COM4501 Android Operating System for Mobile Devices

YAHYA BATURAY SARAÇOĞLU – 18290121

App Fit

About the application

The Main Idea:

It ıs designed for people who have a life routine that requires a certain level of physical activity in their daily life and who need to regulate their eating habits.

Who should use?

-People who work in a job that requires more physical strength than normal

-Those,

who have a regular sports life.

who want to have a healthy life.

who want to establish a healthy diet.

Some information about the application:

-This application is created using android studio editor, java programming language, object-oriented programming and xml properties.

-Run on virtual device, android operating system

-Android file system

-Internal & External storage

-SQLite database

Explanation of the application with code blocks and screenshots:

MainActivity.java >

Including the tool I need while using the database:

```
Stetho.initializeWithDefaults( context: this);

new OkHttpClient.Builder()
        .addNetworkInterceptor(new StethoInterceptor())
        .build();
```

Creating an object from the DBAdapter class I created for the database.

I get the number of foods that are in the table where the database for foods.

if there is no food in the table yet, the food is being defined with DBSetupInsert class.

```
DBAdapter db = new DBAdapter( ctx: this);
db.open();

int numberRows = db.count( table: "food");

if(numberRows < 1){

    DBSetupInsert setupInsert = new DBSetupInsert( ctx: this);
    setupInsert.insertAllCategories();
    setupInsert.insertAllFood();

}
```

This time, I am checking to see if there is any user in the database similarly.

Runs SignUp screen via intent if no user has been added yet.

If the user has been logged in before, HomeFragment under FragmenActivity is redirected and we don't have to see the login screen every time.

```
numberRows = db.count( table: "users");

db.close();

if(numberRows < 1){

    Intent i = new Intent( packageContext: MainActivity.this, SignUp.class);
    startActivity(i);
}
else{
    Intent i = new Intent( packageContext: MainActivity.this, FragmentActivity.class);
    startActivity(i);
```

DBAdapter.java > The class I created for database adaptation

Creating database table for user:

```
try{

    db.execSQL("CREATE TABLE IF NOT EXISTS users (" +
            " _id INTEGER PRIMARY KEY AUTOINCREMENT, " +
            " user_id INTEGER, " +
            " user_email VARCHAR," +
            " user_password VARCHAR, " +
            " user_salt VARCHAR, " +
            " user_alias VARCHAR," +
            " user_dob DATE, " +
            " user_gender INT, " +
            " user_location VARHCAR, " +
            " user_height INT, " +
            " user_mesurment VARHCAR, " +
            " user_last_seen TIME," +
            " user_note VARCHAR);");

}
```

```
try {
    db.execSQL("CREATE TABLE IF NOT EXISTS food (" +
            " _id INTEGER PRIMARY KEY AUTOINCREMENT, " +
            " food_id INTEGER, " +
            " food_name VARCHAR," +
            " food_manufactor_name VARCHAR," +
            " food_store VARCHAR," +
            " food_description VARCHAR," +
            " food_serving_size_gram DOUBLE," +
            " food_serving_size_gram_mesurment VARCHAR," +
            " food_serving_size_pcs DOUBLE," +
            " food_serving_size_pcs_mesurment VARCHAR," +
            " food_energy DOUBLE," +
            " food_proteins DOUBLE," +
            " food_carbohydrates DOUBLE," +
            " food_fat DOUBLE," +
            " food_energy_calculated DOUBLE," +
            " food_proteins_calculated DOUBLE," +
            " food_carbohydrates_calculated DOUBLE," +
            " food_fat_calculated DOUBLE," +
            " food_user_id INT," +
            " food_barcode VARCHAR," +
            " food_category_id INT," +
            " food_thumb VARCHAR," +
            " food_image_a VARCHAR," +
            " food_image_b VARCHAR," +
            " food_image_c VARCHAR," +
            " food_last_used DATE," +
            " food_language VARCHAR," +
            " food_notes VARCHAR);");

}
```

Creating database table for foods

Creating database table for calorie values for eaten foods:

```
try{
    db.execSQL("CREATE TABLE IF NOT EXISTS food_diary_cal_eaten (" +
            " _id INTEGER PRIMARY KEY AUTOINCREMENT, " +
            " fdce_id INTEGER, " +
            " fdce_date DATE, " +
            " fdce_meal_no INT, " +
            " fdce_eaten_energy INT, " +
            " fdce_eaten_proteins INT, " +
            " fdce_eaten_carbs INT, " +
            " fdce_eaten_fat INT);");
}
catch (SQLException e) {
```

Function to create food categories

```
public void setupInsertToCategories(String values){
    try{
        DBAdapter db = new DBAdapter(context);
        db.open();
        db.insert( table: "categories",
                fields: "_id, category_name, category_parent_id, category_icon, category_note",
                values);
        db.close();
    }
    catch (SQLiteException e){
```

Categorizing food

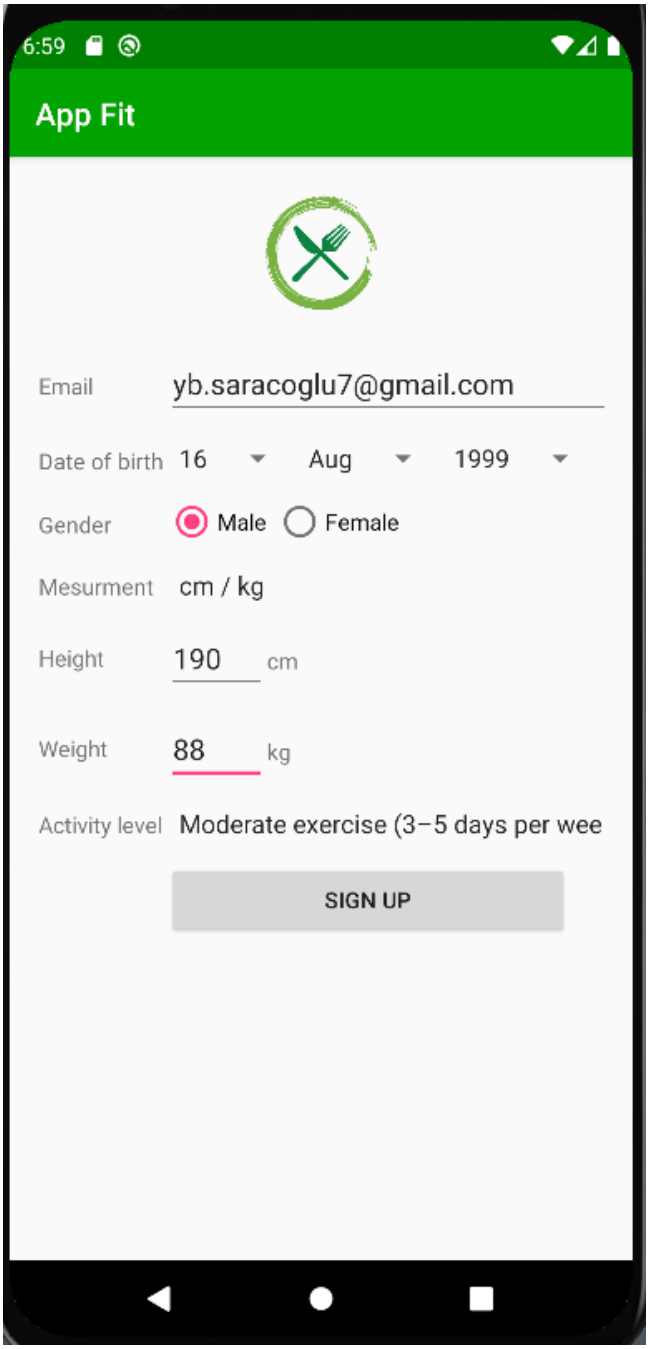
```
public void insertAllCategories(){
    setupInsertToCategories("NULL, 'Bread', '0', '', NULL");
    setupInsertToCategories("NULL, 'Bread', '1', '', NULL");
    setupInsertToCategories("NULL, 'Cereals', '1', '', NULL");
    setupInsertToCategories("NULL, 'Frozen bread and rolls', '1', '', NULL");
    setupInsertToCategories("NULL, 'Crispbread', '1', '', NULL");

    setupInsertToCategories("NULL, 'Dessert and baking', '0', '', NULL");
    setupInsertToCategories("NULL, 'Baking', '6', '', NULL");
    setupInsertToCategories("NULL, 'Biscuit', '6', '', NULL");


    setupInsertToCategories("NULL, 'Drinks', '0', '', NULL");
    setupInsertToCategories("NULL, 'Soda', '9', '', NULL");


    setupInsertToCategories("NULL, 'Fruit and vegetables', '0', ''
    setupInsertToCategories("NULL, 'Frozen fruits and vegetables',
    setupInsertToCategories("NULL, 'Fruit', '11', '', NULL");
    setupInsertToCategories("NULL, 'Vegetables', '11', '', NULL");
    setupInsertToCategories("NULL, 'Canned fruits and vegetables',
```

Function to create food categories:

```
public void setupInsertToFood(String values){

    try {
        DBAdapter db = new DBAdapter(context);
        db.open();
        db.insert( table: "food",
                fields: "_id, food_name, food_manufactor_name, food_serving_size_gram, food_serving_si
                values);
        db.close();
    }
    catch (SQLiteException e){

    }

}
```

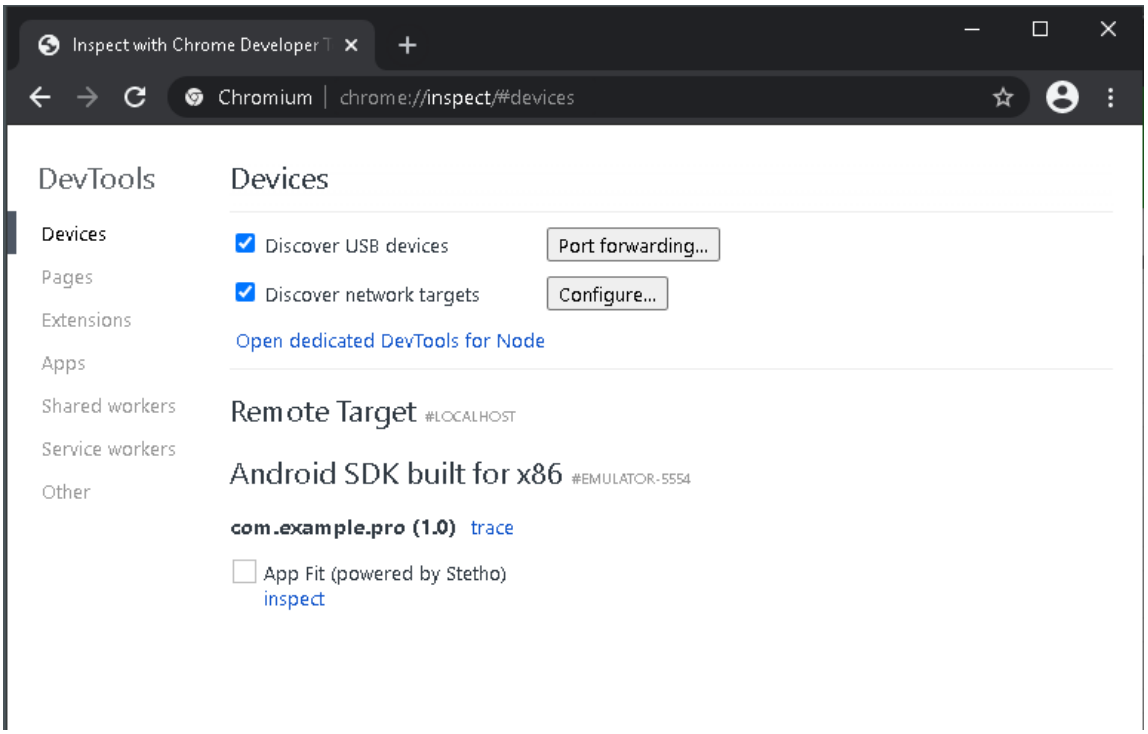
GUI

SignUp Screen:

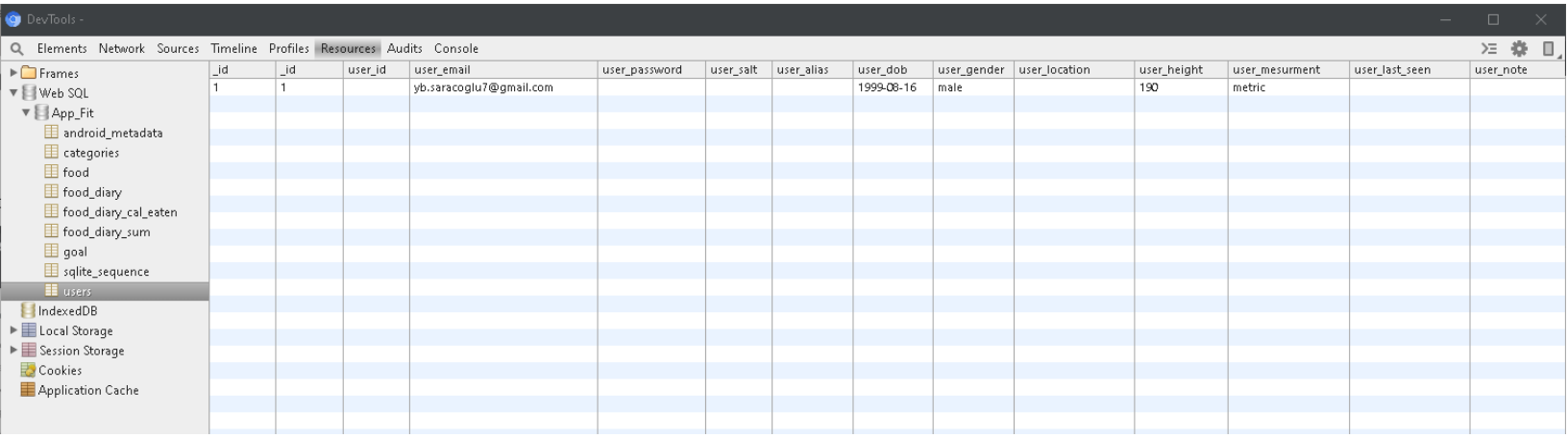


After all the information on this screen is received from the user, it is kept in a database and then used in case of need.

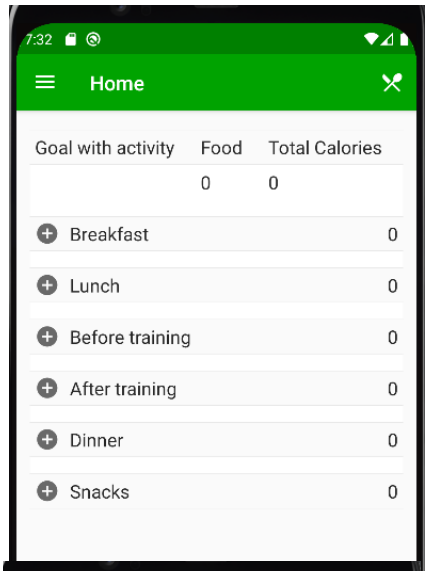
Showing Database with Chromium:



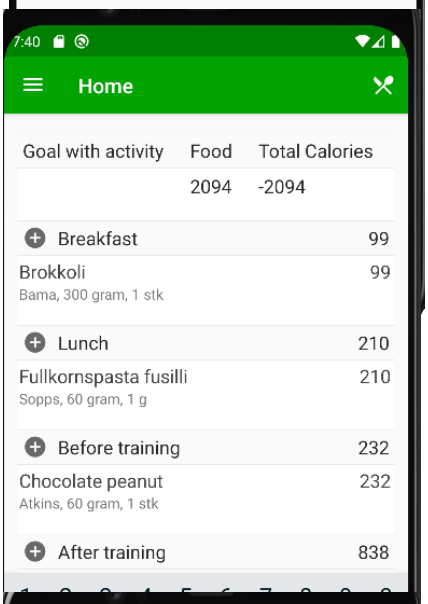
User added after SignUp screen:



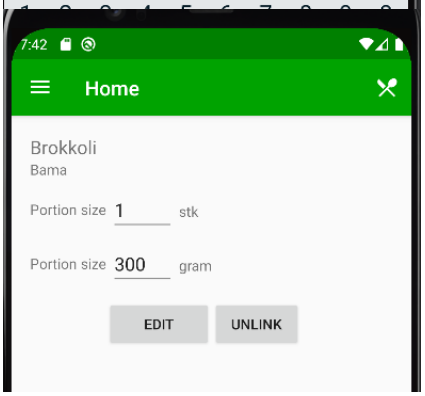
Home Screen:



On this screen, we can add the foods we take daily to the appropriate time period.

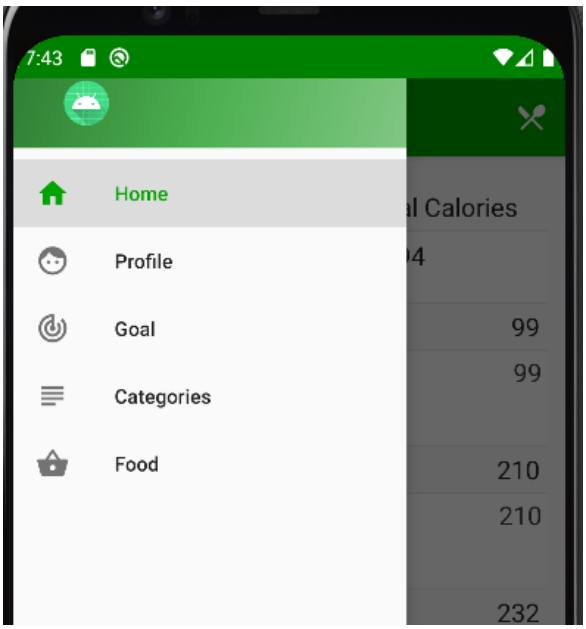


The total amount of calories consumed will be written above.



We can also update the food information after adding it to the list, in this case the database will also be updated.

NavBar:



-Home

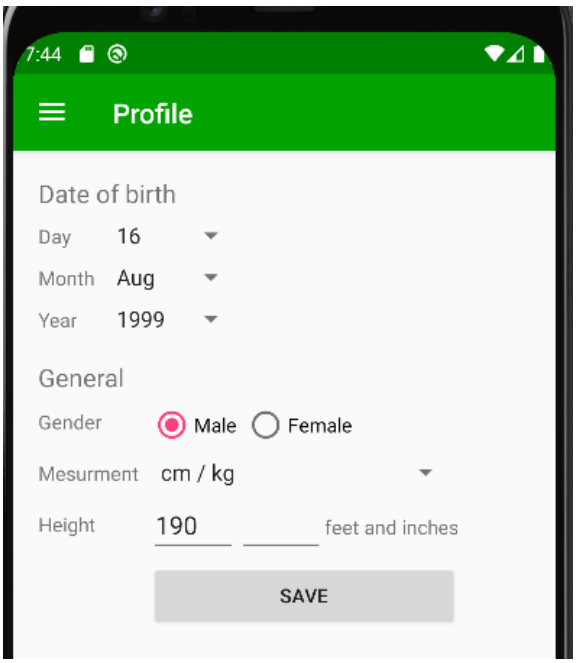
-Profile

-Goal

-Categories

-Food

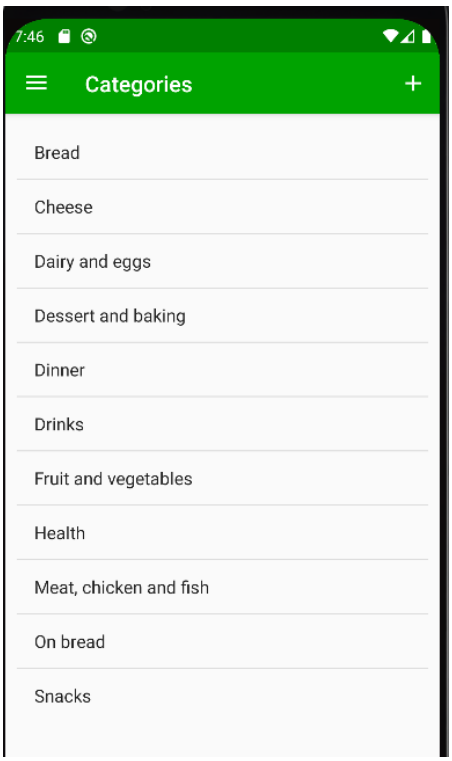
Profile Screen:



The user information on this screen can be updated later, so that more accurate calculations are made according to the changing weight information,

and these changes also take place in the database.

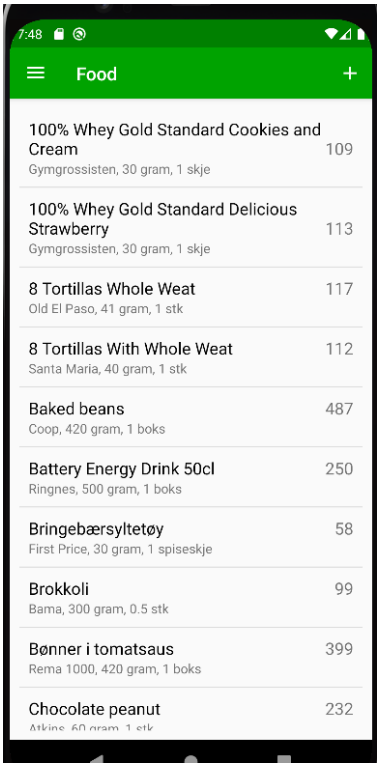
Categories Screen:



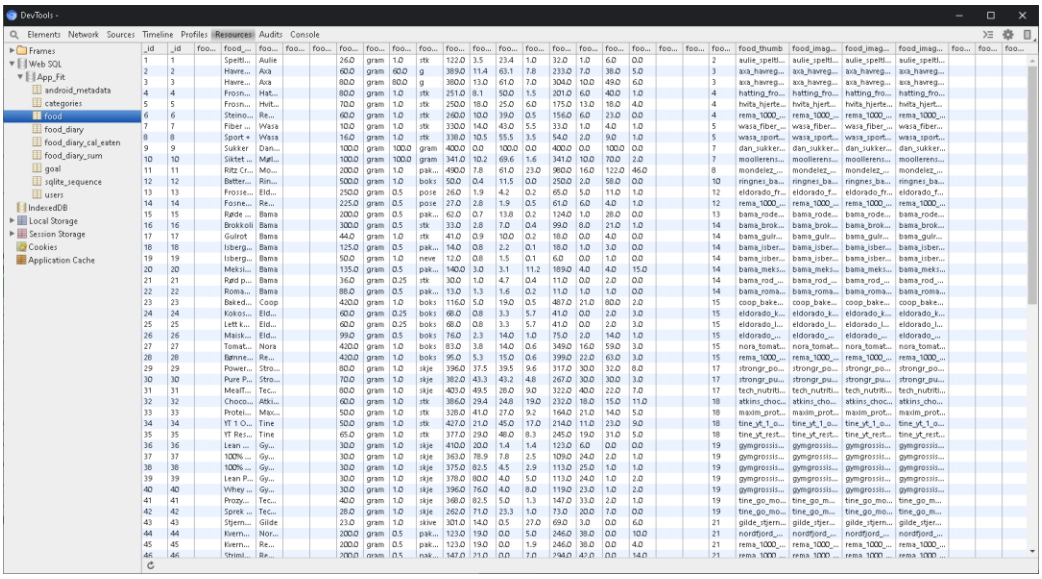
we can update and change these categories in the code blocks we have seen before:

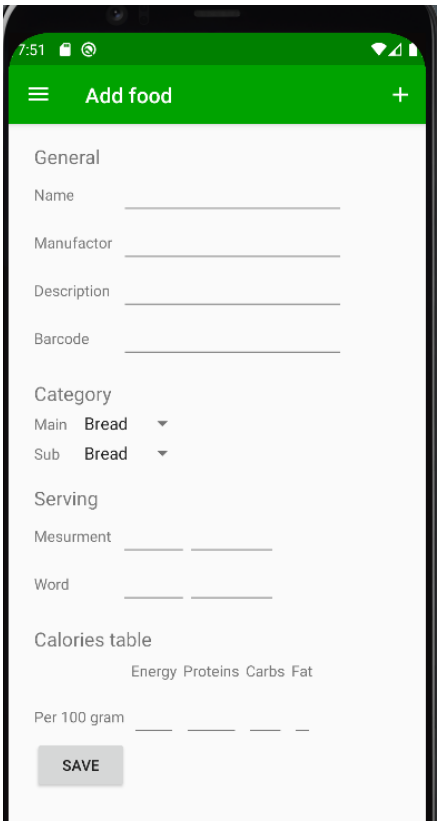
```
public void insertAllCategories(){
    setupInsertToCategories("NULL, 'Bread', '0', '', NULL");
    setupInsertToCategories("NULL, 'Bread', '1', '', NULL");
    setupInsertToCategories("NULL, 'Cereals', '1', '', NULL");
    setupInsertToCategories("NULL, 'Frozen bread and rolls', '1', '', NULL");
    setupInsertToCategories("NULL, 'Crispbread', '1', '', NULL");

    setupInsertToCategories("NULL, 'Dessert and baking', '0', '', NULL");
    setupInsertToCategories("NULL, 'Baking', '6', '', NULL");
    setupInsertToCategories("NULL, 'Biscuit', '6', '', NULL");


    setupInsertToCategories("NULL, 'Drinks', '0', '', NULL");
    setupInsertToCategories("NULL, 'Soda', '9', '', NULL");


    setupInsertToCategories("NULL, 'Fruit and vegetables', '0', '', NULL");
    setupInsertToCategories("NULL, 'Frozen fruits and vegetables', '11', '', NULL");
    setupInsertToCategories("NULL, 'Fruit', '11', '', NULL");
    setupInsertToCategories("NULL, 'Vegetables', '11', '', NULL");
    setupInsertToCategories("NULL, 'Canned fruits and vegetables', '11', '', NULL");
```

Food Screen:



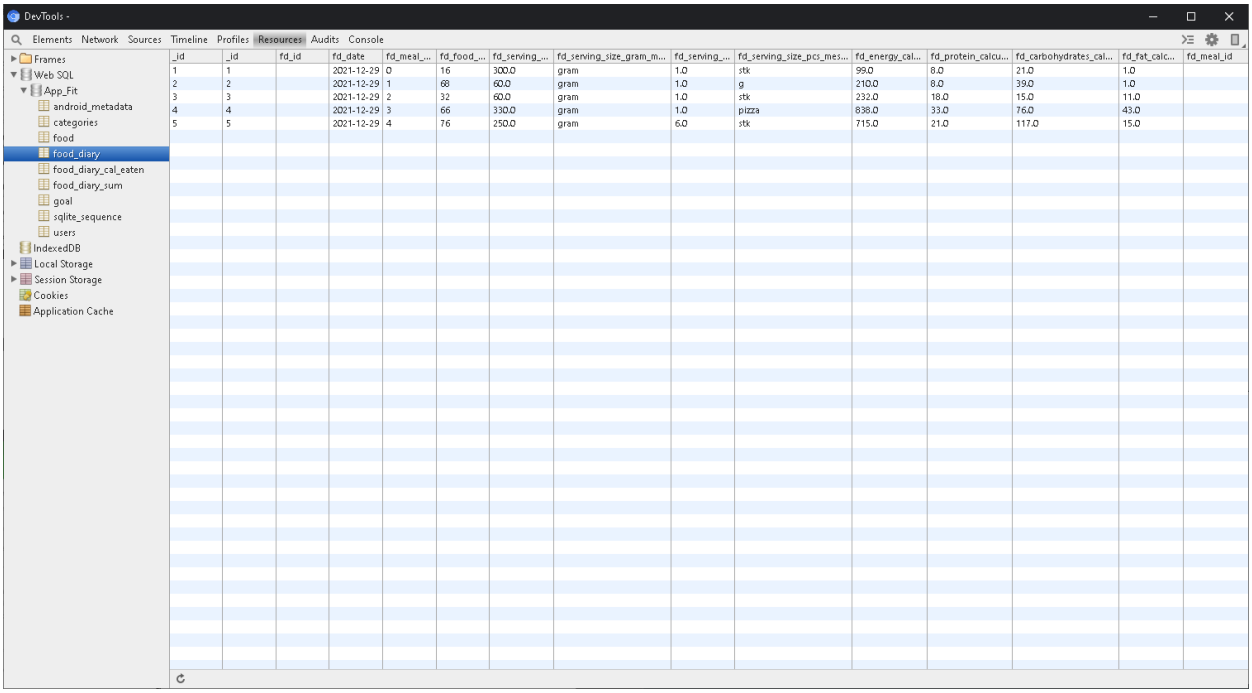
Example foods from database created before:





If the food we want is not found in the database, we can add the food we want from the add food screen by entering the calorie values.

Database from added foods to list:



what can be added to the application later and suggestions for improvement >>

* After the login screen, the guidance of the application can be provided with the target information received from the user.

as an example > the amount of weight desired to be given or to be gained

* the user can be offered different diet and nutrition types options

In this way, the application also addresses different types of nutrition.

Examples:

Ketogenic diet

Vegetarian diet

Vegan diet

Weight watchers diet

South beach diet

Raw food diet

…

***** Exercises can be offered by the app along with workout advice and explanations

* In order to better control the diet in general, weekly and monthly calendars can be added so that we can follow it more easily.

Resources:

Android Development – Full Course

https://www.youtube.com/watch?v=fis26HvvDII&ab_channel=freeCodeCamp.org

Stack Overflow

https://stackoverflow.com/

Android App Development in java – Tutorial Series

https://www.youtube.com/watch?v=tZvjSl9dswg&t=13432s&ab_channel=CalebCurry

SQLite Database for Android – Full Course

https://www.youtube.com/watch?v=312RhjfetP8&ab_channel=freeCodeCamp.org

Chrome DevTools – Course

https://www.youtube.com/watch?v=gTVpBbFWry8&ab_channel=freeCodeCamp.org

Chrome DevTools – Device Mode

https://www.youtube.com/watch?v=FrAZWiMWRa4&ab_channel=GoogleDevelopers

Google Developer Training - Android Developer Fundamentals course

Head First Android Development by Dawn Griffiths and David Griffiths, O'reilly, 2017.

The Busy Coder's Guide to Android Development, Mark L. Murphy, CommonsWare, 201