



AITCS

Homepage: <http://publisher.uthm.edu.my/periodicals/index.php/aitcs>
e-ISSN :2773-5141

SERVICE FINDER: An Android-based Service Finding and Booking Application

Kelly Foo Yen Li¹, Norfaradilla Wahid^{1*}

¹Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: <https://doi.org/10.30880/aitcs.2023.04.01.027>

Received 16 June 2022; Accepted 26 September 2022; Available online 30 June 2023

Abstract: Service Finder is an online service finding and booking Android application developed to ease the public to find the local services such as plumbing and electrical efficiently. Service Finder allows users to search for the service nearby by using the location-based service approach. Location-based services use the global satellite positioning to access the location of the user to provide specific information to the user. Service Finder is developed based on system prototyping methodology by using Android Studio and MySQL database. Android Studio is software used to develop Android applications while MySQL database is a database used to store data of the application. Service Finder includes five modules which are management on profile, account, service, appointment and rate and review. Service Finder allows users to find service needed according to their location and make appointments with preferred time. Overall testing on the developed application is passed except for rate modules.

Keywords: Service Searching, Online Booking, Local Service, Android

1. Introduction

Service is importance to solve problem met in daily life but there is lack of choice for public to get service provider in a fast and convenient way. Most of the people might use to a fixed service provider to solve their problems but when the service provider is not available, they are facing problem of finding other service provider. People used to call service provider to discuss on the problem but there met a problem to describe their location to service provider. Therefore, Service Finder is introduced.

Service Finder is an application that aims to provide quick searching and contact to the local service provider for requesting service and solve the problem met in daily life. This application is also a platform that enables the service provider to promote their service through the internet. It includes five categories of service which is the plumbing, electrical repairing, car repairing, trailer and cleaning service.

The objectives of this project development are to design a platform to find local services like plumber, electrician, car repairing, trailer service and cleaning service, to develop an Android application to help users find a service more efficiently and to evaluate the developed application on target users based on its functionality and user acceptance.

For the following section, Section 2 will discuss the related research made for the application. Section 3 will discuss the methodology used to develop the application and the system analysis and design for the application. Section 4 is about the result and discussion while Session 5 will conclude the project.

2. Literature Review

2.1 Domain Study Background

This section is about the study of the current system. Figure 1 shows the generic business flow when the customer needs the service to solve some daily problems. Customers need to get the contact of service providers from the sources they can reach such as their friends and relatives or search online. After they get the contact, they need to call or contact through WhatsApp to reach the service provider.

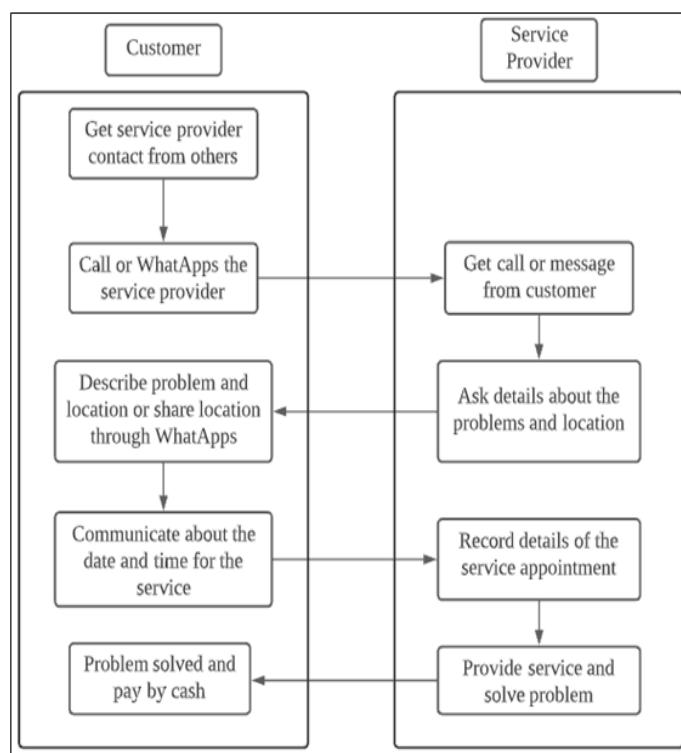


Figure 1: Generic business process diagram for a service request

2.2 Technology Used: Android Studio and Location- Based Service (LBS)

Android Studio is a software that is used to develop mobile applications. To access all of Android's features, one needs to develop applications using tools such as Android Studio, which is provided by Google as a full-fledged development environment for Android development, debugging, testing, and packaging [3]. Android Studio provides a virtual emulator that enables system testing on-screen.

LBS is based on location to provide various services of location related and value-added applications by utilizing positioning technology to determine the location of mobile communication equipment [1]. Location-Based Service is used to filter the information according to location and enable an application to provide more specific information to users. Global satellite positioning system (GPS)

is needed to specify the user location for location-based service to be function. Location-Based Service enables the application to provide more specific suggestions for the services.

2.3 A Study on Similar System

Three existing similar systems are studied to learn about the strength, weakness and features of existing systems. The existing systems studied are ServisHero: On-Demand Service [7], Grab [6] and Foodpanda [5].

2.3.1 ServisHero: On-Demand Service

ServisHero is a mobile application that enables users to book for the service they need on preferred data and time. There are 3 types of booking methods provided in this application which are the book once-off, service plan and quotation. The application provides different categories of which are the cleaning, plumbing, air conditioning, pro home disinfection, handyman, electrical and wiring, moving, contractors and renovation, pest control, home inspection and office disinfection. There is selection on the different price and level of service without selection on the service provider. The services provided are available for three areas in Malaysia which are the Klang Valley, Johor Bahru, and Penang.

2.3.2 Grab

Grab is an application that provides services on transportation, delivery, and electronic payment. When a user requests for a transport service, the request will be sent to a driver nearby and the user will receive the information of the driver that accepts the order. Users can trace the real-time location of the driver. After the order is completed, users can rate the driver and comment if any inconvenience caused or any improvement. Users can choose a restaurant and order food on the application while the delivery man is assigned by the system to the delivery man nearby and willing to accept the order. Users can rate on the service and the restaurant after order is complete.

2.3.3 Foodpanda

Foodpanda is an application that provides service on local food and grocery delivery. The application allows users to order for delivery, self-pickup and dine-in service. The application starts with location accessing to get the current location of the user or location entered by the user followed by the suggestion of a restaurant nearby. Users can view the information, menu, rating, distance from current location and time estimation for the service. After the user places an order, the user can trace the progress of the order and location of the delivery man after the delivery man has picked up the order.

2.4 System Comparison

Service Finder consists of five modules which are profile management, account management, service management, appointment management and rate and review management. The profile management enables users to manage their profile information and apply to become a service provider. The account management enables administrators to receive applications and make decisions to accept a user to become a service provider. The service management allows the service provider to manage the service provided. The appointment management allows the user to manage the request and appointment while the service provider can decide to receive or reject the request. The rate and review management allows the user to review and rate the service.

Table 1 and Table 2 shows the comparison of the existing systems and the proposed system. The tables compare the features of ServisHero, Grab, Foodpanda and Service Finder based on the interface, payment method, approach, modules and service provided. The interface features include the search bar, navigation bar, menu bar, background photo in the system. Payment features include electronic wallet, FPX, Credit card, cash and refund while the approach is the location-based service. The modules are about the modules for each system and service is about the service provided by each system. In the

service features discuss the function of allowing users to apply to become a service provider and service provided in the system. The types of services include delivery, transport, air conditioner service, disinfection, handyman, moving, pest control, plumbing, electrical repairing, car repairing, trailer and cleaning service. All systems allow users to apply to become a service provider except ServisHero.

Table 1: Comparison of existing system and proposed system

Category	Feature	ServisHero	Grab	Foodpanda	Service Finder
Interface	Search Bar	X	/	/	/
	Navigation Bar	/	/	/	/
	Menu Bar	X	X	/	X
	Background Photo	/	X	X	X
Payment	Electronic Wallet	X	/	X	X
	FPX	X	/	/	X
	Credit card	/	/	/	X
	Cash	/	/	/	/
	Refund	/	/	/	X
Approach	Location Based Service	/	/	/	/
Modules	Order History	/	/	/	/
	Payment history	X	/	X	X
	In-App Chat	X	/	X	X
	Rate and review	X	/	/	/
	Profile management	/	/	/	/
Service	Apply as service provider	X	/	/	/
	Delivery	X	/	/	X
	Transport	X	/	X	X
	Air conditional service	/	X	X	X
	Disinfection	/	X	X	X
	Handyman	/	X	X	X
	Moving	/	X	X	X
	Pest Control	/	X	X	X
	Plumbing	/	X	X	/

Electrical repairing	/	X	X	/
----------------------	---	---	---	---

Table 2: (cont.)

Category	Feature	ServisHero	Grab	Foodpanda	Service Finder
Service	Car repairing	/	X	X	/
	Trailer	X	X	X	/
	Cleaning	/	X	X	/

3. Methodology

This section discusses the methodology used for Service Finder project development. The methodology selected to develop Service Finder is system prototyping to develop the system and improve the system based on user feedback on the system. This methodology ensures the result of the application is acceptable to the user and with the features and functions needed by the user.

3.1 System Prototyping

Figure 2 shows the flow of the system prototyping methodology to develop a system.[3]. The system will be built quickly with minimum requirement and tested for the function. The prototype will be reanalyzed, redesigned and reimplemented according to the feedback from users. The next prototype will be corrected and add more features. This cycle will be repeated until they produce a prototype that reaches the goal of the project. This section discusses the methodology used in Service Finder.

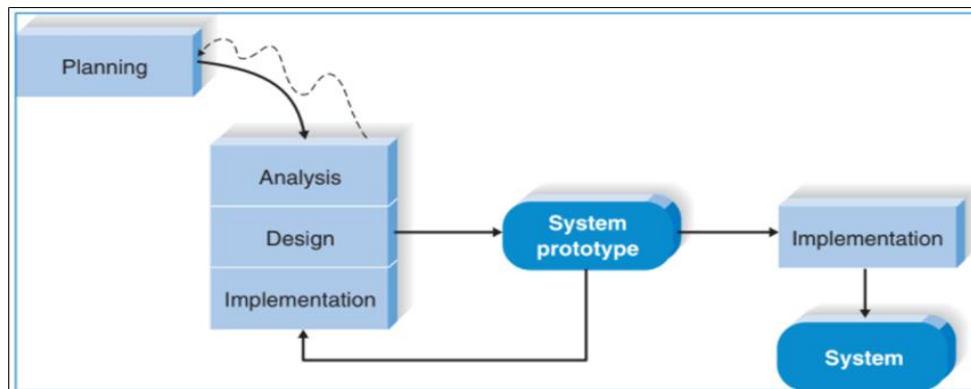


Figure 2: Flow of System Prototyping [2]

The planning phase planned the project about the simple idea and the business value of the Service Finder. The project plan is drawn by Gantt Chart (refer to Appendix) where the project is started from 6th October 2021 and planned to be completed by 30th June 2022. In this phase, the methodology used for the project also decided which is system prototyping methodology. The project continues with the analysis phase to define the problem, objective, scope, user, location of the project applied. The existing related systems also studied to understand the current system and existing problems in the systems. Analysis phase also analyzes the requirement of the application on the software and hardware. Besides, the Unified Modeling Language (UML) diagrams are drawn to analyze the flow of the system and the use case of different roles of the user in the system.

In the design phase, the design of the user interface and database of Service Finder are decided. The user interface design is drawn by using wireframe to show the arrangement of elements of the interface.

The database is described by using Entity-Relationship Diagram (ERD) and data dictionary. The implementation part is carried out using Android Studio and MySQL database. The workflow for each phase is shown in Table 3.

Table 3: Workflow for each phase

Phase	Activity	Output
Planning	<ul style="list-style-type: none"> • Study on project background • Define <ul style="list-style-type: none"> ○ Problem statement ○ Objective ○ Scope ○ Expected result of the project ○ Project significant • Decide development model: system prototyping 	<ul style="list-style-type: none"> • Project proposal • Gantt Chart
Analysis	<ul style="list-style-type: none"> • Study on existing system <ul style="list-style-type: none"> ○ ServisHero ○ Grab ○ Foodpanda • Questionnaire • Interview • System Analysis 	<ul style="list-style-type: none"> • Problem Statement • User requirement • Generic business process diagram • Comparison table between existing system and proposed system • Uses case diagram • Sequence diagram • Class diagram • Data flow diagram
Design	<ul style="list-style-type: none"> • Design <ul style="list-style-type: none"> ○ Flow for the using the application ○ Database ○ User interfaces 	<ul style="list-style-type: none"> • Entity relation diagram • User interface design wireframe
Implementation	<ul style="list-style-type: none"> • Coding on Android Studio • Connect with database 	<ul style="list-style-type: none"> • Profile management • Account management • Service management • Appointment management • Rate and review management

3.2 Hardware and software requirements

In every development, software and hardware are required for the process including Service Finder. The hardware requirements are stated in Table 4 and Table 5 while the software requirements are stated in Table 6. Hardware requirements is about the hardware needed in this project such as laptop used to develop the application and smartphone to do testing on the application developed while software requirement is about the software used to develop the software such as Android Studio, MySQL database and Microsoft Word.

Table 4: Hardware requirement of the proposed system

Hardware	Function
Laptop Huawei D14	Device to develop the proposed system
Processor	Intel(R) Core (TM) i5-10210U CPU @ 1.60GHz 2.11 GHz

Random Access Memory (RAM)

8.00GB

Table 5: (cont.)

Hardware	Function
Operating System	Windows 10
Android smartphone	To test the function of developed system

Table 6: Software requirement of the proposed system

Software	Function
Android Studio	Software to develop Android Application
MySQL database	Database to store data for the proposed system
Microsoft Word	Software to write report

3.3 System Analysis and Design

Service providers are analyzed from different aspects such as the system architecture, database design, interface design and the program design. Figure 3 shows the system architecture of Service Finder. Users interact with the user interface which is the Android application and access the register and login process to use the function of the application. The register process will store the data inserted by the user into the database while the login process will request data from the database to confirm the validation of data input by the user to decide whether the user can login access the data and other functions of the system.

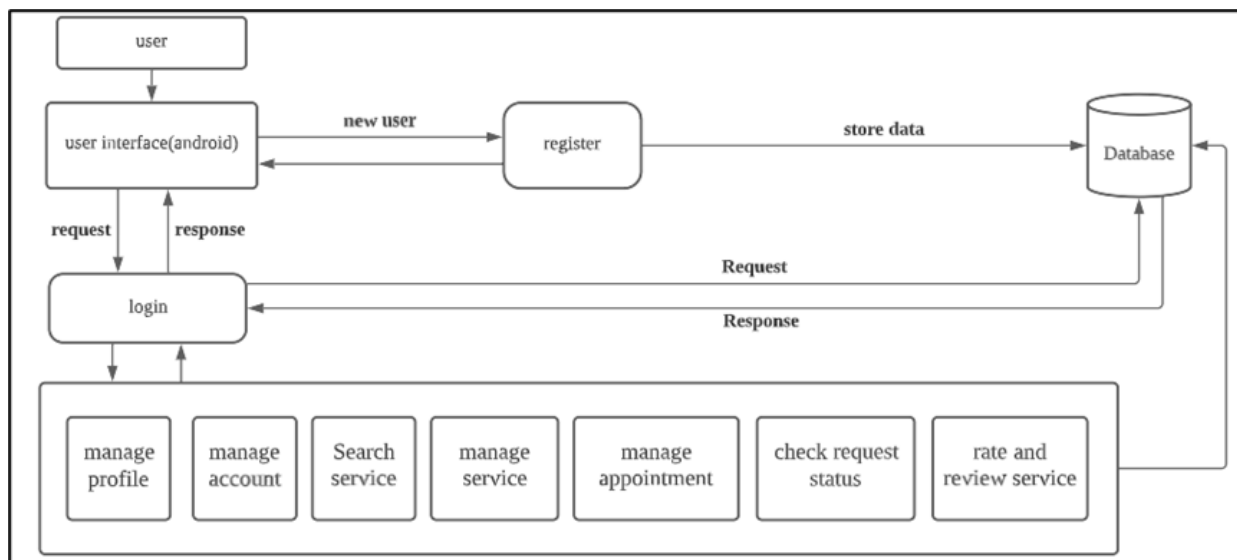


Figure 3: System Architecture diagram

3.3.1 System Analysis

The system is analyzed by using Unified Modeling Language specification (UML). There are many diagrams in UML specification to use in system development while the diagrams used in this project are the use case diagram, sequence diagram, activity diagram and class diagram. The uses case diagram is used to determine the uses case for every role of the user [4]. Figure 4 shows the use case diagram for Service Finder. There are three roles of user in the diagram which are user, administrator and service provider. The use cases of Service Finder include register, login, request service, manage appointment, rate service, manage profile, manage service, manage request, manage account and logout. The sequence diagram is used to determine the message flow and the function used for every use case while the activity diagram is used to demonstrate the workflow for every condition of the application from end-to-end.

Figure 5 shows the sequence diagram for service request of a user to search and request for a service. Users visit the search page and search for the service. The input is sent to the database to get related service results and display on the search page. Users can select a service to view details of the service and make appointments by input the details of the request. After confirmation, the request will be sent to the database. Class diagram is used to describe the classes, attributes, and functions and shows the relationships between classes in the application. Class diagram shows the class for every process and the attributes and functions of the class. Due to the limitation of the page, there are only the use case diagram and sequence diagram for the service request shown in this report.

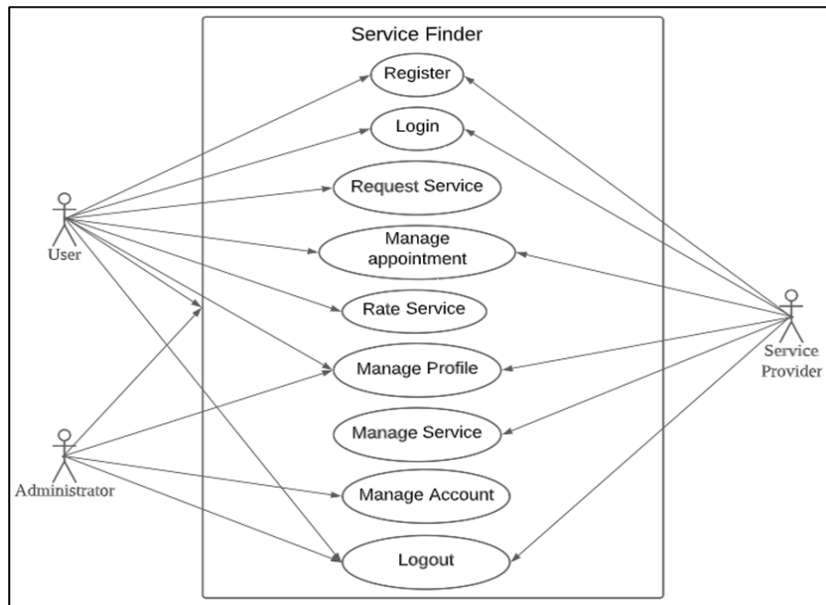


Figure 4: Uses Case Diagram

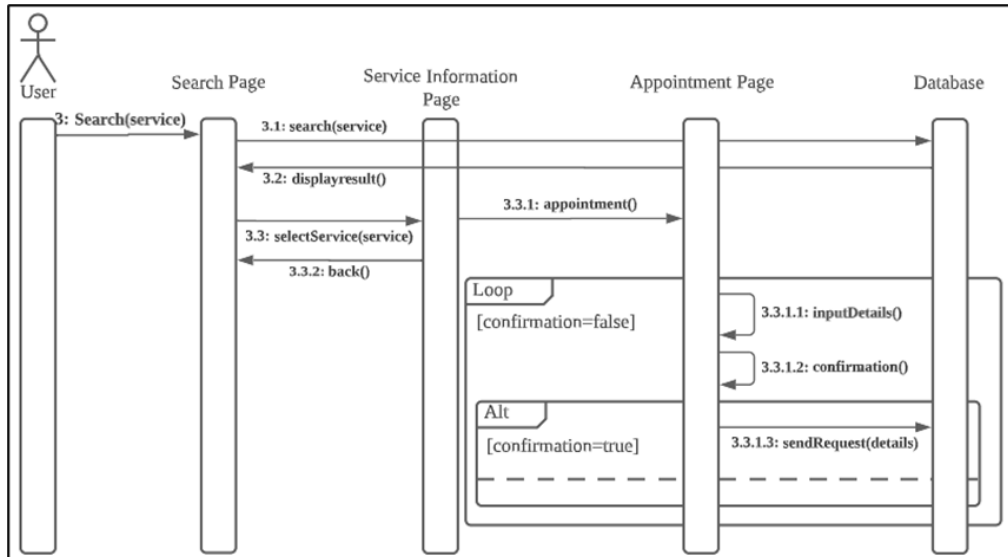


Figure 5: Sequence Diagram for Service Request

3.3.2 System Design

System Design is about the architecture of the system for different aspects such as database and user interface. The system design is important to be used in the implementation phase as to guide the development.

3.3.2.1 Entity-Relationship Diagram and Data Dictionary

The database of the system is designed by using Entity-Relationship Diagram (ERD) to determine the architecture of the database, components in each entity and the relationship between the entities. The data dictionary for each entity of the database is stated in the table while the table consists of the field name, data type, field length, constraint and description for every attribute of the entity. Figure 6 shows the ERD of Service Finder. Due to the limitation of the page, there are only data dictionary of appointment table shown in this report, i.e., in Table 7 and Table 8.

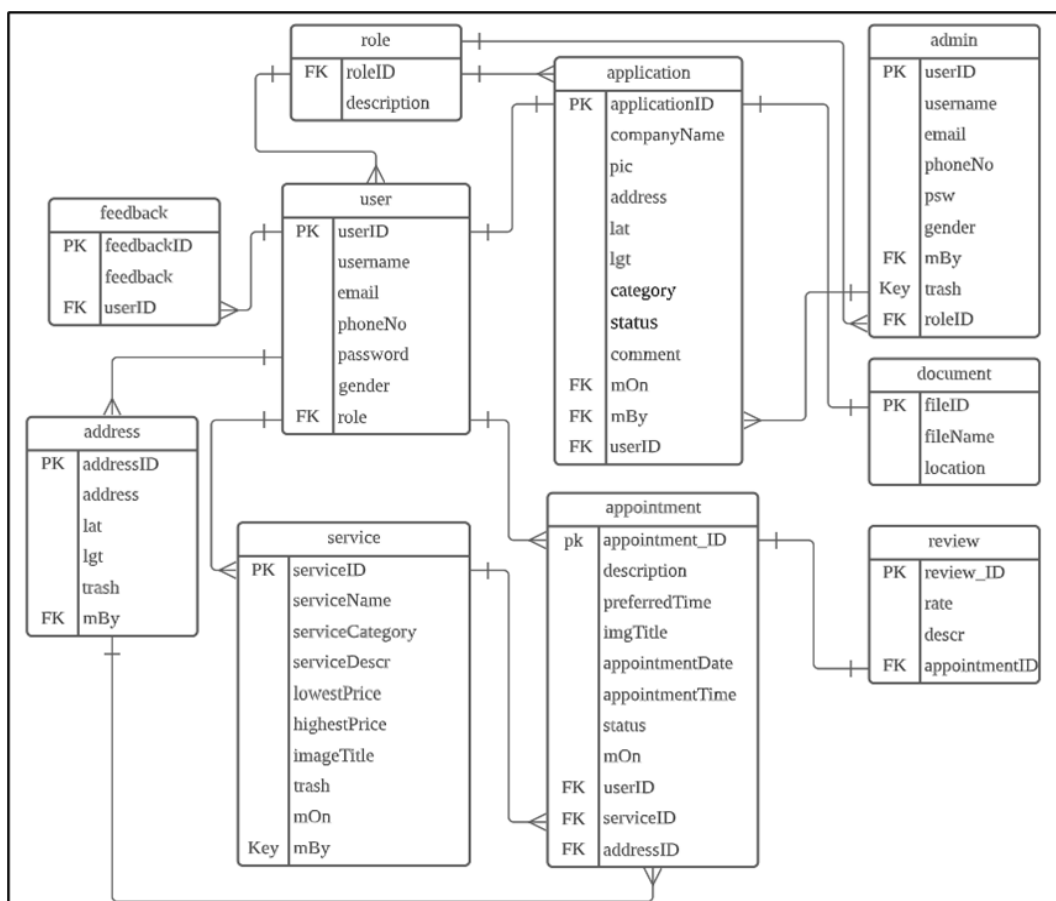


Figure 6: Entity-Relationship Diagram

Table 7: Data Dictionary of appointment table

Field Name	Data Type	Field length	Constraint	Description
appointmentID	Integer	5	Primary Key	Appointment id
description	text	100	Not null	Service location
preferredTime	Varchar	20	Not null	Preferred service time

Table 8: (cont.)

Field Name	Data Type	Field length	Constraint	Description
imgTitle	Varchar	10	Null	Image of appointment
appointmentDate	Varchar	10	Not null	Appointment date
appointmentTime	Varchar	4	Not null	Appointment time
status	Integer	1	Not null	Appointment Status
mOn	Datetime	-	Timestamp	Request time
userID	Integer	5	Foreign key	User id

serviceID	Integer	5	Foreign key	Service id
addressID	Integer	5	Foreign key	Address id

3.3.2.1 User Interface Design

The interface design of the Service Finder is drawn by wireframe. The element of the interface is clearly shown in the wireframe. Figure 7(a) shows the login page design of Service Finder while Figure 7(b) shows the register page design of Service Finder. Figure 7(c) shows the home page design of Service Finder. Figure 8(a) shows the search page design of Service Finder. Figure 8(b) shows the history page design of Service Finder. Figure 8(c) shows the profile page design of Service Finder.

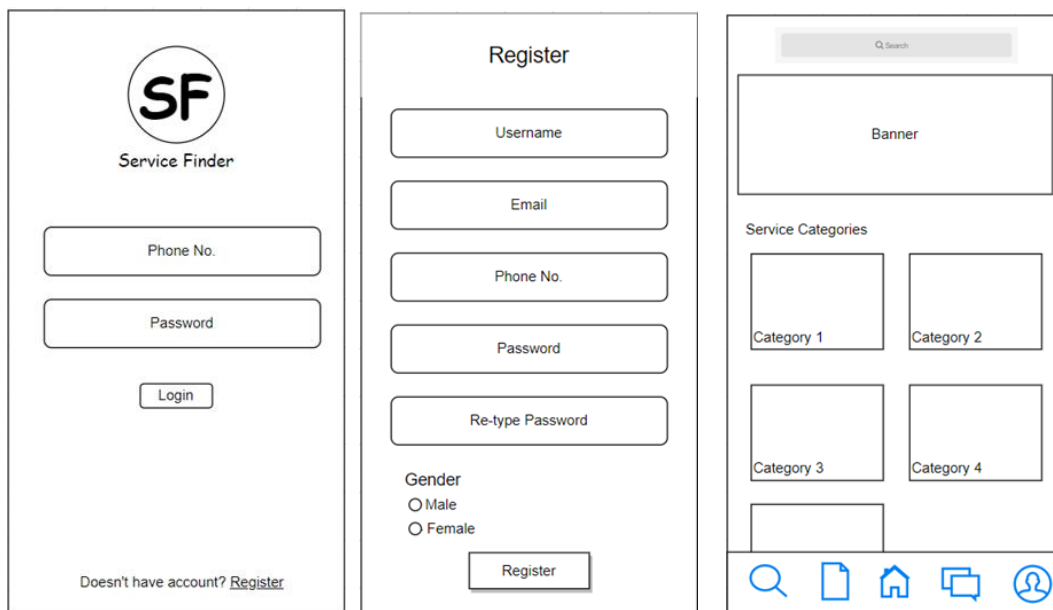


Figure 7(a): Login page design

Figure 7(b): Register page design

Figure 7(c): Home page design

4. Application Implementation and Testing

This section discusses on the implementation of the application developed and the testing phase after the application is implemented to make sure the application can be run as expectation and get notice if there are error in the application.

4.1 Application Implementation

4.1.1 Database Connection

The MySQL database is connected by the PHP file named conn with the code shown in Figure 8 that declared the database details. The file is called by another PHP file to connect to the database. The

MySingleton class in Figure 9 is declared to use Android Volley. Android Volley is used to request the data by HTTP request while the request include the URL to send HTTP request and related data is declared in the class and passed to MySingleton class to get or update the data in the database.

```
<?php
$db_name="id18197891_servicefinder";
$username="id18197891_servicefinder_db";
$password="YuVi1N<--s0QD)Ls";
$servername="localhost";
$conn = mysqli_connect($servername, $username, $password, $db_name);
mysqli_set_charset($conn , "utf-8");
?>
```

Figure 8: PHP code to connect MySQL database

```
private static MySingleton mInstance;
private RequestQueue mRequestQueue;
private Context mContext;

public MySingleton(Context mContext) {
    this.mContext = mContext;
    mRequestQueue = getmRequestQueue();
}

public RequestQueue getmRequestQueue(){
    if(mRequestQueue == null){
        Cache cache = new DiskBasedCache(mContext.getCacheDir(), maxCacheSizeInBytes: 1024*1024);
        Network network = new BasicNetwork(new HurlStack());
        mRequestQueue = new RequestQueue(cache, network);
        mRequestQueue = Volley.newRequestQueue(mContext.getApplicationContext());
    }
    return mRequestQueue;
}

public static synchronized MySingleton getInstance(Context context){

    if (mInstance == null){
        mInstance = new MySingleton(context);
    }
    return mInstance;
}

public<T> void addToRequestQueue(Request<T> request) { mRequestQueue.add(request); }
```

Figure 9: Java code of MySingleton class

4.1.2 Main Function Development Stage

The main function development stage consists of three parts which is the frontend development by Extensible Markup Language XML, backend development by Java and database development by PHP and MySQL database. After the application is developed the testing was carried out for user acceptance testing for target users. The user interface is designed and built with XML that enables the interface to be built easily, the java programming language is used to develop the activity of the application behind the XML code. The interface of main page is as shown in Figure 10(a) while the backend development shown in Figure 12 is used to start a new activity for the user to register when the user clicks on the

register button. A data validation occurred on the user input when the user clicks on the login button. The input for the required field is checked to make sure there is value from user input to continue with further code else an alert “All fields required” will be displayed to the user. User can click on the register button on main page to navigate to the Register page shown in Figure 10(b) to register a new account. If user is logged in, user will navigate to the address page shown in Figure 11(a) to select an address to continue while home page shown in Figure 10(c) will show up after user selected address. The new address page in Figure 11(b) will displayed when user click on the new button in address page to enable user to create new location with their current location or with the address they input. Figure 11(c) shows the user interface for user to input for the appointment details when they request for a service.

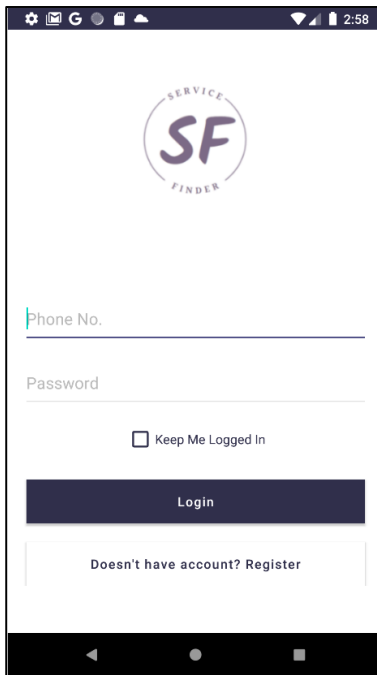


Figure 10(a): Main Page

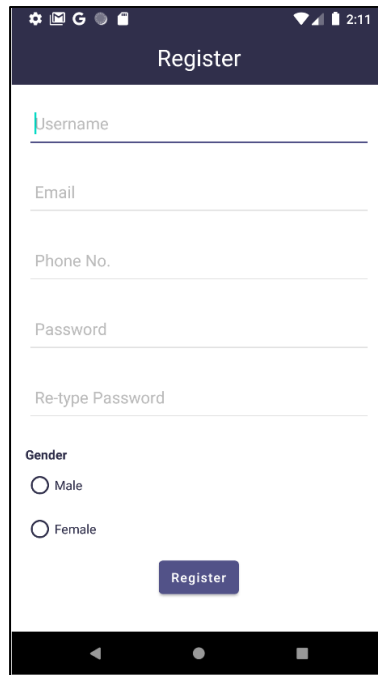


Figure 10(b): Register Page

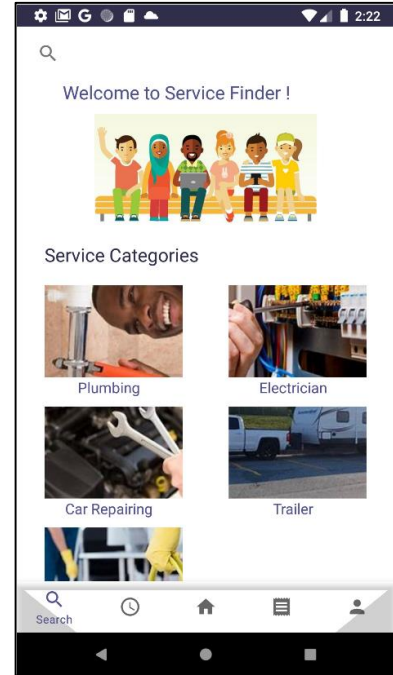


Figure 10(c): Home Page

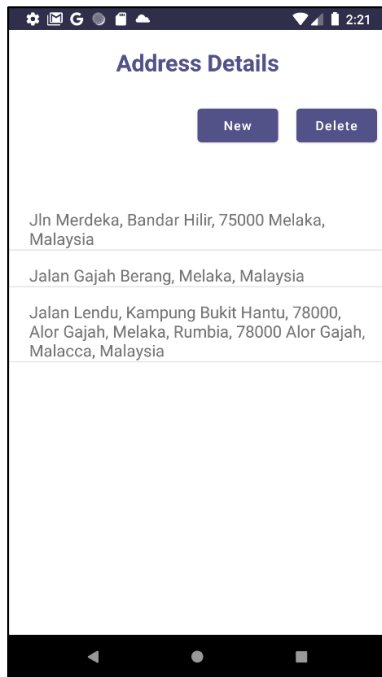


Figure 11(a): Address Page



Figure 11(b): New Address Page

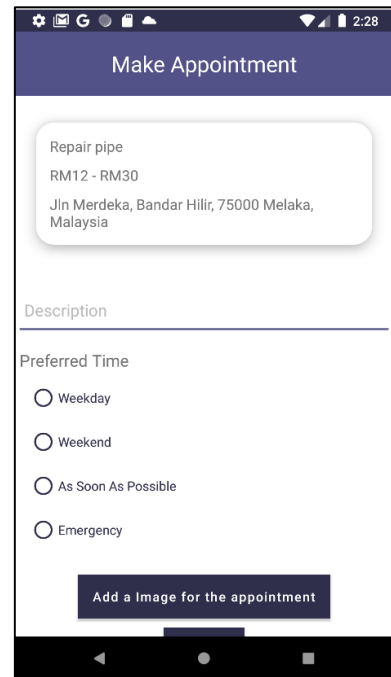


Figure 11(c): New Appointment Page

```
login = findViewById(R.id.btn_login);
login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String txtphone = phoneno.getText().toString();
        String txtpsw = password.getText().toString();
        if (TextUtils.isEmpty(txtphone) || TextUtils.isEmpty(txtpsw)) {
            Toast.makeText(context: MainActivity.this, text: "All fields required", Toast.LENGTH_SHORT).show();
        } else {
            logacc(txtphone, txtpsw);
        }
    }
});

register = findViewById(R.id.btn_register);
register.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent i = new Intent(getApplicationContext(), Register.class);
        startActivity(i);
    }
});
```

Figure 12: Backend development of main interface

4.1.3 GPS Activity

For user profile management, the user can create a new address in the address list with current location. The code in Figure 13 shows a partial code of GPS activity to get current location of the user and to be proceed with appointment and sent the location to service provider.

```

Location location = locationManager.getLastKnownLocation(GPS_PROVIDER);
if(location != null||!location.equals("")){
    lat = location.getLatitude();
    lgt = location.getLongitude();
    Geocoder geocoder = new Geocoder( context: newAddress
        .this, Locale.getDefault());
    try {
        List<Address> addressList = geocoder.getFromLocation(lat, lgt, maxResults: 1);
        address=addressList.get(0).getAddressLine( index: 0);
        LatLng latLng = new LatLng(lat, lgt);
        map.addMarker(new MarkerOptions().position(latLng));
        map.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, v: 15));
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Figure 13: Partial Code to get current location of user

4.1.4 Display Image Activity

In account management modules, the administrator will view the application details submitted by the user with the code in Figure 15 that is used to display the image of the business license of the application at the image view created by XML code in Figure 14. The XML code is called from the backend by id.

```

<ImageView
    android:id="@+id/viewImg"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_centerInParent="true" />

```

Figure 14: XML code to create image view on user interface

```

imgView = findViewById(R.id.viewImg);

Intent intent = getIntent();
String filename = intent.getStringExtra( name: "filename");
String url = "https://sfservicefinder.000webhostapp.com/serviceFinder/document/"+filename+".jpeg";
Glide.with( activity: this).load(url).into(imgView);

```

Figure 15: Backend development to display image

4.2 Testing

This section discussed the functional and user acceptance testing on the developed application and the result of testing.

4.2.1 Functional testing

The functional testing is testing on the application from start to end to make sure the application is fully implemented and ready to run. The testing function is important to make sure the application runs well for every module and fits the expecting outcome. The functional testing was constructed after the

application was implemented. Overall, the test result fulfills the expected result except for rating as the rate button does not appear. Therefore, the rate service cannot be complete.

Table 6: Test plan of the system module and results

Module	Functions	Test	Expected Result	Actual Result
Profile Management	Register	Incomplete data input	Alert message displayed to requested user fill in all fields	pass
		Invalid data input	Alert message about invalid input	pass
		Unique phone number, and email	Alert message if data is used in database	pass
	Login	Invalid data input	Alert message displayed fail to login	pass
		Valid data input	Logged in successfully redirected to home page	pass
		Keep logged in	Always logged in if no log out	pass
	Edit profile	Invalid data input	Alert message on invalid input fail to update	pass
		Update profile with valid input	Update successfully and update current information in application	pass
	Change Password	Invalid data input	Alert message to request correct input	pass
		Update with valid input	Update successfully	pass
	Manage address	add address with address	Marker on map and zoom on map, add successfully	pass
		add address with GPS	Marker on map for current location, on click save to save in address list	pass
		delete address	Address deleted	pass
	Feedback	any input	Feedback saved	pass
	Account Management	Manage administrator	create new admin	Created successfully with success message
edit admin with valid input			Updated successfully	pass
edit admin with invalid input			Alert with error message, fail to update	pass
Reset password			Reset password to default password	pass
delete administrator			Admin deleted	pass

Table 7: Test plan of the system module and results (cont.)

Module	Functions	Test	Expected Result	Actual Result	
Account Management	manage application	accept application	application status and user role updated	pass	
		reject application with reason	application status updated	pass	
		reject application without reason	Alert message required administrator to provide reason	pass	
Service Management	manage service	create service with invalid input	Alert with error message, fail to create	pass	
		Create service with valid input	Create successfully, displayed success message	pass	
		edit service with invalid input	Alert with error message, fail to update	pass	
		Edit service with valid input	Update successfully, displayed success message	pass	
		Delete service	Service deleted	pass	
Appointment Management	search	Search service with category	Display related service in list view	pass	
		Search service with keyword	Display related service in list view	pass	
		Onclick company name	Redirected to company details page	pass	
		Onclick company address	Redirected to google maps with company address	pass	
	Onclick call	Redirected to dial with filled phone number	pass		
	Manage appointment		Create new appointment with incomplete data	Alert message with error message to remind user	pass
			Create appointment with complete data	Created successfully with success message	pass
Edit appointment information			Updated successfully with success message	pass	
		Cancel appointment	Cancel successfully	pass	

Table 8: Test plan of the system module and results (cont.)

Module	Functions	Test	Expected Result	Actual Result
Appointment Management	Manage appointment	Accept appointment without select appointment date and time	Alert messages require date and time	pass
		Accept appointment with appointment date and time	Accepted successfully with success message	pass
		Rejected appointment	Rejected successfully with success message	pass
		On click Start service	Appointment in progress and cannot be cancel and reject	pass
		On click Service Done without price	Alert message required to fill in service price	pass
		Pay by cash	Update status completed, enable rate by user	pass
		Rate service	Updated successfully	fail

5.2.2 User Acceptance Testing

User testing acceptance is a method to test on the verified and accepted level of the user on the application before it is published, due to time limitation only 6 users take part in this testing. The outcome evaluation is stored in the Table 9 and Table 10. It can be concluded that the application is satisfied or fulfils respondents' expectation.

Figure 16: Result of User Acceptance Test on User Interface Evaluation

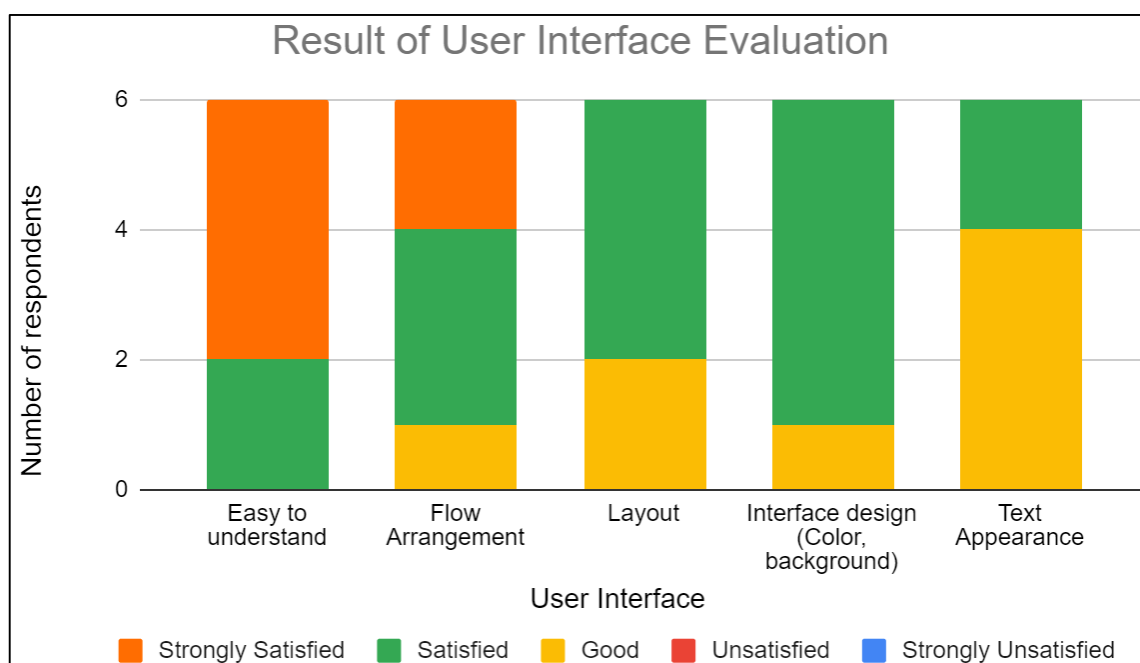
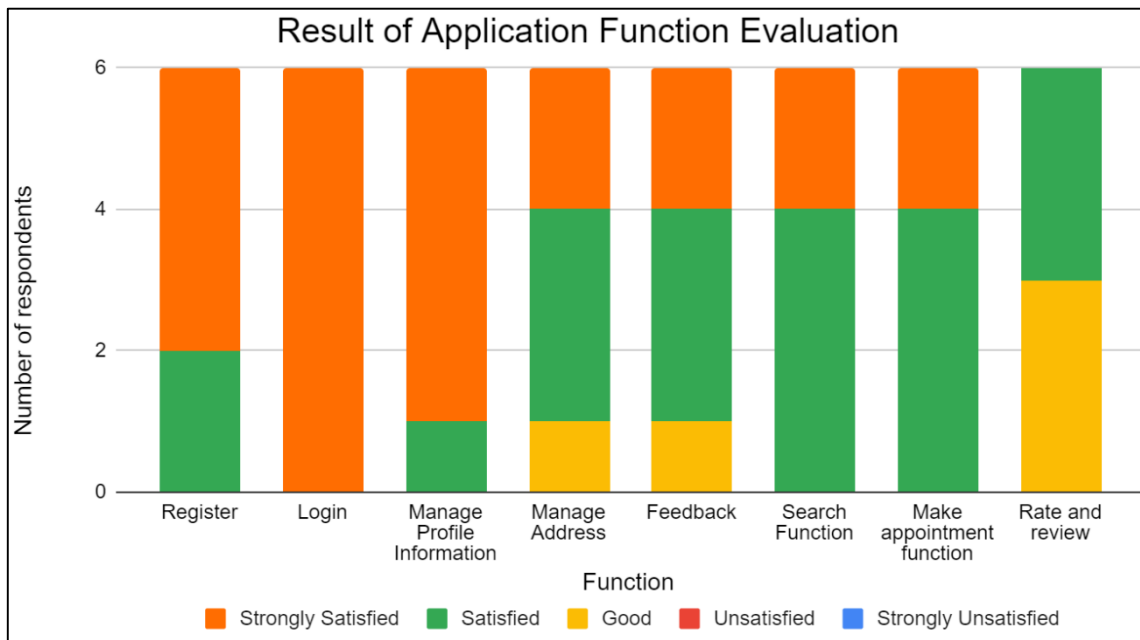


Figure 17: Result of User Acceptance Test on Function Evaluation



Based on the Figure 16 and 17, a conclusion can be made where most of the respondent satisfied with the application’s interface for the flow, arrangement, layout, interface design, ease to understand and text appearance while the result for function is almost same with the interface. User satisfied on most of the function. Nevertheless, some respondents think that the text appearance and rate and review function can be improved more.

5. Conclusion

In conclusion, Service Finder is developed successfully with five modules which are the profile management, account management, service management, appointment management and rate and review management. However, there are some limitations to this application. The limitation of the application is it does not consist of in-app live chat for the user and service provider to discuss in the application. Besides, it does not develop with online payment method to enable user pay by cashless method. Furthermore, there is no notification function in this application to push notification when user send request to service provider.

Some improvement can be carried out to enhance the application in the future. Firstly, online payment method such as Touch n’ Go e-wallet and FPX online banking payment method can be implemented to enhance the application with cashless payment method. Secondly, in-app chat and live chat can be included in the application to ease the communication between service provider and user. Moreover, the notification function should be developed to increase the effectiveness of the application to notice users on new status updated. Lastly, a web-based version of this application can be implemented to ease the user to find service provider without required them to install the application in their devices to reduce storage occupied in user’s device.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Appendix

	Name	Duration	Start	Finish	Predecessors	Qtr 4, 2021			Qtr 1, 2022			Qtr 2, 2022		
						Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
1	<input checked="" type="checkbox"/> Project Planning	7 days	10/6/21 8:00 AM	10/14/21 5:00 PM										
2	Proposal preparation	7 days	10/6/21 8:00 AM	10/14/21 5:00 PM										
3	<input checked="" type="checkbox"/> Analysis	5 days	10/17/21 8:00 AM	10/21/21 5:00 PM	1									
4	Current system study	5 days	10/17/21 8:00 AM	10/21/21 5:00 PM	1									
5	Understand User Requirement	3 days	10/17/21 8:00 AM	10/19/21 5:00 PM	1									
6	<input checked="" type="checkbox"/> Data Gathering	14 days	10/24/21 8:00 AM	11/10/21 5:00 PM	3									
7	Questionnaire	14 days	10/24/21 8:00 AM	11/10/21 5:00 PM	3									
8	Report Writing	7 days	11/11/21 8:00 AM	11/21/21 5:00 PM	6									
9	<input checked="" type="checkbox"/> Data Flow and Decision Analysis	7 days	11/22/21 8:00 AM	11/30/21 5:00 PM	8									
10	Analysis Data Flow	7 days	11/22/21 8:00 AM	11/30/21 5:00 PM	8									
11	<input checked="" type="checkbox"/> Prototyping 1	28 days	12/1/21 8:00 AM	1/9/22 5:00 PM	9									
12	Design	7 days	12/1/21 8:00 AM	12/9/21 5:00 PM	9									
13	Implementation	21 days	12/12/21 8:00 AM	1/9/22 5:00 PM	12									
14	<input checked="" type="checkbox"/> Prototyping 2	28 days	1/10/22 8:00 AM	2/16/22 5:00 PM	11									
15	Design	7 days	1/10/22 8:00 AM	1/18/22 5:00 PM	11									
16	Implementation	21 days	1/19/22 8:00 AM	2/16/22 5:00 PM	15									
17	<input checked="" type="checkbox"/> Prototyping 3	42 days	2/17/22 8:00 AM	4/17/22 5:00 PM	14									
18	Design	7 days	2/17/22 8:00 AM	2/27/22 5:00 PM	14									
19	Implementation	35 days	2/28/22 8:00 AM	4/17/22 5:00 PM	18									
20	<input checked="" type="checkbox"/> Prototyping 4	47 days	4/18/22 8:00 AM	6/21/22 5:00 PM	17									
21	Design	7 days	4/18/22 8:00 AM	4/26/22 5:00 PM	17									
22	Implementation	40 days	4/27/22 8:00 AM	6/21/22 5:00 PM	21									
23	Report Writing	7 days	6/22/22 8:00 AM	6/30/22 5:00 PM	20									

References

- [1] C. Huang and M. Hsu, "Acceptance of location-based service technology—The Facebook Check-in function," 2013 10th International Conference on Service Systems and Service Management, 2013, pp. 809-812, doi: 10.1109/ICSSSM.2013.6602565.
- [2] Dennis, A.R., Wixom, B., & Tegarden, D.P. (2015). Systems Analysis and Design: An Object-Oriented Approach with UML.
- [3] L. Allison and M. M. Fuad, "Inter-App Communication between Android Apps Developed in App-Inventor and Android Studio," 2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft), 2016, pp. 17-18, doi: 10.1109/MobileSoft.2016.018.
- [4] M. N. Alanazi, "Basic Rules to Build Correct UML Diagrams," 2009 International Conference on New Trends in Information and Service Science, 2009, pp. 72-76, doi: 10.1109/NISS.2009.252.
- [5] (2021). foodpanda - Grocery Delivery (Version 5.180.0) [Mobile app]. Retrieved from Google Play Store.
https://play.google.com/store/apps/details?id=com.global.foodpanda.android&hl=en_US&gl=US. [Accessed December 13, 2021].
- [6] (2021). Grab - Transport, Food Delivery, Payments (Version 5.180.0) [Mobile app]. Retrieved from Google Play Store.
<https://play.google.com/store/apps/details?id=com.grabtaxi.passenger>. [Accessed December 13, 2021].
- [7] (2021). ServisHero: On demand services (Version 3.5.25) [Mobile app]. Retrieved from Google Play Store.
https://play.google.com/store/apps/details?id=com.servishero.consumer&hl=en_US&gl=US. [Accessed December 13, 2021].