

# Assessment Cover Sheet

This Assessment Cover Sheet is only to be attached to hard copy submission of assessments.



## ASSESSMENT DETAILS

Unit title	Software Development of Mobile Devices	Tutorial /Lab Group	1	Office use only
Unit code	COS30017	Due date	2 Nov 2022	
Name of lecturer/tutor	Marlene Lu			
Assignment title	Assignment 3			Faculty or school date stamp

## STUDENT(S) DETAILS

	Student Name(s)	Student ID Number(s)
(1)	Alex Ngie Guan Ming	102765770
(2)		
(3)		
(4)		
(5)		

## DECLARATION AND STATEMENT OF AUTHORSHIP

1. I/we have not impersonated, or allowed myself/ourselves to be impersonated by any person for the purposes of this assessment.
2. This assessment is my/our original work and no part of it has been copied from any other source except where due acknowledgement is made.
3. No part of this assessment has been written for me/us by any other person except where such collaboration has been authorised by the lecturer/tutor concerned.
4. I/we have not previously submitted this work for this or any other course/unit.
5. I/we give permission for my/our assessment response to be reproduced, communicated, compared and archived for plagiarism detection, benchmarking or educational purposes.

I/we understand that:

6. Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to exclusion from the University. Plagiarised material can be drawn from, and presented in, written, graphic and visual form, including electronic data and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.

### Student signature/s

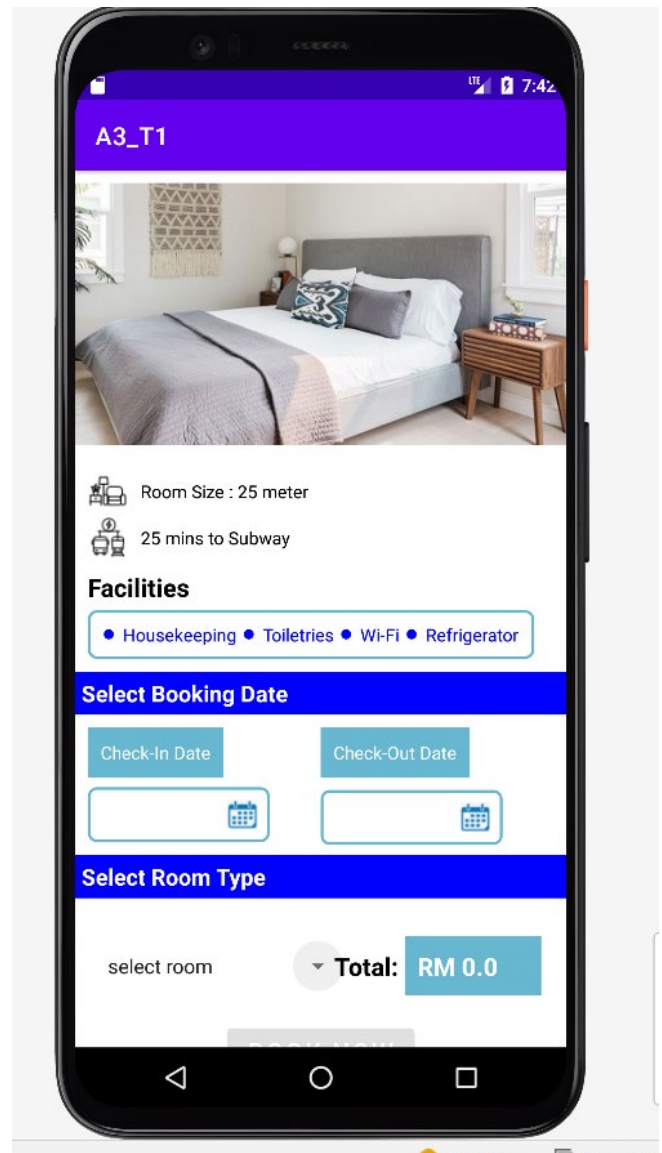
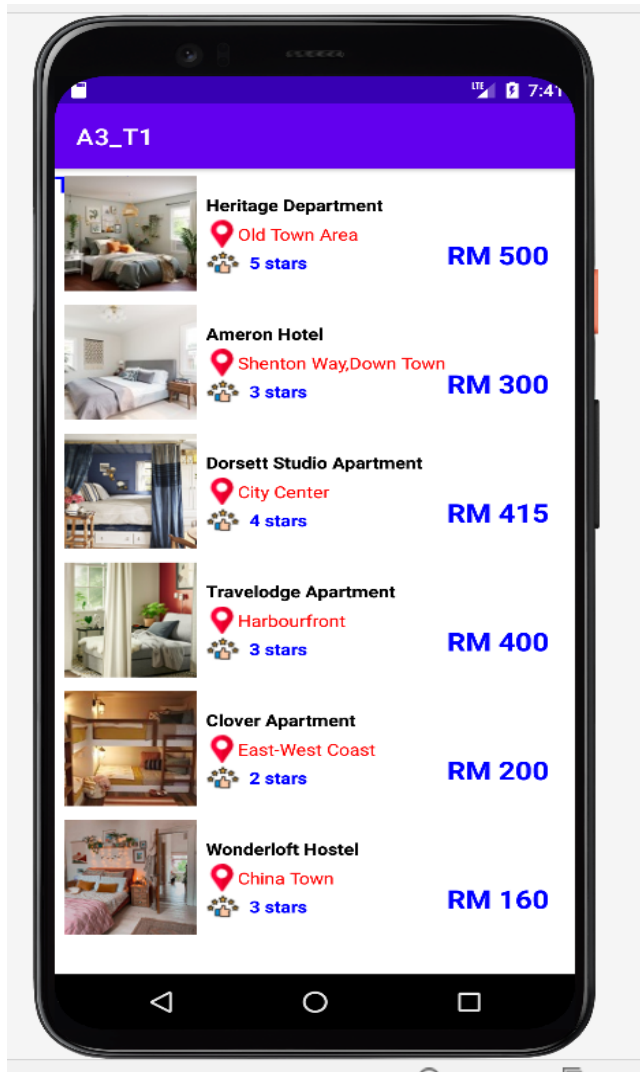
I/we declare that I/we have read and understood the declaration and statement of authorship.

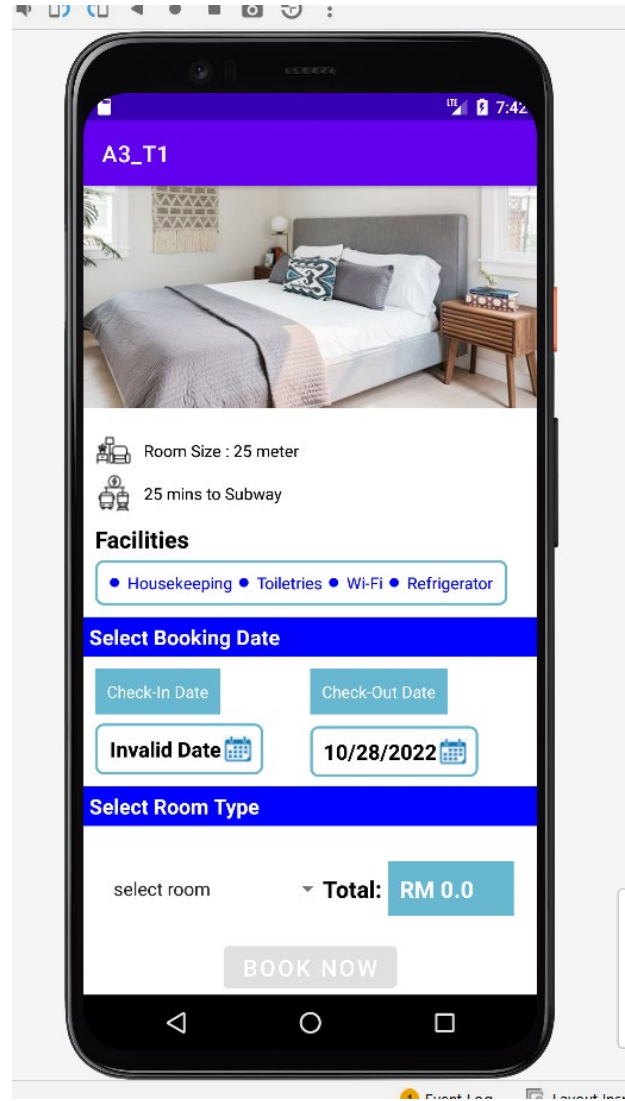
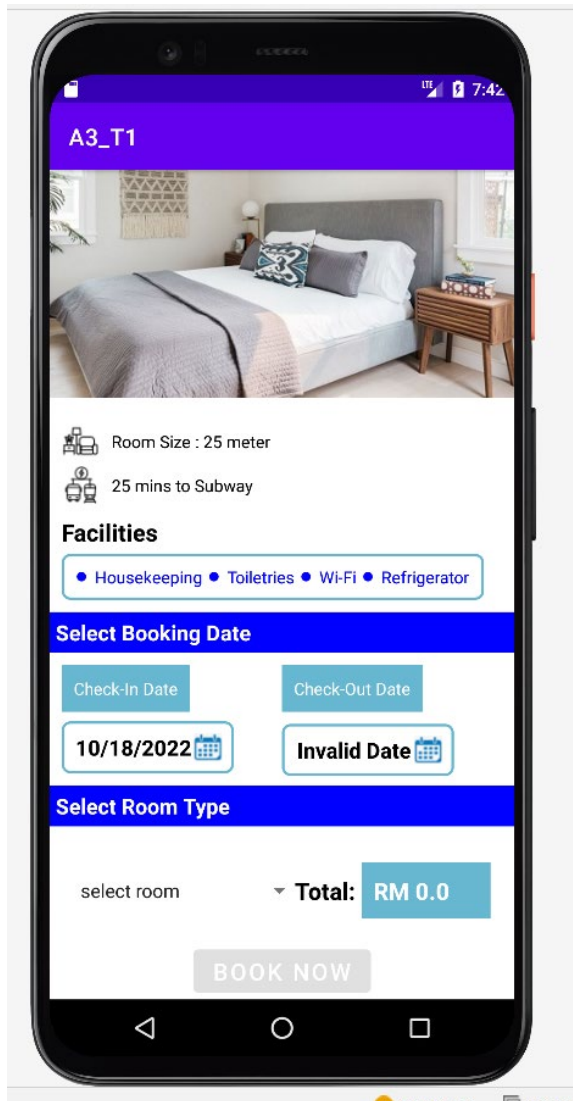
(1)	Alex	(4)	
(2)		(5)	
(3)		(6)	

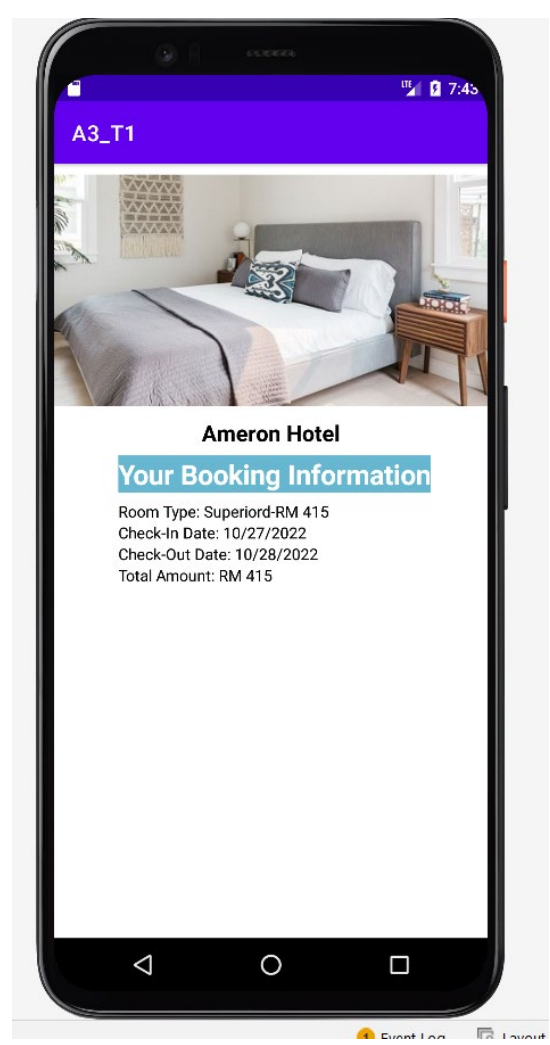
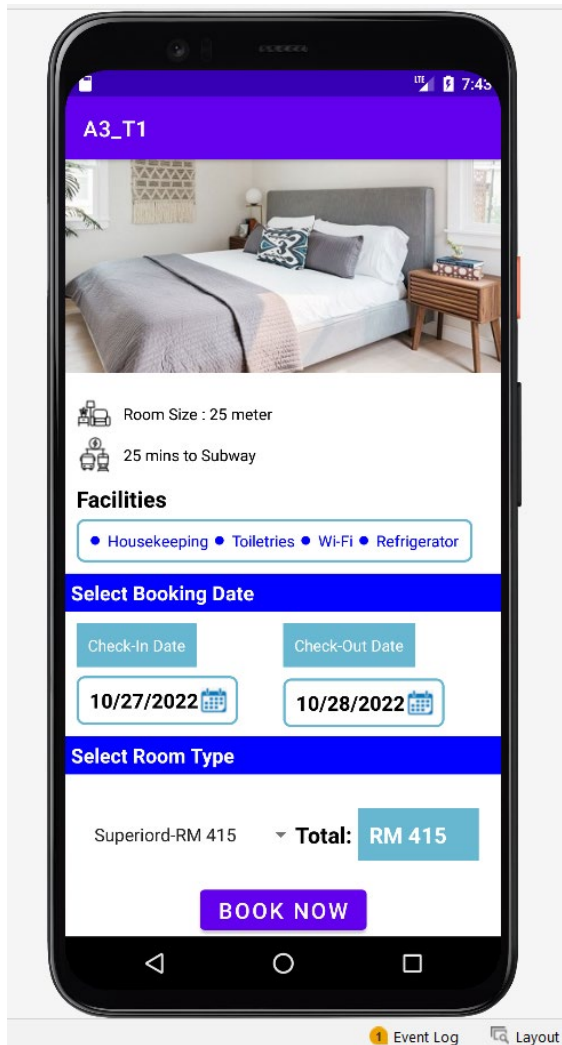
Further information relating to the penalties for plagiarism, which range from a formal caution to expulsion from the University is contained on the Current Students website at <https://www.swinburne.edu.my/current-students/manage-course/exams-results-assessment>

Copies of this form can be downloaded from the Student Forms web page at <https://www.swinburne.edu.my/current-students/manage-course/exams-results-assessment/how-to-submit-work.php>

## Output Task 1







## Source Code Task 1

### MainActivity.kt

```
package com.example.a3_t1

import android.content.Context
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.TextView
import android.widget.Toast
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import org.w3c.dom.Text

class MainActivity : AppCompatActivity() {
    var room = Room()
    lateinit var contacts: ArrayList<Contact>
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        createHostelRV(this)
    }

    fun createHostelRV(me: Context) {
        val rvContact = findViewById<View>(R.id.rv_hostel) as RecyclerView
        contacts = Contact.createContactList(room)
        val adapter = ContactsAdapter(contacts)
        rvContact.adapter = adapter
        rvContact.layoutManager =
            LinearLayoutManager(me, RecyclerView.VERTICAL, false)
    }
}
```

## Room.kt

```
package com.example.a3_t1

class Room {
    val name = listOf("Heritage Department", "Ameron Hotel", "Dorsett Studio Apartment", "Travelodge Apartment", "Clover Apartment", "Wonderloft Hostel")
    val heritageApartment = listOf(name.get(0), "Old Town Area", "5", "28 meter", "10 mins to city center", "Executive-RM 1200, Deluxe-RM 700," + "Superior-RM 500", "Housekeeping, Toiletries, Wi-Fi, Mini Bar", R.drawable.room1_icon, R.drawable.room1);
    val AmeronHotel = listOf(name.get(1), "Shenton Way, Down Town", "3", "25 meter", "25 mins to Subway", "Deluxe-RM 500, Superior-RM 415, Single-RM 300", "Housekeeping, Toiletries, Wi-Fi, Refrigerator", R.drawable.room2_icon, R.drawable.room2)
    val DorsettStudioApart = listOf(name.get(2), "City Center", "4", "28 meter", "5 mins to Bus Station", "Premier-RM 900, Deluxe-RM 600, Superior-RM 415", "Kitchenette, Toiletries, Wi-Fi, Refrigerator", R.drawable.room3_icon, R.drawable.room3)
    val TravelodgeApart = listOf(name.get(3), "Harbourfront", "3", "20 meter", "1.5km to City Center", "Family Room-RM 600, Deluxe-RM 400", "Breakfast, Toiletries, Wi-Fi, Refrigerator", R.drawable.room4_icon, R.drawable.room4)
    val CloverApart = listOf(name.get(4), "East-West Coast", "2", "19 meter", "10 km to City Center", "Deluxe-RM 450, Superior-RM 370, Single-RM 200", "Toiletries, Wi-Fi, Drinking Water", R.drawable.room5_icon, R.drawable.room5)
    val WonderloftHostel = listOf(name.get(5), "China Town", "3", "30 meter", "220 meter to public transportation", "Premium Room-RM 350, Dormitory Room-RM 160", "Wi-Fi, Shower, Laundry", R.drawable.room6_icon, R.drawable.room6)

    fun infoApartOrHotel(name:String): List<Any> {
        if(name == AmeronHotel.get(0)){
            return AmeronHotel
        }else if(name == DorsettStudioApart.get(0)){
            return DorsettStudioApart
        }else if(name == TravelodgeApart.get(0)){
            return TravelodgeApart
        }else if(name == CloverApart.get(0)) {
            return CloverApart
        }else if(name == heritageApartment.get(0)){
            return heritageApartment
        }
        return WonderloftHostel
    }

    fun HostelNameList():List<Any>{
        return name
    }
}
```

## ContactAdapter.kt

```
package com.example.a3_t1

import android.content.Intent
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.annotation.Nullable
import androidx.core.content.ContextCompat.startActivity
import androidx.recyclerview.widget.RecyclerView
import org.w3c.dom.Text

class ContactsAdapter(private val
mContacts:List<Contact>):RecyclerView.Adapter<ContactsAdapter.ViewHolder>()
{
    inner class ViewHolder(itemView:
View):RecyclerView.ViewHolder(itemView){
        val txt_rate:TextView = itemView.findViewById(R.id.txt_rate)
        val txt_name:TextView = itemView.findViewById(R.id.txt_name)
        val txt_area:TextView = itemView.findViewById(R.id.txt_area)
        val txt_price:TextView = itemView.findViewById(R.id.txt_price)
        val img_icon:ImageView = itemView.findViewById(R.id.img_icon)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
ViewHolder {
        val context = parent.context
        val inflater = LayoutInflater.from(context)
        val contactView =
inflater.inflate(R.layout.room_per_row,parent,false)
        return ViewHolder(contactView)
    }

    override fun onBindViewHolder(holder: ContactsAdapter.ViewHolder,
position: Int) {
        val contact:Contact = mContacts.get(position)
        holder.txt_name.setText(contact.name)
        holder.txt_area.setText(contact.area)
        holder.txt_price.setText(contact.price)
        holder.txt_rate.setText(contact.rate + " stars")
        holder.img_icon.setImageResource(contact.img)
        holder.img_icon.setOnClickListener {
            val intent:Intent =
Intent(holder.itemView.context,BookingForm::class.java)
            intent.putExtra("hostelName",contact.name)
            holder.itemView.context.startActivity(intent)
        }
    }

    override fun getItemCount(): Int {
        return mContacts.size
    }
}
```

## Contact.kt

```
package com.example.a3_t1

class Contact(val name:String, val rate:String, val price:String, val
area:String, val img:Int) {
    companion object{
        fun createContactList(room:Room):ArrayList<Contact>{
            val contacts = ArrayList<Contact>()
            for(i in 0..room.HostelNameList().size-1){
                val temp =
room.infoApartOrHotel(room.HostelNameList().get(i).toString())
                val temp_price_with_name =
temp.get(5).toString().split(",").toTypedArray()
                val temp_price_only =
temp_price_with_name.get(temp_price_with_name.size - 1).toString().split("-")
                .toTypedArray()
                contacts.add(Contact(
                    temp.get(0).toString(),
                    temp.get(2).toString(),
                    temp_price_only.get(temp_price_only.size-1).toString(),
                    temp.get(1).toString(),
                    temp.get(7).toString().toInt()
                ))
            }
            return contacts
        }
    }
}
```



## Booking.kt

```
package com.example.a3_t1

class Booking(val listOfRoomInfo:List<Any>) {
    var RoomType:String = "";
    var apartmentName:String = "";
    var CheckInDate:String = "";
    var CheckOutDate:String = "";
    var total:Int = 0;

    fun apartmentName():String{
        return apartmentName;
    }

    fun RoomType():String{
        return RoomType
    }

    fun CheckInDate():String{
        return CheckInDate
    }

    fun CheckOutDate():String{
        return CheckOutDate
    }

    fun total():String{
        return total.toString()
    }

    fun insertApartmentName(temp:String){
        apartmentName = temp
    }

    fun insertRoomType(temp:String){
        RoomType = temp
    }

    fun insertCheckInDate(temp:String){
        CheckInDate = temp
    }

    fun insertCheckOutDate(temp:String){
        CheckOutDate = temp
    }

    fun insertTotal(temp:Int){
        total = temp
    }

    fun roomPriceInfo():MutableList<String?>{
        var temp = listOfRoomInfo.get(5).toString()
        var temp_2 = temp.split(",").toTypedArray()
        var temp_list:MutableList<String?> = ArrayList()
        temp_list.add(0,"select room");
        for(i in 0 .. temp_2.size -1){
            temp_list.add(i+1,temp_2.get(i).toString());
        }
        return temp_list
    }
}
```

```

fun facilitiesInfo():List<Any>{
    var temp = listOfRoomInfo.get(6).toString()
    var temp_2 = temp.split(",").toTypedArray()
    var temp_list:List<Any> = listOfNotNull()
    for(i in 0..temp_2.size -1){
        temp_list += temp_2.get(i)
    }
    return temp_list
}

fun generateRoomPrice(priceWithName:String):Int{
    val temp = priceWithName.split("-").toTypedArray()
    val temp_1 = temp.get(1).split(" ").toTypedArray()
    val price:Int = temp_1.get(1).toInt()
    return price
}

fun roomImage():Int{
    return listOfRoomInfo.get(8).toString().toInt()
}

fun distanceInfo():String{
    return listOfRoomInfo.get(4).toString()
}

fun roomSize():String{
    return "Room Size : " + listOfRoomInfo.get(3).toString()
}
}

```

## BookingForm.kt

```
package com.example.a3_t1

import android.app.DatePickerDialog
import android.graphics.Typeface
import android.os.Bundle
import android.view.View
import android.view.ViewGroup.LayoutParams.WRAP_CONTENT
import android.widget.*
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.marginLeft
import androidx.core.view.setMargins
import java.text.SimpleDateFormat
import java.util.*

class BookingForm : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_booking_form)
        val hostelName = intent.getStringExtra("hostelName")
        val room = Room()
        val booking = Booking(room.infoApartOrHotel(hostelName.toString()))
        val txt_checkInDate:TextView = findViewById(R.id.txt_checkIn)
        val txt_checkOutDate:TextView = findViewById(R.id.txt_checkOut)
        val txt_in:TextView = findViewById(R.id.txt_ShowCheckInDate)
        val txt_out:TextView = findViewById(R.id.txt_ShowCheckOutDate)
        val txt_ttl:TextView = findViewById(R.id.txt_totalPrice)
        val cal = Calendar.getInstance()
        val spn_price:Spinner = findViewById(R.id.spn_RoomPrice)
        val btn_book:Button = findViewById(R.id.btn_book)

        btn_book!!.setOnClickListener(object : View.OnClickListener {
            override fun onClick(view: View) {
                booking.insertApartmentName(hostelName.toString())
                booking.insertCheckInDate(txt_in.text.toString())
                booking.insertCheckOutDate(txt_out.text.toString())
                booking.insertRoomType(spn_price.selectedItem.toString())
                val temp = txt_ttl.text.toString().split("
").toTypedArray()
                booking.insertTotal(temp.get(1).toString().toInt())
                ShowBookingInformation(booking)
            }
        })

        spn_price.onItemSelectedListener = object :
        AdapterView.OnItemSelectedListener {
            override fun onItemSelected(parent: AdapterView<*>?, view:
        View?, position: Int, id: Long) {
                UpdatePrice(txt_in,txt_out,booking,spn_price.selectedItem.toString())
            }

            override fun onNothingSelected(p0: AdapterView<*>?) {
            }
        }

        txt_checkInDate!!.setOnClickListener(object : View.OnClickListener
    {
```

```

        override fun onClick(view: View) {
            DatePickerDialog(this@BookingForm,
                dateListener(cal, "In"),
                // set DatePickerDialog to point to today's date when
it loads up
                cal.get(Calendar.YEAR),
                cal.get(Calendar.MONTH),
                cal.get(Calendar.DAY_OF_MONTH)).show()
                spn_price.setSelection(0)
                txt_ttl.setText("RM 0.0")
                btn_book.isEnabled = false
            }
        })

        txt_checkOutDate!!.setOnClickListener(object : View.OnClickListener
{
            override fun onClick(view: View) {
                DatePickerDialog(this@BookingForm,
                    dateListener(cal, "Out"),
                    // set DatePickerDialog to point to today's date when
it loads up
                    cal.get(Calendar.YEAR),
                    cal.get(Calendar.MONTH),
                    cal.get(Calendar.DAY_OF_MONTH)).show()
                    spn_price.setSelection(0)
                    txt_ttl.setText("RM 0.0")
                    btn_book.isEnabled = false
                }
            })

        FacilitiesShow(booking)
        RoomPrice(booking)
        roomImageSizeDistance(booking)
    }

    fun ShowBookingInformation(booking: Booking) {
        val bookinfo: LinearLayout = findViewById(R.id.ll_content)
        bookinfo.removeAllViews()

        val params: LinearLayout.LayoutParams =
LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
LinearLayout.LayoutParams.WRAP_CONTENT)
        params.setMargins(160, 20, 0, 20)

        val txt = TextView(this)
        txt.setText(booking.apartmentName)
        txt.setTypeface(Typeface.defaultFromStyle(Typeface.BOLD))
        txt.setTextSize(20f)
        txt.textAlignment = View.TEXT_ALIGNMENT_CENTER
        txt.setTextColor(resources.getColor(R.color.black))
        bookinfo.addView(txt)
        bookinfo.marginLeft
        val txt1 = TextView(this)
        txt1.setText("Your Booking Information")
        txt1.setTypeface(Typeface.defaultFromStyle(Typeface.BOLD))
        txt1.setTextSize(25f)
        txt1.setTextColor(resources.getColor(R.color.white))
        txt1.setBackgroundColor(resources.getColor(R.color.lightblue))
        bookinfo.addView(txt1, params)
    }

```

```

        val params_1: LinearLayout.LayoutParams =
            LinearLayout.LayoutParams(LinearLayout.LayoutParams.WRAP_CONTENT,
            LinearLayout.LayoutParams.WRAP_CONTENT)
            params_1.setMargins(160,0,0,0)

            val info:List<String> = listOf("Room Type: ", "Check-In Date: ",
            "Check-Out Date: ", "Total Amount: ")
            val info_data:List<String> =
            listOf(booking.RoomType(), booking.CheckInDate(), booking.CheckOutDate(), "RM
            " + booking.total().toString())
            for(i in 0..info.size -1){
                val temp = TextView(this)
                temp.setText(info.get(i) + info_data.get(i))
                temp.setTextColor(resources.getColor(R.color.black))
                bookinfo.addView(temp,params_1)
            }

        }

        fun dateListener(cal:Calendar,type:String):
        DatePickerDialog.OnDateSetListener {
            val dateSetListener = object : DatePickerDialog.OnDateSetListener {
                override fun onDateSet(view: DatePicker, year: Int,
                monthOfYear: Int,
                                dayOfMonth: Int) {
                    cal.set(Calendar.YEAR, year)
                    cal.set(Calendar.MONTH, monthOfYear)
                    cal.set(Calendar.DAY_OF_MONTH, dayOfMonth)
                    updateDateInView(cal,type)
                }
            }
            return dateSetListener
        }

        fun updateDateInView(cal:Calendar,type:String) {
            val myFormat = "MM/dd/yyyy" // mention the format you need
            val sdf = SimpleDateFormat(myFormat, Locale.US)
            val selectedDate:String = sdf.format(cal.getTime()).toString()
            val txt_in:TextView = findViewById(R.id.txt_ShowCheckInDate)
            val txt_out:TextView = findViewById(R.id.txt_ShowCheckOutDate)

            if(Type == "In"){
                if(checkDate(selectedDate,txt_out.text.toString()) == true){
                    txt_in.setText(selectedDate)
                }else{
                    txt_in.setText("Invalid Date")
                }
            }else if(Type == "Out"){
                if(checkDate(txt_in.text.toString(),selectedDate) == true){
                    txt_out.setText(selectedDate)
                }else {
                    txt_out.setText("Invalid Date")
                }
            }
        }

        fun checkDate(DateIn:String,DateOut:String):Boolean{
            val myFormat = "MM/dd/yyyy" // mention the format you need
            val sdf = SimpleDateFormat(myFormat, Locale.US)

```

```

        var flag:Boolean = false
        if(DateIn.isEmpty() || DateOut.isEmpty() || DateIn == "Invalid
Date" || DateOut == "Invalid Date"){
            flag = true
        }else{
            val dateIn = sdf.parse(DateIn.toString())
            val dateOut = sdf.parse(DateOut.toString())
            val check = dateIn.compareTo(dateOut) // >0 (1 > 2), < 0 (1 <
2) , = 0 (1==2)
            if(check > 0 )
                flag = false
            else if( check < 0)
                flag = true
            else if (check == 0)
                flag = true
        }
        return flag
    }
}

```

```

fun UpdatePrice(txt_in:TextView,txt_out:TextView,booking:
Booking,selectedRoom:String){
    val myFormat = "MM/dd/yyyy" // mention the format you need
    val sdf = SimpleDateFormat(myFormat, Locale.US)
    val btn_book:Button = findViewById(R.id.btn_book)
    if (txt_in.text.isNotEmpty() && txt_out.text.isNotEmpty() &&
txt_out.text != "Invalid Date" && txt_in.text != "Invalid Date" &&
selectedRoom != "select room"){
        val dateIn = sdf.parse(txt_in.text.toString())
        val dateOut = sdf.parse(txt_out.text.toString())
        val diff = kotlin.math.abs(dateOut.time - dateIn.time)
        val diffDay = diff / (24*60*60*1000)
        val txt_price:TextView = findViewById(R.id.txt_totalPrice)
        val x = (diffDay *
booking.generateRoomPrice(selectedRoom)).toString()
        txt_price.setText("RM " + x)
        btn_book.isEnabled = true
    }else{
        btn_book.isEnabled = false
    }
}
}

```

```

fun roomImageSizeDistance(booking: Booking){
    val txt_distance:TextView = findViewById(R.id.txt_distance)
    val txt_roomSize:TextView = findViewById(R.id.txt_roomsize)
    val img_room:ImageView = findViewById(R.id.img_room)
    img_room.setImageResource(booking.roomImage())
    txt_distance.setText(booking.distanceInfo())
    txt_roomSize.setText(booking.roomSize())
}

```

```

fun RoomPrice(booking:Booking){
    val spin_price:Spinner = findViewById(R.id.spn_RoomPrice)
    val x:MutableList<String?> = booking.roomPriceInfo()
    val arrayAdapter:ArrayAdapter<String?> =
ArrayAdapter<String?>(this,android.R.layout.simple_list_item_1,x)

arrayAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdo
wn_item)
    spin_price.adapter = arrayAdapter
}

```

```

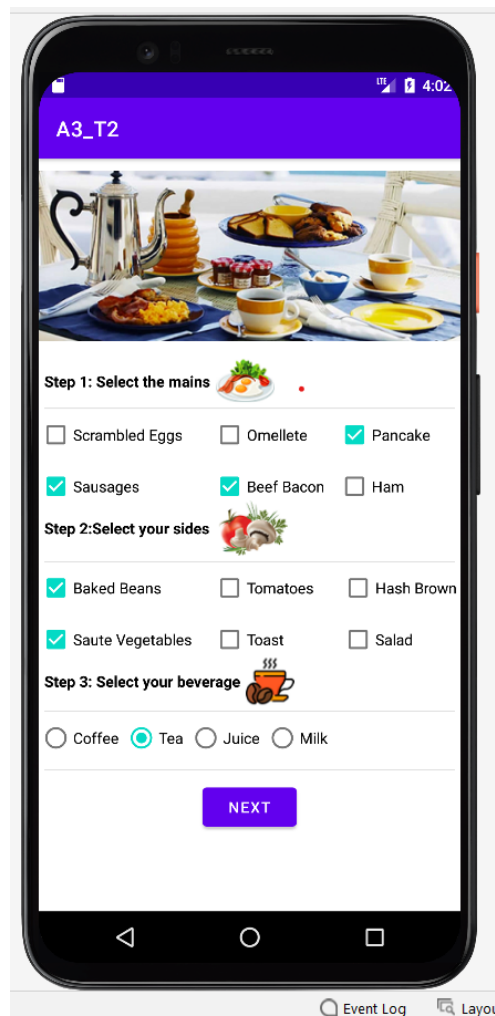
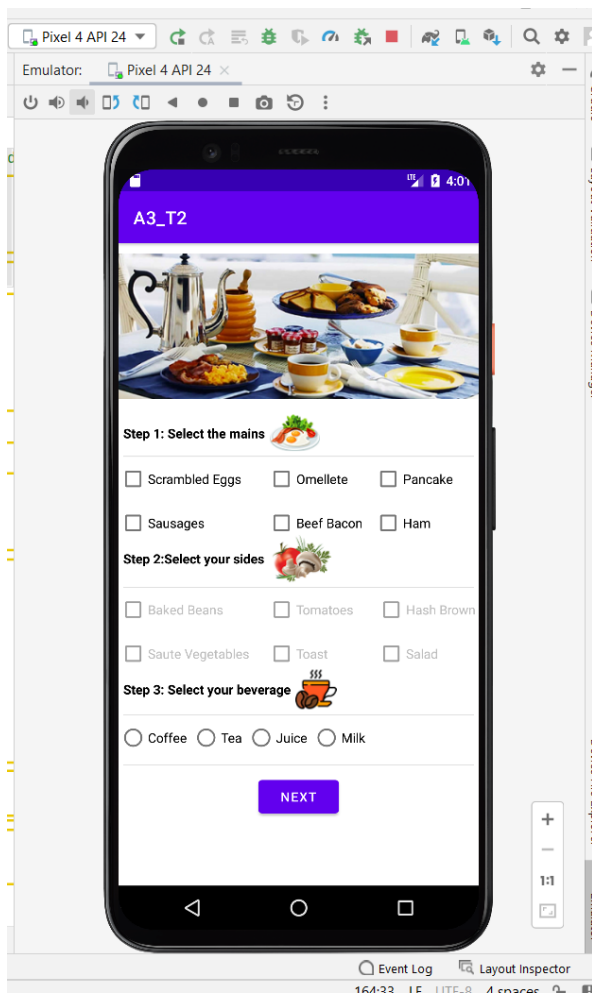
    }

    fun FacilitiesShow(booking:Booking) {
        val txt_fac_item:LinearLayout =
            findViewById(R.id.txt_facilities_item)
        val facilitiesinfo:List<Any> = booking.facilitiesInfo()
        for(i in 0..facilitiesinfo.size-1){
            val txt = TextView(this)

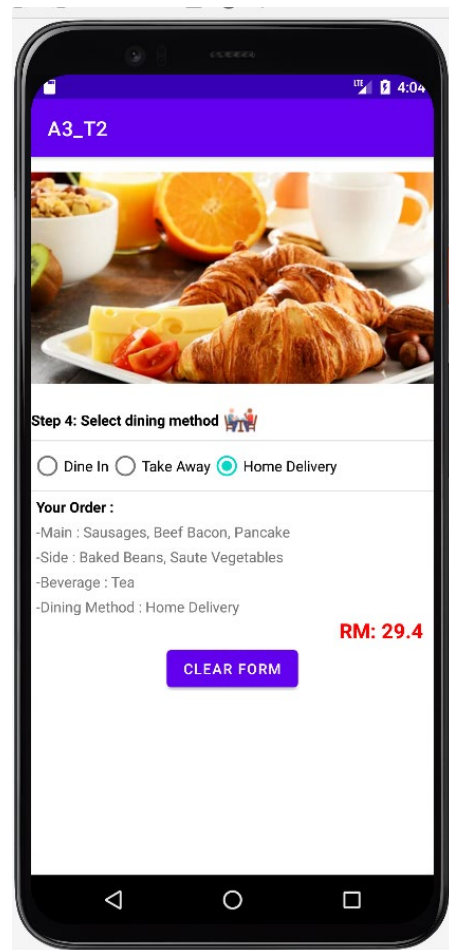
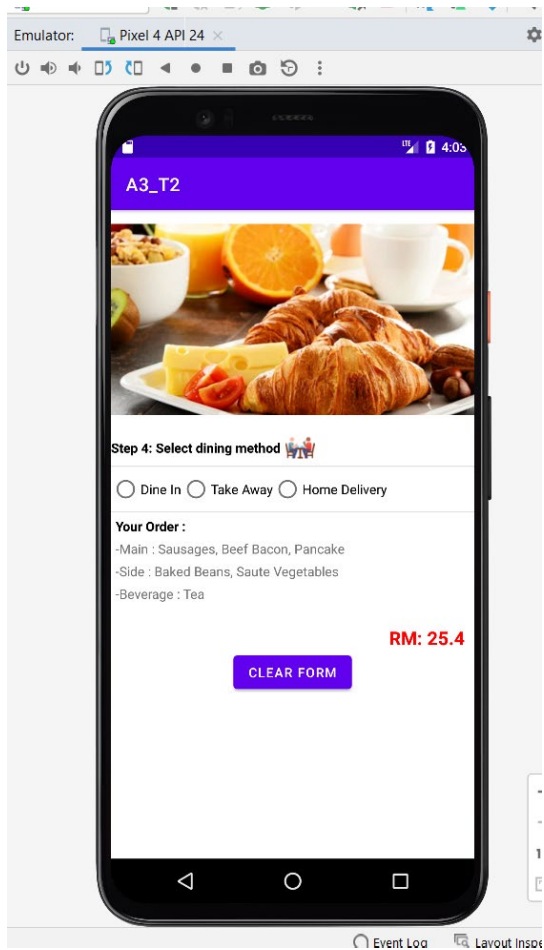
            txt.setCompoundDrawablesWithIntrinsicBounds(R.drawable.bullet,0,0,0)
            txt.setCompoundDrawablePadding(20)
            txt.setText(facilitiesinfo.get(i).toString())
            txt.setTextColor(resources.getColor(R.color.blue))
            txt.setPadding(8,0,8,0)
            txt_fac_item.addView(txt)
        }
    }
}

```

## Assignment 3 Task 2 Output







## Assignment 3 Task 2 Source Code (.kt)

### MainActivity.kt

```
package com.example.a3_t2

import android.content.DialogInterface
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.CheckBox
import android.widget.RadioButton
import android.widget.TextView
import androidx.appcompat.app.AlertDialog
import androidx.fragment.app.Fragment
import androidx.fragment.app.FragmentTransaction

class MainActivity : AppCompatActivity() {
    val first = order1()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        val ft: FragmentTransaction =
supportFragmentManager.beginTransaction()
        ft.add(R.id.fl, first).addToBackStack("ok").commit()
    }

    fun
PassData(main:String,side:String,Beverage:String,cost:String,fragment:
Fragment){
        val bundle = Bundle()
        bundle.putString("main",main)
        bundle.putString("side",side)
        bundle.putString("beverage",Beverage)
        bundle.putString("cost",cost)
        SwitchPage(fragment,bundle)
    }

    fun SwitchPage(fragment:Fragment,bundle: Bundle){
        val ft: FragmentTransaction =
supportFragmentManager.beginTransaction()
        ft.hide(first)
        fragment.arguments = bundle
        ft.add(R.id.fl,fragment).addToBackStack("ok").commit()
    }
}
```

## order1.kt

```
package com.example.a3_t2

import android.content.DialogInterface
import android.content.Intent
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Button
import android.widget.CheckBox
import android.widget.RadioButton
import android.widget.TextView
import androidx.fragment.app.Fragment
import java.text.DecimalFormat

class order1: Fragment() {
    var totalPrice:Double = 0.00;
    var totalMains:Int = 0;
    var price:Double = 0.00;
    var discount:Double = 0.00;
    var txtPrice : String = ""
    var txtSummary: String = ""
    var txtSide: String = ""
    var txtBeverage: String = ""
    private val main = mutableMapOf<Int,Double>();
    private val sideCost = mutableMapOf<Int,Double>();
    private val beverageCost = mutableMapOf<Int,Double>();
    private val cbBreakfastId:Map<Int,Double> = mapOf(R.id.cb_beefbacon to
6.50,R.id.cb_eggs to 5.00,R.id.cb_omellete to 5.00,
        R.id.cb_pancake to 4.00,R.id.cb_ham to 6.00,R.id.cb_sausage to
6.50);
    private val cbSideId:Map<Int,Double> = mapOf(R.id.cb_beans to
2.50,R.id.cb_tomato to 1.50,R.id.cb_hashbrown to 2.00,
        R.id.cb_sauteVege to 3.50,R.id.cb_toast to 2.00, R.id.cb_salad to
3.00);
    private val cvSideVegeID:Map<Int,Double> = mapOf(R.id.cb_beans to
2.50,R.id.cb_hashbrown to 2.00,R.id.cb_sauteVege to 3.50,
        R.id.cb_salad to 3.00);
    private val rbBeverageId:Map<Int,Double> = mapOf(R.id.rd_coffee to
4.00,R.id.rd_juice to 5.50, R.id.rd_tea to 4.00,
        R.id.rd_milk to 5.00);
    private var checkPoint:Boolean = true;
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        var view: View = inflater.inflate(R.layout.order1, container,
false)

        return view
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
    }
}
```

```

        //val btnReset: Button = view.findViewById(R.id.btn_reset);
        //val btnOrder: Button = view.findViewById(R.id.btn_order);
        val btn_next: Button = view.findViewById(R.id.btn_next)
        btn_next.setOnClickListener() {
            (context as
MainActivity).PassData(txtSummary,txtSide,txtBeverage,txtPrice,order2())
        }

        listenOfMainCB(view)
        listenOfSidesCB(view)
        listenOfTeaRB(view)
    }

    fun listenOfTeaRB(view: View) {
        val cof: RadioButton = view.findViewById(R.id.rd_coffee)
        val tea: RadioButton = view.findViewById(R.id.rd_tea)
        val juice: RadioButton = view.findViewById(R.id.rd_juice)
        val milk: RadioButton = view.findViewById(R.id.rd_milk)
        cof.setOnClickListener() {
            selectBeverage(view, cof)
        }
        tea.setOnClickListener() {
            selectBeverage(view, tea)
        }
        juice.setOnClickListener() {
            selectBeverage(view, juice)
        }
        milk.setOnClickListener() {
            selectBeverage(view, milk)
        }
    }

}

    fun listenOfMainCB(view: View) {
        val egg: CheckBox = view.findViewById(R.id.cb_eggs)
        val ome: CheckBox = view.findViewById(R.id.cb_omellete)
        val pan: CheckBox = view.findViewById(R.id.cb_pancake)
        val sau: CheckBox = view.findViewById(R.id.cb_sausage)
        val beef: CheckBox = view.findViewById(R.id.cb_beefbacon)
        val ham: CheckBox = view.findViewById(R.id.cb_ham)
        egg.setOnClickListener() {
            checkedBreakfast(view, egg)
        }
        ome.setOnClickListener() {
            checkedBreakfast(view, ome)
        }
        pan.setOnClickListener() {
            checkedBreakfast(view, pan)
        }
        sau.setOnClickListener() {
            checkedBreakfast(view, sau)
        }
        beef.setOnClickListener() {
            checkedBreakfast(view, beef)
        }
        ham.setOnClickListener() {
            checkedBreakfast(view, ham)
        }
    }
}

```

```

fun listenOfSidesCB(view: View){
    val beans:CheckBox = view.findViewById(R.id.cb_beans)
    val toma:CheckBox = view.findViewById(R.id.cb_tomato)
    val hash:CheckBox = view.findViewById(R.id.cb_hashbrown)
    val vege:CheckBox = view.findViewById(R.id.cb_sauteVege)
    val toast:CheckBox = view.findViewById(R.id.cb_toast)
    val salad:CheckBox = view.findViewById(R.id.cb_salad)
    beans.setOnClickListener(){
        checkedSides(view,beans)
    }
    toma.setOnClickListener(){
        checkedSides(view,toma)
    }
    hash.setOnClickListener(){
        checkedSides(view,hash)
    }
    vege.setOnClickListener(){
        checkedSides(view,vege)
    }
    toast.setOnClickListener(){
        checkedSides(view,toast)
    }
    salad.setOnClickListener(){
        checkedSides(view,salad)
    }
}

fun
printReceipe(title:String,order:MutableMap<Int,Double>,ori:MutableMap<Int,Double>,view:View):String{
    var message:String = title;
    for((key,value) in order){
        var temp: CheckBox = view.findViewById(key);
        message += "      " + temp.text.toString() + " - RM
"+ori[key]+"\\n";
        discount += (ori[key]!! - order[key]!!);
    }
    return message;
}

fun checkedBreakfast(view: View,cb:CheckBox){

    var temp:String = "-Main : ";
    if(cb is CheckBox){
        val checked:Boolean = cb.isChecked;
        for((key,value) in cbBreakfastId){
            if(cb.id == key){
                if(checked){
                    totalPrice += value;
                    totalMains += 1;
                    main[key] = value;
                }else{
                    totalPrice -= value;
                    totalMains -= 1;
                    main.remove(key);
                }
            }
        }
    }
    var n:Int = 0;

```

```

for((key,value ) in main){
    n++;
    val x: CheckBox = view.findViewById(key);
    temp += x.text;
    if(n < main.size){
        temp += ", ";
    }

}

if(totalMains > 0 && checkPoint){
    checkPoint = false;
    enableDisableSide(true,view)
}else if(!checkPoint && totalMains == 0){
    temp = "";
    checkPoint = true;
    enableDisableSide(false,view);
    sideCost.clear();
}

checkDiscountBeverage()
updatePrice();
val df = DecimalFormat("#.##")
txtPrice = df.format(totalPrice).toString();
txtSummary = temp;
}

private fun updatePrice(){
    val df = DecimalFormat("#.##")
    for((key,value) in sideCost){
        var x:Boolean = cvSideVegeID.containsKey(key);
        if(x){ // exist in choosed side list
            if(totalMains >= 3){ // true means main choose equal or
more than 3
                if(cvSideVegeID[key] == value){ // if both price equal
                    totalPrice -= value;
                    totalPrice += df.format(value*0.8).toDouble();
                    sideCost[key] = df.format(value*0.8).toDouble();
                }
            }else{
                if(cvSideVegeID[key] != value){
                    totalPrice -= value;
                    totalPrice += cvSideVegeID[key]!!;
                    sideCost[key] = cvSideVegeID[key]!!;
                }
            }
        }
    }
}

fun checkDiscountBeverage(){
    totalPrice -= price
    val df = DecimalFormat("#.##")
    if(totalPrice >= 20){
        for((key,value) in beverageCost){
            price = df.format(value*0.9).toDouble()
            totalPrice += price
        }
    }else{
        for((key,value) in beverageCost){
            price = value.toDouble()
            totalPrice += price
        }
    }
}

```

```

    }
}

fun checkedSides(view: View, cb: CheckBox) {
    val df = DecimalFormat("#.##")
    if (cb is CheckBox) {
        val checked: Boolean = cb.isChecked;
        val exist: Boolean = sideCost.containsKey(cb.id);
        if (exist) {
            totalPrice -= sideCost[cb.id]!!;
            sideCost.remove(cb.id);
        } else {
            for ((ley, value) in cbSideId) {
                if (cb.id == ley) {
                    if (checked) {
                        if (totalMains >= 3 && cb.id != R.id.cb_toast &&
cb.id != R.id.cb_tomato) {
                            totalPrice +=
df.format(value*0.8).toDouble();
                            sideCost[ley] =
df.format(value*0.8).toDouble();
                        } else {
                            totalPrice += value;
                            sideCost[ley] = value;
                        }
                    }
                }
            }
        }
    }
    var temp: String = "-Side : ";
    var n: Int = 0;
    for ((key, value) in sideCost) {
        n++;
        var x = view.findViewById<TextView>(key);
        temp += x.text;
        if (n != sideCost.size) {
            temp += ", ";
        }
    }
    txtSide = temp;
    txtPrice = totalPrice.toString();
}

private fun enableDisableSide(temp: Boolean, view: View) {
    for ((key, value) in cbSideId) {
        var x: CheckBox = view.findViewById(key);
        if (temp) {
            x.isEnabled = true;
        } else {
            x.isChecked = false;
            x.isEnabled = false;
            totalPrice = price;
        }
    }
    txtSide = "";
}

fun selectBeverage(view: View, cb: RadioButton) {
    var x: Int = 0;

```

```

if (cb is RadioButton) {
    var checked: Boolean = cb.isChecked;
    for ((key, value) in rbBeverageId) {
        if (cb.id == key) {
            if (checked) {
                if (price == 0.0) {
                    price = discountBeverage(value);
                } else {
                    totalPrice -= price;
                    price = discountBeverage(value);
                }
                x = key;
                totalPrice += price;
            }
        }
    }
}

val rd: RadioButton = view.findViewById(x);
txtBeverage = "-Beverage : " + rd.text;
txtPrice = String.format("%.2f", totalPrice).toDouble().toString()
beverageCost.clear();
beverageCost[x] = rbBeverageId[x]!!;
}

fun discountBeverage(value: Double): Double {
    var x: Double = value
    val df = DecimalFormat("#.##")
    if (totalPrice >= 20) {
        x = x * 0.9
    }
    return String.format("%.2f", x).toDouble()
}
}

```



## order2.kt

```
package com.example.a3_t2

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.RadioButton
import android.widget.TextView
import androidx.fragment.app.Fragment

class order2 : Fragment() {
    var main:String = ""
    var side:String = ""
    var beverage:String = ""
    var cost:String = ""
    var DiningMethodCost:Double = 0.00
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        var view: View = inflater.inflate(R.layout.order2, container,
false)
        return view
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        main = arguments?.getString("main").toString()
        side = arguments?.getString("side").toString()
        beverage = arguments?.getString("beverage").toString()
        cost = arguments?.getString("cost").toString()
        val txt_main:TextView = view.findViewById(R.id.txt_summary)
        val txt_side:TextView = view.findViewById(R.id.txt_side)
        val txt_beverage:TextView = view.findViewById(R.id.txt_beverage)
        val txt_cost:TextView = view.findViewById(R.id.txt_cost)
        txt_main.setText(main)
        txt_side.setText(side)
        txt_beverage.setText(beverage)
        txt_cost.setText(cost)
        listenOfDiningMethodRB(view)
    }

    fun listenOfDiningMethodRB(view:View){
        var temp_cost:Double = 0.00
        val rb_dinein:RadioButton = view.findViewById(R.id.rb_dinein)
        val rb_takeaway:RadioButton = view.findViewById(R.id.rb_takeaway)
        val rb_delivery:RadioButton = view.findViewById(R.id.rb_delivery)
        rb_dinein.setOnClickListener(){
            DeliveryMethodPriceAddMinus(view,0.00,"Dine In")
        }
        rb_takeaway.setOnClickListener(){
            DeliveryMethodPriceAddMinus(view,1.50,"Take Away")
        }
        rb_delivery.setOnClickListener(){
```

```

        DeliveryMethodPriceAddMinus(view,4.00,"Home Delivery")
    }
}

fun
DeliveryMethodPriceAddMinus(view:View,price:Double,DineMethod:String){
    var temp_cost:Double = cost.toDouble()
    if(DiningMethodCost == 0.00){
        DiningMethodCost = price
    }else{
        temp_cost = cost.toDouble()
        temp_cost -= DiningMethodCost
        DiningMethodCost = price
    }
    temp_cost += price
    cost = temp_cost.toString()
    val txt_cost:TextView = view.findViewById(R.id.txt_cost)
    txt_cost.setText(cost)
    val txt_dine:TextView = view.findViewById(R.id.txt_dineMethod)
    txt_dine.setText("-Dining Method : " + DineMethod)
}

}

```

## Assignment 3 Task 2 Source Code (.xml)

### MainActivity.xml

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

    <FrameLayout
        android:id="@+id/fl"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

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

## Order1.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">
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <ImageView
                android:id="@+id/img_banner"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent"
                app:srcCompat="@drawable/banner2" />

            <TextView
                android:id="@+id/txt_step1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="5dp"
                android:text="Step 1: Select the mains"
                android:textColor="@color/black"
                android:textStyle="bold"
                app:layout_constraintBottom_toBottomOf="@+id/img_breakfast"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="@+id/img_breakfast" />

            <ImageView
                android:id="@+id/img_breakfast"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="5dp"
                app:layout_constraintStart_toEndOf="@+id/txt_step1"
                app:layout_constraintTop_toBottomOf="@+id/img_banner"
                app:srcCompat="@drawable/breakfast_icon" />

            <View
                android:id="@+id/divider_1"
                android:layout_width="match_parent"
                android:layout_height="1dp"
                android:layout_margin="5dp"
                android:background="?android:attr/listDivider"
                app:layout_constraintTop_toBottomOf="@+id/img_breakfast" />

            <CheckBox
                android:id="@+id/cb_eggs"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Scrambled Eggs"
                app:layout_constraintStart_toStartOf="parent"
```

```

        app:layout_constraintTop_toBottomOf="@+id/divider_1" />

<CheckBox
    android:id="@+id/cb_omellete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Omellete"
    app:layout_constraintEnd_toStartOf="@+id/cb_pancake"
    app:layout_constraintStart_toEndOf="@+id/cb_eggs"
    app:layout_constraintTop_toTopOf="@+id/cb_eggs" />

<CheckBox
    android:id="@+id/cb_pancake"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Pancake"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/cb_omellete"
    app:layout_constraintTop_toTopOf="@+id/cb_omellete" />

<CheckBox
    android:id="@+id/cb_sausage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sausages"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/cb_eggs" />

<CheckBox
    android:id="@+id/cb_beefbacon"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Beef Bacon"
    app:layout_constraintStart_toStartOf="@+id/cb_omellete"
    app:layout_constraintTop_toBottomOf="@+id/cb_omellete" />

<CheckBox
    android:id="@+id/cb_ham"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ham"
    app:layout_constraintStart_toStartOf="@+id/cb_pancake"
    app:layout_constraintTop_toBottomOf="@+id/cb_pancake" />

<TextView
    android:id="@+id/txt_step2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="Step 2:Select your sides "
    android:textColor="@color/black"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/cb_sausage" />

<ImageView
    android:id="@+id/img_side"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    app:layout_constraintBottom_toBottomOf="@id/txt_step2"

```

```

        app:layout_constraintStart_toEndOf="@id/txt_step2"
        app:layout_constraintTop_toTopOf="@+id/txt_step2"
        app:srcCompat="@drawable/side_icon" />

<View
    android:id="@+id/divider2"
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:layout_margin="5dp"
    android:background="?android:attr/listDivider"
    app:layout_constraintTop_toBottomOf="@+id/img_side" />

<CheckBox
    android:id="@+id/cb_beans"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:text="Baked Beans"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/divider2" />

<CheckBox
    android:id="@+id/cb_tomato"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:text="Tomatoes"
    app:layout_constraintBottom_toBottomOf="@id/cb_beans"
    app:layout_constraintStart_toStartOf="@+id/cb_beefbacon"
    app:layout_constraintTop_toTopOf="@id/cb_beans" />

<CheckBox
    android:id="@+id/cb_hashbrown"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:text="Hash Brown"
    app:layout_constraintBottom_toBottomOf="@id/cb_tomato"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="@id/cb_ham"
    app:layout_constraintTop_toTopOf="@id/cb_tomato" />

<CheckBox
    android:id="@+id/cb_sauteVege"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:text="Saute Vegetables"
    app:layout_constraintStart_toStartOf="@id/cb_beans"
    app:layout_constraintTop_toBottomOf="@id/cb_beans" />

<CheckBox
    android:id="@+id/cb_toast"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:enabled="false"
    android:text="Toast"
    app:layout_constraintStart_toStartOf="@id/cb_tomato"
    app:layout_constraintTop_toBottomOf="@id/cb_tomato" />

<CheckBox

```

```

        android:id="@+id/cb_salad"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:enabled="false"
        android:text="Salad"
        app:layout_constraintStart_toStartOf="@id/cb_hashbrown"
        app:layout_constraintTop_toBottomOf="@id/cb_hashbrown" />

<TextView
    android:id="@+id/txt_step3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="Step 3: Select your beverage"
    android:textColor="@color/black"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/cb_sauteVege" />

<ImageView
    android:id="@+id/img_coffee"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    app:layout_constraintBottom_toBottomOf="@id/txt_step3"
    app:layout_constraintStart_toEndOf="@id/txt_step3"
    app:layout_constraintTop_toTopOf="@id/txt_step3"
    app:srcCompat="@drawable/coffee_icon" />

<View
    android:id="@+id/divider3"
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:layout_margin="5dp"
    android:background="?android:attr/listDivider"

    app:layout_constraintTop_toBottomOf="@id/img_coffee" />

<RadioGroup
    android:id="@+id/radioGroup1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/divider3">

    <RadioButton
        android:id="@+id/rd_coffee"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="5dp"
        android:text="Coffee" />

    <RadioButton
        android:id="@+id/rd_tea"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="5dp"
        android:text="Tea" />

    <RadioButton

```

```

        android:id="@+id/rd_juice"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="5dp"
        android:layout_weight="1"
        android:text="Juice" />

        <RadioButton
            android:id="@+id/rd_milk"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Milk" />
    </RadioGroup>

    <View
        android:id="@+id/divider4"
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:layout_margin="5dp"
        android:background="?android:attr/listDivider"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/radioGroup1" />

    <Button
        android:id="@+id/btn_next"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toBottomOf="@+id/divider4"
        android:layout_marginTop="10dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:text="NEXT"/>
    </androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>

```



## Order2.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"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/img_main2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:srcCompat="@drawable/main2" />

    <TextView
        android:id="@+id/txt_step4"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/img_main2"
        android:layout_marginTop="10dp"
        android:textStyle="bold"
        android:textColor="@color/black"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Step 4: Select dining method" />

    <ImageView
        android:id="@+id/img_res_icon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="@id/txt_step4"
        app:layout_constraintBottom_toBottomOf="@id/txt_step4"
        app:layout_constraintStart_toEndOf="@id/txt_step4"
        android:layout_marginLeft="5dp"
        app:srcCompat="@drawable/restaurant_icon" />

    <View
        android:id="@+id/divider1"
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="?android:attr/listDivider"
        app:layout_constraintStart_toStartOf="parent"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp"
        app:layout_constraintTop_toBottomOf="@id/txt_step4" />

    <RadioGroup
        android:id="@+id/rg1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/divider1">

        <RadioButton
            android:id="@+id/rb_dinein"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColor="@color/black"
```

```

        android:text="Dine In" />

<RadioButton
    android:id="@+id/rb_takeaway"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/black"
    android:text="Take Away" />

<RadioButton
    android:id="@+id/rb_delivery"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/black"
    android:text="Home Delivery" />
</RadioGroup>

<View
    android:id="@+id/divider2"
    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="?android:attr/listDivider"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_marginBottom="10dp"
    app:layout_constraintTop_toBottomOf="@id/rg1" />

<TextView
    android:id="@+id/txt_order"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text="Your Order : "
    android:textColor="@color/black"
    android:layout_marginLeft="10dp"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/divider2" />

<TextView
    android:id="@+id/txt_summary"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:text=""
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txt_order" />

<TextView
    android:id="@+id/txt_side"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txt_summary"
    android:text="" />

<TextView
    android:id="@+id/txt_beverage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"

```

```

        app:layout_constraintTop_toBottomOf="@id/txt_side"
        app:layout_constraintStart_toStartOf="parent"
        android:text="" />

<TextView
    android:id="@+id/txt_dineMethod"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@id/txt_beverage"
    android:layout_margin="5dp"
    app:layout_constraintStart_toStartOf="parent"
    android:text="" />

<TextView
    android:id="@+id/txt_rm"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="RM: "
    android:textColor="@color/red"
    android:textSize="20dp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@id/txt_cost"
    app:layout_constraintEnd_toStartOf="@id/txt_cost"
    app:layout_constraintTop_toTopOf="@id/txt_cost" />

<TextView
    android:id="@+id/txt_cost"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="10dp"
    android:text="0.00"
    android:textColor="@color/red"
    android:textSize="20dp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/txt_dineMethod" />

<Button
    android:id="@+id/button"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@id/txt_cost"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="CLEAR FORM" />

</androidx.constraintlayout.widget.ConstraintLayout>

```