



ECOLE HASSANIA  
DES TRAVAUX PUBLICS

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 :**



# CakeCommand

Name :

Password :

Check Password :



Email :

SIGN UP



# CakeCommand

Name :

Password :

Check Password :



Email :

SIGN UP

Write your information



# CakeCommand

Name : abchir

Password : \_\_\_\_\_

Check Password : \_\_\_\_\_



Email : \_\_\_\_\_

SIGN UP

An account already created with the same  
information



# CakeCommand

Name : ouijdane

Password : \_\_\_\_\_

Check Password : \_\_\_\_\_



Email : \_\_\_\_\_

SIGN UP

Complete your information



# CakeCommand

Name : ouijdane

Password : \_\_\_\_\_

Check Password : \_\_\_\_\_



Email : ouijdane.abchir15@outlook.fr

SIGN UP

An account already created with the same  
information



# CakeCommand

Name : ouijdane

Password : .....

Check Password :



Email : ouijdane@gmail.com

SIGN UP

The Checked Password doesn't match the Password





# CakeCommand

Name : ouijdane

Password : .....

Check Password : ..|



Email : ouijdane@gmail.com

SIGN UP

The Checked Password doesn't match the Password



67% 14:57

# CakeCommand

Name :

Password :

SIGN IN



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

SIGN UP

Your account has been created

## 2. Authentification

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



# CakeCommand

Name : \_\_\_\_\_

Password : \_\_\_\_\_

SIGN IN



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

SIGN UP

Write your name and password



# CakeCommand

Name : abchir

Password : \_\_\_\_\_

SIGN IN



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

SIGN UP

Write your password



# CakeCommand

Name : \_\_\_\_\_

Password : .....  
\_\_\_\_\_

SIGN IN



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

SIGN UP

Write your name



# CakeCommand

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 :**





# CakeCommand

abchir

COMMANDS

DISCONNECT

Choose an ingredient from the list below and right it or right your own :

FLOUR

Wheat flour

Almond powder

^ .

Your chosen ingredients for the cake :

REGISTER COMMAND



# CakeCommand

abchir

COMMANDS

DISCONNECT

Choose an ingredient from the list below and right it or right your own :

refined sugar

+

SWEETENERS

Refined sugar

Brown sugar

Honey

Your chosen ingredients for the cake :

REGISTER COMMAND



# CakeCommand

abchir

COMMANDS

DISCONNECT

Choose an ingredient from the list below and right it or right your own :



## SWEETENERS

Refined sugar

Brown sugar

Honey

Your chosen ingredients for the cake :

refined sugar

REGISTER COMMAND



# CakeCommand

abchir

COMMANDS

DISCONNECT

Choose an ingredient from the list below and right it or right your own :



## SWEETENERS

Refined sugar

Brown sugar

Honey

Your chosen ingredients for the cake :

refined sugar

You haven't chosen any ingredient

REGISTER COMMAND



# CakeCommand

abchir

COMMANDS

DISCONNECT

Choose an ingredient from the list below and right it or right your own :



## SWEETENERS

Refined sugar

Brown sugar

Honey

Your chosen ingredients for the cake :

refined sugar

Write your chosen other ingredient

REGISTER COMMAND



# CakeCommand

abchir

COMMANDS

DISCONNECT

Choose an ingredient from the list below and right it or right your own :

+

FLOUR

Wheat flour

Almond powder

^ .

Your chosen ingredients for the cake :

Your command has been registered

REGISTER COMMAND



# CakeCommand

abchir

COMMANDS

DISCONNECT

Choose an ingredient from the list below and right it or  
right your own :

+

FLOUR

Wheat flour

Almond powder

^ .

Your chosen ingredients for the cake :

Write the ingredients

REGISTER COMMAND

#### 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 :**





# CakeCommand

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



68% 15:08

# CakeCommand

abchir

DISCONNECT

Your registered commands :

-> Command 3 by abchir with ingredients :

wheat flour

almond powder

refined sugar

.....

Choose a command number(id) to confirm command :



2

CONFIRM COMMAND

Your confirmed and passed commands :

-> Command 1 by abchir with ingredients :

wheat flour

almond powder

brown sugar

Your command has been confirmed



75% 15:52

# CakeCommand

abchir

DISCONNECT

Your registered commands :

1 -> Command 1

-> 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 :

1 -> Command 1

wheat flour

refined sugar

nuts butter

-> Confirmed command

the id doesn't match any command

## Activity main

```
<?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=".MainActivity">

    <TextView
        android:id="@+id/txt"
        android:layout_width="538dp"
        android:layout_height="38dp"
        android:text="@string/txt"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.504"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/txtPassword"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.018"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.193" />

    <TextView
        android:id="@+id/txtNom"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/txtNom"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.016"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.035" />

    <Button
        android:id="@+id/btnSignin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/btnSignin"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.314" />
```

```

<Button
    android:id="@+id/btnSignup"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/btnSignup"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.69" />

<EditText
    android:id="@+id/strNom"
    android:layout_width="282dp"
    android:layout_height="42dp"
    android:layout_marginStart="112dp"
    android:layout_marginLeft="112dp"
    android:layout_marginBottom="284dp"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.054" />

<EditText
    android:id="@+id/strPassword"
    android:layout_width="245dp"
    android:layout_height="49dp"
    android:ems="10"
    android:inputType="textPassword"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.175" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

## 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:layout_height="wrap_content"
        android:text="@string/txtPassword"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.018"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.193" />

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

```

```
android:layout_height="wrap_content"
android:text="@string/txtNom"
android:textAppearance="@style/TextAppearance.AppCompat.Large"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.016"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.035" />
```

#### <Button

```
android:id="@+id/btnSignup"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/btnSignup"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.69" />
```

#### <EditText

```
android:id="@+id/strNom"
android:layout_width="282dp"
android:layout_height="42dp"
android:layout_marginStart="112dp"
android:layout_marginLeft="112dp"
android:layout_marginBottom="284dp"
android:ems="10"
android:inputType="textPersonName"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.059" />
```

#### <EditText

```
android:id="@+id/strPassword"
android:layout_width="245dp"
android:layout_height="49dp"
android:ems="10"
android:inputType="textPassword"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.175" />
```

#### <TextView

```
android:id="@+id/txtCheckPassword"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/txtCheckPassword"
android:textAppearance="@style/TextAppearance.AppCompat.Large"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.021"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.345" />
```

#### <TextView

```
android:id="@+id/txtEmail"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/txtEmail"
android:textAppearance="@style/TextAppearance.AppCompat.Large"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.014"
```

```

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:ems="10"
    android:inputType="textPassword"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.35" />

<EditText
    android:id="@+id/strEmail"
    android:layout_width="248dp"
    android:layout_height="55dp"
    android:ems="10"
    android:inputType="textEmailAddress"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.499" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

## 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_height="31dp"
        android:text="@string/txtUser"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.014" />

    <EditText
        android:id="@+id/strIngredient"
        android:layout_width="289dp"
        android:layout_height="49dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.181" />

```

```

<Button
    android:id="@+id/btnPlus"
    android:layout_width="44dp"
    android:layout_height="41dp"
    android:text="@string/plus"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.911"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.192" />

<ScrollView
    android:id="@+id/scrollView2"
    android:layout_width="454dp"
    android:layout_height="106dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.397">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:id="@+id/txtIngredients"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

    </LinearLayout>
</ScrollView>

<Button
    android:id="@+id/btnDisconnect"
    android:layout_width="112dp"
    android:layout_height="35dp"
    android:text="@string/disconnect"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.997"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.008" />

<TextView
    android:id="@+id/txtChoose"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/txtChoose"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="@+id/scrollView2"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.104" />

<ScrollView
    android:id="@+id/scrollView3"
    android:layout_width="448dp"
    android:layout_height="104dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.792">

```



```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:id="@+id/txtIngredientsChosen"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
</ScrollView>

<Button
    android:id="@+id/btnRegister"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/btnRegister"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.443"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.943" />

<TextView
    android:id="@+id/txtYourIngredients"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/txtYourIngredients"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.562" />

<Button
    android:id="@+id/btnCommand"
    android:layout_width="116dp"
    android:layout_height="35dp"
    android:text="@string/btnCommand"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.516"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.008" />

</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"

```

```

app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.182">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:id="@+id/txtRegisteredComm"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    </LinearLayout>
</ScrollView>

<ScrollView
    android:id="@+id/scrollView5"
    android:layout_width="408dp"
    android:layout_height="136dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:id="@+id/txtConfirmedComm"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
        </LinearLayout>
    </ScrollView>

<TextView
    android:id="@+id/txtRegisteredCommands"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/txtRegisteredCommands"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.002"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.084" />

<TextView
    android:id="@+id/txtConfirmedCommands"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/txtConfirmedCommands"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.005"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.693" />

<Button
    android:id="@+id/btnConfirm"
    android:layout_width="163dp"
    android:layout_height="43dp"
    android:text="@string/btnConfirm"

```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="1.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.572" />
```

#### <EditText

```
android:id="@+id/strNumCommand"
android:layout_width="200dp"
android:layout_height="48dp"
android:ems="10"
android:inputType="number"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.574" />
```

#### <TextView

```
android:id="@+id/txtChooseCommand"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/txtChooseCommand"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.45" />
```

#### <TextView

```
android:id="@+id/txtnoun"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="bla"
android:textAppearance="@style/TextAppearance.AppCompat.Medium"
app:layout_constraintBottom_toBottomOf="@+id/scrollView5"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.014" />
```

#### <Button

```
android:id="@+id/btnDis"
android:layout_width="113dp"
android:layout_height="35dp"
android:layout_marginStart="248dp"
android:layout_marginLeft="248dp"
android:text="@string/disconnect"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.014" />
```

```
</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;
```

```

import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    EditText strNom;
    EditText strPassword;
    Button btnSignin;
    Button btnSignUp;
    DatabaseManager databaseManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        strNom = findViewById(R.id.strNom);
        strPassword = findViewById(R.id.strPassword);
        btnSignin = findViewById(R.id.btnSignin);
        btnSignUp = findViewById(R.id.btnSignup);
        btnSignin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                databaseManager=new DatabaseManager(MainActivity.this);

                if(databaseManager.AuthUsers(strNom.getText().toString(),strPassword.getText().toString(),MainActivity.this)) {
                    Intent intent = new Intent(MainActivity.this, ActivityThird.class);
                    intent.putExtra("message",strNom.getText().toString());
                    startActivityForResult(intent, 1134);
                }
            }
        });
        btnSignUp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MainActivity.this, ActivitySecond.class);
                startActivityForResult(intent, 1234);
            }
        });
    }
    @Override
    protected void onActivityResult(int requestCode,int resultCode,Intent data){
        super.onActivityResult(requestCode,resultCode,data);
        if(requestCode==1234){}
    }
}

```

## 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;
EditText strCheckPassword;
EditText strEmail;
Button btnSignUp;
DatabaseManager databaseManager;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_second);
    strNom = findViewById(R.id.strNom);
    strPassword = findViewById(R.id.strPassword);
    strCheckPassword=findViewById(R.id.strCheckPassword);
    strEmail=findViewById(R.id.strEmail);
    btnSignUp = findViewById(R.id.btnSignup);
    btnSignUp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view){
            databaseManager=new DatabaseManager(ActivitySecond.this);

            if(!strPassword.getText().toString().equals(strCheckPassword.getText().toString()))
                Toast.makeText(ActivitySecond.this, "The Checked Password doesn't match
the Password ", Toast.LENGTH_LONG).show();

            if(databaseManager.insertUsers(strNom.getText().toString(),strPassword.getText().toString()
, strCheckPassword.getText().toString(), strEmail.getText().toString(),
ActivitySecond.this)) {
                Intent intent = new Intent(ActivitySecond.this, MainActivity.class);
                intent.putExtra("message",strNom.getText().toString());
                startActivityForResult(intent, 1034);
            }
        }
    });
}
@Override
protected void onActivityResult(int requestCode,int resultCode,Intent data){
    super.onActivityResult(requestCode,resultCode,data);
    if(requestCode==1234){}
}
}

```

## ActivityThird.java

```

package com.example.cakecommand;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
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;
    EditText strIngredient;
    Button btnPlus;
    Button btnDisconnect;
    Button btnRegister;
}

```

```

Button btnCommands;
DatabaseManager databaseManager;

@Override
protected void onCreate(Bundle savedInstanceState) {
    final int i=1;
    String appIngredients ="FLOUR"+"\\n\\n"+"Wheat flour"+"\\n\\n"+"Almond
powder"+"\\n\\n"+"Corns crash"+"\\n\\n"+"Mixte nuts powder"+"\\n\\n"
        +"Other"+"\\n\\n"+"none"+"\\n\\n\\n"+"SWEETENERS"+"\\n\\n"+"Refined
sugar"+"\\n\\n"+"Brown sugar"+"\\n\\n"+"Honey"+"\\n\\n"+"Corn syrup"+"\\n\\n"
        +"Other"+"\\n\\n"+"none"+"\\n\\n\\n"+"FATS"+"\\n\\n"+"Regular butter"+"\\n\\n"+"Nuts
butter"+"\\n\\n"+"Coconut oil"+"\\n\\n"+"Other"+"\\n\\n"+"none"
        +"\\n\\n\\n"+"LIQUIDS"+"\\n\\n"+"Regular milk"+"\\n\\n"+"Nuts
milk"+"\\n\\n"+"Regular Cream"+"\\n\\n"+"Puts Cream"+"\\n\\n"+"Other"+"\\n\\n"+"none"
        +"\\n\\n\\n"+"STUFFING"+"\\n\\n"+"Berries"+"\\n\\n"+"Choclate"+"\\n\\n"+"Cream
cheese"+"\\n\\n"+"Other"+"\\n\\n"+"none"
        +"\\n\\n\\n"+"OTHERS"+"\\n\\n"+"Eggs"+"\\n\\n"+"Baking
powder"+"\\n\\n"+"Flavoring"+"\\n\\n"+"Other";
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_third);
    final String User=this.getIntent().getExtras().getString("message");
    txtUser=findViewById(R.id.txtUser);
    txtIngredients=findViewById(R.id.txtIngredients);
    txtIngredientsChosen=findViewById(R.id.txtIngredientsChosen);
    strIngredient=findViewById(R.id.strIngredient);
    btnPlus=findViewById(R.id.btnPlus);
    btnDisconnect=findViewById(R.id.btnDisconnect);
    btnRegister=findViewById(R.id.btnRegister);
    btnCommands=findViewById(R.id.btnCommand);
    txtUser.setText(User);
    txtIngredients.append(appIngredients);
    btnCommands.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(ActivityThird.this, ActivityFourth.class);
            intent.putExtra("message",User);
            startActivityForResult(intent, 1934);
        }
    });
    btnDisconnect.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(ActivityThird.this, MainActivity.class);
            startActivityForResult(intent, 1634);
        }
    });
    btnRegister.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            databaseManager=new DatabaseManager(ActivityThird.this);
            databaseManager.insertCommands(User,txtIngredientsChosen.getText().toString(),0,ActivityThi
rd.this);
            txtIngredientsChosen.setText("");
        }
    });
    btnPlus.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(strIngredient.getText().toString().equals("")) {
                Toast.makeText(ActivityThird.this, "You haven't chosen any ingredient",
Toast.LENGTH_LONG).show();
            }
            if(strIngredient.getText().toString().equals("other")) {
                Toast.makeText(ActivityThird.this, "Write your chosen other
ingredient", Toast.LENGTH_LONG).show();
            }
            else if(strIngredient.getText().toString().equals("flavoring")) {
                Toast.makeText(ActivityThird.this, "Writet your chosen flavoring",
Toast.LENGTH_LONG).show();
            }
        }
    });
}

```

```

        }
        else if(strIngredient.getText().toString().equals("none")) {
            Toast.makeText(ActivityThird.this, "You haven't chosen any ingredient", Toast.LENGTH_LONG).show();
        }
        else
            if(strIngredient.getText().toString().equals("flour") || strIngredient.getText().toString().equals("sweeteners") || strIngredient.getText().toString().equals("fats") ||
                strIngredient.getText().toString().equals("liquids") || strIngredient.getText().toString().equals("stuffing") || strIngredient.getText().toString().equals("others")) {
                Toast.makeText(ActivityThird.this, "Choose and write an ingredient not a category name", Toast.LENGTH_LONG).show();
            }
            else if(i==1)
            {txtIngredientsChosen.append(strIngredient.getText().toString()+"\n\n");
              strIngredient.setText("");
            }
        });
    }
}
}

```

## ActivityFourth.java

```

package com.example.cakecommand;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;

public class ActivityFourth extends AppCompatActivity {

    String User;

    TextView txtRegisteredComm;
    TextView txtConfirmedComm;
    EditText strNumCommand;
    TextView txtnoun;
    Button btnConfirm;
    Button btnDis;
    DatabaseManager databaseManager;
    List<CommandData> list;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        databaseManager=new DatabaseManager(ActivityFourth.this);
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fourth);
        User=this getIntent().getExtras().getString("message");
        txtRegisteredComm=findViewById(R.id.txtRegisteredComm);
        txtConfirmedComm=findViewById(R.id.txtConfirmedComm);
        strNumCommand=findViewById(R.id.strNumCommand);
        btnConfirm=findViewById(R.id.btnConfirm);
        txtnoun=findViewById(R.id.txtnoun);
        btnDis=findViewById(R.id.btnDis);
        txtnoun.setText(User);
        list=databaseManager.getResigteredCommands(User);
    }
}

```

```

        for (CommandData str: list) txtRegisteredComm.append(str+"-> Not confirmed command"
        +"\\n\\n");
        list=databaseManager.getConfirmedCommands(User);
        for (CommandData str: list) txtConfirmedComm.append(str+"-> Confirmed command");
        btnDis.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(ActivityFourth.this, MainActivity.class);
                startActivityForResult(intent, 1134);
            }
        });
        btnConfirm.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                int i=Integer.parseInt(strNumCommand.getText().toString());
                Log.i("DATABASE", "tbedel invoked"+i);
                if (databaseManager.idExist(i, User)) {
                    list = databaseManager.getIngredientsAndMakeConfirm(i, User);
                    for (CommandData str : list)
                        txtConfirmedComm.append(str + "-> Confirmed command");
                    Toast.makeText(ActivityFourth.this, "Your command has been confirmed",
                    Toast.LENGTH_LONG).show();
                }
                else Toast.makeText(ActivityFourth.this, "the id doesn't match any command",
                Toast.LENGTH_LONG).show();
                list=databaseManager.getResigteredCommands(User);
                txtRegisteredComm.setText("");
                for (CommandData str: list) txtRegisteredComm.append(str+"-> Not confirmed
                command"+"\\n\\n");
            }
        });
    }
}
}

```

## DatabaseManager.java

```

package com.example.cakecommand;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;

public class DatabaseManager extends SQLiteOpenHelper {
    private static final String DATABASE_NAME="CakeCommand.DB";
    private static final int DATABASE_VERSION=4;
    public DatabaseManager(Context context){
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        String stringSQL="create table T_Commands ("
            +"id integer primary key autoincrement,"
            +"user text not null,"
            +"ingredients varchar(100"+" ) not null,"
            +"isConfirmed integer not null)";
        String strSQL = "create table T_Users ("
            +"name text primary key,"
            +"password text not null,"
            +"email varchar(20"+" ) not null)";
        sqLiteDatabase.execSQL(strSQL);
    }
}

```



```

        SQLiteDatabase.execSQL(stringSQL);
        Log.i("DATABASE","onCreate invoked");
    }
    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
        String strSQL="drop table T_Users";
        String stringSQL="drop table T_Commands";
        sqLiteDatabase.execSQL(strSQL);
        //sqLiteDatabase.execSQL(stringSQL);
        this.onCreate(sqLiteDatabase);
        Log.i("DATABASE","onUpgrade invoked");
    }
    public boolean insertUsers(String name, String password, String CheckPassword, String
email,Context context){
        String stringSQL="select * from T_Users";
        Cursor cursor =this.getReadableDatabase().rawQuery(stringSQL, null);
        cursor.moveToFirst();
        while(! cursor.isAfterLast()){
            if(name.equals(cursor.getString(0))||email.equals(cursor.getString(2))) {
                cursor.close();
                Toast.makeText(context, "An account already created with the same
information", Toast.LENGTH_LONG).show();
                return false;
            }
            cursor.moveToNext();
        }
        cursor.close();
        if(email.equals("")&&name.equals("")&&password.equals("")){
            Toast.makeText(context, "Write your information", Toast.LENGTH_LONG).show();
            return false;
        }
        if(email.equals("")||name.equals("")||password.equals("")){
            Toast.makeText(context, "Complete your information", Toast.LENGTH_LONG).show();
            return false;
        }
        if(!CheckPassword.equals(password)){
            Toast.makeText(context, "The Checked Password doesn't match the Password",
Toast.LENGTH_LONG).show();
            return false;
        }
        name=name.replace("'", "");
        password=password.replace("'", "");
        CheckPassword=CheckPassword.replace("'", "");
        String strSQL="insert into T_Users (name , password, email) values('"
            +name+"', '"+password+"', '"+email+"')";
        this.getWritableDatabase().execSQL(strSQL);
        Toast.makeText(context, "Your account has been created", Toast.LENGTH_LONG).show();
        Log.i("DATABASE","insertUsers invoked");
        return true;
    }
    public boolean AuthUsers(String nameAuth, String passwordAuth,Context context){
        Log.i("DATABASE","AuthUsers invoked");
        if(nameAuth.equals("")&&passwordAuth.equals("")){
            Toast.makeText(context, "Write your name and password",
Toast.LENGTH_LONG).show();
            return false;
        }
        if(nameAuth.equals("")){
            Toast.makeText(context, "Write your name", Toast.LENGTH_LONG).show();
            return false;
        }
        if(passwordAuth.equals("")){
            Toast.makeText(context, "Write your password", Toast.LENGTH_LONG).show();
            return false;
        }
        String strSQL="select * from T_Users";
        Cursor cursor =this.getReadableDatabase().rawQuery(strSQL, null);
        cursor.moveToFirst();
        while(! cursor.isAfterLast()){
            if(nameAuth.equals(cursor.getString(0))&&passwordAuth.equals(cursor.getString(1))) {

```

```

        cursor.close();
        return true;
    }
    cursor.moveToNext();
}
cursor.close();
Toast.makeText(context, "These information don't match any account !",
Toast.LENGTH_LONG).show();
return false;
}

public boolean insertCommands(String user, String ingredients, int isConfirmed, Context
context){
    if(ingredients.equals("")){
        Toast.makeText(context, "Write the ingredients", Toast.LENGTH_LONG).show();
        return false;
    }
    String stringSQL="select * from T_Commands";
    Cursor cursor =this.getReadableDatabase().rawQuery(stringSQL, null);
    cursor.moveToFirst();
    while(! cursor.isAfterLast()){
        if(user.equals(cursor.getString(1))&&ingredients.equals(cursor.getString(2))) {
            cursor.close();
            Toast.makeText(context, "A command already registered with the same
information", Toast.LENGTH_LONG).show();
            return false;
        }
        cursor.moveToNext();
    }
    cursor.close();
    user=user.replace("'", "");
    ingredients=ingredients.replace("'", "");
    String strSQL="insert into T_Commands (user , ingredients, isConfirmed) values('"
        +user+"', '"+ingredients+"', '"+isConfirmed+"')";
    this.getWritableDatabase().execSQL(strSQL);
    Log.i("DATABASE", "insertCommands invoked");
    Toast.makeText(context, "Your command has been registered",
Toast.LENGTH_LONG).show();
    return true;
}

public List<CommandData> getIngredientsAndMakeConfirm(int id,String user){
    Log.i("DATABASE", "tbedel invoked");
    String strSQL="select * from T_Commands";
    String stringSQL="update T_Commands set isConfirmed=1 where id="+id;
    this.getWritableDatabase().execSQL(stringSQL);
    List<CommandData> ingredients=new ArrayList<>();
    Cursor cursor =this.getReadableDatabase().rawQuery(strSQL, null);
    cursor.moveToFirst();
    while(! cursor.isAfterLast()){
        if(id==cursor.getInt(0)&&cursor.getString(1).equals(user)) {
            Log.i("DATABASE", "tbedel invoked");
            CommandData ing=new CommandData(cursor.getInt(0), cursor.getString(1),
cursor.getString(2), cursor.getInt(3));
            ingredients.add(ing) ;
        }
        cursor.moveToNext();
    }
    cursor.close();
    return ingredients;
}

public List<CommandData> getResigteredCommands(String user){
    String strSQL="select * from T_Commands";
    List<CommandData> commands= new ArrayList<>();
    Cursor cursor =this.getReadableDatabase().rawQuery(strSQL, null);
    cursor.moveToFirst();
    while(! cursor.isAfterLast()){
        if(cursor.getInt(3)==0&&cursor.getString(1).equals(user)){CommandData score=new
CommandData(cursor.getInt(0), cursor.getString(1), cursor.getString(2), cursor.getInt(3));
        commands.add(score);}
        cursor.moveToNext();
    }
    cursor.close();
}

```

```

        return commands;
    }

    public List<CommandData> getConfirmedCommands(String user) {
        String strSQL="select * from T_Commands";
        List<CommandData> commands= new ArrayList<>();
        Cursor cursor =this.getReadableDatabase().rawQuery(strSQL, null);
        cursor.moveToFirst();
        while(! cursor.isAfterLast()){
            if(cursor.getInt(3)==1&&cursor.getString(1).equals(user)){CommandData score=new
CommandData(cursor.getInt(0),cursor.getString(1), cursor.getString(2),cursor.getInt(3));
                commands.add(score);}
            cursor.moveToNext();
        }
        cursor.close();

        return commands;
    }

    public boolean idExist(int id,String user){
        String strSQL="select * from T_Commands";
        Cursor cursor =this.getReadableDatabase().rawQuery(strSQL, null);
        cursor.moveToFirst();
        while(! cursor.isAfterLast()){
            if(cursor.getInt(0)==id&&cursor.getString(1).equals(user)) return true;
            cursor.moveToNext();
        }
        cursor.close();
        return false;
    }
}

```

## CommandData.java

```

package com.example.cakecommand;

import java.util.Date;

public class CommandData {
    private int id;
    private String user;
    private String ingredients;
    private int IsConfirmed;

    public CommandData(int id, String user, String ingredients, int isConfirmed) {
        this.id = id;
        this.user = user;
        this.ingredients = ingredients;
        IsConfirmed = isConfirmed;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getUser() {
        return user;
    }

    public void setUser(String user) {
        this.user = user;
    }

    public String getIngredients() {
        return ingredients;
    }
}

```

```

    }

    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

```

package com.example.cakecommand;

import java.security.AccessController;
import java.security.Provider;

public final class JSSEProvider extends Provider {

    public JSSEProvider() {
        super("HarmonyJSSE", 1.0, "Harmony JSSE Provider");
        AccessController.doPrivileged(new java.security.PrivilegedAction() {
            public Void run() {
                put("SSLContext.TLS",
"org.apache.harmony.xnet.provider.jsse.SSLContextImpl");
                put("Alg.Alias.SSLContext.TLSv1", "TLS");
                put("KeyManagerFactory.X509",
"org.apache.harmony.xnet.provider.jsse.KeyManagerFactoryImpl");
                put("TrustManagerFactory.X509",
"org.apache.harmony.xnet.provider.jsse.TrustManagerFactoryImpl");
                return null;
            }
        });
    }
}

```

## 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

```

package com.example.cakecommand;

import android.os.Message;
import android.se.omapi.Session;

import java.net.PasswordAuthentication;
import java.security.Security;
import java.util.Properties;

public class MailSender extends javax.mail.Authenticator {
    private String mailhost = "smtp.gmail.com"; //Hostname of the SMTP mail server which
you want to connect for sending emails.
    private String user;
    private String password;
    private Session session;

    static {
        Security.addProvider(new JSSEProvider());
    }

    public MailSender(String user, String password) {
        this.user = user; //Your SMTP username. In case of GMail SMTP this is going to be
your GMail email address.
        this.password = password; //Your SMTP password. In case of GMail SMTP this is going
to be your GMail password.

        Properties props = new Properties();
        props.setProperty("mail.transport.protocol", "smtp");
        props.setProperty("mail.host", mailhost);
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.port", "465");
        props.put("mail.smtp.socketFactory.port", "465");
        props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
        props.put("mail.smtp.socketFactory.fallback", "false");
        props.setProperty("mail.smtp.quitwait", "false");

        session = Session.getDefaultInstance(props, this);
    }
}

```

```

    }

    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication(user, password);
    }

    public synchronized void sendMail(String subject, String body,
                                      String sender, String recipients) throws Exception {
        MimeMessage message = new MimeMessage(session);
        DataHandler handler = new DataHandler(new ByteArrayDataSource(body.getBytes(),
"text/plain"));
        message.setSender(new InternetAddress(sender));
        message.setSubject(subject);
        message.setDataHandler(handler);

        if (recipients.indexOf(',') > 0)
            message.setRecipients(Message.RecipientType.TO,
InternetAddress.parse(recipients));
        else
            message.setRecipient(Message.RecipientType.TO, new
InternetAddress(recipients));

        Transport.send(message);
    }
}

```

## Code d'invocation

```

new Thread(new Runnable() {

@Override
public void run() {
    try {
        GMailSender sender = new MailSender("Your_Gmail_UserName@gmail.com",
"Your_Gmail_password");
        sender.sendMail("This is a test subject", "This is the test body content",
"Your_Gmail_UserName@gmail.com", "recipient@example.com");
    } catch (Exception e) {
        Log.e("SendMail", e.getMessage(), e);
    }
}

}).start();

```

## 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.





