Assessment Cover Sheet

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

SWINBURNE UNIVERSITY OF TECHNOLOGY

ASSESSMENT DETAILS

Unit title	Software Development For Mobile Devices	Tutorial /Lab Group	1
Unit code	COS30017	Due date	21 Dec 2022
Name of lecturer/tutor	Dr Marlene Lu		
Assignment title	Assignment 5		

Office use only

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.

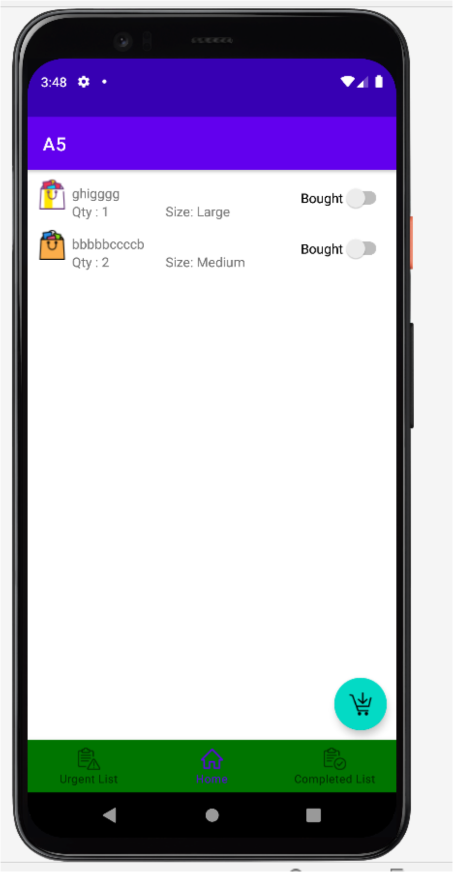
Student signature/s

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**

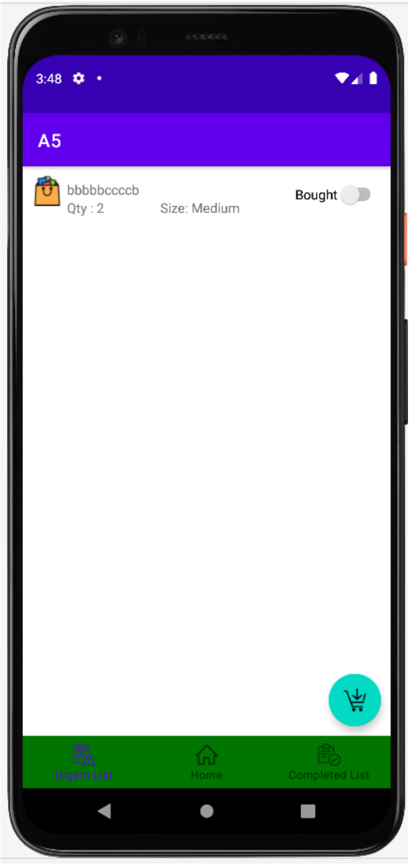


3:48

A5

ghigggg
Qty : 1 Size: Large Bought

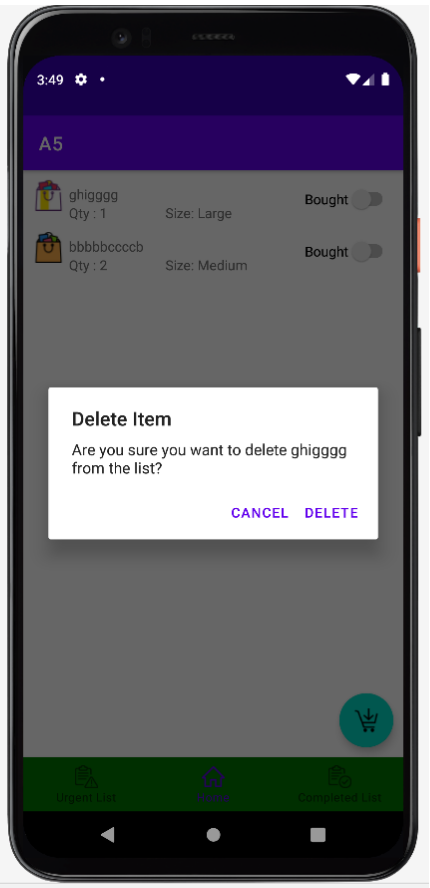
bbbbbbccccb
Qty : 2 Size: Medium Bought



3:48

A5

bbbbbbccccb
Qty : 2 Size: Medium Bought



3:49

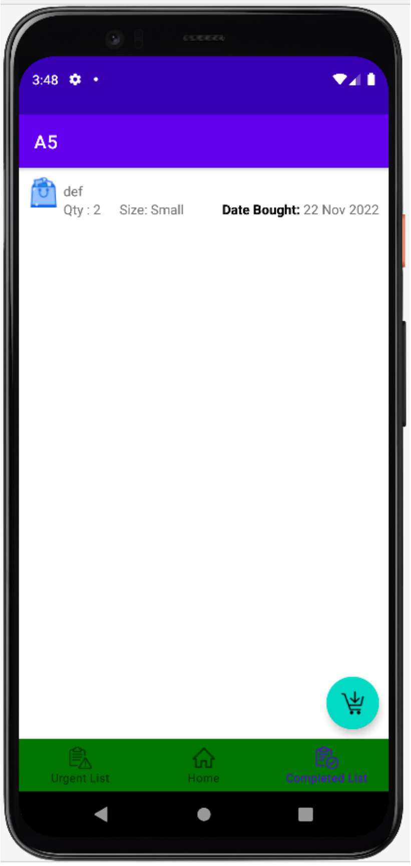
A5

ghigggg
Qty : 1 Size: Large Bought

bbbbbbccccb
Qty : 2 Size: Medium Bought

Delete Item

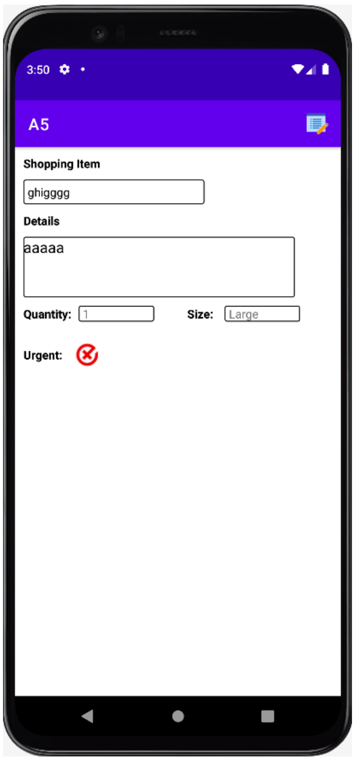
Are you sure you want to delete ghigggg from the list?



3:48

A5

def
Qty : 2 Size: Small **Date Bought:** 22 Nov 2022

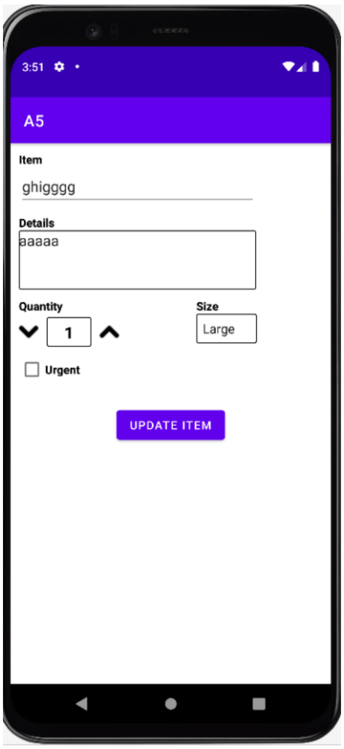


Shopping Item
ghigggg

Details
aaaaa

Quantity: 1 Size: Large

Urgent: ☒



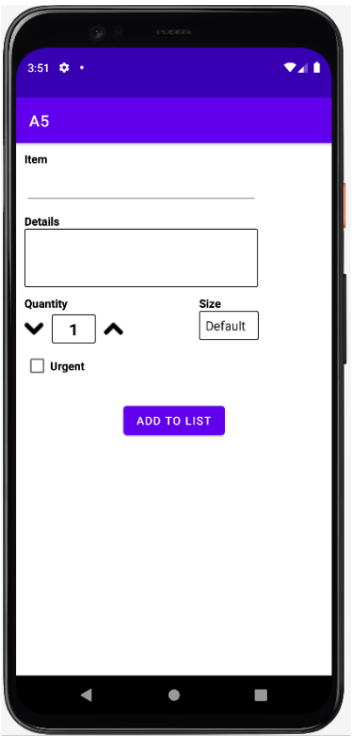
Item
ghigggg

Details
aaaaa

Quantity
1

Size
Large

□ Urgent



Item

Details

Quantity
1

Size
Default

□ Urgent

MainActiviy

```
package com.example.a5

import android.content.Intent
import android.os.Bundle
import android.view.View
import androidx.appcompat.app.AppCompatActivity
import androidx.fragment.app.Fragment
import com.google.android.material.bottomnavigation.BottomNavigationView


class MainActivity : AppCompatActivity() {
    lateinit var bottomNav: BottomNavigationView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        loadFragment(HomeFragment())
        bottomNav = findViewById(R.id.mainBNV) as BottomNavigationView
        bottomNav.setOnItemSelectedListener {item->
            when(item.itemId){
                R.id.btn_complete -> loadFragment(CompleteFragment())
                R.id.btn_home -> loadFragment(HomeFragment())
                R.id.btn_urgent -> loadFragment(UrgentFragment())
            }
            true
        }
        val fab: View = findViewById(R.id.btn_float)
        fab.setOnClickListener { view ->
            val intent  = Intent(this,ShoppingCart::class.java)
            //val intent  = Intent(this,ViewActivity::class.java)
            startActivity(intent)
        }
    }

    fun refreshFragment(fragment: Fragment){
        val transaction = supportFragmentManager.beginTransaction()
        transaction.detach(fragment).attach(fragment).commit()
    }

    fun loadFragment(fragment: Fragment){
        val transaction = supportFragmentManager.beginTransaction()
        transaction.replace(R.id.mainFrame,fragment)
        transaction.addToBackStack(null)
        transaction.commit()
    }
}
```

MyDB

```
package com.example.a5

import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
import android.os.Build
import android.util.Log
import androidx.annotation.RequiresApi
import java.time.LocalDateTime
import java.time.format.DateTimeFormatter

class MyDB(context: Context):
    SQLiteOpenHelper(context, MyDB.DATABASE_NAME,null,
MyDB.DATABASE_VERSION) {
    override fun onCreate(p0: SQLiteDatabase?) {
        val sql = "CREATE TABLE IF NOT EXISTS " + tbName + " (" + id + "
INTEGER PRIMARY KEY, " +
                item + " TEXT," +
                details + " TEXT," +
                quantity + " TEXT," +
                size + " TEXT," +
                urgent + " TEXT," +
                buy + " TEXT," +
                purchasedate + " TEXT" + ")"
        p0?.execSQL(sql)
    }

    override fun onUpgrade(p0: SQLiteDatabase?, p1: Int, p2: Int) {
        p0!!.execSQL("DROP TABLE IF EXISTS " + tbName)
        onCreate(p0)
    }

    fun
updateData(id:Int,name:String,info:String,num:String,ttl:String,flag:String
,flag2:String){
        val temp = ContentValues()
        val db = this.writableDatabase
        temp.put(item,name)
        temp.put(details,info)
        temp.put(quantity,num)
        temp.put(size,ttl)
        temp.put(urgent,flag)
        temp.put(buy,flag2)
        temp.put(purchasedate,"")
        db.update(tbName, temp,"id="+ id,null)
        db.close()
    }

    fun
addData(name:String,info:String,num:String,ttl:String,flag:String,flag2:Str
ing){
        val temp = ContentValues()
        val db = this.writableDatabase
        temp.put(item,name)
        temp.put(details,info)
        temp.put(quantity,num)
        temp.put(size,ttl)
        temp.put(urgent,flag)
        temp.put(buy,flag2)
        temp.put(purchasedate,"")
        db.insert(tbName,null,temp)
        db.close()
    }

    fun amen(temp:Int){
        val db = this.writableDatabase
        val contentValues = ContentValues()
        contentValues.put(id,temp)
        db.delete(tbName,"id=" + temp,null)
        db.close()
    }

    fun getData(): ArrayList<MyData>{
        var temp:ArrayList<MyData> = arrayListOf()
        val db = this.readableDatabase
        val cursor = db.rawQuery("select * from " + tbName + " where buy !=
'true'",null)
        cursor!!.moveToFirst()
        if(cursor.count > 0){
            temp.add(MyData(
                cursor.getInt(cursor.getColumnIndexOrThrow(MyDB.id)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.item)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.details)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.quantity)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.size)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.urgent)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.buy)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.purchasedate))
            ))
        }
        while (cursor!!.moveToNext()){
            temp.add(MyData(
                cursor.getInt(cursor.getColumnIndexOrThrow(MyDB.id)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.item)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.details)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.quantity)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.size)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.urgent)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.buy)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.purchasedate))
            ))
        }
        cursor.close()
        return temp
    }

    fun getData2(x:Int): ArrayList<MyData>{
        var temp:ArrayList<MyData> = arrayListOf()
        val db = this.readableDatabase
        val cursor = db.rawQuery("select * from " + tbName + " where id=" +
x,null)
        cursor!!.moveToFirst()
        if(cursor.count > 0){
            temp.add(MyData(
                cursor.getInt(cursor.getColumnIndexOrThrow(MyDB.id)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.item)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.details)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.quantity)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.size)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.urgent)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.buy)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.purchasedate))
            ))
        }
        cursor.close()
        return temp
    }

    fun getData3(): ArrayList<MyData>{
        var temp:ArrayList<MyData> = arrayListOf()
        val db = this.readableDatabase
        val cursor = db.rawQuery("select * from " + tbName + " where buy !=
'true' and urgent == 'true'",null)
        cursor!!.moveToFirst()
        if(cursor.count > 0){
            temp.add(MyData(
                cursor.getInt(cursor.getColumnIndexOrThrow(MyDB.id)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.item)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.details)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.quantity)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.size)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.urgent)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.buy)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.purchasedate))
            ))
        }
        while (cursor!!.moveToNext()){
            temp.add(MyData(
                cursor.getInt(cursor.getColumnIndexOrThrow(MyDB.id)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.item)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.details)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.quantity)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.size)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.urgent)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.buy)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.purchasedate))
            ))
        }
        cursor.close()
        return temp
    }

    fun getData1(): ArrayList<MyData>{
        var temp:ArrayList<MyData> = arrayListOf()
        val db = this.readableDatabase
        val cursor = db.rawQuery("select * from " + tbName + " where buy ==
'true'",null)
        cursor!!.moveToFirst()
        if(cursor.count > 0){
            temp.add(MyData(
                cursor.getInt(cursor.getColumnIndexOrThrow(MyDB.id)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.item)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.details)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.quantity)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.size)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.urgent)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.buy)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.purchasedate))
            ))
        }
        while (cursor!!.moveToNext()){
            temp.add(MyData(
                cursor.getInt(cursor.getColumnIndexOrThrow(MyDB.id)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.item)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.details)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.quantity)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.size)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.urgent)),
                cursor.getString(cursor.getColumnIndexOrThrow(MyDB.buy)),

cursor.getString(cursor.getColumnIndexOrThrow(MyDB.purchasedate))
            ))
        }
        cursor.close()
        return temp
    }

    fun buybuybuy(md:MyData){
        val current = LocalDateTime.now()
        var formatter = DateTimeFormatter.ofPattern("dd MMMM yyyy")
        var formattedDate = current.format(formatter)

        val db = this.writableDatabase
        val temp = ContentValues()
        temp.put(buy,"true")
        temp.put(purchasedate,formattedDate.toString())
        db.update(tbName,temp,"id=" + md.id,null)
        db.close()
    }

    companion object{
        private val DATABASE_NAME = "test"
        private val DATABASE_VERSION = 1
        val tbName = "wa"
        val id = "id"
        val item = "item"
        val details = "details"
        val quantity = "quantity"
        val size = "size"
        val urgent = "urgent"
        val buy = "buy"
        val purchasedate = "date"
    }
}
```

ViewActivity

```
package com.example.a5

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.Button
import android.widget.EditText
import android.widget.ImageView
import android.widget.TextView
import android.widget.Toast

class ViewActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_view)
        val id = Integer.parseInt(intent.getStringExtra("id"))
        val db = MyDB(this)
        val temp = db.getData2(id)
        val txt_item: TextView = findViewById(R.id.txt_item)
        val txt_details:EditText = findViewById(R.id.ET_detail)
        val txt_quantity:TextView = findViewById(R.id.txt_quantity)
        val txt_size:TextView = findViewById(R.id.txt_size)
        val img:ImageView = findViewById(R.id.img_urgent)
        txt_item.setText(temp[0].item)
        txt_details.setText(temp[0].details)
        txt_quantity.setText(temp[0].quantity)
        txt_size.setText(temp[0].size)
        if(temp[0].urgent == "false"){
            img.setImageResource(R.drawable.unchecked)
        }else{
            img.setImageResource(R.drawable.checked)
        }
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        getMenuInflater().inflate(R.menu.right_corner, menu);
        return super.onCreateOptionsMenu(menu)
    }

    override fun onOptionsItemSelected(item: MenuItem) = when(item.itemId)
{
        R.id.shareButton -> {
            val id = intent.getStringExtra("id")
            val intent = Intent(this,EditActivity::class.java)
            intent.putExtra("id",id)
            startActivity(intent)
            true
        }else ->{
            super.onOptionsItemSelected(item)
        }
    }
}
```

UrgentFragment.kt

```
package com.example.a5

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

// TODO: Rename parameter arguments, choose names that match
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private const val ARG_PARAM1 = "param1"
private const val ARG_PARAM2 = "param2"

/**
 * A simple [Fragment] subclass.
 * Use the [UrgentFragment.newInstance] factory method to
 * create an instance of this fragment.
 */
class UrgentFragment : Fragment() {
    // TODO: Rename and change types of parameters
    private var param1: String? = null
    private var param2: String? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        arguments?.let {
            param1 = it.getString(ARG_PARAM1)
            param2 = it.getString(ARG_PARAM2)
        }
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_urgent, container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        val rvContact = view.findViewById<View>(R.id.rc_urgent) as
RecyclerView
        val md = MyDB(view.context)
        rvContact.adapter = RCAdapter(md.getData3(),view.context)
        rvContact.layoutManager = LinearLayoutManager(view.context,
RecyclerView.VERTICAL,false)
    }

    companion object {
        /**
         * Use this factory method to create a new instance of
         * this fragment using the provided parameters.
         *
         * @param param1 Parameter 1.
         * @param param2 Parameter 2.
         * @return A new instance of fragment UrgentFragment.
         */
        // TODO: Rename and change types and number of parameters
        @JvmStatic
        fun newInstance(param1: String, param2: String) =
            UrgentFragment().apply {
                arguments = Bundle().apply {
                    putString(ARG_PARAM1, param1)
                    putString(ARG_PARAM2, param2)
                }
            }
    }
}
```

ShoppingCart

```
package com.example.a5

import android.content.ContentValues
import android.media.Image
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.text.TextUtils
import android.widget.*

class ShoppingCart : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_shopping_cart)
        val db = MyDB(this)
        val arrayList: ArrayList<String> = ArrayList()
        arrayList.add("Default")
        arrayList.add("Small")
        arrayList.add("Medium")
        arrayList.add("Large")
        val arrayAdapter = ArrayAdapter(this,
android.R.layout.simple_spinner_item, arrayList)
        val spn_size: Spinner = findViewById(R.id.spn_size)
        spn_size.adapter = arrayAdapter

        val etItem:EditText = findViewById(R.id.ET_item)
        val etDetails:EditText = findViewById(R.id.ET_details)
        val txtQuantity:TextView = findViewById(R.id.txt_num)
        val spnSize:Spinner = findViewById(R.id.spn_size)
        val cbUrgent:CheckBox = findViewById(R.id.cb_urgent)
        val btnSubmit: Button = findViewById(R.id.btn_addlist)
        val imgUp: ImageView = findViewById(R.id.img_up)
        val imgDown:ImageView = findViewById(R.id.img_down)
        btnSubmit.setOnClickListener(){
            if(TextUtils.isEmpty(etItem.text)){
                etItem.setError("Please enter the item to be purchased")
                return@setOnClickListener
            }else{

db.addData(etItem.text.toString(),etDetails.text.toString(),txtQuantity.tex
t.toString(),

spnSize.selectedItem.toString(),cbUrgent.isChecked.toString(),"false")
                etItem.text.clear()
                etDetails.text.clear()
                txtQuantity.text = "1"
                spnSize.setSelection(0)
                cbUrgent.isChecked = false
            }
        }
        imgUp.setOnClickListener(){
            var x:Int = Integer.parseInt(txtQuantity.text.toString())
            x += 1
            txtQuantity.text = x.toString()
        }

        imgDown.setOnClickListener(){
            var x:Int = Integer.parseInt(txtQuantity.text.toString())
            if(x > 1){
                x -= 1
                txtQuantity.text = x.toString()
            }
        }



    }
}
```

RcAdapter_1

```
package com.example.a5

import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.Switch
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import java.time.LocalDateTime
import java.time.format.DateTimeFormatter

class RCAdapter_1(private val mContacts:List<MyData>,val context: Context):
RecyclerView.Adapter<RCAdapter_1.ViewHolder>() {
    inner class ViewHolder(itemview: View):
RecyclerView.ViewHolder(itemview){
        val txt_item: TextView = itemview.findViewById(R.id.txt_item)
        val txt_quantity: TextView =
itemview.findViewById(R.id.txt_quantity)
        val txt_size: TextView = itemview.findViewById(R.id.txt_size)
        val img_icon: ImageView = itemview.findViewById(R.id.img_uob)
        val txt_date:TextView = itemview.findViewById(R.id.txt_date)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
ViewHolder {
        val context = parent.context
        val inflater = LayoutInflater.from(context)
        val contactView = inflater.inflate(R.layout.rc_row_1,parent,false)
        return ViewHolder(contactView)
    }

    override fun onBindViewHolder(holder: RCAdapter_1.ViewHolder, position:
Int) {
        val contact:MyData = mContacts.get(position)
        holder.txt_item.setText(contact.item)
        holder.txt_quantity.setText(contact.quantity)
        holder.txt_size.setText(contact.size)
        val current = contact.date.split(" ").toTypedArray()
        holder.txt_date.setText(current[0] + " " + current[1].take(3) + " "
+ current[2])

        holder.img_icon.setImageResource(contact.getPicture2())
    }

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

RcAdapter

```
package com.example.a5

import android.content.Context
import android.content.DialogInterface
import android.content.Intent
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.Switch
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.recyclerview.widget.RecyclerView


class RCAdapter(private val mContacts:List<MyData>,val context:Context):
RecyclerView.Adapter<RCAdapter.ViewHolder>() {
    val db = MyDB(context)
    inner class ViewHolder(itemview: View):
RecyclerView.ViewHolder(itemview){
        val txt_item: TextView = itemview.findViewById(R.id.txt_item)
        val txt_quantity: TextView =
itemview.findViewById(R.id.txt_quantity)
        val txt_size: TextView = itemview.findViewById(R.id.txt_size)
        val sw_buy: Switch = itemview.findViewById(R.id.sw_buy)
        val img_icon: ImageView = itemview.findViewById(R.id.img_uob)

    }

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

    override fun onBindViewHolder(holder: RCAdapter.ViewHolder, position:
Int) {
        val contact:MyData = mContacts.get(position)
        holder.txt_item.setText(contact.item)
        holder.txt_quantity.setText(contact.quantity)
        holder.txt_size.setText(contact.size)

        holder.img_icon.setImageResource(contact.getPicture())
        holder.img_icon.setOnClickListener {
//            val intent: Intent =
Intent(holder.itemView.context,BookingForm::class.java)
//            intent.putExtra("hostelName",contact.name)
//            holder.itemView.context.startActivity(intent)
        }

        holder.sw_buy.setOnClickListener(){
            val x = MyDB(context)
            x.buybuybuy(contact)
            (context as MainActivity).loadFragment(HomeFragment())
        }
        holder.itemView.setOnClickListener(){
            val intent  = Intent(context,ViewActivity::class.java)
            intent.putExtra("id",contact.id.toString())
            //val intent  = Intent(this,ViewActivity::class.java)
            context.startActivity(intent)
        }

        val btnDelete =  {dialog:DialogInterface,which:Int->
            db.amen(contact.id)
            (context as MainActivity).loadFragment(HomeFragment())
        }

        val btnCancel = {dialog:DialogInterface,which:Int->
            Toast.makeText(context,"cancel",Toast.LENGTH_LONG).show()
        }

        holder.itemView.setOnLongClickListener(){
            val builder = AlertDialog.Builder(context)
            builder.setTitle("Delete Item")
            builder.setMessage("Are you sure you want to delete "+
contact.item +" from the list?")

builder.setPositiveButton("Delete",DialogInterface.OnClickListener(function
= btnDelete))

builder.setNegativeButton("Cancel",DialogInterface.OnClickListener(function
= btnCancel))
            builder.show()
            return@setOnLongClickListener true
        }
    }

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

MyData

```
package com.example.a5

data class MyData(val id:Int,val item:String,val details:String,val
quantity:String,val size:String,val urgent:String,val buy:String,val
date:String){

    fun getPicture():Int{
        var pic:Int
        if(urgent == "false"){
            pic = R.drawable.buy
        }else{
            pic = R.drawable.urgent
        }
        return pic
    }

    fun getChecked():Boolean{
        var flag:Boolean
        if(urgent == "false"){
            flag = false
        }else{
            flag = true
        }
        return flag
    }

    fun getPicture2():Int{
        return R.drawable.bought
    }

    fun getPosition():Int{
        var x:Int = 3
        if(size == "Default"){
            x = 0
        }else if(size == "Small"){
            x = 1
        }else if(size == "Medium"){
            x = 2
        }
        return x
    }
}
```

HomeFragment

```
package com.example.a5

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.Button
import android.widget.TextView
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

// TODO: Rename parameter arguments, choose names that match
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private const val ARG_PARAM1 = "param1"
private const val ARG_PARAM2 = "param2"

/**
 * A simple [Fragment] subclass.
 * Use the [HomeFragment.newInstance] factory method to
 * create an instance of this fragment.
 */
class HomeFragment() : Fragment() {
    // TODO: Rename and change types of parameters
    private var param1: String? = null
    private var param2: String? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        arguments?.let {
            param1 = it.getString(ARG_PARAM1)
            param2 = it.getString(ARG_PARAM2)
        }

    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_home, container, false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        val rvContact = view.findViewById<View>(R.id.rc_home) as
RecyclerView
        val md = MyDB(view.context)
        rvContact.adapter = RCAdapter(md.getData(),view.context)
        rvContact.layoutManager =
LinearLayoutManager(view.context,RecyclerView.VERTICAL,false)
    }

    companion object {
        /**
         * Use this factory method to create a new instance of
         * this fragment using the provided parameters.
         *
         * @param param1 Parameter 1.
         * @param param2 Parameter 2.
         * @return A new instance of fragment HomeFragment.
         */
        // TODO: Rename and change types and number of parameters
        @JvmStatic
        fun newInstance(param1: String, param2: String) =
            HomeFragment().apply {
                arguments = Bundle().apply {
                    putString(ARG_PARAM1, param1)
                    putString(ARG_PARAM2, param2)
                }
            }
    }
}
```

EditActiviy

```
package com.example.a5

import android.content.Intent
import android.os.Bundle
import android.text.TextUtils
import android.widget.*
import androidx.appcompat.app.AppCompatActivity


class EditActivity : AppCompatActivity() {


    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_edit)
        val id =  Integer.parseInt(intent.getStringExtra("id"))
        val db = MyDB(this)
        val arrayList: ArrayList<String> = ArrayList()
        arrayList.add("Default")
        arrayList.add("Small")
        arrayList.add("Medium")
        arrayList.add("Large")
        val arrayAdapter = ArrayAdapter(this,
android.R.layout.simple_spinner_item, arrayList)
        val spn_size: Spinner = findViewById(R.id.spn_size)
        spn_size.adapter = arrayAdapter

        val etItem: EditText = findViewById(R.id.ET_item)
        val etDetails: EditText = findViewById(R.id.ET_details)
        val txtQuantity: TextView = findViewById(R.id.txt_num)
        val spnSize: Spinner = findViewById(R.id.spn_size)
        val cbUrgent: CheckBox = findViewById(R.id.cb_urgent)
        val btnSubmit: Button = findViewById(R.id.btn_addlist)
        val imgUp: ImageView = findViewById(R.id.img_up)
        val imgDown: ImageView = findViewById(R.id.img_down)

        val ddd = db.getData2(id)

        etItem.setText(ddd[0].item)
        etDetails.setText(ddd[0].details)
        txtQuantity.setText(ddd[0].quantity)
        cbUrgent.isChecked = ddd[0].getChecked()
        spn_size.setSelection(ddd[0].getPosition())


        btnSubmit.setOnClickListener(){
            if(TextUtils.isEmpty(etItem.text)){
                etItem.setError("Please enter the item to be purchased")
                return@setOnClickListener
            }else{

db.updateData(id,etItem.text.toString(),etDetails.text.toString(),txtQuanti
ty.text.toString(),

spnSize.selectedItem.toString(),cbUrgent.isChecked.toString(),"false")
                val intent = Intent(applicationContext,
MainActivity::class.java)
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)
                startActivity(intent)
            }
        }
        imgUp.setOnClickListener(){
            var x:Int = Integer.parseInt(txtQuantity.text.toString())
            x += 1
            txtQuantity.text = x.toString()
        }

        imgDown.setOnClickListener(){
            var x:Int = Integer.parseInt(txtQuantity.text.toString())
            if(x > 1){
                x -= 1
                txtQuantity.text = x.toString()
            }
        }

    }
}
```

CompleteFragment

```
package com.example.a5

import android.os.Bundle
import androidx.fragment.app.Fragment
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

// TODO: Rename parameter arguments, choose names that match
// the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
private const val ARG_PARAM1 = "param1"
private const val ARG_PARAM2 = "param2"

/**
 * A simple [Fragment] subclass.
 * Use the [CompleteFragment.newInstance] factory method to
 * create an instance of this fragment.
 */
class CompleteFragment : Fragment() {
    // TODO: Rename and change types of parameters
    private var param1: String? = null
    private var param2: String? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        arguments?.let {
            param1 = it.getString(ARG_PARAM1)
            param2 = it.getString(ARG_PARAM2)
        }
    }

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_complete, container,
false)
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        val rvContact = view.findViewById<View>(R.id.rc_complete) as
RecyclerView
        val md = MyDB(view.context)
        rvContact.adapter = RCAdapter_1(md.getData1(),view.context)
        rvContact.layoutManager = LinearLayoutManager(view.context,
RecyclerView.VERTICAL,false)
    }

    companion object {
        /**
         * Use this factory method to create a new instance of
         * this fragment using the provided parameters.
         *
         * @param param1 Parameter 1.
         * @param param2 Parameter 2.
         * @return A new instance of fragment CompleteFragment.
         */
        // TODO: Rename and change types and number of parameters
        @JvmStatic
        fun newInstance(param1: String, param2: String) =
            CompleteFragment().apply {
                arguments = Bundle().apply {
                    putString(ARG_PARAM1, param1)
                    putString(ARG_PARAM2, param2)
                }
            }
    }
}
```