

DEVELOPPEMENT MOBILE

Rapport Mini projet

-CommandCake App-

Réalisé par :

ABCHIR OUIJDANE **Encadré par :**Mme RACHIK ZINEB

INTRODUCTION:

Le mini projet Android qui est dans le cadre de l'élément DEVELOPPEMENT MOBILE enseigné par Mme RACHIK ZINEB, vise à créer une application Android mobile qui traite la gestion du service de création de de livraison de gâteaux sur commande. En outre, l'application gère la gestion des compte et des commandes enregistrées et passées (confirmées pour la procédure de livraison).

Mon application Android nommée CommandCake traite en détail les 4 fonctionnalités demandées pourtant elle est simple esthétiquement mais concise au niveau de gestion des cas extrêmes et les erreurs de saisie grâce à la précision du développement au niveau de la gestion des tables par SQLite.

Elle traite de plus le service de mailing.

Nous allons ainsi manifester la manière de fonctionnement pour chaque fonctionnalité implémentée.

1. Création du compte

On montre tout d'abord les précautions prises pour gérer les cas extrêmes(Robustesse) et éviter les bugs de saisie :

Name :

Password : _____

Check Password :



Email:

SIGN UP





Name:

Password:

Check Password :



Email:

SIGN UP

Write your information





Name: abchir

Password:

Check Password :



Email:

SIGN UP

An account already created with the same information

	_		
Į	*	o	ŀ



Name: ouijdane

Password : _____

Check Password :



Email:

SIGN UP

Complete your information





Name: ouijdane

Password : _____

Check Password:



Email: ouijdane.abchir15@outlook.fr

SIGN UP

An account already created with the same information





Name: ouijdane

Password: *****

Check Password :



Email: ouijdane@gmail.com

SIGN UP

The Checked Password doesn't match the Password





Name: ouijdane

Password: *****

Check Password : ••



Email: ouijdane@gmail.com

SIGN UP

The Checked Password doesn't match the Password

	ি না 67% দ্ৰ 14:57
CakeCommand	
Name :	
Password :	
SIGN IN	
Don't you have an ac	ccount?

SIGN UP

Your account has been created

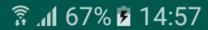
2. Authentification

On montre de même les cas gérés pour la gestion des bugs de saisie :

		និ .d 67% ន 14:57
CakeComm	and	
Name :		
Password:		
	SIGN IN	
Don't yo	ou have an a	ccount?
	SIGN UP	

Write your name and password

_		
	0	þ



Name: abchir

Password : _____

SIGN IN



----- Don't you have an account ? -----

SIGN UP

Write your password

	🚡 📶 67% 🖪 14:57
CakeCommand	
Name :	
Password : *****	
SIGN IN	
Don't you have an ac	ccount?

SIGN UP

Write your name

_		
	o	ı



Name: abchir

Password: ****

SIGN IN

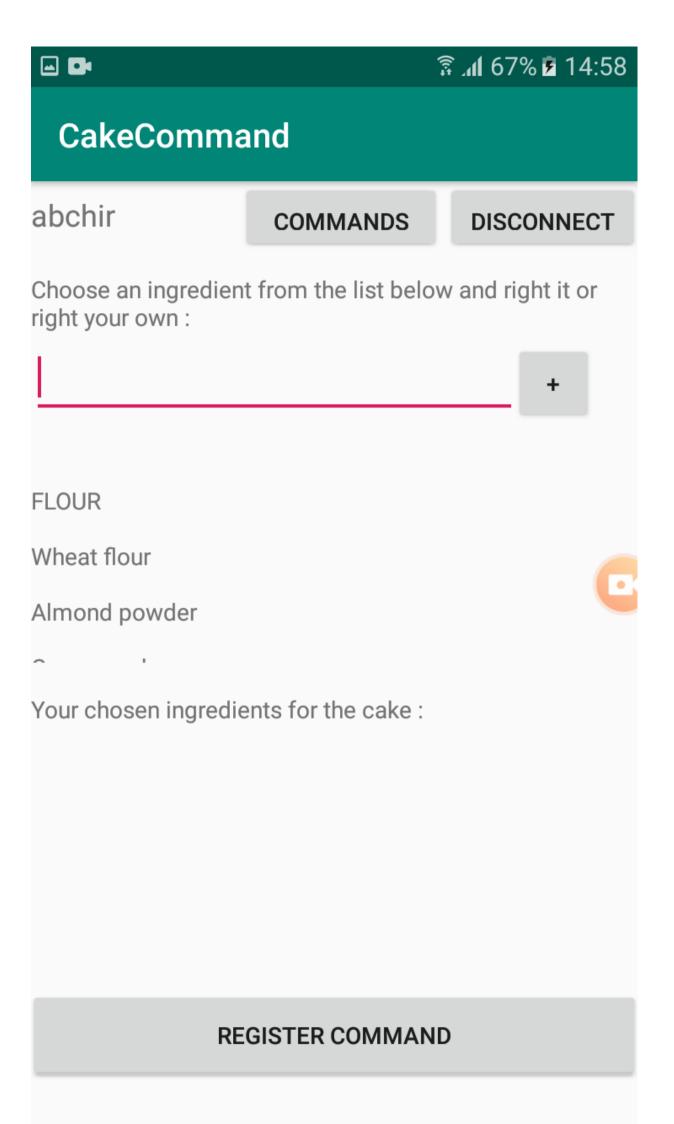


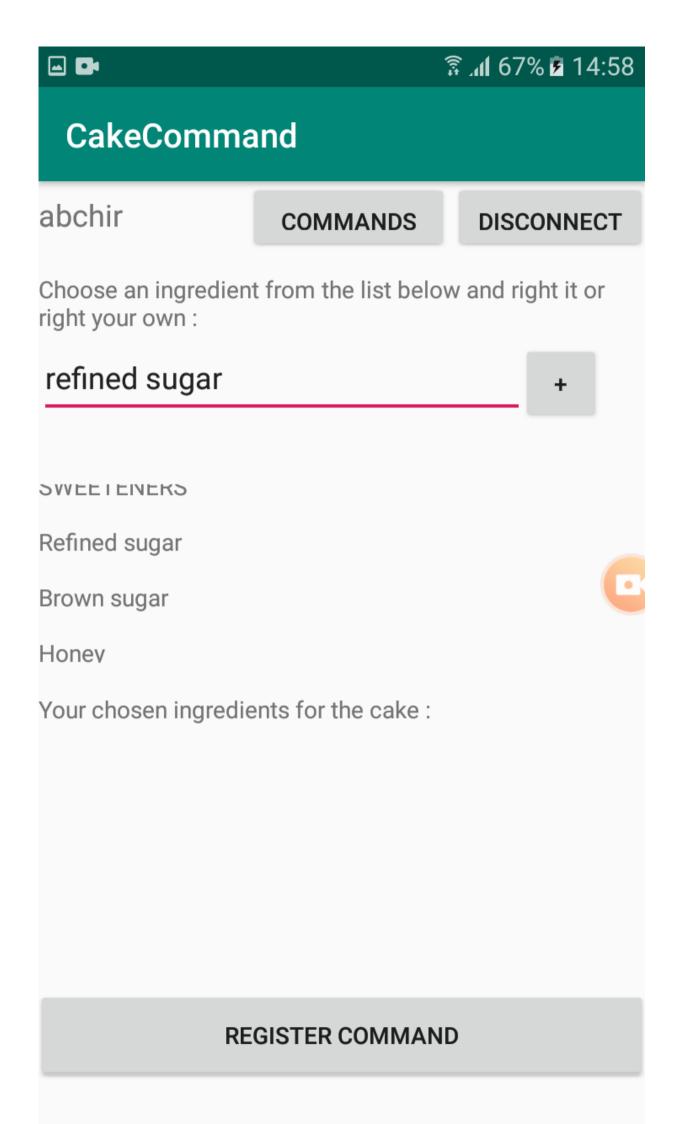
----- Don't you have an account? -----

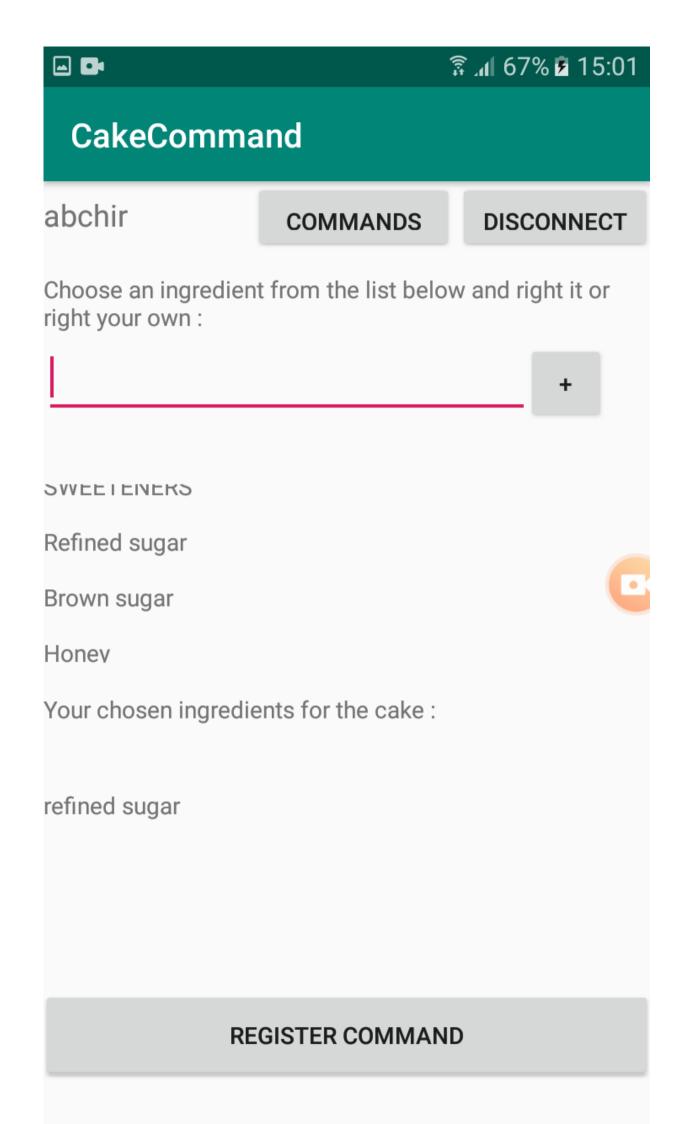
SIGN UP

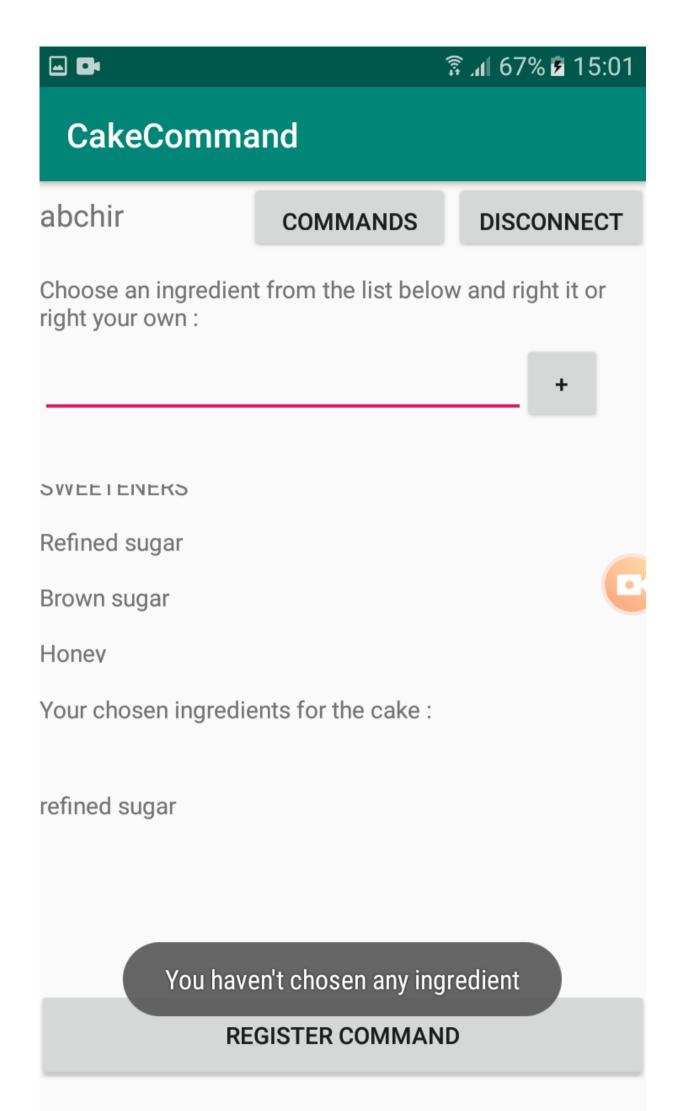
These information don't match any account!

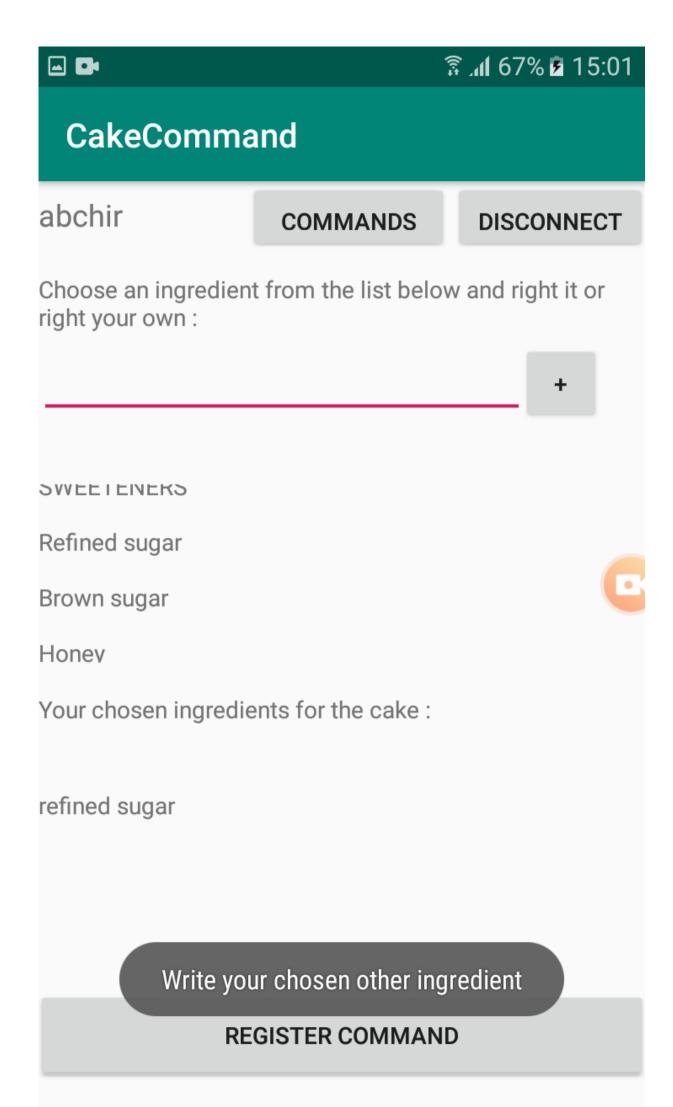
3. Choisir les ingrédients proposés par l'application ou les personnaliser
On montre de même les cas gérés pour la gestion des bugs de saisie :

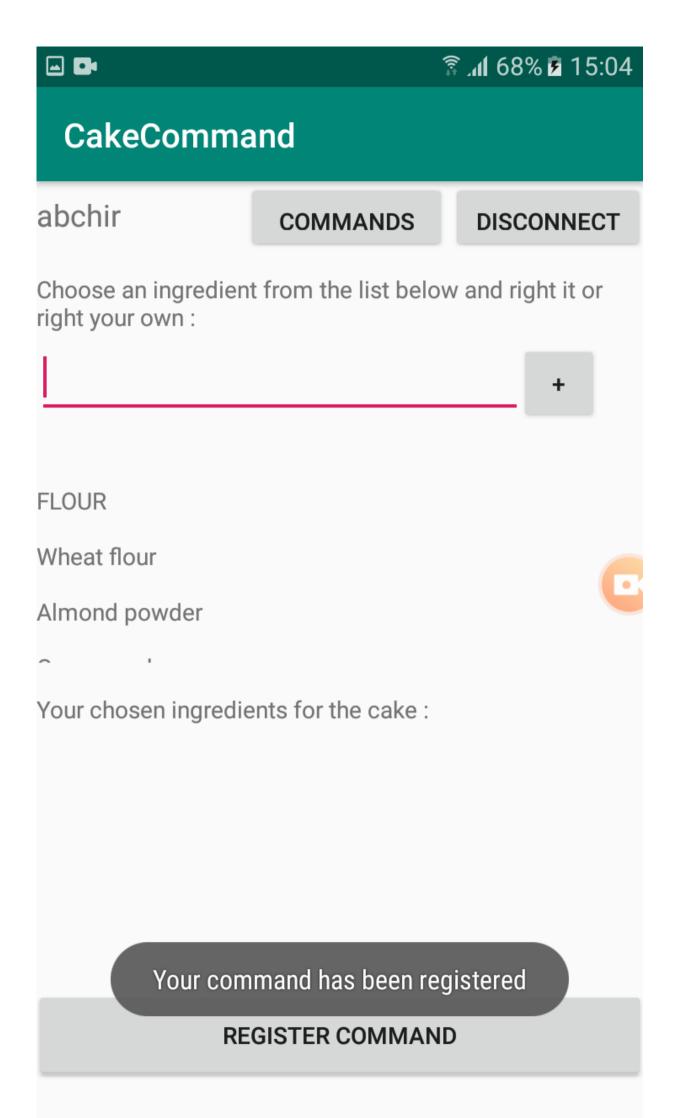


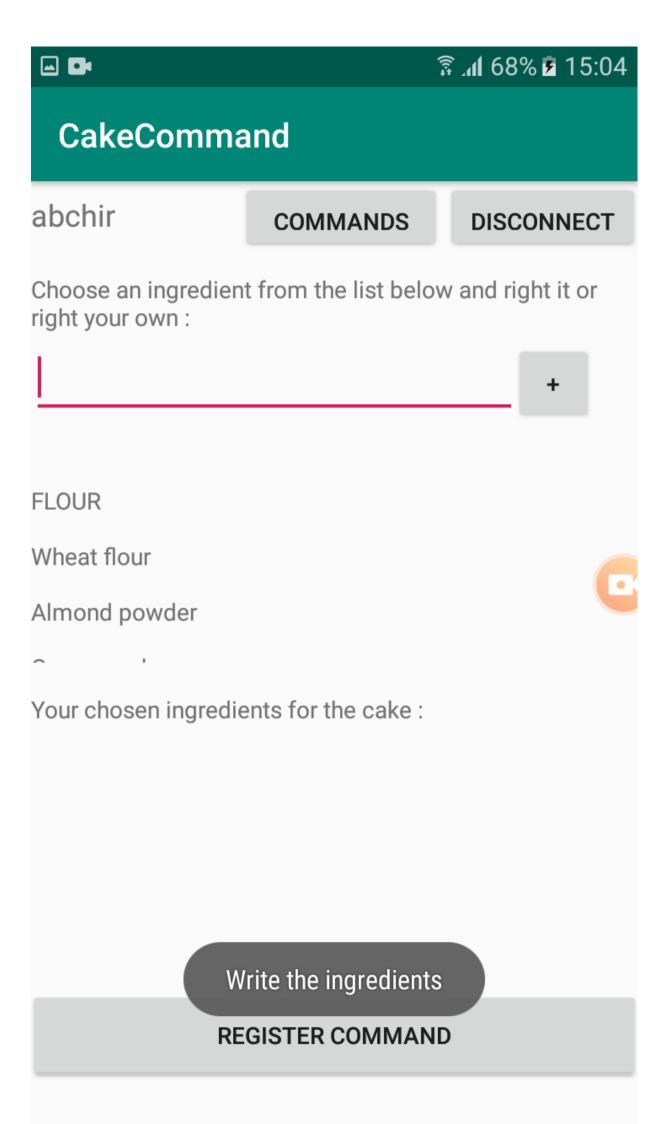








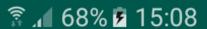




4. Consultation des commandes enregistrées et passées (confirmées pour la livraison) après leur enregistrement et confirmation en temps réel

On montre la facilité fournie par l'interface de l'application permettant d'y circuler aisément et intuitivement :





abchir

DISCONNECT

Your registered commands:

-> Command 4 by abchir with ingredients:

almond powder

brown sugar

corn syrup

Choose a command number(id) to confirm command:



5

CONFIRM COMMAND

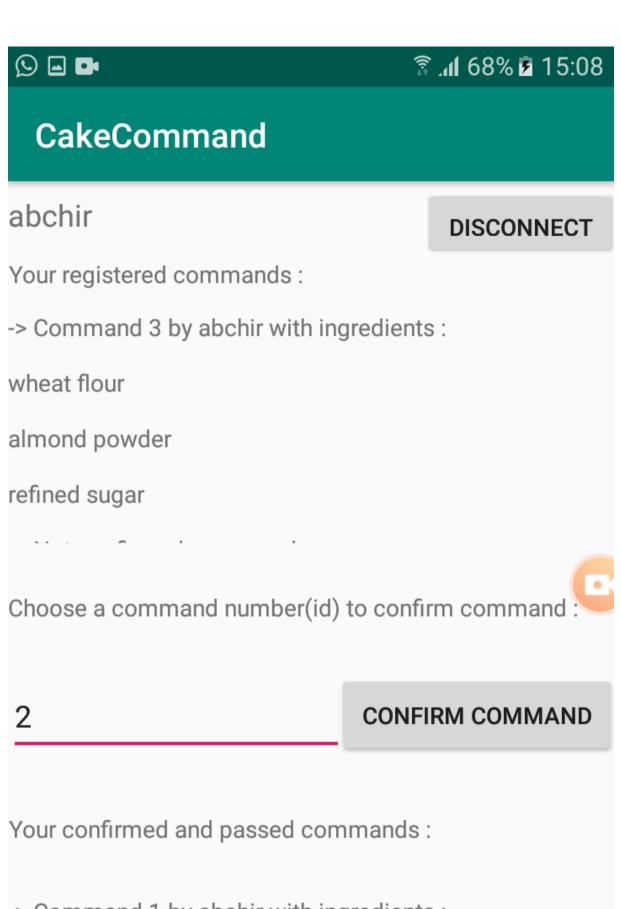
Your confirmed and passed commands:

-> Command 1 by abchir with ingredients:

wheat flour

almond powder

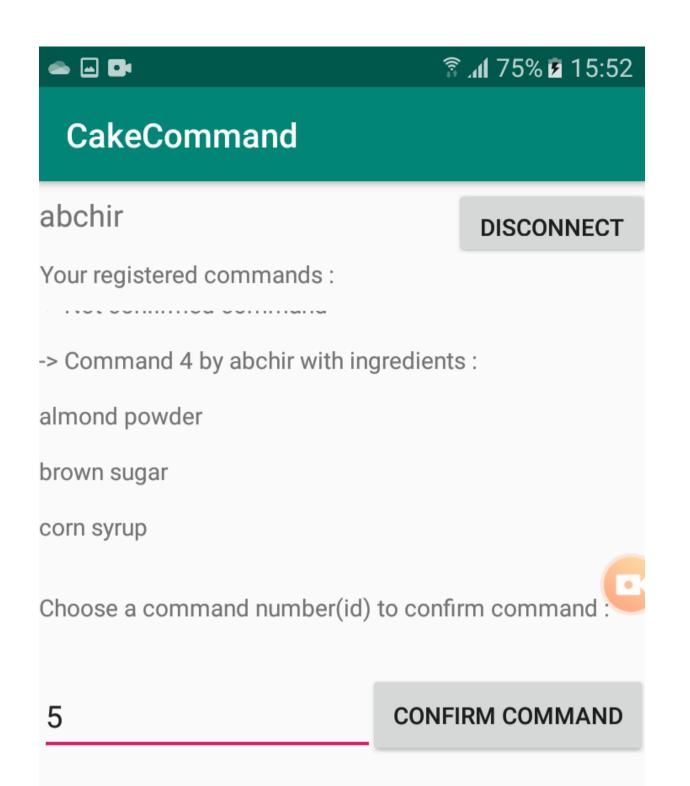
brown sugar



-> Command 1 by abchir with ingredients:

wheat four Your command has been confirmed almond powder

brown sugar



Your confirmed and passed commands:

wheat flour

the id doesen't match any command

nuts butter

-> Confirmed command

Activity_main

```
xmlns:android="http://schemas.android.com/apk/res/android"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
    <TextView
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout constraintVertical bias="0.314" />
```

```
<EditText
    app:layout constraintBottom toBottomOf="parent"
    app:layout constraintStart toStartOf="parent"
    app:layout constraintTop toTopOf="parent"
<EditText
```

Activity second

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

<TextView
    android:id="@+id/txtPassword"
    android:layout_width="wrap_content"
    android:text="@string/txtPassword"
    android:text="@string/txtPassword"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    app:layout_constraintBottom_toBottomof="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintBottom_toBottomof="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintStortom_toSparent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintVertical_bias="0.193" />

<TextView
    android:id="@+id/txtNom"
    android:layout_width="wrap_content"</pre>
```

```
android:layout height="wrap content"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    app:layout constraintEnd toEndOf="parent"
    android:text="@string/btnSignup"
<EditText
    android:inputType="textPersonName"
<TextView
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
```

```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.514" />

<EditText
    android:id="@+id/strCheckPassword"
    android:layout_width="179dp"
    android:layout_height="40dp"
    android:imputType="textPassword"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.35" />

<EditText
    android:id="@+id/strEmail"
    android:layout_height="55dp"
    android:layout_height="55dp"
    android:inputType="textEmailAddress"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintIndor_toTopOf="parent"
    app:layout_constraintIndor_toTopOf="parent"
```

Activity_third

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

<TextView
    android:id="@+id/txtUser"
    android:layout_width="80dp"
    android:layout_width="80dp"
    android:layout_height="31dp"
    android:text="@string/txtUser"
    android:text="@string/txtUser"
    android:text="@string/txtUser"
    android:text="@string/txtUser"
    android:constraintBod tom_toBottomOf="parent"
    app:layout_constraintBod toEndOf="parent"
    app:layout_constraintBod toEndOf="parent"
    app:layout_constraintStatt_toStartOf="parent"
    app:layout_constraintStatt_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.014" />

<EditText
    android:layout_width="289dp"
    android:layout_beight="49dp"
    android:inputType="textPersonName"
    app:layout_constraintBottom toBottomOf="parent"
    app:layout_const
```

```
android:layout width="44dp"
<ScrollView
    <LinearLayout
        <TextView
<TextView
<ScrollView
```

```
</ScrollView>
       android:text="@string/btnRegister"
   <TextView
</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity fourth

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

    <ScrollView
        android:id="@+id/scrollView4"
        android:layout_width="360dp"
        android:layout_height="138dp"</pre>
```

```
app:layout constraintBottom toBottomOf="parent"
        android:orientation="vertical">
    </LinearLayout>
</ScrollView>
<ScrollView
</ScrollView>
<TextView
<TextView
```

```
app:layout constraintBottom toBottomOf="parent"
  <EditText
      android:inputType="number"
      app:layout constraintEnd toEndOf="parent"
  <TextView
  <TextView
      android:textAppearance="@style/TextAppearance.AppCompat.Medium"
/androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.cakecommand;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
```

```
mport android.widget.EditText;
   EditText strNom;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
            public void onClick(View view) {
ainActivity.this)) {
        super.onActivityResult(requestCode, resultCode, data);
```

ActivitySecond.java

```
package com.example.cakecommand;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class ActivitySecond extends AppCompatActivity {
    EditText strNom;
```

```
EditText strPassword;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        strEmail=findViewById(R.id.strEmail);
        btnSignUp.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
ActivitySecond.this)) {
```

ActivityThird.java

```
package com.example.cakecommand;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;
import android.recyclerview.widget.RecyclerView;

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

import org.w3c.dom.Text;

public class ActivityThird extends AppCompatActivity{

    TextView txtUser;
    TextView txtIngredientsChosen;
    TextView txtIngredients;
    Button btnPlus;
    Button btnPlus;
    Button btnRegister;
```

```
protected void onCreate(Bundle savedInstanceState) {
            startActivityForResult(intent, 1634);
```

ActivityFourth.java

```
import androidx.appcompat.app.AppCompatActivity;
   protected void onCreate(Bundle savedInstanceState) {
```

```
for(CommandData str:list)txtConfirmedComm.append(str+"-> Confirmed command");
            public void onClick(View view) {
            public void onClick(View view) {
                        txtConfirmedComm.append(str + "-> Confirmed command");
", Toast. LENGTH LONG) . show();
```

DatabaseManager.java

```
package com.example.cakecommand;
import java.util.List;
public class DatabaseManager extends SQLiteOpenHelper {
   public DatabaseManager(Context context) {
   public void onCreate(SQLiteDatabase sqLiteDatabase) {
```

```
sqLiteDatabase.execSQL(stringSQL);
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
   Cursor cursor =this.getReadableDatabase().rawQuery(stringSQL, null);
   cursor.moveToFirst();
   while(! cursor.isAfterLast()){
   cursor.close();
   public boolean AuthUsers(String nameAuth, String passwordAuth, Context context) {
   Cursor cursor =this.getReadableDatabase().rawQuery(strSQL, null);
   cursor.moveToFirst();
```

```
cursor.close();
public boolean insertCommands (String user, String ingredients, int isConfirmed, Context
    Cursor cursor =this.getReadableDatabase().rawQuery(stringSQL, null);
    cursor.moveToFirst();
    while(! cursor.isAfterLast()){
    this.getWritableDatabase().execSQL(strSQL);
    this.getWritableDatabase().execSQL(stringSQL);
    Cursor cursor =this.getReadableDatabase().rawQuery(strSQL, null);
    cursor.moveToFirst();
        if(id==cursor.getInt(0)&&cursor.getString(1).equals(user)) {
public List<CommandData> getResigteredCommands(String user) {
    cursor.moveToFirst();
    while(! cursor.isAfterLast()){
```

```
String strSQL="select * from T Commands";
Cursor cursor =this.getReadableDatabase().rawQuery(strSQL, null);
cursor.close();
```

CommandData.java

```
IsConfirmed = isConfirmed;
public int getId() {
```

```
public void setIngredients(String ingredients) {
    this.ingredients = ingredients;
}

public int getIsConfirmed() {
    return IsConfirmed;
}

public void setIsConfirmed(int isConfirmed) {
    IsConfirmed = isConfirmed;
}

@Override
public String toString() {
    return "-> Command "+id+" by "+user+" with ingredients : \n\n"+ingredients;
}
}
```

JSSEProvider.java

ByteArrayDataSource.java

```
package com.example.cakecommand;
import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import javax.sql.DataSource;

public class ByteArrayDataSource implements DataSource {
    private byte[] data;
    private String type;

    public ByteArrayDataSource(byte[] data, String type) {
        super();
        this.data = data;
        this.type = type;
    }

    public ByteArrayDataSource(byte[] data) {
```

```
super();
    this.data = data;
}

public void setType(String type) {
    this.type = type;
}

public String getContentType() {
    if (type == null)
        return "application/octet-stream";
    else
        return type;
}

public InputStream getInputStream() throws IOException {
    return new ByteArrayInputStream(data);
}

public String getName() {
    return "ByteArrayDataSource";
}

public OutputStream getOutputStream() throws IOException {
    throw new IOException("Not Supported");
}
```

MailSender.java

Code d'invocation

CONCLUSION:

Tout d'abord, je pense que la demande de réalisation d'un mini projet Android est une très bonne chose car cela ne nous permet pas seulement de pratiquer et tester nos atouts du développement Android, mais également de créer une application complète répondant à des besoins réels à part les petits exercices traités. Partant avec cet état d'esprit, ce projet m'a permis d'acquérir des connaissances intéressantes qui dépassent celles apprises dans les séances des cours.

Je tiens de préciser, que la gestion du temps est un critère déterminant, donc j'ai essayé de mieux organiser mon temps en fonction des fonctionnalités demandées et se focaliser en contenu plus que le côté esthétique.

Je peux donc affirmer que ce projet m'a beaucoup apporté et a été en fin

de compte un travail agréable car le résultat est concret et très satisfaisant pour moi.