Faculty of Electrical and Electronic Engineering Technology

DEVELOPMENT OF VOICE-ACTIVATED ORDERING USING ANDROID APPLICATION FOR RESTAURANT

LOI MING ONN

Bachelor of Computer Engineering Technology (Computer Systems) with Honours

2021

DEVELOPMENT OF VOICE-ACTIVATED ORDERING USING ANDROID APPLICATION FOR RESTAURANT

LOI MING ONN

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Computer Engineering Technology (Computer Systems) with Honours

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I declare that this project report entitled "Development Of Voice-Activated Ordering Using Android Application For Restaurant" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	[illegible]
Student Name	:	LOI MING ONN
Date	:	28/2/2022

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Engineering Technology (Computer Systems) with Honours.

Signature	:	[illegible]
Supervisor Name	:	Dr. Adam Wong Yoon Khang
Date	:	3/3/2022
Signature	:	
Co-Supervisor Name (if any)	:	
Date	:	

DEDICATION

To my dear Mother, Father, family, friends and lecturer:

I dedicate the research project to you all for several reasons.
As my parents and my family, all of you are the main part of my life. Each moment, each decision, each situation and each result and subsequent, I have never been abandoned and even borne together with you all. I can't proceed further without you, my mother, father and my dear family.

For my friends, who accepted me as a company to them in these year of university life, who supported and trusted me in up and down, no matter what happens you all always lean a hand to me, I am truly grateful.

To all the lecturers that involved in my years of degree, everyone of you have supported me throughout the journey, any words of wisdom are contributed and shared by you all will be remembered, and motivates me to be better, especially in conducting this research.

Once again, very thanks to everyone I listed here, as they are the people who supported me and contributed the most to build up this invaluable research project and contribute to the society in the future.

ABSTRACT

Lifestyle today is very challanging nowadays and thus saving time is one of the most important factor for every industry field, especially food and beverage service which mainly from restaurant. A crowded restaurant with a long queue to take order has cause problem to many customers as they wanted to have easier and faster way to make their order without waiting too long. Customer who just need simple and fast checkout order have difficulties to repeat browsing the menu as they needed to take longer time and increase their waiting time. The main objective of this research is to develop a simple and user-friendly system application for menu ordering system that can improve the customer usage by also implementing voice-recognition technology as a possible method for these targeted customers. The system will be also adding in a basic database to apply two-way communication between the system and data input by the users for data collection. The work scope of this project includes system design that will made by free prototyping and designing tools. The application of the menu ordering system will be also developed with AndroidOS platform with Extensible Markup Language (XML) and JAVA programming language. Then the voice-recognition technology which also using Google Speech-to-Text (STT) and its function is implemented and combined into the application. An data collection of result will be analysed based on the accuracy for the voice-recognition method and comparison of time taken between these the common traditional manual method versus voice ordering method. The methodology is involving free assisting tool such as StarUML Draw.io, Adobe XD that are used to illustrate the workflow, system cycle and also the Graphical User Interface (GUI) of the application. Then the the application is programmed and structure with the Android Studio that based on AndroidOS system's native users that use other devices than computer. There is significant consistency on the comparison for the results, where its percentage accuracy between preliminary result and expected result only 10% difference, 1.9seconds of weighted mean in manual ordering and 3 seconds for voice-ordering method, given that some of addition of extra features and improvement. Overall, the methodology has succeed in giving out the main required result from data collection and analysis and showing the potential of improvement on quality of menu ordering system is exist by implementing voice recognition method into it and improve restaurant operation.

ABSTRAK

Kehidupan kini yang sangat bercabar telah menjadikan penjimatan masa salah satu faktor penting bagi setiap bidang industri, terutamanya dalam industri makanan dan minuman yang berkaitan dengan pengurusan restoran. Restoran yang terlebih pelanggan telah memberikan masalah kepada golongan pelanggan yang ingin dapatkan pesanan secara mudah dan cepat tanpa menunggu lama. Golongan pelanggan tersebut juga berasa sukar untuk membuat pesanan dengan menu yang berulang-ulang dan masa diambil untuk memesan juga meningkat. Tujuan utama kajian ini adalah untuk mengembangkan aplikasi sistem pemesanan menu restorang yang mudah dan mesra pengguna dengan mengimplementasi dan menggabungkan teknologi pengecaman suara sebagai salah satu cara pemesanan. Sistem aplikasi ini juga mengandungi fungsi pangkalan data yang mampu dijadikan sebagai cara komunikasi duala hala antara sistem dengan data pesanan daripada pelanggan. Skop projek kajian ini mengandungi reka bentuk sistem yang akan dikembangkan dengan peralatan perisian reka dan prototaip yang percuma. Aplikasi ini juga akan dikembangkan dan dibina dengan platform sistem operasi AndroidOS dengan XML dan JAVA sebagai bahasa pengaturacaraan. Pembentukan carta aliran pelan projek dan sistem serta muka pengantaraan pengguna grafik (GUI) juga dibentuk dengan peralatan yang percuma seperti StarUML, Draw.io dan juga Adobe XD. Penglibatan teknologi pengecaman suara dalam aplikasi adalah "Speech-to-Text" daripada Google dan fungsinya akan digabbungkan ke dalam aplikasi. Pengumpulan data daripada hasil kajian akan dianalisi berdasarkan ketepatan daripada penggunakan fungsi pengecaman suara dan perbandingan antara masa diguna untuk membuat pesanan dengan cara secara manual atau penggunaan fungsi pengecaman suara. Hasil kajian penyelidikan mencapai prestasi yang stabil antara keputusan awal dan keputusan akhir dalam segi ketepatan fungsi pengecaman suara dalam pemesanan dengan perbezaan sebanyak 10%,selain itu, masa diambil untuk pemesanan secara manual juga mempunyai 1.9 saat sahaja dan pemesanan secara pengecaman suara hanya berbeza sebanyak 3 saat berbanding antara keputusan awal dan keputusan akhir kerana perkembangan dan pertambahan fungsi-fungsi baru dalam applikasi. Hasil kajian ini telah menunjukkan potensi dalam peningkatan kualiti pemesanan menu juga wujud dengan pelaksaan konsep penggunaan cara pengecaman suara dalam sistem pemesanan di restoran hari ni.

ACKNOWLEDGEMENTS

In this research project, I would like to take opportunity to express my gratitude and most appreciation to my supervisor, Dr. Adam Wong Yoon Khang for his invaluable guidance and suggestion, wise words and passionate response throughout the progress of the project.

I am also taking the chance to express my thankfulness to my university, Universiti Teknikal Malaysia Melaka (UTeM) that gave me this precious moment for any supported I need and giving me time to complete the project. I will be also remembering my friends' contribution on thoughts, guidelines and idea sharing to ease and help me on my research project.

I give my greatest appreciation to my parents and family, who supported me for everything, guided me and be with me throughout the process of finishing the project. Their support on my journey to complete this project task are the key of my success on the research project.

And lastly, I would like to thank to everyone else who are not listed here and involved in providing any inspiration, giving thoughts and encouragement for me to continue and finish up the project. Every moment spent to contribute me on any progress are all appreciated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

s	-	seconds
%	-	percentage

LIST OF ABBREVIATIONS

STT	-	Speech-to-Text
XML	-	Extensible Markup Language
GUI	-	Graphical User Interface
SHD	-	Smart Home Device

LIST OF APPENDICES

LIST OF FORMULA

Percentage Accuracy, P	-	$percentage, p = \frac{frequency\ (Accurate), f}{Number\ of\ Trial, n}\ x\ 100$
Weighted Mean	-	$Weighted\ Mean, \sum \bar{x} = \frac{Time\ Taken\ to\ Order, X}{Number\ of\ Trial, N}$

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter will describe the introduction of the topic, Development of Simple Two-Way Communication Android Application Systems for Restaurant with Add-on Voice-Recognition Function amd databases. The whole chapter will be covering problem statements, objectives, and the work scope of the project. The whole project goal can be achieved by finding every kind of solutions to solve all the problem stated in this report and completing the objectives that are planned on this project report.

1.2 Project Background

Lifestyle today has a huge advancement, in especially between cities which are always crowded, and busy. As cities are always meant and mentioned to be a busy place, especially for the society advancement that had led everyone who lived in city being time conscious, where every single unit of time have to be fully utilised just not to get wasted for nothing achieved. Thus, it affects every kind of industry field that has to be always improving and evolving in terms of efficiency, productivity and quality.

In this project, food and beverage industry is targeted, where it focuses are on one of the society's daily needs, which is restaurant performances on how much efficiency they can achieved for their menu ordering system. In a summary of the main research topic, restaurants' efficiency has to be improved, especially towards services and convenience, where customer can have the best, easiest and fastest way to make their order collected by the ordering system. Comparing the traditional way which normal workers who in charged of taking orders from customers that uses pen and paper to jot down all the info, By developing a simple mobile application for the customers, the whole operation performance for the restaurant can be possible to improved, and customers will have better impression and satisfaction on the services provided.

For mobile application, there are a lot of menu ordering applications available today, from various branding of restaurants, and presented in different way. For example, we have large franchise company like McDonald’s and KFC where they started to use self-ordering kiosk to pick up orders from customers. Or we have food ordering application like Pizza Hut or Domino’s where we can order by online. To seek an opportunity of improvement something is possible to be added on and make the application more interesting, which would be one of the demanding feature that everyone would like to use on other application nowadays which is involvement of voice-recognition technology.

Overall, the mobile application should contain a simple, easy-to-understand Graphical User Interface (GUI), a voice recognition or assisting method, a functionable and usable application system, a notification system and finally a backend database that will be used for two-way communication functions between customers and the restaurant operators. By having these requirements implemented inside the application, it is estimated that the ordering system of a restaurant can be improved and more efficient in shorter time taken.

1.3 Problem Statement

In order for a research project to have directions how to determine its objectives for conducting it. Problem statements are musts in order for us to find what are the flaws still existing in current available system and what improvement, modification we can improvised.

For a restaurant's ordering system, in customers' perspective, convenience is the key problem that I found out in their concern. Customers in restaurant today lack of convenience for making an order when the restaurant is in high peak session that fully filled with a lot of other customers. Making manual order by physically from the staffs are not going to fulfil every customer. There are also certain category of customers who just wanted to have an alternative method like using voice to order that considered as their preferred method makes most of the existing ordering systems just not able for them to use on. Based on [1], the traditional, manual ordering system that done by staffs will cause some human errors such as giving wrong information of bill, unreadable handwriting by waiter who just clip down everything in a instant and wrong sequences of order that will cause dissatisfaction towards restaurant.

Besides, as the background mentioned about how busy an urban area, or a large city can be, time is their very important factor especially for customers such as workers that have limited time to spend for getting their meal up are forced to take a long queue, just to ordering their food at the counter or a kiosk machine. The time taken made by the system is worse providing that every customer also have to browse through the whole menu which is annoying to loyal or busy customers who just wanted to have their meal ordered as easiest and fastest as possible, where we categorised them as "express customers". Slow ordering system would affect the restaurant reputation, productivity and most importantly, customers' satisfaction towards the services. As the journal made by [2] mentioned, customers often must wait for the waiter when it comes to ordering food, and it will make them anxious about the services and feel exhausted by just waiting and wasting time.

Moreover, in restaurant perspective, what happens when or if a restaurant become famous but still uses a service without technology, such as making order using pens and papers using manpower as what [3] stated, it would be a trouble for a basic restaurant as the amount of manpower is always limited when the whole restaurant is filled with customers. The whole team is short of waiters to take order one by one by running to each customer one by one just to take down each orders. Waiters will eventually not catching up the orders in this short bursting time and it will affecting the whole operation for the restaurant.

1.4 Objectives

There are several objectives that I wanted to achieve throughout this research project, these are the points of goal for my objectives to be achieved to complete my research purpose:

- To study about a system application design with the consideration of restaurant operation and improvement of menu ordering procedure.
- To develop an application that contains at least basic and useful functions for restaurant usage and involving voice-recognition technology.
- To analyse the possibility of two-way communication and order data collection with the help of database management system.

- To observe the accuracy result of voice ordering method and compare it with traditional manual ordering method for effectiveness.

1.5 Scope of Project

The project scope targered in this research will be related to all the objectives stated in this report. They are listed below:

- System is planned to be designed with the aid of free prototyping and Graphical User Interface designing tools.
- The application of the menu ordering system will be built in AndroidOS platform with XML and JAVA programming language.
- Application will be combined with voice-recognition technology which is Google Speech-to-Text (STT) function.
- The result is expected be analysed based on the accuracy of the voice-recognition function, and the time taken to order with different method, and hence to find the achievement and improvement towards menu ordering.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

The literature review section will be explaining and focusing about every idea, inspiration, provided information and helpful tips from various article made by all experts around the world that might be contributed to complete the project. In this study, every literature review that I picked out of my list of studied literatures will briefly cover the main contents they used for their project, the topics involved, advantages, disadvantages and comparison between previous researches. Other than that, useful information that might to be totally related such as analysis data and hardware's experience that used by other research. By analysing each of the related article found out throughout the project timeline, it will certainly contribute a lot of information for me to building up my research project's progression. Few literature and research resources that provided the most beneficial information for reviews on their strength and weaknesses so a comparison can be made among them and thus providing a better study, guidance and information to continue on research.

2.2 Voice-Recognition /Voice-Assistance /Voice-Activation Technology

Voice-recognition technology has been a very interesting field for every industry to migrate into their business and field. There are many researches made in previous time by various researchers and composed by authors around the world regarding the topic.

According to [4], voice usage has become an increasingly popular in Graphical User Interface (GUI) channel, mainly contributing to current trend of wearables, vehicles and home automation system. In his opinion, voice has affected our lifestyle especially in technology field and become our everyday fixture. [5] also mentioned that voice assistants are software agent that can interpret human speech and respond via synthesized voices that helps users to complete specific tasks with voice communication.

There are several kinds of example on how voice-recognition technology is added or migrated to another field of technology, for example based on the book written by [5], voice-recognition technology has improved restaurant operation by implement it into the current existing system available for it. Several functions like chatbot, which is a program that capable of carrying human-like chat conversation data, then it can answers and frequently ask questions, assisting customers and order tracking.

Figure 1: Robot Working as a Concierge at a Japanese Sushi Restaurant [6]

Figure 1 is a robotic worker in a restaurant, they can perform a variety of different function in the restaurant [6]. Applying robotic field with voice-recognition method can be revolutionise a restaurant operation by decreasing manpower on ordering servicing. These kind of technology is very popular in advanced country such as Japan, United State or European regions.

There are several benefits of voice-recognition or assisting technology towards human and social lifestyle. One of it is mentioned by [5], where users can use them to do a lot of tasks easier by voice communication to voice assistance, as they have the ability to send/read messages and answer any basic informational queries such as weather. [7] also gives out good examples of benefit where voice can be translation to text or vice versa, via speech-to-text and text-to-speech functions and map the information received as a data in programming too. [8]'s research also explored how voice-activated smart home devices (SHDs) affect users in voice-activation technology towards daily life.

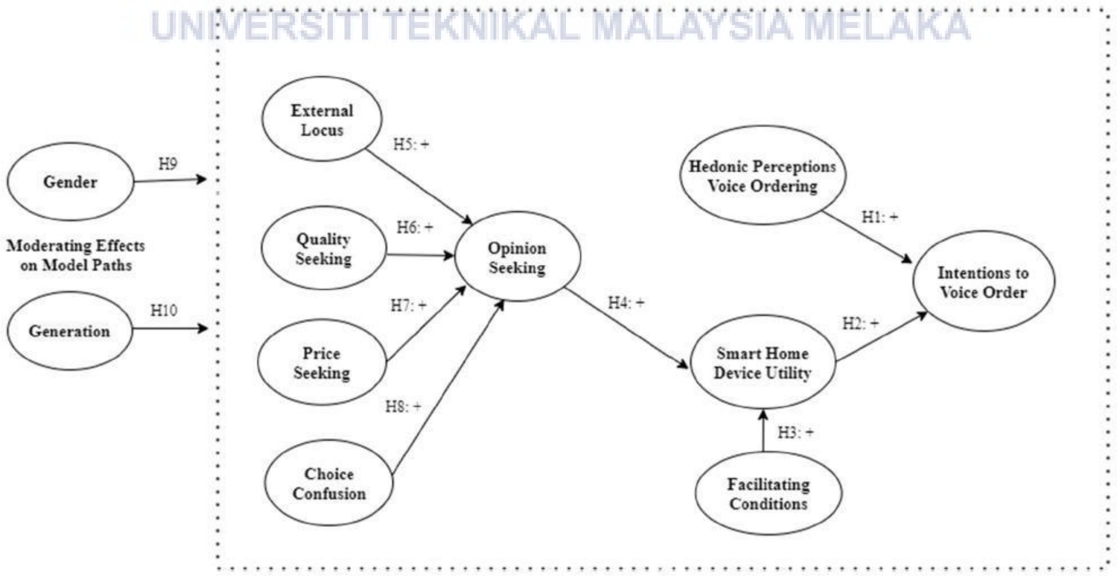


Figure 2: Factors that Influence Intention to Voice Order Goods and Services Using SHDs [8]

As Figure 2 shown, these are the flow of factors on how consumers affected by SHD to have the intention to voice ordering goods and services starting from information seeking until have the intention to use voice as their main method for ordering. SHD enable to answer various type of answers based on their pre-defined library set by the developer company such as customers who use SHDs can seek for opinion from it, and they trust the facillitating conditions provided by SHDs' utility program that will increase their attention towards using them for making order.

Thus, it is shown that important voice is towards current trend of technology implementation. A study is important to make a research on how much potential can voice-recognition technology helps in current workplace, especially in restaurant which is a high communication demand spot on their operations.

2.3 Application Development for Restaurant/ Menu Ordering System

An ordering system from the restaurant nowadays could not survive longer or improved in a better way without the help of modern technology. Traditional way of menu ordering system maintained by restaurant is starting to have a backlog once the operation starts to be busy and reach an uncontrollable peak time.

There are several factors on why a restaurant must have a modern system that relies on technology nowadays. One of the main reason was mentioned by [2], where they found issues like customers often must wait for the waiter to service when it comes to food ordering process, as the restaurant might be too crowded, and this will make customers feel mentally anxious and exhausted about the services. [3] also stated that by developing a restaurant ordering system, it can be possible to replace traditional method that currently used by most of the restaurant, such as pen and paper writing in order to accelerate the ordering and processing food in the restaurant.

There are several technical methods and approaches made by researchers and authors to build the software, one of the way used by [2] is the creation of QR code as data collection method for customers to input their data to the system when they wanted to make orders. Customers scan the QR code then they can access the menu webpage (or online menu card) to pick their meal set or items. The example of the method is shown in Figure 3.

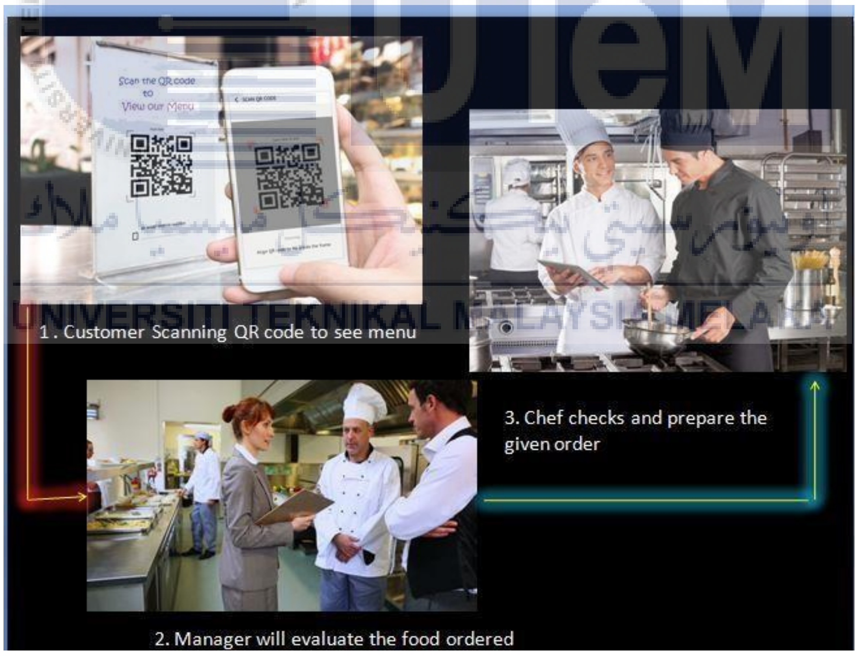
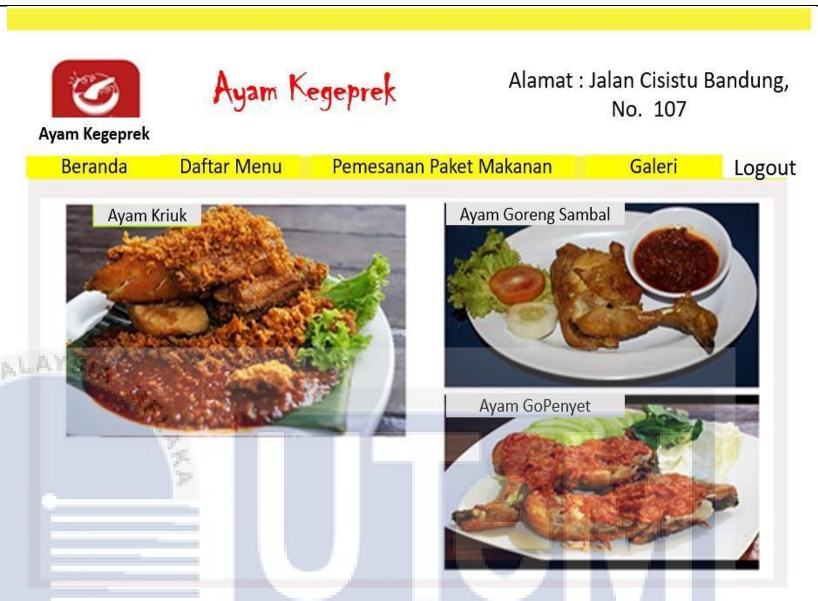
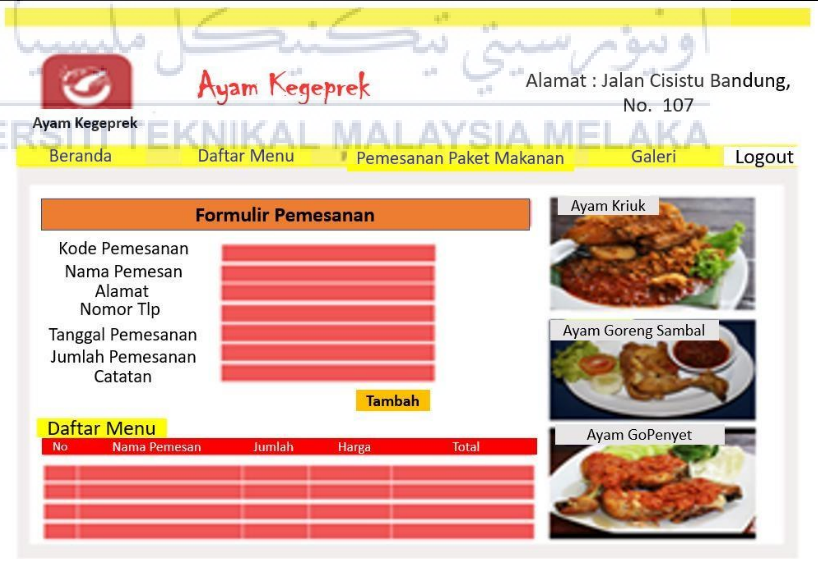

Scan the QR code to View our Menu
1. Customer Scanning QR code to see menu
3. Chef checks and prepare the given order
2. Manager will evaluate the food ordered

Figure 3: Example of QR Code Application [2]

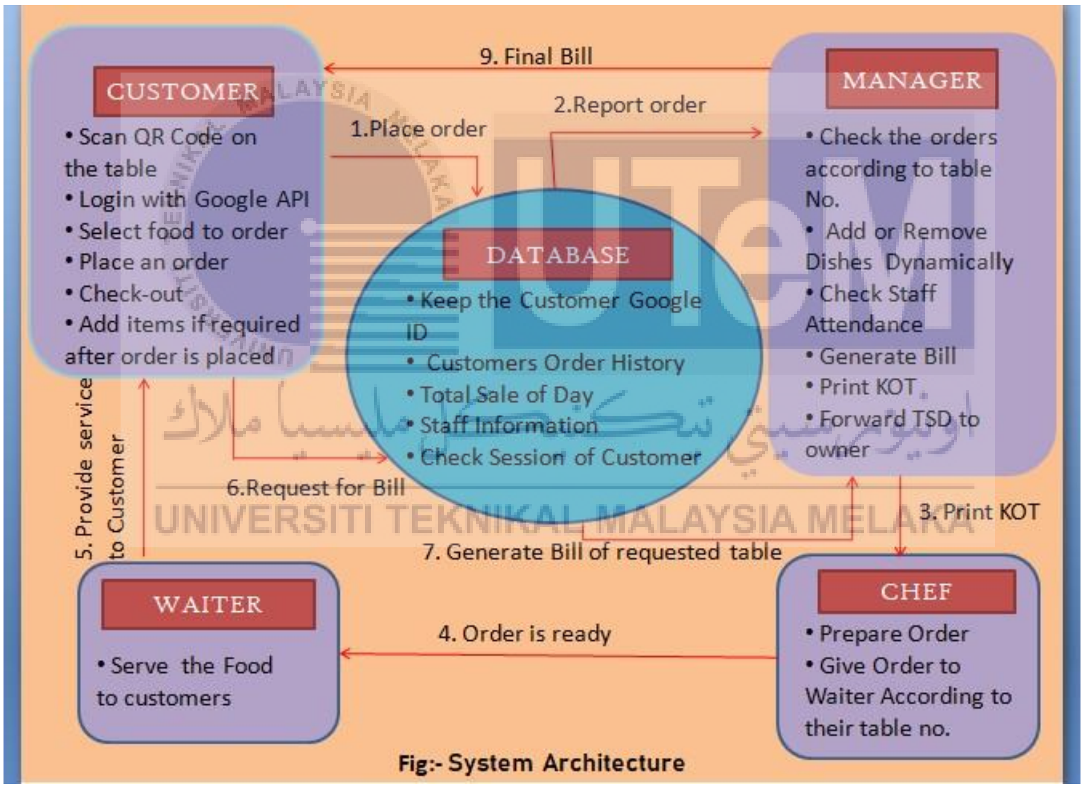
Besides, GUI design played an important role in a menu application, where a well made design can attract more customer usage and a simple GUI can save a lot ot time for interacting with it. Developer can add in the important information, such as the menu item, order registration and confirmation, customer personal information, ordering number and so on. An example can be seen from [9] is shown in Table 1.

Table 1: Example Of GUI Design For Food Ordering System From [9]

Page Content	Figure
1.Homepage	 Ayam Kegeprek Ayam Kegeprek Alamat : Jalan Cisistu Bandung, No. 107 Beranda Daftar Menu Pemesanan Paket Makanan Galeri Logout Ayam Kriuk Ayam Goreng Sambal Ayam GoPenyet Figure 4: Menu Home Page GUI [9].
2. Food Ordering Page	 Ayam Kegeprek Ayam Kegeprek Alamat : Jalan Cisistu Bandung, No. 107 Beranda Daftar Menu Pemesanan Paket Makanan Galeri Logout Formulir Pemesanan Kode Pemesanan Nama Pemesan Alamat Nomor Tlp Tanggal Pemesanan Jumlah Pemesanan Catatan Tambah Daftar Menu No Nama Pemesan Jumlah Harga Total Ayam Kriuk Ayam Goreng Sambal Ayam GoPenyet Figure 5: Food Ordering Page GUI [9].

Referring to Table 1, both Figure 4 and Figure 5 homepage and food ordering GUI made by the respective researchers contain simplet but clear information such as menu item, order number, menu registration and so on. This GUI example shows that the information is clearly represented with label and customer can interact with the menu application easily.

Besides GUI design, system workflow design is also a must to explain on how suppose the application should be work or how the software works in a flow. There are examples like system cycle or system architecture can be applied on the system flow.



9. Final Bill
CUSTOMER
• Scan QR Code on the table
• Login with Google API
• Select food to order
• Place an order
• Check-out
• Add items if required after order is placed
1.Place order
2.Report order
MANAGER
• Check the orders according to table No.
• Add or Remove Dishes Dynamically
• Check Staff Attendance
• Generate Bill
• Print KOT
• Forward TSD to owner
DATABASE
• Keep the Customer Google ID
• Customers Order History
• Total Sale of Day
• Staff Information
• Check Session of Customer
5. Provide service to Customer
6.Request for Bill
7. Generate Bill of requested table
3. Print KOT
WAITER
• Serve the Food to customers
4. Order is ready
CHEF
• Prepare Order
• Give Order to Waiter According to their table no.
Fig:- System Architecture

Figure 6: System Architecture [2]

As we can see from the Figure 6, there are different modules (or actors) of the system (Customer, Manager, Waiter, Chef, Database) represented in each box and each modules have their respective functions with their attributes based on each actors, followed by relations to show how each module interact with each other in system functions. System architecture with well-arranged functions will prevent any misinterpretion of data and misunderstanding between developers during the programming and coding stage.

Another important element inside system design is determination of the workflow cycle. An example by [3], which is represented by a figure of planning stage for system design by using SDLC (Software Development Life Cycle) process concept which is a type of stage processing method to develop an how the workflow stages should be.

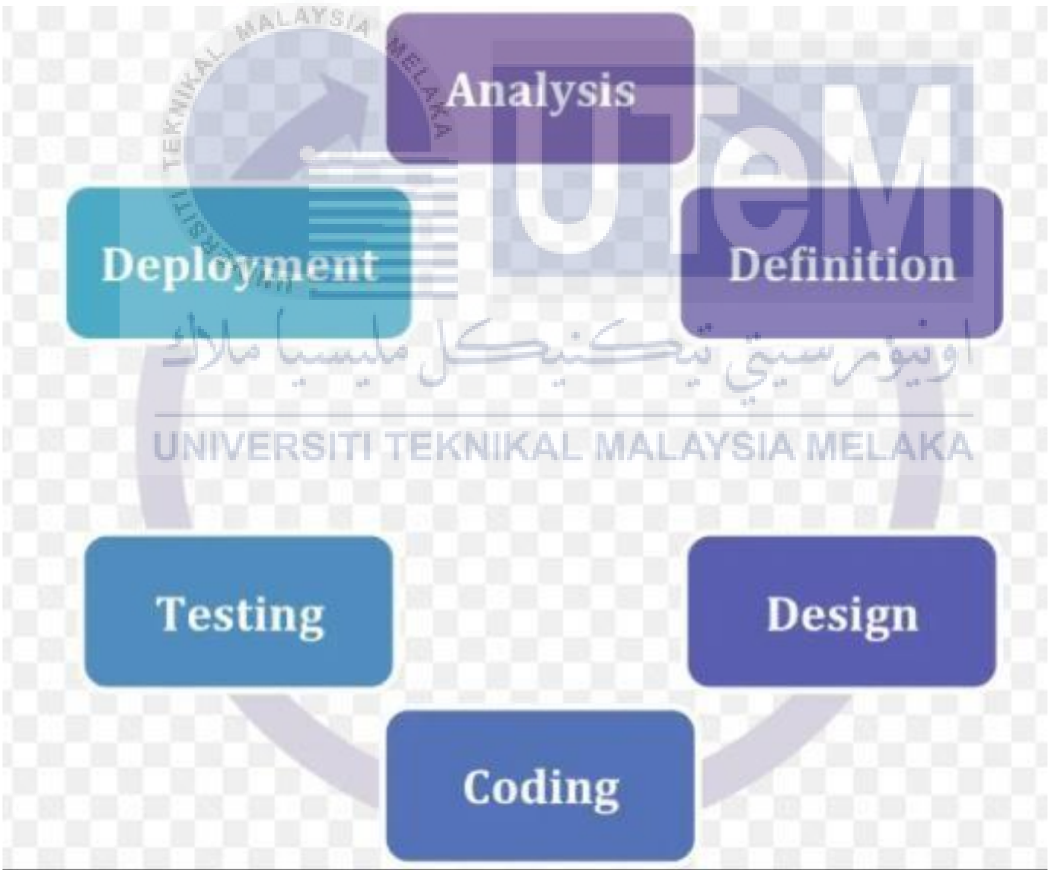


Figure 7: Example of an SDLC Cycle With Different Phases for a System Development [3]

Referring to Figure 7, the purpose of SDLC in system design is to have a goal on the planning stage for explaining activities related to project management and giving information to analyst and programmer how the system works based on the flow planned. Without SDLC, the system has no direction and cannot relate with each other, making the system unable to work together and failure is expected to be happen easily. Figure 7 shows up the whole cycle of SDLC should at least have definition, design, coding, testing, deployment and analysis stages.

Then, every system is supposed to have some basic requirement from other software or component to help the completion of the system process, such as cloud computing services, data parsing method, one of the example of assisting software is used by r who used JQuery to develop the proposed application and AJAX to receive or transfer information between system in the server or databases.

Beside a web-based system that users can access the system through webpage or browsers, native system is also available as a method to build our application or system for other device, such as phone and tablets. [11] made his system which is available in AndroidOS, which is a famous native platform for the application to reach out more targeted users based on their system requirement.

In summary, to build a food ordering system, firstly system design is very important to build the software, follow by its architecture and workflow to determine how the system is going to work. After that, implementing method to transfer or parsing the data to communicate with back-end functions. A suitable platform can decide how well an application goes and targeted in the market, for different purposes and different user target. In restaurant usage, mobile application would be the best and easiest for customers as they doesn’t need to get restricted by browsers or depending other system.

2.4 Relationship of the Systems, Back-end System and Real Time Databases

A data relationship in a system is very important to link all the functions of the system together, so the data can be process smoothly without error, and contains complete information that is required by the developers. Therefore, a back-end development is playing the important role for the system to be functional. [12] has shown a very good example on how a system supposed to be work at behind.

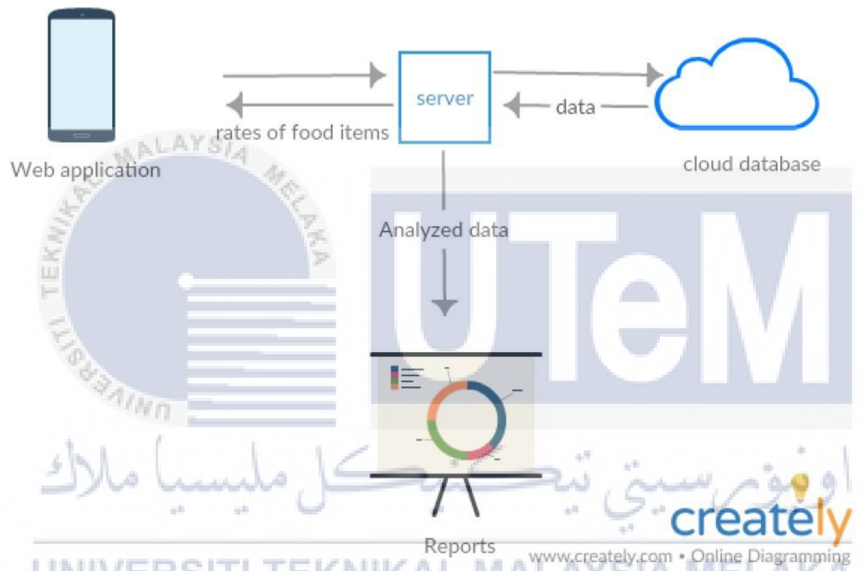


Figure 8: Data Parsing Flow from Application to Cloud Database [12]

Based on Figure 8, the application sends its data to the cloud database through the server and replies any required responses received to the device. The server also collects the data and display out through the application as the result so users can check the result they wanted from the application. The data parsing between application and cloud database through the server can be also collected and analysed in case they are needed for other operation and calculation.

To process this data transferring, some designs or plans that can show each relationship of the data and how the data are supposed to be store inside the memory. These data will be stored inside a centric unit of system storage which is we called as database. Database will collect all the data required to store and manipulates by its administrators. Then these data will be arranged with a specific sequence. The sequence can be planned with a diagram which we called as Entity-Relationship (E-R) diagram, which it could display out all entity inside the database with its sub attributes that related to them.

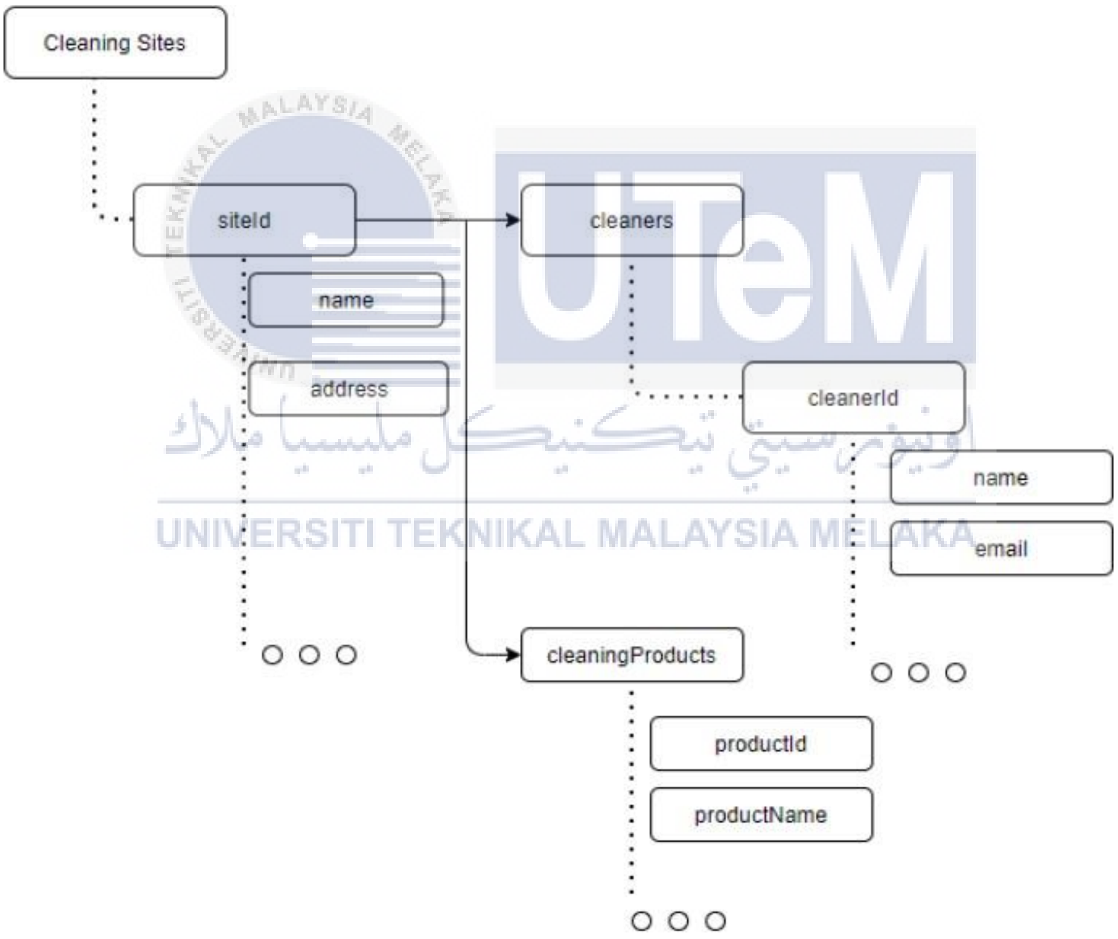
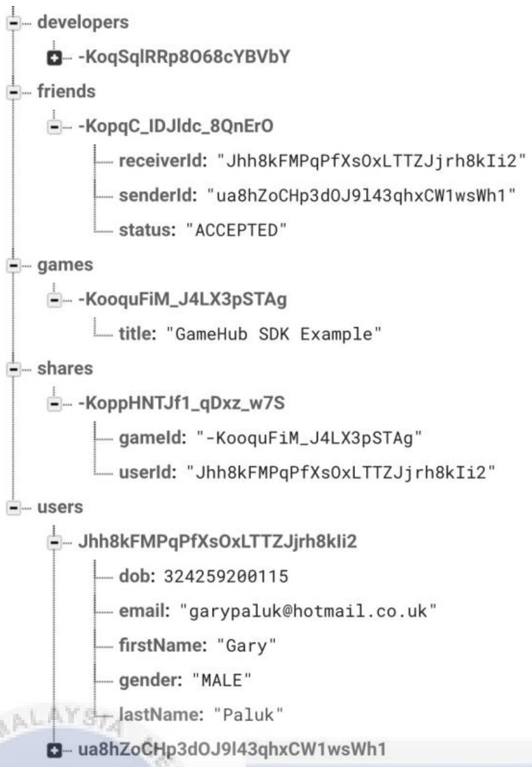


Figure 9: Database structure from a System [13]

Figure 9 shows developer’s database structure clearly representing the system data arrangement and storage in a E-R diagram, which normally linked with few tables of data that have different attribute as all data will share the same data storage or database The main attribute in each table will act as a “category” for every sub-attribute so they can categorise themselves on their respective relatable table. Based on the image, “siteid” is the main table that contain information such as name, address which are main information, then it acts as key attribute to the tables related to which are “cleaners” and “cleaningProducts” table that also have their own attributes inside the table.

Choosing right database platform and implementation of database are also important key to be discussed about. There are some methods used by researchers on their database part of the system. One of it is shown by [14] that used MATLAB as their software tool which contains Audio Database Toolbox to store and filter out auditory database and utilise them to make it easier on focusing on the code algorithmic aspects. There are a lot of different database platform as well, especially real-time database, so administrator can inspect and monitor all data input into the database. One of the best available real-time database service providers is from Firebase from Google Inc.



```
developers
    -KoqSqlRRp8O68cYBVbY
friends
    -KopqC_IDJldc_8QnErO
        receiverId: "Jhh8kFMPqPfXsOxLTTZJjrh8kIi2"
        senderId: "ua8hZoCHp3dOJ9l43qhxCW1wsWh1"
        status: "ACCEPTED"
games
    -KooquFiM_J4LX3pSTAg
        title: "GameHub SDK Example"
shares
    -KoppHNTJf1_qDxz_w7S
        gameId: "-KooquFiM_J4LX3pSTAg"
        userId: "Jhh8kFMPqPfXsOxLTTZJjrh8kIi2"
users
    Jhh8kFMPqPfXsOxLTTZJjrh8kIi2
        dob: 324259200115
        email: "garypaluk@hotmail.co.uk"
        firstName: "Gary"
        gender: "MALE"
        lastName: "Paluk"
    ua8hZoCHp3dOJ9l43qhxCW1wsWh1
```

Figure 10: Example of Firebase Database Structure

Based on Figure 10, it is an example of database made by Firebase where every attribute in their tables (for example “receiverId” attribute in the database table “friends”) have its own unique generated key, then developers can use the generated key to connect and relate with other table, so data input by user can be parsed into the related table (based on the figure the related one is the “user” table).

Then, we can allow Firebase to access into our system, a research book authored by [15], [16] provided some guidance on how to use Firebase started from fundamental. It covers on how Firebase’s functions such as authentication, real time database input/output, cloud storage, hosting, test lab and cloud function works inside Firebase system to control and manipulate our data according to our requirement.

In summary, back-end system also involving plans to structure a system database and server to store and manage data that produced by the system application, Different firebase platform that are invented by famous developer company provides different functions. In the case of restaurant operation, it is suitable on any kind of database platform services as long it provides the best convenience to the developers and administrators to make any changes on it.

2.5 Comparison of Literature on the Advantage and Disadvantages

This section will have a list of advantages and disadvantages I personally found based on some literatures sources that I reviewed. All the listed point are extra advantages and disadvantages based on personal understanding on the context. Each title listed will have a brief explanation on their advantages and disvantage

Table 2: List of Advantages and Disadvantages for Voice-Recognition Literatures

Title	Advantage/Strength	Disadvantage/Weakness
Continuous Authentication for Voice Assistants [4]	• Security concern. • Improvement of security in voice-recognition technology field.	• Only available for certain types of physical product.
Robots, Artificial Intelligence, and Service Automation in Restaurants [6]	• Artificial Intelligence in restaurant operation	• (No disadvantages)
Speech to Text and Vice Versa [7]	• Introduction of API • Google Text-to-Speech and Speech-to-Text	
Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants [5]	• Introduction of voice assistant devices	• Non-technical report
Consumer Acceptance of Voice-Activated Smart Home Devices for Product Information Seeking And Online Ordering [8]	• provide information on effect of voice in daily lives.	

Table 3: List of Advantages and Disadvantages for Restaurant/Menu Ordering/System Building Literatures

Title	Advantage/Strength	Disadvantage/Weakness
Food Ordering System Using Mobile Phone [11]	• The application created is based on operating system (OS) that widely used today	• No involvement of voice-recognition technology
Designing Web-based Food Ordering Information System in Restaurant [9]	• Provided an example of simple GUI design	
An Efficient Digital Ordering System for Restaurant [2]	• QR code to collect orders' data	
Mechanism of Food Ordering in A Restaurant Using Android Technology [3]	• A workflow explantion provided between system application and hardware	
	• Guidance on SDLC of an application	

Table 4: List of Advantages and Disadvantages for Relationship of the Systems and Real Time Databases Literatures

Title	Advantage/Strength	Disadvantage/Weakness
Order-Now System [13]	• Showing Firebase can be also used with other programming language. • Fundamental guideline of Firebase to beginners.	• Using high resouce consuming language which are React.js and Angular.js (Javascript)
The Definite Guide to Firebase [16]		
Deploying to Firebase as the Back End [15]		
Smart Restaurant Management and Ordering System: A Case Study [12]	• Implemented data collection of specific information like ingredient, table seats in a restaurant	• (no disadvantages found)
Autonomous Restaurant Serving System using Image Detection and Voiceprint Detection in Android Application [14]	• Voiceprint detection system with a MATLAB Audio Database	• Consume more data resources and required high hardware support

Refer to Table 2, these are the literatures that involved in voice-recognition technology. [4]’s research topic is mainly involved in security aspect of voice-assistant, so the research provided some security concern and information about this kind of technology in our society field, the prototype product made by them manage improved and enhance the security factor of voice-recognition technology field. However it is used for certain type of products only [6] shared a lot of information about artificial intelligence with involvement of voice-recognition technology, such as robots and service automation in a restaurant. While [5] introduce more about voice-assistance we have today, like Alexa, Siri and many more. Literature [8] also giving out more detail of benefits of voice-related technology in our daily life.

Based on Table 3, these literature are mainly related to research about menu application in restaurant. [3] provided the guidance about the determination of SDLC cycle on each stage during the development progress. [9] also provided their example of GUI (refer to Table which contains Figure 4 and Figure 5) and giving guidance on what basic information is required and should be included in the application, such as order number, customer info and so on. [2]’s research shared about their menu application that use QR code to get order data from customers in the restaurant. Customer just scan the given QR code and they can access and make order in their device. [11] also made his menu application in AndroidOS, which is an operating system platform that is widely used by society nowadays so the application could have large market potential especially in native devices. However, research topics from [3]. [9], [2] and [11] does not involved any implementation of voice-recognition technology or method, giving a sign of showing demanding feature that is possible to be implemented in this research topic

Referring to Table 4, the content of the paper are the list of advantages and disadvantage that are related to relation of system to real-time database. [13], [15] and [16] shared the same advantages, where they provided guidance about Firebase that are suitable for beginners or new developer who wants to familiar themselves in using their real-time database services into their application, especially Android Studio who supported the services. [13] and [15] also show their example of applying Firebase function into their application that is built with other programming language (although the programming languages they used are largely dependent on device memory usage, and cost a lot of time to maintain, update and testing due to their complexity of coding), [12] have an interesting case study that their research have a system that can collect specific data from the users or customers, which are data like the food, ingredient and table seats available in the restaurant. [14] also using another type of real-time database from MATLAB, MATLAB provided a database toolbox which is specifically for audio – named MATLAB Audio Database Toolbox, where it contains data to filter audio.

2.6 Summary of Literature Review

As a summary of the literature review, previous researchers do provide a lot of information and guidance for study on research. Most of the literatures do have its uniqueness and benefits for that research field as well as disadvantages that mostly considered as demands or that might not be able to be fulfilled by the researcher. However, their topics have been given a lot of insights towards methodology and result determination for this research.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this section, Methodology of this research project will be discussed. The content will listed out all of the pre-requirement in software/hardware that are needed to used for completing the project. A list of flow diagram and system design diagram are also included in this section as a plan or prototype for the workflow on how the project is conducted. Then the conceptual figures of procedure and steps are shown here to share about how the research project is executed. Finally, a list of current limitation will be list out that found during the whole process of methodology will be also included into this section.

3.2 List of Requirement of Software/Hardware/Tools/Materials

A list of extra requirement is presented here in order to conduct the development. They are all available free thus the cost can be reduce to only time and manpower cost. The software used for develop the application are Android Studio Development Kit, Google Firebase, Eclipse IDE for Java, Adobe XD, StarUML. There are also alternatives available however these are personal preference to use on developing the project.

- Google Firebase as Real Storage/Database for the application
- Android Studio Development Kit as application developer platform
- Adobe XD for Design Prototype for the application
- StarUML as Activity Diagram of the System

- Hardware: Smartphone with android 5.0 and above, at least 3GB RAM and 200MB of reserved storage, personal computer at least 8GB of RAM and 20GB of storage for development storage.
- Draw.io to create flowchart diagrams for project flow and basic system flow.

In a summary for my listing above, the main development tool required is Android Studio with connection to Google Firebase. Adobe XD, Draw.io and StarUML are important as some essential tools to help on planning and prototyping stages. Then in hardware aspect, the device used must be at least in minimum requirement to avoid instability.

3.3 System Design and Workflow Diagram

System design and workflow diagram is the earlier stage of development process. They are playing roles to give leading direction for developers and designer on how to work out everything in sequential form, to get as close as what Gantt Chart timeline expected for each section. Besides, they are important so developers can determine what they need to do and complete on every stage, a mistake in between might affect the whole development process

The listed figures below are all the system designed diagram that shows a table containing the whole development flow of this research project as well as the initial system design flowchart, followed by user’s Activity Diagram and its Use-Case Specification.

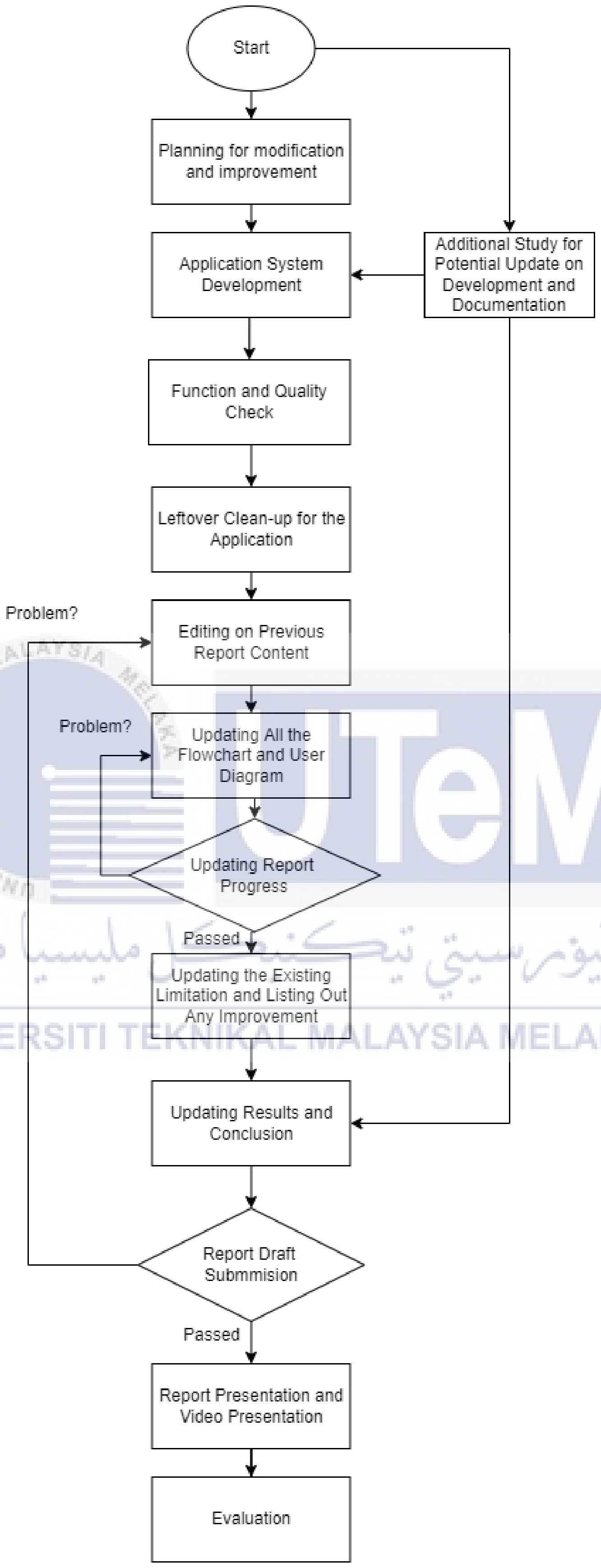

Start
Planning for modification and improvement
Application System Development
Additional Study for Potential Update on Development and Documentation
Function and Quality Check
Leftover Clean-up for the Application
Problem?
Editing on Previous Report Content
Problem?
Updating All the Flowchart and User Diagram
Updating Report Progress
Passed
Updating the Existing Limitation and Listing Out Any Improvement
Updating Results and Conclusion
Report Draft Submmision
Passed
Report Presentation and Video Presentation
Evaluation

Figure 11: Project Development Flowchart

Refer to Figure 11, it is a flowchart for the whole research project documentation stages. When it reaches problem such as errors on documenting each chapter, it will be revert back to the previous stages so correction could be make. A background study for the research purpose is also important so adequate amount of knowledge could be contribute to the research and completes the production of the application that meet the research objectives and project's scopes.

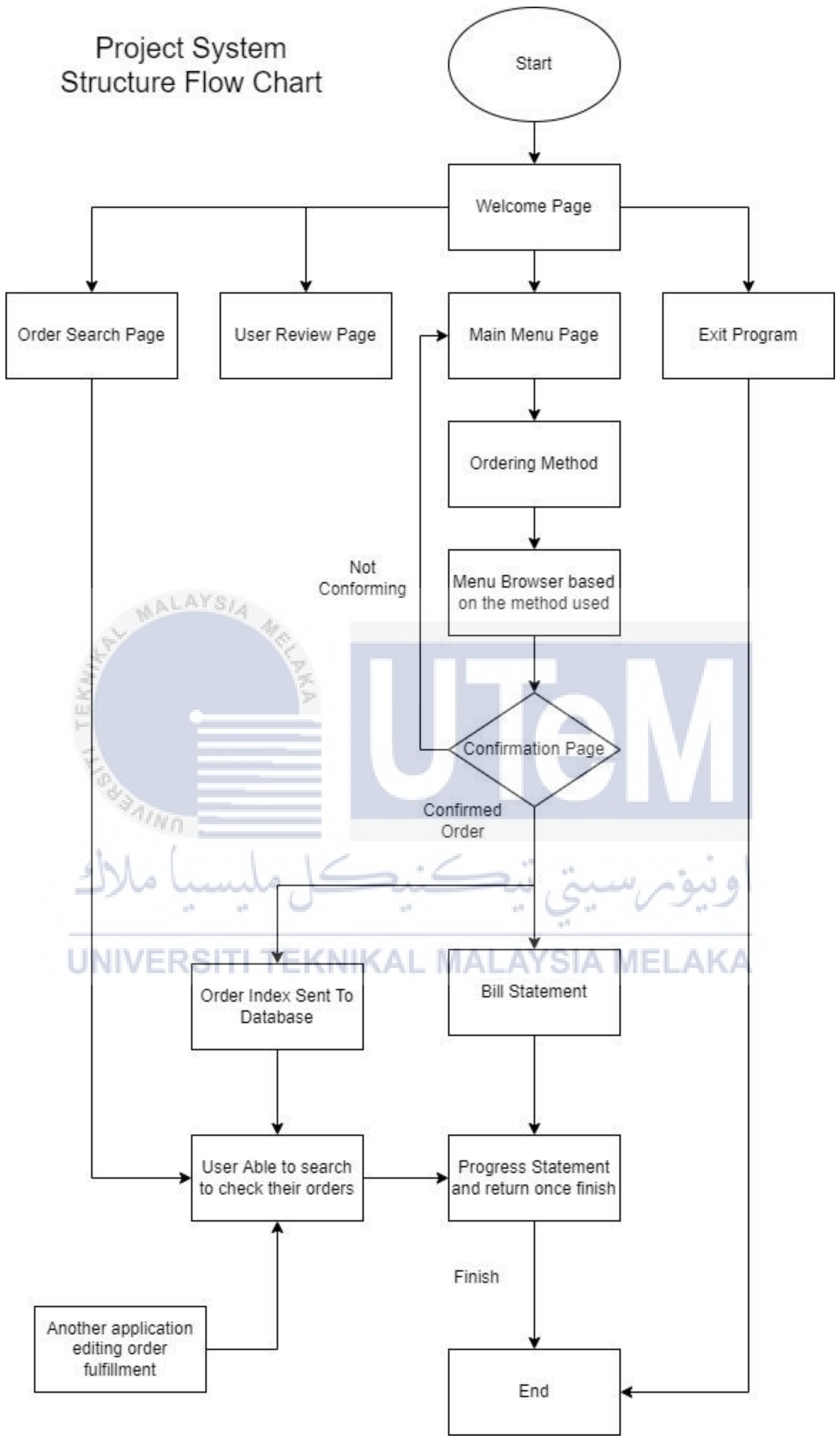


Figure 12: System Design Flowchart for Project Application System

Figure 12 is an flowchart model that contains system design of the project application. This flowchart is an expected sequence on how the application works. Starting from reaching a menu page, then ordering method, following by confirmation and bill, then order updates and searches with databases and application supported. Then there is also another optional site for user review and feedback about the application.

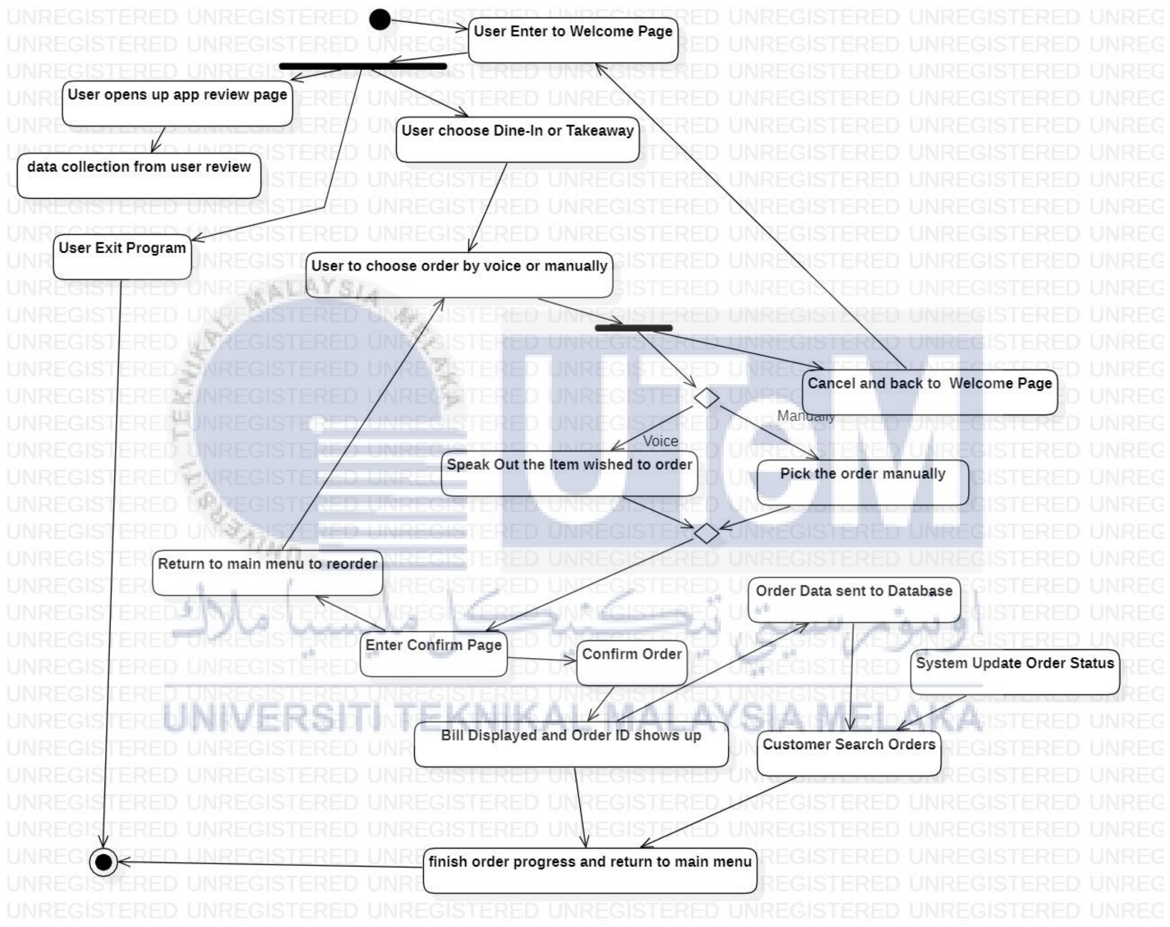

User Enter to Welcome Page
User opens up app review page
User choose Dine-In or Takeaway
data collection from user review
User Exit Program
User to choose order by voice or manually
Cancel and back to Welcome Page
Manually
Voice
Speak Out the Item wished to order
Pick the order manually
Return to main menu to reorder
Order Data sent to Database
Enter Confirm Page
Confirm Order
System Update Order Status
Bill Displayed and Order ID shows up
Customer Search Orders
finish order progress and return to main menu

Figure 13: User's Activity Diagram for the Application

Use-case diagram is important as a display for expected stages on how user interact with the system or application. So customers or users started the application and choose to order for takeaway or dine-in, with manual or express voice-ordering method. Then users can proceed to pick what they wanted with any amount if they used manual ordering, or just voice out anything they wanted with a maximum of 2 items with only 1 for each quantity.

Then confirmation page is out and users have to confirm with the item and prices shown. Bill page will be proceed next with order number for customers to track their order progress whether it is complete or not in the order search function at the main menu. There will be another optional application used to update the order fulfillment based on the restaurant operation and order number.

3.4 Experimental Setup

3.4.1 Prototype of the Application

An application prototype is consist of a GUI design prototype, then followed by the coding of the application that programs out the design prototype as close as possible, without losing its main function as much as possible. The extra variable for the appearance of the GUI can be subject to change based on the restriction on the coding section such as layout, button arrangement, image selection as long it still carries out the data required and the function that can be accessible by the users.

3.4.1.1 Prototype Design

Prototype Designs are very important as they are the drafting for on how application should be looked like before they are created, in order to create the most efficient application, the user interface must be simple and clean, so user can easy to understand which button to use, which object to interact to execute a specific functions, and without taking very long time.

Based of Figure 14-16, the diagrams shows how the application should look like initially before it goes to production for the application. These designs are specifically made with a prototyping software named Adobe XD. The name of the application is just set randomly as the purpose of this research is to carry out the possibility of the functions.

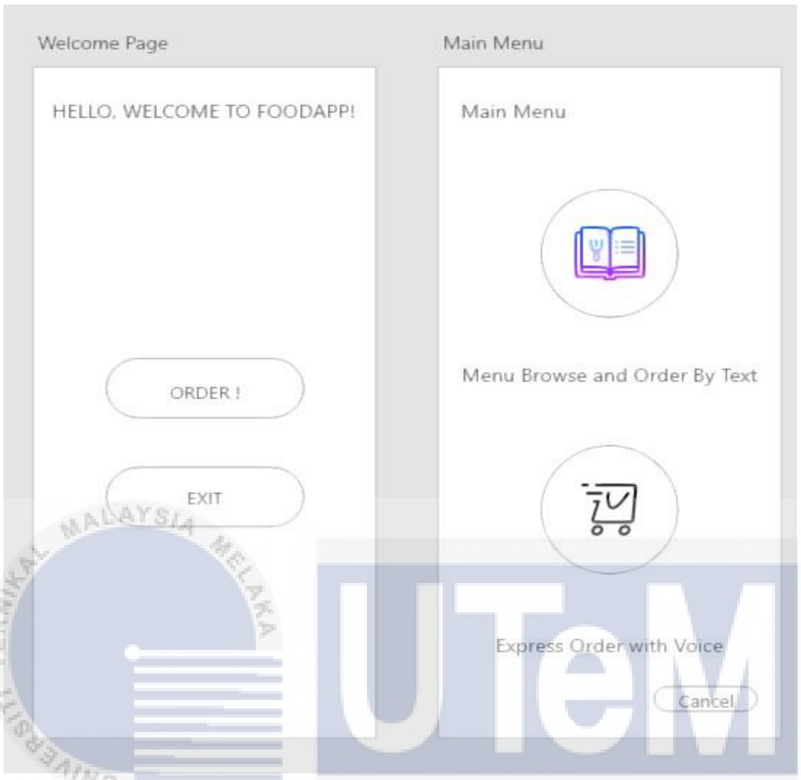


Figure 14: Welcome Page and Main Menu Page

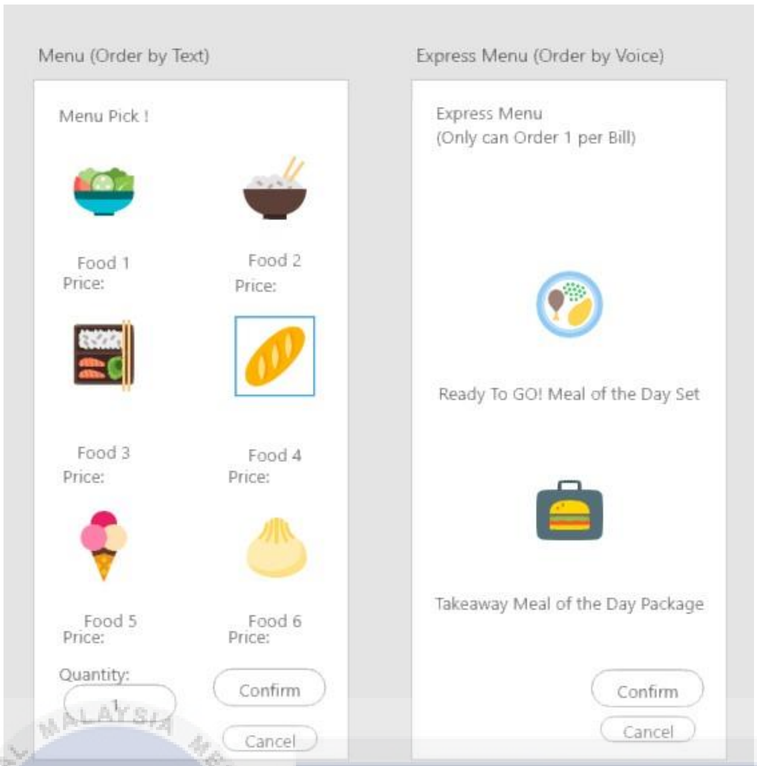


Figure 15: Manual Menu and Express Menu

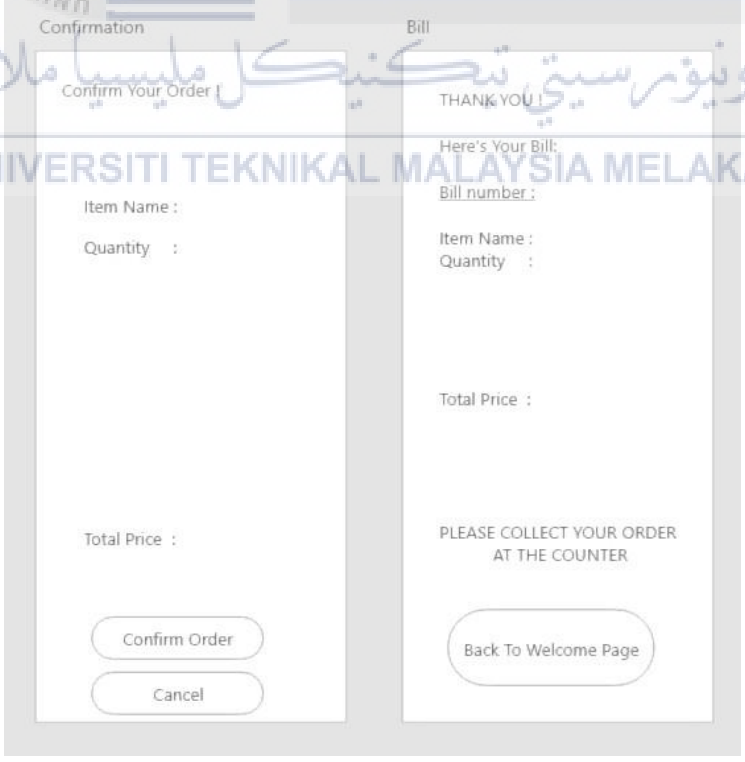
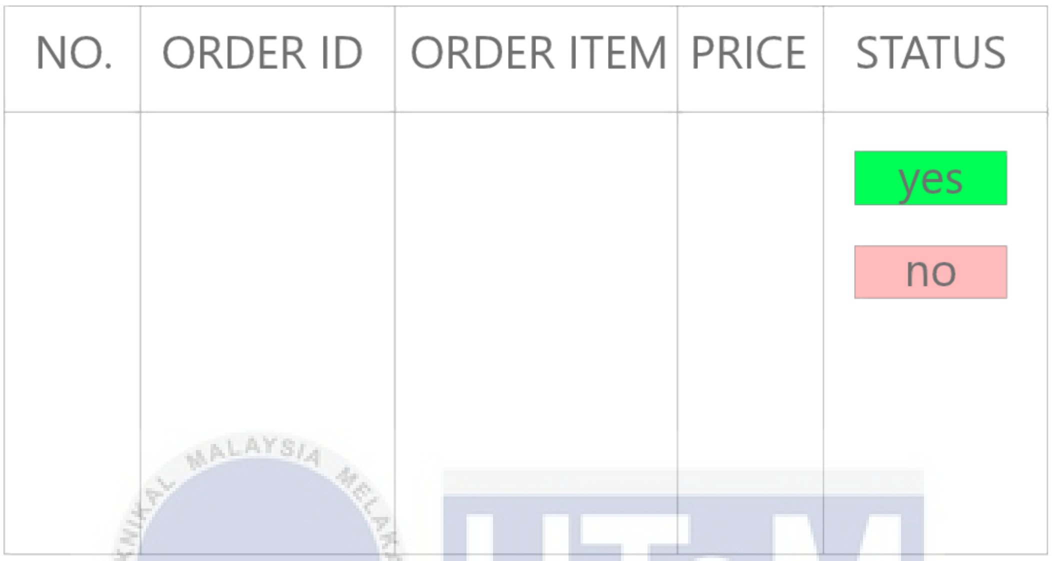


Figure 16: Confirmation Page and Bill Page

There is also a drafting of template design for how Firebase's real-time database should looked like with value stores into it and used to control the result and sent to the system for the users. It is shown in Figure 17.



NO.	ORDER ID	ORDER ITEM	PRICE	STATUS
				yes no

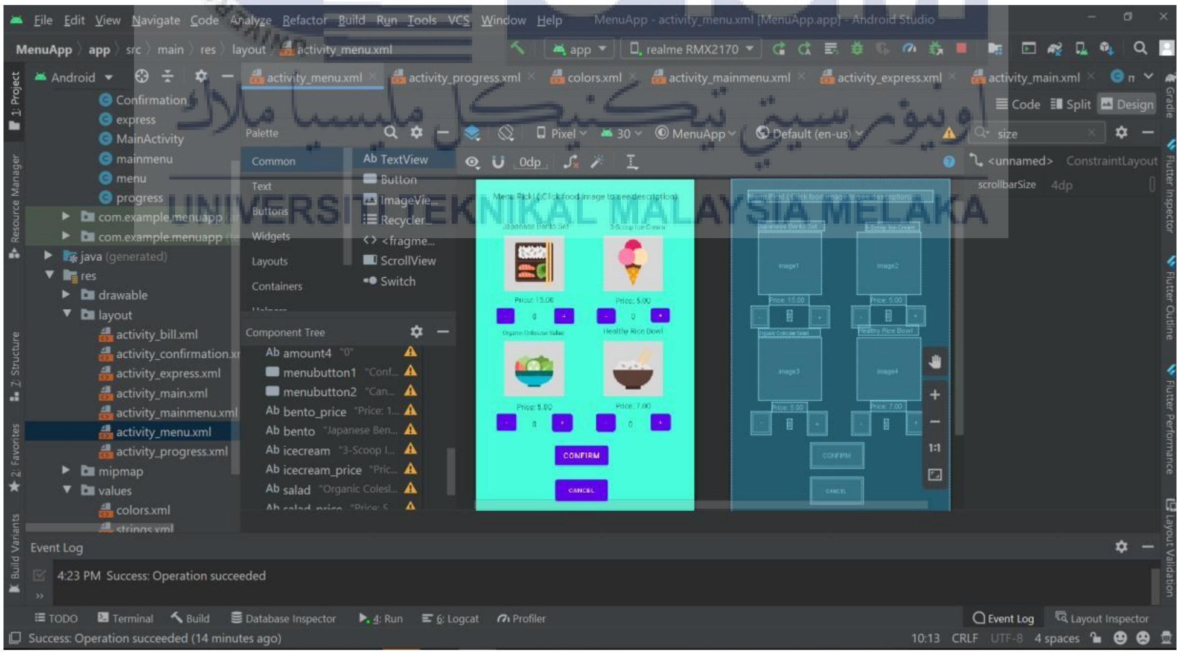
Figure 17: Initial Database Template

3.4.1.2 Actual Procedures of Application Development

This section will cover up examples of procedures and setting up all of the actual implementation of the prototype design, as prototype design might contains uncontrollable limitation that programming cannot be done, and hence have to be replaced, eliminated and substituted, as well as how the programming from Android Studio works to build up the design and the application.

To build every page shown from the prototype, firstly, Android Studio categorised the pages as Activity, each Activity can be processing specific task as programmed and store a lot of information, like text, images, GUI, widgets and so on. The layout design of the Activity will be handled with Extensible Markup Language (XML) and the functions inside can be programmed with JAVA language which is a default language used by Android Studio to program all the Activity. An additional Google Speech-to-Text (STT) function will be available for voice ordering method. These development progress are shown in Figure 18 (Other images can referred to Appendix A for the scenario of programming in Android Studio, the full coding will be on Appendix C).

Images below will be showing some examples of the programming progress, on how Android Studio's example programming looks like, the result of each Activity will be presented on Chapter 4's section. Refer to Figure 18, the leftmost section is the list of the coding files and folders as well as the resouces for the application, then the middle part is the editing section and the right most part is the parameter section. It is an example on how Android Studio's Activity Design Tab looks like.



4:23 PM Success: Operation succeeded

Success: Operation succeeded (14 minutes ago)

10:13 CRLF UTF-8 4 spaces

Figure 18: Example Scenario in Android Studio

3.5 Limitation and Summary

Every programming have its own limitation and flaw, as certain part of the functions, design and layout cannot be implemented and unable to debug, below are the list of findings that I found while developing this application that still need to be surpassed and fixed in the future

- Voice Recognition Method I used which is Speech-To-Text (STT) function for express ordering can only pickup one item order per usage.
- Every object's layout constraint depending on user's devices, different devices have its own dimension and resolution and thus it can't be done by now.
- STT function's accuracy is varied and not totally accurate, and users need to speak the specific term to record down the order. For example: User can only voice out specific term as matched to the system so the function can be working, elaboration and simplification of the term or phrases might not be able to invoke the function.
- Only one item is available to be ordered with STT function for now. Parsing multiple data values are heavily dependent to the possibility to program.
- There is no verification of the order input as it requires real payment gateway to verify the transaction is successful.

In the next part of this project research, the application has some improvements from the limitation of this research. First of all, voice-recognition function managed to enable customer to make 2 type of food ordered in one time. So for example, users can order item A and B for each order. Although the unit per item is stilled fix to single unit, however more variant of item can order now compared to only single item per time

Due to the ability to order two type of item per time, the application accept what is the combination of the items without any specific arrangement, such as user can order item A and item B by calling out either item A or item B first, without specific arrangement. This can be improved more based on the restaurant on what kind of configuration of their menu.

In summary for this whole chapter, all methodology used are personal preference and without any plagiarism. Every stage is conducted with extra cautious to prevent misinformation and miscalculation in between and affecting the result of each step. Additional steps are possible to add in or modify by further researching time to time. Every limitations exist are not permanent as they could be upgraded with future researches to minimise the flaws on the application and research.

CHAPTER 4

RESULT

4.1 Introduction

In this section, I will show all of the result obtained from the progress of my production of my application, and its result with provided parameters and requirement. A discussion will be also included in this section to share about possible thoughts and possible reasoning on how the result is created and its explanation.

4.2 Preliminary Result and Result Analysis

4.2.1 Application Completion and Performance Benchmarking Result

A preliminary result is made on the completion of the application will be presented here, as it covers every key function related to research topic. The menu page's GUI result from application will be shown in Figure 19 and Figure 20 (and the other application pages are on Appendix B section) based on a standard AndroidOS mobile phone to test out all functionalities.

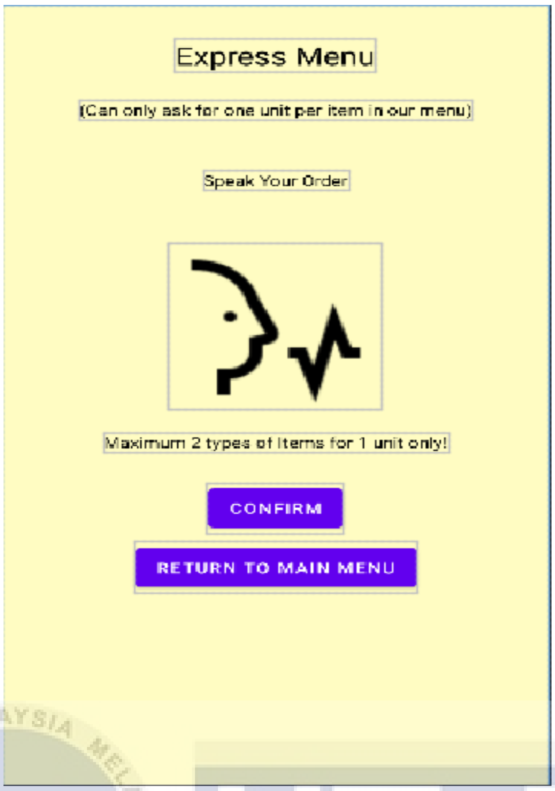

Express Menu
(Can only ask for one unit per item in our menu)
Speak Your Order
Maximum 2 types of Items for 1 unit only!
CONFIRM
RETURN TO MAIN MENU

Figure 19: Normal Manual Clicking Ordering Menu

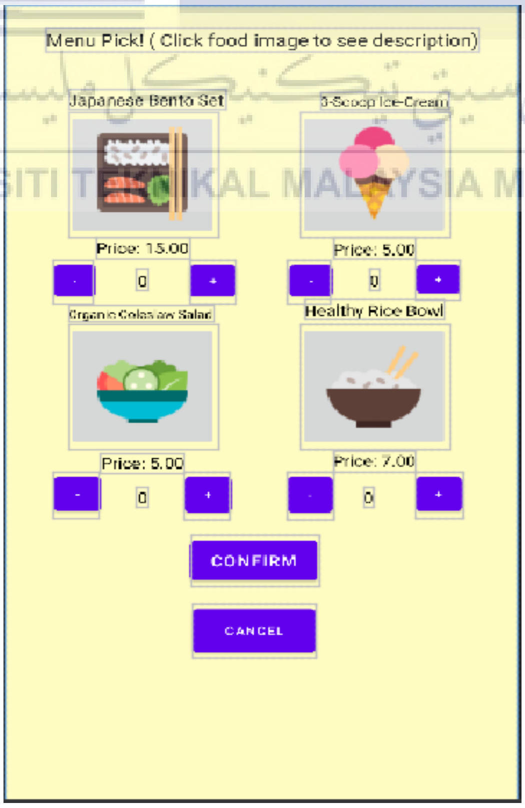


Figure 20: Express Menu Ordering with Voice

Referring to Figure 19, it is the menu page for manual ordering method, so customers have to manually click increment or decrement of the amount needed for the food/drink they wished to order. While on Figure 20, customers have to click the middle button to access the voice ordering (STT) function that detect their voices and translate it into text value to parse. The main difference between both method here is voice ordering only available to make 1 unit of order compared to manual ordering method where it is free to choose any amount.

An collection of data analysed based of [17], where similarly of unit p, stands for percentage, that it is a statistical measurement that is used to know the ratio of distribution and translating frequency counts for comparison. The result of the percentage will determine the importance of this measurement for this research on effectiveness of voice-ordering method. Hence, frequency is based on the accurate result, while total number will be based on number of trial. The data is obtain and recorded in Table 5, followed by its calculation based on the formula.

$$percentage, p = \frac{frequency\ (Accurate), f}{Number\ of\ Trial, n}\ x\ 100 \quad [17]$$

Table 5: Percentage Accuracy of the Voice Ordering Function

Number of Trial,N	Frequency(Accurate), f
1	Accurate
2	Accurate
3	Accurate
4	Not Accurate
5	Accurate

Number of Trial,N	Frequency(Accurate), f
6	Accurate
7	Accurate
8	Accurate
9	Not Accurate
10	Accurate
	$\text{Percentage Accuracy, P} = \frac{8}{10} x\ 100\%$ = 80%

Based on [17], there are also another technique that used to weigh the result of the measurement to every attempt for ordering using manual or voice-ordering method. All timings are recorded and a frequency of certain corresponding verbal interpretion is divded to number of trials in order to obtain the total mean of each ordering method, and compared out which one is more efficient on usage. After that, a weighted mean is obtain based on the calculation between time taken and number of trial to make an order. The only difference of this formula is the frequency for each measurement is only 1 as we can only obtain 1 answer at a time, hence the frequency is negligible.

$$Weighted\ Mean, \sum \bar{x} = \frac{Time\ Taken\ to\ Order, X}{Number\ of\ Trial, N} \quad [17]$$

Table 6: Weighted Mean Time Taken to Make an Order

Number of Trial,N	Time Taken to Order with Manual Ordering, X_M	Time Taken to Order with Voice-Ordering, Xv
1	7	5
2	11	6
3	8	4
4	13	4
5	7	6

Number of Trial,N	Time Taken to Order with Manual Ordering, X_M	Time Taken to Order with Voice-Ordering, Xv
6	10	5
7	5	5
8	15	4
9	9	4
10	10	4
	$Mean\ time, \bar{x} = \frac{95}{10}$ =9.5 seconds	$Mean\ time, \bar{x} = \frac{47}{10}$ =4.7 seconds

A graphical figure is represented as comparison between both average time here in Figure 21 which y-axis represents the time,t and x-axis represents the type of ordering method.

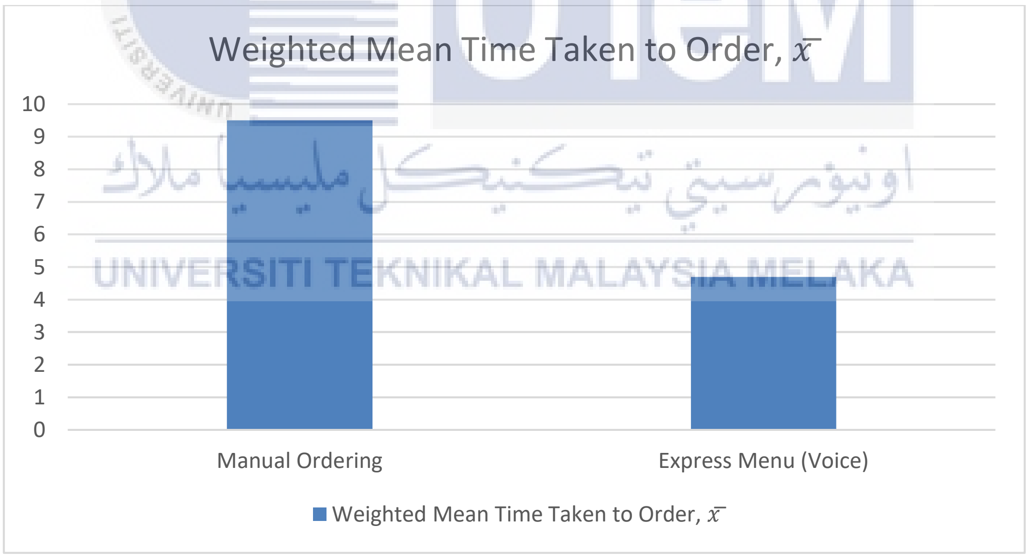


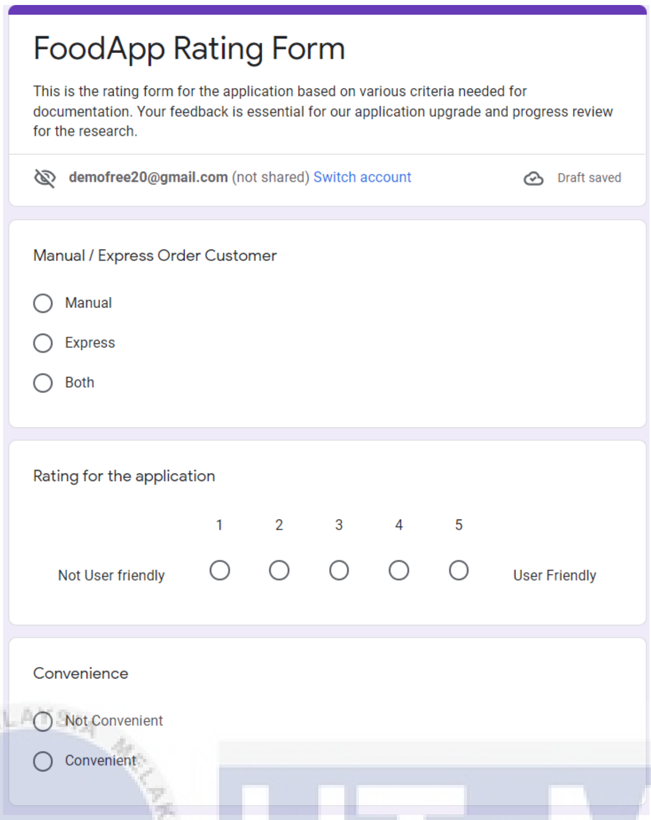
Figure 21: Weighted Mean Time Taken to Order based on Method Chosen.

Referring to Table 5, we can see the accuracy of voice-recognition function could achieve good results which score 80% out of 100, it shows that using voice ordering is muchly possible to make order by just speaking to the device without relying on clicking repetitive menu. The data on Table 6 also shows how much time taken to make order with each respective method and resulted in using voice-recognition method can reduce time taken roughly half of the time taken by normal manual method as shown in Figure 21. The result may be affected by several reasoning based on the users like hesitation of picking item or remaking order due to error input.

4.2.2 User Feedback Review Result

User feedback is one of the essential and high priority result that is directly reflects to the application. Although the result might be purely subjective because it is based on users' preferences, however with a limited range of answers, a rough estimation of result could be presented and giving more transparencies on the direction of the research.

In the application, a feedback form is presented for the users or customers in this research setting, to give their opinions and review about the application. The data are collected with Google Form with various type of questions that related to our objectives and some simple efficiency result.


FoodApp Rating Form

This is the rating form for the application based on various criteria needed for documentation. Your feedback is essential for our application upgrade and progress review for the research.

demofree20@gmail.com (not shared) Switch account

Draft saved

Manual / Express Order Customer

◯ Manual

◯ Express

◯ Both

Rating for the application

	1	2	3	4	5	
Not User friendly	◯	◯	◯	◯	◯	User Friendly

Convenience

◯ Not Convenient

◯ Convenient

Figure 22: FoodApp Rating Form Made With Google Form.

Based on Figure 22, all the questionaires in the feedback and review form are used to have a quantitative result of benchmarking and responses for this application system performance based on their criteria that are aimed to fulfill all objectives' goals. Technically, in this form the main aspect regarding this research, such as application rating as overall, convenience, time saving and accessibility are set here as part of the questions to get how every users, which are customers in a restaurant would review for improvement of their experience when used in the restaurant. All of the questions are giving positive responses from all the respondees and the detailed results are available in Appendix C. As a result, the application system proves that it has improvement in terms of time saving, better accessibility, more convenient and also the application receives high ratings that proved its successfulness.

4.2.3 Case Study Result

In this section, a case study result is carried to prove a qualitative result on how is the application affecting the whole restaurant operations and whether the research materials are sufficient or efficient enough to prove that they are giving good feedback for the restaurant operation. The case study is taken place on a real operating restaurant that has high traffic on the customer flows and having very good performance in terms of operation. An interview with one of the restaurant operator is carried out as an important source of information for reviewing, and providing feedback for the application system whether any valuable comments that could be obtained.

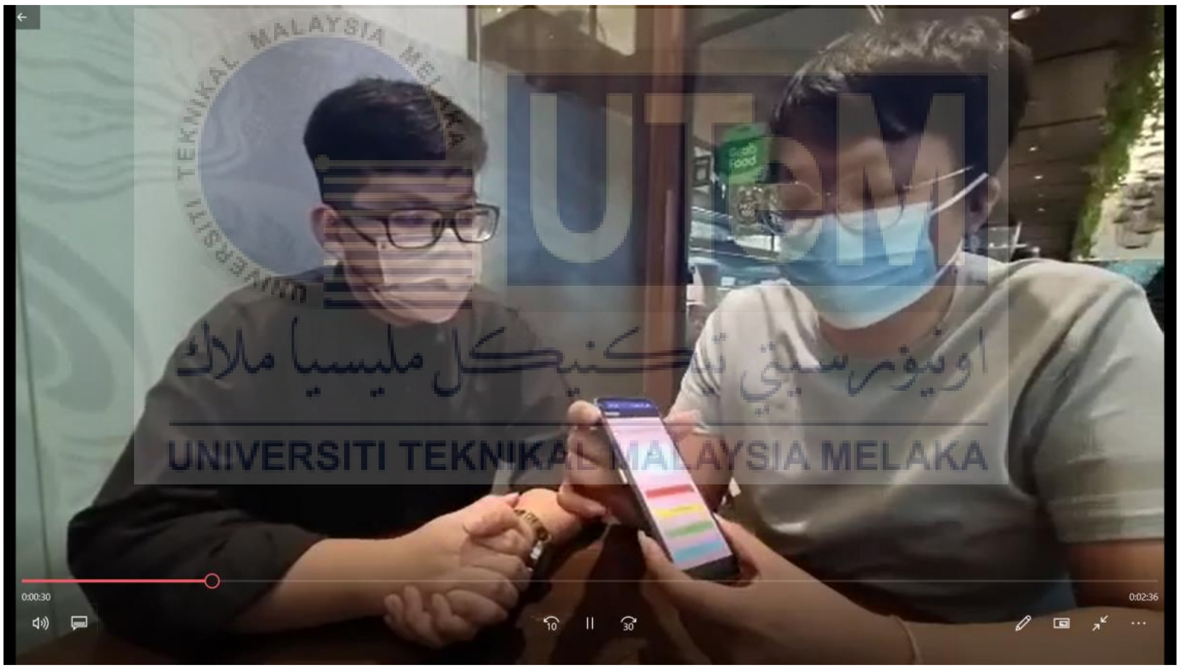


Figure 23: Case Study Interview with a restaurant in Kuala Lumpur.

Figure 23 is an image from the interview which is between researcher and the target interviewee which is an operator member from a famous restaurant which is located in Kuala Lumpur. The interviewee provided a lot of useful review and feedback based on their current existing system, where a good quality of responses can obtained from them to see whether it reaches quality expected for restaurant nowadays.

Based on the result, the restaurant operator member rated 4 out of 5 for the overall ratings, he is also think that this application save time for most of the customers, and also easily available and accessible by any category of customers.

4.3 Expected Result After Completion

The expected result is matching to all criteria in the research objectives, which includes an Android-based application which is a menu application for restaurant. The application is capable for customers to use traditional manual or voice-recognition method to make a basic order for their meal. Customer can confirm and view their bill followed by searching for the bill progress based on their order number stored into the real-time database. Another application is used to support the main application that are specifically for restaurant workers or operators to fulfill their order status and complete the order so customer can get to know when it is done.

Like the analysis available from preliminary result, the same benchmark measurement is done again to compare any consistent result based on improvement of the application system that overcame certain limitations previously and modified. Hence, the accuracy test and time taken test from Table 5 and Table 6 will be measured again for the performance check. Then two comparison charts will be presented to see their difference in result.

Table 7: Percentage Accuracy of the Voice Ordering Function for Completed Application

Number of Trial,N	Frequency(Accurate), f
1	Not Accurate

Number of Trial,N	Frequency(Accurate), f
2	Accurate
3	Accurate
4	Accurate
5	Not Accurate
6	Not Accurate
7	Accurate
8	Accurate
9	Accurate
10	Accurate
	$\text{Percentage Accuracy, P} = \frac{7}{10} x\ 100\%$ = 70%

Table 8: Weighted Mean Time Taken to Make an Order for Completed Application

Number of Trial,N	Time Taken to Order with Manual Ordering, X_M	Time Taken to Order with Voice-Ordering, Xv
1	13	7
2	15	5
3	14	9
4	9	9
5	9	5
6	10	8
7	11	7
8	15	6
9	8	10
10	10	11
Number of Trial,N	Time Taken to Order with Manual Ordering, X_M	Time Taken to Order with Voice-Ordering, Xv
	$Mean\ time, \bar{x} = \frac{114}{10}$ =11.4 seconds	$Mean\ time, \bar{x} = \frac{77}{10}$ =7.7 seconds

As the result of Table 8, the percentage for the accuracy result is decreased by a little, this is because updated version of the voice-ordering system, where users can make more than 1 item per order, hence when more items and criteria included, the result tend to be more accurate and fortunately it still stays in a positive result which is more than half of the trials are giving accurate responses.

The weighted mean for time taken on each ordering method in table 9 is a completed and more accurate version compared to Table 6 which is a preliminary result, as the application system is now added more functions and reduced more limitations. When more features and functions are added on, the mean time taken for both ordering method increases as users can now have more options for ordering and more detail information to look for, such as order checking and multiple items ordering.

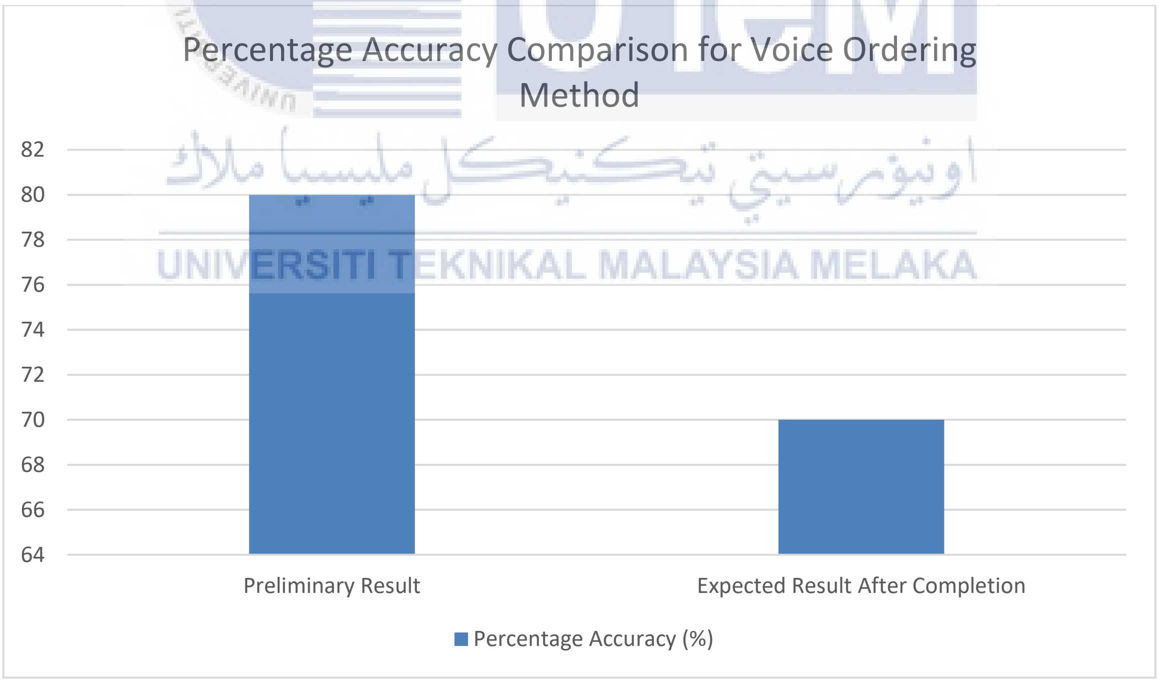


Figure 24: Comparison of Percentage Accuracy Voice Ordering Method Between Preliminary Result and Expected Result

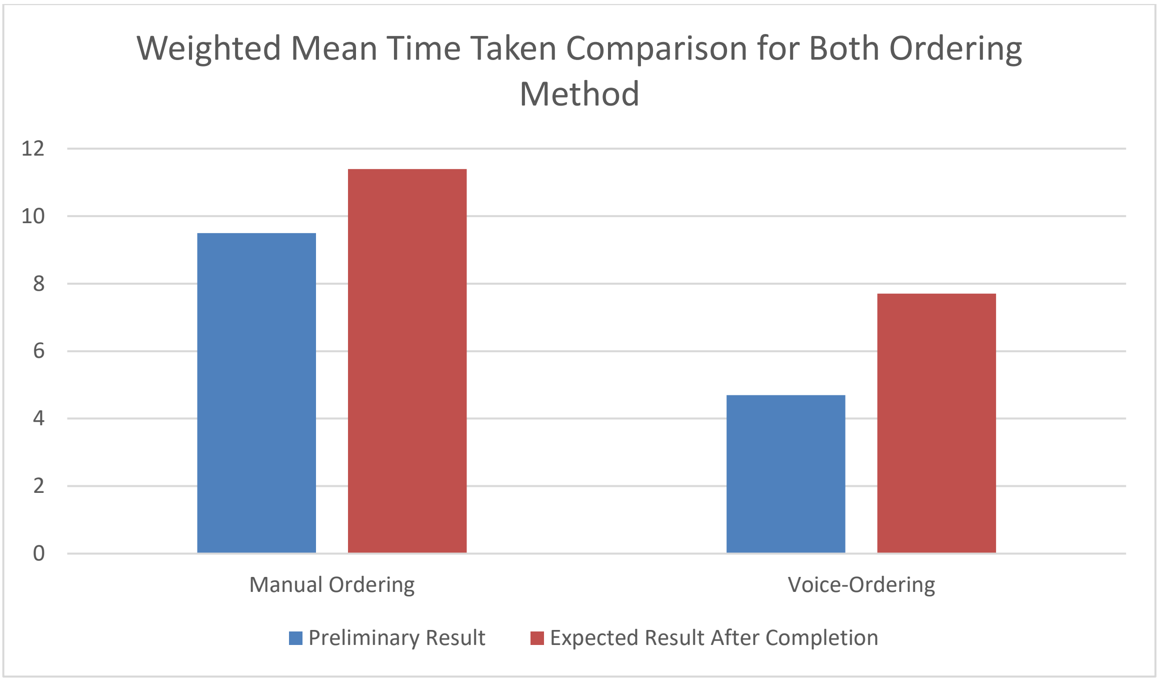


Figure 25: Comparison of Weighted Mean Time Taken to Order Between Preliminary Result and Expected Result

Figure 24 and Figure 25 are the comparisons for same data analysis for preliminary and expected result. In Figure 24, the percentage accuracy for voice-ordering method is lower on expected result compared to preliminary result as it has upgraded to enable itself for ordering two types of items instead of a single item per time. While for Figure 25, the weighted mean time taken for both ordering method shows higher side on the expected result after the application is completed compared to preliminary result which has the application that lack of search function and database system that users or customers have lesser functions and feature to use the application. Overall, expected result has 10% difference in consistency on percentage accuracy on voice-ordering, and 1.9 seconds difference for weighted mean on time taken for order by manual method, and 3 seconds difference for weighted mean on time taken for order by voice-ordering method.

4.4 Summary

In summary, voice-recognition technology has proved its value of improvement on how effective to save customer's time to make ordering in a restaurant, however the result is not on absolute value such as the time taken where only using stopwatch to measure the timing and they will be depending on customers' preference and way to make their orders with the help of application. External factors might be also affecting the result like users' hardware performance that would affect the overall result, hence an average is taken to reduce the bias of error on accuracy and time taken. Qualitative result like having a live-interview with restaurant operator in case study for the application and quantitive result like having a feedback form for any users are also available to collect any subjective but majority result since application feedback from a lot of users are important too.

CHAPTER 5
CONCLUSION

5.1 Conclusion

In conclusion of this research report, the demand of improving restaurant operation with technology is still very high in current society. Previous researches by researchers around the world had provide a lot of beneficial information about this research topic. They implemented different kind of technology into restaurant’s menu ordering system that provide different improvements. The advancement of voice-recognition method is the key information to contribute on this research topic as the main objective is to upgrading current system to a new level that is still highly demand and might not be visible yet inside the restaurant operation.

The methodology is effective and robust to obtain the main result for this research objective to show its possibility to use voice ordering on restaurant menu. The preliminary result shown also once again proved how useful voice ordering can be inside a normal restaurant menu system. The accuracy of voice-recognition method is shown a good sign that it is usable for restaurant operation and the time taken for voice ordering has giving advantages to the targeted users which customers that are in hurry and doesn’t want to go through the whole menu to make order. The result given is also utilised by simple and reasonable data input and calculation formula so future research can use them as reference more easily.

The limitation stated inside the methodology section will be also giving ideas for future research on what can be fixed and improved in further way. Most of the limitation listed could have been solve in future works based on the improvement of the technology used, so it created a new possibility to add-on or improve this current result to more solid system that benefit restaurants or any other society field that have menu ordering system.

5.2 Future Works

The research and application will continue its step to implement more useful function available in current trend of restaurant application, as well as more minimisation of the limitation of this research listed, as well as continue up to improve any available application to more stable and better result for futureproof requirements.

Since nowadays users can search for any application they required online, therefore this research is working towards more possibilities to intergrate with other available restaurant system that is hoped to have more advance and usable application for restaurant operation in the future.

REFERENCES

[1] W. S. Jiuan, “Restaurant Ordering System,” 2019.

[2] R. Paul, R. R. Bhandopia, D. R. Patil, J. D. Raut, and A. D. Gotmare, “An Efficient Digital Ordering System for Restaurant,” *Int. J. Res. Eng. Sci. Manag.*, vol. 3, no. 3, pp. 391–395, 2020.

[3] R. Aulia, A. Zakir, H. Dafitri, D. Siregar, and H. Hasdiana, “Mechanism of Food Ordering in A Restaurant Using Android Technology,” *J. Phys. Conf. Ser.*, vol. 930, no. 1, 2017, doi: 10.1088/1742-6596/930/1/012030.

[4] H. Feng, K. Fawaz, and K. G. Shin, “Continuous authentication for voice assistants,” *Proc. Annu. Int. Conf. Mob. Comput. Networking, MOBICOM*, vol. Part F1312, pp. 343–355, 2017, doi: 10.1145/3117811.3117823.

[5] M. B. Hoy, “Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants,” *Med. Ref. Serv. Q.*, vol. 37, no. 1, pp. 81–88, 2018, doi: 10.1080/02763869.2018.1404391.

[6] K. Berezina, O. Ciftci, and C. Cobanoglu, “Robots, artifiial intelligence, and service automation in restaurants,” in *Robots, Artificial Intelligence and Service Automation in Travel, Tourism and Hospitality*, 2019, pp. 185–219.

[7] N. K. Manaswi, “Speech to Text and Vice Versa,” in *Deep Learning with Applications Using Python*, 2018, pp. 127–144.

[8] B. Canziani and S. MacSween, “Consumer acceptance of voice-activated smart home devices for product information seeking and online ordering,” *Comput. Human Behav.*, vol. 119, no. January 2020, p. 106714, 2021, doi: 10.1016/j.chb.2021.106714.

[9] L. Warlina and S. M. Noersidik, "Designing Web-based Food Ordering Information System in Restaurant," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 407, no. 1, 2018, doi: 10.1088/1757-899X/407/1/012029.

[10] K. Chochiang, P. Ung, and N. Bunsaman, "One Stop Restaurant Service Application," *17th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol. ECTI-CON 2020*, pp. 482–485, 2020, doi: 10.1109/ECTI-CON49241.2020.9158070.

[11] L. W. HONG, "Food Ordering System Using Mobile Phone," 2016.

[12] H.L.V.G.Wasuladaththa and P. N. M. , E.W.M.W.W.A.S.K.Ekanayake, I.P.Hiranthi Yashodha Premasiri, "SMART RESTAURANT MANAGEMENT AND ORDERING SYSTEM: A CASE STUDY," no. April, 2018.

[13] V. Tarus, "ORDER-NOW SYSTEM CENTRIA," no. December, 2020.

[14] S. Dasgupta, M. Samaddar, C. N. Yogalakshmi, K. Vijayan, and Subhiksha, "Autonomous Restaurant Serving System using Image Detection and Voiceprint Detection in Android Application," *Proc. 2020 IEEE Int. Conf. Commun. Signal Process. ICCSP 2020*, pp. 946–951, 2020, doi: 10.1109/ICCSP48568.2020.9182314.

[15] M. Hajian, *Deploying to Firebase as the Back End*. 2019.

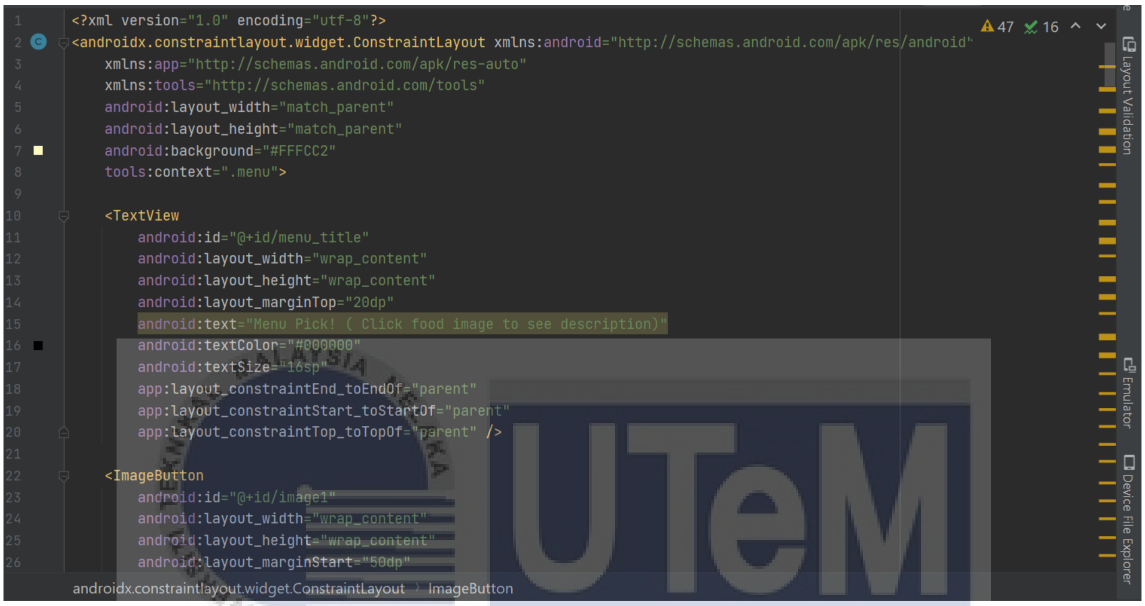
[16] L. Moroney, *The Definitive Guide to Firebase*. 2017.

[17] Molejon, Montaigne G. and Comendador, Benilda Eleonor V., "Augmented Reality Ordering System With Data Analytics To Support Decision Making In The Restaurant Chains", Innovatus, vol. 3, no. 1, pp. 18–28, May 2020, doi: 10.5281/zenodo.5173069.

APPENDICES

Appendix A

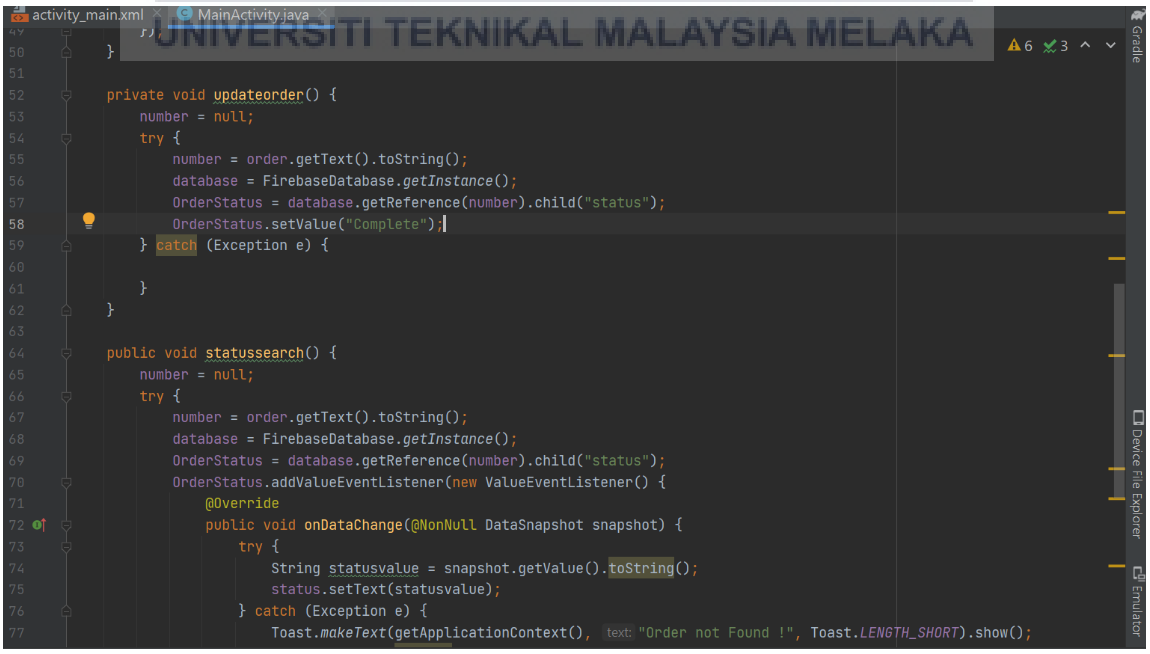
Example Scenario of Android Studio with Structure of XML, JAVA programming and Listing



```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFCC2"
    tools:context=".menu">

    <TextView
        android:id="@+id/menu_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Menu Pick! ( Click food image to see description)"
        android:textColor="#000000"
        android:textSize="16sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageButton
        android:id="@+id/image1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
```

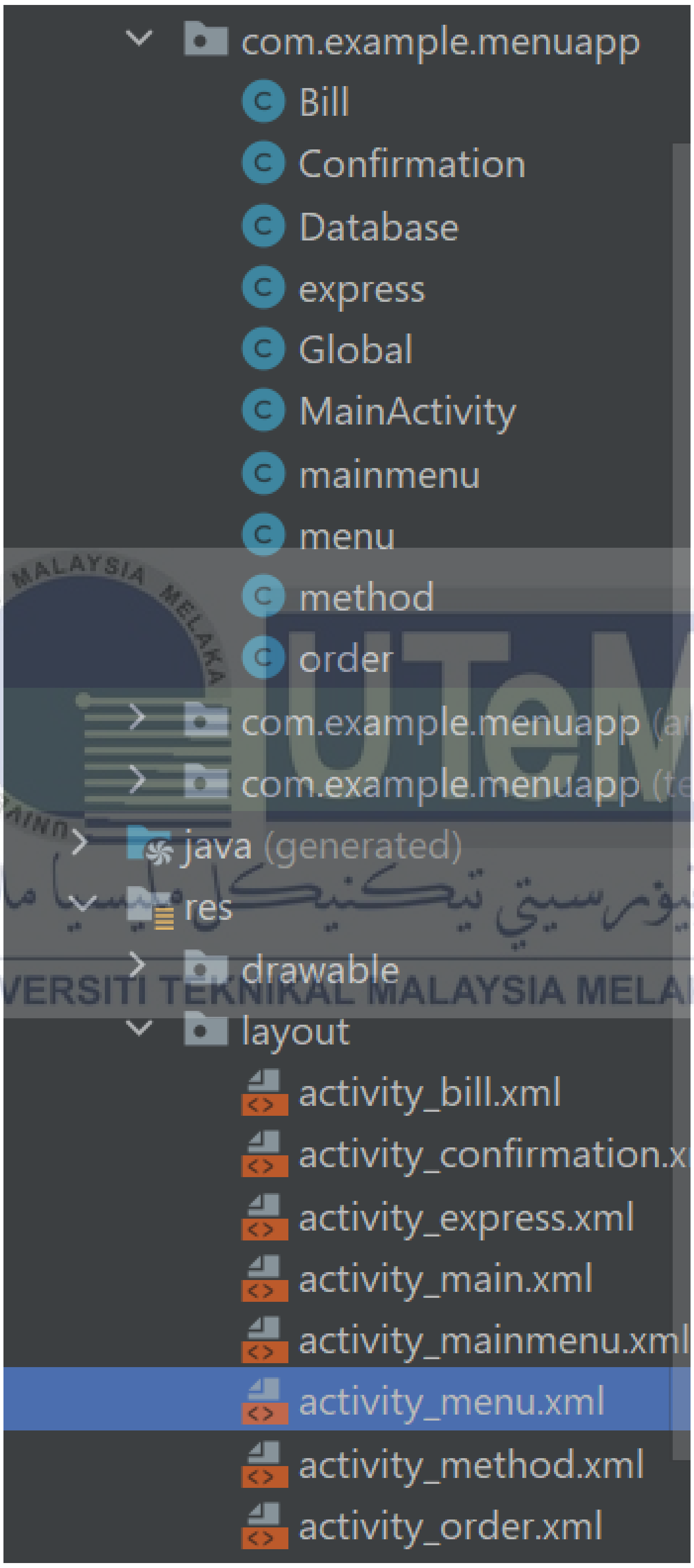
47 16
Layout Validation
Emulator
Device File Explorer
androidx.constraintlayout.widget.ConstraintLayout > ImageButton

XML Coding Scenario

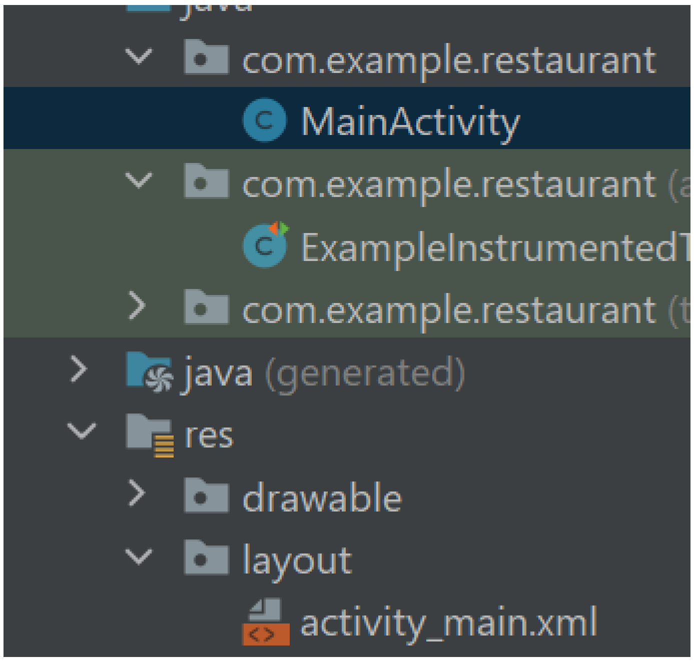


```java
        });
    }

    private void updateorder() {
        number = null;
        try {
            number = order.getText().toString();
            database = FirebaseDatabase.getInstance();
            OrderStatus = database.getReference(number).child("status");
            OrderStatus.setValue("Complete");
        } catch (Exception e) {

        }
    }

    public void statussearch() {
        number = null;
        try {
            number = order.getText().toString();
            database = FirebaseDatabase.getInstance();
            OrderStatus = database.getReference(number).child("status");
            OrderStatus.addValueEventListener(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    try {
                        String statusvalue = snapshot.getValue().toString();
                        status.setText(statusvalue);
                    } catch (Exception e) {
                        Toast.makeText(getApplicationContext(),  text: "Order not Found !", Toast.LENGTH_SHORT).show();
```

Java Coding Scenario



```
⌄ com.example.menuapp
    Bill
    Confirmation
    Database
    express
    Global
    MainActivity
    mainmenu
    menu
    method
    order
> com.example.menuapp (a
> com.example.menuapp (te
> java (generated)
⌄ res
  > drawable
  ⌄ layout
      activity_bill.xml
      activity_confirmation.x
      activity_express.xml
      activity_main.xml
      activity_mainmenu.xml
      activity_menu.xml
      activity_method.xml
      activity_order.xml
```

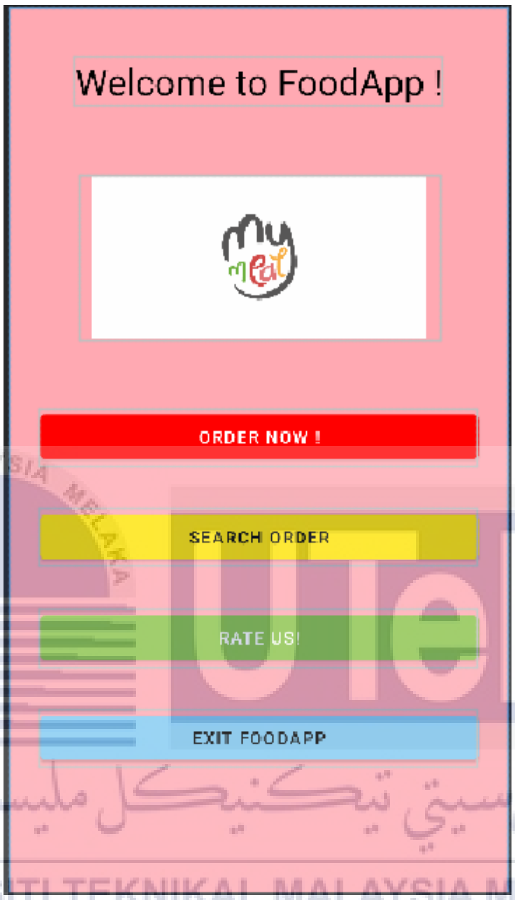


```
java
  com.example.restaurant
    MainActivity
  com.example.restaurant (a
    ExampleInstrumentedT
  com.example.restaurant (t
java (generated)
res
  drawable
  layout
    activity_main.xml
```

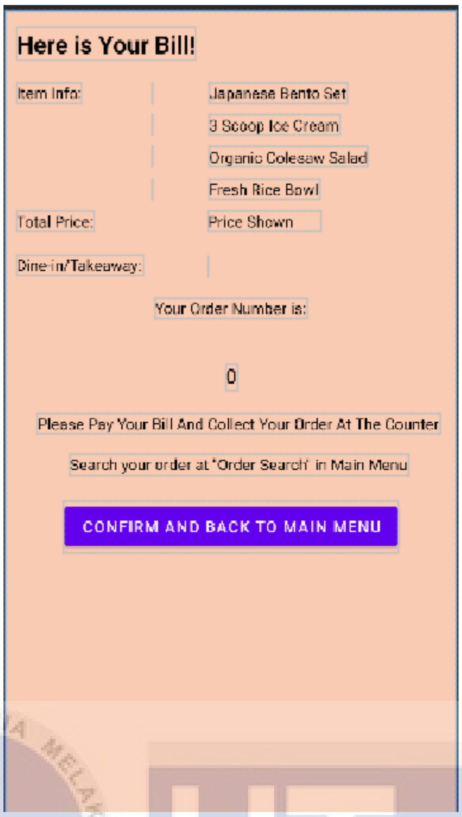
List of XML and JAVA Code Pages

Appendix B

Application GUI Result on Android Devices



Welcome Page


Here is Your Bill!

Item Info: Japanese Bento Set

3 Scoop Ice Cream

Organic Coleslaw Salad

Fresh Rice Bowl

Total Price: Price Shown

Dine-in/Takeaway:

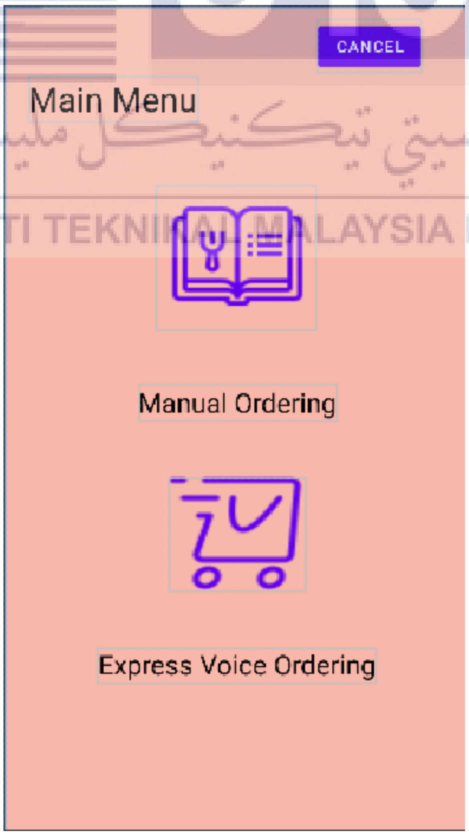
Your Order Number is:

0

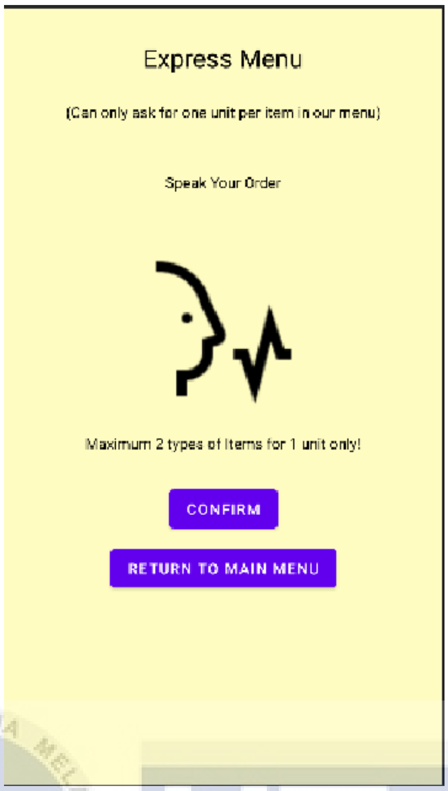
Please Pay Your Bill And Collect Your Order At The Counter

Search your order at 'Order Search' in Main Menu

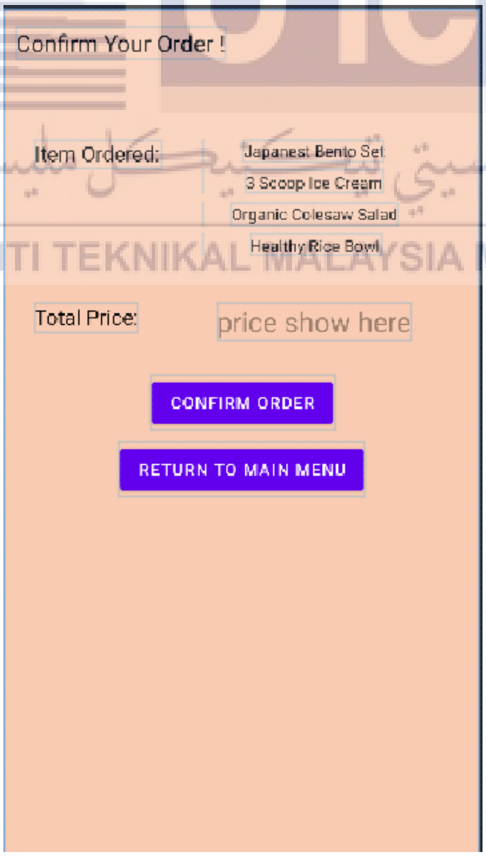
Billing Page



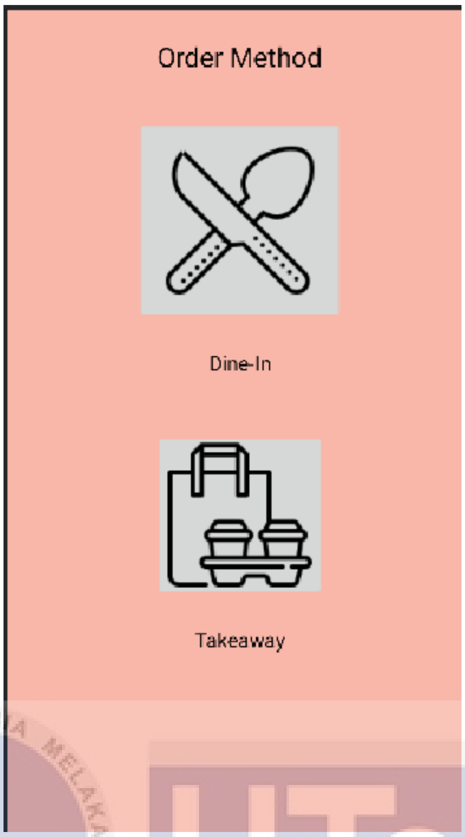
Main Menu (Ordering Method Selection)


Express Menu
(Can only ask for one unit per item in our menu)
Speak Your Order
Maximum 2 types of Items for 1 unit only!
CONFIRM
RETURN TO MAIN MENU

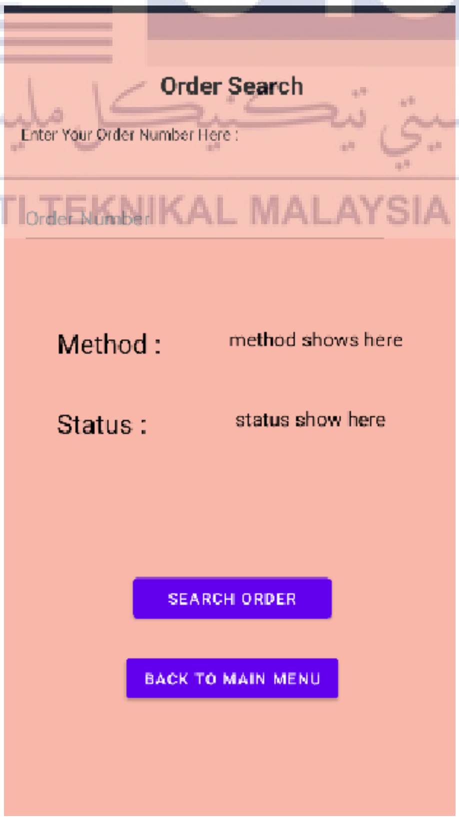
Google Speech-to-Text Function



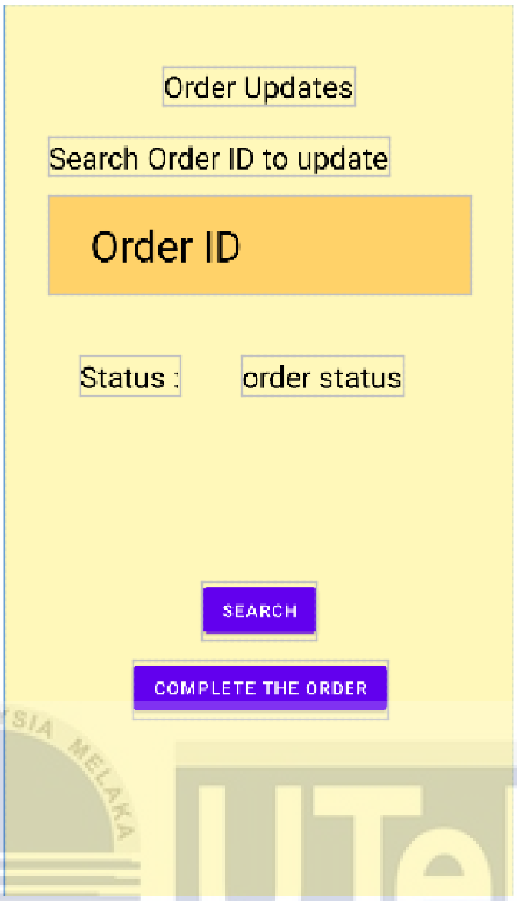
Order Confirmation Page



Takeaway or Dine-in Selection



Order Search



Order Update Page

Appendix C

Programming Code for Every Page of the Application

1. Welcome Page

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFA9B2"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/welcome_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
        android:layout_marginTop="40dp"
        android:layout_marginEnd="30dp"
        android:text="Welcome to FoodApp !"
        android:textColor="#000000"
        android:textSize="30sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/welcome_button"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="25dp"
        android:layout_marginTop="250dp"
        android:layout_marginEnd="25dp"
        android:backgroundTint="#FF0000"
        android:text="Order Now !"
        android:textColorHighlight="#FFFFFF"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/welcome_text" />

    <ImageView
        android:id="@+id/welcome_logo"
        android:layout_width="298dp"
        android:layout_height="136dp"
```

```
app:layout_constraintBottom_toTopOf="@+id/welcome_button"
        app:layout_constraintEnd_toEndOf="@+id/welcome_button"

app:layout_constraintStart_toStartOf="@+id/welcome_button"

app:layout_constraintTop_toBottomOf="@+id/welcome_text"
        app:srcCompat="@drawable/welcome" />

    <Button
        android:id="@+id/main_exit"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="25dp"
        android:layout_marginTop="35dp"
        android:layout_marginEnd="25dp"
        android:backgroundTint="#82CAF2"
        android:text="Exit FoodApp"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/rating" />

    <Button
        android:id="@+id/check_button"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="25dp"
        android:layout_marginTop="35dp"
        android:layout_marginEnd="25dp"
        android:backgroundTint="#FEE405"
        android:text="SEARCH ORDER"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/welcome_button" />

    <Button
        android:id="@+id/rating"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="35dp"
        android:backgroundTint="#8BC34A"
        android:text="Rate Us!"
        app:layout_constraintEnd_toEndOf="@+id/check_button"

app:layout_constraintStart_toStartOf="@+id/check_button"

app:layout_constraintTop_toBottomOf="@+id/check_button" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Java Coding:

```
package com.example.menuapp;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;


public class MainActivity extends AppCompatActivity {

        private Button
welcome_button,check_button,exit,rating;

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

        welcome_button = (Button)
findViewById(R.id.welcome_button);

        welcome_button.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openMainmenu();
            }
        });
        check_button = (Button)
findViewById(R.id.check_button);
        check_button.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openCheck();
            }
        });
        rating = (Button) findViewById(R.id.rating);
        rating.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

gotoUrl("https://forms.gle/HwWdkEpAvNxHmmNQ8");
            }
        });
        exit = (Button) findViewById(R.id.main_exit);
        exit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                moveTaskToBack(true);

android.os.Process.killProcess(android.os.Process.myPid());
                System.exit(1);
            }
        });


    }
    private void gotoUrl(String s) {
        Uri uri = Uri.parse(s);
        startActivity(new Intent(Intent.ACTION_VIEW,uri));
    }

    public void openCheck(){
        Intent intent2 = new Intent(this, order.class);
        startActivity(intent2);
    }
    public void openMainmenu(){
        Intent intent = new Intent(this, method.class);
        startActivity(intent);
    }


}
```

2. Main Menu

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F8B7A9"
    tools:context=".mainmenu">

    <TextView
        android:id="@+id/mainmenu_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="60dp"
        android:text="Main Menu"
        android:textColor="#000000"
        android:textSize="30sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button1"
        android:layout_width="142dp"
        android:layout_height="128dp"
        android:layout_marginStart="130dp"
        android:layout_marginTop="56dp"
        android:layout_marginEnd="130dp"
        android:background="@drawable/menulogo"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/mainmenu_title" />

    <TextView
        android:id="@+id/mainmenu_text1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
        android:layout_marginTop="50dp"
        android:layout_marginEnd="30dp"
        android:text="Manual Ordering"
        android:textColor="#000000"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button1" />

    <Button
        android:id="@+id/button2"
        android:layout_width="122dp"
        android:layout_height="101dp"
        android:layout_marginStart="130dp"
        android:layout_marginTop="50dp"
        android:layout_marginEnd="130dp"
        android:background="@drawable/voiceshopping"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/mainmenu_text1" />

    <TextView
        android:id="@+id/mainmenu_text2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
        android:layout_marginTop="50dp"
        android:layout_marginEnd="30dp"
        android:text="Express Voice Ordering"
        android:textColor="#000000"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button2" />

    <Button
        android:id="@+id/mainmenu_return"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="105dp"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="40dp"
        android:text="Cancel"
        app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toEndOf="@+id/mainmenu_title"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Java Coding:

```
package com.example.menuapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

public class mainmenu extends AppCompatActivity {

    private Button button1,button2,button3;

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

        button1 = (Button) findViewById(R.id.button1);
        button2 = (Button) findViewById(R.id.button2);
        button3 = (Button) findViewById(R.id.mainmenu_return);
        button1.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                openMenu();
            }
        });
        button2.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                openExpress();
            }
        });
        button3.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                returnWelcome();
            }
        });
    }
    public void openMenu(){
        Intent intent = new Intent(this,menu.class);

        startActivity(intent);
    }
    public void openExpress(){
        Intent intent = new Intent(this,express.class);

        startActivity(intent);
    }
    public void returnWelcome(){
        Intent intent = new Intent(this,MainActivity.class);
        startActivity(intent);
    }
}
    }
}
```

3. Manual Order Menu

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFCC2"
    tools:context=".menu">

    <TextView
        android:id="@+id/menu_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Menu Pick! ( Click food image to see
description)"
        android:textColor="#000000"
        android:textSize="16sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <ImageButton
        android:id="@+id/image1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:layout_marginTop="56dp"
        android:contentDescription="Healthy Japanese Standard
Ricebox"
        android:onClick="showBento"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/menu_title"
        app:srcCompat="@drawable/bento" />

    <ImageButton
        android:id="@+id/image2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:layout_marginTop="56dp"
        android:layout_marginEnd="40dp"
        android:contentDescription="Delicious sweet ice-cream
as dessert!"
        android:onClick="showIcecream"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/image1"
        app:layout_constraintTop_toBottomOf="@+id/menu_title"
        app:srcCompat="@drawable/icecream" />

    <Button
        android:id="@+id/increase1"
        android:layout_width="38dp"
        android:layout_height="40dp"
        android:layout_marginTop="20dp"
        android:onClick="increment1"
        android:text="+"
        android:textSize="10sp"
        app:layout_constraintStart_toEndOf="@+id/bento_price"
        app:layout_constraintTop_toBottomOf="@+id/image1" />

    <Button
        android:id="@+id/decrease1"
        android:layout_width="34dp"
        android:layout_height="40dp"
        android:layout_marginTop="20dp"
        android:onClick="decrement1"
        android:text="-"
        android:textSize="10sp"
        app:layout_constraintEnd_toStartOf="@+id/bento_price"
        app:layout_constraintTop_toBottomOf="@+id/image1" />
    <TextView
        android:id="@+id/amount1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0"
        android:textColor="#000000"

app:layout_constraintBottom_toBottomOf="@+id/decrease1"
        app:layout_constraintEnd_toEndOf="@+id/bento_price"

app:layout_constraintStart_toStartOf="@+id/bento_price"
        app:layout_constraintTop_toTopOf="@+id/decrease1" />

    <ImageButton
        android:id="@+id/image3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        android:contentDescription="Fresh and tasty chews of
vege bowl!"
        android:onClick="showSalad"
        app:layout_constraintStart_toStartOf="@+id/image1"
        app:layout_constraintTop_toBottomOf="@+id/image1"
        app:srcCompat="@drawable/salad" />

    <ImageButton
        android:id="@+id/image4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="Glutinous bowl of rice to
fill you up!"
        android:onClick="showRice"
        app:layout_constraintEnd_toEndOf="@+id/image2"
        app:layout_constraintStart_toStartOf="@+id/image2"
        app:layout_constraintTop_toTopOf="@+id/image3"
        app:srcCompat="@drawable/ricebowl" />

    <Button
        android:id="@+id/increment2"
        android:layout_width="36dp"
        android:layout_height="37dp"
        android:layout_marginTop="20dp"
        android:onClick="increment2"
        android:text="+"
        android:textSize="10sp"

app:layout_constraintStart_toEndOf="@+id/icecream_price"
        app:layout_constraintTop_toBottomOf="@+id/image2" />

    <Button
        android:id="@+id/decrement2"
        android:layout_width="35dp"
        android:layout_height="40dp"
        android:layout_marginTop="20dp"
        android:onClick="decrement2"
        android:text="-"
        android:textSize="10sp"

app:layout_constraintEnd_toStartOf="@+id/icecream_price"
        app:layout_constraintTop_toBottomOf="@+id/image2" />

    <TextView
        android:id="@+id/amount2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0"
        android:textColor="#000000"

app:layout_constraintBottom_toBottomOf="@+id/decrement2"
        app:layout_constraintEnd_toEndOf="@+id/icecream_price"

app:layout_constraintStart_toStartOf="@+id/icecream_price"
        app:layout_constraintTop_toTopOf="@+id/decrement2"
        app:layout_constraintVertical_bias="0.476" />

    <Button
        android:id="@+id/decrease3"
        android:layout_width="38dp"
        android:layout_height="43dp"
        android:layout_marginTop="20dp"
        android:onClick="decrement3"
        android:text="-"
        android:textSize="10sp"
        app:layout_constraintEnd_toStartOf="@+id/salad_price"
        app:layout_constraintTop_toBottomOf="@+id/image3" />

    <Button
        android:id="@+id/increase3"
        android:layout_width="38dp"
        android:layout_height="44dp"
        android:layout_marginTop="20dp"
        android:onClick="increment3"
        android:text="+"
        android:textSize="10sp"
        app:layout_constraintStart_toEndOf="@+id/salad_price"
        app:layout_constraintTop_toBottomOf="@+id/image3" />

    <TextView
        android:id="@+id/amount3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:text="0"
        android:textColor="#000000"
```

```
app:layout_constraintBottom_toBottomOf="@+id/decrease3"
        app:layout_constraintEnd_toEndOf="@+id/salad_price"

app:layout_constraintStart_toStartOf="@+id/salad_price"
        app:layout_constraintTop_toTopOf="@+id/decrease3"
        app:layout_constraintVertical_bias="0.0" />

    <Button
        android:id="@+id/decrease4"
        android:layout_width="36dp"
        android:layout_height="44dp"
        android:layout_marginTop="20dp"
        android:onClick="decrement4"
        android:text="-"
        android:textSize="10sp"
        app:layout_constraintEnd_toStartOf="@+id/rice_price"
        app:layout_constraintTop_toBottomOf="@+id/image4" />

    <Button
        android:id="@+id/increase4"
        android:layout_width="37dp"
        android:layout_height="43dp"
        android:layout_marginTop="20dp"
        android:onClick="increment4"
        android:text="+"
        android:textSize="10sp"
        app:layout_constraintStart_toEndOf="@+id/rice_price"
        app:layout_constraintTop_toBottomOf="@+id/image4" />

    <TextView
        android:id="@+id/amount4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:layout_marginEnd="10dp"
        android:text="0"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/rice_price"
        app:layout_constraintStart_toStartOf="@+id/rice_price"
        app:layout_constraintTop_toTopOf="@+id/decrease4" />

    <Button
        android:id="@+id/menubutton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:layout_marginEnd="157dp"
        android:text="Confirm"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="@+id/increase3"
        app:layout_constraintTop_toBottomOf="@+id/increase3"
/>
```

```
    <Button
        android:id="@+id/menubutton2"
        android:layout_width="99dp"
        android:layout_height="51dp"
        android:layout_marginTop="15dp"
        android:text="Cancel"
        android:textSize="11sp"
        app:layout_constraintEnd_toEndOf="@+id/menubutton1"

app:layout_constraintStart_toStartOf="@+id/menubutton1"
        app:layout_constraintTop_toBottomOf="@+id/menubutton1"
/>

    <TextView
        android:id="@+id/bento_price"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Price: 15.00"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/image1"
        app:layout_constraintStart_toStartOf="@+id/image1"
        app:layout_constraintTop_toBottomOf="@+id/image1" />

    <TextView
        android:id="@+id/bento"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="2dp"
        android:text="Japanese Bento Set"
        android:textColor="#000000"
        app:layout_constraintBottom_toTopOf="@+id/image1"
        app:layout_constraintStart_toStartOf="@+id/image1" />

    <TextView
        android:id="@+id/icecream"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="2dp"
        android:text="3-Scoop Ice-Cream"
        android:textColor="#000000"
        android:textSize="12sp"
        app:layout_constraintBottom_toTopOf="@+id/image2"
        app:layout_constraintEnd_toEndOf="@+id/image2"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="@+id/image2" />

    <TextView
        android:id="@+id/icecream_price"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="1dp"
        android:text="Price: 5.00"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/image2"
        app:layout_constraintStart_toStartOf="@+id/image2"
        app:layout_constraintTop_toBottomOf="@+id/image2" />

    <TextView
        android:id="@+id/salad"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="2dp"
        android:text="Organic Coleslaw Salad"
        android:textColor="#000000"
        android:textSize="11sp"
        app:layout_constraintBottom_toTopOf="@+id/image3"
        app:layout_constraintStart_toStartOf="@+id/image3" />

    <TextView
        android:id="@+id/salad_price"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="3dp"
        android:text="Price: 5.00"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/image3"
        app:layout_constraintStart_toStartOf="@+id/image3"
        app:layout_constraintTop_toBottomOf="@+id/image3" />

    <TextView
        android:id="@+id/rice"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="3dp"
        android:text="Healthy Rice Bowl"
        android:textColor="#000000"
        app:layout_constraintBottom_toTopOf="@+id/image4"
        app:layout_constraintEnd_toEndOf="@+id/image4"
        app:layout_constraintStart_toStartOf="@+id/image4" />

    <TextView
        android:id="@+id/rice_price"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="1dp"
        android:text="Price: 7.00"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/image4"
        app:layout_constraintStart_toStartOf="@+id/image4"
        app:layout_constraintTop_toBottomOf="@+id/image4" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Java Coding:

```java
package com.example.menuapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class menu extends AppCompatActivity {

    public static final String
EXTRA1="com.example.menuapp.EXTRA1";

    private Button button1, button2;
    private TextView value1,value2,value3,value4;
    private TextView bento,icecream;
    int count1=0;
    int count2=0;
    int count3=0;
    int count4=0;


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

        button1 = (Button) findViewById(R.id.menubutton1);
        button2 = (Button) findViewById(R.id.menubutton2);
        button1.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                openConfirmation();
            }
        });
        button2.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                returnMain();
            }
        });
        value1 = (TextView) findViewById(R.id.amount1);
        value2 = (TextView) findViewById(R.id.amount2);
        value3 = (TextView) findViewById(R.id.amount3);
        value4 = (TextView) findViewById(R.id.amount4);
    }
    public void openConfirmation() {
        String bentoString=value1.getText().toString().trim();
        String
icecreamString=value2.getText().toString().trim();
        String saladString=value3.getText().toString().trim();
        String riceString=value4.getText().toString().trim();

        Bundle bundle = new Bundle();
        bundle.putString("bento", bentoString);
        bundle.putString("icecream", icecreamString);
        bundle.putString("salad", saladString);
        bundle.putString("rice", riceString);

        Intent intent = new Intent(this, Confirmation.class);
        intent.putExtras(bundle);

        startActivity(intent);

    }
    public void returnMain() {
        Intent intent = new Intent(this, mainmenu.class);
        startActivity(intent);
    }
    public void increment1(View view){
        count1++;
        value1.setText(String.valueOf(count1));
    }
    public void decrement1(View view){
        if(count1<=0){
            count1=0;
        }
        else count1--;
        value1.setText(String.valueOf(count1));

    }
    public void increment2(View view){
        count2++;
        value2.setText("" + count2);
    }
    public void decrement2(View view){
        if(count2<=0){
            count2=0;
        }
        else count2--;
        value2.setText("" + count2);
    }
    public void increment3(View view){
        count3++;
        value3.setText("" + count3);
    }
    public void decrement3(View view){
        if(count3<=0){
            count3=0;
        }
        else count3--;
        value3.setText("" + count3);
    }
    public void increment4(View view){
        count4++;
        value4.setText("" + count4);
    }
    public void decrement4(View view){
        if(count4<=0){
            count4=0;
        }
        else count4--;
        value4.setText("" + count4);
    }
    public void displayToast(String message) {
        Toast.makeText(getApplicationContext(), message,
                Toast.LENGTH_SHORT).show();
    }

    public void showBento(View view) {
        displayToast(getString(R.string.bento_message));
    }
    public void showIcecream(View view) {
        displayToast(getString(R.string.icecream_message));
    }
    public void showSalad(View view) {
        displayToast(getString(R.string.salad_message));
    }
    public void showRice(View view) {
        displayToast(getString(R.string.rice_message));
    }

}
```

4. Express Menu

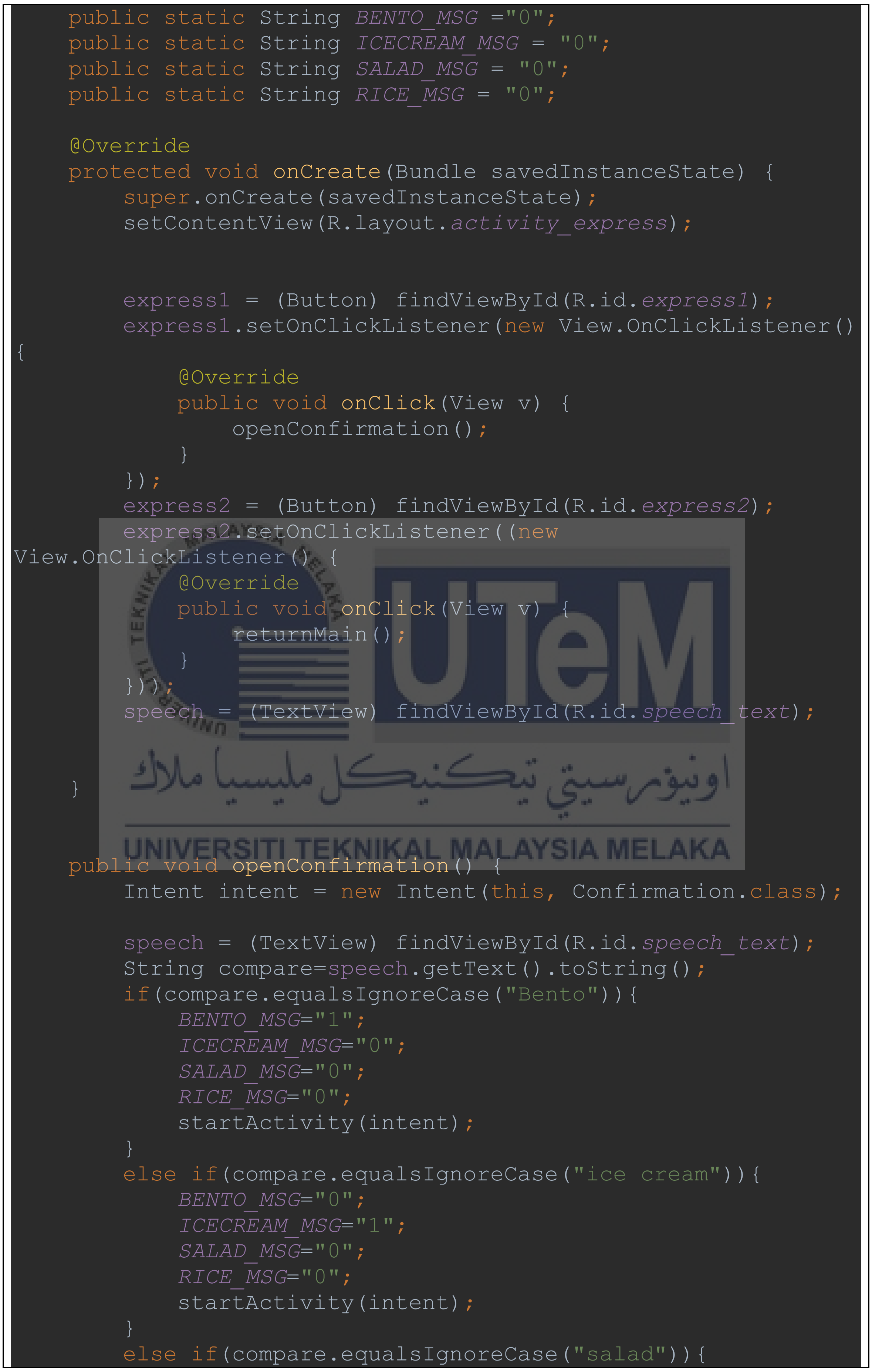
XML:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFCC2"
    tools:context=".express">

    <TextView
```

```
        android:id="@+id/express_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:text="Express Menu"
        android:textColor="#000000"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/express1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="28dp"
        android:text="Confirm"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/express_max"
/>

    <Button
        android:id="@+id/express2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Return to Main Menu"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/express1" />

    <TextView
        android:id="@+id/express_describe"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        android:text="(Can only ask for one unit per item in
our menu)"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/express_title" />

    <ImageView
        android:id="@+id/speak"
        android:layout_width="160dp"
        android:layout_height="156dp"
        android:layout_marginTop="50dp"
        android:onClick="openSpeak"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/speech_text"
        app:srcCompat="@drawable/voice" />

    <TextView
        android:id="@+id/speech_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="47dp"
        android:text="Speak Your Order"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/express_describe" />

    <TextView
        android:id="@+id/express_max"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Maximum 2 types of Items for 1 unit
only!"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/speak" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Java Coding:

```
package com.example.menuapp;

import android.content.Intent;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
import java.util.Locale;

public class express extends AppCompatActivity {

    private Button express1, express2;
    private TextView speech;
    private static final int REQUEST_CODE_SPEECH_INPUT=1000;
```



```
    public static String BENTO_MSG ="0";
    public static String ICECREAM_MSG = "0";
    public static String SALAD_MSG = "0";
    public static String RICE_MSG = "0";

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


        express1 = (Button) findViewById(R.id.express1);
        express1.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                openConfirmation();
            }
        });
        express2 = (Button) findViewById(R.id.express2);
        express2.setOnClickListener((new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                returnMain();
            }
        }));
        speech = (TextView) findViewById(R.id.speech_text);


    }


    public void openConfirmation() {
        Intent intent = new Intent(this, Confirmation.class);

        speech = (TextView) findViewById(R.id.speech_text);
        String compare=speech.getText().toString();
        if(compare.equalsIgnoreCase("Bento")){
            BENTO_MSG="1";
            ICECREAM_MSG="0";
            SALAD_MSG="0";
            RICE_MSG="0";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("ice cream")){
            BENTO_MSG="0";
            ICECREAM_MSG="1";
            SALAD_MSG="0";
            RICE_MSG="0";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("salad")){
            BENTO_MSG="0";
            ICECREAM_MSG="0";
            SALAD_MSG="1";
            RICE_MSG="0";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("rice bowl")){
            BENTO_MSG="0";
            ICECREAM_MSG="0";
            SALAD_MSG="0";
            RICE_MSG="1";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("bento & ice
cream")){
            BENTO_MSG="1";
            ICECREAM_MSG="1";
            SALAD_MSG="0";
            RICE_MSG="0";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("bento and salad")){
            BENTO_MSG="1";
            ICECREAM_MSG="0";
            SALAD_MSG="1";
            RICE_MSG="0";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("bento and rice
bowl")){
            BENTO_MSG="1";
            ICECREAM_MSG="0";
            SALAD_MSG="0";
            RICE_MSG="1";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("ice cream and
bento")){
            BENTO_MSG="1";
            ICECREAM_MSG="1";
            SALAD_MSG="0";
            RICE_MSG="0";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("ice cream and
salad")){
            BENTO_MSG="0";
            ICECREAM_MSG="1";
            SALAD_MSG="1";
            RICE_MSG="0";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("ice cream and rice
```

```java
bowl")){
            BENTO_MSG="0";
            ICECREAM_MSG="1";
            SALAD_MSG="0";
            RICE_MSG="1";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("salad and bento")){
            BENTO_MSG="1";
            ICECREAM_MSG="0";
            SALAD_MSG="1";
            RICE_MSG="0";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("salad and ice
cream")){
            BENTO_MSG="0";
            ICECREAM_MSG="1";
            SALAD_MSG="1";
            RICE_MSG="0";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("salad and rice
bowl")){
            BENTO_MSG="0";
            ICECREAM_MSG="0";
            SALAD_MSG="1";
            RICE_MSG="1";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("rice bowl and
bento")){
            BENTO_MSG="1";
            ICECREAM_MSG="0";
            SALAD_MSG="0";
            RICE_MSG="1";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("rice bowl and ice
cream")){
            BENTO_MSG="0";
            ICECREAM_MSG="1";
            SALAD_MSG="0";
            RICE_MSG="1";
            startActivity(intent);
        }
        else if(compare.equalsIgnoreCase("rice bowl and
salad")){
            BENTO_MSG="0";
            ICECREAM_MSG="0";
            SALAD_MSG="1";
            RICE_MSG="1";
            startActivity(intent);
        }
        else{
            Toast.makeText(this, "Error message received,
please try again !", Toast.LENGTH_SHORT).show();
        }




    }

    public void returnMain() {
        Intent intent = new Intent(this, mainmenu.class);
        startActivity(intent);
    }

    public void openSpeak(View view) {
        Intent intent = new
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
        intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);

intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE,Locale.getDefa
ult());
        intent.putExtra(RecognizerIntent.EXTRA_PROMPT,"Hi,
speak something");

//start intent
        try {
            //if error

startActivityForResult(intent,REQUEST_CODE_SPEECH_INPUT);
        }
        catch(Exception e){
            //if no error
            Toast.makeText(this, ""+e.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    protected void onActivityResult(int requestCode,int
resultCode,@Nullable Intent data){
        super.onActivityResult(requestCode,resultCode,data);

        switch (requestCode){
            case REQUEST_CODE_SPEECH_INPUT:{
                if (resultCode ==RESULT_OK && null!=data) {
                    ArrayList<String> results =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);

                    speech.setText(results.get(0));
                }
            }
```

```
                break;
            }
        }


}
```

5. Confirmation Page

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F9CBB2"
    tools:context=".Confirmation">

    <TextView
        android:id="@+id/confirmation_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="10dp"
        android:layout_marginTop="15dp"
        android:text="Confirm Your Order !"
        android:textColor="#000000"
        android:textSize="20sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/confirmation_item"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="70dp"
        android:layout_marginEnd="30dp"
        android:text="Item Ordered:"
        android:textColor="#000000"
        android:textSize="18sp"
        app:layout_constraintEnd_toStartOf="@+id/textView"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/confirmation_title"
/>

    <TextView
        android:id="@+id/confirmation_price"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp"
        android:text="Total Price:"
        android:textColor="#000000"
        android:textSize="18sp"

app:layout_constraintStart_toStartOf="@+id/confirmation_item"
        app:layout_constraintTop_toBottomOf="@+id/textView4"
/>

    <Button
        android:id="@+id/confirm_button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:text="Confirm Order"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/totalPrice"
/>

    <Button
        android:id="@+id/confirm_button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="Return to Main Menu"

app:layout_constraintEnd_toEndOf="@+id/confirm_button1"

app:layout_constraintStart_toStartOf="@+id/confirm_button1"

app:layout_constraintTop_toBottomOf="@+id/confirm_button1" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="36dp"
        android:textColor="#000000"
        app:layout_constraintEnd_toStartOf="@+id/textView5"

app:layout_constraintTop_toTopOf="@+id/confirmation_item" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:textColor="#000000"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/textView" />
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:textColor="#000000"
        app:layout_constraintStart_toStartOf="@+id/textView2"
        app:layout_constraintTop_toBottomOf="@+id/textView2"
/>

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:textColor="#000000"
        app:layout_constraintStart_toStartOf="@+id/textView3"
        app:layout_constraintTop_toBottomOf="@+id/textView3"
/>

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
        android:layout_marginEnd="97dp"
        android:text="Japanest Bento Set"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/confirmation_title"
        app:layout_constraintTop_toTopOf="@+id/textView" />

    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="3 Scoop Ice Cream"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/textView5"
        app:layout_constraintStart_toStartOf="@+id/textView5"
        app:layout_constraintTop_toBottomOf="@+id/textView5"
/>

    <TextView
        android:id="@+id/textView7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Organic Colesaw Salad"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/textView6"
        app:layout_constraintStart_toStartOf="@+id/textView6"
        app:layout_constraintTop_toBottomOf="@+id/textView6"
/>

    <TextView
        android:id="@+id/textView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Healthy Rice Bowl"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/textView7"
        app:layout_constraintStart_toStartOf="@+id/textView7"
        app:layout_constraintTop_toBottomOf="@+id/textView7"
/>

    <TextView
        android:id="@+id/totalPrice"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="95dp"
        android:hint="price show here"
        android:textColor="#000000"
        android:textSize="24sp"

app:layout_constraintBottom_toBottomOf="@+id/confirmation_pric
e"
        app:layout_constraintEnd_toEndOf="@+id/textView8"

app:layout_constraintStart_toEndOf="@+id/confirmation_price"

app:layout_constraintTop_toTopOf="@+id/confirmation_price"
        app:layout_constraintVertical_bias="0.0" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Java Coding:

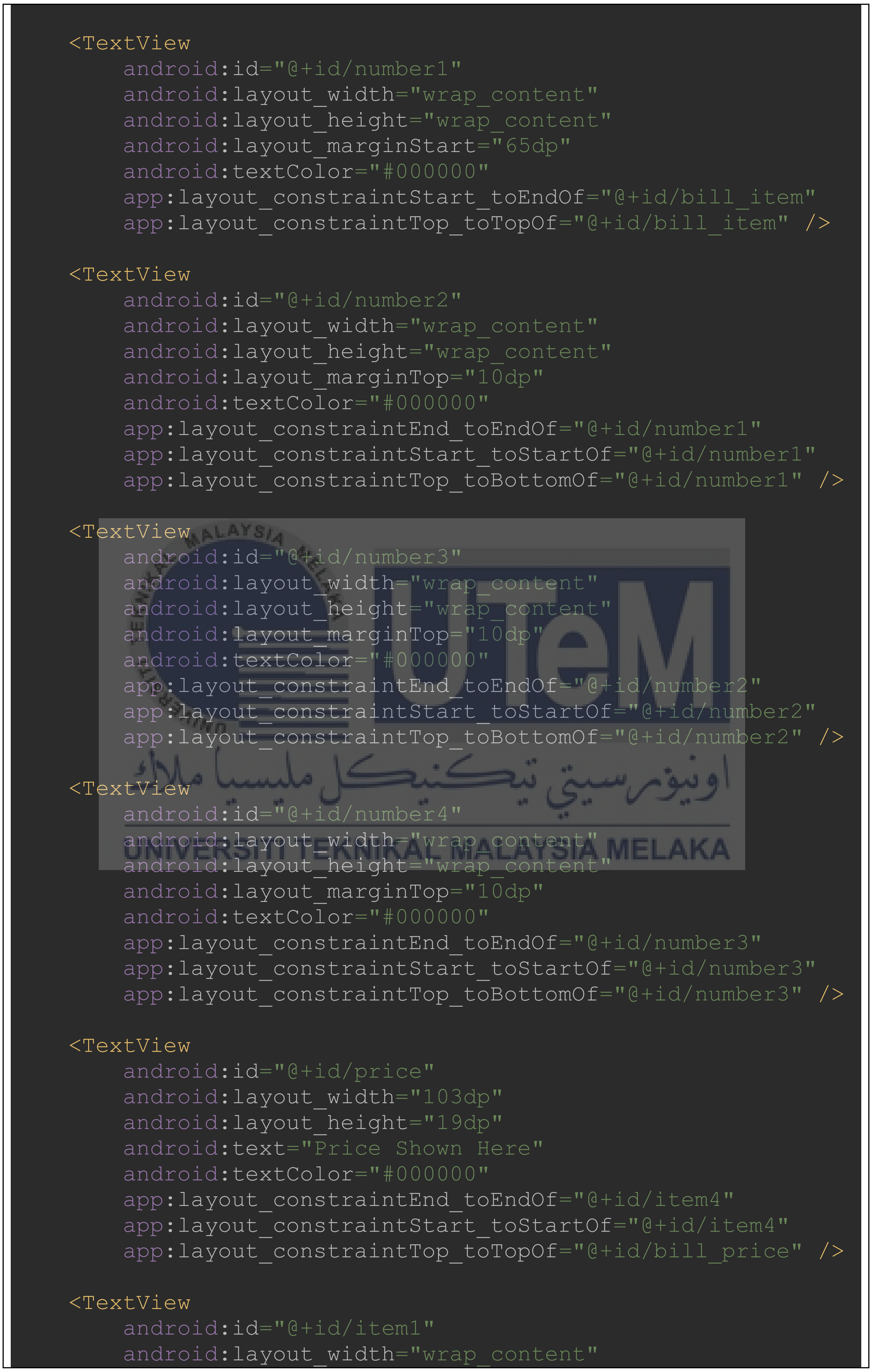
```
package com.example.menuapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import java.text.DecimalFormat;
```

```
public class Confirmation extends AppCompatActivity {

    private Button button1, button2;
    private static DecimalFormat df2 = new
DecimalFormat("#.##");
    private float
totalprice,bentoprice,icecreamprice,saladprice,riceprice;

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

        TextView itemtext=findViewById(R.id.textView);
        itemtext.setText(express.BENTO_MSG);
        TextView itemtext2=findViewById(R.id.textView2);
        itemtext2.setText(express.ICECREAM_MSG);
        TextView itemtext3=findViewById(R.id.textView3);
        itemtext3.setText(express.SALAD_MSG);
        TextView itemtext4=findViewById(R.id.textView4);
        itemtext4.setText(express.RICE_MSG);

        bentoprice=15*Float.parseFloat(express.BENTO_MSG);

icecreamprice=5*Float.parseFloat(express.ICECREAM_MSG);
        saladprice=5*Float.parseFloat(express.SALAD_MSG);
        riceprice=7*Float.parseFloat(express.RICE_MSG);

totalprice=bentoprice+icecreamprice+saladprice+riceprice;

((TextView)findViewById(R.id.totalPrice)).setText(String.value
Of(totalprice)+df2.format(0));

        Bundle bundle=getIntent().getExtras();
        if(bundle!=null){
            String bento=bundle.getString("bento");
            String icecream=bundle.getString("icecream");
            String salad=bundle.getString("salad");
            String rice=bundle.getString("rice");

            TextView bentotext=findViewById(R.id.textView);
            TextView
icecreamtext=findViewById(R.id.textView2);
            TextView saladtext=findViewById(R.id.textView3);
            TextView ricetext=findViewById(R.id.textView4);

            bentotext.setText(bento);
            icecreamtext.setText(icecream);
            saladtext.setText(salad);
            ricetext.setText(rice);

            bentoprice=15*Float.parseFloat(bento);
            icecreamprice=5*Float.parseFloat(icecream);
            saladprice=5*Float.parseFloat(salad);
            riceprice=7*Float.parseFloat(rice);

totalprice=bentoprice+icecreamprice+saladprice+riceprice;


((TextView)findViewById(R.id.totalPrice)).setText(String.value
Of(totalprice)+df2.format(0));
        }

        button1 = (Button) findViewById(R.id.confirm_button1);
        button2 = (Button) findViewById(R.id.confirm_button2);
        button1.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                openBill();
            }
        });
        button2.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                returnMain();
            }
        });
    }
    public void openBill(){
        String
bentoString=((TextView)findViewById(R.id.textView)).getText().
toString();
        String icecreamString=
((TextView)findViewById(R.id.textView2)).getText().toString();
        String saladString=
((TextView)findViewById(R.id.textView3)).getText().toString();
        String riceString=
((TextView)findViewById(R.id.textView4)).getText().toString();

        Bundle bundle = new Bundle();
        bundle.putString("bento", bentoString);
        bundle.putString("icecream", icecreamString);
        bundle.putString("salad", saladString);
        bundle.putString("rice", riceString);
        Intent intent = new Intent(this,Bill.class);
        intent.putExtras(bundle);

        if(totalprice==0.00){
            Toast.makeText(this,"You have not chosen anything
from the menu", Toast.LENGTH_SHORT).show();
        }
        else {
            startActivity(intent);
        }
    }
    public void returnMain(){
        Intent intent = new Intent(this,mainmenu.class);
        startActivity(intent);
    }
}
```

6. Bill Page

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F9CBB2"
    tools:context=".Bill">

    <TextView
        android:id="@+id/Bill_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="10dp"
        android:layout_marginTop="15dp"
        android:text="Here is Your Bill!"
        android:textColor="#000000"
        android:textSize="22sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/bill_item"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Item Info:"
        android:textColor="#000000"
        app:layout_constraintStart_toStartOf="@+id/Bill_title"
        app:layout_constraintTop_toBottomOf="@+id/Bill_title"
/>

    <TextView
        android:id="@+id/bill_price"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="Total Price:"
        android:textColor="#000000"
        app:layout_constraintStart_toStartOf="@+id/bill_item"
        app:layout_constraintTop_toBottomOf="@+id/number4" />

    <TextView
        android:id="@+id/bill_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="22dp"
        android:text="Please Pay Your Bill And Collect Your
Order At The Counter"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.644"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/number" />

    <Button
        android:id="@+id/bill_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="60dp"
        android:text="Confirm and Back to Main Menu"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/bill_text"
/>

    <TextView
        android:id="@+id/bill_number"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="60dp"
        android:text="Your Order Number is:"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/price" />

    <TextView
        android:id="@+id/number"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp"
        android:text="0"
        android:textColor="#000000"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="@+id/bill_number"

app:layout_constraintStart_toStartOf="@+id/bill_number"
        app:layout_constraintTop_toBottomOf="@+id/bill_number"
/>
```



```xml
    <TextView
        android:id="@+id/number1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="65dp"
        android:textColor="#000000"
        app:layout_constraintStart_toEndOf="@+id/bill_item"
        app:layout_constraintTop_toTopOf="@+id/bill_item" />

    <TextView
        android:id="@+id/number2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/number1"
        app:layout_constraintStart_toStartOf="@+id/number1"
        app:layout_constraintTop_toBottomOf="@+id/number1" />

    <TextView
        android:id="@+id/number3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/number2"
        app:layout_constraintStart_toStartOf="@+id/number2"
        app:layout_constraintTop_toBottomOf="@+id/number2" />

    <TextView
        android:id="@+id/number4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/number3"
        app:layout_constraintStart_toStartOf="@+id/number3"
        app:layout_constraintTop_toBottomOf="@+id/number3" />

    <TextView
        android:id="@+id/price"
        android:layout_width="103dp"
        android:layout_height="19dp"
        android:text="Price Shown Here"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/item4"
        app:layout_constraintStart_toStartOf="@+id/item4"
        app:layout_constraintTop_toTopOf="@+id/bill_price" />

    <TextView
        android:id="@+id/item1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="50dp"
        android:text="Japanese Bento Set"
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/number1"
        app:layout_constraintTop_toTopOf="@+id/number1" />

    <TextView
        android:id="@+id/item2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="3 Scoop Ice Cream"
        android:textColor="#000000"
        app:layout_constraintStart_toStartOf="@+id/item1"
        app:layout_constraintTop_toBottomOf="@+id/item1" />

    <TextView
        android:id="@+id/item3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="Organic Colesaw Salad"
        android:textColor="#000000"
        app:layout_constraintStart_toStartOf="@+id/item2"
        app:layout_constraintTop_toBottomOf="@+id/item2" />

    <TextView
        android:id="@+id/item4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="Fresh Rice Bowl"
        android:textColor="#000000"
        app:layout_constraintStart_toStartOf="@+id/item3"
        app:layout_constraintTop_toBottomOf="@+id/item3" />

    <TextView
        android:id="@+id/methodText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Dine-in/Takeaway:"
        android:textColor="#000000"
        app:layout_constraintStart_toStartOf="@+id/bill_price"
        app:layout_constraintTop_toBottomOf="@+id/bill_price"
/>

    <TextView
        android:id="@+id/bill_method"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="23dp"
        android:textColor="#000000"
        app:layout_constraintStart_toStartOf="@+id/price"
        app:layout_constraintTop_toBottomOf="@+id/price" />

    <TextView
        android:id="@+id/bill_text2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text='Search your order at "Order Search" in
Main Menu'
        android:textColor="#000000"
        app:layout_constraintEnd_toEndOf="@+id/bill_text"
        app:layout_constraintStart_toStartOf="@+id/bill_text"
        tools:layout_editor_absoluteY="404dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Java Coding:

```
package com.example.menuapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.text.DecimalFormat;
import java.util.Random;

public class Bill extends AppCompatActivity {

    private Button button;
    private TextView orderText, method;
    private static DecimalFormat df2 = new
DecimalFormat("#.##");
    private float totalprice, bentoprice, icecreamprice,
saladprice, riceprice;
    private DatabaseReference OrderNumber, Status, Method;
    private FirebaseDatabase database;
    String gotString = Global.stringToPass;

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

        Bundle bundle = getIntent().getExtras();
        if (bundle != null) {
            String bento = bundle.getString("bento");
            String icecream = bundle.getString("icecream");
            String salad = bundle.getString("salad");
            String rice = bundle.getString("rice");

            TextView bentotext = findViewById(R.id.number1);
            TextView icecreamtext =
findViewById(R.id.number2);
            TextView saladtext = findViewById(R.id.number3);
            TextView ricetext = findViewById(R.id.number4);

            bentotext.setText(bento);
            icecreamtext.setText(icecream);
            saladtext.setText(salad);
            ricetext.setText(rice);

            bentoprice = 15 * Float.parseFloat(bento);
            icecreamprice = 5 * Float.parseFloat(icecream);
            saladprice = 5 * Float.parseFloat(salad);
            riceprice = 7 * Float.parseFloat(rice);
            totalprice = bentoprice + icecreamprice +
saladprice + riceprice;
            TextView totaltext = findViewById(R.id.price);
            totaltext.setText(String.valueOf(totalprice) +
df2.format(0));
        }

        method = findViewById(R.id.bill_method);
        method.setText(getString);
        button = (Button) findViewById(R.id.bill_button);
        orderText = (TextView) findViewById(R.id.number);
        randomNumber();


        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openMain();
            }
        });

        Toast.makeText(this, "Order Made! Please check at
Order Search", Toast.LENGTH_SHORT).show();

    }

    public void openMain() {
        String a = orderText.getText().toString();
        database = FirebaseDatabase.getInstance();
        Database b = new Database(gotString, "Imcomplete", a);
        database.getReference(a).setValue(b);
        Intent intent = new Intent(this, MainActivity.class);
        startActivity(intent);
    }

    public void randomNumber() {
        Random orderNumber = new Random();
        int random = orderNumber.nextInt(10001 - 1) + 1;
        orderText.setText(Integer.toString(random));

    }

}
```

7. Takeaway / Dine-in Page

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F8B7A9"
    tools:context=".method">

    <ImageButton
        android:id="@+id/method_dineButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/method_title"
        app:srcCompat="@drawable/icons8_restaurant_160" />

    <TextView
        android:id="@+id/method_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        android:text="Order Method"
        android:textColor="#000000"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:id="@+id/method_dine"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        android:text="Dine-In"
        android:textColor="#000000"
        android:textSize="18sp"

app:layout_constraintEnd_toEndOf="@+id/method_dineButton"

app:layout_constraintStart_toStartOf="@+id/method_dineButton"

app:layout_constraintTop_toBottomOf="@+id/method_dineButton"
/>

    <ImageButton
        android:id="@+id/method_takeButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/method_dine"
        app:srcCompat="@drawable/icons8_takeaway_128" />

    <TextView
        android:id="@+id/method_takeaway"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        android:text="Takeaway"
        android:textColor="#000000"
        android:textSize="18sp"

app:layout_constraintEnd_toEndOf="@+id/method_takeButton"

app:layout_constraintStart_toStartOf="@+id/method_takeButton"

app:layout_constraintTop_toBottomOf="@+id/method_takeButton"
/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Java Code:

```
package com.example.menuapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;

import androidx.appcompat.app.AppCompatActivity;

public class method extends AppCompatActivity {
    private ImageButton method_dineButton;
    private ImageButton take_button;

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

        method_dineButton = (ImageButton)
findViewById(R.id.method_dineButton);
        method_dineButton.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openMain();
            }

        });
        take_button = (ImageButton)
findViewById(R.id.method_takeButton);
        take_button.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openMain2();
            }
        });
    }

    public void openMain() {
         Global.stringToPass="Dine-In";
        Intent intent = new Intent(this, mainmenu.class);

        startActivity(intent);

    }

    public void openMain2() {
        Global.stringToPass="Takeaway";
        Intent intent = new Intent(this, mainmenu.class);

        startActivity(intent);
    }
}
```

8. Order Search

XML:

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#F8B7A9">

    <TextView
        android:id="@+id/order_text1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:background="#00FFFFFF"
        android:text="Order Search"
        android:textColor="#000000"
        android:textSize="22sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/order_menu"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="500dp"
        android:text="Back to Main Menu"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/order_text1" />

    <Button
        android:id="@+id/order_search"
        android:layout_width="180dp"
        android:layout_height="48dp"
        android:layout_marginBottom="25dp"
        android:text="Search Order"

app:layout_constraintBottom_toTopOf="@+id/order_menu"
        app:layout_constraintEnd_toEndOf="@+id/order_menu"

app:layout_constraintStart_toStartOf="@+id/order_menu" />

    <EditText
        android:id="@+id/Order_number_input"
        android:layout_width="332dp"
        android:layout_height="54dp"
        android:layout_marginStart="15dp"
        android:layout_marginTop="40dp"
        android:ems="10"
        android:hint="Order Number"
        android:inputType="textPersonName"
        android:textColor="#000000"
        android:textColorHint="#828282"
        app:layout_constraintEnd_toEndOf="@+id/order_text1"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/order_text2" />

    <TextView
        android:id="@+id/order_text2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Enter Your Order Number Here :"
        android:textColor="#000000"

app:layout_constraintStart_toStartOf="@+id/Order_number_inp
ut"

app:layout_constraintTop_toBottomOf="@+id/order_text1" />

    <TextView
        android:id="@+id/order_text3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="70dp"
        android:text="Method :"
        android:textColor="#000000"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="@+id/order_text1"
        app:layout_constraintHorizontal_bias="0.267"
        app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/Order_number_inpu
t" />

    <TextView
        android:id="@+id/order_method"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="10dp"
        android:text="method shows here"
        android:textColor="#000000"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="parent"
```

```
app:layout_constraintStart_toEndOf="@+id/order_text3"
        app:layout_constraintTop_toTopOf="@+id/order_text3"
/>

    <TextView
        android:id="@+id/order_text4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp"
        android:text="Status :"
        android:textColor="#000000"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="@+id/order_text3"
        app:layout_constraintHorizontal_bias="0.0"

app:layout_constraintStart_toStartOf="@+id/order_text3"

app:layout_constraintTop_toBottomOf="@+id/order_text3" />

    <TextView
        android:id="@+id/order_status"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="40dp"
        android:layout_marginEnd="10dp"
        android:text="status show here"
        android:textColor="#000000"
        android:textSize="18sp"

app:layout_constraintBottom_toBottomOf="@+id/order_text4"

app:layout_constraintEnd_toEndOf="@+id/order_method"

app:layout_constraintStart_toStartOf="@+id/order_method"

app:layout_constraintTop_toBottomOf="@+id/order_method" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Java Code:

```
package com.example.menuapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

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

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class order extends AppCompatActivity {

    public Button order_menu, order_search,test;
    public TextView order_method, order_status;
    public DatabaseReference OrderNumber, OrderMethod,
OrderStatus;
    public FirebaseDatabase database;
    public EditText order_number_search;
    public String number;

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


        order_method = findViewById(R.id.order_method);
        order_status = findViewById(R.id.order_status);

        order_number_search =
findViewById(R.id.Order_number_input);
        order_menu = findViewById(R.id.order_menu);

        order_menu.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openMainmenu();
            }
        });
        order_search = findViewById(R.id.order_search);
        order_search.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                openSearch();
            }
        });
       /* test = findViewById(R.id.test);


        test.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                NotificationCompat.Builder mbuilder =
(NotificationCompat.Builder)
                        new
NotificationCompat.Builder(getApplicationContext())
                                .setSmallIcon(R.drawable.ic
_launcher_background)
                .setContentTitle("Order Complete")
               .setContentText("Your order has been
complete ! Please show your order number to the counter.");


                NotificationManager notificationManager =
(NotificationManager)

getSystemService(NOTIFICATION_SERVICE);
                notificationManager.notify(1,
mbuilder.build());
            }
        });*/

    }


    public void openMainmenu() {

        Intent intent = new Intent(this,
MainActivity.class);
        startActivity(intent);

    }

    public void openSearch() {
        number = null;
        try {
            number =
order_number_search.getText().toString();
            database = FirebaseDatabase.getInstance();
            OrderNumber = database.getReference(number);
            OrderStatus =
database.getReference(number).child("status");
            OrderMethod =
database.getReference(number).child("method");
            OrderMethod.addValueEventListener(new
ValueEventListener() {

                @Override
                public void onDataChange(@NonNull
DataSnapshot snapshot) {
                    try {
                        String method =
snapshot.getValue().toString();
                        order_method.setText(method);
                    }
                    catch (Exception a){


                    }
                }


                @Override
                public void onCancelled(@NonNull
DatabaseError error) {

                }
            });}
            catch(Exception a){}
            OrderStatus.addValueEventListener(new
ValueEventListener() {
                @Override
                public void onDataChange(@NonNull
DataSnapshot snapshot) {
                    try {
                        String status =
snapshot.getValue().toString();
                        order_status.setText(status);

                    }
                    catch (Exception c){

Toast.makeText(getApplicationContext(),"Order not
Found !",Toast.LENGTH_SHORT).show();
                        order_status.setText("ERROR");
                        order_method.setText("ERROR");
                    }

                }


                @Override
                public void onCancelled(@NonNull
DatabaseError error) {

                }
            });
        //if(order_status.getText().toString() ==
"Complete"){
            /*NotificationCompat.Builder builder = new
NotificationCompat.Builder(order.this,"My notification");
            builder.setContentTitle("Order Complete");
            builder.setContentText("Your order has been
complete ! Please show your order number to the counter.");
            builder.setSmallIcon(R.drawable.notify);
            builder.setAutoCancel(true);

            NotificationManagerCompat managerCompat =
NotificationManagerCompat.from(order.this);
```

```
            managerCompat.notify(1, builder.build());*/
        //}
        }

    }
```

9. Order Update

XML:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/androi
d"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFF8BA"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/Main_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:text="Order Updates"
        android:textColor="#000000"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/main_order"
        android:layout_width="340dp"
        android:layout_height="80dp"
        android:layout_marginTop="15dp"
        android:background="#FFD269"
        android:ems="10"
        android:hint="    Order ID"
        android:inputType="textShortMessage|number"
        android:shadowColor="#000000"
        android:textColor="#000000"
        android:textColorHint="#000000"
        android:textSize="34sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@+id/main_text1" />

    <TextView
        android:id="@+id/main_text1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        android:text="Search Order ID to update"
        android:textColor="#000000"
        android:textSize="24sp"

app:layout_constraintStart_toStartOf="@+id/main_order"

app:layout_constraintTop_toBottomOf="@+id/Main_title" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="25dp"
        android:layout_marginTop="50dp"
        android:text="Status :"
        android:textColor="#000000"
        android:textSize="24sp"

app:layout_constraintStart_toStartOf="@+id/main_order"

app:layout_constraintTop_toBottomOf="@+id/main_order" />

    <TextView
        android:id="@+id/main_status"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="50dp"
        android:hint="order status"
        android:textColor="#000000"
        android:textColorHint="#000000"
        android:textSize="24sp"

app:layout_constraintStart_toEndOf="@+id/textView3"

app:layout_constraintTop_toTopOf="@+id/textView3" />

    <Button
        android:id="@+id/main_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="149dp"
        android:text="Search"

app:layout_constraintEnd_toEndOf="@+id/main_order"
```

```
app:layout_constraintStart_toStartOf="@+id/main_order"

app:layout_constraintTop_toBottomOf="@+id/textView3" />

    <Button
        android:id="@+id/complete_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:text="Complete the Order"

app:layout_constraintEnd_toEndOf="@+id/main_button"

app:layout_constraintStart_toStartOf="@+id/main_button"

app:layout_constraintTop_toBottomOf="@+id/main_button"
/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

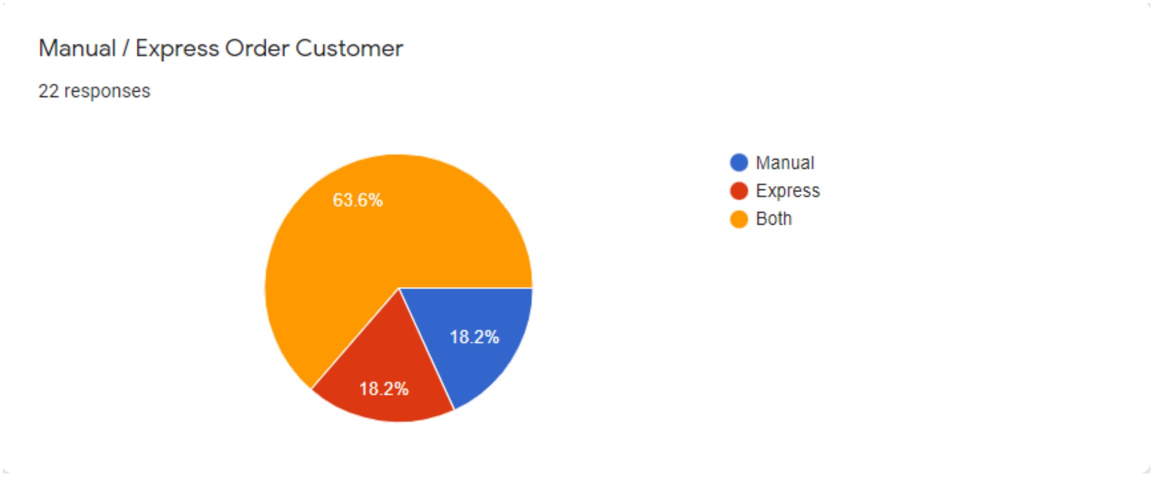
Java Code:

```
package com.example.restaurant;

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

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class MainActivity extends AppCompatActivity {
    public EditText order;
    public TextView status;
    public Button update, complete;
    public DatabaseReference OrderStatus;
    public FirebaseDatabase database;
    public String number;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        order = findViewById(R.id.main_order);
        status = findViewById(R.id.main_status);
        update = findViewById(R.id.main_button);
        update.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                statussearch();
            }
        });

        complete = findViewById(R.id.complete_button);
        complete.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                updateorder();
            }
        });
    }

    private void updateorder() {
        number = null;
        try {
            number = order.getText().toString();
            database = FirebaseDatabase.getInstance();
            OrderStatus =
database.getReference(number).child("status");
            OrderStatus.setValue("Complete");
        } catch (Exception e) {

        }
    }

    public void statussearch() {
        number = null;
        try {
            number = order.getText().toString();
            database = FirebaseDatabase.getInstance();
            OrderStatus =
database.getReference(number).child("status");
            OrderStatus.addValueEventListener(new
ValueEventListener() {
                @Override
                public void onDataChange(@NonNull
DataSnapshot snapshot) {
                    try {
                        String statusvalue =
snapshot.getValue().toString();
                        status.setText(statusvalue);
                    } catch (Exception e) {
```

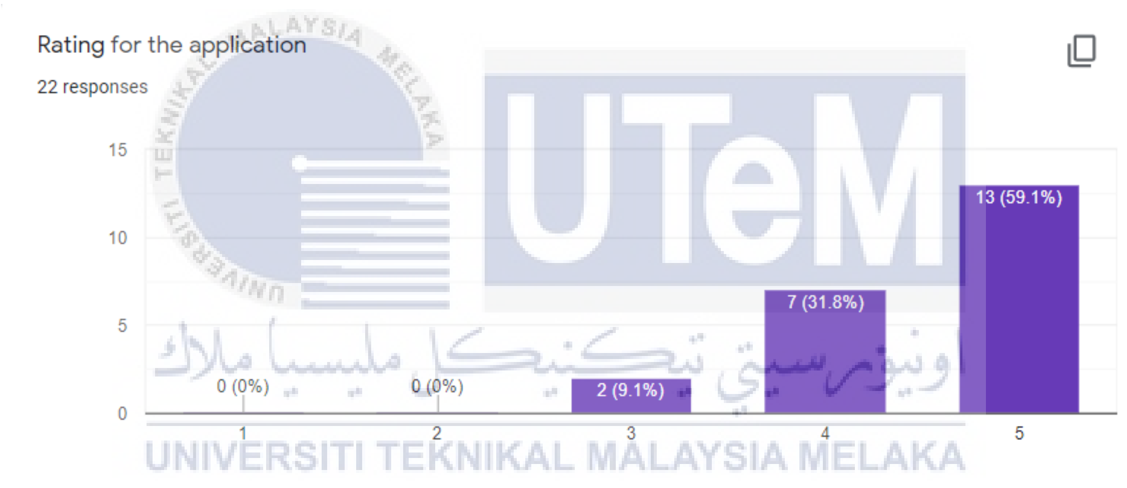
```
Toast.makeText(getApplicationContext(), "Order not
Found !", Toast.LENGTH_SHORT).show();
                        status.setText("Error");
                    }
                }

                @Override
                public void onCancelled(@NonNull
DatabaseError error) {

                }
            });
        } catch (Exception e) {

        }

    }
}
```

Appendix D

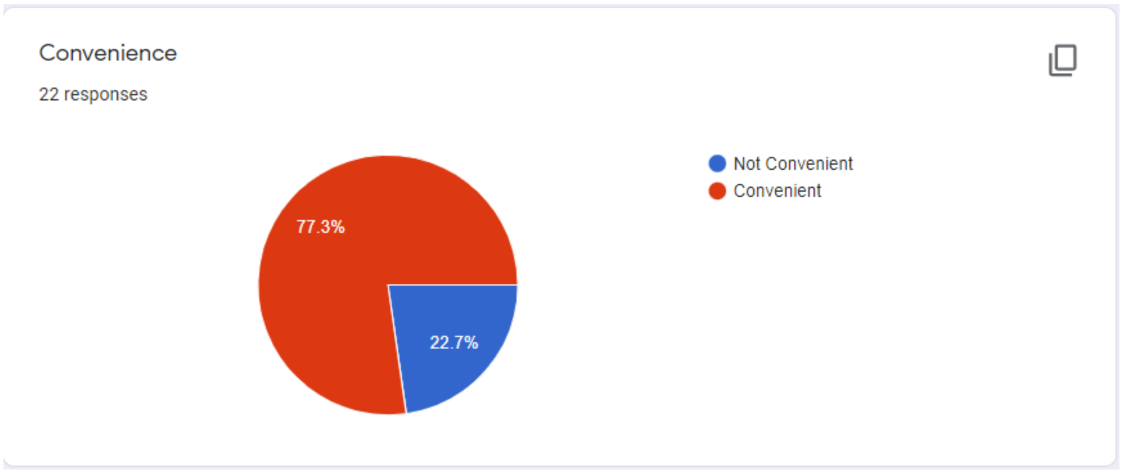
Feedback & Review Result



Manual / Express Order



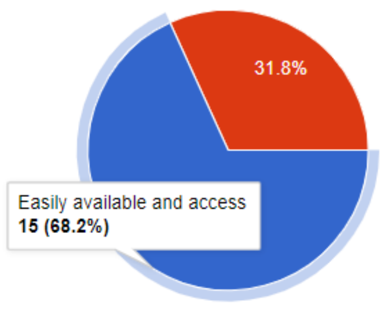
Application Ratings



Convenience Percentage

Availability and Accessibility

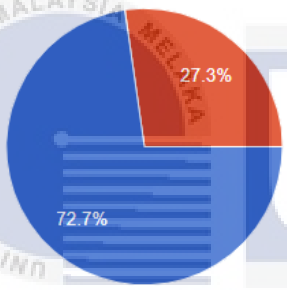
22 responses



Availability and Accessibility Percentage

Time Saving

22 responses



Time Saving Percentage