifying, locating, tracking, monitoring and overseeing things. This work proposes a context-aware platform that provides mobile applications as a service. The application utilizes a few sorts of context information, for example, location information. It plans to build up mobile application that runs on android mobile system which displays current location of mobile app user with the help of the information stored in the database. Latitude and longitude are the forms used to specify a location of cell device. This information is then converted into a full address with the help of this application which includes current location name, location destination address, website, contact phone number and map. If there should arise an occurrence of emergency, the user can basically open up the application and effectively look for destination within a specific area. The design shows the best way to execute, program and build up this application and has been tried on barely three cell phones. The versatile application upholds state protection, where user explicit information and application status are put away for the user's future reference. It additionally empowers application suppliers to progressively control admittance to the usefulness of their applications. Execution assessment results show that the application can utilize server versatile asset provisioning to offer adaptable adaptability, while fulfilling certain QoS requirements. The app is most useful for new students and staffs who are not familiar with their surroundings within the campus environment. It provides information about topics of chosen interest like the school library, cafeteria, faculty, lecture rooms, clinic, bus stations, ATM points and many more. Information will be displayed on the platform based upon the location as determined by the information available on the mobile application. It provides advanced mapping systems that will display different places based on the interest of app user. It would be easy to search nearby place and navigate in a user friendly way. This mobile application is developed to reduce stress for new students and staffs who find themselves on campus. Having said these, we have decided to add an extra feature to our app which connects personal calendars to geo-location on campus for services such as scheduling and showing students and staff their daily lectures and meetings locations on Google map.

Keywords: internet of things; mobility; location; tracking; mobile application; map

ÖZET

Nesnelerin İnterneti (IoT) ayrıca Her Şeyin İnterneti olarak adlandırılan, birbirleriyle işbirliği yapmak için donatılmış makinelerin veya cihazların dünya çapında bir teknolojisi olarak hayal edilen başka bir yenilik dünya görüşüdür. Akıllı tanımlama, bulma, izleme, izleme ve denetleme hedefini gerçekleştirir. Bu çalışma, mobil uygulamaları hizmet olarak sağlayan içeriğe duyarlı bir platform önerir. Uygulama, konum bilgileri gibi birkaç tür bağlam bilgisi kullanır. Veritabanında depolanan bilgilerin yardımıyla mobil uygulama kullanıcısının mevcut konumunu görüntüleyen android mobil sistemde çalışan mobil uygulama oluşturmayı planlıyor. Enlem ve boylam, hücre aygıtının konumunu belirtmek için kullanılan formlardır. Bu bilgiler daha sonra mevcut konum adı, konum hedef adresi, web sitesi, iletişim telefon numarası ve haritayı içeren bu uygulamanın yardımıyla tam bir adrese dönüştürülür. Acil bir durum ortaya çıkarsa, kullanıcı temelde uygulamayı açabilir ve belirli bir alanda hedefi etkili bir şekilde arayabilir. Tasarım, bu uygulamayı yürütmenin, programlamanın ve oluşturmanın en iyi yolunu gösterir ve ancak üç cep telefonunda denenmiştir. Çok yönlü uygulama, kullanıcının gelecekteki başvurusu için kullanıcı açık bilgilerinin ve uygulama durumunun kaldırıldığı durum korumasını savunur. Ayrıca, uygulama tedarikçilerine uygulamalarının yararlılığına kabulü aşamalı olarak kontrol etmelerini de güçlendirir. Yürütme değerlendirme sonuçları, uygulamanın belirli QoS gereksinimlerini yerine getirirken uyarlanabilir uyarlanabilirlik sunmak için sunucu çok yönlü varlık sağlamayı kullanabileceğini göstermektedir. Uygulama, kampüs ortamında çevresine aşina olmayan yeni öğrenciler ve personel için en kullanışlı olanıdır. Okul kütüphanesi, kafeterya, fakülte, dershaneler, klinik, otogarlar, ATM noktaları ve daha birçok ilgi alanı hakkında bilgi vermektedir. Bilgiler, mobil uygulamada bulunan bilgilerle belirlenen konuma göre platformda görüntülenecektir. Uygulama kullanıcısının ilgisine göre farklı yerleri görüntüleyecek gelişmiş haritalama sistemleri sağlar. Yakındaki yeri aramak ve kullanıcı dostu bir şekilde gezinmek kolay olurdu. Bu mobil uygulama, kendilerini kampüste bulan yeni öğrenciler ve personel için stresi azaltmak için geliştirilmiştir. Bunları söyledikten sonra, öğrencilere ve personele Günlük derslerini ve toplantı konumlarını Google haritasında planlama ve gösterme gibi hizmetler için kişisel takvimleri kampüsteki coğrafi konuma bağlayan uygulamamıza ekstra bir özellik eklemeye karar verdik.

***Anahtar Kelimeler*:** nesnelerin interneti; hareketlilik; yer; izleme; mobil uygulama; harita

TABLE OF CONTENTS

Page

ACKNOWLEDGMENT i

ABSTRACT ii

ÖZET iii

TABLE OF CONTENTS iv

LIST OF ABBREVIATIONS v

LIST OF FIGURES vi

LIST OF TABLES vii

CHAPTER 1: INTRODUCTION

1.1 Overview 1

CHAPTER 2: BACKGROUND AND RELATED RESEARCH

2.1 mSpace Mobile: A Mobile Application for Location 4

2.2 mSpace Mobile System Architecture 6

2.3 mSpace Tests Execution and Security 8

CHAPTER 3: SYSTEM ARCHITECTURE

3.1 Material Design and Optimization 11

3.2 Android Studio setup 12

3.3 Backend customization 13

CHAPTER 4: USER INTERFACE

4.1 Installation 22

4.2 Foreground operations 22

4.3 Background operations 23

4.4 Resilience of the NEU Places application 23

4.5 Customization 24

4.6 Data processing and analysis 24

4.7 Response time 25

4.8 Calendar Map with Translator Feature 26

CHAPTER 5: IMPLEMENTATION AND METHODOLOGY

5.1 Android Location API 28

5.2 The API of Google Places 29

5.2.1 Place Searches 29

5.2.2 System Testing 31

CHAPTER 6: SYSTEM PERFORMANCE AND VALIDATION

6.1 Test Environment and Test Plan 32

6.2 Results 33

6.3 Experimental Setups 38

6.3.1 Launch time 39

6.3.2 Time to navigate between pages of the application 40

6.4 The Usage of Central Processing Unit (CPU) 41

CHAPTER 7: CONCLUSION

REFERENCES 47

APPENDICES

Appendix 1: AndroidManifest XML code 52

Appendix 2: AdapterPlaceGrid Java code 55

Appendix 3: AdapterSuggestionSearch Java code 61

Appendix 4: AppConfig Java code 64

Appendix 5: Constant Java code 65

Appendix 6: Similarity report…………………………………………………….. 66

Appendix 7: Ethical Approval letter……………………………………………… 67

LIST OF ABBREVIATIONS

P:	Actively Maintained
FTU:	Free To Use
AS:	Available in App Store
CR:	Continous Recording
C:	Customizable
GPS:	Extracts signal from global positioning system
Wi-Fi:	Computes location from Wi-Fi connection
Ce:	Cellular location tracking
PI:	Assists with accurate and longitudinal point of interest mapping
PM:	Allows for reliable path mapping of movements
OS:	Open source code
MA:	Mobile Application
MQ:	mSpace Query Service
RDF:	Resource Description Framework
MK:	mSpace Knowledge interface
SOAP:	Simple Object Access Protocol
HTTP:	Hypertext Transfer Protocol

LIST OF FIGURES

Figure 1.1: Different features on the app User Interface.. 2

Figure 2.1: App interface sections.. 5

Figure 2.2: mSpace Mobile architecture.. 6

Figure 3.1: Android mobile application activities lifecycle... 10

Figure 3.2: Android Studio Interface... 12

Figure 3.3: Android Studio Interface after opening a project.. 13

Figure 3.4: The structure of NEU Places application... 14

Figure 3.5: Creating SQL Database for NEU Places app.. 14

Figure 3.6: Adding tables to NEU Places database.. 15

Figure 3.7: Propose System Architecture... 16

Figure 3.8: The database architecture... 17

Figure 3.9: Linking the SQL to NEU Places app.. 17

Figure 3.10: Generating the .apk file.. 17

Figure 4.1: Nine screens that explain how the app works... 20

Figure 6.1: Execution Time of the Proposed app.. 33

Figure 6.2: Execution Time of Iterate thought array... 34

Figure 6.3: Execution Time of Iterate thought matrix... 35

Figure 6.4: Execution Time of Fibonacci number... 35

Figure 6.5: Memory Usage of all devices.. 37

Figure 6.6: CPU Usage.. 38

Figure 6.7: Calendar Map.. 43

Figure 6.8: Android Studio Profiler...44

LIST OF TABLES

Table 5.1: The arguments for Place Search API.. 30

Table 5.2: The arguments for Place Search Requests.. 31

Table 6.1: Phones for Testing.. 32

Table 6.2: Launch times (in ms)... 39

Table 6.3: The time taken to open a different activity (in ms)... 40

Table 6.4: The time taken to return to the previous activity (in ms)...................................... 41

Table 6.5: The consumption of the phone memory after launching the app.......................... 41

Table 6.6: Research methods features of how to track locations with Smartphones............. 42

Table 6.7: Comparative results for response time.. 44

CHAPTER 1

INTRODUCTION

1.1 Overview

With the fast increment of Smartphone's gadgets, the users are additionally expanding as it turns into a significant wellspring of data. It is obviously showing the interest of users in portable applications that give storage, analysis and visualization of the information (which for retrieval in the reference is the name of the user who posted the app to Google Play Store, AlienmanTechnologyLLC, 2015), (Hardy, 2014). A large portion of advanced phones furnished with PC preparing capacity are utilized to get to organize and utilize various applications created to meet the user requirement via mobile communication network. (Salman and Saleem, 2015). Distinctive android Mobile applications have been created including top paid and free applications additionally some focus on GPSGlobal Positioning System are developed (Saleem and Khan, 2015). Recent years have seen expanding force for portable applications that can adjust their conduct as indicated by client setting.Setting like such incorporates area data; client profile, client evaluations, gadget profile, and time. There are a few stages for creating advanced cell applications. Majority of the telephones and helpful gadgets uphold Android operating systems (OS).

The aim of this thesis is to develop a mobile app on Android platform and compare its response time with the existing works then analyzing the performance under different mobility conditions. We have used the LTE and WiFi for this comparison and ensure that the proposed work have the best performance. We propose NEU Places, a mobile application that allows users to travel through the campus by utilizing contexts that are significant to them. Someone that just got admitted to the university may find it difficult to locate the nearby restaurant, lecture rooms, his/her faculty/department and other places within the campus vicinity. He/she may decide to open the mobile app and search for query like “International Office”. The NEU Places app (Fig. 1) joins an imaginative interface and design to help prepared investigation of rich information. The mobile application was built with Java using Android Studio and SQL has been used for the backend. The application underpins the distribution of new data in the backend: campus toilet, etc., can be easily published which will dynamically reflect on the mobile application. In these respects, NEU Places uses technologies that provide an effective information exploration experience. The primary novel features in the proposed app that makes it different from other apps are:

- Connection of personal calendars to geo-location on campus for services
- Scheduling and displaying students and staff their daily lectures and meetings locations on the Google map
- Language translator

After critical and rigorous analysis on the response time of the existing similar apps, it is recorded that the existing app have high response time and I am able to pick this limitation with the following novelties:

a) **Implementation of Splash Screen:** Introducing the welcome screen activity (which is also known as the mobile app splash activity) whenever the application is running is one of the steps to help improve the experience of the mobile user by masking the genuine application response. I have implemented a splash screen which shows up when the app is launched. This feature is not available in some of the existing mobile apps.

b) **Removal of unnecessary data and files**: Too much of data/files on an app take a very long time to download. Response time of the proposed mobile app has been fully optimized by diminishing the size of a payload sent from the server to the cell phone; the time a user spends holding up can be decreased. The data that users do not need has been removed so as to reduce the app response time.

c) **Using Google Map**: The map that has been used to display location on the app is the Google Map. I have choose to use this feature for embedding map on the app because of its efficiency and low consumption of phone memory which results in a reduction of the app response time.

Sluggish response time can abate adoption and usage can lower the value of the app and any ensuing return on investment (ROI). Faster response time can enhance cellular achievement both in phrases of worker and client adoption.

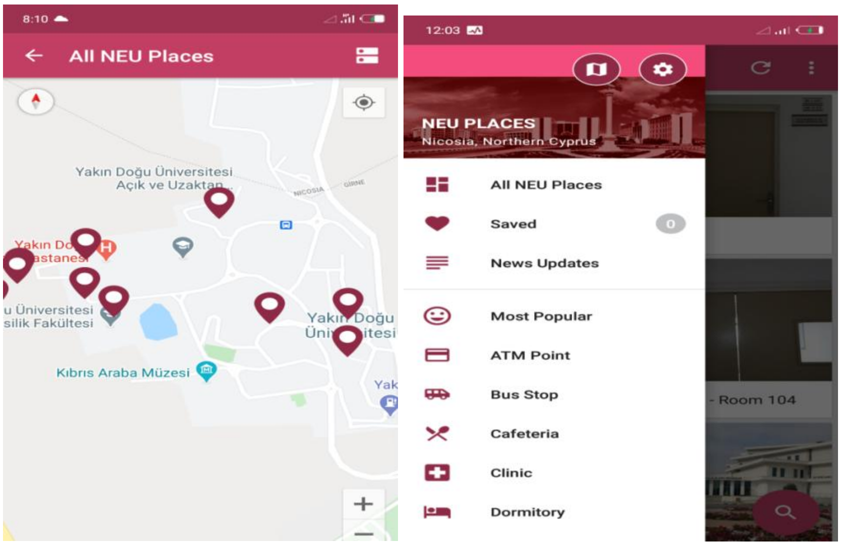


Figure 1.1: Different features on the app User Interface

The rest of this thesis is prepared as follows. Section 2 presents brief background and descriptions associated research. Section 3 describes the proposed structure and architecture. In section 4, we offer the user interface of our system in details. The implementation details of our prototype in section five and a performance evaluation is supplied in section 6. Lastly, section 7 attracts some concluding remarks.

CHAPTER 2

BACKGROUND AND RELATED RESEARCH

Internet of Things (IoT) called as another model correspondence framework, where various sorts of items, for example, cell phones, sensors, or gadgets are communicating with one another over the web. As the human portability turns into a significant perspective in many smart-city applications, information from cell phones has been given a lot of attention (Al-Turjman, 2016). IoT encourages a wide assortment of heterogeneous gadgets which are to be associate, convey and being produce a huge measure of information. The technology is advancing in such a way that everything is now getting connected to the internet which enables people to easily communicate and share information with no stress. (Malchi, 2021). Capacity the board of IoT information is a fundamental part of shrewd information stockpiling IoT-applications.

These days, urban communities face complex difficulties to improve their residents' personal satisfaction. Because of metropolitan fixation, individuals' everyday environments have been affected by expanded gridlocks, carbon dioxide, ozone harming substance outflows, and garbage removal. The idea of keen city has developed famous in the course of recent years. It accepts a few measurements relying upon the importance of "smart" and advantages from creative utilizations of new sorts of data and correspondences innovation to help public sharing (Ullah, 2021). The possibility of the digital/smart city is a response to all the arising issues. Numerous urban communities characterize themselves as "brilliant" when they recognize a portion attributes as being so, (for example, broadband network, advanced incorporation, and information labor force). Enhancement plan is utilized as the activities and procedures to be followed during the plan and activity of correspondence organizations, along these lines contributing decency and throughput. With the initiation of the 5G remote correspondence framework, a completely associated and grounded society is said to persevere, give different kinds of business structures, area ability, and administrations, associated with different sorts of traffic designs and extraordinary requirements (Kumar, 2020).

2.1 mSpace Mobile: A Mobile Application for Location

This is a mobile application that was developed for the same purpose but this mobile was developed for Windows phone. This Mobile application (Fig. 2) joins a creative interface and design to help prepared investigation of rich data spaces. As opposed to compelling iterative, discrete watchword questions, it keeps up the setting of related data when it is in support of the fast investigation of a wide grouping of related sources. This app uses the Software Framework of mSpace (Harris, 2004), that bolsters the cooperative effort together with the portrayal of disseminated related assets. Additionally, this mobile app underpins thin new data distribution: remarks and suggestions might be distributed in any of the component found with the help of the interface. With this, the app utilization advances gives a successful data investigation experience while additionally adding to the development of information stores for the Web.

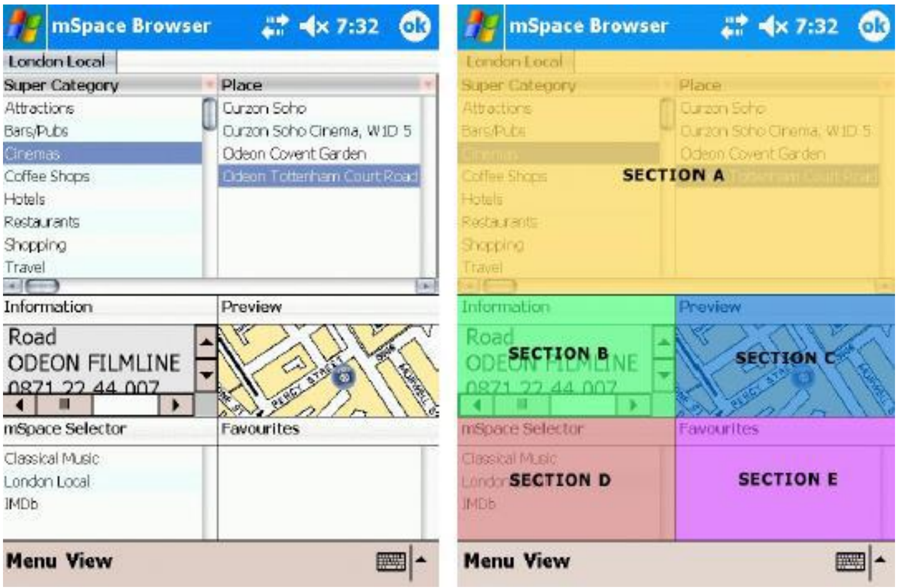


Figure 2.1: There are five highlights inside the UI: A – columnar of the app browser; B – box containing information; C – the cup map preview; D – the app selector; E – the list of favourites. (Max and Alisdair, 2014)

The paper describes the application as far as its usefulness, UI and supporting design. It has three core features which are i) the custom small-screen specific interface (Fig. 2.1) to help connection model of the mobile application (Schraefel, Karam, & Zhao, 2003) for logical investigation of huge data spaces, ii) it supports the cordination of distributed web resources and, iii) it creates awareness when you are in a particular location. The interface of the app is designed to allow the application users of little screen mobile app run complex queries through direct control and composing a text is not needed. This mobile app uses the architecture which is built on three-layer.

2.2 mSpace Mobile System Architecture

The engineering portrayed underneath is intended to question different triplestores. The portable application permits the client to choose their wellspring of data through the framework interface. mSpace Mobile framework abstracts the inner ideas of inquiry age and triplestore questioning to disseminate the framework, removing load from lowpower cell phones. Three-layer design has been created which are the mobile application itself (MA), the mobile app Service for Query (MQ), and the triplestore information which is associated with the Resource Description Framework (MK).

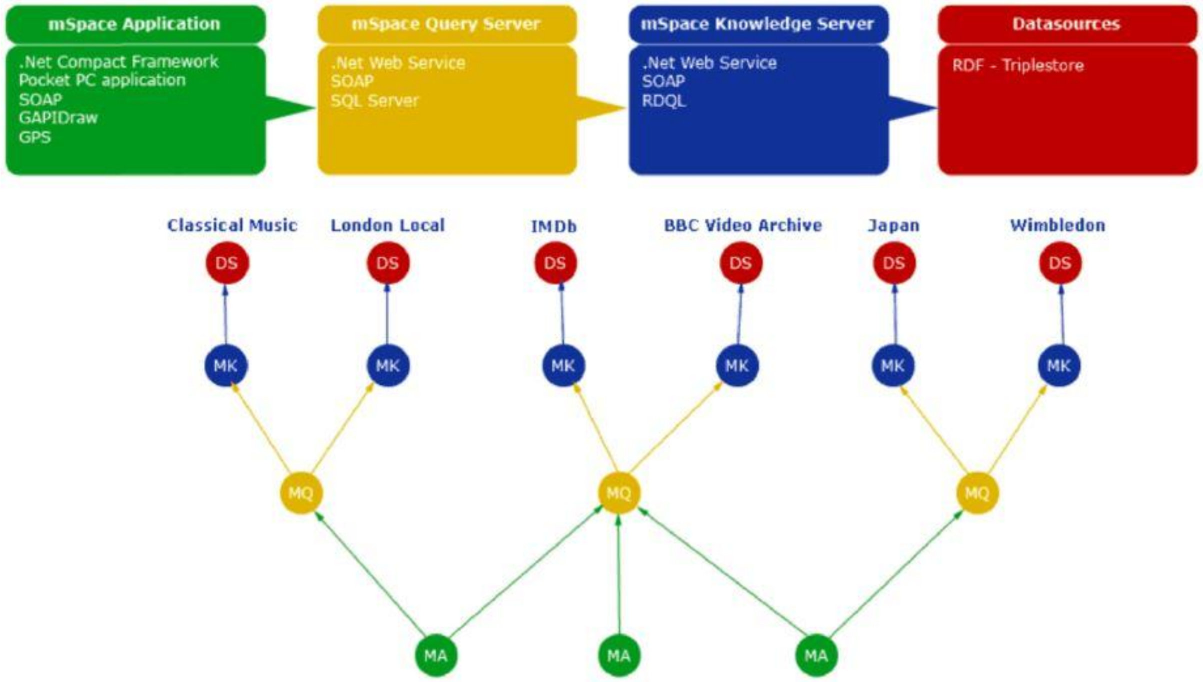


Figure 2.2: mSpace Mobile architecture

The triplestore information which is associated with the Resource Description Framework makes it easy to utilize and deal with information arrangement: to have numerous suppliers that stay controlled independently. The methodology likewise bolsters other Web highlights like creator connecting of assets, simplicity of making of new information, straightforward dissemination and client picked the sources of information. To permit customer programming to be composed for various stages, and to encourage the utilization of the mobile app questioning design, one of most idea of the user is depicted in the engineering as a conventional mobile app (MA). The reasons why the Service Query has been implemented is to encourage maximal similarity among conceivable mobile apps; empower mobile apps to associate with this help effectively, and to keeps the intricacy of the convention low, two explicit access strategies, are utilized for the correspondence between the MA and the MQ. The MA discusses at first with the MQ, which goes about as a representative (or, adequately, a disseminated connect base), with information on spaces and of important mSpace Knowledge Services (MKs). The MA idea is planned with the end goal that the application can take any shape that can execute the correspondence convention. In this construction, the MKs speak with the MA after meeting with a MQ. A MK is what might be compared to the triplestore in the current mSpace usage, and on account of beginning executions, the MK imparts by means of HTTP-RDQL with a 3store triplestore. The quick motivation behind the MK, when found by the MQ as a fitting storehouse, is hence to develop a RDQL inquiry utilizing obliging triples that are indicated by the MA. Likewise, on the off chance that one needed to get data about a spot to eat, they may visit a universally useful neighborhood data provider, (for example, Thomson Local) or an area explicit café control. The equivalent applies for the MQ model; the decision of MQ decides the substance. This is comparable to the capacity to pick what joins you have on your WWW site, making the data applicable, just as forestalling a data over-burden situation. The way in to this is that the customer questions the MQ and all things considered, the customer is just mindful of the connections given by the MQ.

Another approach is the approach that coordinates Markovian interaction with IEEE 802.16 to dissect different QoS measurements, in particular throughput, mistake rate, and the normal bundle delay. In brilliant urban areas, savvy transportation frameworks, cell organization, remote sensor organization, device-to-device (D2D), and large numbers of these keen standards include the remote versatile mixed media correspondences as well as transmissions. Specifically, the device-to-device D2D has gotten expanded consideration from remote cell networks since it can recognizably off-load the organization traffic and diminish the transmission energy since the conveying gadgets are in closeness of one another (Ali and Ahmed, 2018). This nearness makes it a prominent systems administration design for relay-based transmission utilized by energy-constrained gadgets, for example, cell phones, tablets, and individual advanced associates. Thusly, it is viewed as essential to improve the energy proficiency of device-to-device D2D organizations while keeping up their dependability regarding the accomplished throughput and inactivity. The vast majority of the current works have considered the immediate connection circumstance for the corruption of following reasons, in particular, significant distance correspondence,(Ali and Ahmed, 2018) poor propagation,(Baro and Escalera, 2009) and interference.(Beek and Sandell, 1997)

As relay-based transmission can improve the longer distance device-to-device D2D communication, most of the researchers have given their attention for the multihopping D2D communication. Al-Turjman and Abdulkadir(Al-Turjman and Abdulkadir, 2018) managed spatial-density and force transmission to improve bandwidth. Be that as it may, it is pronounced without indicating a steering strategy. Al-Turjman et al(Al-Turjman and Hassanein, 2011) proposed a multihop directing procedure to expand the jump check of the systems administration frameworks. Notwithstanding, it is then demonstrated inconsistent to limit the distance between the clients. Since the current relay-based procedures don't consider the energy productivity for the D2D correspondence, there is a need to form a combinatorial streamlining that improves the energy proficiency of such correspondence frameworks. Appropriately, innovative headways are adjusting the demonstration between transmission capacity use and time delay in any information conveyance approach. Likewise, low intricacy with high unwavering quality correspondence comes as a first key need other than energy productivity, idleness, and security. (Al-Turjman and Abdulkadir, 2018).

In this work, we aim at proposing A Framework for Schools Virtual Tour via the Internet of Things (Iot) Mobile Services. We propose a novel design for the mobile-based location app that provides information about topics of chosen interest like the school library, cafeteria, faculty, lecture rooms, clinic, bus stations, ATM points and many more. Information will be displayed on the platform based upon the location as determined by the information available on the mobile application. It provides advanced mapping systems that will display different places based on the interest of app user. It would be easy to search nearby place and navigate in a user friendly way. This mobile application is developed to reduce stress for new students and staffs who find themselves on campus. This framework utilizes the latest channel coding techniques while utilizing existing cellular infrastructure to easily display a particular location with the use of Google map.

2.3 mSpace Tests Execution and Security

We defined four test cases for each task: access an element of an array, iterate through an array, iterate through a matrix and calculation of Fibonacci number (Andonoska and Jakimoski, 2018). For the test cases that require input parameters we will define three ranges, small, medium and large. For the small range, an arbitrary and small enough value is taken that contributes to changing the utilization of the memory and CPU. For the highest range, a rough upper boundary value is taken, after which the device has insufficient resources to perform the action. The medium range is defined between the upper and lower boundaries. The maximum values are different for devices with different hardware configuration. The following values were defined (Andonoska and Jakimoski, 2018):

a) **Iterate through array:** small: 1000; medium: 1,000,000; large: 40,000,000 (Xiaomi), 30,000,000 (Samsung) and 20,000,000 (LG)

b) **Iterate through matrix:** small: 100; medium: 1000; large: 6000 (Xiaomi), 5000 (Samsung) and 4000 (LG)

c) **Fibonacci number:** small: 10; medium: 30; large: 40 (Xiaomi), 40 (Samsung) and 35 (LG).

Digital Physical Systems are coordinating the systems administration, registering, and actual cycles, where current sensors handle its significant parts proficiently, for example, a digital framework and an actual interaction (Deebak, 2021). However, as the interaction between these two components increases, then the physical systems is more prone to the security vulnerabilities. To handle these vulnerabilities, Blockchain is breakthrough technologies, which secure and execute transactions in an open network environment without the involvement of any centralized third-party system. The changes in industrial production associated with Industry 4.0 are driven by some important advances in technology, two of the most important of which are big data and the IoT, which facilitate the sharing of information and improve the flexibility of the production chain (Al-Turjman et. al, 2020). IoT networks dedicated to smart-cities perform critical tasks such as generatingencryption/decryption keys (Kadjouh et. al, 2020).

CHAPTER 3

SYSTEM ARCHITECTURE

NEU Places mobile application has been built with Java using Android Studio and SQL. The project is as simple as possible for better understanding. It contains few JAVA and XML files. The app has been built on Google standard Google map to ensure highest accuracy and portability and has a dynamic backend based on SQL, this will help the administrator to update and maintain backend data without any additional server or database.

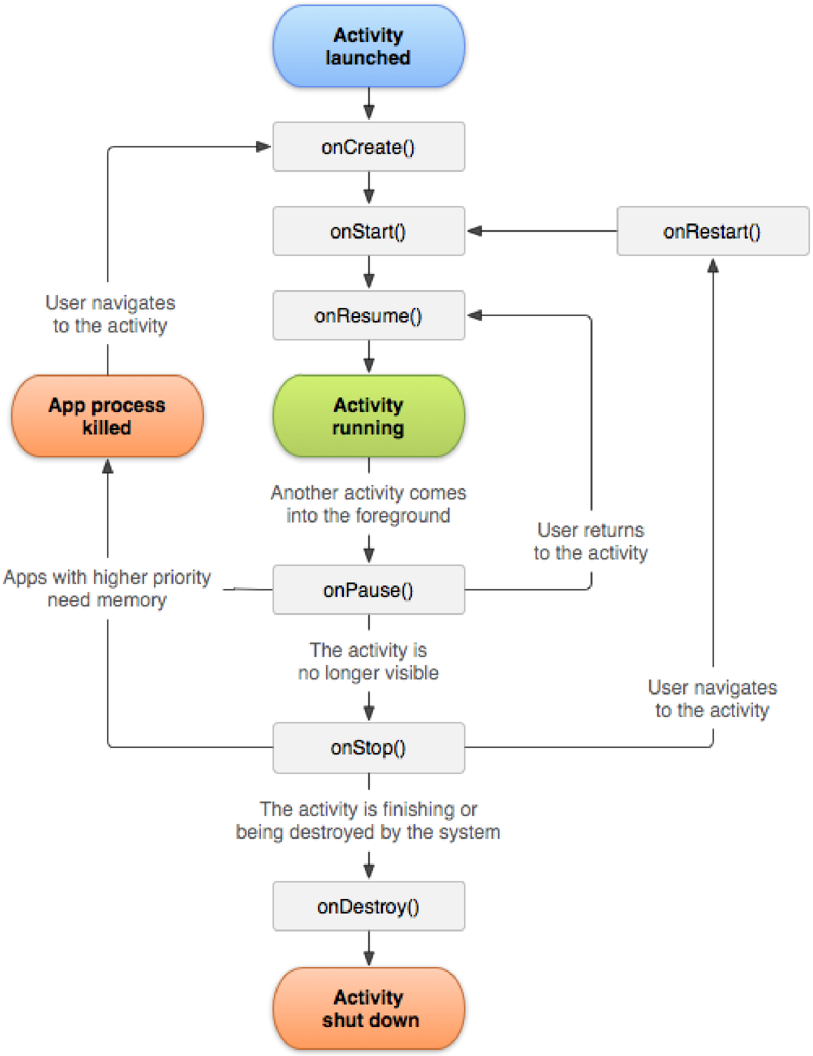

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Figure 3.1: Android mobile application activities lifecycle (which for retrieval in the reference is the name of the user who posted the article)

There are three key loops to use when within activities:

a) ***The entire lifetime of an activity:*** this occurs between the main call to onCreate(Bundle) through to a solitary last call to onDestroy(). A movement will do all arrangement of "worldwide" state in onCreate(), and delivery all excess assets in onDestroy(). For instance, in the event that it has a string running out of sight to download information from the organization, it might make that string in onCreate() and afterward stop the string in onDestroy().

b) ***The visible lifetime of an activity***: it occurs between a call to onStart() until a comparing call to onStop(). During this time the client can see the movement on-screen, however it may not be in the closer view and collaborating with the client. Between these two techniques you can keep up assets that are expected to show the action to the client. For instance, you can enlist a BroadcastReceiver in onStart() to screen for changes that sway your UI, and unregister it in onStop() when the client no longer sees what you are showing. The onStart() and onStop() strategies can be called on different occasions, as the movement gets noticeable and covered up to the client.

c) ***The foreground lifetime of an activity:*** occurs between a call to onResume() until a comparing call to onPause(). During this time the action is in noticeable, dynamic and associating with the client. An action can as often as possible go between the continued and stopped states - for instance when the gadget rests, when a movement result is conveyed, when another aim is conveyed, so the code in these strategies ought to be genuinely lightweight.

The whole lifecycle of an action is characterized by the accompanying Activity techniques. These are snares that we abrogate to accomplish suitable work when the action changes state. All exercises will execute onCreate(Bundle) to do their underlying arrangement; many will likewise actualize onPause() to submit changes to information and get ready to stop communicating with the client, and onStop() to deal with done being obvious on screen.

3.1. Material Design and Optimization

The first thing a developer should consider when working on a project is to design the project structure including the UI/UX design for the system. Material Design is a visual language that integrates the exemplary standards of good plan with the advancement of innovation and science. It is very important to have a presentable user interface by using a well pleasing design. (which for retrieval in the reference is the name of the user who posted the article, Material Design)

To recognize a bunch of UI segment classifications that much of the time happen in Android applications, we referred to famous plan apparatuses and dialects that uncover segment libraries has been referenced such as Balsamiq (Balsamiq, 2018) and Google's Material Design. (Call-Em-All, 2018). The material design has been maintained for Location Finder mobile application to decorate the user interface and to ensure the app is mobile friendly and fully optimized. All basic components are nicely decorated with unique looking and gorgeous colour combination. Optimization of code is essential to make the app run smoothly without lagging and that is why app has been developed with the code fully optimized. Every single implementation has been optimized for highest performance. Also, the code has been beautifully crafted and modularized to enable other developers to easily understand the code. Comments are used where necessary to describe certain line of code.

3.2. Android Studio setup

This is a free software built purposely to develop Android apps. After installing this software, the next thing is to download the necessary plugins including Java IDE and SDK. The next step is to launch the app and start a new project as illustrated below:

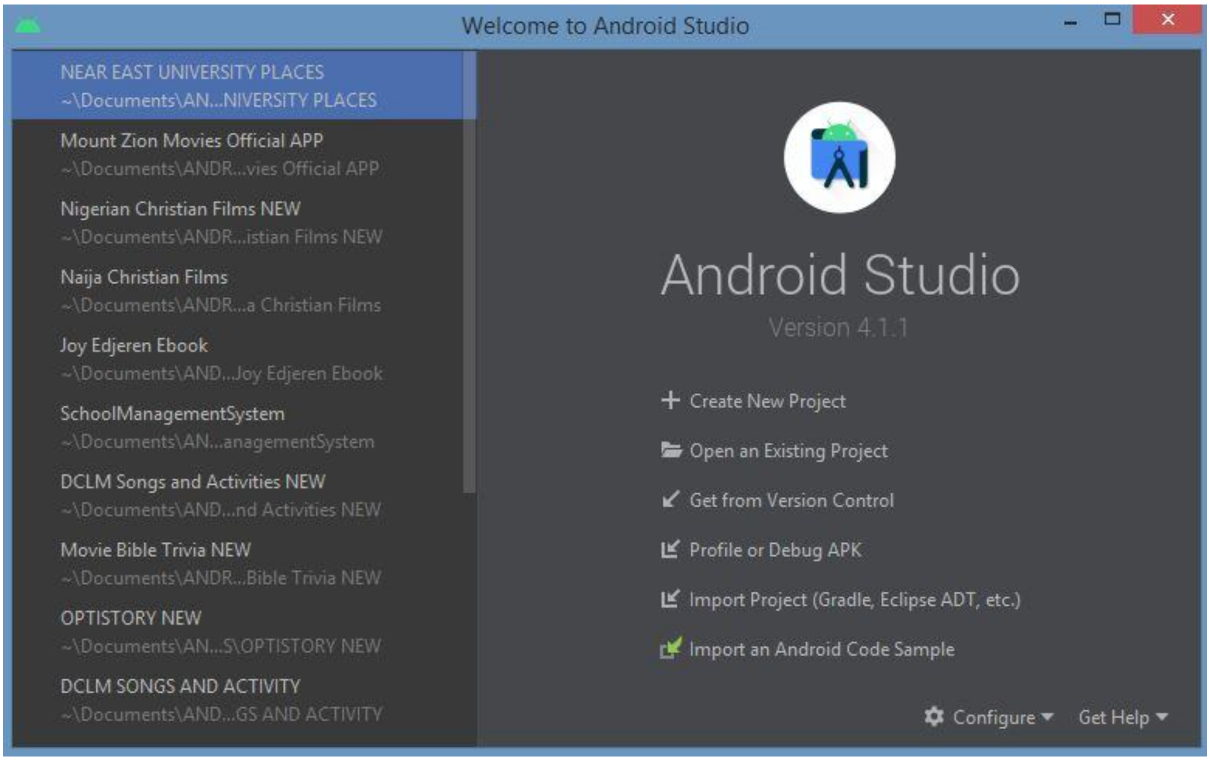
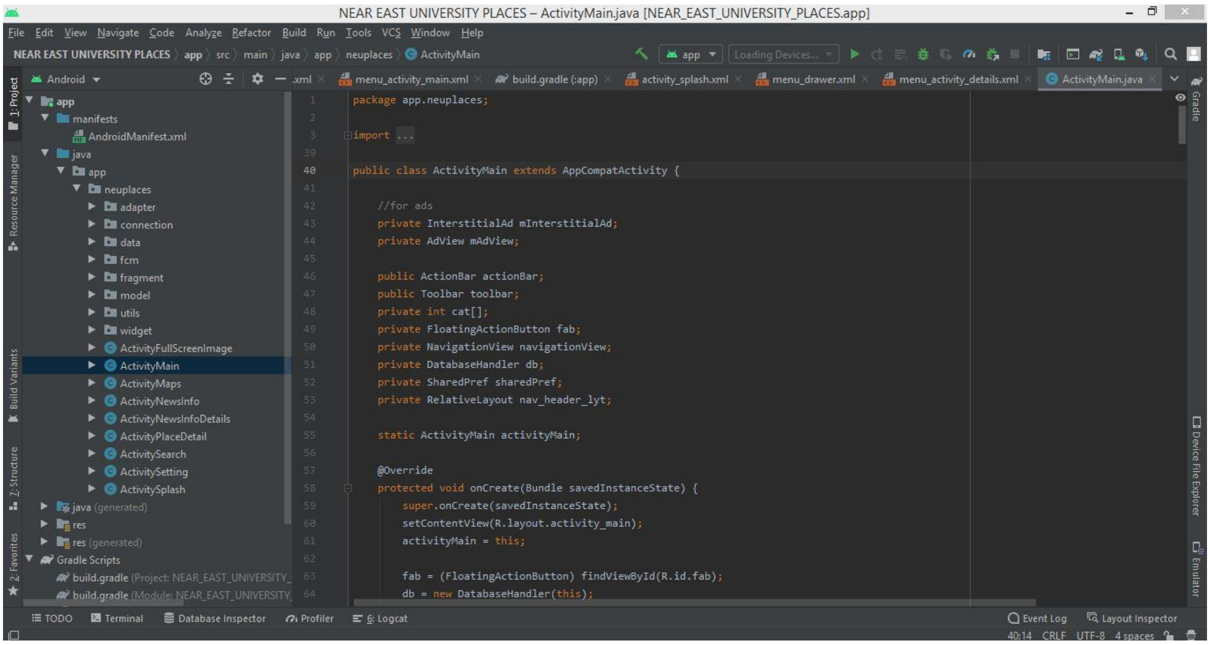

Welcome to Android Studio
NEAR EAST UNIVERSITY PLACES
~\Documents\AN...NIVERSITY PLACES
Mount Zion Movies Official APP
~\Documents\ANDR...vies Official APP
Nigerian Christian Films NEW
~\Documents\ANDR...istian Films NEW
Naija Christian Films
~\Documents\ANDR...a Christian Films
Joy Edjeren Ebook
~\Documents\AND...Joy Edjeren Ebook
SchoolManagementSystem
~\Documents\AN...anagementSystem
DCLM Songs and Activities NEW
~\Documents\AND...nd Activities NEW
Movie Bible Trivia NEW
~\Documents\ANDR...Bible Trivia NEW
OPTISTORY NEW
~\Documents\AN...S\OPTISTORY NEW
DCLM SONGS AND ACTIVITY
~\Documents\AND...GS AND ACTIVITY
Android Studio
Version 4.1.1
+ Create New Project
Open an Existing Project
Get from Version Control
Profile or Debug APK
Import Project (Gradle, Eclipse ADT, etc.)
Import an Android Code Sample
Configure
Get Help

Figure 3.2: Android Studio Interface

If a project has been created already, it will be listed at the left corner of the screen and the project can be easily selected. In this case, the project folder is named NEAR EAST

UNIVERSITY PLACES. Once that has been selected, the project will be synced by Gradle and the screenshot below will be displayed.



```java
package app.neuplaces;

import ...

public class ActivityMain extends AppCompatActivity {

    //for ads
    private InterstitialAd mInterstitialAd;
    private AdView mAdView;

    public ActionBar actionBar;
    public Toolbar toolbar;
    private int cat[];
    private FloatingActionButton fab;
    private NavigationView navigationView;
    private DatabaseHandler db;
    private SharedPref sharedPref;
    private RelativeLayout nav_header_lyt;

    static ActivityMain activityMain;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        activityMain = this;

        fab = (FloatingActionButton) findViewById(R.id.fab);
        db = new DatabaseHandler(this);
```

Figure 3.3: Android Studio Interface after opening a project

The files in the folder contain the Java and XML files.

3.3. Backend customization

The MySQL design depicts how the various parts of a MySQL framework identify with each other. The MySQL engineering is essentially a client – server framework. MySQL information base server is the server and the applications which are associating with MySQL data set server are clients.

An internet connection and a centralized database (MySQL) are needed for NEU Places mobile app. This app represents client/server architecture. Apps developed for Android devices serve as client while the combination of MySQL database and PHP APIs scripts forms the server. PHP APIs script acts as a link between the Mobile application and the MySQL database. Figure 3.4 shows the structure of this application. The main functions of PHP API are to do the following:

- Accepting the read and write requests from the clients.
- Run and manipulate the request to MySQL database.
- Formatting the output as JavaScript Object Notation (JSON).

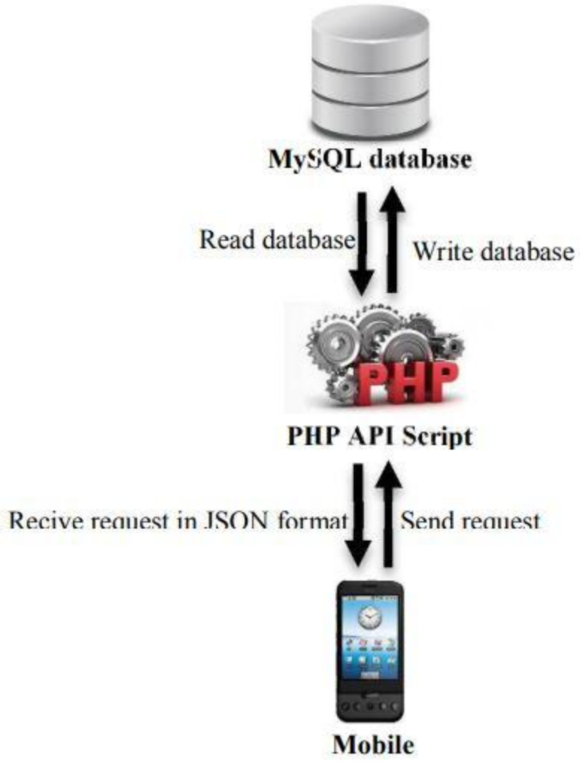
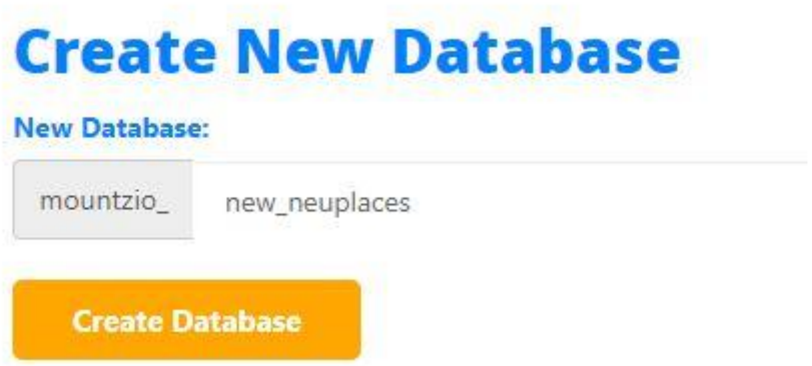


Figure 3.4: The structure of NEU Places application

Operations that are applied on the map: The developed application presents some operations that has been created to make some interaction with users. These operations are:

- Add a new location marker on the map.
- Delete a location marker from the map.
- Display pre-specific location information.
- Draw a path between two locations.

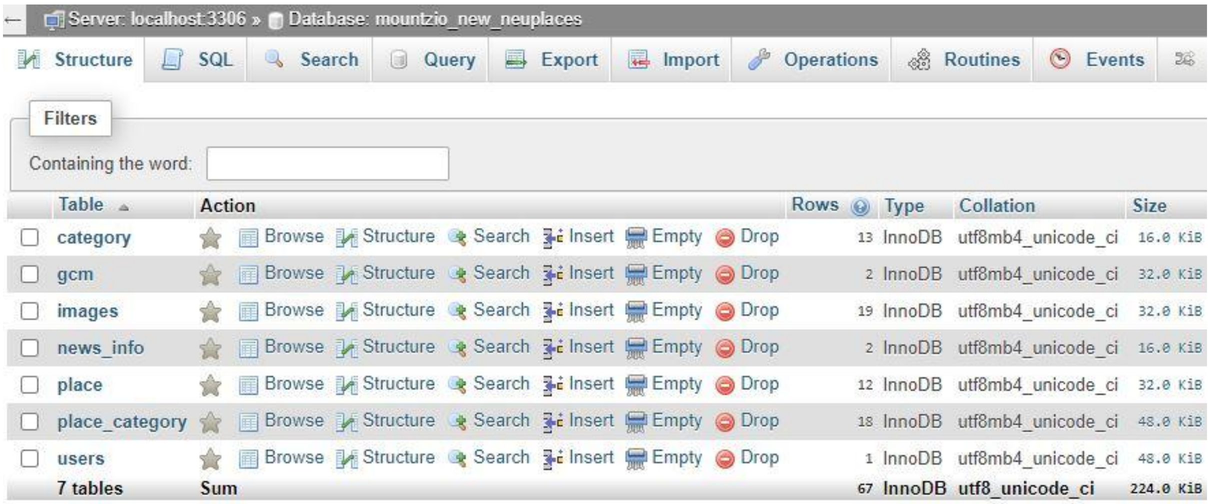
This is where we add the data that will appear on NEU Places application dynamically. We start by creating the database as shown below:


Create New Database

New Database: mountzio_ new_neuplaces

Figure 3.5: Creating SQL Database for NEU Places app

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modelling techniques. The database management system (DBMS) is the software that interacts with end users, applications, and the database itself to capture and analyze the data. Database is needed in this project for data storage. After creating the database, user is also created to enable permissions for reading, writing and so on.Once we are done creating the database and user, we open phpMyAdmin dashboard to create tables.



Server: localhost:3306 » Database: mountzio_new_neuplaces

Containing the word:

	Table	Action	Rows	Type	Collation	Size
□	category	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8mb4_unicode_ci	16.0 KiB
□	gcm	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	32.0 KiB
□	images	Browse Structure Search Insert Empty Drop	19	InnoDB	utf8mb4_unicode_ci	32.0 KiB
□	news_info	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	16.0 KiB
□	place	Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_unicode_ci	32.0 KiB
□	place_category	Browse Structure Search Insert Empty Drop	18	InnoDB	utf8mb4_unicode_ci	48.0 KiB
□	users	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_unicode_ci	48.0 KiB
	7 tables	Sum	67	InnoDB	utf8_unicode_ci	224.0 KiB

Figure 3.6: Adding tables to NEU Places database

We have created seven tables which are category, gcm, images, news_info, place, place_category and users.

a) ***Category:*** under this section, we have each category of places on campus, places like International Office, Cafeteria, and Dormitory and so on.

b) ***Gcm:*** this is the notification table; this table has been created to track the number of registered users which receives notification when a new location has been added to the app.

c) ***Images:*** this is where the images are stored

d) ***News_info:*** a table that stores the latest school news

e) ***Place:***this is where subcategories are stored like Yakın Doğu Üniversitesi Tıp Fakültesi, International Student Office and so on.

f) ***Place_category:***place identity and category ID are stored here

g) ***Users:***this table takes the administrator details

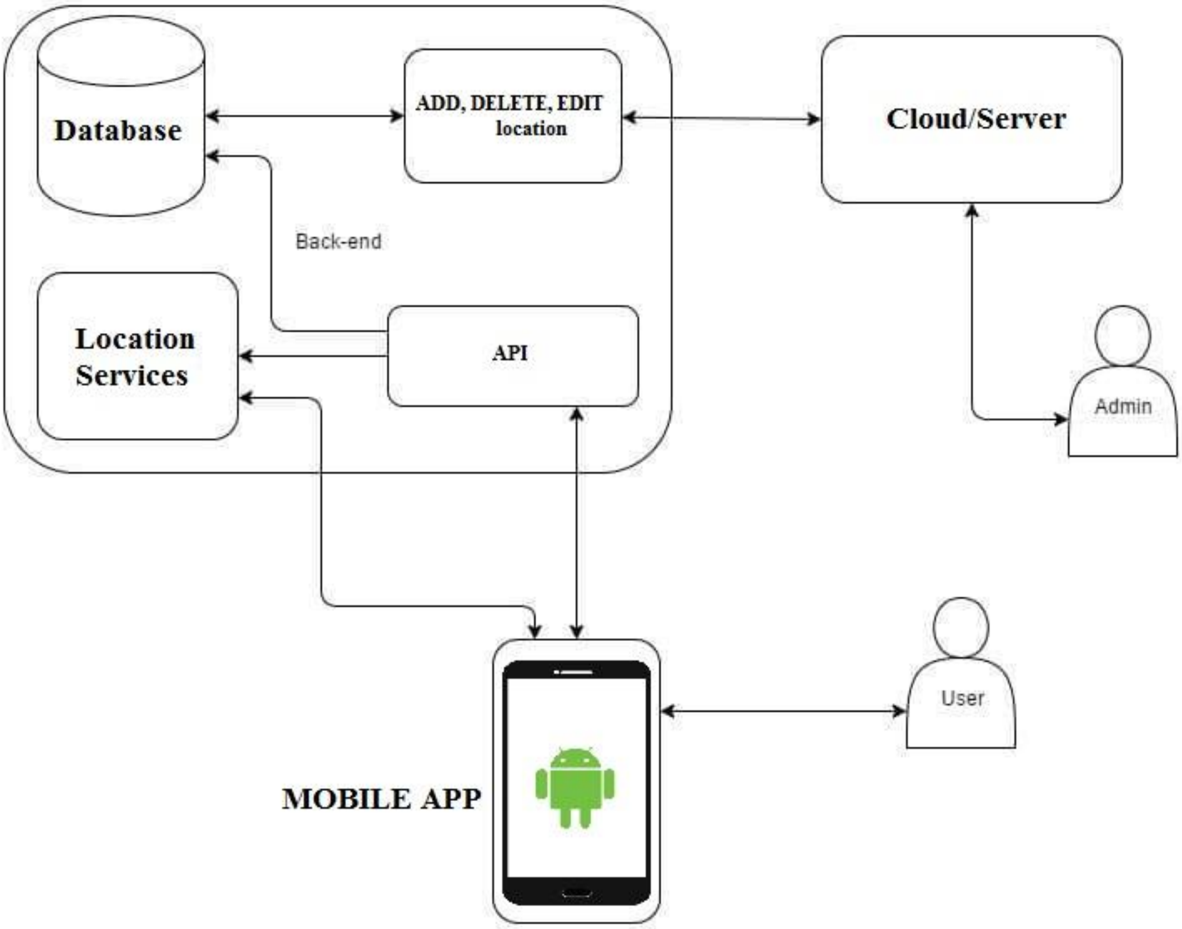
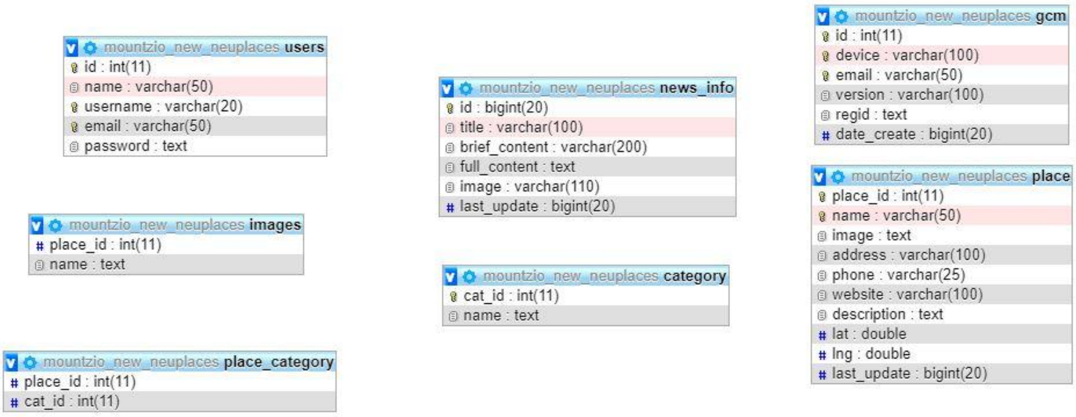


Figure 3.7: Propose System Architecture

Fig 3.7 shows the architecture of our system. We have developed a prototype to test the fundamental functionality of our mobile app. The prototype implementsthe context and data management on the server side while the mobile user interface resides on the mobile device. The user interface is implemented using the latest Android SDK. The server components are responsible for managing the database and dispatching relevant applications along with recommended actions.



mountzio_new_neuplaces users
id : int(11)
name : varchar(50)
username : varchar(20)
email : varchar(50)
password : text

mountzio_new_neuplaces images
place_id : int(11)
name : text

mountzio_new_neuplaces place_category
place_id : int(11)
cat_id : int(11)

mountzio_new_neuplaces news_info
id : bigint(20)
title : varchar(100)
brief_content : varchar(200)
full_content : text
image : varchar(110)
last_update : bigint(20)

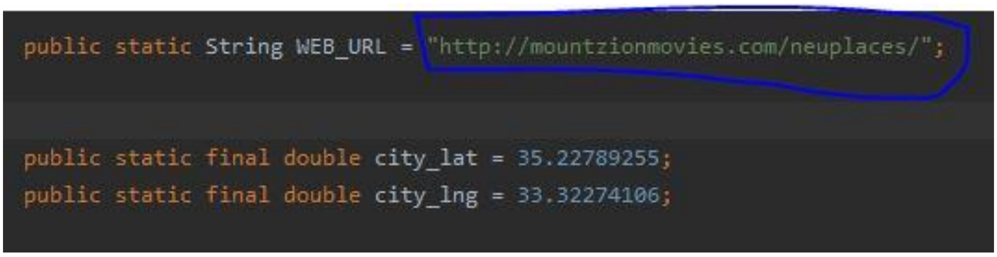
mountzio_new_neuplaces category
cat_id : int(11)
name : text

mountzio_new_neuplaces gcm
id : int(11)
device : varchar(100)
email : varchar(50)
version : varchar(100)
regid : text
date_create : bigint(20)

mountzio_new_neuplaces place
place_id : int(11)
name : varchar(50)
image : text
address : varchar(100)
phone : varchar(25)
website : varchar(100)
description : text
lat : double
lng : double
last_update : bigint(20)

Figure 3.8: The database architecture

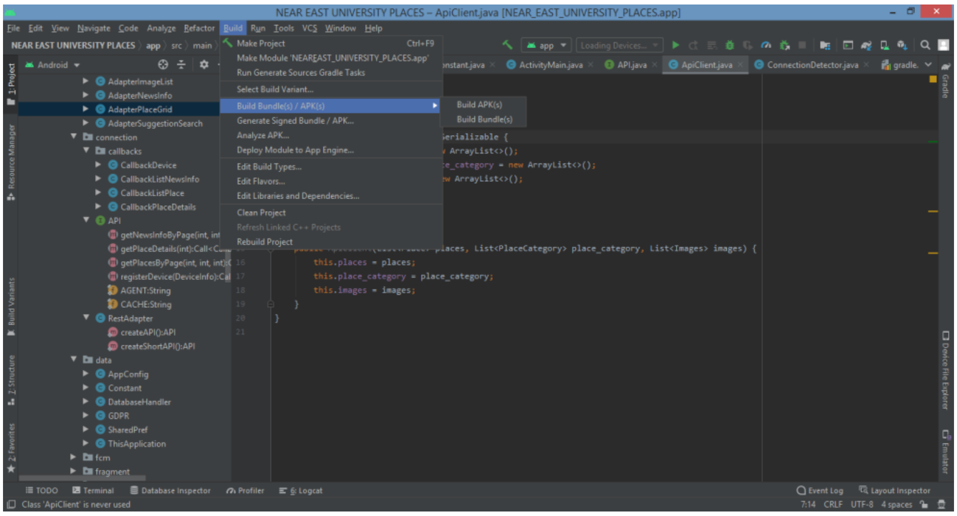
Above is the structure of the database and the tables with the values that has been used for the creation of the database. After creating the database, the server URL will be copied and this is what will be used in Android Studio to display information on the app. A Java class has been created in Android Studio called Constant.java. This class takes the parameters of the SQL database tables. The code is given below:



```
public static String WEB_URL = "http://mountzionmovies.com/neuplaces/";

public static final double city_lat = 35.22789255;
public static final double city_lng = 33.32274106;
```

Figure 3.9: Linking the SQL to NEU Places app

After adding this project id to Android Studio, the apk file is generated by selecting Build > Generate Signed Bundle/APK, as illustrated below:



NEAR EAST UNIVERSITY PLACES – ApiClient.java [NEAR_EAST_UNIVERSITY_PLACES.app]

```
        places, List<PlaceCategory> place_category, List<Images> images) {
            this.places = places;
            this.place_category = place_category;
            this.images = images;
        }
    }
```

Figure 3.10: Generating the .apk file

After generating the apk file, the app can be either installed on any Android device or published on Google Play Store for people to install.

Having explained the architecture design of NEU Places application, we can now move to the user interface.

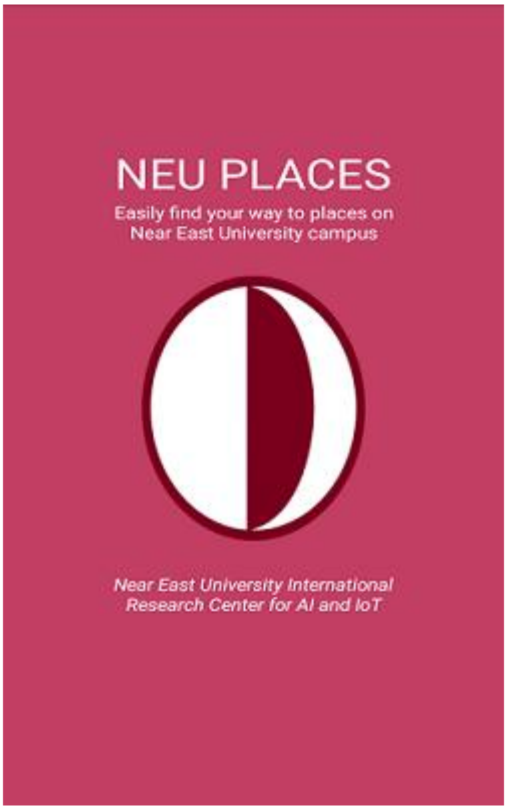
CHAPTER 4

USER INTERFACE

The NEU Places app interface is designed to allow users of small screen devices run a search for a particular query like “Dormitory, Cafeteria and so on”. To this end, the application utilises the primary features of the app interaction model. The features of NEU Places app are:

- App will automatically detect user current location.
- The contact number of each business can be easily dialled for more information about a particular location.
- It allows user to easily navigate to desired location.
- List of locations are already created.
- Users can save an item or a particular location of interest for further exploration in future.
- Categories of different items (like names of available cafeteria) are created.

The screenshots of the user interface are listed below:

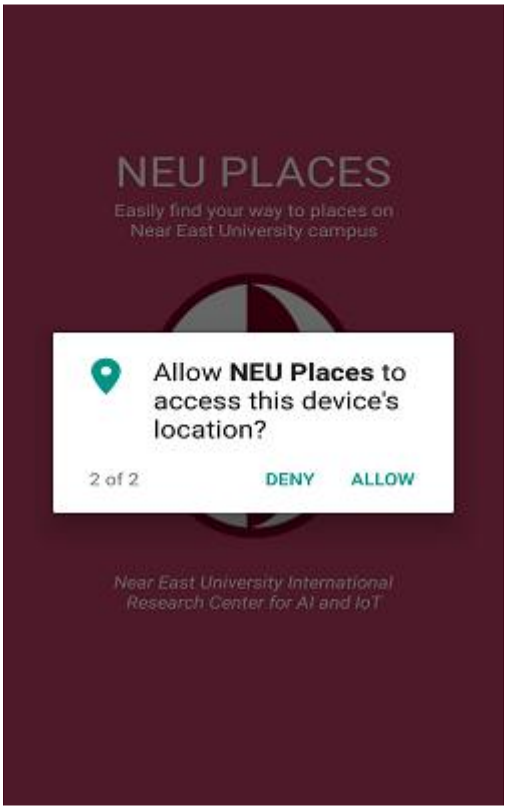


NEU PLACES

Easily find your way to places on Near East University campus

Near East University International Research Center for AI and IoT

A



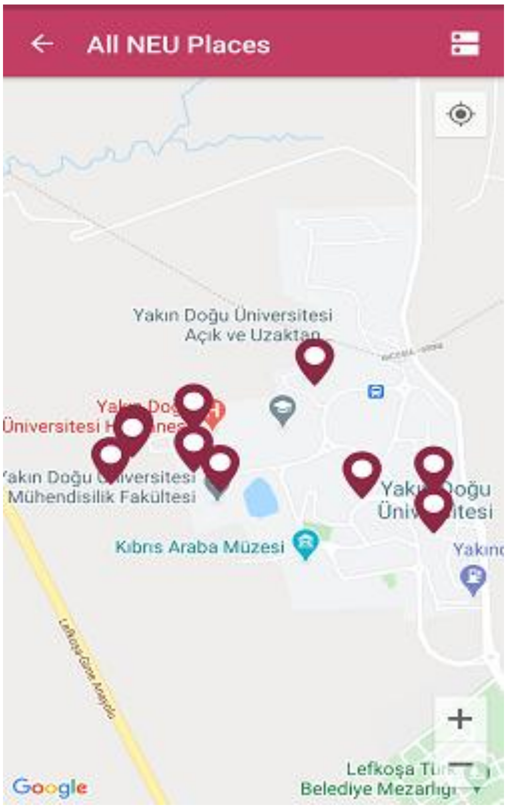
NEU PLACES

Easily find your way to places on Near East University campus

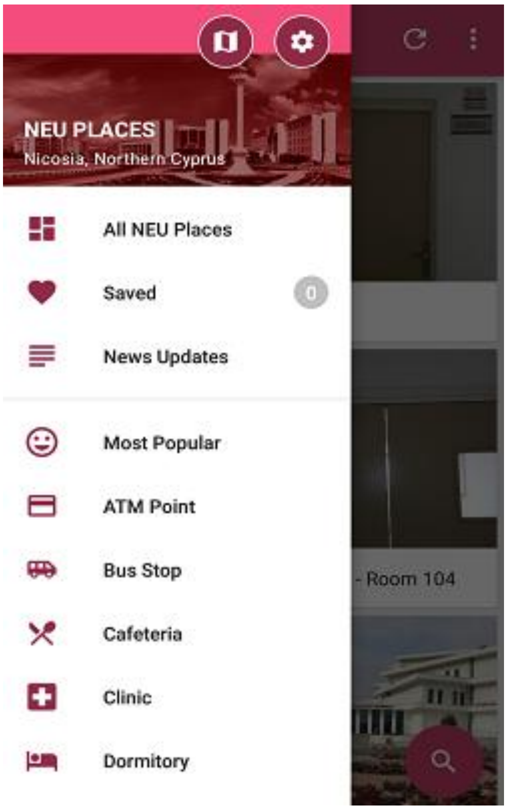
Allow **NEU Places** to access this device's location?

Near East University International Research Center for AI and IoT

B

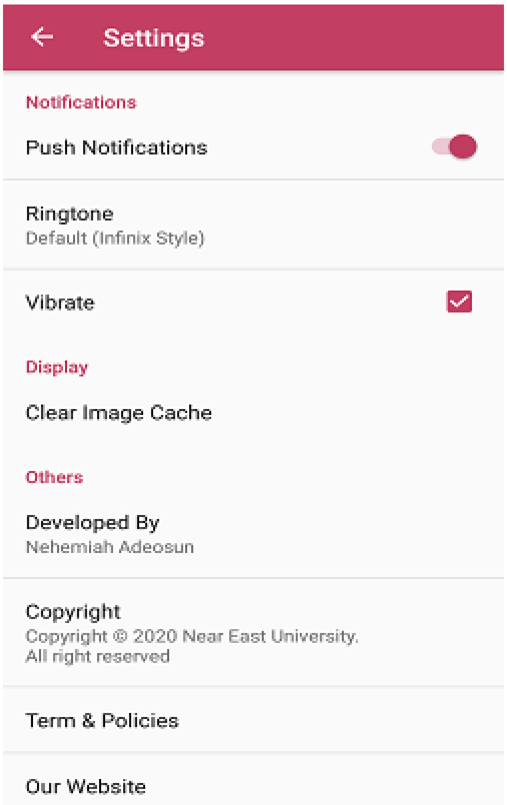

All NEU Places

C


NEU PLACES
Nicosia, Northern Cyprus

- All NEU Places
- Saved 0
- News Updates
- Most Popular
- ATM Point
- Bus Stop
- Cafeteria
- Clinic
- Dormitory

D


Settings

Notifications

Push Notifications ●

Ringtone
Default (Infinix Style)

Vibrate ☑

Display

Clear Image Cache

Others

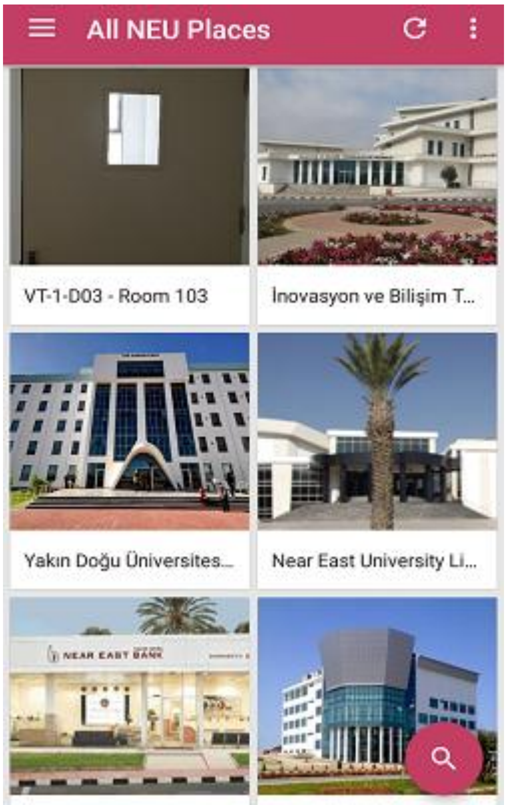
Developed By
Nehemiah Adeosun

Copyright
Copyright © 2020 Near East University.
All right reserved

Term & Policies

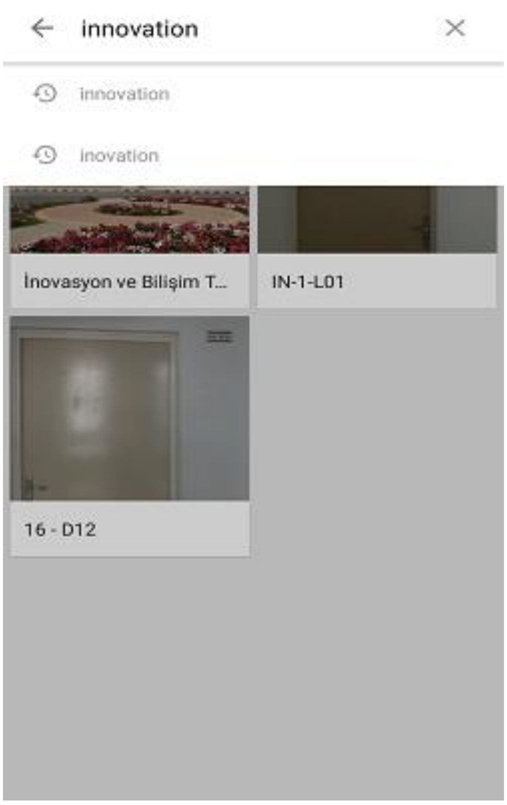
Our Website

E

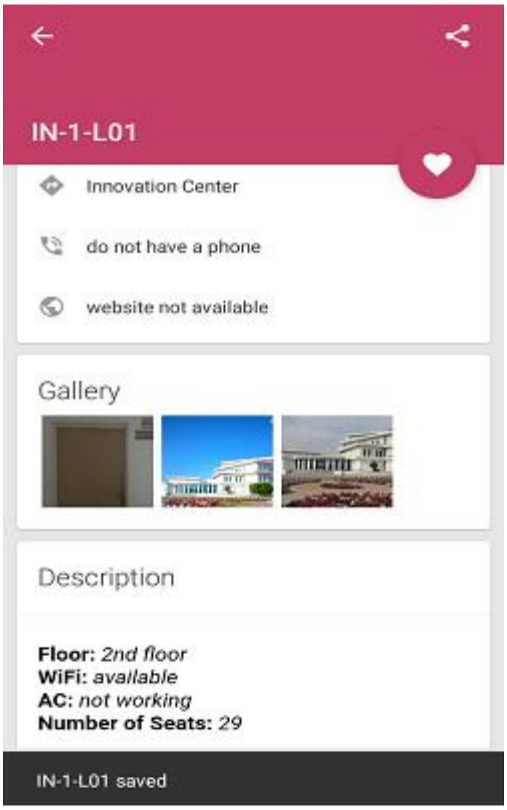

All NEU Places

- VT-1-D03 - Room 103
- İnovasyon ve Bilişim T...
- Yakın Doğu Üniversites...
- Near East University Li...

F



G


IN-1-L01

Innovation Center

do not have a phone

website not available

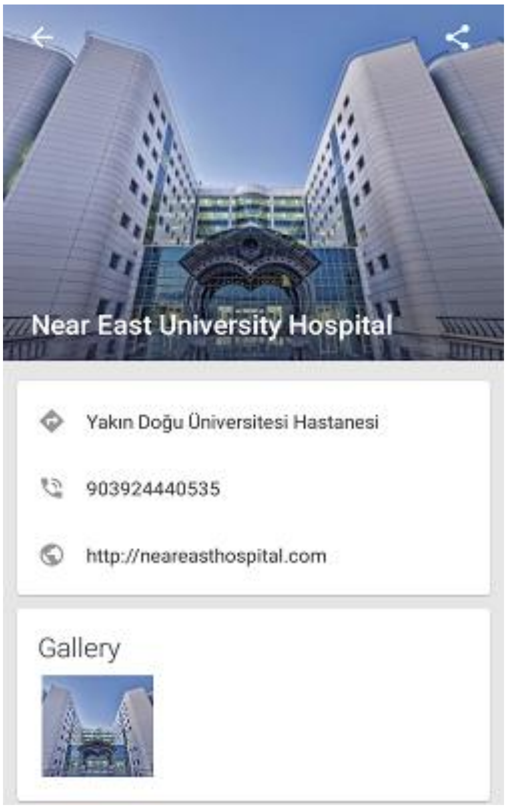
Gallery

Description

Floor: *2nd floor*
WiFi: *available*
AC: *not working*
Number of Seats: *29*

IN-1-L01 saved

H


Near East University Hospital

Yakın Doğu Üniversitesi Hastanesi

903924440535

http://neareasthospital.com

Gallery

I

Figure 4.1: Nine screens that explain how the app works. The user just open up NEU Places app and about to find a lecture room. A – The first screen that user will see is the splash screen activity with a beautifully designed logo with some description. B – Gives permission to app to access the user's current location. C – User's location has been detected. D – User navigates using the navigation bar. E – Settings screen. F – List of available places on campus. G – User trying to locate a lecture room called IN-1-L01 in the building of Innovation centre by using the search bar. H – A short description about the lecture room with the map is displayed. I – The campus main clinic is displayed.

Using NEU Places mobile app to locate a lecture room that is located in Innovation Center, within the campus (Fig. 14), the user at first open up NEU Places app and the first screen that shows up is the splash screen with the app logo and a short description (Fig. 14 A); the user then grant the app a permission to access the location, if not granted, app will not give the precise location (Fig. 14 B); user's location will be detected after granting the permission to access location (Fig. 14 C); then user sees the list of available places (Fig. 14 F); user then search for innovation centre which automatically displays the list of lecture rooms in this building with the help of the search algorithm (Fig. 14 G); IN-1-L01 is selected which shows up with a short description, phone number to call, website URL, and map (Fig. 14 H). The school hospital is displayed (Fig. 14 I).

4.1 Installation

The process of the app installation is planned to be direct and should not take long or any responsibility from user. This mobile app should initially be downloaded and installed from the Google Play store and requires under 10 MB of space. Immediately after installing the app, it can be launched for usage. It permits app users to see how the mobile application functions, approve suitable authorizations, and affirm that information assortment can start. Users will commonly be approached to give authorizations identified with the utilization of location. A window will also popup asking for the user permission to access current location, if the permission is granted by the user, the first activity will be displayed. If the app user did not grant the permission, the mobile app will automatically close.

4.2 Foreground operations

The data security and consent: Related conduct information and Location are measures which are sensitive, and consequently conventions should guarantee that the security and information of the user is protected when collecting user's data (James, Jankowska, Marx, & Laden, 2016). When the app is launched at first, users are given a concise data screen that indicates what data will be gathered and how to stop information assortment. This data and the terms of administration and security strategy can be reviewed whenever from inside the application. During the assortment stage, all area information are put away in a 256-bit SQL data set. This guarantees that regardless of whether the application source code was undermined, no information can be recovered in the absence of the first data. Just the portable application engineer can get to this data set.

4.3 Background operations

NEU Places mobile application relies on the ACCESS_NETWORK_STATE, ACCESS_FINE_LOCATION, READ_GSERVICES and ACCESS_COARSE_LOCATION permissions. This gives admittance to GPS, Wi-Fi, and organization investigation to recover scope, longitude, precision levels in meters estimated by a range of certainty. The application is viewed as high need, which implies that the most precise perusing accessible is given, paying little mind to battery use. The request for positivity of follow (comparable to precision) is consequently: GPS, at that point the wireless network, trailed by network examination (Canzian and Musolesi, 2015). Every five minutes, location update is requested by default. A lengthy table is returned and it is automatically stored in the SQL database.

4.4 Resilience of the NEU Places application

Seven potential ways that the background operations of the mobile app which can be prevented from functioning has been identified. Data collection can be stopped inadvertently by the app users with the following steps: (i) switching off their mobile device, (ii) existing or shutting down the app, (iii) stopping the foreground tasks which are running, (iv) forcing the shutting down of all active applications, (v) disabling the location services, (vi) activating the power saving modes, or (vii) uninstalling/removing the mobile app completely from the device. Additionally, if the frontal area segment of the application is shut, the foundation administration will keep on running. Regardless of whether all closer view applications are cleared, foundation administrations won't be interfered. In any case, on the off chance that a power conclusion of all applications happens, at that point the member will be needed to open the application again to proceed with information assortment. In the activity that a user doesn't have area consents empowered, then the app will send the user a warning. This reminds the user that area authorizations ought to be empowered. User can tap on the warning, which will guide them toward the pertinent settings through which applicable authorizations can be re-empowered. Furthermore, the force saving modes present in some Android gadgets may restrict the quantity of area focuses recorded by a gadget on the off chance that it has not been utilized for a protracted timeframe (which for retrieval in the reference is the name of the user who posted the article to Android Developer Website, Android Developer, 2018). In any case, this can halfway be relieved by guaranteeing that clients physically whitelist the application, which expands the quantity of accessible information logging windows. Lastly, removing/uninstalling the application is interpreted as a desire to withdraw from the usage, and this will stop the collection of data and delete all the files that is associated to the app.

4.5 Customization

The source code can be modified and customized. The frequency of location and the location data source such as GPS, Wi-Fi, etc., is used by default. We have illustrated these files in the Constant.java file. By following the customization process, the application can then be redistributed on the Google Play store.

Data will not be shared with the other apps on the device. All the information gathered by the mobile app can be analyzed with some other streams of data which has been obtained from different devices or apps. This might include methodologies that capture time-stamped objective measures of behaviour (e.g., physical activity from an accelerometer), or survey response items over longer periods of time (e.g., mood assessment from an experience sampling application) (Jankowska, Schipperijn, & Kerr, 2015).

4.6 Data processing and analysis

Data of any location can be analyzed in three key ways. The first way is to place the points of a particular location in the database, like cafeteria, lecture rooms, clinic, etc. We can characterize the identified locations with this method on the basis of how they are related to some other geographic databases; examples are the longitude, latitude and the full address. The second method is by characterizing movements as a form of behaviour in some number of ways (Carlson, Jankowska, Meseck, Godbole, Natarajan, Raab, & Kerr, 2015). Information related to radius of gyration, travel distance can be provided. According to the research carried out by the psychological experts of recent, it appears that the analysis which includes information related to both destination and journey is more valuable incrementally (Huang and Barnes, 2016). Lastly, the consideration of time can provide information as regards when an individual is engaged in specific behaviours or activities. As location data become easier to collect, there are still potentials for all these analyses to develop/improve further.

4.7 Response time

The mobile API response time matters when it comes to mobile app development. Slow response times can be hindering to the adoption of mobile apps. A delay of just a few seconds is sufficient to make a few users relinquish an application. Application response time is a significant factor that can impact the reception and utilization of portable. This can be especially obvious in the venture, where making a versatile application can be costly, partially due to complex mix with back-end frameworks. The expense of portable application improvement ought to be offset with the worth it makes for the business to help produce a quantifiable profit.

With mobile applications that employees rely upon to finish their jobs, slow load times can decrease efficiency and lead to representative disappointment and customer disappointment. Portable applications, in contrast to work area applications, have restricted assets, for example, battery life, preparing pace, and transfer speed use, and UI/UX plan choices need to represent this. Luckily, a few stunts and acclimations to the RESTful application programming interfaces (APIs) that regularly give information to portable applications can help improve load times while being mindful to asset utilization. NEU Places mobile app has been fully optimized in order to get a better and fast response time, some of the methods used are listed below:

a) **Implementation of Splash Screen:** A significant part of mobile UI/UX configuration is the apparent reaction season of portable APIs in stacking information. Introducing a sprinkle screen when the application is dispatched is one stunt to help improve client experience by veiling the genuine application load time. These beginning up screens look like the fundamental subject of the application yet is without content, giving the client consolation that something are going on while the application loads content.

b) **Mobile-centric APIs to connect back end:** Numerous organizations actually depend on inheritance frameworks that were not planned in light of versatile. Information coming from inheritance APIs may not be advanced for show or preparing on a cell phone and can debase the client's experience. In any case, numerous undertaking portable arrangements need the information and data that is put away on heritage frameworks. MBaaS contributions give the instruments and climate important to make versatile driven RESTful APIs that are intended to coordinate with heritage frameworks, offer quicker burden times, and create portable cordial information.

c) **Removal of unnecessary data and files:** Huge volumes of information set aside a long effort to download. By limiting the size of a payload sent from the worker to a cell phone, the time a client spends holding up can be diminished. Numerous methods exist to decrease payloads. One technique is to eliminate information that the cell phone and client needn't bother with. While this may appear glaringly evident, it can undoubtedly be neglected during improvement. Current MBaaS arrangements frequently uphold the Node.js JavaScript runtime, which can rearrange mentioning and marshaling heritage information to eliminate superfluous fields and change to gadget inviting JSON designs. This can diminish the payload and consequently cut the time a cell phone spends downloading information. Moderate reaction time or anything that can hamper appropriation or use can bring down the estimation of the application and any subsequent ROI. Quicker reaction time can improve portable achievement both as far as representative and client selection. Portable driven APIs, storing instruments, and some other great UI/UX practices can assist with accomplishing this.

4.8 Calendar Map with Translator Feature

Being more organized can bring a host of benefits to our life, whether we are doing it for our own betterment or to help things run more smoothly. It can free up time we did not know we had, identify gaps in scheduling that we could use more efficiently and cut down on unnecessary stress. We have decided to add an extra feature to our app which connects personal calendars to geo-location on campus for services such as scheduling and showing students and staff their daily lectures and meetings locations on Google map. This is an extra feature that has been implemented on the mobile app which can be used to track events. Events can be scheduled on the mobile phone which displays the location on the app.

Android Permissions

Our app needs to have access permissions in order for it to access calendar provider. App can request required permissions in manifest file. We have used READ_CALENDAR and

WRITE_CALENDAR permissions in our application for this purpose. The app can get the calendar data using calendar provider or calendar intents. Calendar provider is a content provider which is one of the components in android and lets app share data with other apps. We have linked the mobile app with the mobile calendar with these two permissions.

CHAPTER 5

IMPLEMENTATION AND METHODOLOGY

The Location-based service is a key functionality which is used in smart phone applications. To give a good experience to the user about various locations, we often combine locations with maps. The Android devices support Application Programming Interfaces which is generally known as APIs (which for retrieval in the reference is the name of the user who posted the article to Android Developer official website, Location Manager Application Programming Interfaces). The current location can be discovered on a mobile app by using the location service. The periodic updates of device location information can be requested on any app that is location-based.

5.1. The Application Programming Interface of Android Location

The Android location APIs are classes different from some other APIs. These classes are available in the package of Location API. Location APIs is also used for retrieving the Location information of users (which for retrieval in the reference is the name of the user who posted the article to Android Developer official website, Location Manager APIs).

a) ***The LocationManager:*** This is a class which gives the location services an access. Additionally, this also gives room to discover the criteria of the best Location Provider.

b) ***The LocationProvider:*** This is the location provider abstract super class. It gives the reports of the period on location geography of a mobile device.

c) ***The LocationListener***: A class which gives call-bck methods. This class is called whenever the location changes. The location manager has to be registered under the listener object.

d) ***The Criteria***: This is a class which provides the mobile app to select a suitable LocationProvider by giving access to a set of required properties of the Locationprovider.

The access to Google maps is also provided by Android with the help of API. Required places can be displayed on the screen to the user on the map with the help of APIs for location and Google maps.

5.2. The API of Google Places

The API of Google Places (which for retrieval in the reference is the name of the user who posted the article to Google official website, Google Places API) is a service that returns information/data about the places defined within a mobile app or Web service. It also returns places data which is defined within a spetial locations, or points of preffered interest with the help of HTTP requests. Latitude and longitude coordinates are used to specify a particular location. There are four major types of requests which are available with the Google Places API. The four fundamental Places available services are:

a) ***The Searches for Places:*** This is a service that returns an array of nearby Places based on the location that has been defined by a user.

b) ***The Details of Places:*** This is a service which returns a specific information/data about a Place that has been defined by a user.

c) ***The Check-ins of Places:*** A service that allows the request a user has checked in to a specific Place

d) It is used to guage the popularity of a Place. If a Place is check-in frequently, it boosts a Place's priority in the app's Place Search responses

e) ***The Place Reports:*** Users are allowed to add new location to the service of a Place. Places can also be deleted if a mistake has been made.

Some of the limitations for processing query over the API for Google Places (which for retrieval in the reference is the name of the user who posted the article to Google official website, The API for Google Places) are:

- Only 1000 requests per 24hrs period only are allowed by users which have the Key for API.
- Only 100 000 requests for 24hrs period are allowed by users who have validated their identity with the help of the APIs console. For authentication which gives access to enable billing, an active credit card is required.

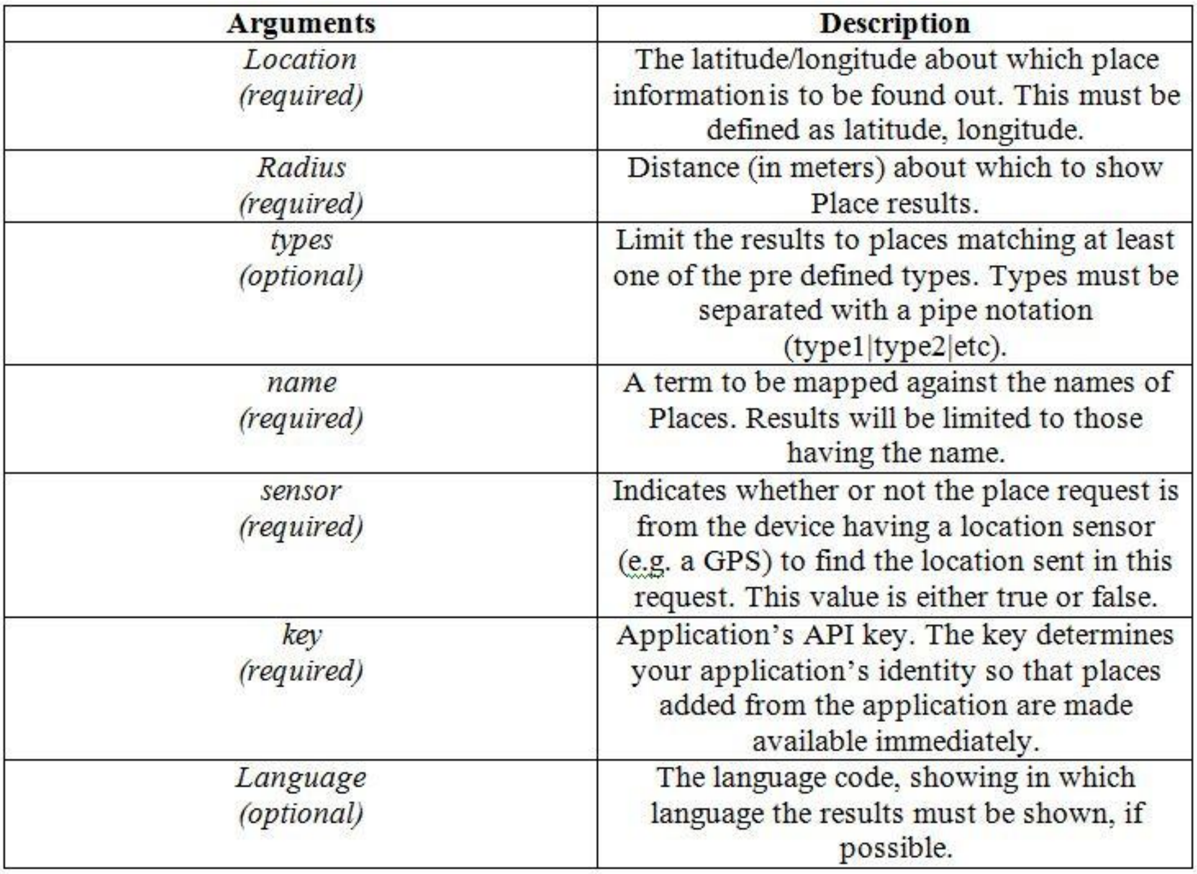
5.2.1. Places Searches

The HTTP URL is used to make request for a Place Search. It is defined in the following way (which for retrieval in the reference is the name of the user who posted the article to Google official website, Google Places API):

https://maps.googleapis.com/maps/api/place/search/output?arguments

The output of these requests can be either JavaScript Object Notation (JSON) or eXtensible Markup Language which is popularly known as XML.

Table 5.1: The arguments for Place Search API



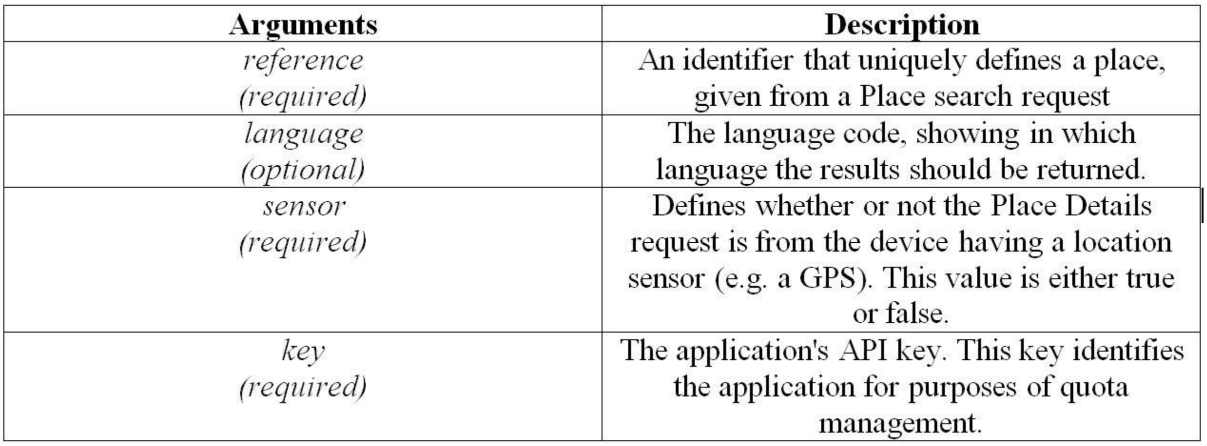
Arguments	**Description**
Location *(required)*	The latitude/longitude about which place informationis to be found out. This must be defined as latitude, longitude.
Radius *(required)*	Distance (in meters) about which to show Place results.
types *(optional)*	Limit the results to places matching at least one of the pre defined types. Types must be separated with a pipe notation (type1|type2|etc).
name *(required)*	A term to be mapped against the names of Places. Results will be limited to those having the name.
sensor *(required)*	Indicates whether or not the place request is from the device having a location sensor (e.g. a GPS) to find the location sent in this request. This value is either true or false.
key *(required)*	Application's API key. The key determines your application's identity so that places added from the application are made available immediately.
Language *(optional)*	The language code, showing in which language the results must be shown, if possible.

The details of a place is needed to return more detailed information about the defined place of the user(which for retrieval in the reference is the name of the user who posted the article to Google official website, Google Places API) such as the address, category, website, contact number, etc. HTTP URL is used for Place Details requests such as:

https://maps.googleapis.com/maps/api/place/details/output?arguements

The output of these requests can be either JavaScript Object Notation (JSON) or eXtensible Markup Language which is popularly known as XML.

Table 5.2: The arguments for Place Search Requests



Arguments	Description
reference *(required)*	An identifier that uniquely defines a place, given from a Place search request
language *(optional)*	The language code, showing in which language the results should be returned.
sensor *(required)*	Defines whether or not the Place Details request is from the device having a location sensor (e.g. a GPS). This value is either true or false.
key *(required)*	The application's API key. This key identifies the application for purposes of quota management.

5.2.2 System Testing

We have developed the proposed app using the Android Studio software. This is a software for mobile app developers for Android apps. All the APIs that has been mentioned covered the app development. This app has been tested on Samsung Galaxy S7 (this device is A-GPS mobile device). The CPU of the device is Android 6.0 (Marshmallow).

Android Permissions

To perform network operations in our application, we have included the *android.permission.INTERNET* in our manifest file. Without this permission, the monile app will not be able to connect to the internet. The location permissions offered by Android are two which are *ACCESS_COARSE_LOCATION* and *ACCESS_FINE_LOCATION*. We have used these two permissions to determine the best accuracy of the location returned by the API.

CHAPTER 6

SYSTEM PERFORMANCE AND VALIDATION

For any mobile app, performance is very critical. We ensure that the performance of the proposed mobile app is fully optimized because if the mobile app is slow, the end user will uninstall the app and find other related application that performs better. Our system is compatible and able to take benefits of existing cloud services. We have chosen this approach because we are interested in the communication between the mobile application and the remote server.

6.1 Test Environment and Test Plan

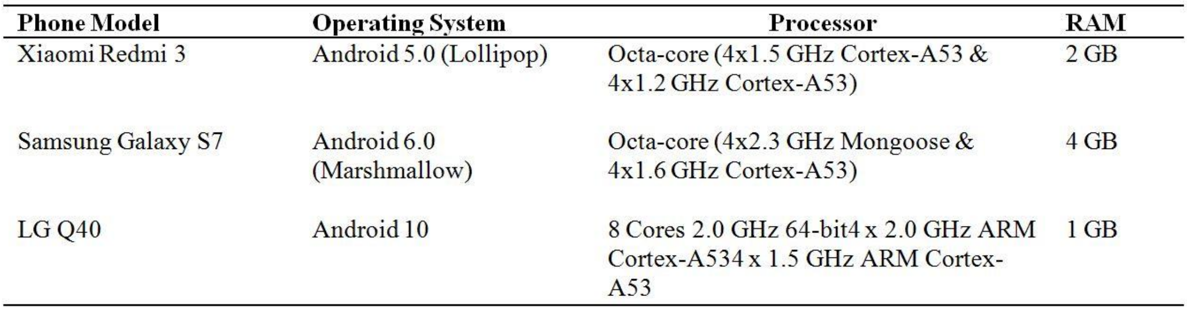
The environment where the tests will run, together with the tools and required hardware constitute the test environment.

The test environment in this research consists of:

- Cloud that hosts the mobile app
- Mobile devices for testing
- Tools for monitoring the utilization of hardware resources in the mobile devices
- Tools to analyze and visualize the data.

The web cloud has 4GB of RAM and Quad-core 3.3GHz CPU. The list of mobile phones and their specifications that has been used for testing the app is displayed in Table 6.1.

Table 6.1: Phones for Testing



Phone Model	Operating System	Processor	RAM
Xiaomi Redmi 3	Android 5.0 (Lollipop)	Octa-core (4x1.5 GHz Cortex-A53 & 4x1.2 GHz Cortex-A53)	2 GB
Samsung Galaxy S7	Android 6.0 (Marshmallow)	Octa-core (4x2.3 GHz Mongoose & 4x1.6 GHz Cortex-A53)	4 GB
LG Q40	Android 10	8 Cores 2.0 GHz 64-bit4 x 2.0 GHz ARM Cortex-A534 x 1.5 GHz ARM Cortex-A53	1 GB

For each test we measure the time required for a task to execute, the usage of RAM and CPU. The tests were run on Wi-Fi and 4G network. The time required for a task to execute is calculated within the code. The execution time for cloud calculations will include the overall time needed for the request to finish. Additionally the server time needed for processing will be measured so that we can observe the delay for each cloud request. To measure the utilization of the device's resources, RAM and processor, the Android Profiler tool will be used. Android Profiler is part of Android Studio 3.0; it provides real-time graphical display of CPU usage, memory and network usage for the running mobile application.

The process of accurately measuring mobile phone power consumption is a difficult task. One approach is to take physical power measurements on a piece of real hardware with specialized measurement equipment. There are software tools to perform this type of measurements, but it is hard to get an accurate value. So the focus here is on execution time, CPU and memory utilization of mobile application functions, where more accurate values can be gathered.

6.2 Results

The speed of the device is listed on the y-axis in Mbps while the devices with their parameters which we call devices and the network type are listed on the x-axis. We have analyzed the connection speed on three different devices which are Xiaomi, Samsung and LG. We have decided to use these devices because they are the most popular and widely used Android devices. After which we run the app on all these three devices, the output is given in Figure 6.1. The best connection is 4G according to our result which means that our app can run on the latest connection speed.

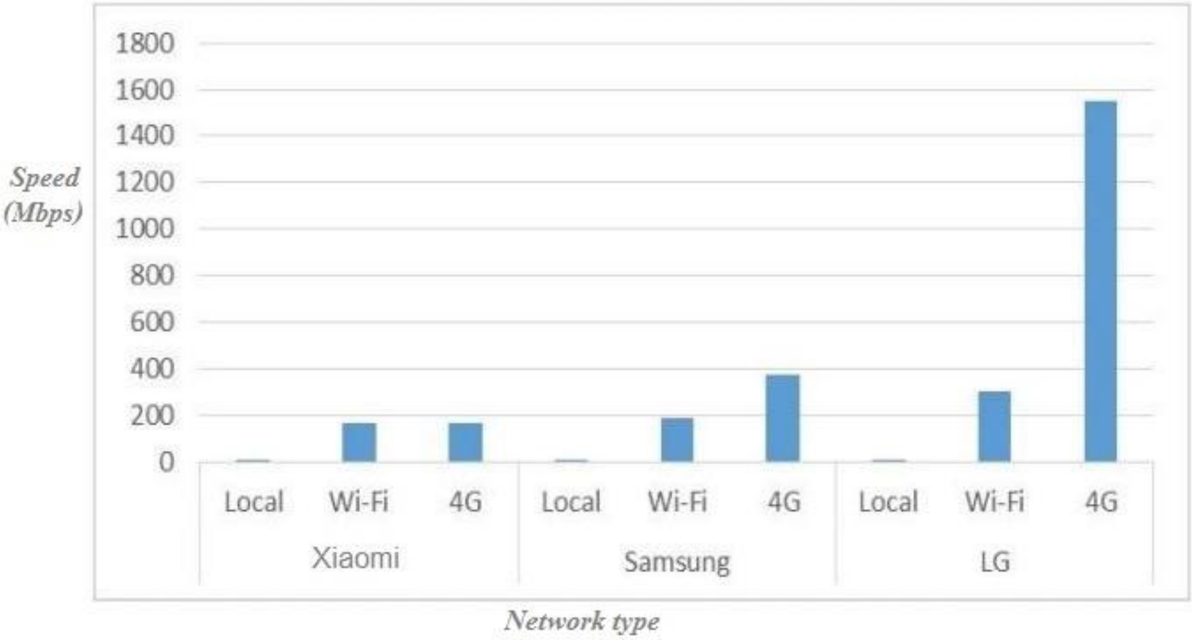


Figure 6.1: Execution Time of the proposed app

The tests were executed on network with the following average download/upload speed: 15Mbps/16Mbps for Wi-Fi connection, and 3Mbps/1Mbps for 4G connection. The average execution time is taken for each test. Values are displayed in milliseconds. In Figure 6.1, the execution time of accessing an element is shown, for local, Wi-Fi and 4G. In this test, an array of predefined length is created, and its first element is returned.

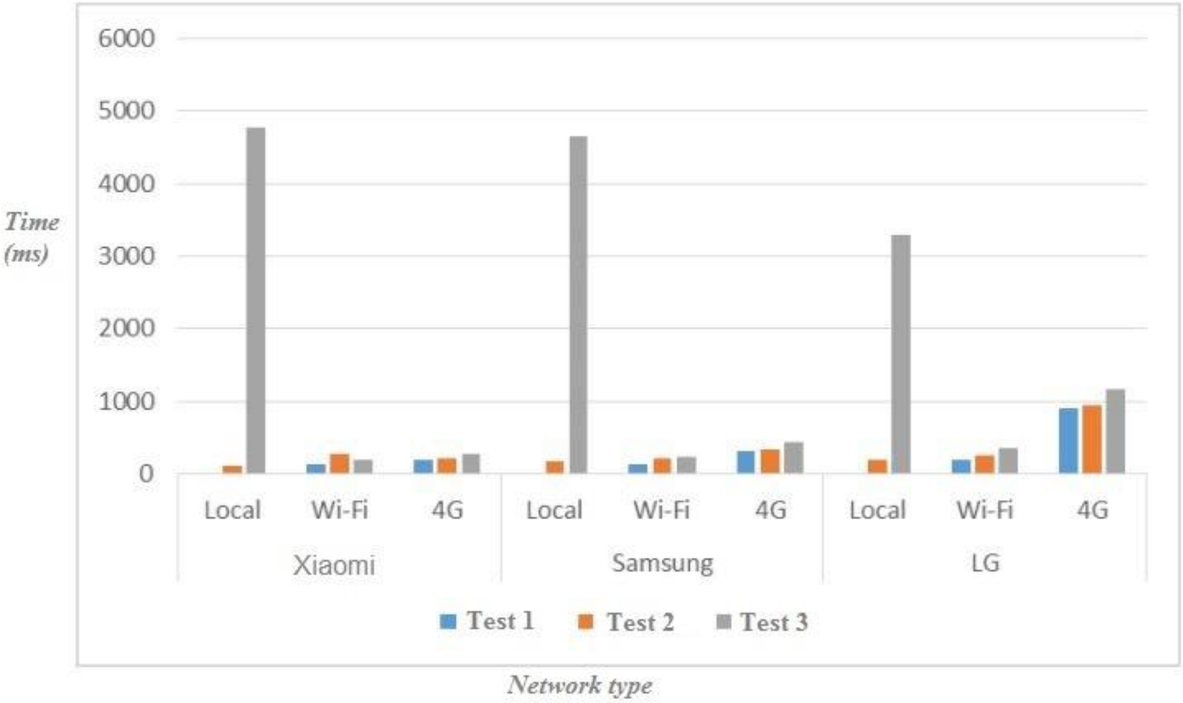


Figure 6.2: Execution Time of Iterate thought array

In the second test, local time takes from 119ms-191ms, and cloud execution time takes from 212ms-275ms, of which 3ms is server time. Cloud calculations are much faster than local in the third range, where local time takes around 4235ms, while the cloud time is around 264ms, with 56ms server time.

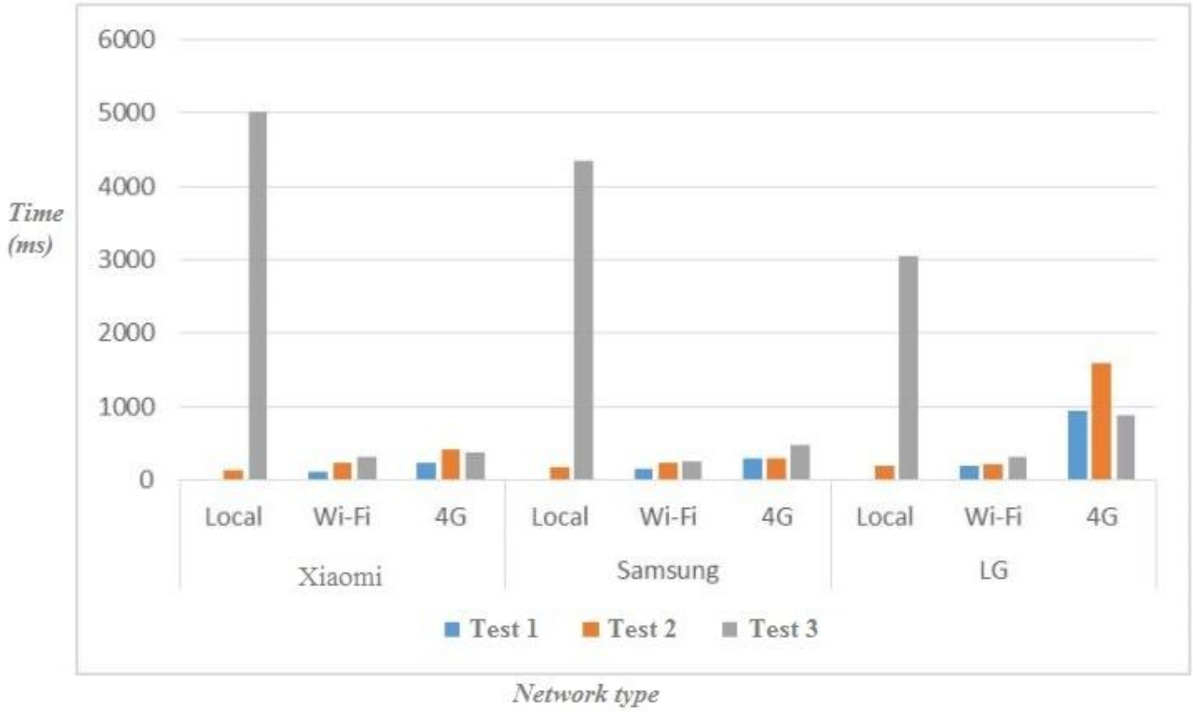


Figure 6.3: Execution Time of Iterate thought matrix

Figure 6.4 displays the execution time of Fibonacci number calculation when we run the app on Android Studio. The local calculations are faster only in the first test. In the second and third test, the cloud calculations are considerably faster.

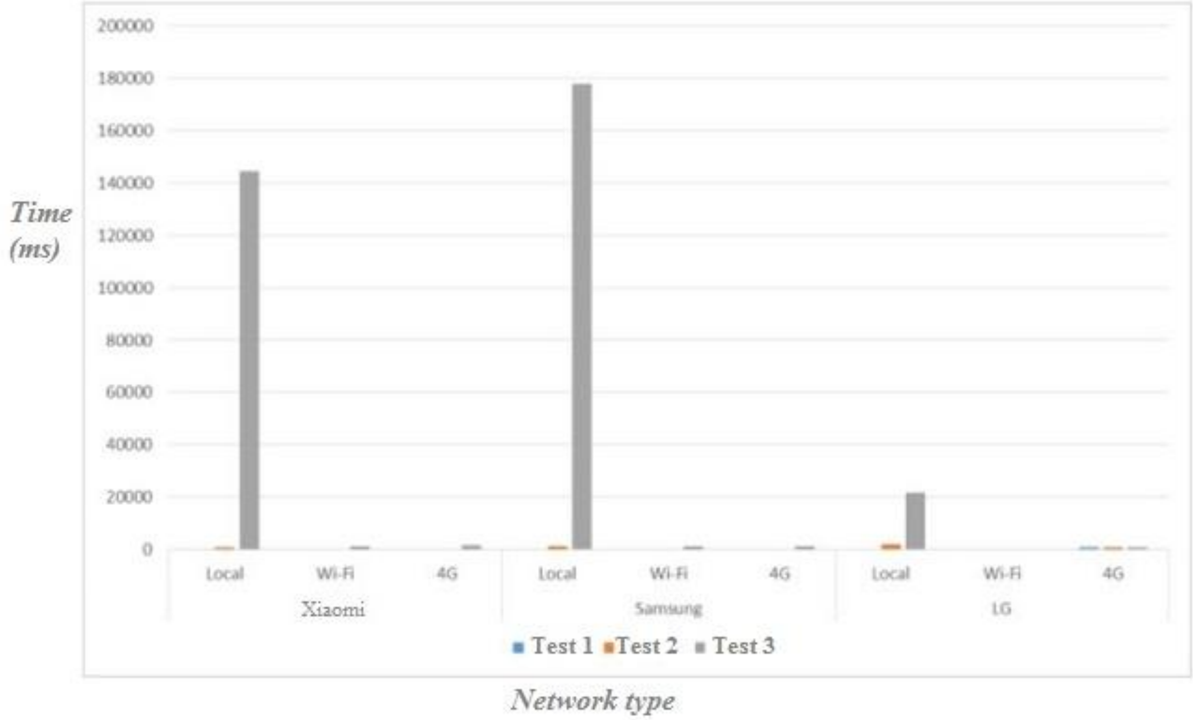
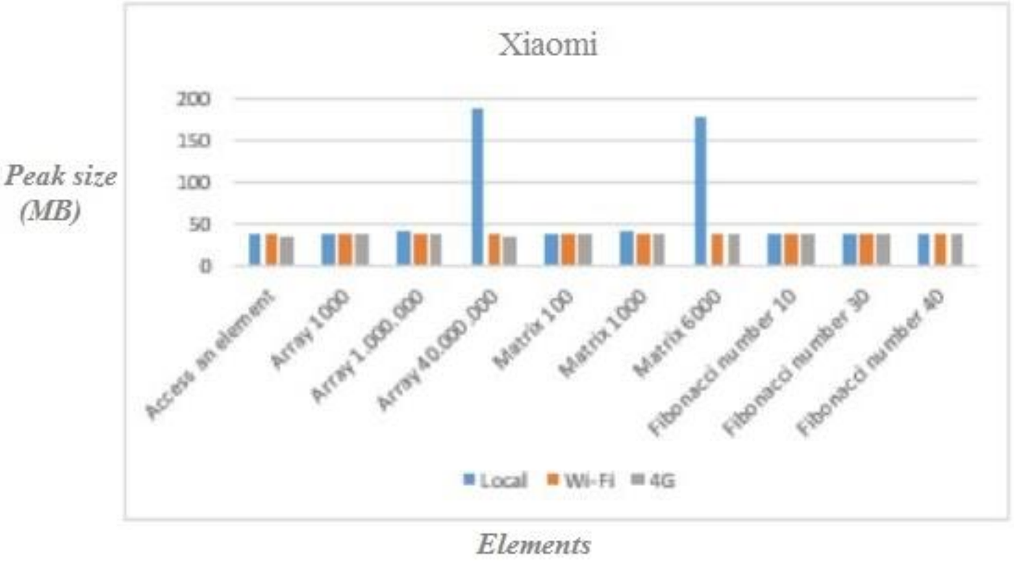
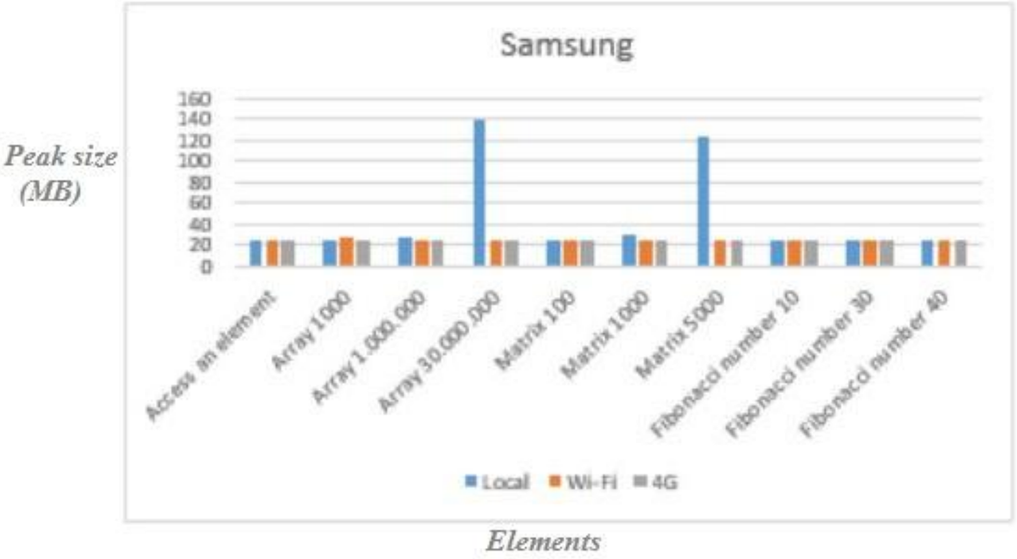


Figure 6.4: Execution Time of Fibonacci number

The local time in the first test takes 1ms-4.8ms, and the cloud time takes 138ms-206ms, with 0ms server time. In the second test, local time takes from 1166ms-1974ms, and cloud time takes from 143ms-276ms, with 4ms server time. In the third test the local time takes around 114s, while cloud takes around 1s, of which 575ms is server time. If we analyze the execution time of all test cases, the highest difference is in the third test of the test cases where local time takes from 3 seconds to 178 seconds (about 3 minutes), which degrades user experience. On average, Wi-Fi network processing is always faster than the 4G, about 55%. The local execution time in the third test shows that the LG execution time is quite small compared to the other phones. This is because on Xiaomi and Samsung the test is run for value 40, and on LG the test is run for value 35. LG has lower hardware specifications and for higher value the application crashes. Figure 6.4 shows a memory usage across all three devices.





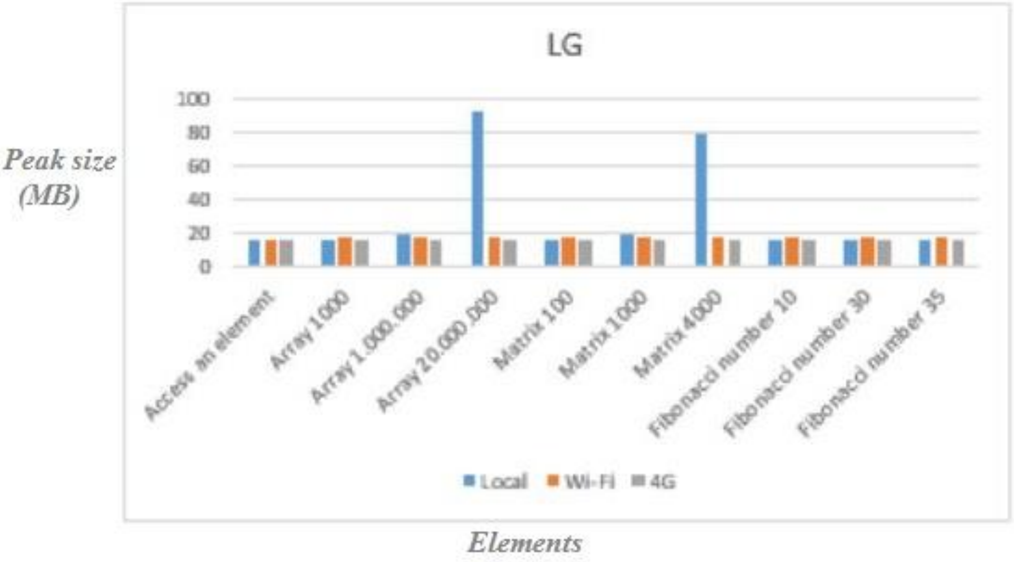
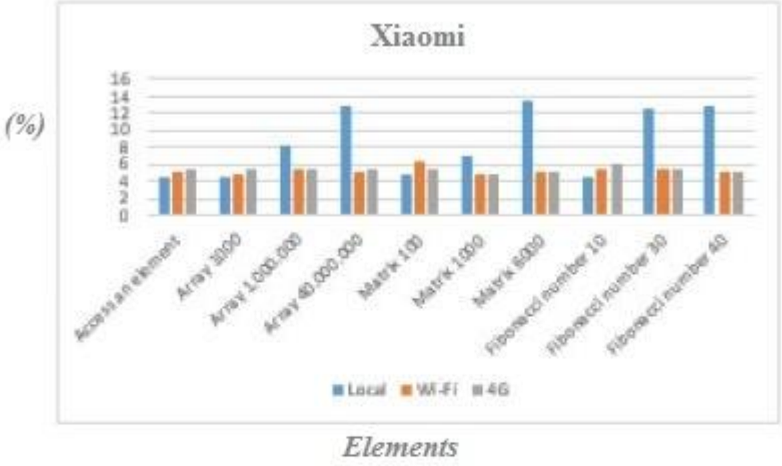


Figure 6.5: Memory Usage of all devices

Memory utilization bounces for local calculations for the largest tests in the test cases for array and matrix iteration. The highest memory peaks are 188MB for an array of 40.000.000 elements in Xiaomi, and 139MB for 30.000.000 elements in Samsung. For all other cases there is no significant difference between local and cloud calculations. There is no significant difference between the usage of memory in Wi-Fi and 4G calculations on all devices.

Figure 6.6 shows CPU utilization of all devices expressed in percentage.



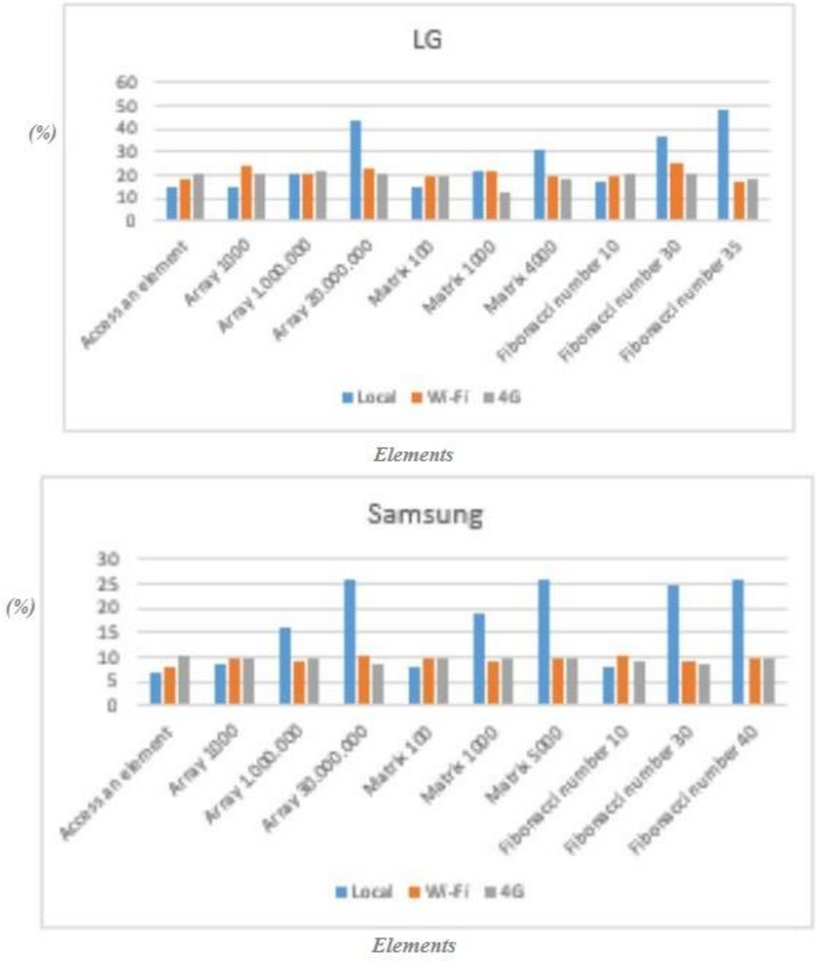


Figure 6.6: CPU Usage

The presented CPU utilization percentage is in terms of the whole CPU capacity. Since we use single-threaded code, one core is used for processing. The CPU utilization bounces for local processing, in test cases with the last two tests in Xiaomi and Samsung. In the remaining cases, there is no significant difference in the utilization between local and cloud. In Xiaomi, the utilization ranges from 4.4%-13.6% for local, and 4.8% to 6.4% for cloud calculations. In Samsung, the CPU ranges from 7%-26% for local, and from 8.2%-10.4% for cloud calculations. In LG, local processing bounces in the third test of the second and third test case, and in the last two ranges of the fourth test case. In the remaining cases, there is no significant difference in the utilization between local and cloud processing. The local CPU utilization takes from 14.6%-48.6%, while in the cloud from 18%-25.2%. There is no significant difference in the utilization between Wi-Fi and 4G.

6.3 Experimental Setups

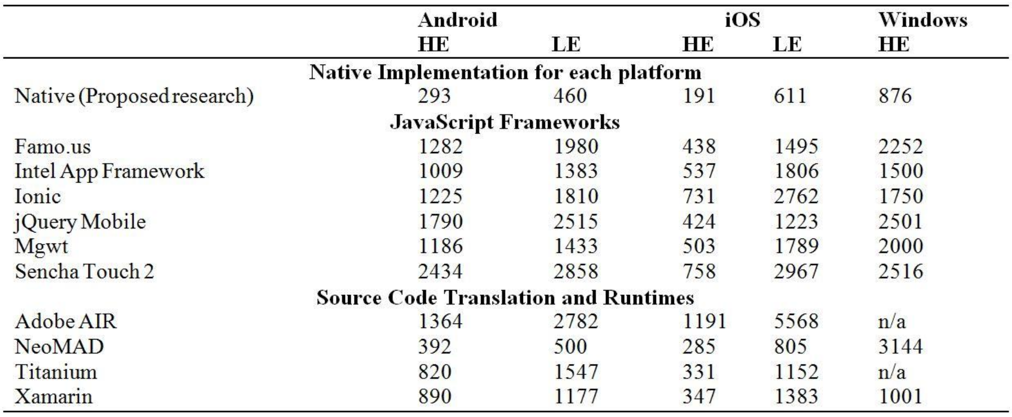
There are a couple of rules that talks about an application response time and the user experience. Response time that is below 100ms will be very fast on user's device. Any response time that is up to one second(s) are adequate to the users. Mobile application testing is a sort of A/B testing wherein diverse user portions are given different assortments of an in-application experience to figure out which one initiates the ideal activity from them. Mobile application testing incorporates all capacities tests which run on the server-side, as well as permiting advertisers with the managers of item products to optimize the experiences in the end-to-end user interface.

A few models are considered to determine the tools for the cross-platforms to exhibit the examination. Most importantly, there is the accessibility of a PropertyCross usage utilizing the device. This doesn't confine the exploration, as most notable and regularly utilized cross-platform devices are upheld (Retrieved, 2012).

6.3.1 Launch Time

In Table 6.2, we have talked about the launch time of the proposed app when if it has been developed on different platforms. This table shows the presented results of cross-platform devices in a slow dispatch time contrasted with local turn of events. Generally, the execution with Touch Sencha 2 is the slowest to begin. The general request of the other JavaScript structures is reliant on the platform on which the application is running. For example, jQuery portable has the best dispatch time on iOS while it performs a lot of more terrible on Android and Windows Phone. Albeit less noticeable contrasted with the webview for JavaScript system executions, the runtime is for the biggest part answerable for the dispatch time overhead of the cross-stage apparatus usage in the runtime classification. NeoMAD is a source code interpreter that doesn't utilize a runtime yet deciphers the source code of the application to devoted source code for the various platforms. The proposed application is a Native application which runs on Android platform.

Table 6.2: Launch times in ms.



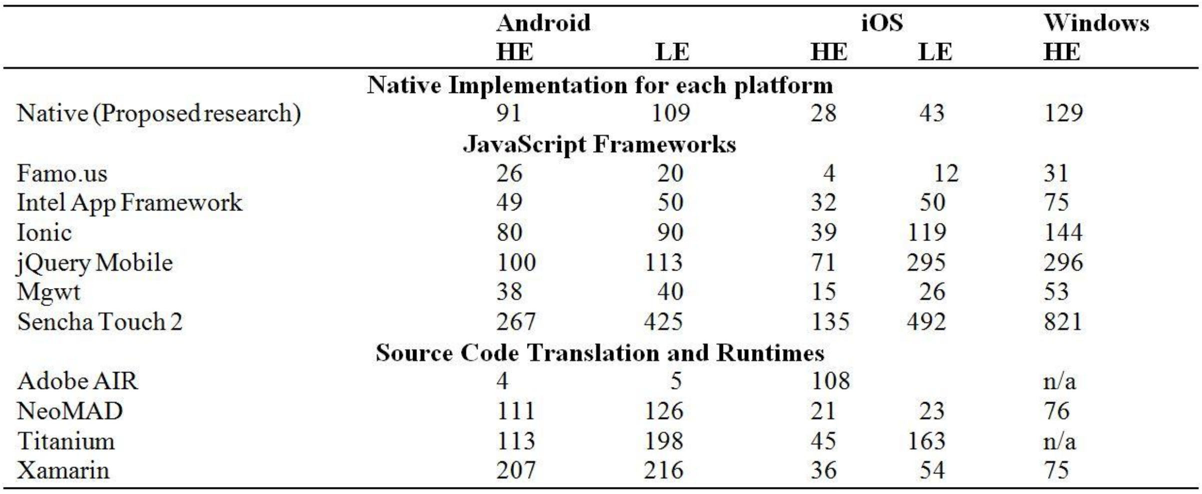
	Android		iOS		Windows
	HE	LE	HE	LE	HE
Native Implementation for each platform					
Native (Proposed research)	293	460	191	611	876
JavaScript Frameworks					
Famo.us	1282	1980	438	1495	2252
Intel App Framework	1009	1383	537	1806	1500
Ionic	1225	1810	731	2762	1750
jQuery Mobile	1790	2515	424	1223	2501
Mgwt	1186	1433	503	1789	2000
Sencha Touch 2	2434	2858	758	2967	2516
Source Code Translation and Runtimes					
Adobe AIR	1364	2782	1191	5568	n/a
NeoMAD	392	500	285	805	3144
Titanium	820	1547	331	1152	n/a
Xamarin	890	1177	347	1383	1001

Consequently, the launch time of the NeoMAD is expected to closely look like the native launch time which works like the Native platforms. We have used the Native implementation to develop our proposed app.

6.3.2 Time to navigate between pages of the application

In the section, we investigate the response times for exploring to an activity then getting back to another activity when the application is created on various programming platforms. In Table 6.3, utilizing JavaScript usage systems (such as Famo.us, Mgwt) for the most part accomplish great reaction seasons of under 100ms. Nonetheless, when we open up an activity, it is held in the phone memory and when we switch to the next activity, the response time is calculated. The proposed application is a Native application which runs on Android devices.

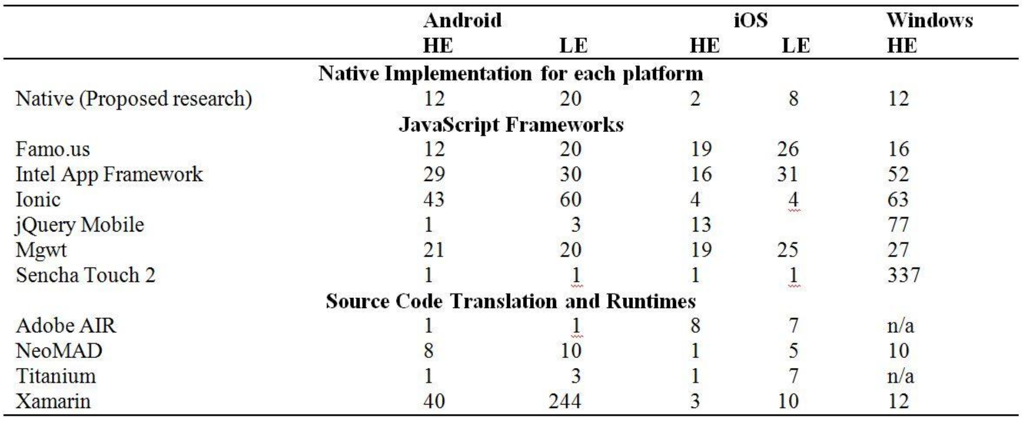
Table 6.3: The time taken to open a different activity (in ms).



	Android		iOS		Windows
	HE	LE	HE	LE	HE
Native Implementation for each platform					
Native (Proposed research)	91	109	28	43	129
JavaScript Frameworks					
Famo.us	26	20	4	12	31
Intel App Framework	49	50	32	50	75
Ionic	80	90	39	119	144
jQuery Mobile	100	113	71	295	296
Mgwt	38	40	15	26	53
Sencha Touch 2	267	425	135	492	821
Source Code Translation and Runtimes					
Adobe AIR	4	5	108		n/a
NeoMAD	111	126	21	23	76
Titanium	113	198	45	163	n/a
Xamarin	207	216	36	54	75

The response time of some programming platform usage have better response time contrasted with the local variant. This can be credited to the quick delivering of some activity on the screen, the activity that happens when exchanging between different pages on local apps. In Table 6.4, reaction delays for getting back to previous activity are for the most part in a way that is better than when opening another activity. We have utilized the Native execution to build up our proposed application which runs on Android gadgets.

Table 6.4: The time taken to return to the previous activity (in ms).



	Android		iOS		Windows
	HE	LE	HE	LE	HE
Native Implementation for each platform					
Native (Proposed research)	12	20	2	8	12
JavaScript Frameworks					
Famo.us	12	20	19	26	16
Intel App Framework	29	30	16	31	52
Ionic	43	60	4	4	63
jQuery Mobile	1	3	13		77
Mgwt	21	20	19	25	27
Sencha Touch 2	1	1	1	1	337
Source Code Translation and Runtimes					
Adobe AIR	1	1	8	7	n/a
NeoMAD	8	10	1	5	10
Titanium	1	3	1	7	n/a
Xamarin	40	244	3	10	12

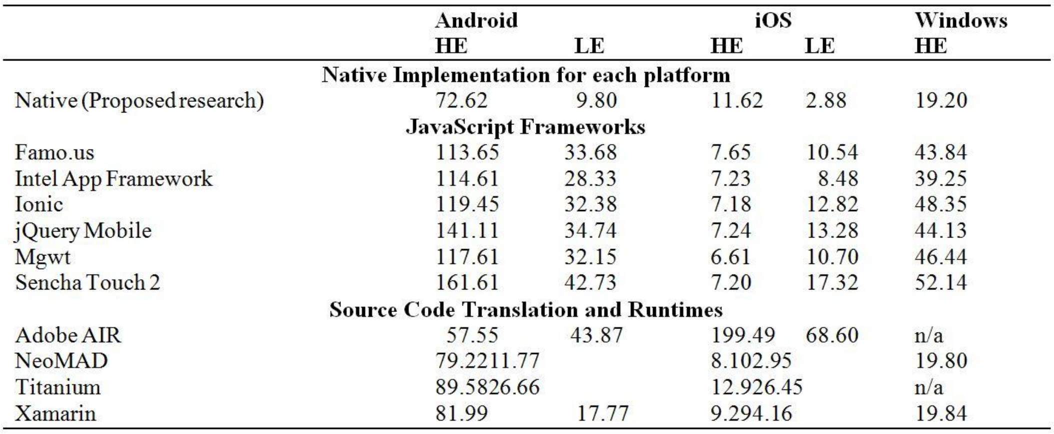
Aside from the seventh Framework on the table above, rendition on the low-end devices, every reaction delay is underneath 100 milliseconds. Henceforth, users must definetely encounter this progress as momentary.

The tables plainly show contrasts in memory the executives methodologies utilized by the various platforms. Android distribute essentially additional Random Access Memory (RAM) for every execution contrasted with iOS. We have utilized the Native usage to build up our proposed application which runs on Android devices.

6.4 The Usage of Central Processing Unit (CPU)

The usage of CPU is the piece of the complete accessible CPU time spent by an application during a specific time span.

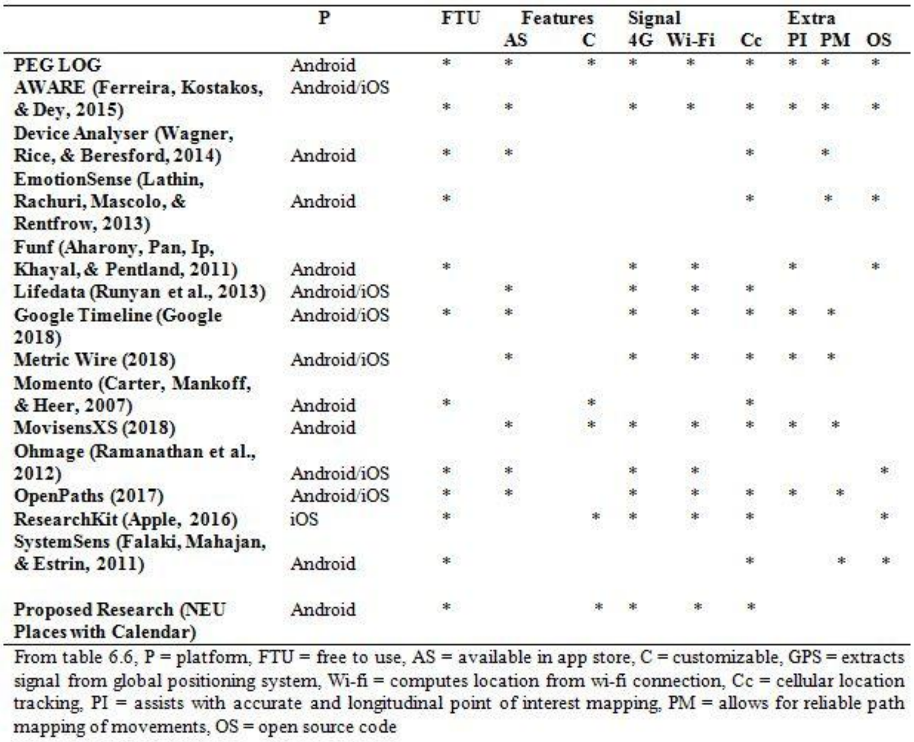
Table 6.5: The consumption of the phone memory after launching the app (in MB).



	Android		iOS		Windows
	HE	LE	HE	LE	HE
Native Implementation for each platform					
Native (Proposed research)	72.62	9.80	11.62	2.88	19.20
JavaScript Frameworks					
Famo.us	113.65	33.68	7.65	10.54	43.84
Intel App Framework	114.61	28.33	7.23	8.48	39.25
Ionic	119.45	32.38	7.18	12.82	48.35
jQuery Mobile	141.11	34.74	7.24	13.28	44.13
Mgwt	117.61	32.15	6.61	10.70	46.44
Sencha Touch 2	161.61	42.73	7.20	17.32	52.14
Source Code Translation and Runtimes					
Adobe AIR	57.55	43.87	199.49	68.60	n/a
NeoMAD	79.2211.77		8.102.95		19.80
Titanium	89.5826.66		12.926.45		n/a
Xamarin	81.99	17.77	9.294.16		19.84

We have compared all these results with the proposed research. The mobile app has been developed on Android platform. We ensure that the response time of the proposed research is lower than the other results. This has been achieved by fully optimizing our mobile app by getting rid of codes that are not adding any value to the app, checking the apps' efficiency, trying and testing the code, use profiling tools for monitoring, emphasizing on increasing app usability, and focusing on the user interface. Having putting these into consideration, we are able to have a better response time which makes our app faster that the other ones. The response time for the proposed research over the Wi-Fi is 5.28s and 5.71s over 4G network. We have used the Native implementation to develop our proposed app which runs on Android devices.

Table 6.6: Research methods features of how to track locations with Smartphones



	P	FTU	Features		Signal			Extra		
			AS	C	4G	Wi-Fi	Cc	PI	PM	OS
PEG LOG	Android	*	*	*	*	*	*	*	*	*
AWARE (Ferreira, Kostakos, & Dey, 2015)	Android/iOS	*	*		*	*	*	*	*	*
Device Analyser (Wagner, Rice, & Beresford, 2014)	Android	*	*				*		*	
EmotionSense (Lathin, Rachuri, Mascolo, & Rentfrow, 2013)	Android	*					*		*	*
Funf (Aharony, Pan, Ip, Khayal, & Pentland, 2011)	Android	*			*	*		*		*
Lifedata (Runyan et al., 2013)	Android/iOS		*		*	*	*			
Google Timeline (Google 2018)	Android/iOS	*	*		*	*	*	*	*	
Metric Wire (2018)	Android/iOS		*		*	*	*	*	*	
Momento (Carter, Mankoff, & Heer, 2007)	Android	*		*			*			
MovisensXS (2018)	Android		*	*	*	*	*	*	*	
Ohmage (Ramanathan et al., 2012)	Android/iOS	*	*		*	*				*
OpenPaths (2017)	Android/iOS	*	*		*	*	*	*	*	
ResearchKit (Apple, 2016)	iOS	*		*	*	*	*			*
SystemSens (Falaki, Mahajan, & Estrin, 2011)	Android	*					*		*	*
Proposed Research (NEU Places with Calendar)	Android	*		*	*	*	*			

From table 6.6, P = platform, FTU = free to use, AS = available in app store, C = customizable, GPS = extracts signal from global positioning system, Wi-fi = computes location from wi-fi connection, Cc = cellular location tracking, PI = assists with accurate and longitudinal point of interest mapping, PM = allows for reliable path mapping of movements, OS = open source code

Table 6.6 shows the comparative results for different researchers’ response time and the proposed research. AWARE is a mobile app which has been built on mobile platform for context and data collection. AWARE mobile app runs on two platforms (Android and iOS). The response time of this mobile app when it is launched either with Wi-Fi or 4G are (7.09s and 9.03s) respectively. The second result that has been generated is the Device Analyser. This is a vigorous information assortment device which can dependably gather data on Android Smartphone use from an open local area of contributors. Device analyser runs on Android device only. The response time of this mobile app when it is launched either with Wi-Fi or 4G are (15.04s and 16.11s) respectively. The third result is the Momento mobile app. This is an apparatus that offers incorporated help for arranged assessment of universal processing applications. This app runs on Android platform. The response time of this mobile app when it is launched either with Wi-Fi or 4G are (8.91s and 9.02s). The fourth result is the Metric Wire mobile application which has been developed on Android and iOS platforms. The response time of this mobile app when it is launched either with Wi-Fi or 4G are (10.19s and 10.34s) respectively. The last result generated is the Google Timeline mobile app which has been built on Android and iOS platforms. This is a mobile app that has been developed by Google. It is a location tracking app. The response time of this mobile app when it is launched either with Wi-Fi or 4G are (7.23s and 7.31s) respectively.

We have proposed NEU Places app which has been linked with the calendar. This is a Native mobile app which runs on Android devices. It can schedule and display students and staff their daily lectures and meetings locations on the Google map and also display locations within the environment. The calendar feature lets us keep track of events and the location. It is the ideal tool for managing personal and professional schedules. An example of a view is given below:

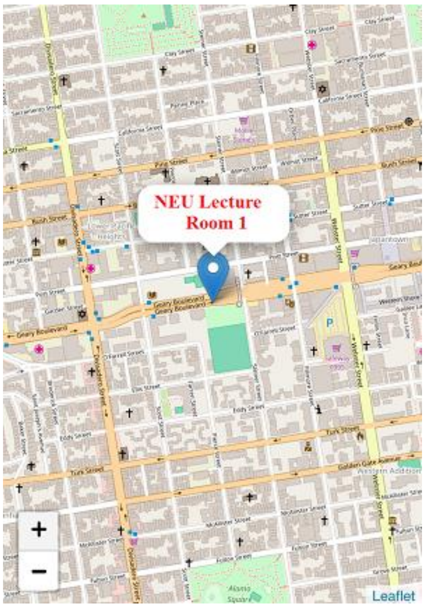
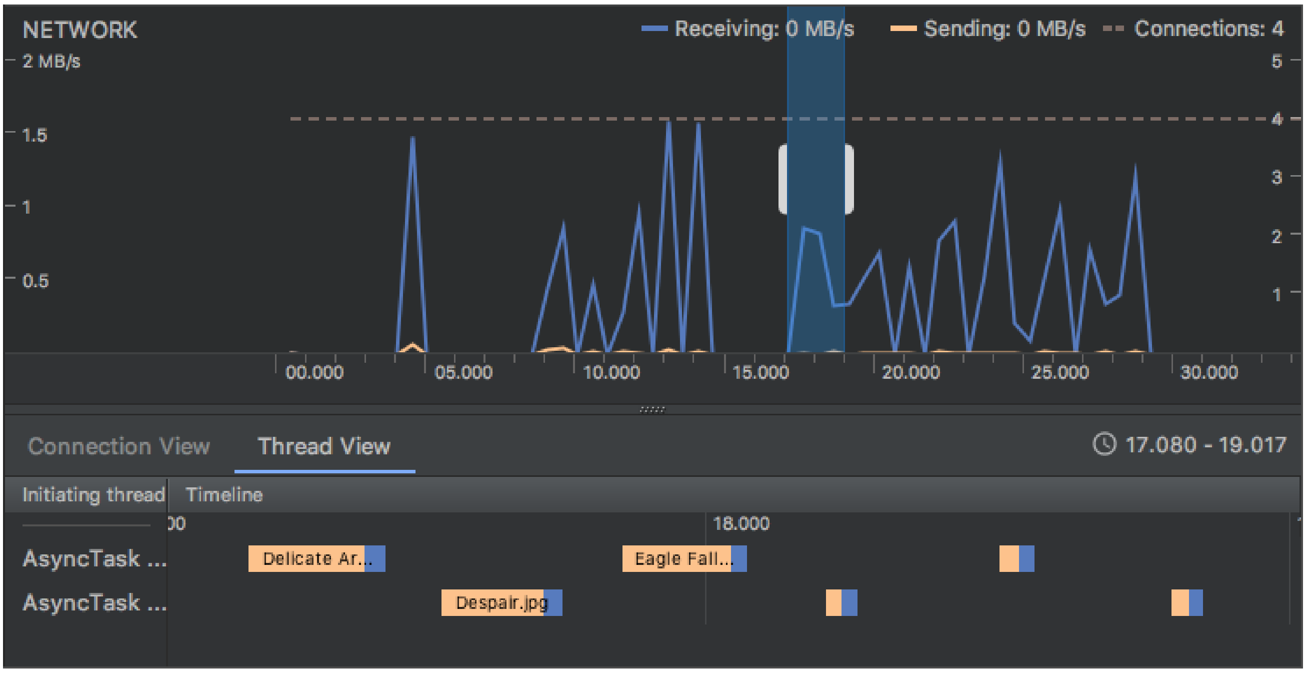


Figure 6.7: Calendar Map

Definition: From table 6.6, P represents the mobile device platform in which the mobile app runs, FTU describe whether the mobile app has been developed to be listed on App Store as a free to use or paid app, AS specifies if the app has been published on App Store or not, C is used to confirm if a mobile app is customizable or not, GPS extricates signal from worldwide situating framework, Wi-fi figures area from wi-fi association, Cc signifies cell area following, PI helps with exact and longitudinal focal point planning, PM takes into consideration solid way planning of developments, OS specifies if an app code is an open source code.

The Android Profiler provides real-time data to help in understanding how the app uses CPU, memory, network, and battery resources. This is a tool that runs on Android Studio. We have used this tool to analyze the performance of our mobile app. The Wi-Fi and LTE networks have been used to carry out this analysis. We have run our app on three different Android devices and the result obatined after doing these analyses on both Wi-Fi and LTE networks has been compared with some other existing mobile apps. The interface of the tool used for this analysis is shown below:

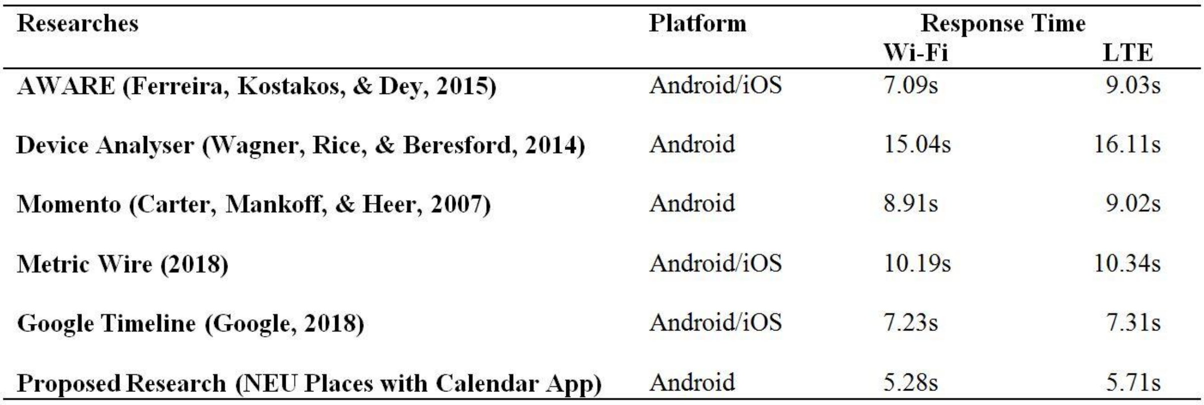


6.8: Android Studio Profiler

The Network Profiler displays realtime network activity on a timeline, showing data sent and received, as well as the current number of connections. This lets us examine how and when the app transfers data, and optimize the underlying code appropriately. The main focus is the network section, once we run our app on emulator, while connected to the Wi-Fi, a particular result is given, which is also applicable to the LTE network.

We have compared all these results with the proposed research. The mobile app has been developed on Android platform. We ensure that the response time of the proposed research is lower than the other results. This has been achieved by fully optimizing our mobile app by getting rid of codes that are not adding any value to the app, checking the apps' efficiency, trying and testing the code, use profiling tools for monitoring, emphasizing on increasing app usability, and focusing on the user interface. Having putting these into consideration, we are able to have a better response time which makes our app faster that the other ones. The response time for the proposed research over the Wi-Fi is 5.28s and 5.71s over LTE network.

Table 6.7: Comparative results for response time



Researches	Platform	Response Time	
		Wi-Fi	LTE
AWARE (Ferreira, Kostakos, & Dey, 2015)	Android/iOS	7.09s	9.03s
Device Analyser (Wagner, Rice, & Beresford, 2014)	Android	15.04s	16.11s
Momento (Carter, Mankoff, & Heer, 2007)	Android	8.91s	9.02s
Metric Wire (2018)	Android/iOS	10.19s	10.34s
Google Timeline (Google, 2018)	Android/iOS	7.23s	7.31s
Proposed Research (NEU Places with Calendar App)	Android	5.28s	5.71s

At the point when area information cannot be gathered, this mobile device app tries analyzing the wellspring issue. Take for example; a recording is made when we restart the mobile phone. Past these assignments, a touch of preparing of area information is completed inside the proposed application itself. Subsequently, the application insignificantly affects battery execution, in any event. This contrasts well and numerous other famous applications, which run an enormous various foundation cycles and information sharing components naturally, which are once in a while clarified to the end client (Kleek, Liccardi, Binns, Zhao, Weitzner, & Shadbolt, 2017).

CHAPTER 7

CONCLUSION

Cell phone's application changed the method of lives more than nothing particularly android mobile applications. In excess of 190 nations around the planet a great many Android mobile phones are established and these numbers are growing daily (Meier,2012). NEU Places mobile application has been developed and discussed in this report for easy usage on Android devices. This application works as different locations within the campus vicinity can be automatically detected. It is a complete native android application to find different locations such as lecture rooms, cafeteria, dormitory, bus stops, ATM, university clinic, and many more. It provides advanced mapping systems that will show different places of interest for app users. The performance was assessed on five unique gadgets, two Android and iOS devices. In light of the analysis of the performance, general finishes of which application developers ought to know while choosing a particular tool for cross-platform are drawn. Notwithstanding, utilization of cross-platform tool can affect explicit useful parts of the applications that is developed for mobile. Henceforth, the work that will be done in future will contain the examining of these perspectives when estimating, such as, executing complex calculations, getting to device assets, for example, sensors or steady memory and correspondence with different devices.

REFERENCE

Ali S, Ahmad A, Iqbal R, Saleem S, Umer T. Joint RRH-association, sub-channel assignment and power allocation in multi-tier 5G C-Rans. *IEEE Access*. 2018;6:34393-34402.

Al-Turjman, F. (2019). Smart-city medium access for smart mobility applications in Internet of Things. *Transactions on Emerging Telecommunications Technologies*, e3723.

Al-Turjman F, Hassanein H, Alsalih W, Ibnkahla M. *Optimized relay placement for wirelesssensor networks federation in environmental applications*. Wireless Communication Mobile Computing. 2011;11(12):1677-1688.

Al-Turjman F, Abdulkadir AA. *Smart-grid and solar energy harvesting in the IoT era: an overview*. Concurrency Computat Pract Exper. 2018.https://doi.org/10.1002/cpe.4896

AlienmanTechnologyLLC. (2015). Where My Droid. Retrieved from play.google.com: https://play.google.com/store/apps/details?id=com.alienmanfc6.wheresmyandroid&hl=en

Android.(2018). Services overview. Retrieved from https://developer.android.com/guide/components/services

Android mobile application activities lifecycle. https://developer.android.com/reference/android/app/Activity

Android SDK. http://developer.android.com/sdk/index.html. Accessed: February, 2013

Andonoska, A., & Jakimoski, K. (2018). Performance Evaluation of Mobile Applications. https://www.researchgate.net/publication/337437805

B. D. Deebak, et. al., “Privacy-Preserving in Smart Contracts Using Blockchain and Artificial Intelligence for Cyber Risk Measurements”, *Journal of Information Security and Applications*, 2021

Balsamiq Studios. 2018. basalmiq. (2018). https://balsamiq.com/

Baro X, Escalera S, Vitria J, Pujol O, Radeva P. Traffic sign recognition using evolutionary AdaBoost detection and Forest-ECOC classification. *IEEE Trans Intell Transp Syst*. 2009;10:113-126.

Call-Em-All. 2018. Material-UI. (2018). https://material-ui-next.com/

Canzian, L., & Musolesi, M. (2015). Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. *In Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing* (pp. 1293–1304). New York, NY: ACM Press

Carlson, J. A., Jankowska, M. M., Meseck, K., Godbole, S., Natarajan, L., Raab, F. . . . Kerr, J. (2015). Validity of PALMS GPS scoring of active and passive travel compared to SenseCam. *Medicine and Science in Sports and Exercise, 47*, 662–667.

F. Al-Turjman, et; al., "Ubiquitous Cloud-based Monitoring via a Mobile App in Smartphones: An Overview", *IEEE International Conference on Smart Cloud*, New York, USA, pp. 1-6, 2016

F. Al-Turjman, et. al., "Seamless Authentication: For IoT-Big Data Technologies in Smart Industrial Application Systems", *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2919-2927, 2020

F. Ullah, et. al., "*A conceptual framework for blockchain smart contract adoption to manage real estate deals in smart cities*", Neural Computing and Applications, 2021.

Google Places API http://code.google.com/apis/maps/documentation/places/

Hardy, B. (2014). The Big Nerd Ranch Guide

Harris, C., et al., *mSpace: Exploring The Semantic Web. A Technical Report in Support of the mSpace software framework.* 2004, IAM Group, University of Southampton: Southampton. p. 98.

Huang, Y., Xiong, H., Leach, K., Zhang, Y., Chow, P., Fua, K. . . . Barnes, L. E. (2016). Assessing social anxiety using GPS trajectories and point-of-interest data. *In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (pp. 898–903)*. New York, NY: ACM Press.

James, P., Jankowska, M., Marx, C., Hart, J. E., Berrigan, D., Kerr, J., . . ., & Laden, F. (2016). BSpatial energetics: Integrating data from GPS, accelerometry, and GIS to address obesity and inactivity. *American Journal of Preventive Medicine*, 51, 792–800.

K. Kumar, et. al, "A Reinforcement Learning Optimization for Future Smart Cities Using Software Defined Networking", *International Journal of Machine Learning and Cybernetics*, 2020. DOI. 10.1007/s13042-020-01245-w

Jankowska, M. M., Schipperijn, J., & Kerr, J. (2015). *A framework for using GPS data in physical activity and sedentary behaviour studies.* Exercise and Sport Sciences Reviews, 43, 48–56.

Location Manager APIs– Android Developer http://developer.android.com/reference/android/location/LocationManager.html

Meier, R. (2012). *Professional Android for Application Development*. ISBN: 978-1-118-10227-5.

Material Design. https://material.io/design/

Max, W., Alistair R., Daniel A., Alisdair O., *mSpace Mobile: A Mobile Application for the Semantic Web*. IAM Research Group School of Electronics and Computer Science University of Southampton, SO17 1BJ, http://mspace.fm/

N. Kadjouh; et. al, "A Dominating Tree based Leader Election Algorithm for Smart Cities IoT Infrastructure", *Springer Mobile Networks and Applications*, 2020

Pizarro, A. N., Schipperijn, J., Ribeiro, J. C., Figueiredo, A., Mota, J., & Santos, M. P. (2017). Gender differences in the domain-specific contributions to moderate-to-vigorous physical activity, accessed by GPS. *Journal of Physical Activity and Health*, 14, 474–478.

S. Malchi, et. al., "A Trust-based Fuzzy Neural Network for Smart Data Fusion in Internet of Things", *Elsevier Computers & Electrical Engineering Journal*, vol. 89, pp. 106901, 2021

Schraefel, m.c., M. Karam, and S. Zhao. *mSpace: interaction design for userdetermined, adaptable domain exploration in hypermedia*. in *International Workshop on Adaptive Hypermedia*. 2003. Nottingham, UK.

Salman, K., Ali, H., & Saleem, S. (2015). Research on Auto Mobile-Pc Upload Images Application through Bluetooth Using Java. VAWKUM Transaction on Computer Sciences. Retrieved from http://vfast.org/journals/index.php/VTCS/article/view/356

Saleem, S., Khan, I., khan, S., & Rahman, A. (2015). Comparison of Cooperative Diversity Protocols in Various Relay Locations through Network Coding. VFAST Transactions on Software Engineering. Retrieved from http://vfast.org/journals/index.php/VTSE/article/view/357

van de Beek J, Sandell M, Börjesson P. ML estimation of timing and frequency offset in OFDM systems. *IEEE Trans Signal Process*. 1997; 45:1800-1805.

Van Kleek, M., Liccardi, I., Binns, R., Zhao, J., Weitzner, D. J., & Shadbolt, N. (2017). Better the devil you know: Exposing the data sharing practices of smartphone apps. *In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 5208–5220). New York, NY: ACM Press.

Vision Mobile. Cross-platform developer tools 2012: Bridging the worlds of mobile apps and the web. February, 2012.

APPENDICES

Appendix 1: AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
package="app.neuplaces"
tools:ignore="GoogleAppIndexingWarning">
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-feature android:name="android.hardware.location.gps" />
<application
android:name="app.neuplaces.data.ThisApplication"
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:largeHeap="true"
android:usesCleartextTraffic="true"
android:supportsRtl="true"
android:theme="@style/AppTheme">
```

```
<uses-library android:name="org.apache.http.legacy" android:required="false"/>

<activity android:name="app.neuplaces.ActivitySplash">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity

android:name="app.neuplaces.ActivityMain"

android:label="@string/app_name" />

<activity

android:name="app.neuplaces.ActivitySetting"

android:label="@string/activity_title_settings"

android:theme="@style/AppThemeSetting" />

<activity

android:name="app.neuplaces.ActivityMaps"

android:label="@string/activity_title_maps" />

<activity android:name="app.neuplaces.ActivityPlaceDetail" />

<activity android:name="app.neuplaces.ActivityFullScreenImage" />

<activity android:name="app.neuplaces.ActivitySearch" />

<activity android:name="app.neuplaces.ActivityNewsInfo" />

<activity android:name="app.neuplaces.ActivityNewsInfoDetails" />

<activity

android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|uiMode|screenS
ize|smallestScreenSize"

android:theme="@android:style/Theme.Translucent" />
```

```
<meta-data
android:name="com.google.android.gms.version"
android:value="@integer/google_play_services_version" />
<uses-library android:name="com.google.android.maps" />
<meta-data
android:name="com.google.android.maps.v2.API_KEY"
android:value="@string/api_key" />
</application>
</manifest>
```

Appendix 2: AdapterPlaceGrid.java

```
package app.neuplaces.adapter;
import android.content.Context;
import androidx.recyclerview.widget.RecyclerView;
import androidx.recyclerview.widget.StaggeredGridLayoutManager;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import com.balysv.materialripple.MaterialRippleLayout;
import java.util.ArrayList;
import java.util.List;
import app.neuplaces.R;
import app.neuplaces.data.Constant;
import app.neuplaces.model.Place;
import app.neuplaces.utils.Tools;
public class AdapterPlaceGrid extends RecyclerView.Adapter<RecyclerView.ViewHolder> {
private final int VIEW_ITEM = 1;
    private final int VIEW_PROG = 0;
    private boolean loading;
    private Context ctx;
    private List<Place>items = new ArrayList<>();
    private OnLoadMoreListener onLoadMoreListener;
    private OnItemClickListener onItemClickListener;
    private int lastPosition = -1;
    private boolean clicked = false;
    public interface OnItemClickListener {
void onItemClick(View view, Place viewModel);}
public void setOnItemClickListener(OnItemClickListener onItemClickListener) {
this.onItemClickListener = onItemClickListener;}
public class ViewHolder extends RecyclerView.ViewHolder {
public TextView title;
        public ImageView image;
        public TextView distance;
        public LinearLayout lyt_distance;
        public MaterialRippleLayout lyt_parent;
        public ViewHolder(View v) {
super(v);
title = (TextView) v.findViewById(R.id.title);
image = (ImageView) v.findViewById(R.id.image);
distance = (TextView) v.findViewById(R.id.distance);
lyt_distance = (LinearLayout) v.findViewById(R.id.lyt_distance);
lyt_parent = (MaterialRippleLayout) v.findViewById(R.id.lyt_parent);}}
public static class ProgressViewHolder extends RecyclerView.ViewHolder {
public ProgressBar progressBar;
        public ProgressViewHolder(View v) {
super(v);
progressBar = (ProgressBar) v.findViewById(R.id.progressBar1);}}
```

```
public AdapterPlaceGrid(Context ctx, RecyclerView view, List<Place> items) {
this.ctx = ctx;
        this.items = items;
lastItemViewDetector(view);}
@Override
public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        RecyclerView.ViewHolder vh;
        if (viewType == VIEW_ITEM) {
            View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_place,
                    parent, false);
vh = new ViewHolder(v);
} else {
            View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_loading,
                    parent, false);
vh = new ProgressViewHolder(v);}
return vh;}
@Override
public void onBindViewHolder(RecyclerView.ViewHolder holder, int position) {
if (holder instanceof ViewHolder) {
            ViewHolder vItem = (ViewHolder) holder;
            final Place p = items.get(position);
vItem.title.setText(p.name);
Tools.displayImageThumb(ctx, vItem.image, Constant.getURLimgPlace(p.image), 0.5f);
            if (p.distance == -1) {
                vItem.lyt_distance.setVisibility(View.GONE);
} else {
                vItem.lyt_distance.setVisibility(View.VISIBLE);
vItem.distance.setText(Tools.getFormatedDistance(p.distance));}
setAnimation(vItem.lyt_parent, position);
vItem.lyt_parent.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(final View v) {
if (!clicked &&onItemClickListener != null) {
clicked = true;
onItemClickListener.onItemClick(v, p);
}}});
clicked = false;
} else {
        ((ProgressViewHolder) holder).progressBar.setIndeterminate(true);}
if (getItemViewType(position) == VIEW_PROG) {
        StaggeredGridLayoutManager.LayoutParams layoutParams =
              (StaggeredGridLayoutManager.LayoutParams)
              holder.itemView.getLayoutParams();
layoutParams.setFullSpan(true);
} else {
        StaggeredGridLayoutManager.LayoutParams layoutParams =
              (StaggeredGridLayoutManager.LayoutParams)
              holder.itemView.getLayoutParams();
layoutParams.setFullSpan(false);}}
@Override
public int getItemCount() {
return items.size();}
@Override
public int getItemViewType(int position) {
return this.items.get(position) != null ? VIEW_ITEM : VIEW_PROG;}
```

```
@Override
public long getItemId(int position) {
return position;}
private void setAnimation(View viewToAnimate, int position) {
if (position >lastPosition) {
            Animation animation = AnimationUtils.loadAnimation(ctx, R.anim.slide_in_bottom);
viewToAnimate.startAnimation(animation);
lastPosition = position;}}
public void insertData(List<Place> items) {
        setLoaded();
        int positionStart = getItemCount();
        int itemCount = items.size();
        this.items.addAll(items);
notifyItemRangeInserted(positionStart, itemCount);}
public void setLoaded() {
loading = false;
        for (int i = 0; i < getItemCount(); i++) {
if (items.get(i) == null) {
items.remove(i);
notifyItemRemoved(i);}}}
public void setLoading() {
if (getItemCount() != 0) {
this.items.add(null);
notifyItemInserted(getItemCount() - 1);}}
public void resetListData() {
this.items = new ArrayList<>();
notifyDataSetChanged();}
```

```
public void setOnLoadMoreListener(OnLoadMoreListener onLoadMoreListener) {

this.onLoadMoreListener = onLoadMoreListener;}

private void lastItemViewDetector(RecyclerView recyclerView) {

if (recyclerView.getLayoutManager() instanceof StaggeredGridLayoutManager) {

final StaggeredGridLayoutManager layoutManager = (StaggeredGridLayoutManager)
                 recyclerView.getLayoutManager();

recyclerView.addOnScrollListener(new RecyclerView.OnScrollListener() {

@Override

public void onScrolled(RecyclerView recyclerView, int dx, int dy) {

super.onScrolled(recyclerView, dx, dy);

                int lastPos =
                 getLastVisibleItem(layoutManager.findLastVisibleItemPositions(null));

                if (!loading && lastPos == getItemCount() - 1 &&onLoadMoreListener != null)
                 {

int current_page = getItemCount() / Constant.LIMIT_LOADMORE;

onLoadMoreListener.onLoadMore(current_page);

loading = true;}}});}}

public interface OnLoadMoreListener {

void onLoadMore(int current_page);}

private int getLastVisibleItem(int[] into) {

int last_idx = into[0];

      for (int i : into) {

if (last_idx < i) last_idx = i;}

return last_idx;}}
```

Appendix 3: AdapterSuggestionSearch.java

```
package app.neuplaces.adapter;

import android.content.Context;

import android.content.SharedPreferences;

import androidx.recyclerview.widget.RecyclerView;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.LinearLayout;

import android.widget.TextView;

import com.google.gson.Gson;

import java.io.Serializable;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import app.neuplaces.R;

public class AdapterSuggestionSearch extends
                RecyclerView.Adapter<AdapterSuggestionSearch.ViewHolder> {

private static final String SEARCH_HISTORY_KEY = "_SEARCH_HISTORY_KEY";

    private static final int MAX_HISTORY_ITEMS = 5;

    private List<String>items = new ArrayList<>();

    private OnItemClickListener onItemClickListener;

    private SharedPreferences prefs;

    public class ViewHolder extends RecyclerView.ViewHolder {

public TextView title;

public LinearLayout lyt_parent;
```

```
    public ViewHolder(View v) {
super(v);
title = (TextView) v.findViewById(R.id.title);
lyt_parent = (LinearLayout) v.findViewById(R.id.lyt_parent);}}
public void setOnItemClickListener(OnItemClickListener onItemClickListener) {
this.onItemClickListener = onItemClickListener;}
public AdapterSuggestionSearch(Context context) {
prefs = context.getSharedPreferences("PREF_RECENT_SEARCH",
          Context.MODE_PRIVATE);
    this.items = getSearchHistory();
Collections.reverse(this.items);}
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_suggestion,
          parent, false);
ViewHolder vh = new ViewHolder(v);
    return vh;}
@Override
public void onBindViewHolder(ViewHolder holder, int position) {
final String p = items.get(position);
    final int pos = position;
holder.title.setText(p);
holder.lyt_parent.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(final View v) {
onItemClickListener.onItemClick(v, p, pos);} });}
@Override
public int getItemCount() {
```

```
return items.size();}
public interface OnItemClickListener {
void onItemClick(View view, String viewModel, int pos);}
public void refreshItems() {
this.items = getSearchHistory();
Collections.reverse(this.items);
notifyDataSetChanged();}
private class SearchObject implements Serializable {
public SearchObject(List<String> items) {
this.items = items;}
public List<String>items = new ArrayList<>();}
public void addSearchHistory(String s) {
    SearchObject searchObject = new SearchObject(getSearchHistory());
    if (searchObject.items.contains(s)) searchObject.items.remove(s);
searchObject.items.add(s);
    if (searchObject.items.size() >MAX_HISTORY_ITEMS) searchObject.items.remove(0);
String json = new Gson().toJson(searchObject, SearchObject.class);
prefs.edit().putString(SEARCH_HISTORY_KEY, json).apply();}
private List<String>getSearchHistory() {
    String json = prefs.getString(SEARCH_HISTORY_KEY, "");
    if (json.equals("")) return new ArrayList<>();
SearchObject searchObject = new Gson().fromJson(json, SearchObject.class);
    return searchObject.items;}}
```

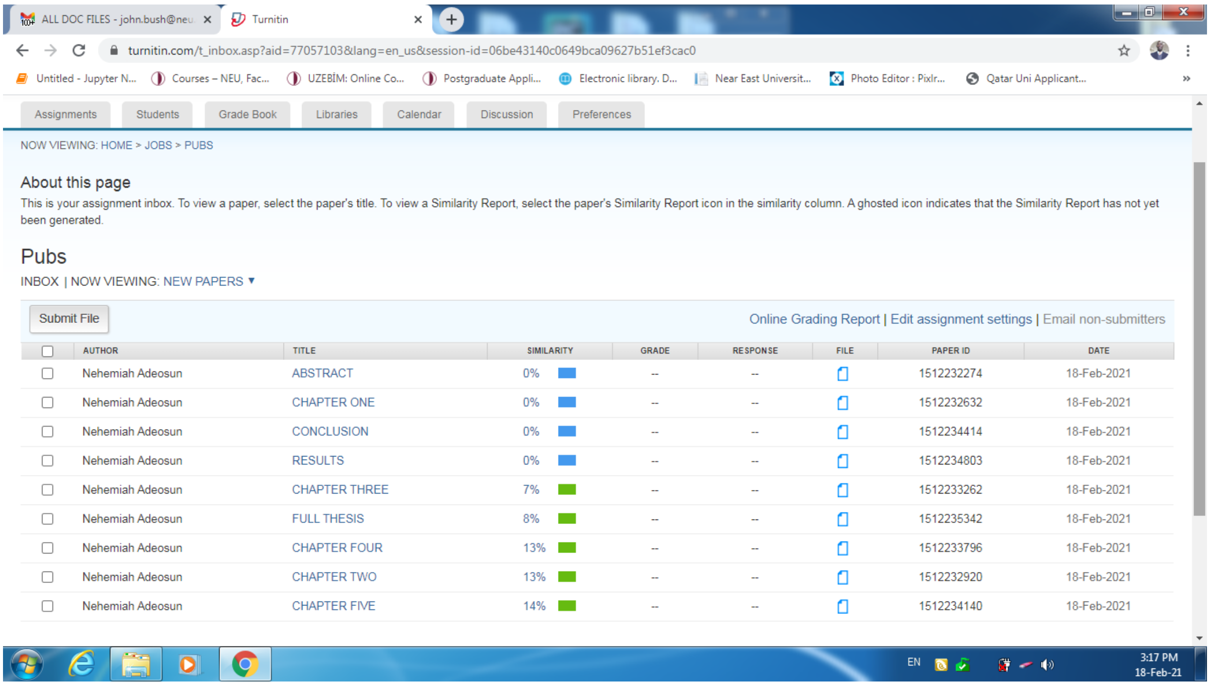
Appendix 4: AppConfig.java

```
package app.neuplaces.data;

public class AppConfig {

public static final boolean ENABLE_GDPR = true;

public static final boolean ENABLE_NEWS_INFO = true;

public static final boolean IMAGE_CACHE = true;

public static final boolean LAZY_LOAD = false;

public static final boolean ENABLE_ANALYTICS = true;

public static final boolean REFRESH_IMG_NOTIF = true;

public static final boolean SORT_BY_DISTANCE = true;

public static final String DISTANCE_METRIC_CODE = "KILOMETER";

public static final String DISTANCE_METRIC_STR = "Km";

public static final boolean THEME_COLOR = false;}
```

Appendix 5: Constant.java

```
package app.neuplaces.data;
public class Constant {
public static String WEB_URL = "http://mountzionmovies.com/neuplaces/";
public static final double city_lat = 35.22789255;
public static final double city_lng = 33.32274106;
public static String getURLimgPlace(String file_name) {
return WEB_URL + "uploads/place/" + file_name;}
public static String getURLimgNews(String file_name) {
return WEB_URL + "uploads/news/" + file_name;}
public static final int LIMIT_PLACE_REQUEST = 40;
public static final int LIMIT_LOADMORE = 40;
public static final int LIMIT_NEWS_REQUEST = 40;
public static int LOAD_IMAGE_NOTIF_RETRY = 3;
public static final String LOG_TAG = "CITY_LOG";
public enum Event {
FAVORITES,
THEME,
NOTIFICATION,
REFRESH}}
```

Appendix 6: Similarity report



NOW VIEWING: HOME > JOBS > PUBS

About this page

This is your assignment inbox. To view a paper, select the paper's title. To view a Similarity Report, select the paper's Similarity Report icon in the similarity column. A ghosted icon indicates that the Similarity Report has not yet been generated.

Pubs

INBOX | NOW VIEWING: NEW PAPERS ▾

Submit File
Online Grading Report | Edit assignment settings | Email non-submitters

□	AUTHOR	TITLE	SIMILARITY	GRADE	RESPONSE	FILE	PAPER ID	DATE
□	Nehemiah Adeosun	ABSTRACT	0% 🟦	--	--	🗋	1512232274	18-Feb-2021
□	Nehemiah Adeosun	CHAPTER ONE	0% 🟦	--	--	🗋	1512232632	18-Feb-2021
□	Nehemiah Adeosun	CONCLUSION	0% 🟦	--	--	🗋	1512234414	18-Feb-2021
□	Nehemiah Adeosun	RESULTS	0% 🟦	--	--	🗋	1512234803	18-Feb-2021
□	Nehemiah Adeosun	CHAPTER THREE	7% 🟩	--	--	🗋	1512233262	18-Feb-2021
□	Nehemiah Adeosun	FULL THESIS	8% 🟩	--	--	🗋	1512235342	18-Feb-2021
□	Nehemiah Adeosun	CHAPTER FOUR	13% 🟩	--	--	🗋	1512233796	18-Feb-2021
□	Nehemiah Adeosun	CHAPTER TWO	13% 🟩	--	--	🗋	1512232920	18-Feb-2021
□	Nehemiah Adeosun	CHAPTER FIVE	14% 🟩	--	--	🗋	1512234140	18-Feb-2021

Appendix 7: Ethical Approval letter

NEAR EAST UNIVERSITY

ETHICAL APPROVAL DOCUMENT

Date: 27/02/2021

To the Institute of Graduate Studies

For the thesis project entitled as "Analysis of LTE Downlink Performance under Different Mobility Conditions", the researchers declare that they did not collect any data from human/animal or any other subjects. Therefore, this project does not need to go through the ethics committee evaluation.

Title: Prof. Dr.

Name Surname: Fadi Al-Turjman

Signature: F. Alturjman

Role in the Research Project: Supervisor