# BaseCallback class

package com.menu.menu.Classes;  
  
import java.util.concurrent.Callable;  
  
public class BaseCallback implements Callable<Void>  
{  
 protected String m\_message = null;  
 public void SetMessage(String message)  
 {  
 m\_message = message;  
 }  
 String GetMessage()  
 {  
 return m\_message;  
 }  
  
 @Override  
 public Void call() throws Exception  
 {  
 return null;  
 }  
}

# BasketItem class

package com.menu.menu.Classes;  
  
import java.io.Serializable;  
  
public class BasketItem implements Serializable  
{  
 private String m\_id = null;  
 private String m\_mealId = null; //Even though we have m\_meal, it may not be set at all times...  
 private String m\_numberOfMeals = null;  
 private Meal m\_meal = null;  
 private Boolean m\_isTakeaway = null;  
  
 public BasketItem()  
 {  
 }  
  
 //Creating a basket order  
 public BasketItem(Meal meal, String numberOfMeals, boolean isTakeaway)  
 {  
 m\_meal = meal;  
 m\_mealId = meal.getId();  
 m\_numberOfMeals = numberOfMeals;  
 m\_isTakeaway = isTakeaway;  
 }  
  
 public String getId()  
 {  
 return m\_id;  
 }  
  
 public void SetId(String id)  
 {  
 m\_id = id;  
 }  
  
 public Meal GetMeal()  
 {  
 return m\_meal;  
 }  
  
 public void SetMeal(Meal meal)  
 {  
 m\_meal = meal;  
 }  
  
 public void SetMealId(String id)  
 {  
 m\_mealId = id;  
 }  
  
 public String GetMealId()  
 {  
 return m\_mealId;  
 }  
  
 public String GetNumberOfMeals()  
 {  
 return m\_numberOfMeals;  
 }  
  
 public int GetNumberOfMeals\_n()  
 {  
 try  
 {  
 return Integer.*parseInt*(m\_numberOfMeals);  
 }  
 catch (Exception e)  
 {  
 return 0;  
 }  
 }  
  
  
 public void SetNumberOfMeals(String numberOfMeals)  
 {  
 m\_numberOfMeals = numberOfMeals;  
 }  
  
 public void SetNumberOfMeals\_n(int num)  
 {  
 m\_numberOfMeals = String.*valueOf*(num);  
 }  
  
 public Boolean GetTakeaway()  
 {  
 return m\_isTakeaway;  
 }  
  
 public void SetTakeaway(Boolean takeaway)  
 {  
 m\_isTakeaway = takeaway;  
 }  
}

# DatabaseCommunicator class

package com.menu.menu.Classes;  
  
import android.os.AsyncTask;  
import org.apache.commons.io.IOUtils;  
import java.io.InputStream;  
import java.io.OutputStreamWriter;  
import java.net.URL;  
import java.net.URLConnection;  
import java.nio.charset.StandardCharsets;  
import java.util.ArrayList;  
  
public class DatabaseCommunicator  
{  
 public static final String *m\_userTable* = "[menudatabase].[dbo].[User]";  
 public static final String *m\_orderTable* = "[menudatabase].[dbo].[Order]";  
 public static final String *m\_mealTable* = "[menudatabase].[dbo].[meal]";  
 private final String m\_dbRequestUrl = "http://themenuapp.gearhostpreview.com/databaseAPI.php?request=";  
 private final String m\_dbPostUrl = "http://themenuapp.gearhostpreview.com/databaseAPIpost.php";  
  
 public DatabaseCommunicator()  
 {  
 }  
  
 public void RequestUserData(UsersCallback usersCallback)  
 {  
 new ReqUserData().execute(usersCallback);  
 }  
  
 public void RequestOrderData(OrdersCallback ordersCallback)  
 {  
 new ReqOrderData().execute(ordersCallback);  
 }  
  
 public void RequestMealData(MealsCallback mealsCallback)  
 {  
 new ReqMealData().execute(mealsCallback);  
 }  
  
 public void GenericUpload(BaseCallback cb)  
 {  
 new UploadData().execute(cb);  
 }  
  
 private class ReqUserData extends AsyncTask<UsersCallback, String, String>  
 {  
 @Override  
 protected String doInBackground(UsersCallback... params)  
 {  
 try  
 {  
 URLConnection connection = new URL(m\_dbRequestUrl + params[0].GetMessage()).openConnection();  
 params[0].SetMessage("Error");//Shouldn't ever be used, but need to set something different than connection messaged since used as a debug message for recipient class  
 InputStream res = connection.getInputStream();  
  
 String webResponse = IOUtils.*toString*(res, StandardCharsets.*UTF\_8*);  
 if(webResponse != null)  
 params[0].m\_bInternetIssue = false;  
  
 if(webResponse!=null && new Character(webResponse.charAt(0)).equals('['))  
 {  
 ArrayList<User> users = new ArrayList<>();  
  
 String[] results = ParseWebResponse(webResponse);  
  
 for (String user: results)  
 {  
 String userElements[] = user.split(",");  
  
 User u = new User();  
 u.setId(userElements[0]);  
 u.setUsername(userElements[1]);  
 u.setPassword(userElements[2]);  
 u.setFullName(userElements[3]);  
 u.setAddressLine1(userElements[4]);  
 u.setAddressLine2(userElements[5]);  
 u.setAddressLine3(userElements[6]);  
 u.setAddressPostCode(userElements[7]);  
 u.setAddressDescription(userElements[8]);  
 u.setDOB(userElements[9]);  
 u.setLoggedIn(userElements[10]);  
 u.setEmail(userElements[11]);  
 u.setPhone(userElements[12]);  
 u.setRating(userElements[13]);  
 u.setIsAdmin(userElements[14]);  
 u.setIsChef(userElements[15]);  
 u.setLatitude(userElements[16]);  
 u.setLongitude(userElements[17]);  
 u.setFoodType(userElements[18]);  
 u.setPicutreFromSql(userElements[19]);  
  
 users.add(u);  
 }  
 params[0].AddUsers(users);  
 }  
 }  
 catch (Exception e)  
 {  
 params[0].SetMessage("Failed. Check internet connection");  
 }  
  
 if(params[0].m\_users == null)  
 {  
 params[0].SetMessage("Failed. Check internet connection");  
 }  
  
 //Call callback  
 try  
 {  
 params[0].call();  
 }  
 catch (Exception e)  
 {  
 e.printStackTrace();  
 }  
 return "Executed";  
 }  
 }  
  
 private class ReqMealData extends AsyncTask<MealsCallback, String, String>  
 {  
 @Override  
 protected String doInBackground(MealsCallback... params)  
 {  
 try  
 {  
 URLConnection connection = new URL(m\_dbRequestUrl + params[0].GetMessage()).openConnection();  
 InputStream res = connection.getInputStream();  
  
 String webResponse = IOUtils.*toString*(res, StandardCharsets.*UTF\_8*);  
 if(webResponse!=null && new Character(webResponse.charAt(0)).equals('['))  
 {  
 ArrayList<Meal> meals = new ArrayList<>();  
  
 String[] results = ParseWebResponse(webResponse);  
  
 for (String meal: results)  
 {  
 String userElements[] = meal.split(",");  
  
 Meal m = new Meal();  
 m.setOwnerId(userElements[0]);  
 m.setName(userElements[1]);  
 m.setHalal(userElements[2].equals("true"));  
 m.setVegan(userElements[3].equals("true"));  
 m.setVegiterian(userElements[4].equals("true"));  
 m.setContainsMilk(userElements[5].equals("true"));  
 m.setContainsGluten(userElements[6].equals("true"));  
 m.setIngredients(userElements[7]);  
 m.setCalories(userElements[8]);  
 m.setPrice(userElements[9]);  
 m.setMaxNoPortions(userElements[10]);  
 m.setId(userElements[11]);  
 m.setOwnerUsername(userElements[12]);  
 m.setEatIn(userElements[13]);  
 m.setHoursAvaliableFrom(userElements[14]);  
 m.setHoursAvaliableTo(userElements[15]);  
 m.setPicutreFromSql(userElements[16]);  
 m.setRating(userElements[17]);  
  
 meals.add(m);  
 }  
 params[0].AddMeals(meals);  
 }  
 }  
 catch (Exception e)  
 {  
 params[0].SetMessage("Failed. Check internet connection");  
 }  
  
 if(params[0].m\_meals == null)  
 {  
 params[0].SetMessage("No meals. Check internet connection?");  
 }  
  
 //Call callback  
 try  
 {  
 params[0].call();  
 }  
 catch (Exception e)  
 {  
 e.printStackTrace();  
 }  
 return "Executed";  
 }  
 }  
  
 private class ReqOrderData extends AsyncTask<OrdersCallback, String, String>  
 {  
 @Override  
 protected String doInBackground(OrdersCallback... params)  
 {  
 try  
 {  
 URLConnection connection = new URL(m\_dbRequestUrl + params[0].GetMessage()).openConnection();  
 InputStream res = connection.getInputStream();  
  
 String webResponse = IOUtils.*toString*(res, StandardCharsets.*UTF\_8*);  
 if(webResponse!=null && new Character(webResponse.charAt(0)).equals('['))  
 {  
 ArrayList<Order> basketItems = new ArrayList<>();  
  
 String[] results = ParseWebResponse(webResponse);  
  
 for (String order: results)  
 {  
 String userElements[] = order.split(",");  
  
 Order o = new Order();  
 o.setId(userElements[0]);  
 o.setMealId(userElements[1]);  
 o.setChefId(userElements[2]);  
 o.setEaterId(userElements[3]);  
 o.setNumberOfPortions(userElements[4]);  
 o.setCurrentState(userElements[5]);  
 o.setIsTakeaway(userElements[6]);  
 o.setMessages\_sql(userElements[7]);  
  
 basketItems.add(o);  
 }  
 params[0].AddOrders(basketItems);  
 }  
 }  
 catch (Exception e)  
 {  
 params[0].SetMessage(e.getMessage());  
 }  
  
 try  
 {  
 params[0].call();  
 } catch (Exception e)  
 {  
 e.printStackTrace();  
 }  
 return "Executed";  
 }  
 }  
  
 private class UploadData extends AsyncTask<BaseCallback, String, String>  
 {  
 @Override  
 protected String doInBackground(BaseCallback... params)  
 {  
 try  
 {  
 URLConnection connection = new URL(m\_dbPostUrl).openConnection();  
 connection.setDoOutput(true);  
 OutputStreamWriter writer = new OutputStreamWriter(connection.getOutputStream());  
 writer.write("request="+params[0].GetMessage());  
 writer.flush();  
 InputStream res = connection.getInputStream();  
 String result = IOUtils.*toString*(res, StandardCharsets.*UTF\_8*);  
 writer.close();  
  
 params[0].SetMessage(result);  
 }  
 catch (Exception e)  
 {  
 params[0].SetMessage(e.getMessage());  
 }  
  
 try  
 {  
 params[0].call();  
 }  
 catch (Exception e)  
 {  
 e.printStackTrace();  
 }  
  
 return "Executed";  
 }  
 }  
  
 private String[] ParseWebResponse(String response)  
 {  
 //Get rid of first [  
 response = response.substring(2);  
  
 //Get rid of last ]  
 response = response.substring(0, response.length()-1);  
  
 //Put into array seperated by "["  
 String results[];  
 if(!response.contains("["))  
 {  
 results = new String[1];  
 results[0] = response;  
 }  
 else  
 {  
 results = response.split("\\[");  
 }  
  
 for (int x = 0; x < results.length; x++)  
 {  
 results[x] = results[x].replaceAll("]", "");  
 }  
  
 return results;  
 }  
}

LocalSettings class

package com.menu.menu.Classes;  
  
import android.content.Context;  
import android.content.SharedPreferences;  
  
public class LocalSettings  
{  
 private static User *m\_localUser* = null;  
  
 public static void LoadSettings(Context context)  
 {  
 *m\_localUser* = new User();  
  
 SharedPreferences userSettings = context.getSharedPreferences("UserInfo", 0);  
  
 *m\_localUser*.setUsername(userSettings.getString("Username", ""));  
 *m\_localUser*.setPassword(userSettings.getString("Password", ""));  
 }  
  
 public static void SignOutUser(Context context)  
 {  
 *SaveLoginDetails*(null, null, context);  
 *m\_localUser* = new User();  
 }  
  
 public static void UpdateLocalUser(User u)  
 {  
 *m\_localUser* = u;  
 }  
  
 public static User GetLocalUser()  
 {  
 return *m\_localUser*;  
 }  
  
 public static boolean IsLoginSaved()  
 {  
 return (*m\_localUser* != null || (*m\_localUser*.getPassword() != null && *m\_localUser*.getUsername() != null));  
 }  
  
 public static void SaveLoginDetails(String user, String pass, Context context)  
 {  
 SharedPreferences userSettings = context.getSharedPreferences("UserInfo", 0);  
 SharedPreferences.Editor editor = userSettings.edit();  
 editor.putString("Username",user);  
 editor.putString("Password",pass);  
 editor.commit();  
 }  
  
}

Meal class

package com.menu.menu.Classes;  
  
import java.io.Serializable;  
  
public class Meal extends SerializableBitmap implements Serializable  
{  
 private String m\_ownerId = null;  
 private String m\_name = "No name";  
 private Boolean m\_bIsHalal = false;  
 private Boolean m\_bIsVegan = false;  
 private Boolean m\_bIsVegiterian = false;  
 private Boolean m\_bContainsMilk = false;  
 private Boolean m\_bContainsGluten = false;  
 private String m\_ingredients = "No ingredients";  
 private String m\_calories = "0";  
 private String m\_price = "0";  
 private String m\_maxNoPortions = "0";  
 private String m\_id = null;  
 private String m\_ownerUsername = null;  
 private String m\_eatIn = "false";  
 private String m\_hoursAvaliableFrom = "0";  
 private String m\_hoursAvaliableTo = "0";  
 private String m\_rating = "5";  
 private final static String *m\_mealInsert* = "INSERT INTO [menudatabase].[dbo].[meal] (owner\_user\_id,meal\_name,is\_halal,is\_vegan,is\_vegiterian,contains\_milk,contains\_gluten,ingredients\_list,estimated\_calories,price,number\_of\_portions\_avaliable,id,ownerUsername,eatIn,hoursAvaliableFrom,hoursAvaliableTo,picture,rating) VALUES ";  
  
 public String GetInsertString()  
 {  
 final String d = "','";  
 return *m\_mealInsert* + "('" + getOwnerId() + d + getName() + d + btos(getHalal()) + d + btos(getVegan()) + d + btos(getVegiterian()) + d + btos(getContainsMilk()) + d + btos(getContainsGluten()) + d +  
 getIngredients() + d + getCalories() + d + getPrice() + d + getMaxNoPortions() + "', NEWID(),'"  
 + getOwnerUsername() + d + getEatIn() + d + getHoursAvaliableFrom() + d + getHoursAvaliableTo() + d + pictureToSql() + d + getRating() + "')";  
 }  
  
 public String GetUpdateString()  
 {  
 String d = "',";  
 return "owner\_user\_id='" + getOwnerId() + d + "meal\_name='" + getName() + d + "is\_halal='" + btos(getHalal()) + d + "is\_vegan='" + btos(getVegan()) + d + "is\_vegiterian='" + btos(getVegiterian()) + d + "contains\_milk='" +  
 btos(getContainsMilk()) + d + "contains\_gluten='" + btos(getContainsGluten()) + d + "ingredients\_list='" + getIngredients() + d + "estimated\_calories='" + getCalories() + d + "price='" +  
 getPrice() + d + "number\_of\_portions\_avaliable='" + getMaxNoPortions() + d + "id='" + getId() + d + "OwnerUsername='" + getOwnerUsername() + d + "eatIn='" + getEatIn() +  
 d + "hoursAvaliableFrom='" + getHoursAvaliableFrom() + d + "hoursAvaliableTo='" + getHoursAvaliableTo() + d + "picture='" + pictureToSql() + d + "rating='" + getRating() + "'";  
 }  
  
 private String btos(Boolean b)  
 {  
 return b ? "true" : "false";  
 }  
  
 public void SetEatIn(boolean eatIn, boolean takeaway)  
 {  
 if(eatIn && takeaway)  
 {  
 setEatIn("BOTH");  
 }  
 else if(eatIn)  
 {  
 setEatIn("EAT-IN");  
 }  
 else  
 {  
 setEatIn("TAKEAWAY");  
 }  
 }  
  
 public Boolean CurrentlyOnSale()  
 {  
 //*todo* return true;  
 }  
  
 public Boolean IsEatIn()  
 {  
 return (getEatIn().equals("BOTH") || getEatIn().equals("EAT-IN"));  
 }  
  
 public Boolean IsTakeaway()  
 {  
 return (getEatIn().equals("BOTH") || getEatIn().equals("TAKEAWAY"));  
 }  
  
 public String getOwnerId()  
 {  
 return m\_ownerId;  
 }  
  
 public void setOwnerId(String ownerId)  
 {  
 m\_ownerId = ownerId;  
 }  
  
 public String getName()  
 {  
 return m\_name;  
 }  
  
 public void setName(String name)  
 {  
 m\_name = name;  
 }  
  
 public Boolean getHalal()  
 {  
 return m\_bIsHalal;  
 }  
  
 public void setHalal(Boolean halal)  
 {  
 m\_bIsHalal = halal;  
 }  
  
 public Boolean getVegan()  
 {  
 return m\_bIsVegan;  
 }  
  
 public void setVegan(Boolean vegan)  
 {  
 m\_bIsVegan = vegan;  
 }  
  
 public Boolean getVegiterian()  
 {  
 return m\_bIsVegiterian;  
 }  
  
 public void setVegiterian(Boolean vegiterian)  
 {  
 m\_bIsVegiterian = vegiterian;  
 }  
  
 public Boolean getContainsMilk()  
 {  
 return m\_bContainsMilk;  
 }  
  
 public void setContainsMilk(Boolean containsMilk)  
 {  
 m\_bContainsMilk = containsMilk;  
 }  
  
 public Boolean getContainsGluten()  
 {  
 return m\_bContainsGluten;  
 }  
  
 public void setContainsGluten(Boolean containsGluten)  
 {  
 m\_bContainsGluten = containsGluten;  
 }  
  
 public String getIngredients()  
 {  
 String retval = "";  
 for(int i = 0; i < m\_ingredients.length(); i++)  
 {  
 if(m\_ingredients.charAt(i) == 92)  
 {  
 retval += "\n";  
 i++;  
 }  
 else  
 {  
 retval += m\_ingredients.charAt(i);  
 }  
 }  
  
 return retval;  
 }  
  
 public void setIngredients(String ingredients)  
 {  
 m\_ingredients = ingredients;  
 }  
  
 public String getCalories()  
 {  
 return m\_calories;  
 }  
  
 public void setCalories(String calories)  
 {  
 m\_calories = calories;  
 }  
  
 public String getPrice()  
 {  
 return m\_price;  
 }  
  
 public void setPrice(String price)  
 {  
 m\_price = price;  
 }  
  
 public String getMaxNoPortions()  
 {  
 return m\_maxNoPortions;  
 }  
  
 public int getMaxNoPortions\_n()  
 {  
 try  
 {  
 return Integer.*parseInt*(m\_maxNoPortions);  
 }  
 catch (Exception e)  
 {  
 return 0;  
 }  
 }  
  
 public void setMaxNoPortions(String maxNoPortions)  
 {  
 m\_maxNoPortions = maxNoPortions;  
 }  
  
 public String getId()  
 {  
 return m\_id;  
 }  
  
 public void setId(String id)  
 {  
 m\_id = id;  
 }  
  
 public String getOwnerUsername()  
 {  
 return m\_ownerUsername;  
 }  
  
 public void setOwnerUsername(String ownerUsername)  
 {  
 m\_ownerUsername = ownerUsername;  
 }  
  
 public String getEatIn()  
 {  
 return m\_eatIn;  
 }  
  
 public void setEatIn(String eatIn)  
 {  
 m\_eatIn = eatIn;  
 }  
  
 public String getHoursAvaliableFrom()  
 {  
 return m\_hoursAvaliableFrom;  
 }  
  
 public void setHoursAvaliableFrom(String hoursAvaliableFrom)  
 {  
 m\_hoursAvaliableFrom = hoursAvaliableFrom;  
 }  
  
 public String getHoursAvaliableTo()  
 {  
 return m\_hoursAvaliableTo;  
 }  
  
 public void setHoursAvaliableTo(String hoursAvaliableTo)  
 {  
 m\_hoursAvaliableTo = hoursAvaliableTo;  
 }  
  
 public String getRating()  
 {  
 if(m\_rating == null || m\_rating.equals("null"))  
 return "0";  
 return m\_rating;  
 }  
  
 public void setRating(String rating)  
 {  
 m\_rating = rating;  
 }  
}

MealListViewItem class

package com.menu.menu.Classes;  
  
import android.content.Context;  
import android.graphics.Bitmap;  
import android.graphics.Color;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.menu.menu.R;  
import java.util.ArrayList;  
  
//Class used to create a corresponding UI element for each ic\_meal in an arraylist of meals  
//These UI elements are then added into displayList  
public class MealListViewItem extends ArrayAdapter<Meal>  
{  
 ArrayList<Meal> m\_mealListRef;  
  
 public MealListViewItem(Context c, ArrayList<Meal> mealListRef)  
 {  
 super(c, R.layout.*layout\_meal\_info*, mealListRef);  
 m\_mealListRef = mealListRef;  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent)  
 {  
 //get view from activity\_forms.xml  
 View itemView = convertView;  
 if (itemView == null)  
 {  
 itemView = (LayoutInflater.*from*(getContext())).inflate(R.layout.*layout\_meal\_info*, parent, false);  
 }  
  
 Meal currentMeal = m\_mealListRef.get(position);  
  
 //Subject text  
 TextView subjectText = itemView.findViewById(R.id.*li\_text*);  
 subjectText.setText(currentMeal.getName());  
  
 //On sale  
 TextView onSale = itemView.findViewById(R.id.*li\_infoRight*);  
 if (currentMeal.CurrentlyOnSale())  
 {  
 onSale.setText("On Sale");  
 onSale.setTextColor(Color.*GREEN*);  
 }  
 else  
 {  
 onSale.setText("Marked Unavailable");  
 onSale.setTextColor(Color.*RED*);  
 }  
  
 //Price  
 ((TextView)itemView.findViewById(R.id.*li\_infoLeft*)).setText(currentMeal.getRating() + "/5");  
  
 //Img  
 ImageView img = itemView.findViewById(R.id.*img*);  
 Bitmap bmp = currentMeal.getPicture();  
 if(bmp!=null)  
 img.setImageBitmap(bmp);  
 else  
 img.setImageResource(R.drawable.*ic\_meal*);  
  
 return itemView;  
 }  
}

MealsCallback class

package com.menu.menu.Classes;  
  
import java.util.ArrayList;  
  
public class MealsCallback extends BaseCallback  
{  
 protected ArrayList<Meal> m\_meals = null;  
 public void AddMeals(ArrayList<Meal> m)  
 {  
 m\_meals = m;  
 }  
}

Order class

package com.menu.menu.Classes;  
  
import java.util.ArrayList;  
  
public class Order  
{  
 private String m\_id = "NEWID()";  
 private String m\_mealId = null;  
 private Meal m\_meal = null;  
 private String m\_mealChefId = null;  
 private String m\_mealEaterId = null;  
 private String m\_numOfPortions = null;  
 private String m\_currentState = null;  
 private String m\_isTakeaway = null;  
 private ArrayList<String> m\_messages = new ArrayList<>();  
  
 public Order()  
 {  
 }  
  
 public Order(Meal meal, String mealChefId, String mealEaterId, String numberOfPortions, boolean isTakeaway)  
 {  
 m\_mealId = meal.getId();  
 m\_meal = meal;  
 m\_mealChefId = mealChefId;  
 m\_mealEaterId = mealEaterId;  
 m\_numOfPortions = numberOfPortions;  
 setIsTakeaway\_b(isTakeaway);  
 }  
  
 private final static String *m\_startInsert* = "INSERT INTO " + DatabaseCommunicator.*m\_orderTable* + " (id, meal\_id, meal\_chef\_id, meal\_eater\_id, num\_portions\_ordered, current\_state, is\_takeaway, messages) VALUES ";  
 public String getInsertString()  
 {  
 final String d = "','";  
 return *m\_startInsert* + "(" + m\_id + ",'" + m\_mealId + d + m\_mealChefId + d + m\_mealEaterId + d + m\_numOfPortions + d + m\_currentState + d + m\_isTakeaway + d + getMessages\_sql() + "');";  
 }  
  
 public String getUpdateString()  
 {  
 final String d = "','";  
 return "UPDATE" + DatabaseCommunicator.*m\_orderTable* + " SET meal\_id='"+m\_mealId+"', meal\_chef\_id='"+m\_mealChefId+"', meal\_eater\_id='"+m\_mealEaterId+"', num\_portions\_ordered='"+m\_numOfPortions+"', current\_state='"+m\_currentState+"', is\_takeaway='"+m\_isTakeaway+"', messages='" + getMessages\_sql() + "' WHERE id='"+m\_id+"'";  
 }  
  
 public String getId()  
 {  
 return m\_id;  
 }  
  
 public void setId(String id)  
 {  
 this.m\_id = id;  
 }  
  
 public void setNumberOfPortions(String numOfPortions)  
 {  
 m\_numOfPortions = numOfPortions;  
 }  
  
 public void setNumberOfPortions(int numberOfPortions)  
 {  
 m\_numOfPortions = String.*valueOf*(numberOfPortions);  
 }  
  
 public String getNumberOfPortions()  
 {  
 return m\_numOfPortions;  
 }  
  
 public int getNumberOfPortions\_n()  
 {  
 if(m\_numOfPortions == null)  
 return 0;  
  
 return Integer.*parseInt*(m\_numOfPortions);  
 }  
  
 public String getMealId()  
 {  
 return m\_mealId;  
 }  
  
 public void setMealId(String mealId)  
 {  
 m\_mealId = mealId;  
 }  
  
 public String getCurrentState()  
 {  
 return m\_currentState;  
 }  
  
 public void setCurrentState(String currentState)  
 {  
 this.m\_currentState = currentState;  
 }  
  
 public void setEaterId(String id)  
 {  
 m\_mealEaterId = id;  
 }  
  
 public String getEaterId()  
 {  
 return m\_mealEaterId;  
 }  
  
 public boolean getIsTakeaway()  
 {  
 return m\_isTakeaway.equals("true");  
 }  
  
 public void setIsTakeaway\_b(Boolean isTakeaway)  
 {  
 m\_isTakeaway = isTakeaway ? "true" : "false";  
 }  
  
 public void setIsTakeaway(String isTakeaway)  
 {  
 m\_isTakeaway = isTakeaway;  
 }  
  
 public void setChefId(String id)  
 {  
 m\_mealChefId = id;  
 }  
  
 public String getChefId()  
 {  
 return m\_mealChefId;  
 }  
  
 public enum State  
 {  
 *AwatingResponse*,  
 *InBasket*,  
 *BeingMade*,  
 *OnRoute*,  
 *Delivered* }  
  
 public State getState\_c()  
 {  
 switch (getCurrentState())  
 {  
 case "Awaiting Response From Chef":  
 return State.*AwatingResponse*;  
 case "Being Made":  
 return State.*BeingMade*;  
 case "On Route":  
 return State.*OnRoute*;  
 case "Delivered":  
 return State.*Delivered*;  
 case "In Basket":  
 return State.*InBasket*;  
 default:  
 return null;  
 }  
 }  
  
 public void setState\_c(State cState)  
 {  
 switch (cState)  
 {  
 case *AwatingResponse*:  
 SetState("Awaiting Response From Chef");  
 break;  
 case *BeingMade*:  
 SetState("Being Made");  
 break;  
 case *OnRoute*:  
 SetState("On Route");  
 break;  
 case *Delivered*:  
 SetState("Delivered");  
 break;  
 case *InBasket*:  
 SetState("In Basket");  
 break;  
 }  
 }  
  
 public String GetState()  
 {  
 return getCurrentState();  
 }  
  
 public void SetState(String currentState)  
 {  
 setCurrentState(currentState);  
 }  
  
 public void setMessages\_sql(String stringList)  
 {  
 String[] messages = stringList.split(" --- ");  
 for (String m : messages)  
 {  
 m\_messages.add(m);  
 }  
 }  
  
 public void setMessages(ArrayList<String> list)  
 {  
 m\_messages = list;  
 }  
  
 public ArrayList<String> getMessages()  
 {  
 return m\_messages;  
 }  
  
 public String getMessages\_sql()  
 {  
 if(m\_messages.size() == 0)  
 return "null";  
  
 String combinedMessages = "";  
 for (String s : m\_messages)  
 {  
 combinedMessages += s + " --- ";  
 }  
  
 return combinedMessages.substring(0, combinedMessages.length() - 5);  
 }  
  
 public void setMeal(Meal meal)  
 {  
 m\_meal = meal;  
 }  
  
 public Meal getMeal()  
 {  
 return m\_meal;  
 }  
}

OrdersCallback class

package com.menu.menu.Classes;  
  
import java.util.ArrayList;  
  
public class OrdersCallback extends BaseCallback  
{  
 protected ArrayList<Order> m\_orders = null;  
 public void AddOrders(ArrayList<Order> u)  
 {  
 m\_orders = u;  
 }  
}

SerializableBitmap class

package com.menu.menu.Classes;  
  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.util.Base64;  
  
import java.io.ByteArrayOutputStream;  
import java.io.IOException;  
import java.io.Serializable;  
  
public class SerializableBitmap implements Serializable  
{  
 private Bitmap m\_picture;  
  
 public String pictureToSql()  
 {  
 if(m\_picture == null)  
 {  
 return "null";  
 }  
  
 ByteArrayOutputStream baos = new ByteArrayOutputStream();  
 m\_picture.compress(Bitmap.CompressFormat.*JPEG*, 10, baos);  
 String temp = Base64.*encodeToString*(baos.toByteArray(), Base64.*URL\_SAFE*);  
 temp = temp.replaceAll("\n", "^");  
 return temp;  
 }  
  
 public void setPicutreFromSql(String sql)  
 {  
 if (sql.equals("null"))  
 {  
 m\_picture = null;  
 }  
 else  
 {  
 sql = sql.replaceAll("^","\n");  
  
 try  
 {  
 sql = new String(sql.getBytes());  
 byte[] decodedString = Base64.*decode*(sql, Base64.*URL\_SAFE*);  
 m\_picture = BitmapFactory.*decodeByteArray*(decodedString, 0, decodedString.length);  
 }  
 catch (Exception e)  
 {  
 m\_picture = null;  
 }  
 }  
 }  
  
 public Bitmap getPicture()  
 {  
 return m\_picture;  
 }  
  
 public void setPicture(Bitmap b)  
 {  
 m\_picture = b;  
 }  
  
 // Converts the Bitmap into a byte array for serialization  
 private void writeObject(java.io.ObjectOutputStream out) throws IOException  
 {  
 if(m\_picture != null)  
 {  
 ByteArrayOutputStream byteStream = new ByteArrayOutputStream();  
 m\_picture.compress(Bitmap.CompressFormat.*PNG*, 0, byteStream);  
 byte bitmapBytes[] = byteStream.toByteArray();  
 out.write(bitmapBytes, 0, bitmapBytes.length);  
 }  
 }  
  
 // Deserializes a byte array representing the Bitmap and decodes it  
 private void readObject(java.io.ObjectInputStream in) throws IOException, ClassNotFoundException  
 {  
 ByteArrayOutputStream byteStream = new ByteArrayOutputStream();  
 int b;  
 while((b = in.read()) != -1)  
 byteStream.write(b);  
 byte bitmapBytes[] = byteStream.toByteArray();  
 m\_picture = BitmapFactory.*decodeByteArray*(bitmapBytes, 0, bitmapBytes.length);  
 }  
}

User class

package com.menu.menu.Classes;  
  
import java.io.Serializable;  
  
public class User extends SerializableBitmap implements Serializable  
{  
 private String m\_id = "NEWID()";  
 private String m\_username = null; //*todo testing...* private String m\_password = null;  
 private String m\_fullName = null;  
 private String m\_addressLine1 = null;  
 private String m\_addressLine2 = null;  
 private String m\_addressLine3 = null;  
 private String m\_addressPostCode = null;  
 private String m\_addressDescription = null;  
 private String m\_DOB = null;  
 private String m\_loggedIn = null;  
 private String m\_email = null;  
 private String m\_phone = null;  
 private String m\_rating = "5";  
 private String m\_isAdmin = null;  
 private String m\_latitude = null;  
 private String m\_longitude = null;  
 private String m\_isChef = null;  
 private String m\_foodType = null;  
 private final static String *m\_userInsert* = "INSERT INTO [menudatabase].[dbo].[User] (id,name,password,full\_name,address\_line\_1,address\_line\_2,address\_city,address\_post\_code,address\_description,date\_of\_birth,logged\_in,contact\_email,contact\_phone,rating,is\_admin,is\_chef,latitude,longitude,food\_type,picture\_id) VALUES ";  
  
 public String GetInsertString()  
 {  
 final String d = "','";  
 return *m\_userInsert* + "(" + getId() + ",'" + getUsername() + d + getPassword() + d + getFullName() + d + getAddressLine1() + d + getAddressLine2() + d + getAddressLine3() + d +  
 getAddressPostCode() + d + getAddressDescription() + d + getDOB() + d + getLoggedIn() + d + getEmail() + d + getPhone() + d + getRating() + d + getIsAdmin() + d + getIsChef() + d + getLatitude() + d + getLongitude() + d + getFoodType() + d + pictureToSql() + "')";  
 }  
  
 public String GetUpdateString()  
 {  
 String d = "',";  
 return "name='" + getUsername() + d + "password='" + getPassword() + d + "full\_name='" + getFullName() + d + "address\_line\_1='" + getAddressLine1() + d + "address\_line\_2='" + getAddressLine2() + d + "address\_city='" +  
 getAddressLine3() + d + "address\_post\_code='" + getAddressPostCode() + d + "address\_description='" + getAddressDescription() + d + "date\_of\_birth='" + getDOB() + d + "logged\_in='" +  
 getLoggedIn() + d + "contact\_email='" + getEmail() + d + "contact\_phone='" + getPhone() + d + "rating='" + getRating() + d + "is\_admin='" + getIsAdmin() +  
 d + "picture\_id='" + pictureToSql() + d + "is\_chef='" + getIsChef() + d + "latitude='" + getLatitude() + d + "longitude='" + getLongitude() + d + "food\_type='" + getFoodType() + "'";  
 }  
  
 public String getId() {  
 return m\_id;  
 }  
  
 public void setId(String id) {  
 m\_id = id;  
 }  
  
 public String getUsername() {  
 return m\_username;  
 }  
  
 public void setUsername(String username) {  
 m\_username = username;  
 }  
  
 public String getPassword() {  
 return m\_password;  
 }  
  
 public void setPassword(String password) {  
 m\_password = password;  
 }  
  
 public String getFullName() {  
 return m\_fullName;  
 }  
  
 public void setFullName(String fullName) {  
 m\_fullName = fullName;  
 }  
  
 public String getAddressLine1() {  
 return m\_addressLine1;  
 }  
  
 public void setAddressLine1(String addressLine1) {  
 m\_addressLine1 = addressLine1;  
 }  
  
 public String getAddressLine2() {  
 return m\_addressLine2;  
 }  
  
 public void setAddressLine2(String addressLine2) {  
 m\_addressLine2 = addressLine2;  
 }  
  
 public String getAddressLine3() {  
 return m\_addressLine3;  
 }  
  
 public void setAddressLine3(String addressLine3) {  
 m\_addressLine3 = addressLine3;  
 }  
  
 public String getAddressPostCode() {  
 return m\_addressPostCode;  
 }  
  
 public void setAddressPostCode(String addressPostCode) {  
 m\_addressPostCode = addressPostCode;  
 }  
  
 public String getAddressDescription() {  
 return m\_addressDescription;  
 }  
  
 public void setAddressDescription(String addressDescription) {  
 m\_addressDescription = addressDescription;  
 }  
  
 public String getDOB() {  
 return m\_DOB;  
 }  
  
 public void setDOB(String DOB)  
 {  
 this.m\_DOB = DOB;  
 }  
  
 public String getLoggedIn() {  
 return m\_loggedIn;  
 }  
  
 public void setLoggedIn(String loggedIn) {  
 m\_loggedIn = loggedIn;  
 }  
  
 public String getEmail() {  
 return m\_email;  
 }  
  
 public void setEmail(String email) {  
 m\_email = email;  
 }  
  
 public String getPhone() {  
 return m\_phone;  
 }  
  
 public void setPhone(String phone) {  
 m\_phone = phone;  
 }  
  
 public String getRating() {  
 return m\_rating;  
 }  
  
 public void setRating(String rating) {  
 m\_rating = rating;  
 }  
  
 public String getIsAdmin() {  
 return m\_isAdmin;  
 }  
  
 public Boolean getIsAdmin\_b()  
 {  
 return m\_isAdmin.equals("true");  
 }  
  
 public void setIsAdmin(String isAdmin)  
 {  
 m\_isAdmin = isAdmin;  
 }  
  
 public void setIsAdmin\_b(Boolean isAdmin)  
 {  
 m\_isAdmin = isAdmin ? "true" : "false";  
 }  
  
 public String getLatitude() {  
 return m\_latitude;  
 }  
  
 public void setLatitude(String latitude) {  
 m\_latitude = latitude;  
 }  
  
 public String getLongitude() {  
 return m\_longitude;  
 }  
  
 public void setLongitude(String longitude) {  
 m\_longitude = longitude;  
 }  
  
 public String getIsChef() {  
 return m\_isChef;  
 }  
  
 public Boolean getIsChef\_b()  
 {  
 if(m\_isChef == null)  
 return false;  
 return m\_isChef.equals("true");  
 }  
  
 public void setIsChef(String isChef) {  
 m\_isChef = isChef;  
 }  
  
 public void setIsChef\_b(Boolean isChef)  
 {  
 m\_isChef = isChef ? "true" : "false";  
 }  
  
 public String getFoodType() {  
 return m\_foodType;  
 }  
  
 public void setFoodType(String foodType) {  
 m\_foodType = foodType;  
 }  
}

UsersCallback class

package com.menu.menu.Classes;  
  
import java.util.ArrayList;  
  
public class UsersCallback extends BaseCallback  
{  
 protected ArrayList<User> m\_users = null;  
 public boolean m\_bInternetIssue = true;  
 public void AddUsers(ArrayList<User> u)  
 {  
 m\_users = u;  
 }  
}

AddressEdit class

package com.menu.menu;  
  
import android.location.Address;  
import android.location.Geocoder;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.User;  
  
public class AddressEdit extends AppCompatActivity  
{  
 User m\_currentUser = LocalSettings.*GetLocalUser*();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_address\_edit*);  
  
 final EditText input\_addressLine1 = findViewById(R.id.*input\_addressLine1*);  
 final EditText input\_addressLine2 = findViewById(R.id.*input\_addressLine2*);  
 final EditText input\_addressLine3 = findViewById(R.id.*input\_addressLine3*);  
 final EditText input\_postCode = findViewById(R.id.*input\_postCode*);  
  
 input\_addressLine1.setText((m\_currentUser.getAddressLine1() != null) ? m\_currentUser.getAddressLine1() : "");  
 input\_addressLine2 .setText((m\_currentUser.getAddressLine2() != null) ? m\_currentUser.getAddressLine2() : "");  
 input\_addressLine3.setText((m\_currentUser.getAddressLine3() != null) ? m\_currentUser.getAddressLine3() : "");  
 input\_postCode.setText((m\_currentUser.getAddressPostCode() != null) ? m\_currentUser.getAddressPostCode() : "");  
  
 findViewById(R.id.*btn\_save*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 final String addLn1 = input\_addressLine1.getText().toString();  
 final String addLn2 = input\_addressLine2.getText().toString();  
 final String addLn3 = input\_addressLine3.getText().toString();  
 final String addPostCode = input\_postCode.getText().toString();  
  
 try  
 {  
 Geocoder coder = new Geocoder(AddressEdit.this);  
 Address address = coder.getFromLocationName(addLn1 + " " + addLn2 + " " + addLn3 + " " + addPostCode,1).get(0);  
  
 m\_currentUser.setLatitude(String.*valueOf*(address.getLatitude()) );  
 m\_currentUser.setLongitude(String.*valueOf*(address.getLongitude()));  
 m\_currentUser.setAddressLine1(addLn1);  
 m\_currentUser.setAddressLine2(addLn2);  
 m\_currentUser.setAddressLine3(addLn3);  
 m\_currentUser.setAddressPostCode(addPostCode);  
 LocalSettings.*UpdateLocalUser*(m\_currentUser);  
 onBackPressed();  
 }  
 catch (Exception e)  
 {  
 e.printStackTrace();  
 SetError("Incorrect address");  
 }  
 }  
 });  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(AddressEdit.this, errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
}

Basket class

package com.menu.menu;  
  
import android.content.Context;  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.BaseCallback;  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.Meal;  
import com.menu.menu.Classes.MealsCallback;  
import com.menu.menu.Classes.Order;  
  
import java.util.ArrayList;  
  
public class Basket extends Fragment  
{  
 private static ArrayList<Order> *m\_basketItems* = new ArrayList<>();  
 public static void addOrder(Order order)  
 {  
 *m\_basketItems*.add(order);  
 }  
  
 private View.OnClickListener m\_drawerListener;  
 private ListView m\_displayList;  
  
 public void SetDrawerButtonListner(View.OnClickListener listener)  
 {  
 m\_drawerListener = listener;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState)  
 {  
 View root = inflater.inflate(R.layout.*fragment\_basket*, container, false);  
 root.findViewById(R.id.*drawerToggle*).setOnClickListener(m\_drawerListener);  
 m\_displayList = root.findViewById(R.id.*listView\_basket*);  
 m\_displayList.setOnItemClickListener(new AdapterView.OnItemClickListener()  
 {  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id)  
 {  
 Intent intent = new Intent(getContext(), MealView.class);  
 intent.putExtra("meal", *m\_basketItems*.get(position).getMealId());  
 startActivity(intent);  
 }  
 });  
  
 root.findViewById(R.id.*btn\_checkout*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 if(*m\_basketItems*.isEmpty())  
 {  
 SetError("No items to order.");  
 }  
 else  
 {  
 DatabaseCommunicator dbComms = new DatabaseCommunicator();  
 for(Order order : *m\_basketItems*)  
 {  
 UploadOrderCallback uoc = new UploadOrderCallback();  
 uoc.setOrder(order);  
 uoc.SetMessage(order.getInsertString());  
 dbComms.GenericUpload(uoc);  
 }  
 }  
 }  
 });  
  
 Update();  
  
 return root;  
 }  
  
 private void Update()  
 {  
 BasicUpdate();  
 DatabaseCommunicator dbComms = new DatabaseCommunicator();  
  
 for (Order basketItem : *m\_basketItems*)  
 {  
 UpdateMealOfOrder umoo = new UpdateMealOfOrder();  
 umoo.SetMessage("SELECT \* FROM " + dbComms.*m\_mealTable* + " WHERE id='"+ basketItem.getMealId()+"';");  
 dbComms.RequestMealData(umoo);  
 }  
 }  
  
 private void BasicUpdate()  
 {  
 m\_displayList.setAdapter(new BasketListViewItem(getContext(), *m\_basketItems*));  
 }  
  
 private class UpdateMealOfOrder extends MealsCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 ((MainHub)getActivity()).runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(!m\_meals.isEmpty())  
 {  
 for (Order basketItem : *m\_basketItems*)  
 {  
 if (basketItem.getMealId().equals(m\_meals.get(0).getId()))  
 {  
 basketItem.setMeal(m\_meals.get(0));  
 }  
 }  
 m\_displayList.setAdapter(new BasketListViewItem(getContext(), *m\_basketItems*));  
 }  
 else  
 {  
 SetError("Failed to update an order!");  
 }  
 }  
 });  
 return null;  
 }  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast.*makeText*(getActivity(), errorString, Toast.*LENGTH\_LONG*).show();  
 }  
  
 class UploadOrderCallback extends BaseCallback  
 {  
 private Order m\_order;  
  
 @Override  
 public Void call() throws Exception  
 {  
 getActivity().runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(m\_message.equals("null"))  
 {  
 *m\_basketItems*.remove(m\_order);  
 m\_displayList.setAdapter(new BasketListViewItem(getContext(), *m\_basketItems*));  
  
 if(*m\_basketItems*.size() == 0)  
 {  
 ((MainHub)getActivity()).NavigateToFragment(MainHub.*MyOrdersFragmentTag*);  
 }  
 }  
 else  
 {  
 SetError("Failed to request order, check internet.");  
 }  
 }  
 });  
 return null;  
 }  
  
 public void setOrder(Order order)  
 {  
 m\_order = order;  
 }  
 }  
  
 //Class used to create a corresponding UI element for each ic\_meal in an arraylist of meals  
 //These UI elements are then added into displayList  
 class BasketListViewItem extends ArrayAdapter<Order>  
 {  
 ArrayList<Order> m\_basketItemListRef;  
  
 public BasketListViewItem(Context c, ArrayList<Order> basketItemListRef)  
 {  
 super(c, R.layout.*layout\_meal\_info*, basketItemListRef);  
 m\_basketItemListRef = basketItemListRef;  
 }  
  
 @Override  
 public View getView(final int position, View convertView, ViewGroup parent)  
 {  
 //get view from activity\_forms.xml  
 View itemView = convertView;  
 if (itemView == null)  
 {  
 itemView = (LayoutInflater.*from*(getContext())).inflate(R.layout.*layout\_basket\_item*, parent, false);  
 }  
  
 final Order currentBasketItem = m\_basketItemListRef.get(position);  
 final Meal currentMeal = currentBasketItem.getMeal();  
  
 //Subject text  
 final TextView txt\_subject = itemView.findViewById(R.id.*li\_top*);  
 txt\_subject.setText(currentMeal.getName());  
  
 //Ammount  
 final TextView txt\_orderAmount = itemView.findViewById(R.id.*li\_amount*);  
 txt\_orderAmount.setText(currentBasketItem.getNumberOfPortions());  
  
 //Price  
 String price = currentMeal.getPrice();  
 Double cost = 0.00;  
 try  
 {  
 cost = Double.*parseDouble*(price) \* currentBasketItem.getNumberOfPortions\_n();  
 }  
 catch (Exception e)  
 {  
 }  
 ((TextView)itemView.findViewById(R.id.*li\_text*)).setText(currentMeal.getPrice() + "£");  
  
 //Img  
 ImageView img = itemView.findViewById(R.id.*img*);  
 Bitmap bmp = currentMeal.getPicture();  
 if(bmp!=null)  
 img.setImageBitmap(bmp);  
 else  
 img.setImageResource(R.drawable.*ic\_meal*);  
  
 itemView.findViewById(R.id.*btn\_portionsAdd*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 int mealNumber = currentBasketItem.getNumberOfPortions\_n();  
 if(mealNumber < currentBasketItem.getMeal().getMaxNoPortions\_n())  
 {  
 mealNumber+=1;  
 currentBasketItem.setNumberOfPortions(mealNumber);  
 txt\_orderAmount.setText(currentBasketItem.getNumberOfPortions());  
 }  
 BasicUpdate();  
 }  
 });  
  
 itemView.findViewById(R.id.*btn\_portionsSub*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 int mealNumber = currentBasketItem.getNumberOfPortions\_n();  
 if(mealNumber > 0)  
 {  
 mealNumber-=1;  
 currentBasketItem.setNumberOfPortions(mealNumber);  
 }  
 BasicUpdate();  
 }  
 });  
  
 itemView.findViewById(R.id.*btn\_remove*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 m\_basketItemListRef.remove(position);  
 BasicUpdate();  
 }  
 });  
  
 return itemView;  
 }  
 }  
}

ChefAccountSettings class

package com.menu.menu;  
  
  
import android.app.TimePickerDialog;  
//import android.content.Intent;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.widget.TimePicker;  
//import com.menu.menu.Classes.BaseCallback;  
//import com.menu.menu.Classes.DatabaseCommunicator;  
  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.User;  
  
public class ChefAccountSettings extends AppCompatActivity  
{  
 User m\_currentUser = LocalSettings.*GetLocalUser*();  
  
 TextView m\_txt\_pickDateFrom = null;  
 TextView m\_txt\_pickDateTo = null;  
 TextView m\_txt\_hoursAvaliableFrom = null;  
 TextView m\_txt\_hoursAvaliableTo = null;  
 TimePickerDialog.OnTimeSetListener m\_onHoursFromSetListener = null;  
 TimePickerDialog.OnTimeSetListener m\_onHoursToSetListener = null;  
 String m\_hoursAvaliableFrom = null;  
 String m\_hoursAvaliableTo = null;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chef\_account\_settings*);  
  
 final Switch switch\_chef = findViewById(R.id.*switch\_chef*);  
 final EditText input\_foodType = findViewById(R.id.*input\_foodType*);  
  
 m\_txt\_pickDateFrom = findViewById(R.id.*txt\_chooseDateFrom*);  
 m\_txt\_pickDateTo = findViewById(R.id.*txt\_chooseDateTo*);  
 m\_txt\_hoursAvaliableFrom = findViewById(R.id.*txt\_hoursAvaliableFrom*);  
 m\_txt\_hoursAvaliableTo = findViewById(R.id.*txt\_hoursAvaliableTo*);  
  
 //May remove...  
 //m\_txt\_hoursAvaliableFrom.setVisibility(View.INVISIBLE);  
 //m\_txt\_hoursAvaliableTo.setVisibility(View.INVISIBLE);  
  
 if(m\_currentUser != null)  
 {  
 Boolean isChef = m\_currentUser.getIsChef\_b();  
 String foodType = m\_currentUser.getFoodType();  
 if (foodType != null)  
 input\_foodType.setText(foodType);  
 else  
 input\_foodType.setText("");  
 switch\_chef.setChecked(isChef);  
  
 if(!isChef)  
 {  
 input\_foodType.setVisibility(View.*INVISIBLE*);  
 findViewById(R.id.*txt\_foodSpeciality*).setVisibility(View.*INVISIBLE*);  
 m\_txt\_pickDateFrom.setVisibility(View.*INVISIBLE*);  
 m\_txt\_pickDateTo.setVisibility(View.*INVISIBLE*);  
 m\_txt\_hoursAvaliableFrom.setVisibility(View.*INVISIBLE*);  
 m\_txt\_hoursAvaliableTo.setVisibility(View.*INVISIBLE*);  
 }  
 }  
  
 findViewById(R.id.*btn\_save*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 m\_currentUser.setIsChef\_b(switch\_chef.isChecked());  
  
 if(switch\_chef.isChecked())  
 {  
 if(input\_foodType.getText() != null && input\_foodType.getText().toString() != "")  
 {  
 m\_currentUser.setFoodType(input\_foodType.getText().toString());  
  
 if(m\_hoursAvaliableFrom != null && m\_hoursAvaliableTo != null)  
 {  
 //to-do set loading bar  
 //UpdateMealCallback umc = new UpdateMealCallback();  
 //DatabaseCommunicator dbComms = new DatabaseCommunicator();  
 //umc.SetMessage("UPDATE " + dbComms.m\_mealTable + " SET hoursAvaliableFrom='" + m\_hoursAvaliableFrom + "' WHERE owner\_user\_id='" + LocalSettings.LocalUser.Id + "';");  
 //dbComms.GenericUpload(umc);  
  
 //For now...  
 LocalSettings.*UpdateLocalUser*(m\_currentUser);  
 onBackPressed();  
 }  
 else  
 {  
 SetError("Make sure both time to & from are set for meal times.");  
 }  
 }  
 else  
 {  
 SetError("Please enter in a food speciality.");  
 }  
 }  
 else  
 {  
 LocalSettings.*UpdateLocalUser*(m\_currentUser);  
 onBackPressed();  
 }  
 }  
 });  
  
 switch\_chef.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 if(switch\_chef.isChecked())  
 {  
 input\_foodType.setVisibility(View.*VISIBLE*);  
 findViewById(R.id.*txt\_foodSpeciality*).setVisibility(View.*VISIBLE*);  
 m\_txt\_pickDateFrom.setVisibility(View.*VISIBLE*);  
 m\_txt\_pickDateTo.setVisibility(View.*VISIBLE*);  
 m\_txt\_hoursAvaliableFrom.setVisibility(View.*VISIBLE*);  
 m\_txt\_hoursAvaliableTo.setVisibility(View.*VISIBLE*);  
 }  
 else  
 {  
 input\_foodType.setVisibility(View.*INVISIBLE*);  
 findViewById(R.id.*txt\_foodSpeciality*).setVisibility(View.*INVISIBLE*);  
 m\_txt\_pickDateFrom.setVisibility(View.*INVISIBLE*);  
 m\_txt\_pickDateTo.setVisibility(View.*INVISIBLE*);  
 m\_txt\_hoursAvaliableFrom.setVisibility(View.*INVISIBLE*);  
 m\_txt\_hoursAvaliableTo.setVisibility(View.*INVISIBLE*);  
 }  
 }  
 });  
  
  
 m\_onHoursFromSetListener = new TimePickerDialog.OnTimeSetListener()  
 {  
 @Override  
 public void onTimeSet(TimePicker timePicker, int i, int i1)  
 {  
 m\_hoursAvaliableFrom = (i + ":" + i1);  
 m\_txt\_pickDateFrom.setText(i + ":" + i1);  
 }  
 };  
  
 m\_onHoursToSetListener = new TimePickerDialog.OnTimeSetListener()  
 {  
 @Override  
 public void onTimeSet(TimePicker timePicker, int i, int i1)  
 {  
 m\_hoursAvaliableTo = (i + ":" + i1);  
 m\_txt\_pickDateTo.setText(i + ":" + i1);  
 }  
 };  
  
 View.OnClickListener pickDateFromListener = new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 TimePickerDialog timePickerDialog = new TimePickerDialog(ChefAccountSettings.this, m\_onHoursFromSetListener, 0, 0, true);  
 timePickerDialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*WHITE*));  
 timePickerDialog.show();  
 }  
 };  
 m\_txt\_pickDateFrom.setOnClickListener(pickDateFromListener);  
 m\_txt\_hoursAvaliableFrom.setOnClickListener(pickDateFromListener);  
  
 View.OnClickListener pickDateToListener = new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 TimePickerDialog timePickerDialog = new TimePickerDialog(ChefAccountSettings.this, m\_onHoursToSetListener, 0, 0, true);  
 timePickerDialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*WHITE*));  
 timePickerDialog.show();  
 }  
 };  
 m\_txt\_pickDateTo.setOnClickListener(pickDateToListener);  
 m\_txt\_hoursAvaliableTo.setOnClickListener(pickDateToListener);  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(ChefAccountSettings.this, errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
  
 /\*private class UpdateMealCallback extends BaseCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(m\_message.equals("null"))  
 {  
 onBackPressed();  
 }  
 else  
 {  
 SetError("Update failed! Check internet connection.");  
 }  
 }  
 });  
  
 return null;  
 }  
 }\*/  
}

ChefMeals class

package com.menu.menu;  
  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.Meal;  
import com.menu.menu.Classes.MealListViewItem;  
import com.menu.menu.Classes.MealsCallback;  
  
import java.util.ArrayList;  
  
public class ChefMeals extends AppCompatActivity  
{  
 private String m\_chefId = null;  
 private DatabaseCommunicator m\_dbComms = new DatabaseCommunicator();  
 private ArrayList<Meal> m\_mealInfoArray = new ArrayList<>();  
 private ListView m\_displayList;  
 private ProgressBar m\_progressBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_chef\_meals*);  
  
 m\_progressBar = findViewById(R.id.*progressBar*);  
  
 Bundle extras = getIntent().getExtras();  
 if(extras == null || !extras.containsKey("chefId") || !extras.containsKey("chefUsername"))  
 {  
 SetError("Error. Page loaded incorrectly.");  
 }  
 else  
 {  
 m\_chefId = extras.getString("chefId");  
 ((TextView)findViewById(R.id.*txt\_username*)).setText(extras.getString("chefUsername"));  
  
 m\_displayList = findViewById(R.id.*mealsList*);  
 UpdateList();  
  
 m\_displayList.setOnItemClickListener(new AdapterView.OnItemClickListener()  
 {  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id)  
 {  
 Intent intent = new Intent(ChefMeals.this, MealView.class);  
 Meal parcelMeal = m\_mealInfoArray.get(position);  
 MealView.*FailedBitmap* = parcelMeal.getPicture();  
 parcelMeal.setPicture(null);  
 intent.putExtra("meal", parcelMeal);  
 startActivity(intent);  
 }  
 });  
 }  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(ChefMeals.this, errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
  
 private void UpdateList()  
 {  
 m\_progressBar.startNestedScroll(1);  
 m\_progressBar.setVisibility(View.*VISIBLE*);  
  
 ChefMeals.GetMealsListCallback gmlc = new ChefMeals.GetMealsListCallback();  
 gmlc.SetMessage("SELECT \* FROM " + m\_dbComms.*m\_mealTable* + " WHERE owner\_user\_id = '" + m\_chefId + "';");  
 m\_dbComms.RequestMealData(gmlc);  
 }  
  
 private class GetMealsListCallback extends MealsCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(m\_meals != null && !m\_meals.isEmpty())  
 {  
 m\_mealInfoArray = m\_meals;  
 m\_displayList.setAdapter(new MealListViewItem(getApplicationContext(),m\_mealInfoArray));  
 }  
 else  
 {  
 SetError(m\_message);  
 }  
  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
 }  
 });  
 return null;  
 }  
 }  
}

# Home class

package com.menu.menu;  
  
import android.content.Intent;  
import android.location.Location;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.support.annotation.NonNull;  
import android.support.v4.app.Fragment;  
import android.widget.SearchView;  
import android.widget.Toast;  
  
import com.google.android.gms.location.LocationCallback;  
import com.google.android.gms.location.LocationRequest;  
import com.google.android.gms.location.LocationResult;  
import com.google.android.gms.location.LocationServices;  
import com.google.android.gms.maps.CameraUpdateFactory;  
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.MapView;  
import com.google.android.gms.maps.MapsInitializer;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.model.CameraPosition;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.Marker;  
import com.google.android.gms.maps.model.MarkerOptions;  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.User;  
import com.menu.menu.Classes.UsersCallback;  
  
import java.util.ArrayList;  
  
public class Home extends Fragment  
{  
 private MapView m\_MapView;  
 private GoogleMap m\_googleMap;  
 private View.OnClickListener m\_drawerListener;  
 private DatabaseCommunicator m\_dbComms = new DatabaseCommunicator();  
 private Marker m\_userMarker = null;  
  
 class UsernameIdPair  
 {  
 UsernameIdPair(String username, String id)  
 {  
 Username = username;  
 Id = id;  
 }  
 String Username;  
 String Id;  
 }  
 private ArrayList<UsernameIdPair> m\_markerInformaitonList = new ArrayList<>();  
  
 public void SetDrawerButtonListner(View.OnClickListener listener)  
 {  
 m\_drawerListener = listener;  
 }  
  
 public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)  
 {  
 View root = inflater.inflate(R.layout.*fragment\_home*, container, false);  
  
 //  
 //Search view  
 //  
 final SearchView searchView = root.findViewById(R.id.*searchView*);  
 searchView.setOnSearchClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 Intent intent = new Intent(getActivity(), SearchResults.class);  
 intent.putExtra("search", searchView.getQuery());  
 startActivity(intent);  
 }  
 });  
  
  
 //  
 //Map view  
 //  
 m\_MapView = root.findViewById(R.id.*mapView*);  
 m\_MapView.onCreate(savedInstanceState);  
 m\_MapView.onResume();  
  
 try  
 {  
 MapsInitializer.*initialize*(getActivity().getApplicationContext());  
 }  
 catch (Exception e)  
 {  
 e.printStackTrace();  
 }  
  
 m\_MapView.getMapAsync(new OnMapReadyCallback()  
 {  
 @Override  
 public void onMapReady(GoogleMap mMap)  
 {  
 m\_googleMap = mMap;  
  
 m\_googleMap.setOnInfoWindowClickListener(new GoogleMap.OnInfoWindowClickListener()  
 {  
 @Override  
 public void onInfoWindowClick(Marker marker)  
 {  
 for(UsernameIdPair uip : m\_markerInformaitonList)  
 {  
 if(uip.Username.equals(marker.getTitle()))  
 {  
 Intent intent = new Intent(getActivity(), ChefMeals.class);  
 intent.putExtra("chefId", uip.Id);  
 intent.putExtra("chefUsername", uip.Username);  
 startActivity(intent);  
 }  
 }  
 }  
 });  
  
 }  
 });  
  
  
 //  
 //Buttons  
 //  
 root.findViewById(R.id.*drawerToggle*).setOnClickListener(m\_drawerListener);  
  
 root.findViewById(R.id.*btn\_locate*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 UpdateLocation();  
 }  
 });  
  
 root.findViewById(R.id.*btn\_refresh*).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 UpdateList();  
 m\_userMarker = m\_googleMap.addMarker(new MarkerOptions().position(m\_userMarker.getPosition()).title("you"));  
 }  
 });  
  
  
 UpdateLocation();  
  
 UpdateList();  
  
 return root;  
 }  
  
 private void UpdateLocation()  
 {  
 LocationRequest mLocationRequest = LocationRequest.*create*();  
 mLocationRequest.setPriority(LocationRequest.*PRIORITY\_HIGH\_ACCURACY*);  
 LocationCallback mLocationCallback = new LocationCallback()  
 {  
 @Override  
 public void onLocationResult(LocationResult locationResult)  
 {  
 if (locationResult != null)  
 {  
 for (Location location : locationResult.getLocations())  
 {  
 if (location != null && m\_googleMap != null)  
 {  
 LatLng userLatLng = new LatLng(location.getLatitude(), location.getLongitude());  
  
 if(m\_userMarker != null)  
 m\_userMarker.remove();  
  
 m\_userMarker = m\_googleMap.addMarker(new MarkerOptions().position(userLatLng).title("You"));  
  
 // For zooming automatically to the location of the marker  
 CameraPosition cameraPosition = new CameraPosition.Builder().target(userLatLng).zoom(12).build();  
 m\_googleMap.animateCamera(CameraUpdateFactory.*newCameraPosition*(cameraPosition));  
 }  
 }  
 }  
  
 }  
 };  
 LocationServices.*getFusedLocationProviderClient*(getActivity()).requestLocationUpdates(mLocationRequest, mLocationCallback, null);  
 }  
  
 @Override  
 public void onResume()  
 {  
 super.onResume();  
 m\_MapView.onResume();  
 }  
  
 @Override  
 public void onPause()  
 {  
 super.onPause();  
 m\_MapView.onPause();  
 }  
  
 @Override  
 public void onDestroy()  
 {  
 super.onDestroy();  
 m\_MapView.onDestroy();  
 }  
  
 @Override  
 public void onLowMemory()  
 {  
 super.onLowMemory();  
 m\_MapView.onLowMemory();  
 }  
  
 private void UpdateList()  
 {  
 Home.GetChefsListCallback gmlc = new Home.GetChefsListCallback();  
 gmlc.SetMessage("SELECT \* FROM " + m\_dbComms.*m\_userTable* + " WHERE is\_chef = 'true';");  
 m\_dbComms.RequestUserData(gmlc);  
  
 if(m\_googleMap != null)  
 m\_googleMap.clear();  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast.*makeText*(getActivity(), errorString, Toast.*LENGTH\_LONG*).show();  
 }  
  
 private class GetChefsListCallback extends UsersCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 ((MainHub) getActivity()).runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(m\_users != null && !m\_users.isEmpty())  
 {  
 if(m\_googleMap != null)  
 {  
 for (User user : m\_users)  
 {  
 try  
 {  
 LatLng latLong = new LatLng(Float.*parseFloat*(user.getLatitude()), Float.*parseFloat*(user.getLongitude()));  
 m\_googleMap.addMarker(new MarkerOptions().position(latLong).title(user.getUsername()).snippet(user.getFoodType()));  
 m\_markerInformaitonList.add(new UsernameIdPair(user.getUsername(), user.getId()));  
 }  
 catch (Exception e)  
 {  
 SetError("At least one user failed to load!");  
 }  
 }  
 }  
 else  
 {  
 SetError("Map not loaded yet, can't fill.");  
 }  
 }  
 else  
 {  
 SetError(m\_message);  
 }  
 }  
 });  
 return null;  
 }  
 }  
}

# Loader class

package com.menu.menu;  
  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.UsersCallback;  
  
public class Loader extends AppCompatActivity  
{  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
  
 LocalSettings.*LoadSettings*(this);  
 if (LocalSettings.*IsLoginSaved*())  
 {  
 UserDataCallback ucb = new UserDataCallback();  
 DatabaseCommunicator dbComms = new DatabaseCommunicator();  
 ucb.SetMessage("SELECT \* FROM " + dbComms.*m\_userTable* + " WHERE name = '" + LocalSettings.*GetLocalUser*().getUsername() + "' and password = '" + LocalSettings.*GetLocalUser*().getPassword() + "';");  
 dbComms.RequestUserData(ucb);  
 }  
 else  
 {  
 startActivity(new Intent(Loader.this, Login.class));  
 }  
  
 setContentView(R.layout.*activity\_loader*);  
 }  
  
 private class UserDataCallback extends UsersCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 if (m\_users != null && !m\_users.isEmpty())  
 {  
 LocalSettings.*UpdateLocalUser*(m\_users.get(0));  
 startActivity(new Intent(Loader.this, MainHub.class));  
 }  
 else  
 {  
 startActivity(new Intent(Loader.this, Login.class));  
 }  
 return null;  
 }  
 }  
}

# Login class

package com.menu.menu;  
  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ProgressBar;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.UsersCallback;  
  
public class Login extends AppCompatActivity  
{  
 DatabaseCommunicator m\_dbComms = new DatabaseCommunicator();  
 private ProgressBar m\_progressBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_login*);  
  
 final Button btn\_login = findViewById(R.id.*btn\_login*);  
 final Button btn\_signUp = findViewById(R.id.*btn\_signUp*);  
 final EditText input\_usernameOrEmail = findViewById(R.id.*input\_usernameOrEmail*);  
 final EditText input\_password = findViewById(R.id.*input\_password*);  
  
 m\_progressBar = findViewById(R.id.*progressBar*);  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
  
 btn\_login.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 String usernameOrEmail = input\_usernameOrEmail.getText().toString();  
 String password = input\_password.getText().toString();  
  
 String select = "SELECT \* FROM " + m\_dbComms.*m\_userTable* + " WHERE name = '" + usernameOrEmail + "' and password = '" + password + "' or contact\_email = '"  
 + usernameOrEmail + "' and password = '" + password + "';";  
  
 UserDataCallback ucb = new UserDataCallback();  
 ucb.SetMessage(select);  
 m\_dbComms.RequestUserData(ucb);  
  
 m\_progressBar.startNestedScroll(1);  
 m\_progressBar.setVisibility(View.*VISIBLE*);  
 }  
 });  
  
 btn\_signUp.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 startActivity(new Intent(Login.this, SignUp.class));  
 }  
 });  
 }  
  
 @Override  
 public void onBackPressed()  
 {  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(Login.this, errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
  
 private class UserDataCallback extends UsersCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if (m\_users != null && !m\_users.isEmpty()) //Correct user found  
 {  
 LocalSettings.*UpdateLocalUser*(m\_users.get(0));  
 LocalSettings.*SaveLoginDetails*(m\_users.get(0).getUsername(), m\_users.get(0).getPassword(), Login.this);  
 startActivity(new Intent(Login.this, MainHub.class));  
 }  
 else  
 {  
 if(m\_bInternetIssue)  
 SetError("Failed to connect! Check internet?");  
 else  
 SetError("Incorrect Details");  
  
 }  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
 }  
 });  
 return null;  
 }  
 }  
}

# MainHub class

package com.menu.menu;  
  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
  
import androidx.navigation.NavController;  
import androidx.navigation.Navigation;  
import androidx.navigation.ui.AppBarConfiguration;  
import androidx.navigation.ui.NavigationUI;  
  
import android.support.annotation.NonNull;  
import android.support.design.widget.NavigationView;  
import android.support.v4.app.Fragment;  
import android.support.v4.view.GravityCompat;  
import android.support.v4.widget.DrawerLayout;  
  
import android.support.v7.app.AlertDialog;  
import android.support.v7.app.AppCompatActivity;  
import android.text.InputType;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.OrdersCallback;  
  
public class MainHub extends AppCompatActivity  
{  
 //Deal with fragment management  
 public static final String *HomeFragmentTag* = "HOME\_FRAGMENT";  
 public static final String *BasketFragmentTag* = "BASKET\_FRAGMENT";  
 public static final String *SettingsFragmentTag* = "SETTINGS\_FRAGMENT";  
 public static final String *MyMealsFragmentTag* = "MY\_MEALS\_FRAGMENT";  
 public static final String *MyOrdersFragmentTag* = "MY\_ORDERS\_FRAGMENT";  
  
 private NavigationView m\_navigationView;  
  
 final View.OnClickListener m\_drawerLisner = new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 ((DrawerLayout)findViewById(R.id.*drawer\_layout*)).openDrawer(GravityCompat.*START*);  
 }  
 };  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main\_hub*);  
  
 m\_navigationView = findViewById(R.id.*nav\_view*);  
 m\_navigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener()  
 {  
 @Override  
 public boolean onNavigationItemSelected(@NonNull MenuItem menuItem)  
 {  
 ((DrawerLayout)findViewById(R.id.*drawer\_layout*)).closeDrawer(GravityCompat.*START*);  
  
 switch (menuItem.getItemId())  
 {  
 case R.id.*nav\_chefSettigns*:  
 m\_navigationView.getMenu().getItem(1).setChecked(true);  
 NavigateToFragment(*MyMealsFragmentTag*);  
 break;  
  
 case R.id.*nav\_signOut*:  
 LocalSettings.*SignOutUser*(getApplicationContext());  
 startActivity(new Intent(MainHub.this, Login.class));  
 break;  
  
 case R.id.*nav\_settings*:  
 m\_navigationView.getMenu().getItem(2).setChecked(true);  
 NavigateToFragment(*SettingsFragmentTag*);  
 break;  
  
 case R.id.*nav\_basket*:  
 m\_navigationView.getMenu().getItem(3).setChecked(true);  
 NavigateToFragment(*BasketFragmentTag*);  
 break;  
  
 case R.id.*nav\_currentOrders*:  
 m\_navigationView.getMenu().getItem(0).setChecked(true);  
 NavigateToFragment(*MyOrdersFragmentTag*);  
 break;  
  
 case R.id.*nav\_home*:  
  
 NavigateToFragment(*HomeFragmentTag*);  
 break;  
  
 case R.id.*nav\_reportIssue*:  
  
 AlertDialog.Builder builder = new AlertDialog.Builder(MainHub.this);  
 builder.setTitle("Send Error Report");  
  
 // Set up the buttons  
 builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which)  
 {  
 Intent intent = new Intent(Intent.*ACTION\_SENDTO*);  
 intent.setData(Uri.*parse*("mailto:"));  
 intent.putExtra(Intent.*EXTRA\_EMAIL*, "MeNU.contact.us@gmail.com");  
 intent.putExtra(Intent.*EXTRA\_SUBJECT*, "Issue from " + LocalSettings.*GetLocalUser*().getId());  
 if (intent.resolveActivity(getPackageManager()) != null)  
 {  
 startActivity(intent);  
 }  
 else  
 {  
 SetError("Failed to send report. Check internet?");  
 }  
  
 }  
 });  
 builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 dialog.cancel();  
 }  
 });  
  
 builder.show();  
  
 break;  
 }  
 return true;  
 }  
 });  
  
 TextView title = m\_navigationView.getHeaderView(0).findViewById(R.id.*nav\_bar\_title*);  
 title.setText(LocalSettings.*GetLocalUser*().getUsername());  
  
 TextView subTitle = m\_navigationView.getHeaderView(0).findViewById(R.id.*nav\_bar\_subTitle*);  
 subTitle.setText(LocalSettings.*GetLocalUser*().getEmail());  
  
 ImageView imgTitle = m\_navigationView.getHeaderView(0).findViewById(R.id.*nav\_bar\_image*);  
 imgTitle.setImageBitmap(LocalSettings.*GetLocalUser*().getPicture());  
  
 if(!LocalSettings.*GetLocalUser*().getIsChef\_b())  
 m\_navigationView.findViewById(R.id.*nav\_chefSettigns*).setVisibility(View.*INVISIBLE*);  
  
 Bundle extras = getIntent().getExtras();  
 if(extras == null || !extras.containsKey("fragment"))  
 {  
 NavigateToFragment(*HomeFragmentTag*);  
 }  
 else  
 {  
 NavigateToFragment(extras.getString("fragment"));  
 }  
 }  
  
 public void NavigateToFragment(String tag)  
 {  
 Fragment fragment = getSupportFragmentManager().findFragmentByTag(tag);  
 if (fragment == null || !fragment.isVisible())  
 {  
 switch (tag)  
 {  
 case *HomeFragmentTag*:  
 fragment = new Home();  
 ((Home)fragment).SetDrawerButtonListner(m\_drawerLisner);  
 tag = *HomeFragmentTag*;  
 break;  
  
 case *BasketFragmentTag*:  
 fragment = new Basket();  
 ((Basket)fragment).SetDrawerButtonListner(m\_drawerLisner);  
 tag = *BasketFragmentTag*;  
 break;  
  
 case *SettingsFragmentTag*:  
 fragment = new Settings();  
 ((Settings)fragment).SetDrawerButtonListner(m\_drawerLisner);  
 tag = *SettingsFragmentTag*;  
 break;  
  
 case *MyMealsFragmentTag*:  
 fragment = new MyMeals();  
 ((MyMeals)fragment).SetDrawerButtonListner(m\_drawerLisner);  
 tag = *MyMealsFragmentTag*;  
 break;  
  
 case *MyOrdersFragmentTag*:  
 fragment = new MyOrders();  
 ((MyOrders)fragment).SetDrawerButtonListner(m\_drawerLisner);  
 tag = *MyOrdersFragmentTag*;  
 break;  
 }  
  
 android.support.v4.app.FragmentTransaction ft = getSupportFragmentManager().beginTransaction();  
 ft.replace(R.id.*fragmentHolder*, fragment, tag);  
 ft.commit();  
 }  
 }  
  
 @Override  
 public void onBackPressed()  
 {  
 }  
  
 @Override  
 protected void onResume()  
 {  
 //Check not already on that fragment  
 Fragment hf = getSupportFragmentManager().findFragmentByTag(*HomeFragmentTag*);  
 if (hf == null || !hf.isVisible())  
 {  
 m\_navigationView.getMenu().getItem(0).setChecked(true);  
 }  
 super.onResume();  
 }  
  
 public void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(MainHub.this, errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu)  
 {  
 return true;  
 }  
  
 private AppBarConfiguration mAppBarConfiguration;  
 @Override  
 public boolean onSupportNavigateUp()  
 {  
 NavController navController = Navigation.*findNavController*(this, R.id.*nav\_host\_fragment*);  
 return NavigationUI.*navigateUp*(navController, mAppBarConfiguration)  
 || super.onSupportNavigateUp();  
 }  
}

# MealRegistration class

package com.menu.menu;  
  
import android.app.Activity;  
import android.app.Dialog;  
import android.app.TimePickerDialog;  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.graphics.Color;  
import android.graphics.drawable.BitmapDrawable;  
import android.graphics.drawable.ColorDrawable;  
import android.net.Uri;  
import android.provider.MediaStore;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.TimePicker;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.BaseCallback;  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.Meal;  
  
import java.io.FileNotFoundException;  
import java.io.IOException;  
  
public class MealRegistration extends AppCompatActivity  
{  
 private Meal m\_currentMeal = null;  
 private DatabaseCommunicator m\_dbComms = new DatabaseCommunicator();  
 public static Bitmap *FailedBitmap* = null;  
  
 ImageView m\_img\_image = null;  
 EditText m\_input\_name = null;  
 EditText m\_input\_maxNumberOfDishes = null;  
 EditText m\_input\_price = null;  
 EditText m\_input\_ingredients = null;  
 EditText m\_input\_calories = null;  
 Switch m\_switch\_takeaway = null;  
 Switch m\_switch\_eatIn = null;  
 Switch m\_switch\_isHalal = null;  
 Switch m\_switch\_isVegan = null;  
 Switch m\_switch\_isVegiterian = null;  
 Switch m\_switch\_containsMilk = null;  
 Switch m\_switch\_containsGluten = null;  
 TextView m\_txt\_pickDateFrom = null;  
 TextView m\_txt\_pickDateTo = null;  
 TextView m\_txt\_hoursAvaliableFrom = null;  
 TextView m\_txt\_hoursAvaliableTo = null;  
  
 TimePickerDialog.OnTimeSetListener m\_onHoursFromSetListener = null;  
 TimePickerDialog.OnTimeSetListener m\_onHoursToSetListener = null;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_meal\_registration*);  
  
 m\_img\_image = findViewById(R.id.*img\_image*);  
 m\_input\_name = findViewById(R.id.*input\_name*);  
 m\_input\_maxNumberOfDishes = findViewById(R.id.*input\_numerOfDishes*);  
 m\_input\_price = findViewById(R.id.*input\_price*);  
 m\_input\_ingredients = findViewById(R.id.*input\_ingredients*);  
 m\_switch\_takeaway = findViewById(R.id.*switch\_takeaway*);  
 m\_switch\_eatIn = findViewById(R.id.*switch\_eatIn*);  
 m\_txt\_pickDateFrom = findViewById(R.id.*txt\_pickDateFrom*);  
 m\_txt\_pickDateTo = findViewById(R.id.*txt\_pickDateTo*);  
 m\_txt\_hoursAvaliableFrom = findViewById(R.id.*txt\_HoursAvaliableFrom*);  
 m\_txt\_hoursAvaliableTo = findViewById(R.id.*txt\_HoursAvaliableTo*);  
 m\_input\_calories = findViewById(R.id.*input\_calories*);  
  
 final Dialog allergenDialog = new Dialog(this);  
 allergenDialog.setContentView(R.layout.*dialog\_alergen\_info\_editable*);  
 findViewById(R.id.*btn\_viewAllergenInfo*).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 allergenDialog.show();  
 }  
 });  
 m\_switch\_isHalal = allergenDialog.findViewById(R.id.*radio\_isHalal*);  
 m\_switch\_containsGluten = allergenDialog.findViewById(R.id.*radio\_containsGluten*);  
 m\_switch\_containsMilk = allergenDialog.findViewById(R.id.*radio\_containsMilk*);  
 m\_switch\_isVegan = allergenDialog.findViewById(R.id.*radio\_isVegan*);  
 m\_switch\_isVegiterian = allergenDialog.findViewById(R.id.*radio\_isVegetarian*);  
  
 final Button btn\_add = findViewById(R.id.*btn\_order*);  
 final Button btn\_delete = findViewById(R.id.*btn\_delete*);  
 final Button btn\_update = findViewById(R.id.*btn\_update*);  
  
 Bundle extras = getIntent().getExtras();  
 if(extras != null && extras.containsKey("meal"))  
 {  
 m\_currentMeal = (Meal) extras.getSerializable("meal");  
 m\_input\_ingredients.setText(m\_currentMeal.getIngredients());  
 m\_input\_name.setText(m\_currentMeal.getName());  
 m\_input\_maxNumberOfDishes.setText(m\_currentMeal.getMaxNoPortions());  
 m\_input\_price.setText(m\_currentMeal.getPrice());  
 m\_switch\_takeaway.setChecked(m\_currentMeal.IsTakeaway());  
 m\_switch\_eatIn.setChecked(m\_currentMeal.IsEatIn());  
 m\_switch\_isHalal.setChecked(m\_currentMeal.IsEatIn());  
 m\_switch\_containsGluten.setChecked(m\_currentMeal.getContainsGluten());  
 m\_switch\_containsMilk.setChecked(m\_currentMeal.getContainsMilk());  
 m\_switch\_isVegan.setChecked(m\_currentMeal.getVegan());  
 m\_switch\_isVegiterian.setChecked(m\_currentMeal.getVegiterian());  
 m\_txt\_pickDateTo.setText(m\_currentMeal.getHoursAvaliableTo());  
 m\_txt\_pickDateFrom.setText(m\_currentMeal.getHoursAvaliableFrom());  
 m\_input\_calories.setText(m\_currentMeal.getCalories());  
  
 Bitmap bmp = m\_currentMeal.getPicture();  
 if (bmp != null)  
 m\_img\_image.setImageBitmap(m\_currentMeal.getPicture());  
 else if (*FailedBitmap* != null)  
 {  
 m\_img\_image.setImageBitmap(*FailedBitmap*);  
 *FailedBitmap* = null;  
 }  
 else  
 {  
 m\_img\_image.setImageResource(R.drawable.*ic\_meal*);  
 }  
  
 btn\_add.setVisibility(View.*INVISIBLE*);  
 btn\_update.setVisibility(View.*VISIBLE*);  
 }  
 else  
 {  
 m\_currentMeal = new Meal();  
 m\_currentMeal.setOwnerUsername(LocalSettings.*GetLocalUser*().getUsername());  
 m\_currentMeal.setOwnerId(LocalSettings.*GetLocalUser*().getId());  
  
 m\_img\_image.setImageResource(R.drawable.*ic\_meal*);  
  
 //Since the meal is new, don't need to delete it...  
 btn\_delete.setVisibility(View.*INVISIBLE*);  
 btn\_update.setVisibility(View.*INVISIBLE*);  
 }  
  
 m\_img\_image.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 startActivityForResult(new Intent(Intent.*ACTION\_PICK*, android.provider.MediaStore.Images.Media.*INTERNAL\_CONTENT\_URI*), 3);  
 }  
 });  
  
 btn\_add.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 if(GetAndValidateMeal())  
 {  
 AddMealCallback amc = new AddMealCallback();  
 amc.SetMessage("IF NOT EXISTS (SELECT \* FROM " + m\_dbComms.*m\_mealTable* + " WHERE meal\_name = '" + m\_currentMeal.getName() + "') " + m\_currentMeal.GetInsertString() + ";");  
 m\_dbComms.GenericUpload(amc);  
 }  
 }  
 });  
  
 btn\_update.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 if(GetAndValidateMeal())  
 {  
 UpdateMealCallback umc = new UpdateMealCallback();  
 umc.SetMessage("UPDATE " + m\_dbComms.*m\_mealTable* + " SET " + m\_currentMeal.GetUpdateString() + "WHERE id = '" + m\_currentMeal.getId() + "';");  
 m\_dbComms.GenericUpload(umc);  
 }  
 }  
 });  
  
 btn\_delete.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 DeleteMealCallback dmc = new DeleteMealCallback();  
 dmc.SetMessage("DELETE FROM " + m\_dbComms.*m\_mealTable* + " WHERE id='" + m\_currentMeal.getId() + "';");  
 m\_dbComms.GenericUpload(dmc);  
 }  
 });  
  
 m\_onHoursFromSetListener = new TimePickerDialog.OnTimeSetListener()  
 {  
 @Override  
 public void onTimeSet(TimePicker timePicker, int i, int i1)  
 {  
 m\_currentMeal.setHoursAvaliableFrom((i + ":" + i1));  
 m\_txt\_pickDateFrom.setText(i + ":" + i1);  
 }  
 };  
  
 m\_onHoursToSetListener = new TimePickerDialog.OnTimeSetListener()  
 {  
 @Override  
 public void onTimeSet(TimePicker timePicker, int i, int i1)  
 {  
 m\_currentMeal.setHoursAvaliableTo((i + ":" + i1));  
 m\_txt\_pickDateTo.setText(i + ":" + i1);  
 }  
 };  
  
  
 View.OnClickListener pickDateFromListener = new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 TimePickerDialog timePickerDialog = new TimePickerDialog(MealRegistration.this, m\_onHoursFromSetListener, 0, 0, true);  
 timePickerDialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*WHITE*));  
 timePickerDialog.show();  
 }  
 };  
 m\_txt\_pickDateFrom.setOnClickListener(pickDateFromListener);  
 m\_txt\_hoursAvaliableFrom.setOnClickListener(pickDateFromListener);  
  
 View.OnClickListener pickDateToListener = new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 TimePickerDialog timePickerDialog = new TimePickerDialog(MealRegistration.this, m\_onHoursToSetListener, 0, 0, true);  
 timePickerDialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*WHITE*));  
 timePickerDialog.show();  
 }  
 };  
 m\_txt\_pickDateTo.setOnClickListener(pickDateToListener);  
 m\_txt\_hoursAvaliableTo.setOnClickListener(pickDateToListener);  
 }  
  
 private Boolean GetAndValidateMeal()  
 {  
 m\_currentMeal.setMaxNoPortions(m\_input\_maxNumberOfDishes.getText().toString());  
 m\_currentMeal.setName(m\_input\_name.getText().toString());  
 m\_currentMeal.setPicture(((BitmapDrawable)m\_img\_image.getDrawable()).getBitmap());  
 m\_currentMeal.setIngredients(m\_input\_ingredients.getText().toString());  
 m\_currentMeal.setPrice(m\_input\_price.getText().toString());  
 m\_currentMeal.SetEatIn(m\_switch\_eatIn.isChecked(), m\_switch\_takeaway.isChecked());  
 m\_currentMeal.setContainsMilk(m\_switch\_containsMilk.isChecked());  
 m\_currentMeal.setHalal(m\_switch\_isHalal.isChecked());  
 m\_currentMeal.setContainsGluten(m\_switch\_containsGluten.isChecked());  
 m\_currentMeal.setContainsMilk(m\_switch\_containsMilk.isChecked());  
 m\_currentMeal.setVegan(m\_switch\_isVegan.isChecked());  
 m\_currentMeal.setVegiterian(m\_switch\_isVegiterian.isChecked());  
 m\_currentMeal.setCalories(m\_input\_calories.getText().toString());  
  
 m\_switch\_containsMilk.isChecked();  
 m\_switch\_isVegan.isChecked();  
 m\_switch\_isVegiterian.isChecked();  
  
 String errorString = ValidateMeal(m\_currentMeal);  
 if (errorString.equals("NO-ERROR"))  
 {  
 return true;  
 }  
 else  
 {  
 SetError(errorString);  
 return false;  
 }  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(MealRegistration.this, errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
  
 private void NavigateToPreviousPage()  
 {  
 Intent intent = new Intent(MealRegistration.this, MainHub.class);  
 intent.putExtra("fragment", MainHub.*MyMealsFragmentTag*);  
 finish();  
 startActivity(intent);  
 }  
  
 private String ValidateMeal(Meal meal)  
 {  
 //*todo validation...* //name should be unique  
 return "NO-ERROR";  
 }  
  
 //Overridden so we can get the uploaded image  
 protected void onActivityResult(int requestCode, int resultCode, Intent data)  
 {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 //GET\_FROM\_GALLERY = 3  
 if(requestCode==3 && resultCode == Activity.*RESULT\_OK*)  
 {  
 Uri selectedImage = data.getData();  
 try  
 {  
 m\_currentMeal.setPicture(MediaStore.Images.Media.*getBitmap*(this.getContentResolver(), selectedImage));  
 m\_img\_image.setImageBitmap(m\_currentMeal.getPicture());  
 }  
 catch (FileNotFoundException e)  
 {  
 SetError("Image file could not be found");  
 }  
 catch (IOException e)  
 {  
 SetError("An I/O error occured while acquiring the image");  
 }  
 }  
 }  
  
 private class UpdateMealCallback extends BaseCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 if(m\_message.equals("null"))  
 {  
 NavigateToPreviousPage();  
 }  
 else  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 SetError("Update failed! Check internet connection.");  
 }  
 });  
 }  
  
 return null;  
 }  
 }  
  
 private class AddMealCallback extends BaseCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 if(m\_message.equals("null"))  
 {  
 NavigateToPreviousPage();  
 }  
 else  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 SetError("Adding meal failed! Check internet connection.");  
 }  
 });  
 }  
 return null;  
 }  
 }  
  
 private class DeleteMealCallback extends BaseCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 if(m\_message.equals("null"))  
 {  
 NavigateToPreviousPage();  
 }  
 else  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 SetError("Delete failed! Check internet connection.");  
 }  
 });  
 }  
 return null;  
 }  
 }  
}

# MealView class

package com.menu.menu;  
  
import android.app.Dialog;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.support.v7.app.AlertDialog;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.RadioButton;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.Meal;  
import com.menu.menu.Classes.BasketItem;  
import com.menu.menu.Classes.Order;  
import com.menu.menu.Classes.UsersCallback;  
  
public class MealView extends AppCompatActivity  
{  
 private Meal m\_meal = null;  
 public static Bitmap *FailedBitmap* = null;  
 private TextView txt\_address = null;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_meal\_view*);  
  
 Bundle extras = getIntent().getExtras();  
 if(extras == null || !extras.containsKey("meal"))  
 {  
 SetError("Error. Page loaded incorrectly.");  
 }  
 else  
 {  
 final EditText input\_numDishesOrdered = findViewById(R.id.*input\_numerOfDishes*);  
 final Switch switch\_takeaway = findViewById(R.id.*switch\_takeaway*);  
 final Switch switch\_eatIn = findViewById(R.id.*switch\_eatIn*);  
  
 m\_meal = (Meal)extras.getSerializable("meal");  
  
 txt\_address = findViewById(R.id.*txt\_addressFill*);  
 switch\_takeaway.setActivated(m\_meal.IsTakeaway());  
 switch\_eatIn.setActivated(m\_meal.IsEatIn());  
 ((TextView)findViewById(R.id.*txt\_name*)).setText(m\_meal.getName());  
 ((TextView)findViewById(R.id.*txt\_pricePerDish*)).setText(m\_meal.getPrice() + "£");  
 ((TextView)findViewById(R.id.*txt\_numberMeals*)).setText("Number of dishes (Max : " + m\_meal.getMaxNoPortions() + ")");  
 ((TextView)findViewById(R.id.*input\_ingredients*)).setText(m\_meal.getIngredients());  
  
 Bitmap bmp = m\_meal.getPicture();  
 if(bmp != null)  
 ((ImageView)findViewById(R.id.*img\_image*)).setImageBitmap(bmp);  
 else if(*FailedBitmap* != null)  
 {  
 ((ImageView)findViewById(R.id.*img\_image*)).setImageBitmap(*FailedBitmap*);  
 *FailedBitmap* = null;  
 }  
 else  
 {  
 ((ImageView)findViewById(R.id.*img\_image*)).setImageResource(R.drawable.*ic\_meal*);  
 }  
  
 ((TextView)findViewById(R.id.*txt\_rating*)).setText(m\_meal.getRating() +"/5");  
  
 DatabaseCommunicator dbComms = new DatabaseCommunicator();  
 OtherUserInfoCallback OUIC = new OtherUserInfoCallback();  
 OUIC.SetMessage("SELECT \* FROM " + dbComms.*m\_userTable* + " WHERE id='"+m\_meal.getOwnerId()+"';");  
 dbComms.RequestUserData(OUIC);  
  
  
 final Dialog allergenDialog = new Dialog(this);  
 allergenDialog.setContentView(R.layout.*dialog\_alergen\_info*);  
 ((RadioButton)allergenDialog.findViewById(R.id.*radio\_containsGluten*)).setChecked(m\_meal.getContainsGluten());  
 allergenDialog.findViewById(R.id.*radio\_containsGluten*).setEnabled(false);  
 ((RadioButton)allergenDialog.findViewById(R.id.*radio\_containsMilk*)).setChecked(m\_meal.getContainsMilk());  
 allergenDialog.findViewById(R.id.*radio\_containsMilk*).setEnabled(false);  
 ((RadioButton)allergenDialog.findViewById(R.id.*radio\_isHalal*)).setChecked(m\_meal.getHalal());  
 allergenDialog.findViewById(R.id.*radio\_isHalal*).setEnabled(false);  
 ((RadioButton)allergenDialog.findViewById(R.id.*radio\_isVegan*)).setChecked(m\_meal.getVegan());  
 allergenDialog.findViewById(R.id.*radio\_isVegan*).setEnabled(false);  
 ((RadioButton)allergenDialog.findViewById(R.id.*radio\_isVegetarian*)).setChecked(m\_meal.getVegiterian());  
 allergenDialog.findViewById(R.id.*radio\_isVegetarian*).setEnabled(false);  
 findViewById(R.id.*btn\_viewAllergenInfo*).setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 allergenDialog.show();  
 }  
 });  
  
 findViewById(R.id.*btn\_order*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 try  
 {  
 int orderedDishes = Integer.*parseInt*(input\_numDishesOrdered.getText().toString());  
 if(!(orderedDishes <= Integer.*parseInt*(m\_meal.getMaxNoPortions()) && orderedDishes > 0))  
 {  
 SetError("Number of dishes to order surpasses the maximum " + m\_meal.getMaxNoPortions());  
 return;  
 }  
 }  
 catch (Exception e)  
 {  
 SetError("Number of dishes to order is empty or invalid");  
 return;  
 }  
  
 if(switch\_takeaway.isChecked() || switch\_eatIn.isChecked())  
 {  
 DialogInterface.OnClickListener dialogClickListener = new DialogInterface.OnClickListener()  
 {  
 @Override  
 public void onClick(DialogInterface dialog, int which)  
 {  
 switch (which)  
 {  
 case DialogInterface.*BUTTON\_NEGATIVE*:  
 onBackPressed();  
 break;  
  
 case DialogInterface.*BUTTON\_POSITIVE*:  
  
 Intent intent = new Intent(MealView.this, MainHub.class);  
 intent.putExtra("fragment", MainHub.*BasketFragmentTag*);  
 finish();  
 startActivity(intent);  
 break;  
 }  
 }  
 };  
  
 //Add order  
 Basket.*addOrder*(new Order(m\_meal, m\_meal.getOwnerId(), LocalSettings.*GetLocalUser*().getId(), input\_numDishesOrdered.getText().toString(), switch\_takeaway.isChecked()));  
  
 AlertDialog.Builder builder = new AlertDialog.Builder(MealView.this);  
 builder.setMessage("Continue shopping?")  
 .setPositiveButton("Check Out", dialogClickListener)  
 .setNegativeButton("Yes", dialogClickListener).show();  
 }  
 else  
 {  
 SetError("Please select a meal order option.");  
 }  
 }  
 });  
  
 switch\_takeaway.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 switch\_eatIn.setChecked(false);  
 }  
 });  
  
 switch\_eatIn.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 switch\_takeaway.setChecked(false);  
 }  
 });  
 }  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(MealView.this, errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
  
 private class OtherUserInfoCallback extends UsersCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(m\_users != null && !m\_users.isEmpty())  
 {  
 txt\_address.setText(m\_users.get(0).getAddressLine1() + "\n" + m\_users.get(0).getAddressLine2() + "\n" +m\_users.get(0).getAddressLine3() + "\n" + m\_users.get(0).getAddressPostCode());  
 }  
 else  
 {  
 SetError("Data not found! Check internet?");  
 }  
 }  
 });  
 return null;  
 }  
 }  
}

# MeetupChat class

package com.menu.menu;  
  
import android.app.Dialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.support.v7.app.AlertDialog;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.EditText;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.widget.RatingBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.BaseCallback;  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.Order;  
import com.menu.menu.Classes.OrdersCallback;  
import com.menu.menu.Classes.User;  
import com.menu.menu.Classes.UsersCallback;  
  
import java.util.ArrayList;  
  
public class MeetupChat extends AppCompatActivity  
{  
 private User m\_eater = null;  
 private User m\_chef = null;  
 private boolean m\_bTakeaway = false;  
  
 private DatabaseCommunicator m\_dbComms = new DatabaseCommunicator();  
 private TextView txt\_addressLine1;  
 private TextView txt\_addressLine2;  
 private TextView txt\_addressLine3;  
 private TextView txt\_postCode;  
 private TextView txt\_meetingName;  
 private EditText input\_message;  
 private ListView listView\_messages;  
 private ProgressBar m\_progressBar;  
 private String m\_orderId;  
 private ArrayList<String> m\_messages = new ArrayList<>();  
 private Thread m\_updateThread = null;  
 private boolean m\_bContinueUpdate = false;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_meetup\_chat*);  
  
 txt\_addressLine1 = findViewById(R.id.*txt\_addressLine1*);  
 txt\_addressLine2 = findViewById(R.id.*txt\_addressLine2*);  
 txt\_addressLine3 = findViewById(R.id.*txt\_addressLine3*);  
 txt\_postCode = findViewById(R.id.*txt\_addressPostCode*);  
 txt\_meetingName = findViewById(R.id.*txt\_meetingName*);  
 input\_message = findViewById(R.id.*txt\_messageInput*);  
 listView\_messages = findViewById(R.id.*listView\_messages*);  
 m\_progressBar = findViewById(R.id.*progressBar*);  
  
 txt\_meetingName.setVisibility(View.*INVISIBLE*);  
 txt\_addressLine1.setVisibility(View.*INVISIBLE*);  
 txt\_addressLine2.setVisibility(View.*INVISIBLE*);  
 txt\_addressLine3.setVisibility(View.*INVISIBLE*);  
 txt\_postCode.setVisibility(View.*INVISIBLE*);  
  
 Bundle extras = getIntent().getExtras();  
 if(extras != null && extras.containsKey("orderId"))  
 {  
 m\_orderId = extras.getString("orderId");  
  
 GetOrderInfoCallback goic = new GetOrderInfoCallback();  
 goic.SetMessage("Select \* FROM " + m\_dbComms.*m\_orderTable* + " WHERE id = '" + m\_orderId + "';");  
 m\_dbComms.RequestOrderData(goic);  
  
 m\_progressBar.startNestedScroll(1);  
 m\_progressBar.setVisibility(View.*VISIBLE*);  
 }  
 else  
 {  
 SetError("Data not found! Check internet?");  
 }  
  
 findViewById(R.id.*btn\_postMessage*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 if(m\_eater == null)  
 return;  
  
 //Get message & add to list  
 String message = (LocalSettings.*GetLocalUser*().getId() == m\_eater.getId() ? "e -z- " : "c -z- ") + input\_message.getText().toString();  
 m\_messages.add(message);  
  
 //Create temp order object to utilize sql function for uploading  
 Order tempOrder = new Order();  
 tempOrder.setMessages(m\_messages);  
  
 //Upload new message list  
 AddMessageCallback amcb = new AddMessageCallback();  
 amcb.SetMessage("UPDATE " + m\_dbComms.*m\_orderTable* + " SET messages='" + tempOrder.getMessages\_sql() + "' WHERE id='" + m\_orderId + "';");  
 m\_dbComms.GenericUpload(amcb);  
  
 //Clear message input box  
 input\_message.setText("");  
 }  
 });  
  
 findViewById(R.id.*btn\_finish*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 final Dialog ratingDialog = new Dialog(MeetupChat.this);  
 ratingDialog.setContentView(R.layout.*dialog\_rating*);  
 ratingDialog.show();  
  
 ratingDialog.findViewById(R.id.*btn\_done*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View v)  
 {  
 RatingBar ratingBar = ratingDialog.findViewById(R.id.*ratingBar*);  
 float numStars = ratingBar.getRating();  
 //*todo update counter user's rating with this rating* DialogInterface.OnClickListener dialogClickListener = new DialogInterface.OnClickListener()  
 {  
 @Override  
 public void onClick(DialogInterface dialog, int which)  
 {  
 switch (which)  
 {  
 case DialogInterface.*BUTTON\_NEGATIVE*:  
 MarkCompleteCallback mcc = new MarkCompleteCallback();  
 mcc.SetMessage("DELETE FROM " + m\_dbComms.*m\_orderTable* + " WHERE id='" + m\_orderId + "';");  
 m\_dbComms.GenericUpload(mcc);  
  
 m\_progressBar.startNestedScroll(1);  
 m\_progressBar.setVisibility(View.*VISIBLE*);  
  
 break;  
  
 case DialogInterface.*BUTTON\_POSITIVE*:  
 break;  
 }  
 }  
 };  
  
 ratingDialog.hide();  
  
 AlertDialog.Builder builder = new AlertDialog.Builder(MeetupChat.this);  
 builder.setMessage("Finish Order?")  
 .setPositiveButton("No", dialogClickListener)  
 .setNegativeButton("Yes", dialogClickListener).show();  
 }  
 });  
 }  
 });  
  
 //Update messages every second  
 m\_bContinueUpdate = true;  
 m\_updateThread = new Thread()  
 {  
 @Override  
 public void run()  
 {  
 try  
 {  
 while (!isInterrupted() && m\_bContinueUpdate)  
 {  
 Thread.*sleep*(1000);  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 UpdateMessages();  
 }  
 });  
 }  
 } catch (InterruptedException e) {  
 }  
 }  
 };  
 }  
  
 @Override  
 public void onBackPressed()  
 {  
 m\_bContinueUpdate = false;  
 m\_updateThread = null;  
 super.onBackPressed();  
 }  
  
 @Override  
 protected void onStop()  
 {  
 m\_bContinueUpdate = false;  
 m\_updateThread = null;  
 super.onStop();  
 }  
  
 private void UpdateMessages()  
 {  
 UpdateMessagesCallback umcb = new UpdateMessagesCallback();  
 umcb.SetMessage("SELECT \* FROM " + m\_dbComms.*m\_orderTable* + " WHERE id='" + m\_orderId + "';");  
 m\_dbComms.RequestOrderData(umcb);  
 }  
  
 private class MarkCompleteCallback extends BaseCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
 if(m\_message.equals("null"))  
 {  
 m\_bContinueUpdate = false;  
 m\_updateThread = null;  
 startActivity(new Intent(MeetupChat.this, MainHub.class));  
 }  
 else  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 SetError("Failed. Check internet?");  
 }  
 });  
 }  
 return null;  
 }  
 }  
  
 private class AddMessageCallback extends BaseCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 if(!m\_message.equals("null"))  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 SetError("Failed to add message. Check internet?");  
 }  
 });  
 }  
 return null;  
 }  
 }  
  
 private class UpdateMessagesCallback extends OrdersCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 if(m\_bContinueUpdate)  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(m\_orders != null && !m\_orders.isEmpty())  
 {  
 m\_messages = m\_orders.get(0).getMessages();  
 if(m\_messages.get(0).equals("null"))  
 m\_messages.remove(0);  
  
 listView\_messages.setAdapter(new MessageListViewAdapter(getApplicationContext(), m\_messages));  
 }  
 else  
 {  
 SetError("Failed to update chat! Check internet?");  
 }  
 }  
 });  
 return null;  
 }  
 }  
  
 private class GetOrderInfoCallback extends OrdersCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 if(m\_orders != null && !m\_orders.isEmpty())  
 {  
 OtherUserInfoCallback ouic = new OtherUserInfoCallback();  
 if(m\_orders.get(0).getEaterId().equals(LocalSettings.*GetLocalUser*().getId()))  
 {  
 m\_eater = LocalSettings.*GetLocalUser*();  
 ouic.SetMessage("SELECT \* FROM " + m\_dbComms.*m\_userTable* + " WHERE id='" + m\_orders.get(0).getChefId() +"';");  
 }  
 else  
 {  
 m\_chef = LocalSettings.*GetLocalUser*();  
 ouic.SetMessage("SELECT \* FROM " + m\_dbComms.*m\_userTable* + " WHERE id='" + m\_orders.get(0).getEaterId() +"';");  
 }  
  
 m\_dbComms.RequestUserData(ouic);  
 }  
 else  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 SetError("Data not found! Check internet?");  
 }  
 });  
  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
 }  
 return null;  
 }  
 }  
  
 private class OtherUserInfoCallback extends UsersCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(m\_users != null && !m\_users.isEmpty())  
 {  
 if(m\_eater == null)  
 {  
 m\_eater = m\_users.get(0);  
  
 txt\_meetingName.setText("Meeting: " + m\_eater.getUsername());  
 txt\_meetingName.setVisibility(View.*VISIBLE*);  
  
 if(m\_bTakeaway)  
 {  
 txt\_addressLine1.setText(m\_eater.getAddressLine1());  
 txt\_addressLine2.setText(m\_eater.getAddressLine2());  
 txt\_addressLine3.setText(m\_eater.getAddressLine3());  
 txt\_postCode.setText(m\_eater.getAddressPostCode());  
  
 txt\_addressLine1.setVisibility(View.*VISIBLE*);  
 txt\_addressLine2.setVisibility(View.*VISIBLE*);  
 txt\_addressLine3.setVisibility(View.*VISIBLE*);  
 txt\_postCode.setVisibility(View.*VISIBLE*);  
 }  
 }  
 else  
 {  
 m\_chef = m\_users.get(0);  
  
 txt\_meetingName.setText("Meeting: " + m\_chef.getUsername());  
 txt\_meetingName.setVisibility(View.*VISIBLE*);  
  
 if(!m\_bTakeaway)  
 {  
 txt\_addressLine1.setText(m\_chef.getAddressLine1());  
 txt\_addressLine2.setText(m\_chef.getAddressLine2());  
 txt\_addressLine3.setText(m\_chef.getAddressLine3());  
 txt\_postCode.setText(m\_chef.getAddressPostCode());  
  
 txt\_addressLine1.setVisibility(View.*VISIBLE*);  
 txt\_addressLine2.setVisibility(View.*VISIBLE*);  
 txt\_addressLine3.setVisibility(View.*VISIBLE*);  
 txt\_postCode.setVisibility(View.*VISIBLE*);  
 }  
 }  
 }  
 else  
 {  
 SetError("Data not found! Check internet?");  
 }  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
 m\_bContinueUpdate = true;  
 m\_updateThread.start();  
 }  
 });  
 return null;  
 }  
 }  
  
 public class MessageListViewAdapter extends ArrayAdapter<String>  
 {  
 public ArrayList<String> m\_messageListRef;  
  
 public MessageListViewAdapter(Context c, ArrayList<String> messageListRef)  
 {  
 super(c, R.layout.*layout\_message\_eater*, messageListRef);  
 m\_messageListRef = messageListRef;  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent)  
 {  
 String message = m\_messageListRef.get(position);  
  
 String[] messageAndType = message.split(" -z- ");  
  
 View itemView = convertView;  
 if (itemView == null)  
 {  
 itemView = (LayoutInflater.*from*(getContext())).inflate(messageAndType[0].equals("c") ? R.layout.*layout\_message\_chef* : R.layout.*layout\_message\_eater*, parent, false);  
 }  
  
 //Subject text  
 TextView subjectText;  
 if(messageAndType[0].equals("c"))  
 {  
 subjectText = itemView.findViewById(R.id.*txt\_chefMessage*);  
 }  
 else  
 {  
 subjectText = itemView.findViewById(R.id.*txt\_eaterMessage*);  
 }  
 subjectText.setText(messageAndType[1]);  
  
 return itemView;  
 }  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(MeetupChat.this, errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
}

# MyMeals class

package com.menu.menu;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.Button;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.Meal;  
import com.menu.menu.Classes.MealListViewItem;  
import com.menu.menu.Classes.MealsCallback;  
  
import java.util.ArrayList;  
  
public class MyMeals extends Fragment  
{  
 private DatabaseCommunicator m\_dbComms = new DatabaseCommunicator();  
 private ArrayList<Meal> m\_mealInfoArray = new ArrayList<>();  
 private ListView m\_displayList;  
 private ProgressBar m\_progressBar;  
 private View.OnClickListener m\_drawerListener;  
  
 public void SetDrawerButtonListner(View.OnClickListener listener)  
 {  
 m\_drawerListener = listener;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState)  
 {  
 View root = inflater.inflate(R.layout.*fragment\_my\_meals*, container, false);  
 root.findViewById(R.id.*drawerToggle*).setOnClickListener(m\_drawerListener);  
  
 m\_progressBar = root.findViewById(R.id.*progressBar*);  
  
 m\_displayList = root.findViewById(R.id.*mealsList*);  
 UpdateList();  
  
 Button btn\_addMeal = root.findViewById(R.id.*btn\_addMeal*);  
 btn\_addMeal.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 //Not adding intent with meal key signifies new meal...  
 startActivity(new Intent(getActivity(), MealRegistration.class));  
 }  
 });  
  
 m\_displayList.setOnItemClickListener(new AdapterView.OnItemClickListener()  
 {  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id)  
 {  
 Intent intent = new Intent(getActivity(), MealRegistration.class);  
 Meal parcelMeal = m\_mealInfoArray.get(position);  
 MealRegistration.*FailedBitmap* = parcelMeal.getPicture();  
 //parcelMeal.setPicture(null);  
 intent.putExtra("meal", parcelMeal);  
 startActivity(intent);  
 }  
 });  
  
 return root;  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(getActivity(), errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
  
 private void UpdateList()  
 {  
 m\_progressBar.startNestedScroll(1);  
 m\_progressBar.setVisibility(View.*VISIBLE*);  
  
 GetMealsListCallback gmlc = new GetMealsListCallback();  
 gmlc.SetMessage("SELECT \* FROM " + m\_dbComms.*m\_mealTable* + " WHERE owner\_user\_id = '" + LocalSettings.*GetLocalUser*().getId() + "';");  
 m\_dbComms.RequestMealData(gmlc);  
 }  
  
 private class GetMealsListCallback extends MealsCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 getActivity().runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(m\_meals != null && !m\_meals.isEmpty())  
 {  
 m\_mealInfoArray = m\_meals;  
 m\_displayList.setAdapter(new MealListViewItem(getActivity(),m\_mealInfoArray));  
 }  
 else  
 {  
 SetError(m\_message);  
 }  
  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
 }  
 });  
 return null;  
 }  
 }  
}

# MyOrders class

package com.menu.menu;  
  
import android.content.Context;  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.os.Bundle;  
import android.support.v4.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.MealsCallback;  
import com.menu.menu.Classes.Order;  
import com.menu.menu.Classes.OrdersCallback;  
  
import java.util.ArrayList;  
  
public class MyOrders extends Fragment  
{  
 private View.OnClickListener m\_drawerListener;  
 private ListView m\_displayList;  
 private ArrayList<Order> m\_displayedOrders = new ArrayList<>();  
 private ProgressBar m\_progressBar;  
 private DatabaseCommunicator m\_dbComms = new DatabaseCommunicator();  
  
 public void SetDrawerButtonListner(View.OnClickListener listener)  
 {  
 m\_drawerListener = listener;  
 }  
  
 @Override  
 public void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 }  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container,  
 Bundle savedInstanceState)  
 {  
 View root = inflater.inflate(R.layout.*fragment\_my\_orders*, container, false);  
 root.findViewById(R.id.*drawerToggle*).setOnClickListener(m\_drawerListener);  
 m\_displayList = root.findViewById(R.id.*listView\_orders*);  
 m\_progressBar = root.findViewById(R.id.*progressBar*);  
 m\_displayList.setOnItemClickListener(new AdapterView.OnItemClickListener()  
 {  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id)  
 {  
 Intent intent = new Intent(getContext(), MeetupChat.class);  
 intent.putExtra("orderId", m\_displayedOrders.get(position).getId());  
 startActivity(intent);  
 }  
 });  
  
 return root;  
 }  
  
 @Override  
 public void onResume()  
 {  
 update();  
 super.onResume();  
 }  
  
 private void update()  
 {  
 m\_progressBar.startNestedScroll(1);  
 m\_progressBar.setVisibility(View.*VISIBLE*);  
  
 m\_displayedOrders = new ArrayList<>();  
  
 GetOrdersCallback goc = new GetOrdersCallback();  
 goc.SetMessage("SELECT \* FROM " + m\_dbComms.*m\_orderTable* + " WHERE meal\_chef\_id = '"+ LocalSettings.*GetLocalUser*().getId() +"' OR meal\_eater\_id='"+LocalSettings.*GetLocalUser*().getId()+"'");  
 m\_dbComms.RequestOrderData(goc);  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast.*makeText*(getActivity(), errorString, Toast.*LENGTH\_LONG*).show();  
 }  
  
 class GetOrdersCallback extends OrdersCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 (getActivity()).runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if (m\_orders != null)  
 {  
 if(!m\_orders.isEmpty())  
 {  
 for (Order order : m\_orders)  
 {  
 GetMealOfOrderCallback gmooc = new GetMealOfOrderCallback();  
 gmooc.SetMessage("SELECT \* FROM " + m\_dbComms.*m\_mealTable* + " WHERE id='"+order.getMealId()+"'");  
 gmooc.setOrder(order);  
 m\_dbComms.RequestMealData(gmooc);  
 }  
 }  
 else  
 {  
 SetError("No orders to show!");  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
 }  
 }  
 else  
 {  
 SetError("Failed to retrieve orders!");  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
 }  
 }  
 });  
 return null;  
 }  
 }  
  
 class GetMealOfOrderCallback extends MealsCallback  
 {  
 Order m\_order = null;  
 @Override  
 public Void call() throws Exception  
 {  
 (getActivity()).runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if (m\_meals != null && m\_meals.size() > 0)  
 {  
 m\_order.setMeal(m\_meals.get(0));  
 m\_displayedOrders.add(m\_order);  
 m\_displayList.setAdapter(new OrderListViewItem(getContext(), m\_displayedOrders));  
 }  
 else  
 {  
 SetError("Failed to retrieve an order! Check internet.");  
 }  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
 }  
 });  
 return null;  
 }  
  
 public void setOrder(Order order)  
 {  
 m\_order = order;  
 }  
 }  
  
 //Class used to create a corresponding UI element for each ic\_meal in an arraylist of meals  
 //These UI elements are then added into displayList  
 class OrderListViewItem extends ArrayAdapter<Order>  
 {  
 ArrayList<Order> m\_orderItemListRef;  
  
 public OrderListViewItem(Context c, ArrayList<Order> orderItemListRef)  
 {  
 super(c, R.layout.*layout\_order\_info*, orderItemListRef);  
 m\_orderItemListRef = orderItemListRef;  
 }  
  
 @Override  
 public View getView(final int position, View convertView, ViewGroup parent)  
 {  
 //get view from activity\_forms.xml  
 View itemView = convertView;  
 if (itemView == null)  
 {  
 itemView = (LayoutInflater.*from*(getContext())).inflate(R.layout.*layout\_order\_info*, parent, false);  
 }  
  
 final Order currentOrder = m\_orderItemListRef.get(position);  
  
 //Subject text  
 final TextView txt\_subject = itemView.findViewById(R.id.*li\_top*);  
 String price = currentOrder.getMeal().getPrice();  
 Double cost = 0.00;  
 try  
 {  
 cost = Double.*parseDouble*(price) \* currentOrder.getNumberOfPortions\_n();  
 }  
 catch (Exception e)  
 {  
 }  
 txt\_subject.setText(String.*format*("%.2f", cost) + "£");  
  
 //Ammount  
 final TextView txt\_orderAmount = itemView.findViewById(R.id.*li\_top2*);  
 txt\_orderAmount.setText("Portions: " + currentOrder.getNumberOfPortions());  
  
 //Price  
 ((TextView)itemView.findViewById(R.id.*li\_text*)).setText(currentOrder.getMeal().getName());  
  
 //Img  
 ImageView img = itemView.findViewById(R.id.*img*);  
 Bitmap bmp = currentOrder.getMeal().getPicture();  
 if(bmp!=null)  
 img.setImageBitmap(bmp);  
 else  
 img.setImageResource(R.drawable.*ic\_meal*);  
  
 return itemView;  
 }  
 }  
}

# PaymentProcessing class

package com.menu.menu;  
  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.BaseCallback;  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.Meal;  
import com.menu.menu.Classes.BasketItem;  
  
public class PaymentProcessing extends AppCompatActivity  
{  
 Meal m\_meal = null;  
 int m\_numberOfMeals = 0; //can probs remove if end up using order  
 BasketItem m\_basketItem = null;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_payment\_processing*);  
  
 final Button btn\_pay = findViewById(R.id.*btn\_pay*);  
 final TextView txt\_name = findViewById(R.id.*txt\_name*);  
 final TextView txt\_price = findViewById(R.id.*txt\_price*);  
  
 Bundle extras = getIntent().getExtras();  
 if(extras != null && extras.containsKey("meal"))  
 {  
 m\_meal = (Meal)extras.getSerializable("meal");  
  
 txt\_name.setText("ic\_meal : " + m\_meal.getName());  
 txt\_price.setText("Price : " + Double.*toString*(m\_numberOfMeals \* Double.*parseDouble*(m\_meal.getPrice())));  
  
 btn\_pay.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 //*todo set message* UploadOrderCallBack uocb = new UploadOrderCallBack();  
 uocb.SetMessage("");  
 new DatabaseCommunicator().GenericUpload(uocb);  
 }  
 });  
 }  
 else  
 {  
 SetError("ic\_meal not found! Check internet?");  
 }  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(PaymentProcessing.this, errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
  
 class UploadOrderCallBack extends BaseCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(m\_message.equals("null"))  
 {  
 Intent intent = new Intent(PaymentProcessing.this, MeetupChat.class);  
 intent.putExtra("meal", m\_meal);  
 intent.putExtra("order", m\_basketItem);  
 startActivity(intent);  
 }  
 else  
 {  
 SetError("Failed to place order! Check internet.");  
 }  
 }  
 });  
 return null;  
 }  
 }  
}

# SearchResults class

package com.menu.menu;  
  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.KeyEvent;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.Meal;  
import com.menu.menu.Classes.MealListViewItem;  
import com.menu.menu.Classes.MealsCallback;  
import com.menu.menu.Classes.User;  
import com.menu.menu.Classes.UsersCallback;  
import java.util.ArrayList;  
import java.util.List;  
  
public class SearchResults extends AppCompatActivity  
{  
 private DatabaseCommunicator m\_dbComms = new DatabaseCommunicator();  
 private ArrayList<Meal> m\_mealInfoArray = new ArrayList<>();  
 private ArrayList<User> m\_userInfoArray = new ArrayList<>();  
 private ListView m\_displayList;  
 private EditText m\_searchText;  
 private Boolean m\_bShowingUsers = true;  
 private ProgressBar m\_progressBar;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_search\_results*);  
  
 m\_searchText = findViewById(R.id.*input\_search*);  
 m\_displayList = findViewById(R.id.*mealsList*);  
 m\_progressBar = findViewById(R.id.*progressBar*);  
  
 Bundle extras = getIntent().getExtras();  
 if(extras == null || !extras.containsKey("search"))  
 {  
 UpdateListWithUsers(null);  
 }  
 else  
 {  
 UpdateListWithUsers(extras.getString("search"));  
 }  
  
 Button btn\_home = findViewById(R.id.*btn\_home*);  
 btn\_home.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 startActivity(new Intent(SearchResults.this, MainHub.class));  
 }  
 });  
  
 m\_searchText.setOnKeyListener(new View.OnKeyListener()  
 {  
 @Override  
 public boolean onKey(View v, int keyCode, KeyEvent event)  
 {  
 if (event.getAction()==KeyEvent.*ACTION\_DOWN* && keyCode == KeyEvent.*KEYCODE\_ENTER*)  
 {  
 if(m\_bShowingUsers)  
 {  
 UpdateListWithUsers(m\_searchText.getText().toString());  
 }  
 else  
 {  
 UpdateListWithMeals(m\_searchText.getText().toString());  
 }  
 return true;  
 }  
 return false;  
 }  
 });  
  
 m\_displayList.setOnItemClickListener(new AdapterView.OnItemClickListener()  
 {  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id)  
 {  
 if(m\_bShowingUsers)  
 {  
 Intent intent = new Intent(SearchResults.this, ChefMeals.class);  
 intent.putExtra("chefId", m\_userInfoArray.get(position).getId());  
 intent.putExtra("chefUsername", m\_userInfoArray.get(position).getUsername());  
 startActivity(intent);  
 }  
 else  
 {  
 Intent intent = new Intent(SearchResults.this, MealView.class);  
 intent.putExtra("meal", m\_mealInfoArray.get(position));  
 startActivity(intent);  
 }  
 }  
 });  
 }  
  
 private void UpdateListWithUsers(String where)  
 {  
 m\_progressBar.startNestedScroll(1);  
 m\_progressBar.setVisibility(View.*VISIBLE*);  
  
 m\_userInfoArray.clear();  
  
 SearchResults.GetChefsListCallback gclc = new SearchResults.GetChefsListCallback();  
 String message = "SELECT \* FROM " + m\_dbComms.*m\_userTable* + " WHERE is\_chef = 'true' ";  
 if(where != null)  
 {  
 message += "AND name LIKE '%"+where+"%'";  
 }  
 gclc.SetMessage(message + ";");  
 m\_displayList.setAdapter(new SearchResults.UserListAdaptor());  
 m\_dbComms.RequestUserData(gclc);  
 }  
  
 private void UpdateListWithMeals(String where)  
 {  
 m\_progressBar.startNestedScroll(1);  
 m\_progressBar.setVisibility(View.*VISIBLE*);  
  
 m\_mealInfoArray.clear();  
  
 SearchResults.GetMealsListCallback gmlc = new SearchResults.GetMealsListCallback();  
 String message = "SELECT \* FROM " + m\_dbComms.*m\_mealTable* + " ";  
 if(where != null)  
 {  
 message += "WHERE meal\_name LIKE '%"+where+"%''";;  
 }  
 gmlc.SetMessage(message + ";");  
 m\_displayList.setAdapter(new MealListViewItem(getApplicationContext(),m\_mealInfoArray));  
 m\_dbComms.RequestMealData(gmlc);  
 }  
  
 //Class used to create a corresponding UI element for each ic\_meal in m\_mealInfoArray  
 //These UI elements are then added into m\_displayList  
 private class UserListAdaptor extends ArrayAdapter<User>  
 {  
 public UserListAdaptor()  
 {  
 super(SearchResults.this, R.layout.*layout\_meal\_info*, m\_userInfoArray);  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent)  
 {  
 //get view from activity\_forms.xml  
 View itemView = convertView;  
 if (itemView == null)  
 {  
 itemView = getLayoutInflater().inflate(R.layout.*layout\_meal\_info*, parent, false);  
 }  
  
 User currentUser = m\_userInfoArray.get(position);  
  
 TextView subjectText = itemView.findViewById(R.id.*li\_infoLeft*);  
 subjectText.setText(currentUser.getUsername());  
  
 TextView foodType = itemView.findViewById(R.id.*li\_text*);  
 foodType.setText("Food speciality: " + currentUser.getFoodType());  
  
 TextView rating = itemView.findViewById(R.id.*li\_infoRight*);  
 rating.setText(currentUser.getRating() + "/5");  
  
 //Img  
 ImageView img = itemView.findViewById(R.id.*img*);  
 img.setImageBitmap(currentUser.getPicture());  
  
 return itemView;  
 }  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(SearchResults.this, errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
  
 private class GetMealsListCallback extends MealsCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(!m\_meals.isEmpty())  
 {  
 m\_mealInfoArray = m\_meals;  
 m\_displayList.setAdapter(new MealListViewItem(getApplicationContext(), m\_mealInfoArray));  
 }  
 else  
 {  
 SetError(m\_message);  
 }  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
 }  
 });  
 return null;  
 }  
 }  
  
 private class GetChefsListCallback extends UsersCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(m\_users != null && !m\_users.isEmpty())  
 {  
 m\_userInfoArray = m\_users;  
 m\_displayList.setAdapter(new SearchResults.UserListAdaptor());  
 }  
 else  
 {  
 if(m\_users == null)  
 SetError("Check internet connection!");  
 else  
 SetError("No chefs found.");  
 }  
 m\_progressBar.setVisibility(View.*INVISIBLE*);  
 m\_progressBar.stopNestedScroll();  
 }  
 });  
 return null;  
 }  
 }  
}

# Settings class

package com.menu.menu;  
  
import android.app.Activity;  
import android.app.DatePickerDialog;  
import android.content.Intent;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.net.Uri;  
import android.os.Bundle;  
import android.provider.MediaStore;  
import android.support.v4.app.Fragment;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.DatePicker;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.BaseCallback;  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.User;  
  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.time.Instant;  
import java.util.Date;  
  
public class Settings extends Fragment  
{  
 private DatabaseCommunicator m\_dbComms = new DatabaseCommunicator();  
 private User m\_currentUser = LocalSettings.*GetLocalUser*();  
 private ImageView m\_img\_image = null;  
 private DatePickerDialog.OnDateSetListener m\_onDobSetListener = null;  
 private View.OnClickListener m\_drawerListener;  
  
 @Override  
 public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)  
 {  
 View root = inflater.inflate(R.layout.*fragment\_settings*, container, false);  
 root.findViewById(R.id.*drawerToggle*).setOnClickListener(m\_drawerListener);  
  
 final EditText input\_email = root.findViewById(R.id.*input\_name*);  
 final EditText input\_phone = root.findViewById(R.id.*input\_phone*);  
 final EditText input\_fullName = root.findViewById(R.id.*input\_fullName*);  
 final EditText input\_password = root.findViewById(R.id.*input\_password*);  
 final TextView txt\_edit\_dob = root.findViewById(R.id.*txt\_edit\_dob*);  
 m\_img\_image = root.findViewById(R.id.*img\_image*);  
  
 input\_email.setText(m\_currentUser.getEmail());  
 input\_phone.setText(m\_currentUser.getPhone());  
 input\_fullName.setText(m\_currentUser.getFullName());  
 input\_password.setText(m\_currentUser.getPassword());  
 txt\_edit\_dob.setText(m\_currentUser.getDOB());  
 if(m\_currentUser.getPicture() != null)  
 {  
 m\_img\_image.setImageBitmap(m\_currentUser.getPicture());  
 }  
  
 root.findViewById(R.id.*btn\_saveSettigns*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 m\_currentUser.setEmail(input\_email.getText().toString());  
 m\_currentUser.setPhone(input\_phone.getText().toString());  
 m\_currentUser.setFullName(input\_fullName.getText().toString());  
 m\_currentUser.setPassword(input\_password.getText().toString());  
  
 String errorString = SignUp.*ValidateSettings*(m\_currentUser);  
 if (errorString.equals("NO-ERROR"))  
 {  
 UpdateUserCallback umc = new UpdateUserCallback();  
 umc.SetMessage("UPDATE " + m\_dbComms.*m\_userTable* + " SET " + m\_currentUser.GetUpdateString() + " WHERE id = '" + m\_currentUser.getId() + "';");  
 m\_dbComms.GenericUpload(umc);  
 }  
 else  
 {  
 SetError(errorString);  
 }  
 }  
 });  
  
 root.findViewById(R.id.*btn\_addressSettings*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 startActivity(new Intent(getContext(), AddressEdit.class));  
 }  
 });  
  
 m\_img\_image.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 startActivityForResult(new Intent(Intent.*ACTION\_PICK*, android.provider.MediaStore.Images.Media.*INTERNAL\_CONTENT\_URI*), 3);  
 }  
 });  
  
 root.findViewById(R.id.*btn\_chef*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 startActivity(new Intent(getContext(), ChefAccountSettings.class));  
 }  
 });  
  
 m\_onDobSetListener = new DatePickerDialog.OnDateSetListener()  
 {  
 @Override  
 public void onDateSet(DatePicker datePicker, int day, int month, int year)  
 {  
 Date d = new Date(day, month, year);  
 if (d.after(Date.*from*(Instant.*now*())))  
 {  
 String dob = year + "-" + month + "-" + day;  
 m\_currentUser.setDOB(dob);  
 txt\_edit\_dob.setText(dob);  
 }  
 else  
 {  
 SetError("Invalid date of birth.");  
 }  
 }  
 };  
  
 txt\_edit\_dob.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 int day = 1;  
 int month = 1;  
 int year = 2000;  
  
 String dob = m\_currentUser.getDOB();  
 if(dob != null)  
 {  
 try  
 {  
 day = Integer.*parseInt*(dob.split("-")[0]);  
 month = Integer.*parseInt*(dob.split("-")[1]);  
 year = Integer.parseInt(dob.split("-")[2]);  
 }  
 catch (Exception e)  
 {  
 }  
 }  
  
 DatePickerDialog datePickerDialog = new DatePickerDialog(getActivity(), m\_onDobSetListener, day, month, year);  
 datePickerDialog.setTitle("Date of Birth");  
 datePickerDialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.WHITE));  
 datePickerDialog.show();  
 }  
 });  
  
 return root;  
 }  
  
 public void SetDrawerButtonListner(View.OnClickListener listener)  
 {  
 m\_drawerListener = listener;  
 }  
  
 @Override  
 public void onResume()  
 {  
 super.onResume();  
 m\_currentUser = LocalSettings.GetLocalUser();  
 }  
  
 //Overridden so we can get the uploaded image  
 public void onActivityResult(int requestCode, int resultCode, Intent data)  
 {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 //GET\_FROM\_GALLERY = 3  
 if(requestCode==3 && resultCode == Activity.RESULT\_OK)  
 {  
 Uri selectedImage = data.getData();  
 try  
 {  
 m\_currentUser.setPicture(MediaStore.Images.Media.getBitmap(getActivity().getContentResolver(), selectedImage));  
 m\_img\_image.setImageBitmap(m\_currentUser.getPicture());  
 }  
 catch (FileNotFoundException e)  
 {  
 SetError("Image file could not be found");  
 }  
 catch (IOException e)  
 {  
 SetError("An I/O error occured while acquiring the image");  
 }  
 }  
 }  
  
 private void NavigateHome()  
 {  
 Intent intent = new Intent(getContext(), MainHub.class);  
 intent.putExtra("fragment", MainHub.HomeFragmentTag);  
 startActivity(intent);  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.makeText(getActivity(), errorString, Toast.LENGTH\_LONG);  
 t.show();  
 }  
  
 private class UpdateUserCallback extends BaseCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 if(m\_message.equals("null"))  
 {  
 LocalSettings.UpdateLocalUser(m\_currentUser);  
 NavigateHome();  
 }  
 else  
 {  
 getActivity().runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 SetError("Update failed! Check internet connection.");  
 }  
 });  
 }  
  
 return null;  
 }  
 }  
  
}

# SignUp class

package com.menu.menu;  
  
import android.app.Activity;  
import android.app.DatePickerDialog;  
import android.content.Intent;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.net.Uri;  
import android.provider.MediaStore;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.DatePicker;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.menu.menu.Classes.BaseCallback;  
import com.menu.menu.Classes.DatabaseCommunicator;  
import com.menu.menu.Classes.LocalSettings;  
import com.menu.menu.Classes.User;  
import com.menu.menu.Classes.UsersCallback;  
  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.time.Instant;  
import java.util.Date;  
  
public class SignUp extends AppCompatActivity  
{  
 private final DatabaseCommunicator m\_dbComms = new DatabaseCommunicator();  
 private User m\_currentUser = null;  
 private DatePickerDialog.OnDateSetListener m\_onDobSetListener = null;  
 private ImageView m\_img\_image = null;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState)  
 {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_sign\_up*);  
  
 final EditText input\_username = findViewById(R.id.*input\_name*);  
 final EditText input\_email = findViewById(R.id.*input\_email*);  
 final EditText input\_phone = findViewById(R.id.*input\_phone*);  
 final EditText input\_fullName = findViewById(R.id.*input\_fullName*);  
 final TextView txt\_edit\_dob = findViewById(R.id.*txt\_edit\_dob*);  
 final EditText input\_password = findViewById(R.id.*input\_password*);  
 m\_img\_image = findViewById(R.id.*img\_image*);  
 m\_img\_image.setImageResource(R.drawable.*ic\_meal*);  
  
 m\_currentUser = new User();  
 LocalSettings.*UpdateLocalUser*(m\_currentUser);  
  
 findViewById(R.id.*btn\_signUp*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 m\_currentUser.setUsername(input\_username.getText().toString());  
 m\_currentUser.setEmail(input\_email.getText().toString());  
 m\_currentUser.setPhone(input\_phone.getText().toString());  
 m\_currentUser.setFullName(input\_fullName.getText().toString());  
 m\_currentUser.setPassword(input\_password.getText().toString());  
 m\_currentUser.setLoggedIn("true");  
  
 String errorString = *ValidateSettings*(m\_currentUser);  
 if (errorString.equals("NO-ERROR"))  
 {  
 //Upload user  
 LocalSettings.*UpdateLocalUser*(m\_currentUser);  
 RegisterCallback rcb = new RegisterCallback();  
 rcb.SetMessage("IF NOT EXISTS (SELECT \* FROM " + m\_dbComms.*m\_userTable* + " WHERE name = '" + m\_currentUser.getUsername() + "')" + m\_currentUser.GetInsertString() + ";");  
 m\_dbComms.GenericUpload(rcb);  
 }  
 else  
 {  
 SetError(errorString);  
 }  
 }  
 });  
  
 m\_img\_image.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 startActivityForResult(new Intent(Intent.*ACTION\_PICK*, android.provider.MediaStore.Images.Media.*INTERNAL\_CONTENT\_URI*), 3);  
 }  
 });  
  
 findViewById(R.id.*btn\_addressSettings*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 startActivity(new Intent(SignUp.this, AddressEdit.class));  
 }  
 });  
  
 findViewById(R.id.*btn\_beAChef*).setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 startActivity(new Intent(SignUp.this, ChefAccountSettings.class));  
 }  
 });  
  
 m\_onDobSetListener = new DatePickerDialog.OnDateSetListener()  
 {  
 @Override  
 public void onDateSet(DatePicker datePicker, int year, int month, int day)  
 {  
 Date d = new Date(day, month, year);  
 if (d.before(Date.*from*(Instant.*now*())))  
 {  
 String dob = day + "-" + (month+1) + "-" + year;  
 m\_currentUser.setDOB(dob);  
 txt\_edit\_dob.setText(dob);  
 }  
 else  
 {  
 SetError("Invalid date of birth.");  
 }  
 }  
 };  
  
 txt\_edit\_dob.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View view)  
 {  
 int day = 1;  
 int month = 1;  
 int year = 2000;  
  
 DatePickerDialog datePickerDialog = new DatePickerDialog(SignUp.this, m\_onDobSetListener, year, month, day);  
 datePickerDialog.setTitle("Date of Birth");  
 datePickerDialog.getWindow().setBackgroundDrawable(new ColorDrawable(Color.*WHITE*));  
 datePickerDialog.show();  
 }  
 });  
 }  
  
 //Overridden so we can get the uploaded image  
 protected void onActivityResult(int requestCode, int resultCode, Intent data)  
 {  
 super.onActivityResult(requestCode, resultCode, data);  
  
 //GET\_FROM\_GALLERY = 3  
 if(requestCode==3 && resultCode == Activity.*RESULT\_OK*)  
 {  
 Uri selectedImage = data.getData();  
 try  
 {  
 m\_currentUser.setPicture(MediaStore.Images.Media.*getBitmap*(this.getContentResolver(), selectedImage));  
 m\_img\_image.setImageBitmap(m\_currentUser.getPicture());  
 }  
 catch (FileNotFoundException e)  
 {  
 SetError("Image file could not be found");  
 }  
 catch (IOException e)  
 {  
 SetError("An I/O error occured while acquiring the image");  
 }  
 }  
 }  
  
 @Override  
 protected void onResume()  
 {  
 super.onResume();  
 }  
  
 public static String ValidateSettings(User u)  
 {  
 if(u.getLongitude() == null || u.getLatitude() == null)  
 {  
 return "Need to add address";  
 }  
  
 //*todo validation...* return "NO-ERROR";  
 }  
  
 private void SetError(String errorString)  
 {  
 Toast t = Toast.*makeText*(SignUp.this, errorString, Toast.*LENGTH\_LONG*);  
 t.show();  
 }  
  
 private void NavigateLogin()  
 {  
 startActivity(new Intent(SignUp.this, Login.class));  
 }  
  
 @Override  
 public void onBackPressed()  
 {  
 NavigateLogin();  
 }  
  
 class RegisterCallback extends BaseCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 if(m\_message.equals("null"))  
 {  
 String select = "SELECT \* FROM " + m\_dbComms.*m\_userTable* + " WHERE name = '" + m\_currentUser.getUsername() + "' and password = '" + m\_currentUser.getPassword() + "' or contact\_email = '"  
 + m\_currentUser.getEmail() + "' and password = '" + m\_currentUser.getPassword() + "';";  
  
 UserDataCallback ucb = new UserDataCallback();  
 ucb.SetMessage(select);  
 m\_dbComms.RequestUserData(ucb);  
  
 LocalSettings.*SaveLoginDetails*(m\_currentUser.getUsername(), m\_currentUser.getPassword(), getApplicationContext());  
 startActivity(new Intent(SignUp.this, MainHub.class));  
 }  
 else  
 {  
 SetError("Failed to add user! Check internet connection");  
 }  
 }  
 });  
 return null;  
 }  
 }  
  
 private class UserDataCallback extends UsersCallback  
 {  
 @Override  
 public Void call() throws Exception  
 {  
 if (m\_users != null && !m\_users.isEmpty()) //Correct user found  
 {  
 LocalSettings.*UpdateLocalUser*(m\_users.get(0));  
 LocalSettings.*SaveLoginDetails*(m\_users.get(0).getUsername(), m\_users.get(0).getPassword(), SignUp.this);  
 startActivity(new Intent(SignUp.this, MainHub.class));  
 }  
 else  
 {  
 runOnUiThread(new Runnable()  
 {  
 @Override  
 public void run()  
 {  
 SetError("Incorrect Details");  
 }  
 });  
 }  
 return null;  
 }  
 }  
}

# activity\_address\_edit.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:theme="@android:style/Theme.Dialog"  
 tools:context="com.menu.menu.AddressEdit">  
  
 <EditText  
 android:id="@+id/input\_addressLine1"  
 style="@style/Widget.AppCompat.EditText"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_addressLine1" />  
  
 <EditText  
 android:id="@+id/input\_addressLine2"  
 style="@style/Widget.AppCompat.EditText"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_addressLine2" />  
  
 <EditText  
 android:id="@+id/input\_addressLine3"  
 style="@style/Widget.AppCompat.EditText"  
 android:layout\_width="0dp"  
 android:layout\_height="45dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_addressLine3" />  
  
 <EditText  
 android:id="@+id/input\_postCode"  
 style="@style/Widget.AppCompat.EditText"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_postCode" />  
  
 <TextView  
 android:id="@+id/txt\_postCode"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Post Code"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_addressLine3" />  
  
 <TextView  
 android:id="@+id/txt\_addressLine1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Address Line 1"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_addressLine2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Address Line 2"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_addressLine1" />  
  
 <TextView  
 android:id="@+id/txt\_addressLine3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Address Line 3"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_addressLine2" />  
  
 <Button  
 android:id="@+id/btn\_save"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 android:layout\_marginEnd="160dp"  
 android:layout\_marginStart="160dp"  
 android:text="Save"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
</android.support.constraint.ConstraintLayout>

# activity\_chef\_account\_settings.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context=".ChefAccountSettings">  
  
 <TextView  
 android:id="@+id/txt\_foodSpeciality"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Food Speciality"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/switch\_chef" />  
  
 <Switch  
 android:id="@+id/switch\_chef"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:text="Become a chef "  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_warning" />  
  
 <EditText  
 android:id="@+id/input\_foodType"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:text="Name"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_foodSpeciality" />  
  
 <TextView  
 android:id="@+id/txt\_disclaimer"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Disclaimer"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <android.support.v4.widget.NestedScrollView  
 android:id="@+id/nestedScrollView"  
 android:layout\_width="409dp"  
 android:layout\_height="194dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="2dp"  
 android:layout\_marginEnd="20dp"  
 android:fillViewport="true"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_disclaimer">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical" >  
  
 </LinearLayout>  
 </android.support.v4.widget.NestedScrollView>  
  
 <Button  
 android:id="@+id/btn\_save"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="20dp"  
 android:text="Save"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_warning"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="20dp"  
 android:lines="2"  
 android:singleLine="false"  
 android:text="By selecting this option you agree to the disclaimer above."  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/nestedScrollView" />  
  
 <TextView  
 android:id="@+id/txt\_chooseDateTo"  
 android:layout\_width="175dp"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="Choose Date"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="@+id/txt\_hoursAvaliableTo"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_hoursAvaliableTo" />  
  
 <TextView  
 android:id="@+id/txt\_hoursAvaliableTo"  
 android:layout\_width="175dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:gravity="center"  
 android:text="Hours Avaliable To"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_chooseDateFrom" />  
  
 <TextView  
 android:id="@+id/txt\_hoursAvaliableFrom"  
 android:layout\_width="175dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="16dp"  
 android:gravity="center"  
 android:text="Hours Avaliable From"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_mealTimes" />  
  
 <TextView  
 android:id="@+id/txt\_chooseDateFrom"  
 android:layout\_width="175dp"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="Choose Date"  
 android:textSize="18sp"  
 app:layout\_constraintEnd\_toEndOf="@+id/txt\_hoursAvaliableFrom"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_hoursAvaliableFrom" />  
  
 <TextView  
 android:id="@+id/txt\_mealTimes"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="30dp"  
 android:text="Set meal times"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_foodType" />  
  
</android.support.constraint.ConstraintLayout>

# activity\_chef\_meals.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context=".ChefMeals">  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="150dp"  
 android:layout\_height="150dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ListView  
 android:id="@+id/mealsList"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="10dp"  
 android:divider="@null"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_username">  
  
 </ListView>  
  
 <TextView  
 android:id="@+id/txt\_username"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="50dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="50dp"  
 android:text="TextView"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</android.support.constraint.ConstraintLayout>

# activity\_loader.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context=".Loader">  
  
 <ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="65dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="66dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/logo" />  
</android.support.constraint.ConstraintLayout>

# activity\_login.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context="com.menu.menu.Login">  
  
 <Button  
 android:id="@+id/btn\_login"  
 android:layout\_width="126dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="8dp"  
 android:layout\_marginEnd="148dp"  
 android:layout\_marginStart="148dp"  
 android:text="Login"  
 app:layout\_constraintBottom\_toTopOf="@+id/btn\_signUp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <EditText  
 android:id="@+id/input\_password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="60dp"  
 android:layout\_marginEnd="85dp"  
 android:layout\_marginStart="85dp"  
 android:ems="10"  
 android:inputType="textPassword"  
 app:layout\_constraintBottom\_toTopOf="@+id/btn\_login"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <EditText  
 android:id="@+id/input\_usernameOrEmail"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="20dp"  
 android:layout\_marginEnd="85dp"  
 android:layout\_marginStart="85dp"  
 android:ems="10"  
 android:inputType="textEmailAddress"  
 app:layout\_constraintBottom\_toTopOf="@+id/input\_password"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <Button  
 android:id="@+id/btn\_signUp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="24dp"  
 android:layout\_marginEnd="148dp"  
 android:layout\_marginStart="148dp"  
 android:text="Sign Up"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="329dp"  
 android:layout\_height="232dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="20dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/input\_usernameOrEmail"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/logo" />  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:layout\_marginStart="5dp"  
 android:progressTint="#F8F8F8"  
 app:layout\_constraintBottom\_toBottomOf="@+id/btn\_login"  
 app:layout\_constraintStart\_toEndOf="@+id/btn\_login"  
 app:layout\_constraintTop\_toTopOf="@+id/btn\_login" />  
</android.support.constraint.ConstraintLayout>

# activity\_main\_hub.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.v4.widget.DrawerLayout 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:id="@+id/drawer\_layout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:openDrawer="start">  
  
 <include  
 layout="@layout/app\_bar\_main\_hub"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
 <FrameLayout  
 android:id="@+id/fragmentHolder"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"/>  
  
 <android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_main\_hub"  
 app:menu="@menu/activity\_main\_hub\_drawer" />  
  
</android.support.v4.widget.DrawerLayout>

# activity\_meal\_registration.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context="com.menu.menu.MealRegistration">  
  
 <Button  
 android:id="@+id/btn\_viewAllergenInfo"  
 android:layout\_width="25dp"  
 android:layout\_height="25dp"  
 android:layout\_marginStart="8dp"  
 android:background="@android:drawable/ic\_menu\_info\_details"  
 android:backgroundTint="#FD0F0F0F"  
 app:layout\_constraintBottom\_toBottomOf="@+id/txt\_ingredients"  
 app:layout\_constraintStart\_toEndOf="@+id/txt\_fullName"  
 app:layout\_constraintTop\_toTopOf="@+id/txt\_fullName" />  
  
 <EditText  
 android:id="@+id/input\_name"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginEnd="16dp"  
 android:ems="10"  
 android:inputType="textEmailAddress"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_name" />  
  
 <TextView  
 android:id="@+id/txt\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="10dp"  
 android:text="Name"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/switch\_eatIn" />  
  
 <TextView  
 android:id="@+id/txt\_numberMeals"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="10dp"  
 android:text="Max Dishes"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_name" />  
  
 <TextView  
 android:id="@+id/txt\_fullName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="10dp"  
 android:text="Ingredients"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_numerOfDishes" />  
  
 <EditText  
 android:id="@+id/input\_numerOfDishes"  
 android:layout\_width="110dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:ems="10"  
 android:inputType="number"  
 android:textSize="14sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_numberMeals" />  
  
 <EditText  
 android:id="@+id/input\_ingredients"  
 android:layout\_width="347dp"  
 android:layout\_height="100dp"  
 android:layout\_marginStart="40dp"  
 android:layout\_marginTop="-10dp"  
 android:layout\_marginEnd="40dp"  
 android:ems="10"  
 android:inputType="textMultiLine"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.488"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_fullName" />  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/img\_image"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_marginStart="50dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginEnd="50dp"  
 app:civ\_border\_color="#FF000000"  
 app:civ\_border\_width="2dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/ic\_meal"  
 tools:srcCompat="@drawable/ic\_meal" />  
  
 <TextView  
 android:id="@+id/txt\_pricePerDish"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="16dp"  
 android:text="Price Per Dish"  
 android:textSize="18sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_name" />  
  
 <TextView  
 android:id="@+id/txt\_calories"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:text="Calories"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="@+id/input\_calories"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_name" />  
  
 <EditText  
 android:id="@+id/input\_price"  
 android:layout\_width="110dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="16dp"  
 android:ems="10"  
 android:inputType="numberDecimal"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_pricePerDish" />  
  
 <EditText  
 android:id="@+id/input\_calories"  
 android:layout\_width="110dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:inputType="numberDecimal"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toStartOf="@+id/input\_price"  
 app:layout\_constraintStart\_toEndOf="@+id/input\_numerOfDishes"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_calories" />  
  
 <Button  
 android:id="@+id/btn\_order"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 android:layout\_marginEnd="16dp"  
 android:text="Add"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
 <Button  
 android:id="@+id/btn\_delete"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="20dp"  
 android:layout\_marginEnd="20dp"  
 android:text="Delete"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
 <Switch  
 android:id="@+id/switch\_takeaway"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="27dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="60dp"  
 android:text="Takeaway "  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/img\_image" />  
  
 <Switch  
 android:id="@+id/switch\_eatIn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="60dp"  
 android:layout\_marginTop="10dp"  
 android:text="Eat In "  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/img\_image" />  
  
 <Button  
 android:id="@+id/btn\_update"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="10dp"  
 android:text="@string/common\_google\_play\_services\_update\_button"  
 app:layout\_constraintBottom\_toTopOf="@+id/btn\_order"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_HoursAvaliableFrom"  
 android:layout\_width="175dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:gravity="center"  
 android:text="Hours Avaliable From"  
 android:textSize="16sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_ingredients" />  
  
 <TextView  
 android:id="@+id/txt\_HoursAvaliableTo"  
 android:layout\_width="175dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:gravity="center"  
 android:text="Hours Avaliable To"  
 android:textSize="16sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/txt\_HoursAvaliableFrom"  
 app:layout\_constraintTop\_toTopOf="@+id/txt\_HoursAvaliableFrom" />  
  
 <TextView  
 android:id="@+id/txt\_pickDateFrom"  
 android:layout\_width="175dp"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="Choose Date"  
 android:textSize="18sp"  
 app:layout\_constraintEnd\_toEndOf="@+id/txt\_HoursAvaliableFrom"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_HoursAvaliableFrom" />  
  
 <TextView  
 android:id="@+id/txt\_pickDateTo"  
 android:layout\_width="175dp"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:text="Choose Date"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="@+id/txt\_HoursAvaliableTo"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_HoursAvaliableTo" />  
  
</android.support.constraint.ConstraintLayout>

# activity\_meal\_view.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context="com.menu.menu.MealView">  
  
 <TextView  
 android:id="@+id/txt\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="33dp"  
 android:layout\_marginEnd="10dp"  
 android:text="Name"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toEndOf="@+id/img\_image"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_ingredients"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Ingredients"  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/img\_image" />  
  
 <TextView  
 android:id="@+id/txt\_Address"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Address"  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_ingredients" />  
  
 <TextView  
 android:id="@+id/txt\_addressFill"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Address"  
 android:textSize="14sp"  
 app:layout\_constraintStart\_toStartOf="@+id/txt\_Address"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_Address" />  
  
 <EditText  
 android:id="@+id/input\_numerOfDishes"  
 android:layout\_width="162dp"  
 android:layout\_height="42dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginBottom="20dp"  
 android:ems="10"  
 android:inputType="number"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/switch\_takeaway"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_numberMeals"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="32dp"  
 android:layout\_marginStart="124dp"  
 android:layout\_marginEnd="124dp"  
 android:text="Number of dishes"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/input\_numerOfDishes"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:id="@+id/img\_image"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:civ\_border\_width="2dp"  
 app:civ\_border\_color="#FF000000"/>  
  
 <TextView  
 android:id="@+id/txt\_pricePerDish"  
 android:layout\_width="130dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="28dp"  
 android:text="Price Per Dish"  
 android:textAlignment="center"  
 android:textSize="20sp"  
 app:layout\_constraintStart\_toEndOf="@+id/img\_image"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_name" />  
  
 <TextView  
 android:id="@+id/txt\_rating"  
 android:layout\_width="130dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="28dp"  
 android:layout\_marginEnd="20dp"  
 android:text="0/5"  
 android:textAlignment="center"  
 android:textSize="20sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_name" />  
  
 <Button  
 android:id="@+id/btn\_order"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="20dp"  
 android:layout\_marginEnd="20dp"  
 android:text="Order"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
 <TextView  
 android:id="@+id/input\_ingredients"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_ingredients" />  
  
 <Switch  
 android:id="@+id/switch\_eatIn"  
 android:layout\_width="140dp"  
 android:layout\_height="32dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginBottom="20dp"  
 android:text="Eat In"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <Switch  
 android:id="@+id/switch\_takeaway"  
 android:layout\_width="140dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:text="Takeaway"  
 android:textSize="18sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/switch\_eatIn"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <View  
 android:id="@+id/divider2"  
 android:layout\_width="409dp"  
 android:layout\_height="1dp"  
 android:layout\_marginBottom="16dp"  
 android:background="?android:attr/listDivider"  
 app:layout\_constraintBottom\_toTopOf="@+id/txt\_numberMeals"  
 tools:layout\_editor\_absoluteX="1dp" />  
  
 <Button  
 android:id="@+id/btn\_viewAllergenInfo"  
 android:layout\_width="25dp"  
 android:layout\_height="25dp"  
 android:layout\_marginStart="3dp"  
 android:background="@android:drawable/ic\_menu\_info\_details"  
 android:backgroundTint="#FD0F0F0F"  
 app:layout\_constraintBottom\_toBottomOf="@+id/txt\_ingredients"  
 app:layout\_constraintStart\_toEndOf="@+id/txt\_ingredients"  
 app:layout\_constraintTop\_toTopOf="@+id/txt\_ingredients" />  
  
</android.support.constraint.ConstraintLayout>

# activity\_meetup\_chat.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context="com.menu.menu.MeetupChat">  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="150dp"  
 android:layout\_height="150dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/btn\_postMessage"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginBottom="20dp"  
 android:text="Add"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
 <EditText  
 android:id="@+id/txt\_messageInput"  
 android:layout\_width="0dp"  
 android:layout\_height="46dp"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginEnd="24dp"  
 android:layout\_marginBottom="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/btn\_postMessage"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_addressLine1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:text="TextView"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_meetingName" />  
  
 <TextView  
 android:id="@+id/txt\_addressLine2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 app:layout\_constraintStart\_toStartOf="@+id/txt\_addressLine1"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_addressLine1" />  
  
 <TextView  
 android:id="@+id/txt\_meetingName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginEnd="40dp"  
 android:text="TextView"  
 android:textSize="18sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_addressLine3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 app:layout\_constraintStart\_toStartOf="@+id/txt\_addressLine1"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_addressLine2" />  
  
 <TextView  
 android:id="@+id/txt\_addressPostCode"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 app:layout\_constraintStart\_toStartOf="@+id/txt\_addressLine1"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_addressLine3" />  
  
 <View  
 android:id="@+id/divider"  
 android:layout\_width="409dp"  
 android:layout\_height="1dp"  
 android:layout\_marginStart="1dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="1dp"  
 android:background="?android:attr/listDivider"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.166"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_addressPostCode" />  
  
 <ListView  
 android:id="@+id/listView\_messages"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="10dp"  
 android:divider="@null"  
 app:layout\_constraintBottom\_toTopOf="@+id/txt\_messageInput"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/divider" />  
  
 <Button  
 android:id="@+id/btn\_finish"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="14dp"  
 android:layout\_marginEnd="14dp"  
 android:text="Done"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</android.support.constraint.ConstraintLayout>

# activity\_payment\_processing.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context="com.menu.menu.PaymentProcessing">  
  
 <Button  
 android:id="@+id/btn\_pay"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Pay"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Meal : "  
 android:textSize="24sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_price"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="8dp"  
 android:text="Price : "  
 android:textSize="24sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_name" />  
  
</android.support.constraint.ConstraintLayout>

# activity\_search\_results.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context="com.menu.menu.SearchResults">  
  
 <Switch  
 android:id="@+id/switch\_users"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="5dp"  
 android:text="Users / Meals "  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_search" />  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="150dp"  
 android:layout\_height="150dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/btn\_home"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Home"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ListView  
 android:id="@+id/mealsList"  
 android:layout\_width="368dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:divider="@null"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/switch\_users">  
  
 </ListView>  
  
 <EditText  
 android:id="@+id/input\_search"  
 android:layout\_width="238dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 app:layout\_constraintEnd\_toStartOf="@+id/btn\_home"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</android.support.constraint.ConstraintLayout>

# activity\_sign\_up.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 tools:context="com.menu.menu.SignUp">  
  
 <TextView  
 android:id="@+id/txt\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:text="Username"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/img\_image" />  
  
 <EditText  
 android:id="@+id/input\_name"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:inputType="textEmailAddress"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_name" />  
  
 <TextView  
 android:id="@+id/txt\_email"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Email"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_name" />  
  
 <EditText  
 android:id="@+id/input\_email"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:inputType="textEmailAddress"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_email" />  
  
 <TextView  
 android:id="@+id/txt\_phone"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Phone"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_email" />  
  
 <EditText  
 android:id="@+id/input\_phone"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:inputType="phone"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_phone" />  
  
  
 <TextView  
 android:id="@+id/txt\_fullName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Full Name"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_phone" />  
  
 <EditText  
 android:id="@+id/input\_fullName"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.037"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_fullName" />  
  
 <TextView  
 android:id="@+id/txt\_dob"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Date Of Birth"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_fullName" />  
  
 <TextView  
 android:id="@+id/txt\_edit\_dob"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="16dp"  
 android:text="00/00/00"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_dob" />  
  
 <TextView  
 android:id="@+id/txt\_password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Password"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_edit\_dob" />  
  
 <EditText  
 android:id="@+id/input\_password"  
 android:layout\_width="350dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:inputType="textPassword"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.523"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_password" />  
  
 <Button  
 android:id="@+id/btn\_addressSettings"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="50dp"  
 android:layout\_marginEnd="50dp"  
 android:text="Address"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/btn\_beAChef" />  
  
 <Button  
 android:id="@+id/btn\_signUp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="20dp"  
 android:text="Sign Up"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
 <Button  
 android:id="@+id/btn\_beAChef"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:text="Be a Chef"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/btn\_signUp" />  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/img\_image"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_marginStart="50dp"  
 android:layout\_marginTop="20dp"  
 android:layout\_marginEnd="50dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:civ\_border\_color="#FF000000"  
 app:civ\_border\_width="2dp"  
 app:srcCompat="@drawable/ic\_meal"  
 tools:srcCompat="@tools:sample/avatars[0]" />  
  
</android.support.constraint.ConstraintLayout>

# app\_bar\_main\_hub.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.design.widget.CoordinatorLayout 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"  
 tools:context=".MainHub">  
  
 <include layout="@layout/content\_main\_hub" />  
  
</android.support.design.widget.CoordinatorLayout>

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:showIn="@layout/app\_bar\_main\_hub">  
  
 <fragment  
 android:id="@+id/nav\_host\_fragment"  
 android:name="androidx.navigation.fragment.NavHostFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:defaultNavHost="true"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
</android.support.constraint.ConstraintLayout>

# content\_main\_hub.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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="140dp"  
 android:layout\_height="200dp"  
 android:orientation="vertical">  
  
 <RadioButton  
 android:id="@+id/radio\_isVegan"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="20dp"  
 android:enabled="true"  
 android:text="Vegan "  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <RadioButton  
 android:id="@+id/radio\_isVegetarian"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:enabled="true"  
 android:text="Vegetarian "  
 app:layout\_constraintStart\_toStartOf="@+id/radio\_isVegan"  
 app:layout\_constraintTop\_toBottomOf="@+id/radio\_isVegan" />  
  
 <RadioButton  
 android:id="@+id/radio\_containsGluten"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:enabled="true"  
 android:text="Gluten "  
 app:layout\_constraintStart\_toStartOf="@+id/radio\_containsMilk"  
 app:layout\_constraintTop\_toBottomOf="@+id/radio\_containsMilk" />  
  
 <RadioButton  
 android:id="@+id/radio\_containsMilk"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:enabled="true"  
 android:text="Milk "  
 app:layout\_constraintStart\_toStartOf="@+id/radio\_isVegetarian"  
 app:layout\_constraintTop\_toBottomOf="@+id/radio\_isVegetarian" />  
  
 <RadioButton  
 android:id="@+id/radio\_isHalal"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:enabled="true"  
 android:text="Halal "  
 app:layout\_constraintStart\_toStartOf="@+id/radio\_containsGluten"  
 app:layout\_constraintTop\_toBottomOf="@+id/radio\_containsGluten" />  
  
</android.support.constraint.ConstraintLayout>

# dialog\_alergen\_info\_editable.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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="140dp"  
 android:layout\_height="180dp"  
 android:orientation="vertical">  
  
 <Switch  
 android:id="@+id/radio\_isVegan"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="20dp"  
 android:enabled="true"  
 android:text="Vegan "  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Switch  
 android:id="@+id/radio\_isVegetarian"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:enabled="true"  
 android:text="Vegetarian "  
 android:textAlignment="center"  
 app:layout\_constraintStart\_toStartOf="@+id/radio\_isVegan"  
 app:layout\_constraintTop\_toBottomOf="@+id/radio\_isVegan" />  
  
 <Switch  
 android:id="@+id/radio\_containsGluten"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:enabled="true"  
 android:text="Gluten "  
 app:layout\_constraintStart\_toStartOf="@+id/radio\_containsMilk"  
 app:layout\_constraintTop\_toBottomOf="@+id/radio\_containsMilk" />  
  
 <Switch  
 android:id="@+id/radio\_containsMilk"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:enabled="true"  
 android:text="Milk "  
 app:layout\_constraintStart\_toStartOf="@+id/radio\_isVegetarian"  
 app:layout\_constraintTop\_toBottomOf="@+id/radio\_isVegetarian" />  
  
 <Switch  
 android:id="@+id/radio\_isHalal"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:enabled="true"  
 android:text="Halal "  
 app:layout\_constraintStart\_toStartOf="@+id/radio\_containsGluten"  
 app:layout\_constraintTop\_toBottomOf="@+id/radio\_containsGluten" />  
  
</android.support.constraint.ConstraintLayout>

# dialog\_rating.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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="300dp"  
 android:layout\_height="180dp"  
 android:orientation="vertical">  
  
 <RatingBar  
 android:id="@+id/ratingBar"  
 android:layout\_width="248dp"  
 android:layout\_height="57dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_weight="1"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/btn\_done"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="5dp"  
 android:text="Done"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/ratingBar" />  
  
</android.support.constraint.ConstraintLayout>

# fragment\_basket.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 tools:context=".Basket">  
  
 <Button  
 android:id="@+id/btn\_checkout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="20dp"  
 android:text="Checkout"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent" />  
  
 <android.support.design.widget.FloatingActionButton  
 android:id="@+id/drawerToggle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginHorizontal="5dp"  
 android:layout\_marginVertical="5dp"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:clickable="true"  
 app:layout\_anchorGravity="end|bottom"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:maxImageSize="20dp"  
 app:srcCompat="?attr/actionModeCloseDrawable" />  
  
 <TextView  
 android:id="@+id/txt\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:gravity="center"  
 android:text="Basket"  
 android:textSize="24sp" />  
  
 <ListView  
 android:id="@+id/listView\_basket"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginBottom="10dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/btn\_checkout"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_title">  
  
 </ListView>  
  
</android.support.constraint.ConstraintLayout>

# fragment\_home.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <android.support.design.widget.FloatingActionButton  
 android:id="@+id/drawerToggle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginHorizontal="5dp"  
 android:layout\_marginVertical="5dp"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:clickable="true"  
 app:layout\_anchorGravity="end|bottom"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:maxImageSize="20dp"  
 app:srcCompat="?attr/actionModeCloseDrawable" />  
  
 <com.google.android.gms.maps.MapView  
 android:id="@+id/mapView"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:visibility="visible"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.454"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:visibility="visible" />  
  
 <SearchView  
 android:id="@+id/searchView"  
 android:layout\_width="310dp"  
 android:layout\_height="61dp"  
 android:layout\_marginStart="15dp"  
 android:layout\_marginTop="4dp"  
 android:layout\_marginEnd="10dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.6"  
 app:layout\_constraintStart\_toEndOf="@+id/drawerToggle"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:visibility="visible" />  
  
 <android.support.design.widget.FloatingActionButton  
 android:id="@+id/btn\_locate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:clickable="true"  
 app:layout\_constraintStart\_toStartOf="@+id/btn\_refresh"  
 app:layout\_constraintTop\_toBottomOf="@+id/btn\_refresh"  
 app:srcCompat="@android:drawable/ic\_menu\_mylocation" />  
  
 <android.support.design.widget.FloatingActionButton  
 android:id="@+id/btn\_refresh"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:clickable="true"  
 app:layout\_constraintStart\_toStartOf="@+id/drawerToggle"  
 app:layout\_constraintTop\_toBottomOf="@+id/drawerToggle"  
 app:srcCompat="@drawable/ic\_meal" />  
  
</android.support.constraint.ConstraintLayout>

# fragment\_my\_meals.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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=".MyMeals">  
  
 <android.support.design.widget.FloatingActionButton  
 android:id="@+id/drawerToggle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginHorizontal="5dp"  
 android:layout\_marginVertical="5dp"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="16dp"  
 android:clickable="true"  
 app:layout\_anchorGravity="end|bottom"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:maxImageSize="20dp"  
 app:srcCompat="?attr/actionModeCloseDrawable" />  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="150dp"  
 android:layout\_height="150dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ListView  
 android:id="@+id/mealsList"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="16dp"  
 android:divider="@null"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btn\_addMeal" />  
  
 <Button  
 android:id="@+id/btn\_addMeal"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginEnd="20dp"  
 android:text="New"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_myMeals"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="40dp"  
 android:text="My Meals"  
 android:textSize="24sp"  
 app:layout\_constraintBottom\_toBottomOf="@+id/btn\_addMeal"  
 app:layout\_constraintEnd\_toStartOf="@+id/btn\_addMeal"  
 app:layout\_constraintHorizontal\_bias="0.537"  
 app:layout\_constraintStart\_toEndOf="@+id/drawerToggle"  
 app:layout\_constraintTop\_toTopOf="@+id/btn\_addMeal" />  
  
</android.support.constraint.ConstraintLayout>

# fragment\_my\_orders.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 tools:context=".MyOrders">  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="150dp"  
 android:layout\_height="150dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/txt\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:gravity="center"  
 android:text="My Orders"  
 android:textSize="24sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ListView  
 android:id="@+id/listView\_orders"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="60dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_title">  
  
 </ListView>  
  
 <android.support.design.widget.FloatingActionButton  
 android:id="@+id/drawerToggle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginHorizontal="5dp"  
 android:layout\_marginVertical="5dp"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:clickable="true"  
 app:layout\_anchorGravity="end|bottom"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:maxImageSize="20dp"  
 app:srcCompat="?attr/actionModeCloseDrawable" />  
  
</android.support.constraint.ConstraintLayout>

# fragment\_settings.xml

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.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=".Settings">  
  
 <android.support.design.widget.FloatingActionButton  
 android:id="@+id/drawerToggle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginHorizontal="5dp"  
 android:layout\_marginVertical="5dp"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 android:clickable="true"  
 app:layout\_anchorGravity="end|bottom"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:maxImageSize="20dp"  
 app:srcCompat="?attr/actionModeCloseDrawable" />  
  
 <EditText  
 android:id="@+id/input\_name"  
 android:layout\_width="350dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginStart="20dp"  
 android:ems="10"  
 android:inputType="textEmailAddress"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_name" />  
  
 <EditText  
 android:id="@+id/input\_phone"  
 android:layout\_width="350dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginStart="20dp"  
 android:ems="10"  
 android:inputType="phone"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_numberMeals" />  
  
 <EditText  
 android:id="@+id/input\_password"  
 android:layout\_width="350dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginStart="20dp"  
 android:ems="10"  
 android:inputType="textPassword"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_password" />  
  
 <EditText  
 android:id="@+id/input\_fullName"  
 android:layout\_width="350dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 android:ems="10"  
 android:inputType="textPersonName"  
 android:textSize="14sp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_fullName" />  
  
 <TextView  
 android:id="@+id/txt\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:text="Email"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/img\_image" />  
  
 <TextView  
 android:id="@+id/txt\_numberMeals"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Phone"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_name" />  
  
 <TextView  
 android:id="@+id/txt\_fullName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Full Name"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_phone" />  
  
 <TextView  
 android:id="@+id/txt\_password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="20dp"  
 android:text="Password"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_edit\_dob" />  
  
 <Button  
 android:id="@+id/btn\_addressSettings"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:text="Address"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/btn\_chef" />  
  
 <Button  
 android:id="@+id/btn\_saveSettigns"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:layout\_marginBottom="20dp"  
 android:text="Save"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent" />  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/img\_image"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_marginStart="50dp"  
 android:layout\_marginTop="16dp"  
 android:layout\_marginEnd="50dp"  
 app:civ\_border\_color="#FF000000"  
 app:civ\_border\_width="2dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/ic\_meal"  
 tools:srcCompat="@drawable/ic\_meal" />  
  
 <Button  
 android:id="@+id/btn\_chef"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="50dp"  
 android:layout\_marginEnd="50dp"  
 android:text="Be A Chef"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/btn\_saveSettigns" />  
  
 <TextView  
 android:id="@+id/txt\_dob"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="20dp"  
 android:text="Date Of Birth"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/input\_fullName" />  
  
 <TextView  
 android:id="@+id/txt\_edit\_dob"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="12dp"  
 android:text="00/00/00"  
 android:textSize="18sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_dob" />  
  
</android.support.constraint.ConstraintLayout>

# layout\_basket\_item.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <TextView  
 android:id="@+id/li\_text"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:layout\_below="@+id/li\_top"  
 android:layout\_alignBottom="@+id/img"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginEnd="2dp"  
 android:layout\_marginBottom="0dp"  
 android:layout\_toStartOf="@+id/btn\_portionsSub"  
 android:layout\_toEndOf="@+id/img"  
 android:text="Text"  
 android:textAlignment="textStart"  
 android:textSize="18sp" />  
  
 <TextView  
 android:id="@+id/li\_amount"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:layout\_alignTop="@+id/btn\_portionsAdd"  
 android:layout\_marginTop="0dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_toStartOf="@+id/btn\_portionsAdd"  
 android:text="0"  
 android:gravity="center"  
 android:textSize="16sp" />  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/img"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 app:civ\_border\_color="#FF000000"  
 app:civ\_border\_width="2dp"  
 app:srcCompat="@drawable/ic\_meal" />  
  
 <Button  
 android:id="@+id/btn\_remove"  
 style="@style/Widget.AppCompat.Button.Small"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="0dp"  
 android:gravity="center"  
 android:text="x" />  
  
 <Button  
 android:id="@+id/btn\_portionsAdd"  
 style="@style/Widget.AppCompat.Button.Small"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:layout\_below="@+id/btn\_remove"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginEnd="0dp"  
 android:gravity="center"  
 android:text="+" />  
  
 <Button  
 android:id="@+id/btn\_portionsSub"  
 style="@style/Widget.AppCompat.Button.Small"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp"  
 android:layout\_alignTop="@+id/li\_amount"  
 android:layout\_marginTop="0dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_toStartOf="@+id/li\_amount"  
 android:gravity="center"  
 android:text="-" />  
  
 <TextView  
 android:id="@+id/li\_top"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="5dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_toStartOf="@+id/btn\_remove"  
 android:layout\_toEndOf="@+id/img"  
 android:text="TextView" />  
  
</RelativeLayout>

# layout\_meal\_info.xml

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <TextView  
 android:id="@+id/txt\_chefMessage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="110dp"  
 android:layout\_marginLeft="5dp"  
 android:layout\_marginTop="5dp"  
 android:layout\_marginRight="5dp"  
 android:background="@drawable/listitembackground" />  
  
 <TextView  
 android:id="@+id/li\_text"  
 android:layout\_width="280dp"  
 android:layout\_height="55dp"  
 android:layout\_below="@+id/li\_infoRight"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginStart="120dp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginTop="12dp"  
 android:layout\_marginEnd="10dp"  
 android:layout\_marginRight="15dp"  
 android:layout\_marginBottom="10dp"  
 android:text="Text"  
 android:textAlignment="textStart"  
 android:textSize="18sp" />  
  
 <TextView  
 android:id="@+id/li\_infoRight"  
 android:layout\_width="141dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginStart="260dp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="15dp"  
 android:layout\_marginRight="15dp"  
 android:layout\_marginBottom="10dp"  
 android:text="-"  
 android:textAlignment="textEnd"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/li\_infoLeft"  
 android:layout\_width="141dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginStart="120dp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginTop="11dp"  
 android:layout\_marginEnd="150dp"  
 android:layout\_marginRight="10dp"  
 android:layout\_marginBottom="10dp"  
 android:text="-"  
 android:textSize="16sp" />  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/img"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 app:civ\_border\_color="#FF000000"  
 app:civ\_border\_width="2dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/ic\_meal"  
 tools:srcCompat="@drawable/ic\_meal" />  
  
</RelativeLayout>

# layout\_message\_chef.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/txt\_chefMessage"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="60dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginRight="5dp"  
 android:layout\_weight="1"  
 android:background="#D1D0D0"  
 android:paddingLeft="4dp"  
 android:paddingTop="4dp"  
 android:paddingRight="4dp"  
 android:paddingBottom="4dp"  
 android:text="TextView"  
 android:textAlignment="textEnd"  
 android:textColor="#000000" />  
</LinearLayout>

# layout\_message\_eater.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/txt\_eaterMessage"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="5dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginRight="60dp"  
 android:layout\_weight="1"  
 android:background="#D1D0D0"  
 android:paddingLeft="4dp"  
 android:paddingTop="4dp"  
 android:paddingRight="4dp"  
 android:paddingBottom="4dp"  
 android:text="TextView"  
 android:textColor="#000000"  
 android:visibility="visible" />  
</LinearLayout>

# layout\_order\_info.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <TextView  
 android:id="@+id/li\_text"  
 android:layout\_width="294dp"  
 android:layout\_height="46dp"  
 android:layout\_below="@+id/li\_top"  
 android:layout\_alignBottom="@+id/img"  
 android:layout\_marginStart="5dp"  
 android:layout\_marginTop="35dp"  
 android:layout\_marginEnd="2dp"  
 android:layout\_marginBottom="0dp"  
 android:layout\_toStartOf="@+id/btn\_portionsSub"  
 android:layout\_toEndOf="@+id/img"  
 android:text="Text"  
 android:textAlignment="textStart"  
 android:textSize="18sp" />  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/img"  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginTop="10dp"  
 app:civ\_border\_color="#FF000000"  
 app:civ\_border\_width="2dp"  
 app:srcCompat="@drawable/ic\_meal" />  
  
 <TextView  
 android:id="@+id/li\_top"  
 android:layout\_width="204dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_marginStart="5dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_toStartOf="@+id/btn\_remove"  
 android:layout\_toEndOf="@+id/img"  
 android:text="TextView" />  
  
 <TextView  
 android:id="@+id/li\_top2"  
 android:layout\_width="294dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginStart="5dp"  
 android:layout\_marginTop="36dp"  
 android:layout\_marginEnd="2dp"  
 android:layout\_toEndOf="@+id/img"  
 android:text="TextView" />  
  
</RelativeLayout>

# nav\_header\_main\_hub.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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="@dimen/nav\_header\_height"  
 android:background="@drawable/side\_nav\_bar"  
 android:gravity="bottom"  
 android:orientation="vertical"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:theme="@style/ThemeOverlay.AppCompat.Dark">  
   
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/nav\_bar\_image"  
 android:contentDescription="@string/nav\_header\_desc"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 app:civ\_border\_color="#FF000000"  
 app:civ\_border\_width="2dp"  
 android:paddingTop="8dp"  
 app:srcCompat="@drawable/ic\_meal" />  
  
 <TextView  
 android:id="@+id/nav\_bar\_title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingTop="@dimen/nav\_header\_vertical\_spacing"  
 android:text="MyNavigationTitle"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1" />  
  
 <TextView  
 android:id="@+id/nav\_bar\_subTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="MyNavigationSubtitle" />  
  
</LinearLayout>

# activity\_main\_hub\_drawer.xml

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 tools:showIn="navigation\_view">  
  
 <group android:checkableBehavior="single">  
 <item  
 android:id="@+id/nav\_home"  
 android:icon="@drawable/ic\_home"  
 android:title="Home" />  
 <item  
 android:id="@+id/nav\_chefSettigns"  
 android:icon="@drawable/ic\_meal"  
 android:title="My Meals" />  
 <item  
 android:id="@+id/nav\_settings"  
 android:icon="@drawable/ic\_settings"  
 android:title="Settings"/>  
 <item  
 android:id="@+id/nav\_basket"  
 android:icon="@drawable/ic\_basket"  
 android:title="Basket"/>  
 <item  
 android:id="@+id/nav\_currentOrders"  
 android:icon="@drawable/ic\_meal"  
 android:title="Current Orders"/>  
 </group>  
  
 <item android:title="Options">  
 <menu>  
 <item  
 android:id="@+id/nav\_signOut"  
 android:icon="@drawable/ic\_menu\_share"  
 android:title="Sign Out" />  
 <item  
 android:id="@+id/nav\_reportIssue"  
 android:icon="@android:drawable/ic\_menu\_report\_image"  
 android:title="Report Issue" />  
 </menu>  
 </item>  
  
</menu>

# databaseAPI.php

<?php

$serverName = "den1.mssql7.gear.host";

$uid = "menudatabase";

$pwd = "Ss6y7-?5dgjN";

$databaseName = "menudatabase";

$conn = sqlsrv\_connect($serverName, array( "UID"=>$uid, "PWD"=>$pwd, "Database"=>$databaseName));

/\*$tsql = "SELECT \* FROM [menudatabase].[dbo].[User];";\*/

$tsql = $\_REQUEST['request'];

/\* Execute the query. \*/

$stmt = sqlsrv\_query($conn, $tsql);

if ( $stmt )

{

/\* Iterate through the result set printing a row of data upon each iteration.\*/

while( $row = sqlsrv\_fetch\_array( $stmt, SQLSRV\_FETCH\_NUMERIC))

{

$output[]=$row;

}

echo str\_replace('"', '', json\_encode($output));

}

else

{

echo "Error \n";

die( print\_r( sqlsrv\_errors(), true));

}

/\* Free statement and connection resources. \*/

sqlsrv\_free\_stmt( $stmt);

sqlsrv\_close( $conn);

?>

# databaseAPIpost.php

<?php

$serverName = "den1.mssql7.gear.host";

$uid = "menudatabase";

$pwd = "Ss6y7-?5dgjN";

$databaseName = "menudatabase";

$conn = sqlsrv\_connect($serverName, array( "UID"=>$uid, "PWD"=>$pwd, "Database"=>$databaseName));

/\*$tsql = "SELECT \* FROM [menudatabase].[dbo].[User];";\*/

$tsql = $\_POST['request'];

/\* Execute the query. \*/

$stmt = sqlsrv\_query($conn, $tsql);

if ( $stmt )

{

/\* Iterate through the result set printing a row of data upon each iteration.\*/

while( $row = sqlsrv\_fetch\_array( $stmt, SQLSRV\_FETCH\_NUMERIC))

{

$output[]=$row;

}

echo str\_replace('"', '', json\_encode($output));

}

else

{

echo "Error \n";

die( print\_r( sqlsrv\_errors(), true));

}

/\* Free statement and connection resources. \*/

sqlsrv\_free\_stmt( $stmt);

sqlsrv\_close( $conn);

?>