***“By submitting this electronically, using an account & password only I  have access to, I hereby certify that this material which I now submit for assessment is entirely my own work and has not been taken from the work of others, save and to the extent, that such work has been cited and acknowledged within the text of my work.”***

Mobile Assignment Two - WaitPerson v1.1 Nov2014 [goo.gl/BROXPu](http://goo.gl/BROXPu)

**UserName : pio Password:pio**

<Pio O’Connell/H Dip Cloud Mobile Development/pio\_o\_con@hotmail.com>

1. **Design and implement an Android App for a waitperson taking orders in a restaurant.**  The app should include the following:
   1. Basic functionality that addresses App’s purpose: at least 3 Activities and navigation between them. Simple UI components e.g. TextView, EditText & Button.

The Application has 6 activities, namely –

Splash – on entry

Security – for password access

Prefs – to allow user to set user details and also toppings available for the pizzas for the day(not fully integrated)

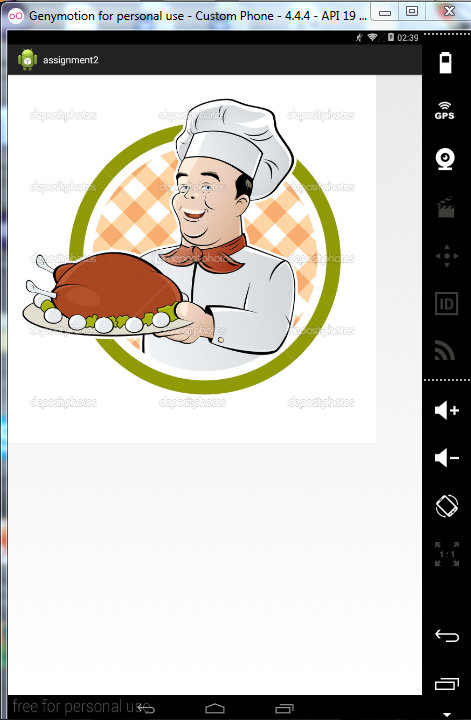
FoodForTheDay – allows user to select which items on the menu for the day(not fully integrated)

AboutUs – details the restaurant

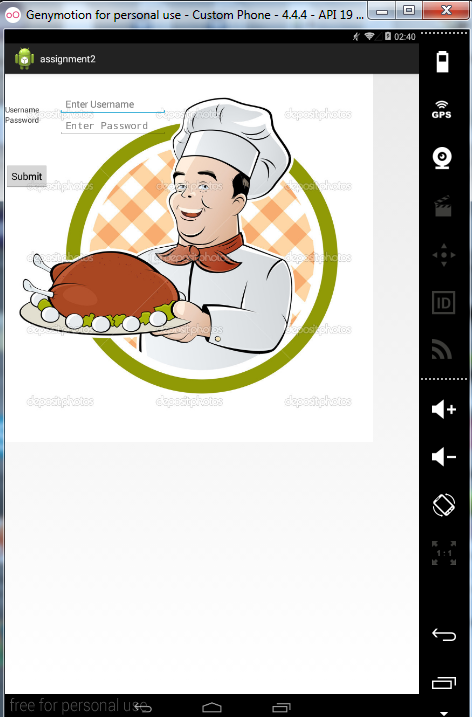
RestaurantTables – which displays the tables available for the restaurant

MainActivity – which handles the actual order.

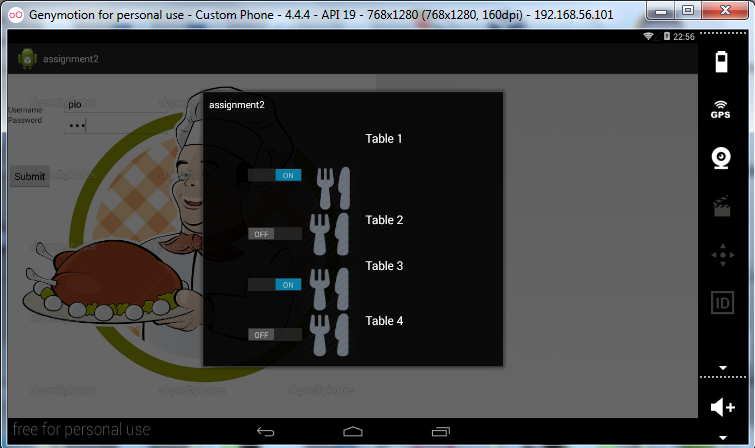
On Launch, the splash is displayed briefly before the login window.

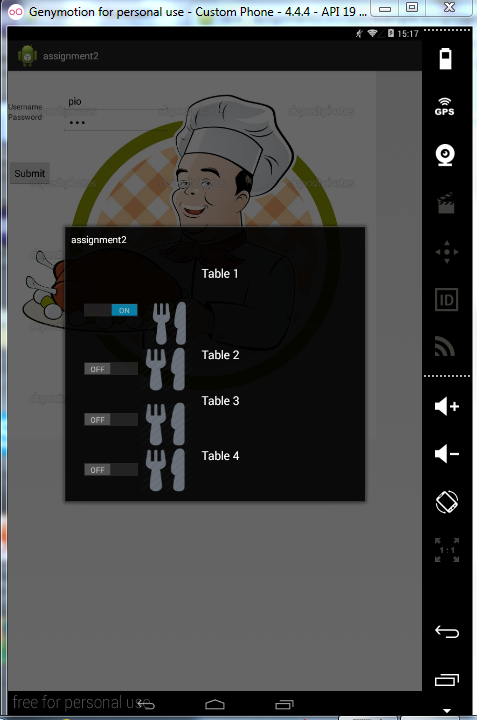


Login Window-

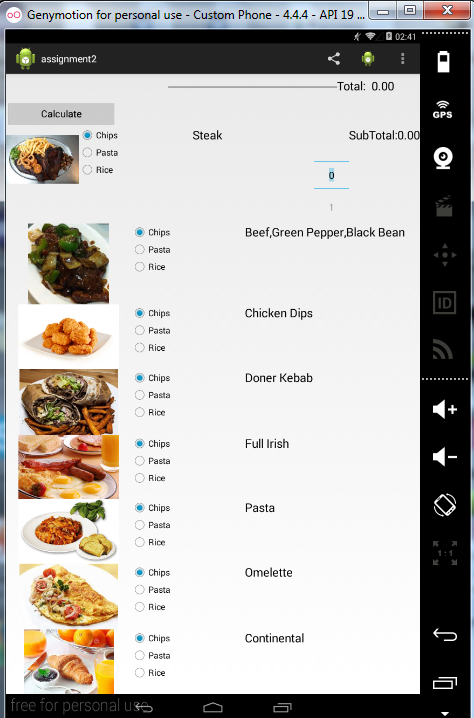


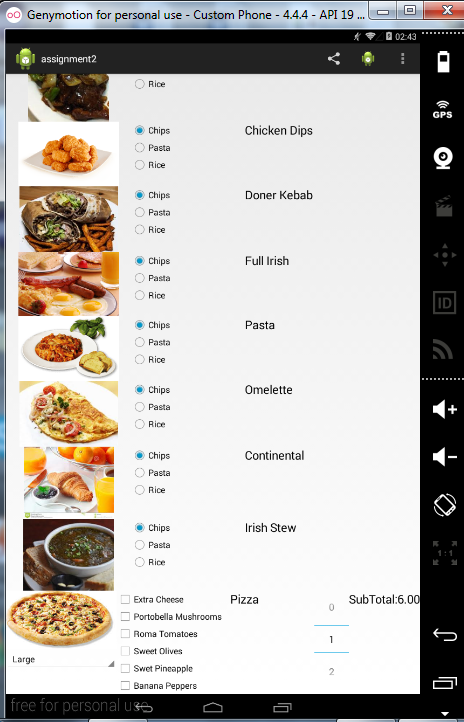
Once user logged in, the Tables are displayed disabled:





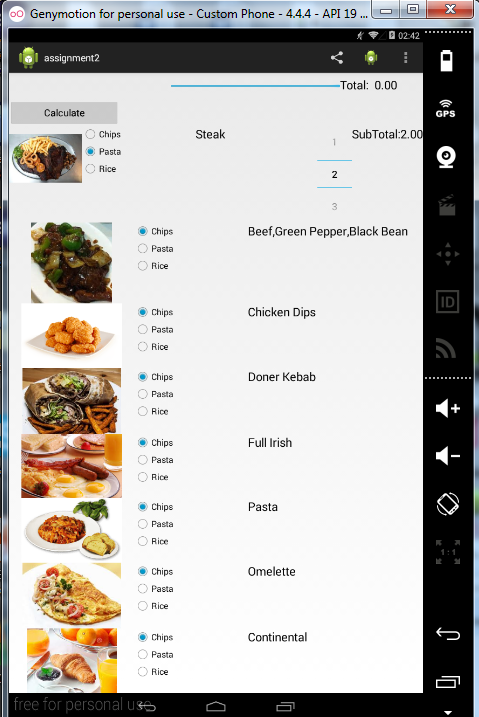
Clicking on an enabled table- the user can then enter the preset menu for the table window.



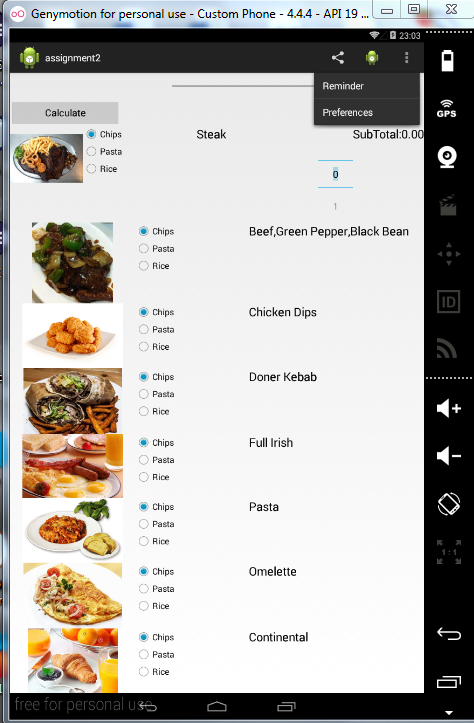


Once the items have been selected – code there only for Steak and Pizza – the user can calculate the bill. Subtotal automatically calculated.

The progress bar indicates the computation is complete..

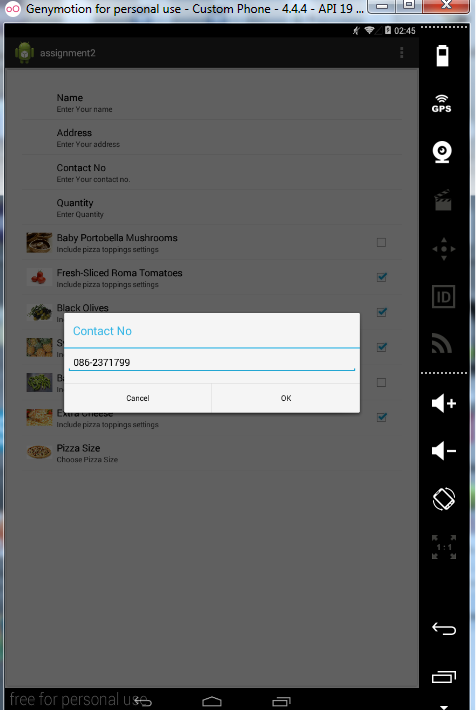


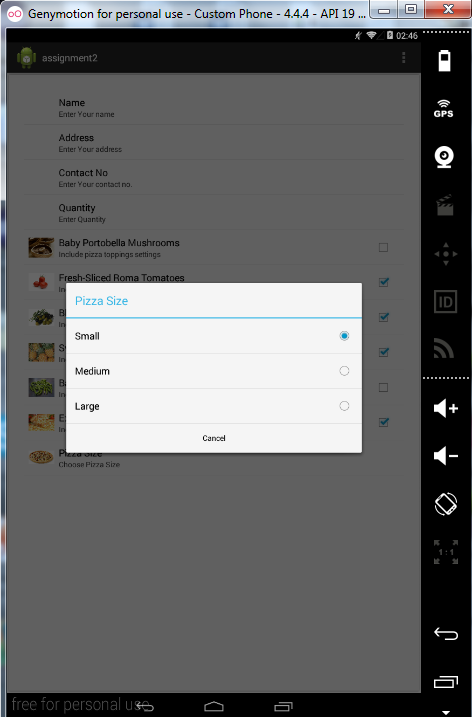
Notification manager and Pending Intents-



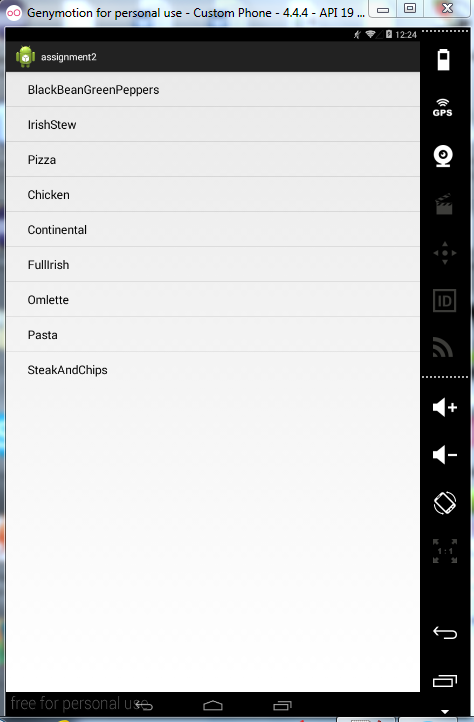
The Reminder submenu allows user to set a message that the order has been taken. On selecting the user returns to the main menu..

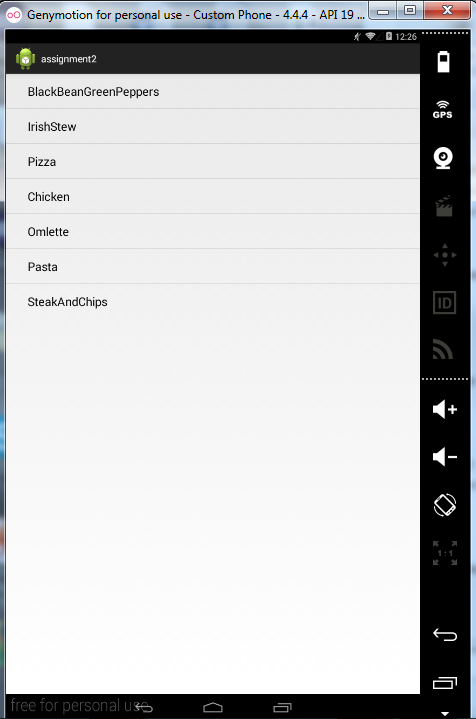
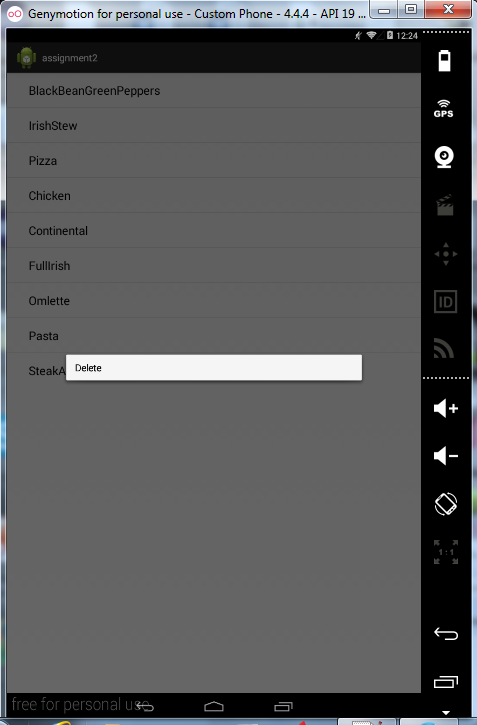
Preferences – selecting Preferences the user is allowed set preferences for user and menu toppings. These are saved but a yet not integrated into the system.





Clicking on the Share Icon, it’s possible(using a long click) to remove items from the menu.I have an array adapter written for this but need to modify it a little to display an icon corresponding to each item.





The Code – Layout Managers

***Horizontal/ vertical Linear Layout Managers***

Used for the **restaurant** tables window in landscape mode and portrait modes

tableButton1 = (Button) findViewById(R.id.Button01);

tableButton1.setEnabled(false);

tableButton1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivityForResult(intentGetMessage,2);

}

});

***FrameLayout / Relative Layout Managers***

For the security window,uses a combination of FrameLayout and RelativeLayout.

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<FrameLayout android:id=*"@+id/chef02"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

xmlns:android=*"http://schemas.android.com/apk/res/android"*

>

<ImageView android:id=*"@+id/chef01"*

android:src=*"@drawable/chef"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:scaleType=*"center"* />

<RelativeLayout

android:id=*"@+id/RelativeLayout01"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"* >

***Button,TextView,EditText***

<Button

android:id=*"@+id/bResults"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/TextView02"*

android:layout\_below=*"@+id/etPassword"*

android:layout\_marginTop=*"50dp"*

android:text=*"@+string/Submit"* />

<EditText

android:id=*"@+id/etUserName"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentTop=*"true"*

android:layout\_marginTop=*"36dp"*

android:layout\_toRightOf=*"@+id/TextView02"*

android:ems=*"10"*

android:hint=*"@string/Username"*

android:width=*"200dp"* />

<TextView

android:id=*"@+id/TextView01"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignBottom=*"@+id/etUserName"*

android:layout\_alignParentLeft=*"true"*

android:text=*"@string/Username1"*

android:width=*"100dp"* />

***Frame Layout Manager***

The **FrameLayout** Window Manager is used for the loginwindow called security here-allowing the image to be mixed with the edit controls

<FrameLayout android:id=*"@+id/chef02"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*

xmlns:android=*"http://schemas.android.com/apk/res/android"*

>

<ImageView android:id=*"@+id/chef01"*

android:src=*"@drawable/chef"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:scaleType=*"center"* />

***TableLayout Manager***

The TableLayout Manager is used for the activity\_main which displays all the items on the menu for the day.

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<ScrollView xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*>

<TableLayout

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"* >

<TableLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"* >

<TableRow

android:id=*"@+id/tableRow99"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"* >

<Button

android:id=*"@+id/bCalculate"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginTop=*"50dp"*

android:text=*"@+string/Calculate"* />

<View

android:layout\_width=*"100dp"*

android:layout\_height=*"match\_parent"*

android:layout\_weight=*"1"*/>

<ProgressBar

android:id=*"@+id/progressbar"*

style=*"?android:attr/progressBarStyleHorizontal"*

android:layout\_width=*"270px"*

android:layout\_height=*"50px"*

android:progress=*"0"*

android:secondaryProgress=*"0"* />

<TextView

android:id=*"@+id/textView970"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/Total"*

android:textAppearance=*"?android:attr/textAppearanceLarge"* />

<TextView

android:id=*"@+id/textView972"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/Zero"*

android:textAppearance=*"?android:attr/textAppearanceLarge"* />

</TableRow>

***Notification Manager and Pending Events***

**private** NotificationManager mNManager;

**private** **static** **final** **int** *NOTIFY\_ID*=1100;

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

nm = (NotificationManager) getSystemService(*NOTIFICATION\_SERVICE*);

nm.cancel(*UniqueID*);

When user selects Reminder it from activity window, it closes the window and returns to main window sending a pending event. It’s the users choice whether they wish to send a pending event.This code has been deprecated since API 11.

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

PendingIntent pIntent = PendingIntent.*getActivity*(**this**, 0,intent, 0);

String body = "Order taken for table";

String title ="Message from cook";

Notification n = **new** ~~Notification~~(R.drawable.*chef*,body,System.*currentTimeMillis*());

n.~~setLatestEventInfo~~(**this**, title, body, pIntent);

n.defaults = Notification.*DEFAULT\_ALL*;

m.notify(*UniqueID*,n);

finish();

**break**;

***Spinner***

Spinner is achieved in the Activity main window

@SuppressWarnings({ "unchecked", "rawtypes" })

ArrayAdapter<String> adapter07 = **new** ArrayAdapter(MainActivity.**this**,android.R.layout.*simple\_spinner\_item*,pizzaValues);

spinner07 = (Spinner)findViewById(R.id.*spinner07*);

spinner07.setAdapter(adapter07);

spinner07.setOnItemSelectedListener(**this**);

An onItemSelected event is then sent each time spinner used.

**public** **void** onItemSelected(AdapterView<?> parent, View view, **int** position,

**long** id) {

**int** position1 = spinner07.getSelectedItemPosition();

**switch** (parent.getId()) {

**case** R.id.*spinner07*:

**switch**(position1) {

**case** 0:

SubTotal07.setText(String.*format*( "%.2f",(pizzaPrices[2])));

**break**;

**case** 1:

SubTotal07.setText(String.*format*( "%.2f",(pizzaPrices[1])));

**break**;

**case** 2:

SubTotal07.setText(String.*format*( "%.2f",(pizzaPrices[0])));

**break**;

}

}

}

***Toggle Button***

Switch onOffSwitch1,onOffSwitch2,onOffSwitch3,onOffSwitch4;

onOffSwitch1 = (Switch) findViewById(R.id.*switch1*);

onOffSwitch1.setOnCheckedChangeListener(**new** OnCheckedChangeListener() {

**public** **void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {

**if**(isChecked)

tableButton2.setEnabled(**true**);

**else**

tableButton2.setEnabled(**false**);

}

});

***ProgressBar***

**final** Runnable mUpdateResults = **new** Runnable(){

@Override

**public** **void** run() {

**int** size=20000;

**double** tmp;

**for**(**int** i=0;i<1000;i++){

**for**(**int** j=0;j<1000;j++){

tmp = 10 \* Math.*log*(i+1)/Math.*log*(j+1);

}

}

*mprogressBar*.setProgress(percent\_done);

}

};

*mprogressBar*= (ProgressBar) findViewById(R.id.*progressbar*) ;

**public** **void** onClick(View v) {

percent\_done = 0;

mHandler.post(mUpdateResults);

}

**ActionBar**

***Menus/Submenus***

The activity\_main.layout for MainActivity..

<menu xmlns:android=*"http://schemas.android.com/apk/res/android"*

xmlns:tools=*"http://schemas.android.com/tools"*

tools:context=*"com.example.assignment2.MainActivity"* >

<item

android:title=*"@string/MenuForDay"*

android:id=*"@+id/menuforday"*

android:showAsAction=*"always"*

android:icon=*"@drawable/ic\_action\_share"*

/>

<item

android:title=*"@string/Aboutus"*

android:id=*"@+id/aboutUs"*

android:showAsAction=*"always"*

android:icon=*"@drawable/ic\_launcher"*

android:alphabeticShortcut=*"a"* />

<item

android:title=*"@string/Quit"*

android:id=*"@+id/quit"*

/>

<item

android:title=*"@string/Preferences"*

android:id=*"@+id/preferences1"*

/>

</menu>

@Override

**public** **boolean** onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

**switch**(item.getItemId()){

**case** R.id.*aboutUs*:

Intent i = **new** Intent("com.example.assignment2.ABOUTUS");

startActivity(i);

**break**;

**case** R.id.*menuforday*:

// Intent i = new Intent("com.example.assignment2.PREFS");

// startActivity(i);

Intent intentGetMessage = **new** Intent(**this**,Foodforday.**class**);

startActivityForResult(intentGetMessage,6);

**break**;

**case** R.id.*preferences1*:

// Intent i = new Intent("com.example.assignment2.PREFS");

// startActivity(i);

Intent intentGetMessage1 = **new** Intent(**this**,Prefs.**class**);

startActivityForResult(intentGetMessage1,2);

**break**;

**case** R.id.*quit*:

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

PendingIntent pIntent = PendingIntent.*getActivity*(**this**, 0, intent, 0);

String body = "Order taken for table";

String title ="Message from cook";

Notification n = **new** Notification(R.drawable.*chef*,body,System.*currentTimeMillis*());

n.setLatestEventInfo(**this**, title, body, pIntent);

n.defaults = Notification.*DEFAULT\_ALL*;

nm.notify(*UniqueID*,n);

finish();

**break**;

}

**return** **super**.onOptionsItemSelected(item);

}

The menus for preferences .app is in the pizza\_menu.xml

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<menu xmlns:android=*"http://schemas.android.com/apk/res/android"* >

<item

android:title=*"@string/Aboutus"*

android:id=*"@+id/aboutUs"*

android:alphabeticShortcut=*"a"* />

<item

android:title=*"@string/Preferences"*

android:id=*"@+id/preferences"*

/>

<item

android:title=*"@string/Quit"*

android:id=*"@+id/quit"* android:icon=*"@drawable/ic\_launcher"*

android:showAsAction=*"never"*/>

</menu>

The submenu for preferences is stored in xml/preferences…

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<menu xmlns:android=*"http://schemas.android.com/apk/res/android"* >

<item

android:title=*"@string/Aboutus"*

android:id=*"@+id/aboutUs"*

android:alphabeticShortcut=*"a"* />

<item

android:title=*"@string/Preferences"*

android:id=*"@+id/preferences"*

/>

<item

android:title=*"@string/Quit"*

android:id=*"@+id/quit"* android:icon=*"@drawable/ic\_launcher"*

android:showAsAction=*"never"*/>

</menu>

**public** **boolean** onOptionsItemSelected(MenuItem item) {

// **TODO** Auto-generated method stub

**switch**(item.getItemId()){

**case** R.id.*aboutUs*:

Intent i = **new** Intent("com.example.assignment2.ABOUTUS");

startActivity(i);

**break**;

**case** R.id.*quit*:

Intent intentMessage=**new** Intent();

String message= "logout";

intentMessage.putExtra("MESSAGE",message);

setResult(2,intentMessage);

finish();

**break**;

}

**return** **super**.onOptionsItemSelected(item);

}

***RadioButton***

radioGroup01 = (RadioGroup) findViewById(R.id.*radioGroup01*);

radioGroup01.setOnCheckedChangeListener(**new** RadioGroup.OnCheckedChangeListener()

{

**public** **void** onCheckedChanged(RadioGroup group, **int** checkedId) {

RadioButton rb=(RadioButton)findViewById(checkedId);

Staples theDay = Staples.*PASTA*;

**switch**(checkedId){

**case** R.id.*radio010*:

menuItem[1].setItemExtra( Staples.*CHIPS*)

**break**;

**case** R.id.*radio011*:

menuItem[1].setItemExtra( Staples.*PASTA*);

**break**;

**case** R.id.*radio012*:

menuItem[1].setItemExtra( Staples.*RICE*);

**break**;

}

}

});

***ImageButton:***

<Button

android:id=*"@+id/Button02"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:layout\_marginLeft=*"15dp"*

android:layout\_marginTop=*"66dp"*

android:background=*"@drawable/table2"* />

Button tableButton1,tableButton2,tableButton3,tableButton4;

tableButton1 = (Button) findViewById(R.id.Button01);

tableButton1.setEnabled(false);

tableButton1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivityForResult(intentGetMessage,2);

}

});

**Screen Layout with layout created pragmatically-**

Both **MenuOfTheDay**(which I haven’t allowed access to – uses an arrayadaper.

Also **FoodForTheDay**

mport java.util.ArrayList;

import java.util.List;

import android.app.Activity;

import android.os.Bundle;

import android.view.ContextMenu;

import android.view.ContextMenu.ContextMenuInfo;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.view.View;

import android.widget.AdapterView.AdapterContextMenuInfo;

import android.widget.ArrayAdapter;

import android.widget.ListView;

public class Foodforday extends Activity{

ListView list\_View;

List<String> list= new ArrayList<String>();

ArrayAdapter<String> adapter;

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.food);

list\_View = (ListView) findViewById(R.id.list\_view);

list.add("BlackBeanGreenPeppers");

list.add("IrishStew");

list.add("Pizza");

list.add("Chicken");

list.add("Continental");

list.add("FullIrish");

list.add("Omlette");

list.add("Pasta");

list.add("SteakAndChips");

adapter = new ArrayAdapter<String>(this,android.R.layout.simple\_expandable\_list\_item\_1,list);

list\_View.setAdapter(adapter);

registerForContextMenu(list\_View);

}

@Override

public void onCreateContextMenu(ContextMenu menu, View v,

ContextMenuInfo menuInfo) {

// TODO Auto-generated method stub

super.onCreateContextMenu(menu, v, menuInfo);

MenuInflater inflater= getMenuInflater();

inflater.inflate(R.menu.contextmenu,menu);

}

@Override

public boolean onContextItemSelected(MenuItem item) {

// TODO Auto-generated method stub

AdapterContextMenuInfo info= (AdapterContextMenuInfo) item.getMenuInfo();

switch(item.getItemId())

{

case R.id.delete\_id:

System.out.println("\n\n ..trying to delete");

list.remove(info.position);

adapter.notifyDataSetChanged();

return true;

default:

return super.onContextItemSelected(item);

}

}

}

**Storage**

I implemented preferences for the preference menu, it does retain the settings using Shared Preferences.

I also implemented databases in Database.java for login. Unfortunately, I ran into some difficulty with the emulator, there’s a bug with the emulator that if u change the column name. Time consuming.

I did not implement java.io but I would have liked to also implement JSON or parcelable. No time.I was intending to all the data I needed to pass from the Main\_activity to the Reataurant tables. The MainWindow has a full class behind it in itemDetails(for each menu item) and PizzaItem(for the different options in the pizza)

**package** com.example.assignment2;

**public** **class** itemDetails {

Staples itemExtra;

**int** itemQuantity;

**double** Subtotal;

String itemName;

**double** itemPrice;

}

**public** itemDetails(Staples itemExtra, **int** itemQuantity, **double** subtotal,String itemName,**double** itemPrice) {

**this**.itemName = itemName;

**this**.itemExtra = itemExtra;

**this**.itemQuantity = itemQuantity;

**this**.itemPrice = itemPrice;

**this**.Subtotal = subtotal;

}

Also for the pizza item-

**public** **class** PizzaItem {

pizzaSize PizzaSize;

**boolean** ExtraCheese,Mushrooms,Tomatoes,Olives,Pineapple,Peppers;

String PizzaName;

**int** Quantity;

**public** PizzaItem(pizzaSize pizzaSize, **boolean** extraCheese, **boolean** mushrooms,

**boolean** tomatoes, **boolean** olives, **boolean** pineapple,

**boolean** peppers, String pizzaName, **int** quantity) {

**super**();

PizzaSize = pizzaSize;

ExtraCheese = extraCheese;

Mushrooms = mushrooms;

Tomatoes = tomatoes;

Olives = olives;

Pineapple = pineapple;

Peppers = peppers;

PizzaName = pizzaName;

Quantity = quantity;

}

Evaluation:

Whilst I learnt a lot by doing this exercise, the **annoying** thing is that I have **no time** to complete all tasks.

I would have liked to –

* Be able to pass structures between activities using json or parcelable.
* Both preferences and itemsofthe day should allow user make settings for the day
  + - For items of the day menu, that the menu also displayed an icon. And changes through the system
    - For preferences that the settings would be throughout the system
* Have the database function able.
* Implement fragments
* Implement widgets