A MOBILE APPLICATION FOR FINDING SHORTEST ROUTE IN ONLINE DELIVERY

(ROde)

NUR FAKHIRA BT MUHAMMAD NAWAWI

BACHELOR OF COMPUTER SCIENCE (COMPUTER NETWORK SECURITY) WITH HONOURS

UNIVERSITI SULTAN ZAINAL ABIDIN

2021

A MOBILE APPLICATION FOR FINDING SHORTEST ROUTE IN ONLINE DELIVERY (ROde)

NUR FAKHIRA BT MUHAMMAD NAWAWI

BACHELOR OF COMPUTER SCIENCE (COMPUTER NETWORK SECURITY) WITH HONOURS

Universiti Sultan Zainal Abidin

2021

DECLARATION

I hereby declare that this report is based on my original work except for quotations and citations, which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Sultan Zainal Abidin or other institutions.

_____

Name: …….

Date: ……

CONFIRMATION

This is to confirm that: The research conducted and the writing of this report was under my supervision**.**

_____

Name: …….

Date: ……

DEDICATION

Bismillahirrahmanirrahim,

Alhamdulillah. Thanks to Allah SWT, for giving me the opportunity to complete my Final Year Project entitled A Mobile Application for Finding Shortest Route in Online Delivery. Firstly, I would like to express my deepest thanks to my supervisor, Dr. Aznida Hayati bt Zakaria@Mohamad for her guidance and support. Not to forget, I also to thanks all my panel for the precious comment and suggestions pertaining on this project. Last but not least, I would like to extend my thanks to parents and friends that always support and encourage me to complete this project.

ABSTRACT

Movement Control Order (MCO) has been implemented by the federal government of Malaysia in response to the COVID-19 pandemic in the country. Due to this, people prefer online delivery services or applications in solving their daily affairs. Companies engaged in online delivery services face the problem of the delivery process because people are focused on using their services. They have been receiving a high volume of orders for delivery services during the MCO. In this case, distance and time should be considered for the online delivery process to ensure that the services provide satisfaction to the public. An application needs to be developed to achieve this goal. This application is able to provide optimal travel routes so that online delivery services can be done in a time that is not only minimal, but also within close travel distance. In this application, the A* algorithm will be used. This application is expected to help online delivery services in their delivery services.

ABSTRAK

*Perintah Kawalan Pergerakan (MCO) telah dilaksanakan oleh kerajaan pusat Malaysia sebagai tindak balas terhadap wabak COVID-19 di negara ini. Hal ini demikian, ramai pengguna yang terdiri daripada rakyat Malaysia lebih gemar menggunakan perkhidmatan atau aplikasi penghantaran secara talian dalam menyelesaikan urusan harian mereka. Oleh itu, Syarikat yang terlibat dalam perkhidmatan penghantaran dalam talian menghadapi masalah dalam proses penghantaran kerana perkhidmatan mereka telah menjadi tumpuan utama kepada pengguna. Mereka telah menerima pesanan yang tinggi untuk perkhidmatan penghantaran selama MCO. Dalam kes ini, jarak dan waktu harus dipertimbangkan untuk proses penghantaran dalam talian untuk memastikan bahawa perkhidmatan memberikan kepuasan kepada orang ramai. Aplikasi perlu dibangunkan untuk mencapai tujuan ini. Aplikasi ini dapat memberikan laluan perjalanan yang optimum sehingga perkhidmatan pengiriman dalam talian dapat dilakukan dalam waktu yang tidak hanya minimum, tetapi juga dalam jarak perjalanan yang lebih optimum. Dalam aplikasi ini, algoritma A * akan digunakan. Aplikasi ini diharapkan dapat membantu perkhidmatan penghantaran dalam talian dalam perkhidmatan penghantaran mereka.*

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS/ TERMS/ SYMBOLS

ROde	Route for Online delivery
API	Application Programming Interface
OS	Operating System
OFD	Online Food Delivery
3PL	Third-party Logistic
BCE	Before the Common Era
AI	Artificial Intelligent
IDE	Integrated Development Environment
NDK	Native Development Kit
JDK	Java Development Kit
MADLC	Mobile Application Development Life Cycle
UI	User Interface
UX	User Experience
JSON	Javascript Object Notation
ID	Identity
KPI	Key Performance Indicator

CHAPTER 1

INTRODUCTION

1.1 Project Background

This chapter aims to describe the project background, problem statement, objectives, and scope. The aim of this research is to help online delivery service to find the shortest route for their services.

Online delivery services are an important industry as technology develops. In this case, companies that contribute to delivery have many problems in the process of delivery services, one example is time and distance that are not optimal because the route information still lacking. This information includes the shortest distance between two nodes.

A mobile application for finding the shortest route in online delivery, Route in Online delivery (ROde) enables users to choose the shortest route they will take. The online delivery service will have information in advance. currently, some of the online delivery services do not have any choice to choose different routes to take when going from one location to another location even some do not provide maps for the location at all.

Besides, the Route in Online delivery (ROde) application enables online delivery services to choose the best route to get minimal distance with minimal time. This project

will be used Android Studio to develop the Route in Online delivery (ROde) application, and Google Map Android APIs JavaScript V2 will be used to get a location on maps. Moreover, the A* algorithm is one of the best shortest path searches, so for this application, the A* algorithm will be implemented to help for finding the shortest path. It is hoped that the development of this application can solve the problem.

1.2 Problem Statement

Practically, often confuse online delivery services lack of information include the shortest route via city from the origin to the destination. Online delivery service usually, have fewer options to choose what route to take from one location to another location. Moreover, they have fixed time to travel, for example, they have only 30 minutes to deliver the order to the customer, due to this they need to find the shortest route for them shorten the traveling time and get the minimal distance.

1.3 Objective

Generally, the objective is to develop a mobile application to identify the shortest route in online delivery service from a particular source to a destination.

1. To design a mobile application for Route in Online delivery (ROde) that can provide details on the route to choose from the point of origin location to the destination.
2. To study and implement A* algorithm in Rode apps to provide optimal routes.
3. To test and evaluate the effectiveness of the mobile application, so online delivery service will have an option to choose the best shortest route.

1.4 Scope

The scope of this study divided into 4:

1. Admin. The role of an admin is login account of the application to check a monthly report, manage users, and arrange a database
2. User. The user need to register or login before they can search the location where they want to go and get the shortest route to their destination. They have an option to choose shortest distance path through a different route.
3. Android based. This application will develop by using Android Studio and for sure use Android OS. This application will be used in mobile devices.
4. Google Map Android APIs JavaScript V2 will be used to get the location on maps. This application will be implement the A*algorithm.

1.5 Expected result

This application is expected to be implemented in a mobile application. The apps can use A* algorithm to list out the choice of route to get the shortest route to reach the destination. At the same time, the user also can minimize their time travel to the destination by using this application. These results represented to achieve the objective project which is to design a mobile application for finding the shortest route in online delivery. Supposing this application will assist the online delivery service in finding the shortest route for delivery services.

1.6 Limitation

There are a few of the limitation of work for this application:

- This application can be used by Android users only.
- This application can only be used through internet connection only
- This application only able used in Pulau Pinang.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses the literature review, searching, collecting, and analysing any issues related to previous journals, research papers, articles, internet or, other resources. The information that has been gathered is about what method and what contributions that the research has been achieved in improving the uses of the applications. The purpose of this literature review is to provide a general overview of a mobile application for finding the shortest route in online delivery.

2.2 Online Food Delivery Services

There is an emerging new wave, the online food delivery (OFD) service, within the food and beverage industry in Malaysia. Online food ordering is a modern dining out, not only confined to take-away and eating out. There are numerous food distribution companies in Malaysia, with many providing food delivery services online. Food Panda, the first distribution company to be actively introduced in Malaysia, is among the firms. In urban cities such as Kuala Lumpur, Penang, and Johor Bharu, the majority of food delivery services are based. This is understandable because food delivery services face the challenge of location and coverage constraints, unlike other e-commerce services that are easier to scale with the reliance on 3PL delivery, while retaining high customer loyalty with on-demand delivery. Perhaps this is the reason that there are only a few strong players in this industry without anyone being entirely dominant [6].

2.3 A* algorithm

A* algorithm is based on two conventional algorithms, the Dijkstra algorithm and the Greedy Best-First Search algorithm. A* is like Dijkstra's algorithm in that it finds the shortest path without fail and it is like Greedy Best First Search in that it uses a heuristic function to estimate the distance to the goal. A* is a successful algorithm because it has the good qualities of both Dijkstra's and Best First-Search algorithms but it has the drawbacks of none. It is as fast as Best-First- Search algorithm and finds a path as good as Dijkstra's algorithm. It unites parts of Dijkstra's algorithm which uses the exact distance g(n) from the start point to any vertex n, and of the Best-First-Search algorithm which uses an estimated distance h(n) from any node n to the target. A* brings about an equilibrium between these two algorithms as its frontier expands from the starting point to the target. In its main loop, the algorithm repeatedly examines the vertex n that has the lowest value of f(n), where f(n) = g(n) + h(n) [5].

2.3.1 A* algorithm process

A* algorithm needs to set up two tables an OPEN table and a CLOSE table. The Open Table saves all the nodes that have been generated but have not yet been examined. The CLOSE table records the nodes that have been visited [7]. The A* algorithm pseudocode is shown in Figure 2.3.1.

1. Add the starting node to the open list
2. Repeat the following steps:
   a. Look for the node that has the lowest f on the open list. Refer to this node as the current node
   b. Switch it to the closed list
   c. For each reachable node from the current node:
      i. If it is on the closed list, ignore it
      ii. If it isn't on the open list add it to the open list. Make the current node the parents of this node. Record the f, g and h value of this node
      iii. If it is on the open list already, check to see if this is a better path. If so, change its parents to the current node, and recalculate the f and g value
   d. Stop when
      i. Add the target node to the closed list
      ii. Fail to find the target, node, and the open list is empty
      iii. Trace backwards from the target node to the starting node. That is your path.

Figure 2.3.1: Pseudocode A* algorithm

2.3.2 Euclidean distance

The Euclidean distance is the measure of the Euclidean space distance between 2 points. Euclid, a mathematician from Greece around 300 B.C.E., invented Euclidean space to learn the relationship between angle and distance. This Euclidean is related to the theorem of Pythagoras and is commonly applied to dimensions 1, 2, and 3. When extended to a higher dimension, but also simple on 1 dimension. The Euclidean distance formula will be used to get the value of heuristic, h(n) [12].

The formula of Euclidean is:

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

2.4 Route optimization

The problem of routing optimization on networks consists of determining paths (simple paths, closed paths, tours, or walks) efficiently between network nodes. In app lications as diverse as logistics, transportation, computer networking, internet routing, to name a few, these problems and their solutions have proven to be important elemen ts for solving many real-world decisions. The deterministic version of many routing optimization problems (such as shortest path problems, traveling salesman problems, vehicle routing problems) has been studied extensively over many decades. In view of the recognized practical importance of incorporating uncertainty, the uncertain versions of routing problems have also attracted increasing attention. Researchers have selected out different selection criteria. One intuitive and well-discussed way is to select a route with the largest probability of arriving on time. For logistics and distribution companies,

the rising cost of fuel means that they need to become efficient in the way they plan their transportation routes and schedule. Traditional route planning methods do not address real-time events that affect business daily. Route planning must be able to respond quickly to any event to ensure the lowest cost of transport in order to meet customer requirements for short notices, route availability and vehicle issues. The basis for route optimization is the use of models to describe the transport network that needs to be planned [2].

2.4.1 Optimizing using Heuristics Functions

The heuristic search function is a key field of AI and has been extensively used in game-play, path-planning, and agent control. The definition of the heuristic search function can be simplified as a set of guidelines for the selection of state-space pathways which resulted in an acceptable possible resolution by decreasing the number of options movements [8]. As we have seen before, Dijkstra's algorithm is the best algorithm available to us to find the shortest path between two endpoints but it is not always the optimal solution as it examines all the possible states. So, the A* algorithm makes use of the heuristic approach to improve its efficiency. It can do so because the heuristic approach only focuses on the states that appear the most suitable options. It does not go on to examine all the possible approaches. It works in the following way where the heuristic function estimates the cost from any node on the graph to the respective destination. If this cost is equal to the actual cost, only then the function proceeds with the nodal path. Thus, it can make the pathfinding process much quicker [5].

2.5 A* algorithm vs Dijkstra algorithm

A* search algorithm is an extension of Dijkstra's algorithm without heuristics. Both the algorithm give the shortest path but the A* gives a faster and efficient result. Dijkstra algorithm is slower as compared to A* because Dijkstra repeatedly attempts to improve an initial approximation (cost) of each node. Over this, it takes more time to reach the target node [9].

Based on the article [9] to support the difference between these two algorithms, figure 2.5.1 and figure 2.5.2 can be considered.

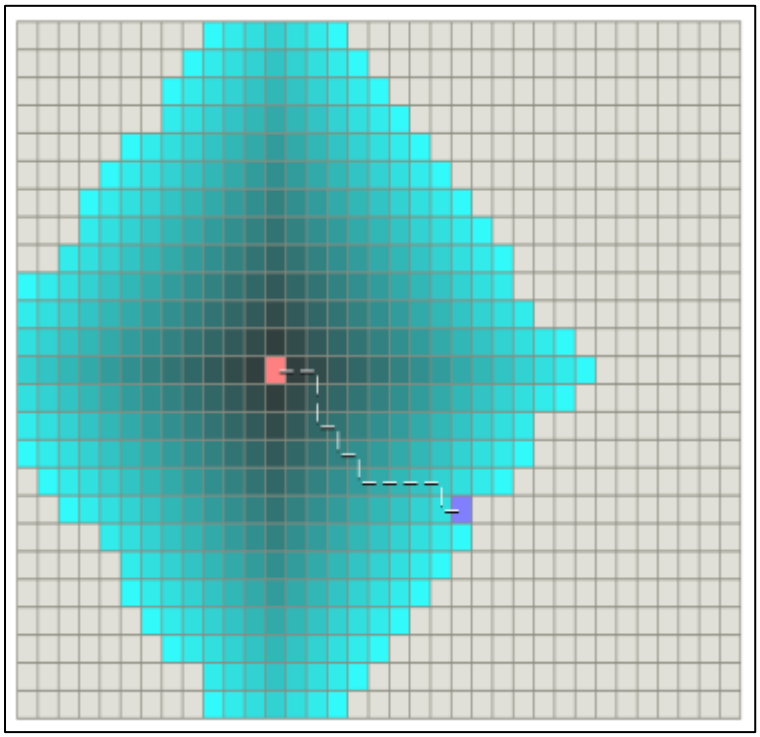


Figure 2.5.1: Route through Dijkstra's algorithm

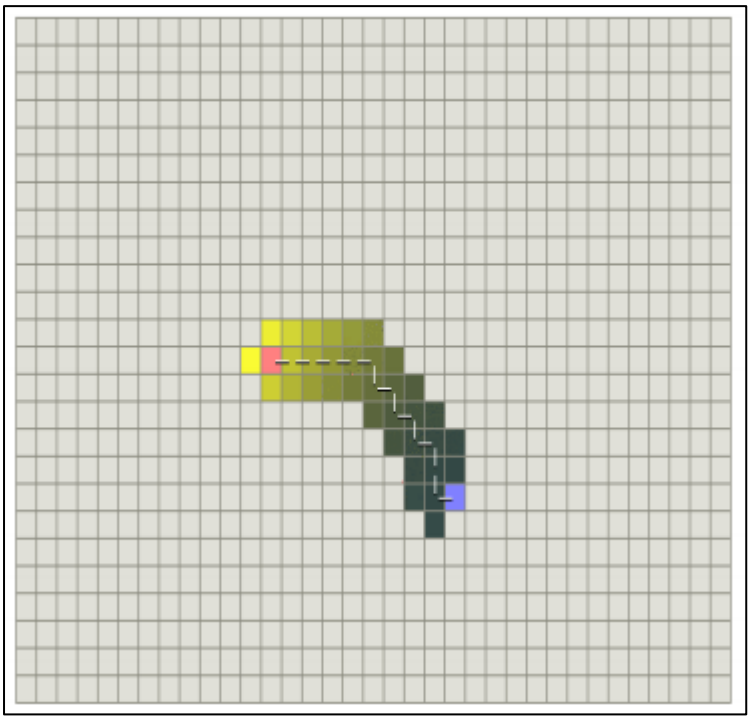


Figure 2.5.2: Route through A* Algorithm

As a conclusion, we have seen the difference between both the algorithm A* algorithm is much more similar to Dijkstra's, the only difference in the two algorithms is that A* gives a better path by using a heuristic function, while Dijkstra only explores all possible paths. A* accomplishes better performance by using heuristics to guide its search and gives the optimal result much faster [9].

2.6 Android based Mobile Application Development

In the advancing world of technology, mobile applications are involving at a fast pace to give users a uniquely personal experience. Android Mobile Application Development is based on Java Language Codes, allowing developers to write Java Language Codes. These codes can control mobile devices via Google-enable Java libraries provided in the Google Android SDK. As a developer, we only need to call the relevant Application Programming interface (APIs) to develop applications on the Android platform as we build the product the interface via a layered approach. As of early 2015, 48.61 percent of mobile devices are running on Android OS, 11.04 percent on iOS, 14 percent on Windows, and 26.34 percent on other operating systems. This made a clear distinction on the platform to create a mobile application, apart from the reality that Android application courses are easier to access, creating cheaper platforms for Android applications learning, an open marketplace, and simple to inculcate for a mobile application beginner. Mobile and e-commerce application are tools for accessing the internet and for buying product and services. This application are constantly involving due to the high rate of technological advances being made. With digital technology rapid development in the last decade, more and more corporation build their e-business to absorb more customers in worldwide without geographical limitation [2].

2.6.1 Android Studio

The Android Studio is a tool for Android app development, IntelliJ IDEA-based official Integrated Development Environment (IDE). Every aspect of the IntelliJ IDEA has been designed to maximize developer productivity. Intelligent coding assistance and ergonomic design together make growth not only efficient but also enjoyable. Android Studio offers even more features that increase your productivity when developing Android apps, in addition to IntelliJ's powerful code editor and developer tools, such as [13]:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where it can develop for all Android devices
- Apply Changes to push code and resource changes to run the app without restarting the app
- Code templates and GitHub integration to help build common app features and import sample code
- Extensive testing tools and frameworks
- Lint Output capture tools, reliability, version compatibility, and other issues
- C++ and support for NDK
- Integrated Google Cloud Platform support, making it simple to incorporate Google Cloud Messaging and App Engine

2.6.2 Google maps

Google Maps API as a service tool from Google application for the development of a web application and mobile application (Android, iOS) [15]. The JavaScript API allows you to customize your own content and image maps for viewing on web pages and mobile devices. The JavaScript API maps contain four simple types of maps (roadmap, satellite, hybrid and terrain) that can be changed using layers and styles, controls and events, and different utilities and libraries [14]. Many applications can access google map and services via the internet. Google Map API composes of 6 functions: Google Maps Geocoding API, Google Maps Geolocation API, Google Places API, Google Maps Distance Matrix API, Google Maps Directions API, Google Maps Roads API [15].

2.7 Comparison between methods

Table 2.7.1: Comparison between methods

Author	Title	Description	Methodology
O.E Oduwole[1,] O.S. Asaolu[1*] and M.S. Osigbemeh[2]	SHORTEST ROUTE: A MOBILE APPLICATION FOR ROUTE OPTIMIZING USING DIGITAL MAP (2019)	Developed a mobile application in java, HTML and google SDKs, find the shortest route between various numbers of locations enumerated on digital maps on a smart device [2]	Nearest Neighbour Algorithm
Hagai Nuansa Ginting[1], Andrew Brian Osmond[2] and Annisa Aditsania[3]	Item Delivery Simulation Using Dijkstra Algorithm for Solving Traveling Salesman Problem (2019)	Simulation how Dijkstra algorithm work [3].	Dijkstra algorithm
E.dere1 and A.Durdu	Usage of the A*algorithm to find the shortest path in Transportation System (2018)	Find the shortest path between a starting-point and ending-point on the map which is taken from Google Maps and segmented as grid-cells [11].	A*algorithm
Kiki Setiawan1, Supriyandi2, imam santoso3, Roy Buana4	MENGHITUNG RUTE TERPENDEK MENGGUNAKAN ALGORITMA A* DENGAN FUNGSI EUCLIDEAN DISTANCE (2018)	Menggunakan perhitungan algoritma A star (A*) dan metode Heuristik. menggunakan aplikasi Google Maps dalam penentuan jalur yang akan dipilih [12].	A* algorithm
Prof. Shiv Kumar Goel, Mr. Sufiyan Ansari, Ms. Tejashree Kuwalekar	Using A* algorithm to find shortest path in Indoor positioning system (2017)	Research the general working of A* algorithm and second section shows that how A* is better than Dijkstra's algorithm [9].	A* algorithm

CHAPTER 3

METHODOLOGY

3.1 Project methodology

The research methodology is essential to ensure the project achieves the objective. This chapter will explain the specific details of the methodology being used in order to develop this project. Methodology plays an important role as a guide for the project to complete and working well as a plan. In order to ensure the effectiveness of the Route for Online delivery services (ROde) application, in this project, we're using the Mobile Application Development Life Cycle (MADLC) model.

3.2 Mobile Development life cycle

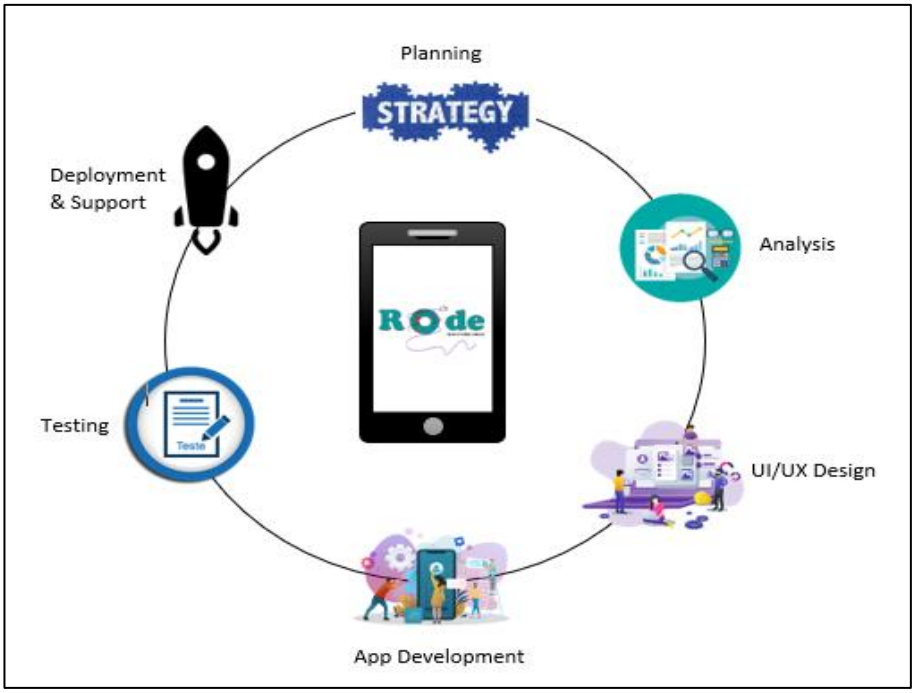


Figure 3.2.1: Mobile Development Life cycle

1. Planning

Planning is the first step in the mobile development life cycle. In this phase, it will go through an initial planning stage to map out the specification documents, hardware, or software requirements that will be used in this project. In this project, the android studio will be development tools, while Google Map Android APIs JavaScript V2 will be implement in the ROde apps. For hardware laptop and smartphone are definitely being used for developing and testing the project. Generally, prepare for the upcoming stages of the life cycle.

Analysis

The analysis phase is encompassing the requirement to perform the tasks. In this phase, the limitation of this project is identified. This project is based android operating system, that means only android user can use it. Besides, if GPS and internet is offline, this application is useless because the apps cannot find the location. In addition, this apps will be only covered in Pulau Pinang. Furthermore, objective, details of scope, and function system development project are also analysed in this phase. The concept of A* algorithm which is finding the shortest route will be implement in this project.

2. UI/UX design

first of all, the developing the user experience (UX), a common practice is wireframing. Wireframing is a rough draft of what users see. It helps in giving a rough idea regarding the looks and functionality of the apps based on the prescribed operating system The user interface design will be sketch in this phase using Microsoft power point. Next, designing user interface should be focusing on two aspects: functionality and the user interface (UI). User friendly, attractive, and informative are the characteristics of user interface as it covers what users see in the app and what they interact with, the functionality aspect focuses on how the app works and what users can do with it. It can be the help screen, keyboard, or buttons. The graphics and colours are finalized in this phase.

3. App development

The stage of development consists of two distinct parts: the first, prototyping, is intended to be quick, while the second, building, is the actual production of the end product. All the complexities regarding building an app are included in this section. For Route for Online delivery, the application is coded using Android Studio with JDK. The code is developed first for the core functionalities. Next, the A*algorithm will be implement in this phase by using JavaScript programming language to solve the problem of finding the shortest route. In most cases, smartphone applications running on Android platforms are designed using Java. Any of the features that make it the most common language for developing applications are easy to understand and enormous community support.

4. Testing

Testing is a method of discovering software bugs or bugs in the program code, while debugging refers to eliminating bugs or bugs from the source code. We have to make sure that everything in the app works. Mobile app testing goes through every single action in the apps from the most used to the least to verify that nothing “breaks”. It should be aiming to minimize the number of errors or breaks as much as possible before the app launch.

5. Deployment and support

Following the conceptualization, prototyping, design, and testing of a mobile app. Now, the program will be released to end-users to assess if the application is working well. The smartphone downloads the program. After checking all the features of the application and running well, the application is ready for an official launch.

3.3 System design

System design is the process of identifying system elements such as modules, architecture, components, and their interface and data for a system based on particular requirements. In this project, the design will be presented by the framework and flowchart.

Framework design

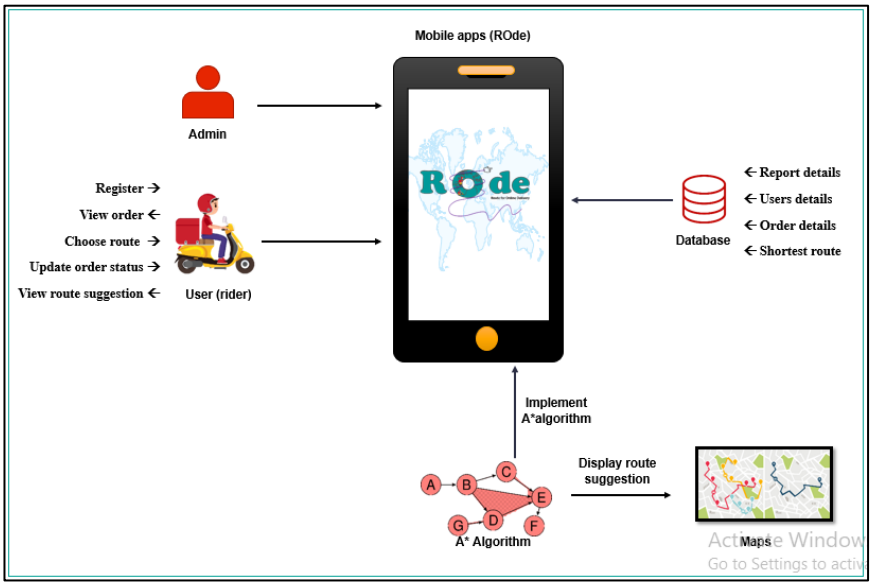
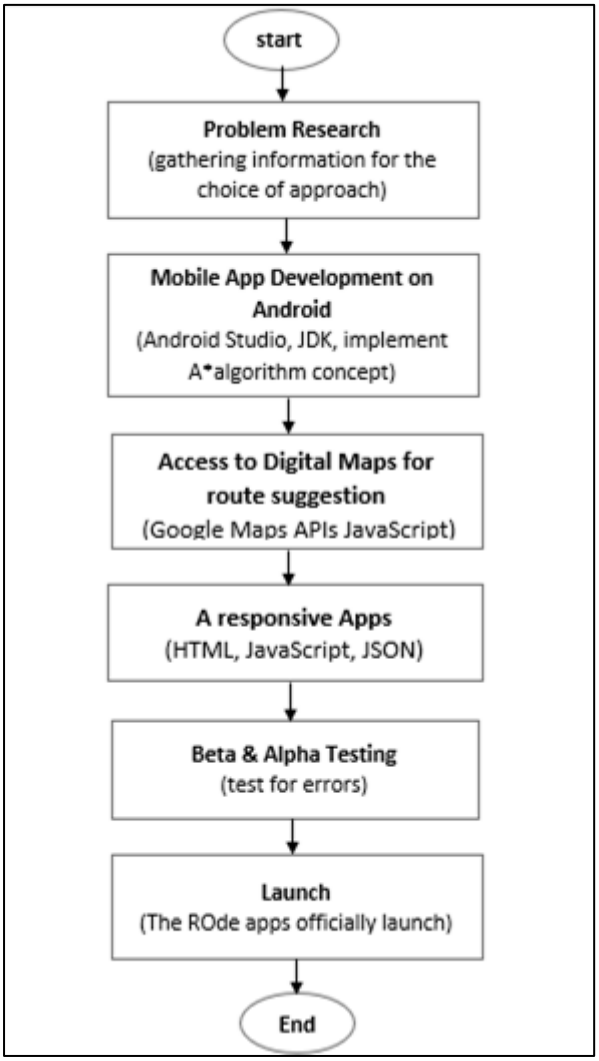


Figure 3.3.1: Framework Route for Online delivery (ROde)

Route Online for delivery (Rode) application is for a rider as a user of this system. Firstly, a rider will be provided the customer's address after they accept the customer order. The application will display the suggestion of the routes Admin will be able to log in, update location details, user details and report details and the application will display the report details.

3.3.1 Flowchart



3.3.2: Flowchart Route for Online delivery (ROde)

The Route for Online delivery (ROde) application were develop to find the shortest route in online food delivery services. Programming, testing and deployment tasks were accomplished to build the mobile app operating system by using Android Studio and JDK (Java SE Development Kit). The concept of A*algorithm will be implement in the phase of Mobile apps development on Android. Google Maps APIs JavaScript were called for the access to Digital Maps using Java programming language to display the route suggestion to the user. The ROde app will be used JavaScript Object Notation (JSON) to give full instruction

on how to navigate and use the ROde application. At the phase of Beta and Alpha testing, the apps will be test for error before upload for user to use. After testing, the ROde apps will be officially launch.

3.4 System requirement

System requirement is the required specifications a device must have to use certain hardware or software to ensure that all system development work smoothly without any interruption and problem. Several requirements were used in this project, include:

3.4.1 Software requirement

Table 3.4.1: Software requirement

No	Software name	Description
1.	Android studio	Tools for building apps
2.	Google Map Android APIs JavaScript V2	Display maps on mobile devices
3.	Firebase	Database
4.	Microsoft Word 2016, Microsoft PowerPoint 2016	Presentation and word processing

3.4.2 Hardware requirement

Table 3.4.2: Hardware requirement

No	Hardware name	Description
1.	Laptop	- Intel Inside core i3 - Memory 8 GB - OS: windows 10
2.	Smartphone	Android version 10
3.	Hard disk	To back up the data of the project

3.5 Proof of concept

3.5.1 User Interface

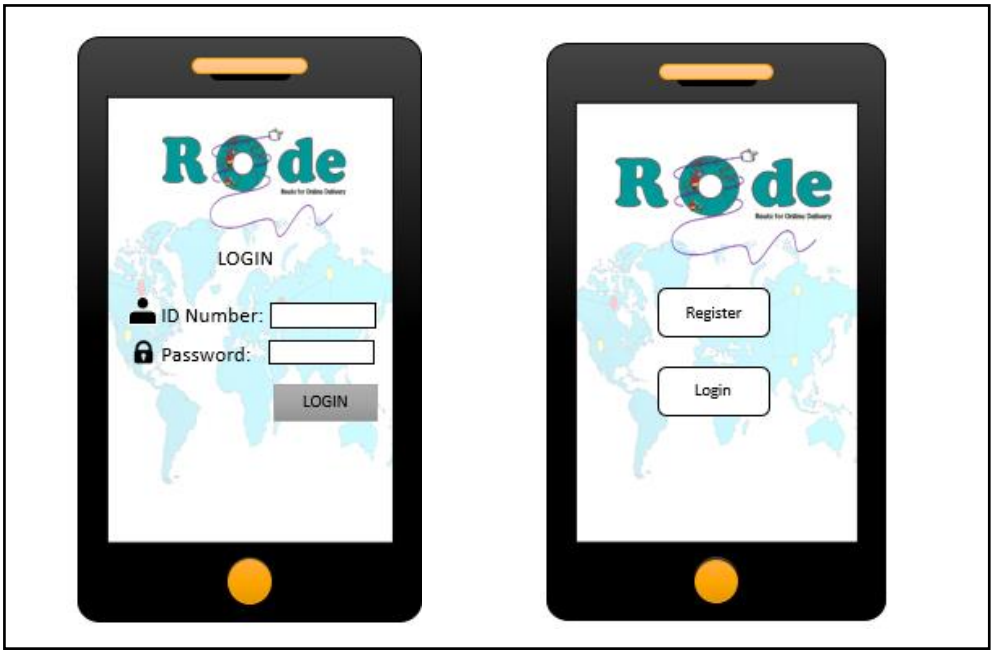


Figure 3.5.1: Interface for register and login

User have to register to the application. ID number will be generate after registration.

User need to Log in to the application by using their ID number and password

Figure 3.5.2: Interface for homepage 1

Homepage : Discovery page where user can receive current issues.

Homepage : User can view and receive order from the button “YOUR TASKS”.

Figure 3.5.3: Interface homepage 2

Homepage : User can view customer and order details and accept order.

Figure 3.5.4: Interface for maps

Maps : The route suggestion will be display to the user by click on the button "ACCEPT"

Maps : Maps will be view the path of route that have been choose

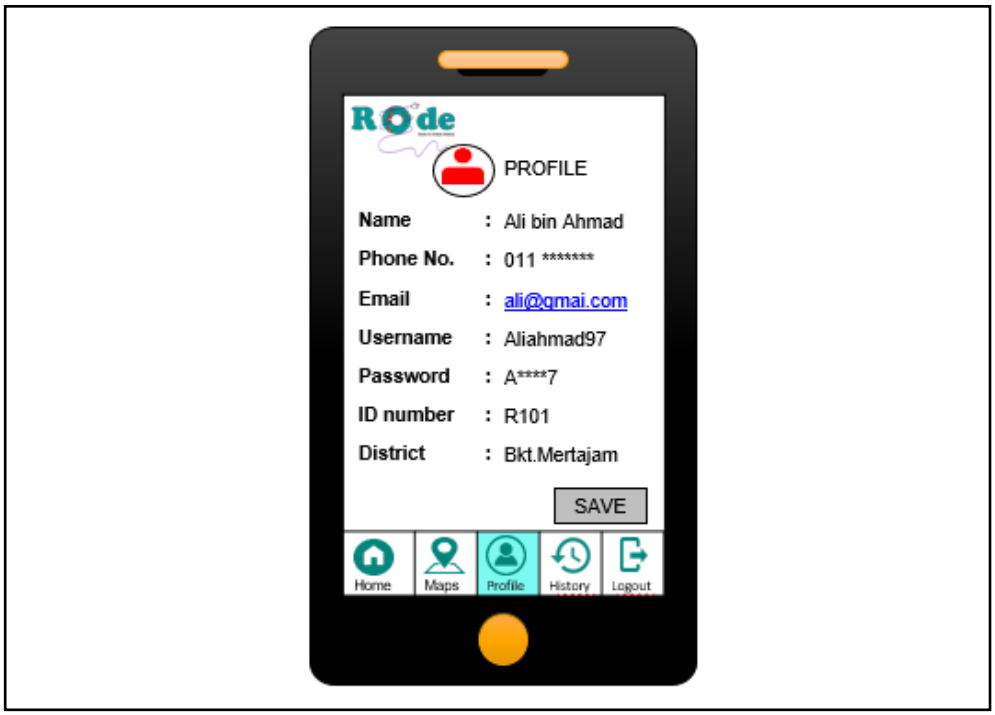


Figure 3.5.5: Interface profile

Profile : User can view and edit their profile.

Figure 3.5.6: Interface for history

History : Record all order history, include date/day and UserID

Figure 3.5.7: Interface for logout

Logout : user can logout from the apps.

3.5.2 Admin interface

Figure 3.5.8: Interface for admin login

Admin have to login by using their username and password.

Admin can manage user details, customer details and report details.

Figure 3.5.9: Interface for user and customer details

User details: admin can view, update, delete and retrieve user details.

Customer details : admin can view, update, delete and retrieve customers details

Figure 3.5.10: Interface for report details

Admin: make a final report for user's Key performance indicator (KPI)

3.6 Method/ Technique

For Route for Online delivery (ROde), A* algorithm will be used to achieve the objective of the project. A* algorithm is jointly proposed by P.E. Hart, N.J. Nilson and B. Raphael [7]. A*algorithm is a compact and efficient algorithm. A*algorithm is a typical artificial intelligence algorithm of heuristic search. Compared to other artificial algorithms [5] [15], it has many advantages, such as the shortest running time, high efficiency, easy implementation. Therefore A*algorithm has been widely used in various fields [16]. Currently, the way of improving the A* algorithm uses two methods, lowering the algorithm running time and reducing the storage space. Usually, by improving the traversal way, the storage space is reduced [7].

In Route for Online delivery (ROde) application should be assisted the users to get the shortest path from an initial location to the destination. Here, we are going to find out how the A* algorithm can be used to find the most cost-effective path in a graph.

1. **Collect data**

   The map taken from Google Maps, node A is the start node and node G is a destination.

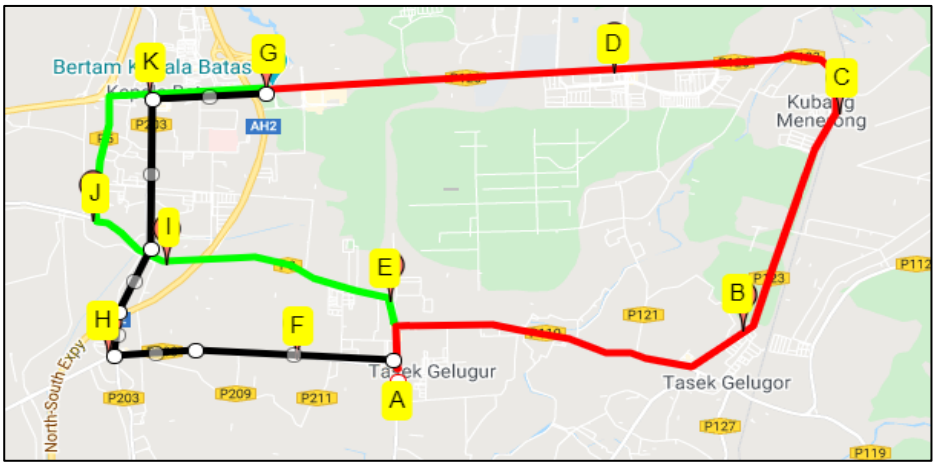


Figure 3.6.1: Route on maps

Assume every place and intersection as a node, and the path is the path to the node. Next, consider the path length of each node as weight.

Selected paths, obtained from google maps

Table 3.6.1: Selected path

Node	Distance (km)
A → B	0.85
B → C	3.6
C → D	3.7
D → G	4.5
A → E	4.6
E → I	2.8
I → J	2
J → G	4.9
A --> F	5.6
F → H	2.7
H → K	6.1
K → G	2.4

Represent the path in the form of a grid to get the coordinate value for each node.

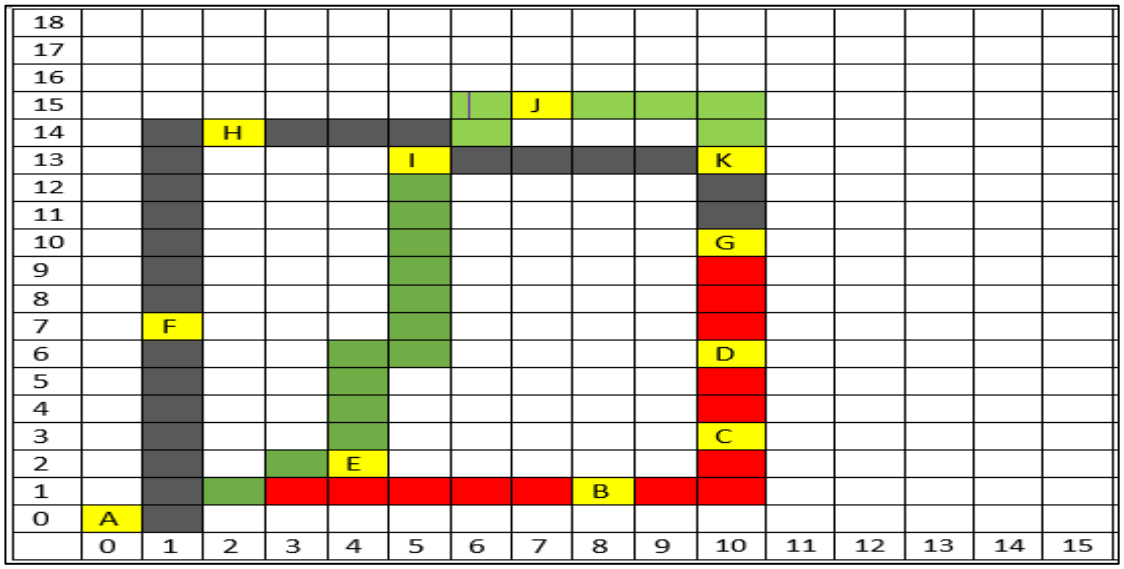


Figure 3.6.2: Route on Grid

Table 3.6.2: Coordinate

Node	Name	Coordinate
A	Kiosk Bertam	(0,0)
B	Bandar Putra Bertam	(8,1)
C	Kubang Semang	(10,3)
D	Tmn. Sepadu	(10,6)
E	Tmn Alamanda	(4,2)
F	Kepala Batas	(1,7)
G	Tasek Gelugor	(10,10)
H	Tmn. Haji Ahmad Badawi	(2,14)
I	Simpang Empat Pmtg.Buloh	(5,13)
J	Pokok Sena	(7,15)
K	Giat Mara Prima TSK	(10,13)

2. Calculate heuristic value

Next, calculate the distance between two node using Euclidean distance formula

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

By using the formula above, the calculation of all points can be seen as follows:

Route 1 : Red

Node	Euclidean distance	Total
A → B	$\sqrt{(0-8)^2 + (0-1)^2}$	8
B → C	$\sqrt{(8-10)^2 + (1-3)^2}$	2.8
C → D	$\sqrt{(10-10)^2 + (3-6)^2}$	2
D → G	$\sqrt{(10-10)^2 + (6-10)^2}$	4

Route 2 : Green

Node	Euclidean distance	Total
A → E	$\sqrt{(0-4)^2 + {}^2(0-2)}$	4.4
E → I	$\sqrt{(4-5)^2 + (2-13)^2}$	11
I → J	$\sqrt{(5-7)^2 + (13-15)^2}$	2.8
J → G	$\sqrt{(7-10)^2 + (15-10)^2}$	5.8

Route 3 : Black

Node	Euclidean distance	Total
A → F	$\sqrt{(0-1)^2 + (0-7)^2}$	7
F → H	$\sqrt{(1-2)^2 + (7-14)^2}$	7
H → K	$\sqrt{(2-10)^2 + (14-13)^2}$	8
K → G	$\sqrt{(10-10)^2 + (13-10)^2}$	3

Figure 3.6.3: Calculation Heuristic

3. Implement A*algorithm

After the heuristic value of each node is obtained, the next step is to find f (n)

using the A * algorithm with the formula:

$$f(n) = g(n) + h(n)$$

Route 1 : Red

Node	A*algorithm	Total
A → B	0.85 + 8	8.85
B → C	3.6 + 2.8	6.4
C → D	3.7 + 2	5.7
D → G	4.5 + 4	8.5
Total		29.45

Route 2 : Green

Node	A*algorithm	Total
A → E	4.6 + 4.4	9
E → I	2.8 + 11	13.8
I → J	2 + 2.8	4.8
J → G	4.9 + 5.8	10.7
Total		38.3

Route 3 : Black

Node	A*algorithm	Total
A → F	5.6 + 7	12.6
F → H	2.7 +7	9.7
H → K	6.1 + 8	14.1
K → G	2.4 + 3	5.4
Total		41.8

Figure 3.6.4: Calculation A* Algorithm

4. Comparison

After calculating the heuristics and performing the search steps using the A * algorithm, then f (n) total obtained from:

route 1 is 29.45 km, from node A → B → C → D → G

route 2 is 38.3 km, from node A → E → I → J → G

and, route 3 is 41.8 km, from node A → F → H → K → G

based on the result we can conclude that the shortest route is from node A → B → C → D → G.

CHAPTER 4

CONCLUSION

In conclusion, the Shortest Route in Online Delivery Services (ROde) applications are built to assist users as these apps can provide the user with recommendations of the shortest route so that a user does not need to waste time and go to other apps to look for information because the application functions as a one-stop center that provides all the necessary information for delivery services. The ROde application is not optimized enough because it cannot handle all variables, such as the use of this app, for android users only. Besides, to search for a position on maps, an internet connection is needed. Other than that, only in Pulau Pinang are these apps covered. Recommendation for future project continuation the application should offer users different preferences for route search, as the user not only can choose a path based on the shortest distance path but can also choose a path based on the shortest time path. Next, update more features on this application, such as this application will be able to notify the customer or these apps can link to any store to make it easier for riders. Hopefully, for future projects, the ROde apps can be designed successfully and produce the expected result as they can be used to solve the problem of finding the shortest route in online delivery services.

REFERENCES

1- Kikelomo, A. A., Asafe, Y. N., Paul, A., & Olawale, L. N. (2017). Design and implementation of mobile map application for finding shortest direction between two pair locations using shortest path algorithm: a case study. International Journal of Advanced Networking and Applications, 9(1), 3300

2- Oduwole, O. E., Asaolu, O. S., & Osigbemeh, M. S. (2019). Shortest Route: A Mobile Application for Route Optimization using Digital Map. Arid Zone Journal of Engineering, Technology and Environment, 15(4), 1038-1048.

3- Ginting, H. N., Osmond, A. B., & Aditsania, A. (2019, May). Item Delivery Simulation Using Dijkstra Algorithm for Solving Traveling Salesman Problem. In Journal of Physics: Conference Series (Vol. 1201, No. 1, p. 012068). IOP Publishing.

4- Isa, K., 1, Shah, M., 2, Palpanadan, S. T., 3, & Isa, F., 4. (march - april 2020). Malaysians' Popular Online Shopping Websites during Movement Control Order (MCO). International Journal of Advanced Trends in Computer Science and Engineering, Volume 9(No.2), issn 2278-3091, 2154-2158. doi:DOI: 10.30534/ijatcse/2020/19092202

5- Mehta, P., Shah, H., Shukla, S., & Verma, S. A Review on Algorithms for Pathfinding in Computer Games.

6- Chai, L. T., & Yat, D. N. C. Online Food Delivery Services: Making Food Delivery the New Normal.

7- Peng, J., Huang, Y., & Luo, G. (2015). Robot path planning based on improved A* algorithm. Cybernetics and Information Technologies, 15(2), 171-180.

8- Kapi, A. Y., Sunar, M. S., & Zamri, M. N. (2020). A Review on Informed Search Algorithms for Video Games Pathfinding. International Journal, 9(3).

9- Goel, S. K., Ansari, M. S., & Kuwalekar, M. T. (2017). Using A* algorithm to find shortest path in Indoor positioning system.

10- Juniawan, F. P., Sylfania, D. Y., & Perkasa, E. B. Pemanfaatan Algoritme A* Menggunakan AIspace Dalam Pencarian Rute Terpendek.

11- Dere, Esat & Durdu, Akif. (2018). Usage of the A* Algorithm to Find the Shortest Path in Transportation Systems.

12- Setiawan, K., 1, S., 2, Santoso, I., 3, & Buana, R., 4. (2018). MENGHITUNG RUTE TERPENDEK MENGGUNAKAN ALGORITMA A* DENGAN FUNGSI EUCLIDEAN DISTANCE. Seminar Nasional Teknologi Informasi Dan Komunikasi 2018 (SENTIKA 2018), 70-79.

13- Google, J. (2013, May 16). *Android Studio* . Retrieved from Android Studio: https://developer.android.com/studio

14- *Maps JavaScript API*. (2021, 1 25). Retrieved from Google Maps Platform: https://developers.google.com/maps/documentation

15- Charoenporn, P. Comparison of Algorithms for Searching Shortest Path and Implementations for the Searching Routing System via Web Services.

16- N. H. Barnouti, S. S. M. Al-Dabbagh, and M. A. Sahib Naser. Pathfinding in Strategy Games and Maze Solving Using A* Search Algorithm, J. Comput. Commun., Vol. 04, No. 11, pp. 15–25, 2016.

17- Sharma, S. K., & Pal, B. L. (2015). Shortest path searching for road network using a* algorithm. International Journal of Computer Science and Mobile Computing, 4(7), 513-522.

18- Mathew, G. E. (2015). Direction based heuristic for pathfinding in video games. *Procedia Computer Science*, *47*, 262-271.

19- Esmaeel, H. R. (2015). Apply android studio (SDK) tools. *International Journal of Advanced Research in Computer Science and Software Engineering*, *5*(5).