The presentation of project “Calculating calories”.

The purpose of application to give instrument for analyzing calories that body gets and spends during the day and changing the weight.

Structure of application for calculating calories.

Each application that calculates calories consists of 4 parts:

1. To input parameters of body to calculate recommended number calories to change the weight.
2. To get number calories during the meals of day and calculate.
3. In day report to calculate balance calories that body gets and spends during the day and to compare with recommended that we calculated early.
4. To get history on a basis of information that we got early.

Features application about body parameters.

- User input gender, weight, age, height, style activity and the application calculates according to 3 methods average, minimum and maximum calories. For slimming application takes minimum from 3 methods. For weight gain application takes maximum from 3 methods.
- Recommended calories minimum is minus 20% and Extra calories is minus 40% for slimming.
- The main feature of application is: saving results of calculating in Firebase using the time stamp and getting and managing it from Firebase with Recycler View dynamically even during the day.

Calculating BMR (basal metabolic rate). Method Harris-Benedict old.

- The old Method of Harris-Bebedict.(1918year)

For female gender:

BMR = 655,0955 + (9,5634 * weight in kg.) + (1,8496 * height in sm.) – (4.6756 * age in years)

For male gender:

BMR = 66,4730 + (13,7516 * weight in kg.) + (5.0033 * height in sm.) – (6,7550 * age in years)

Calculating BMR (basal metabolic rate). Method Harris-Benedict new.

The new Method of Harris-Bebedict.(1984year)

BMR = 447,593 + (9,247 * weight in kg.) + (3,098 * height in sm.) – (4.330 * age in years)

For male gender:

BMR = 88,362 + (13,397 * weight in kg.) + (4,799 * height in sm.) – (5617 * age in years)

Calculating AMR (active metabolic rate).

- It depends on which style of life body has.
- 1.2 – passive lifestyle.
- 1.375 – moderate activity 1-3 times in a week.
- 1.55 – intense load 3-5 times in a week.
- 1.725 – Sportsman 6-7 times in a week.

Calculating daily norm of calories.

- Number of day calories = BMR * AMR.

The method of Miffilin-Sangore.

- Formula Miffilin.
- For female gender:

BMR = (10 * weight in kg.) + (6.25 * Height in sm.) – (5 * age in years) – 161

- For male gender:

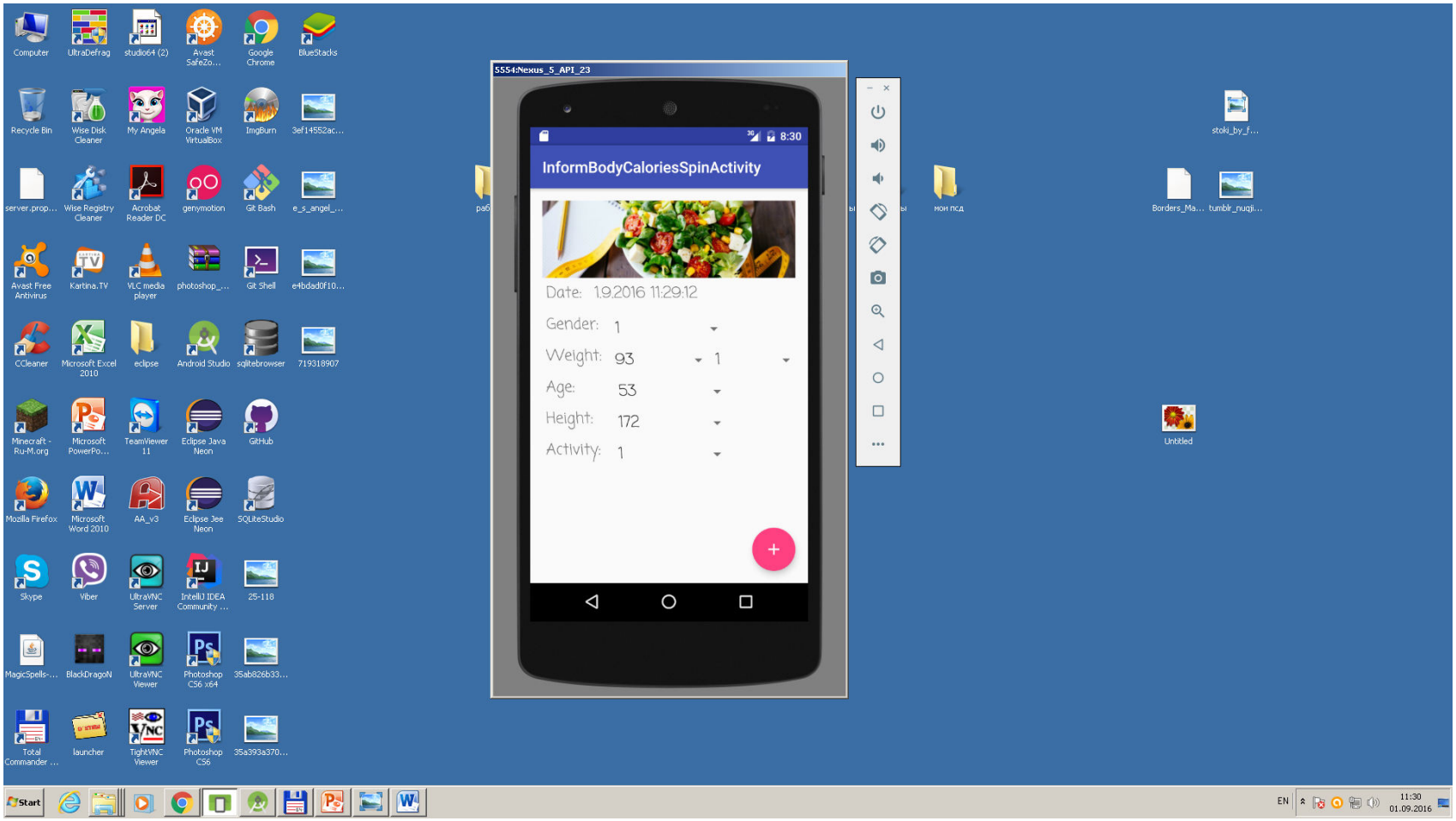
BMR = (10 * weight in kg.) + (6.25 * Height in sm.) – (5 * age in years) + 5.

Number of day calories = BMR * AMR.

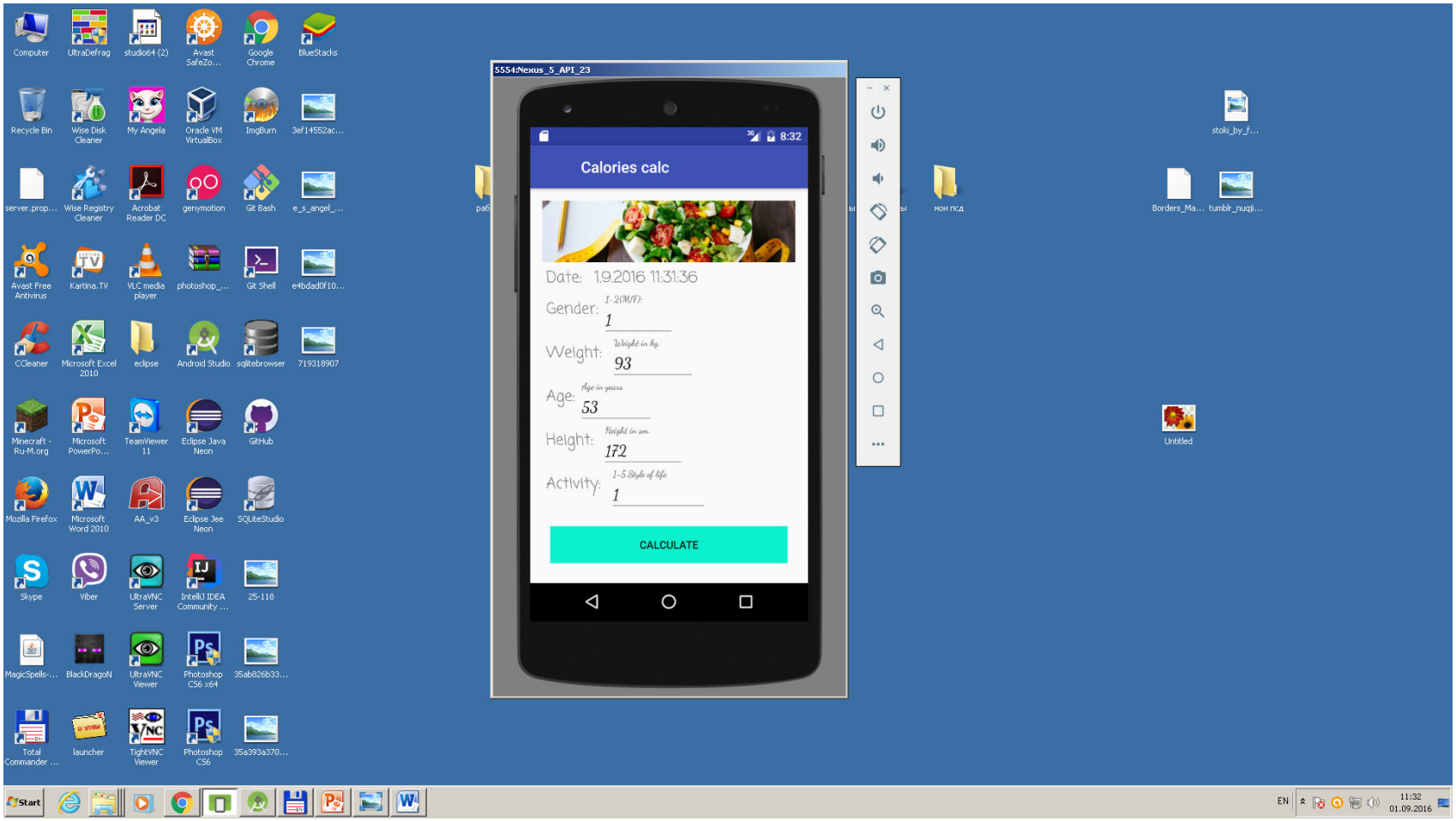
Calculating BMI (Body Mass Index)

- BMI = m / (h * h).
- If BMI < 15 – acute underweight.
- If BMI (15…20) – underweight.
- If BMI (20…25) – normal weight.
- If BMI (25…30) – overweight.
- If BMI > 30 – obesity.

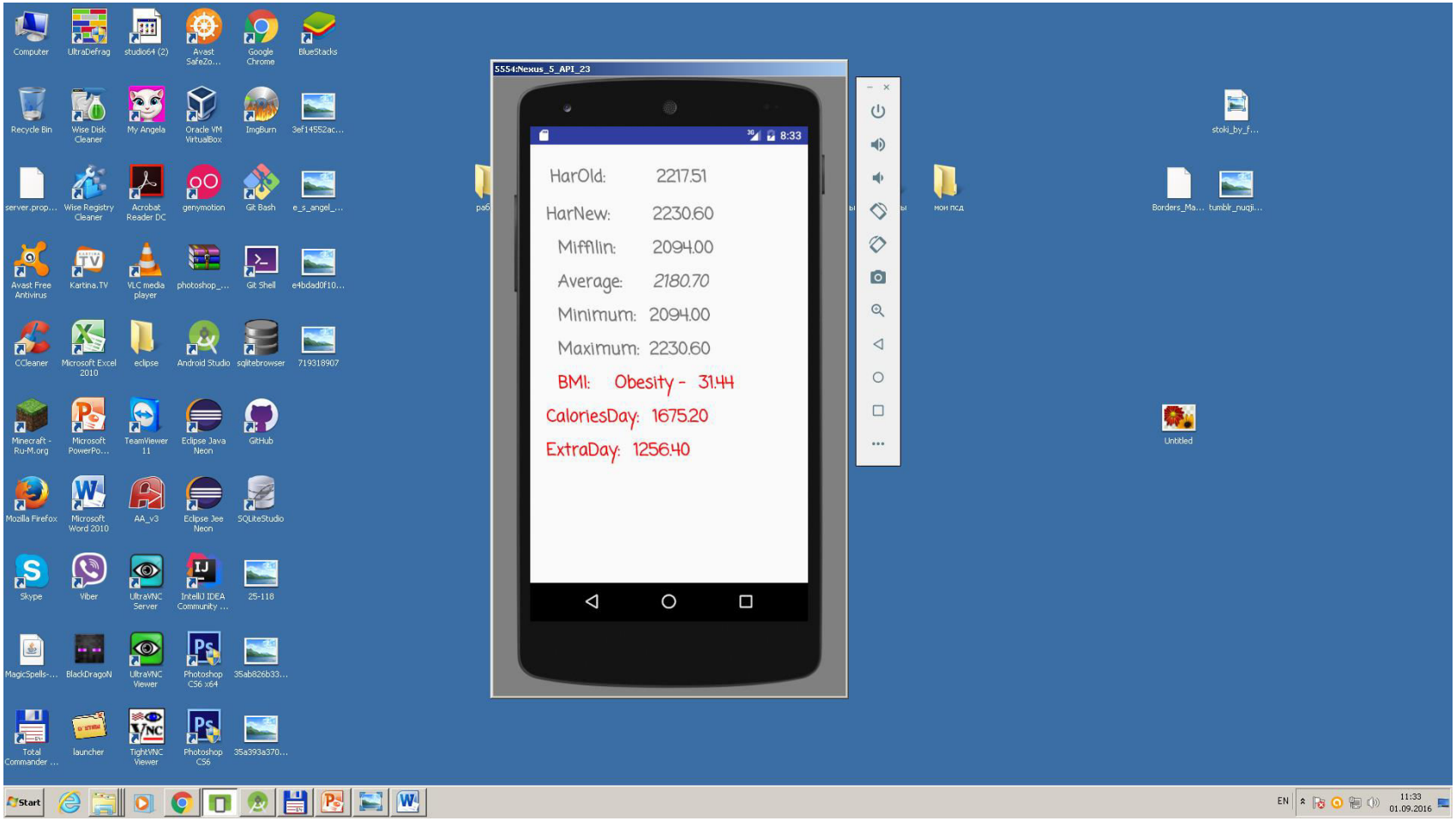
Input body parameters



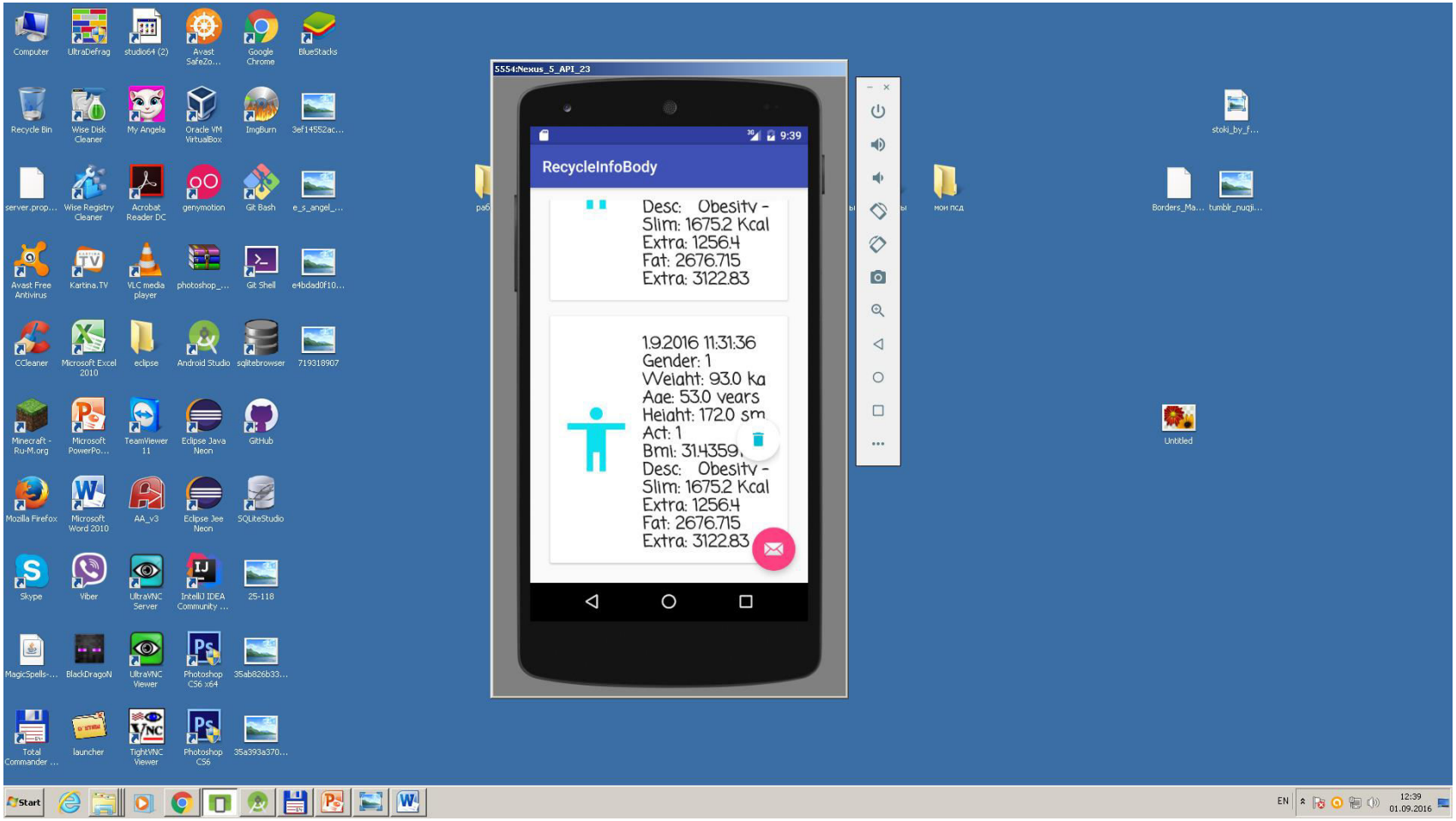
Another input form for body parameters.



Result of calculating and saving results in Firebase.



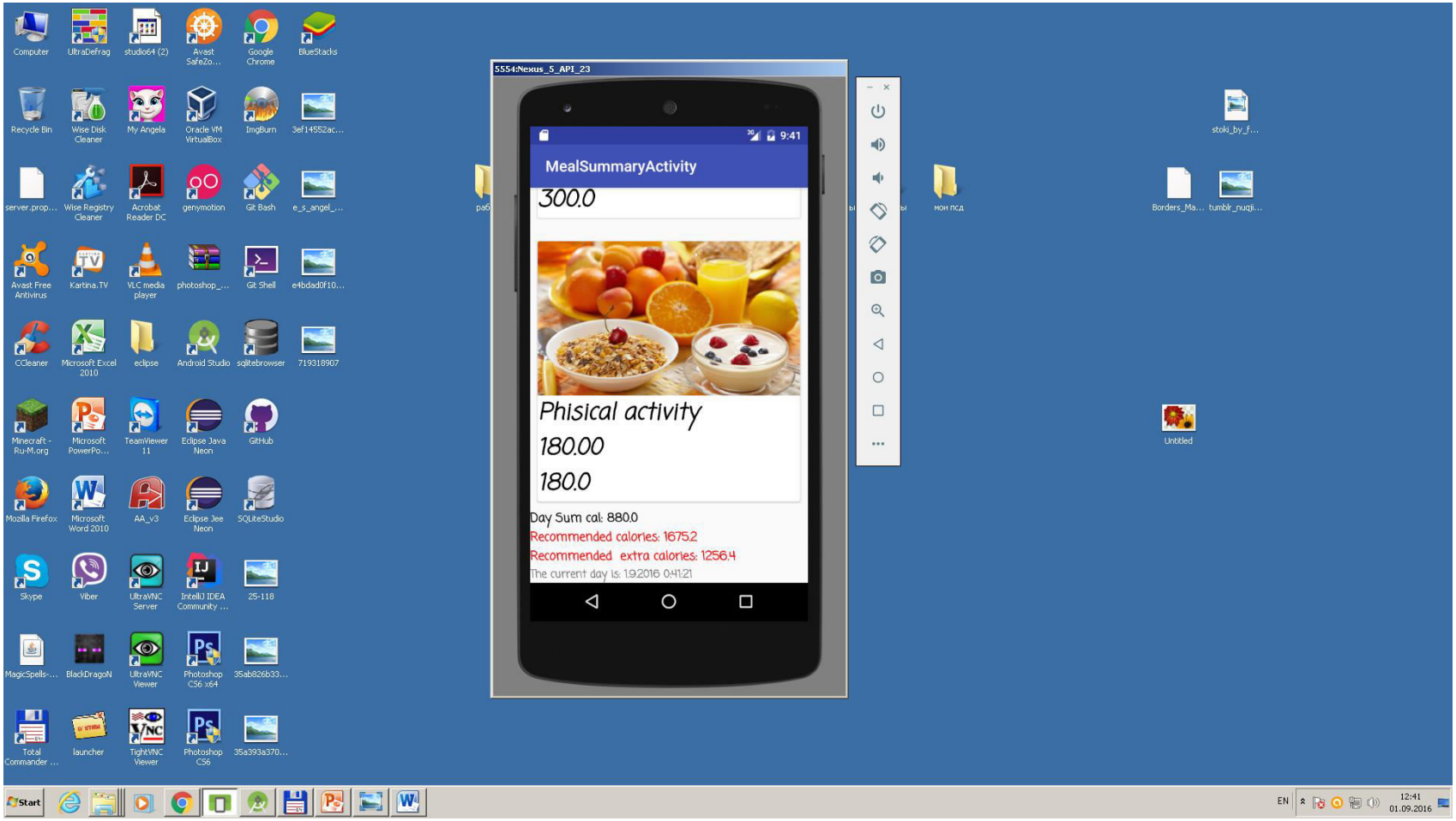
Getting results of body's parameters and calculate recommended numbers calories



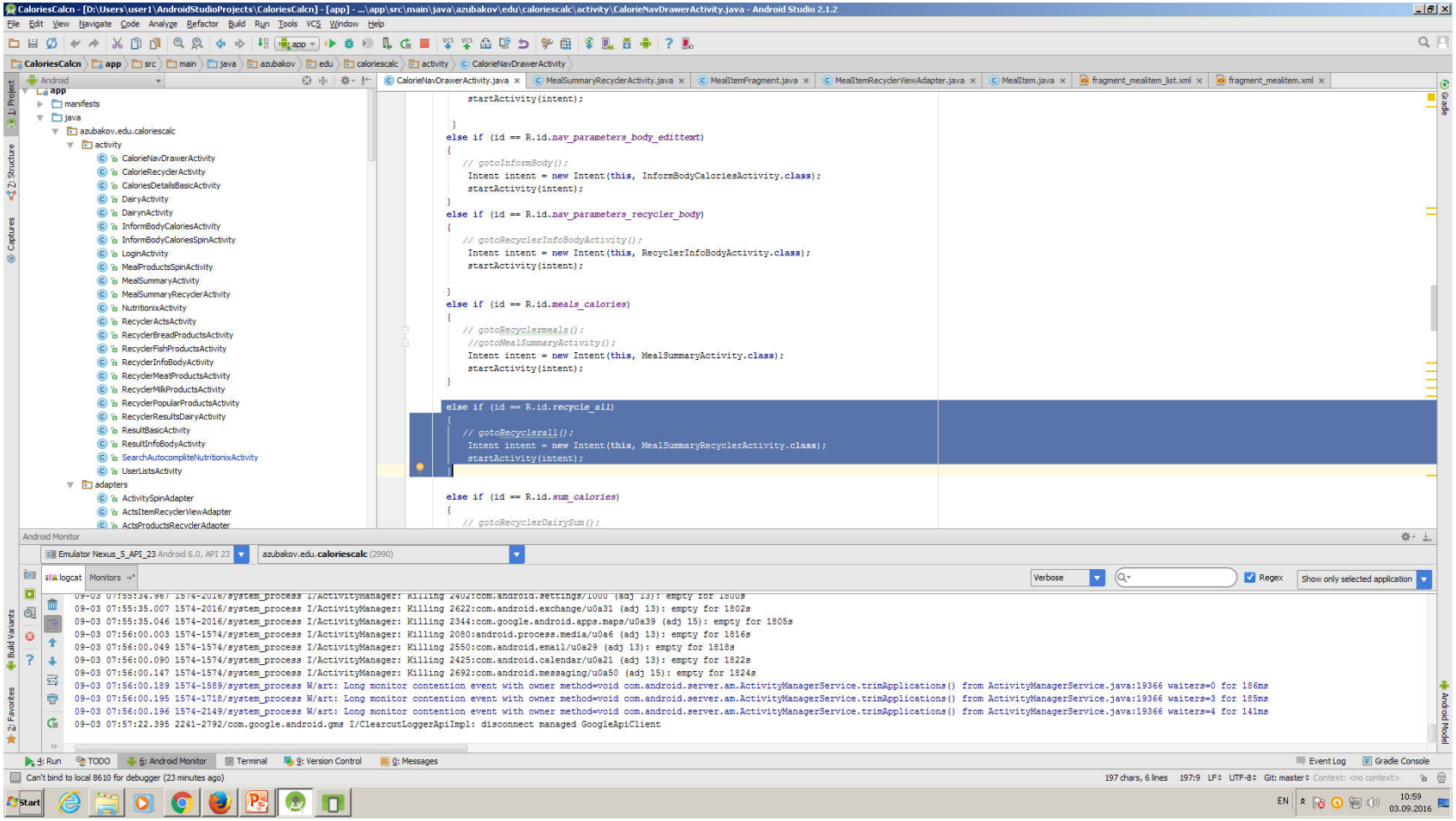
Features of part that gets calories during the day.

- We get number calories during the breakfast, lunch, dinner, snakes and activity.
- We create report that calculate number of calories according to meals and day, compare it with recommended and save information in Firebase using the time stamp.
- It gives possibility to save information in Firebase dynamically to get statistic about calories during the day.
- Another applications save report in the end of day only in mysqli.

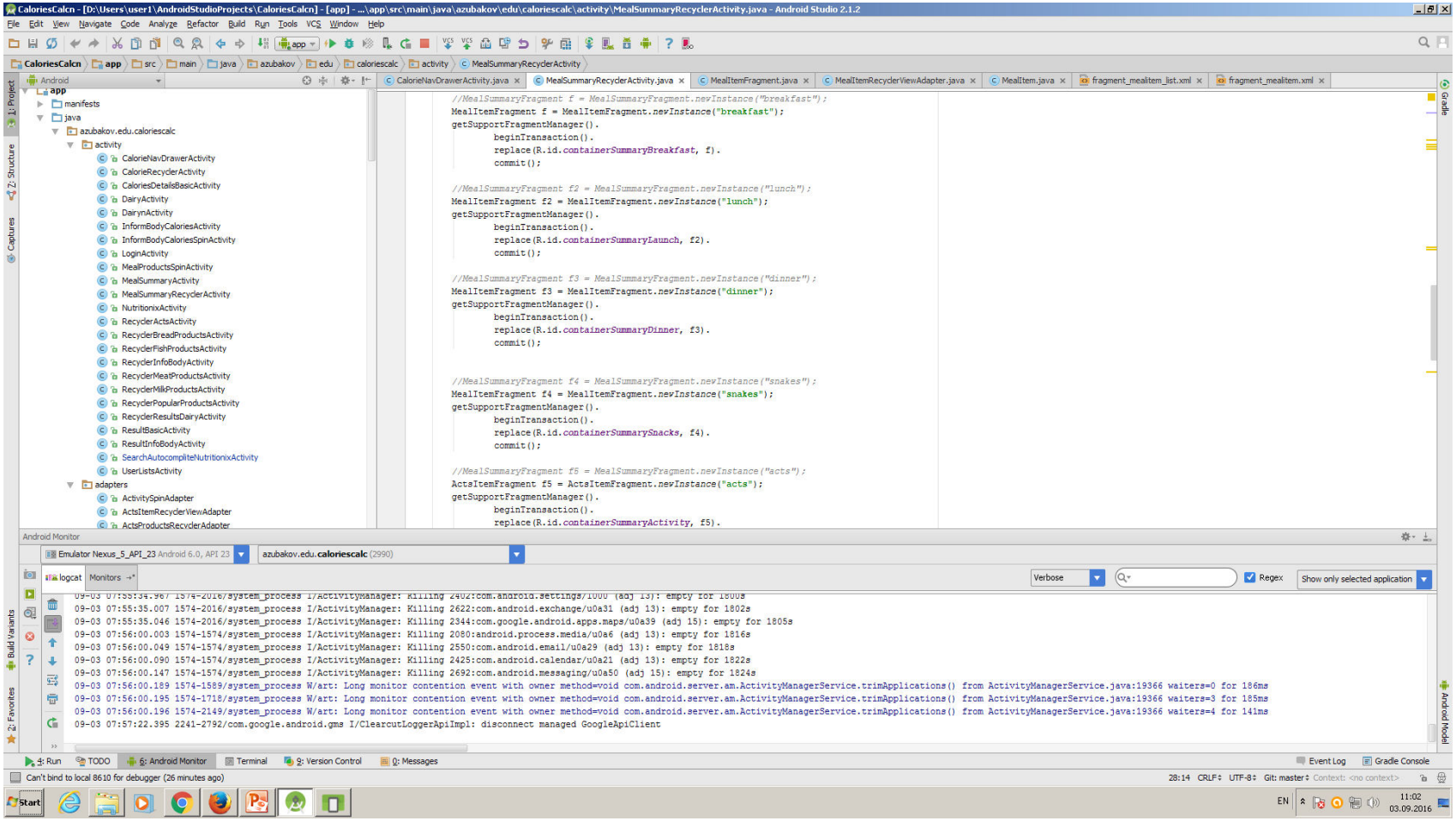
Report of day results. It gives balance calories during the day compared with recommended calories.



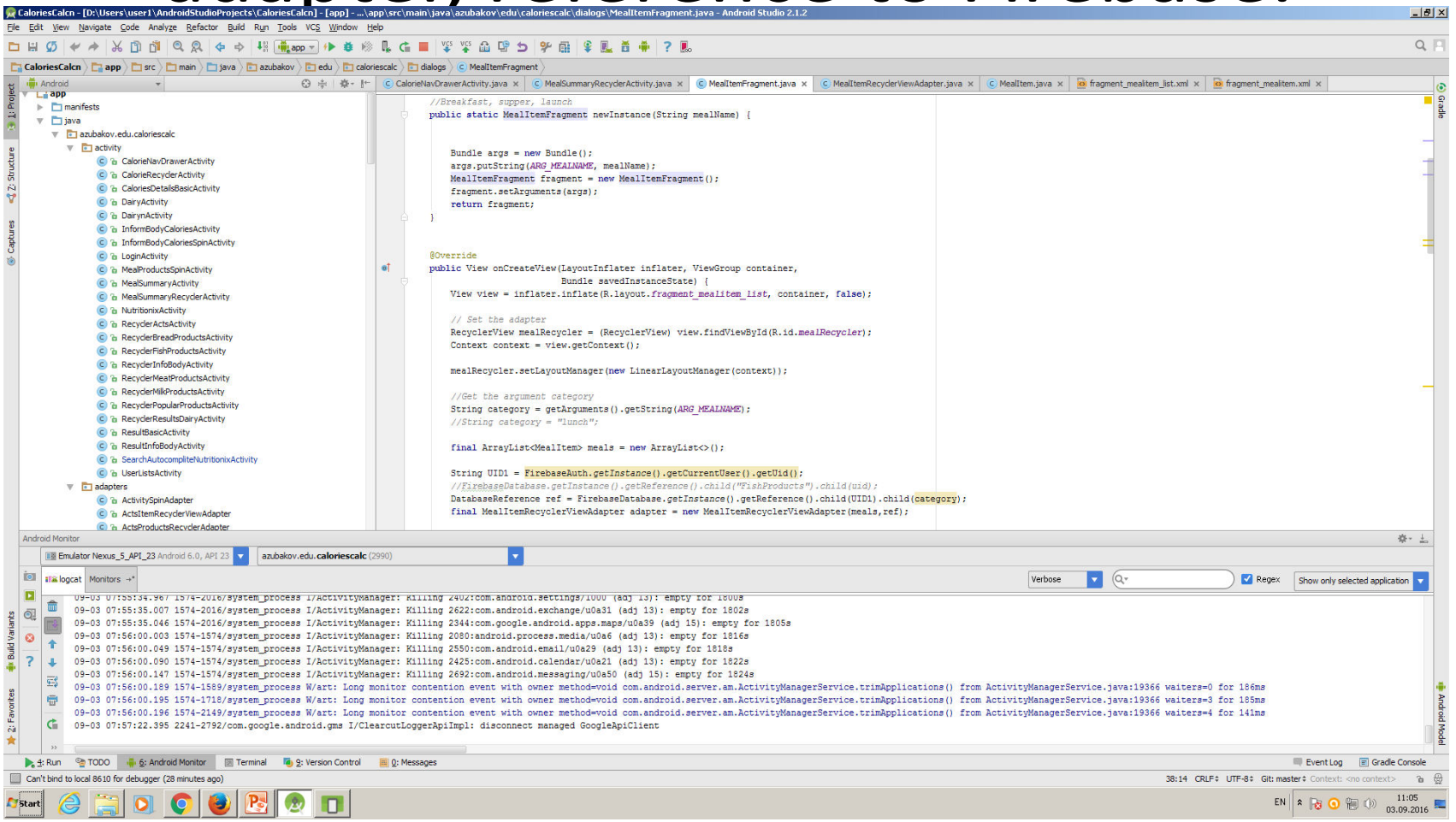
It begins from menu in left panel of CalorieNavDrawerActivity. We open MealSummaryRecyclerActivity.



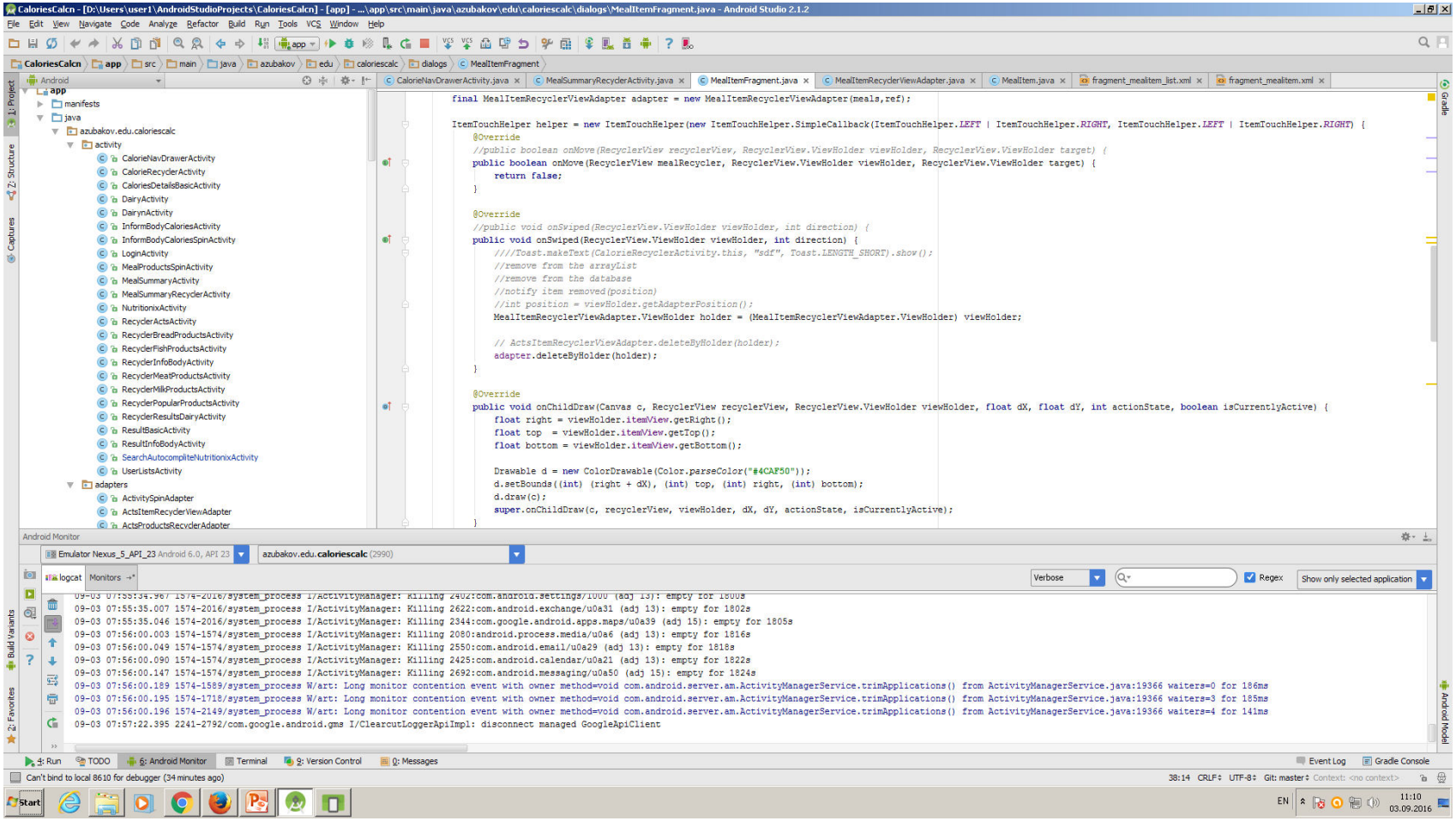
We input information to containers from fragment MealItemFragment.



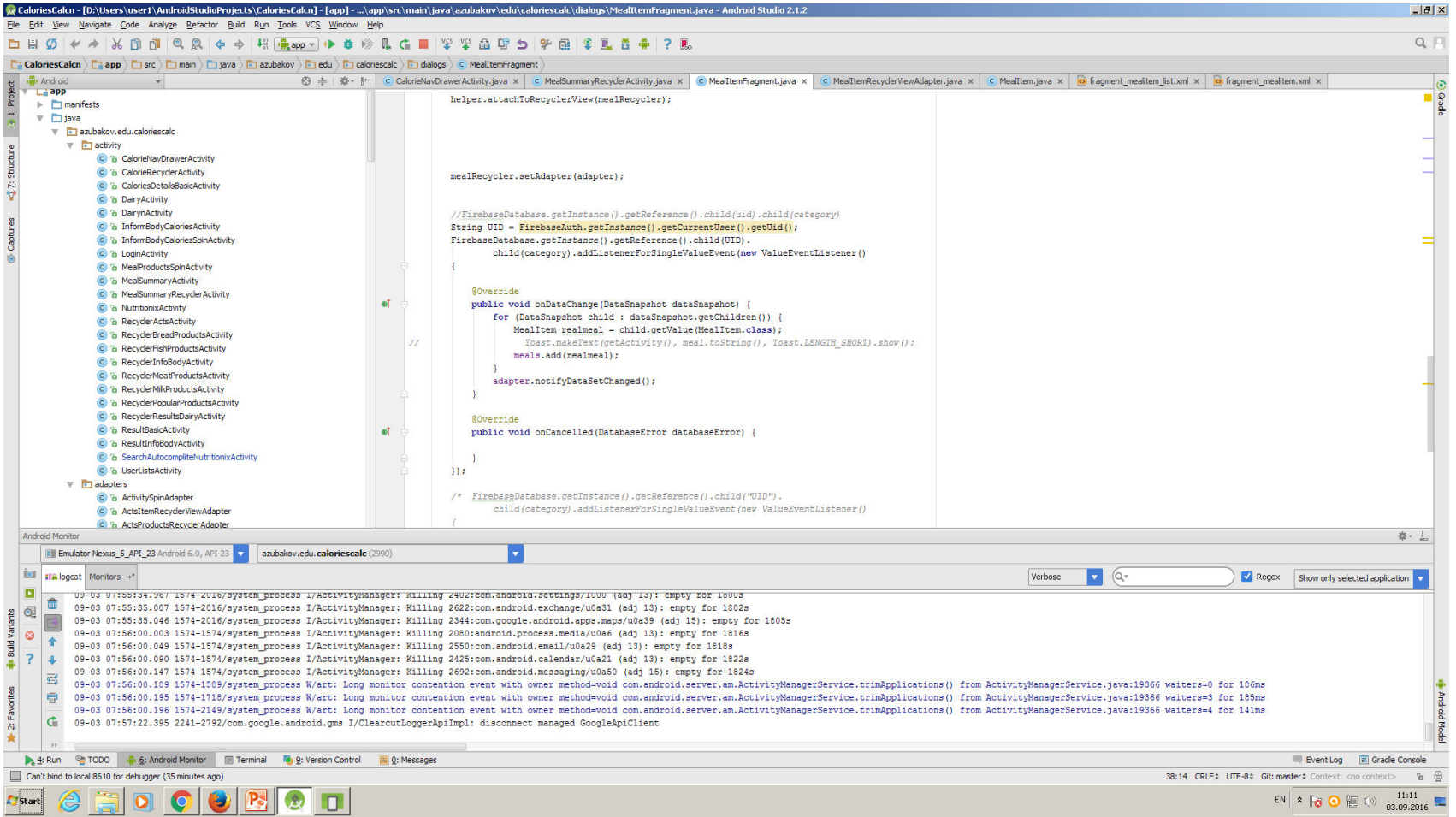
In MealItemFragment we define newInstance, inflate xml file, RecyclerView, Arraylist for meals, adapter, reference to Firebase.



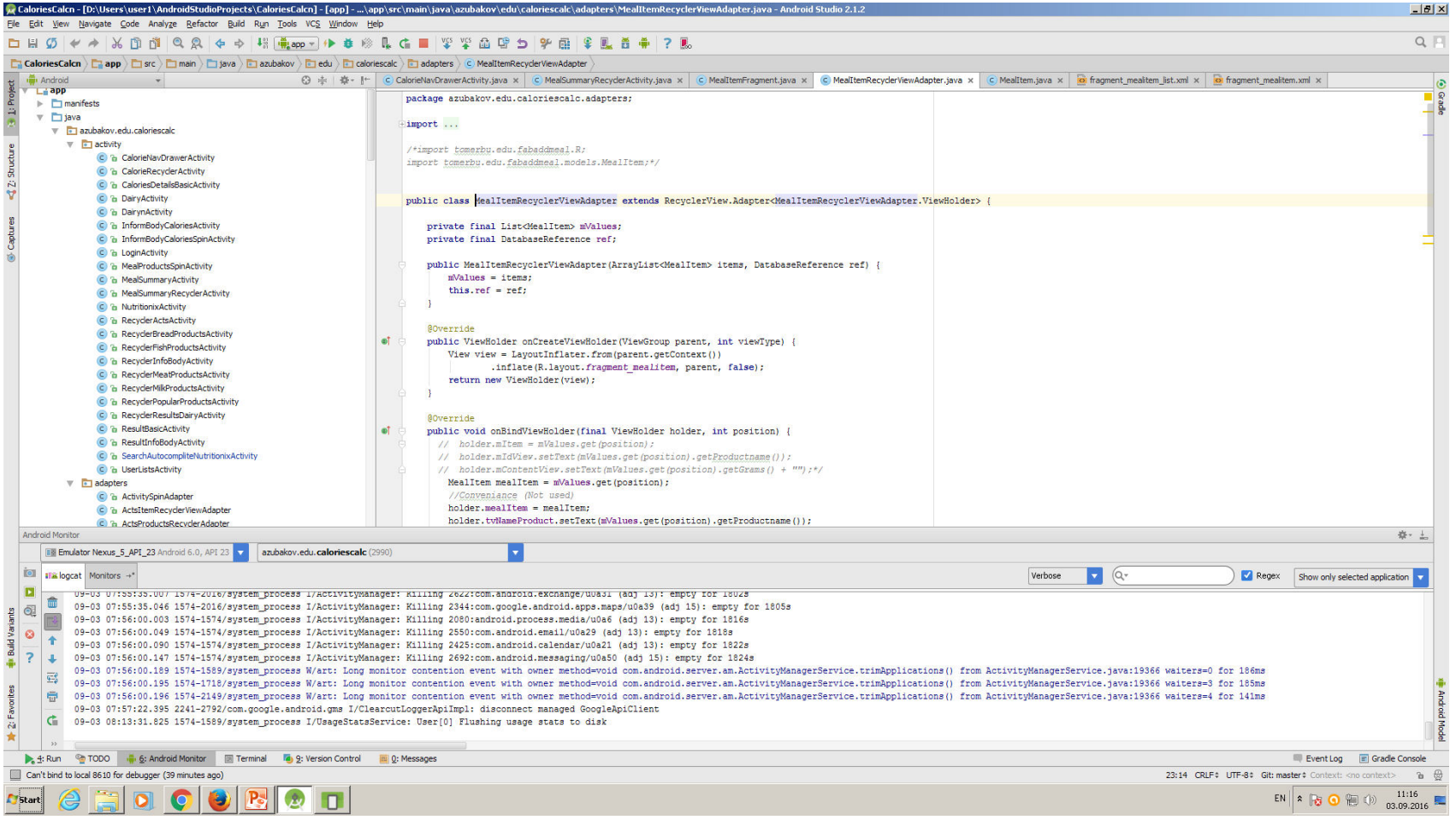
It is part for defining swipe delete.



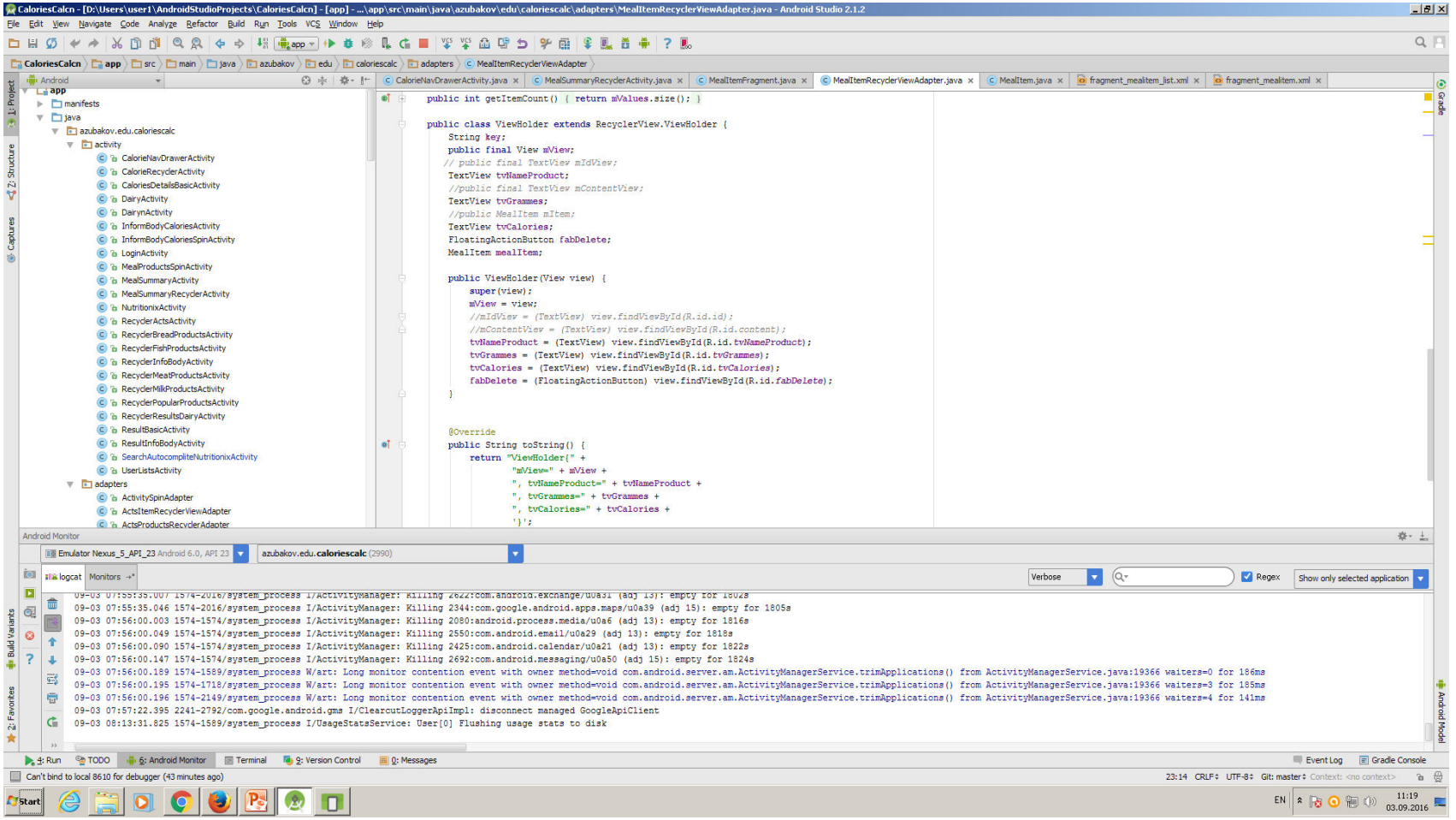
We define adapter and fill information to array meals from Firebase Datasnapshot child according to model MealItem.



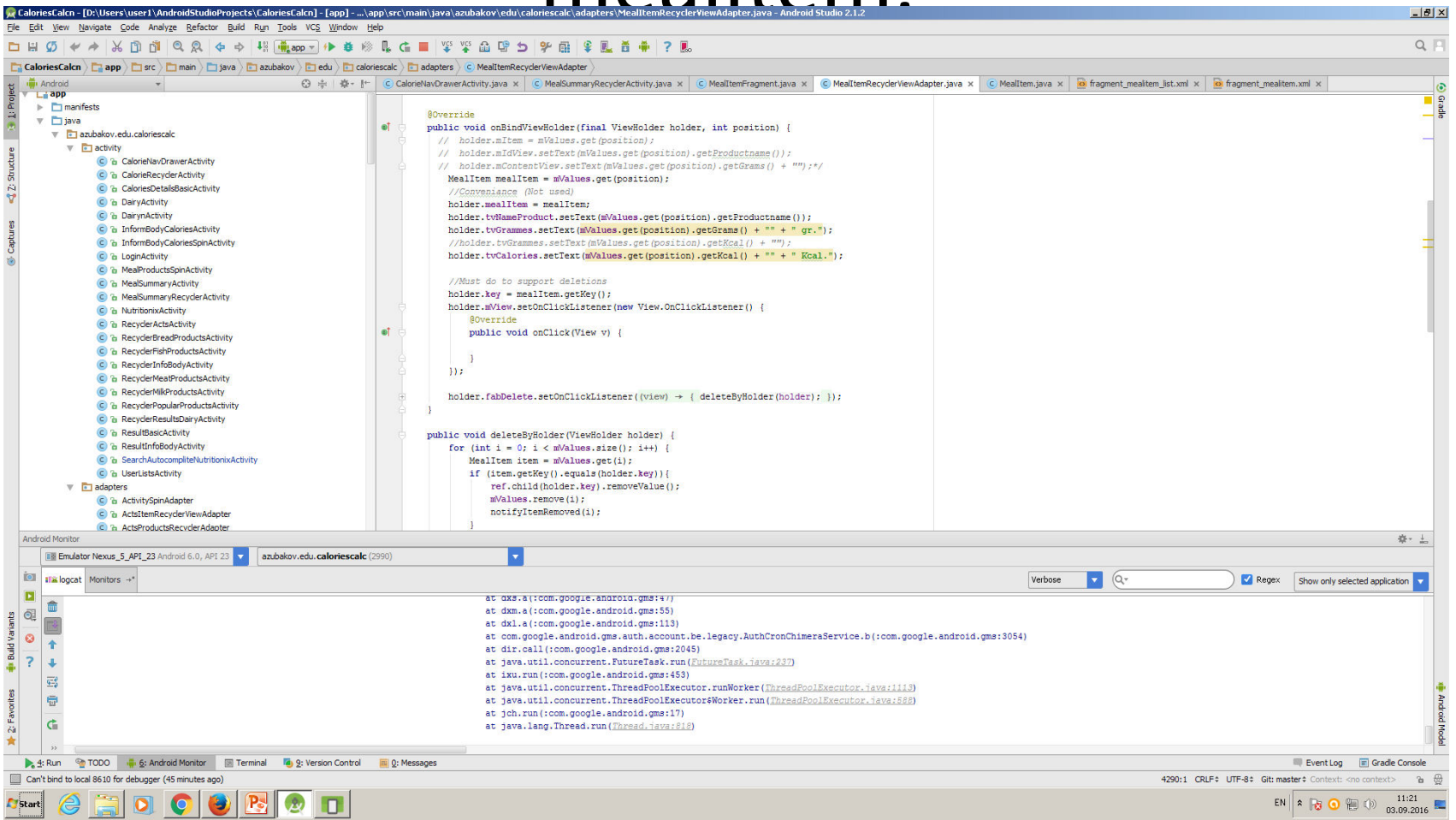
In MealRecyclerViewAdapter we get ArrayList<MealItem> and DatabaseReference in constructor.



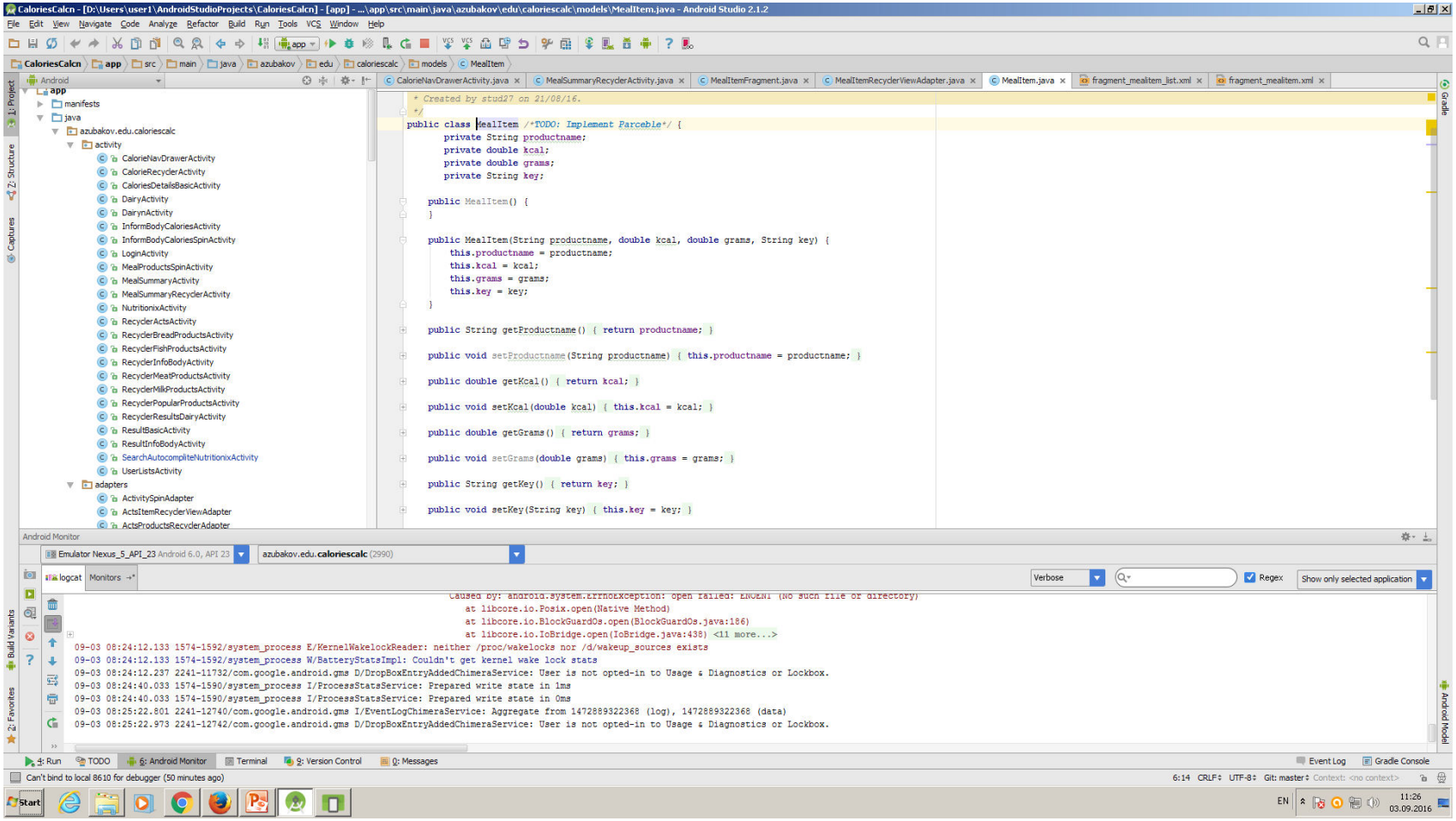
In MealRecyclerViewAdapter we register fields of from fragment mealitem.xml file in ViewHolder.



On BindViewHolder we fill fields of form from position of array according to methods of model of class mealItem.



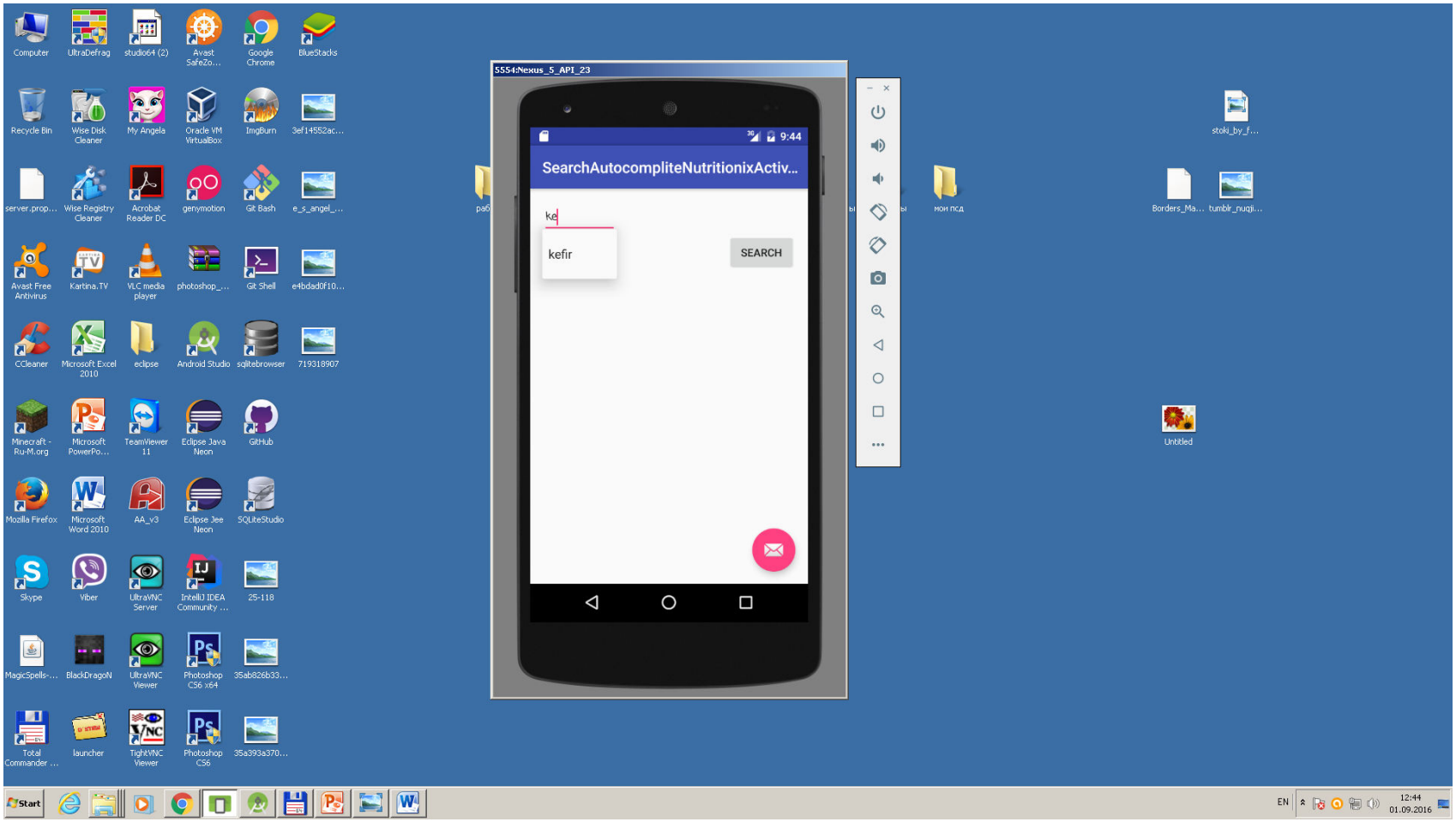
It is a model- class MealItem.



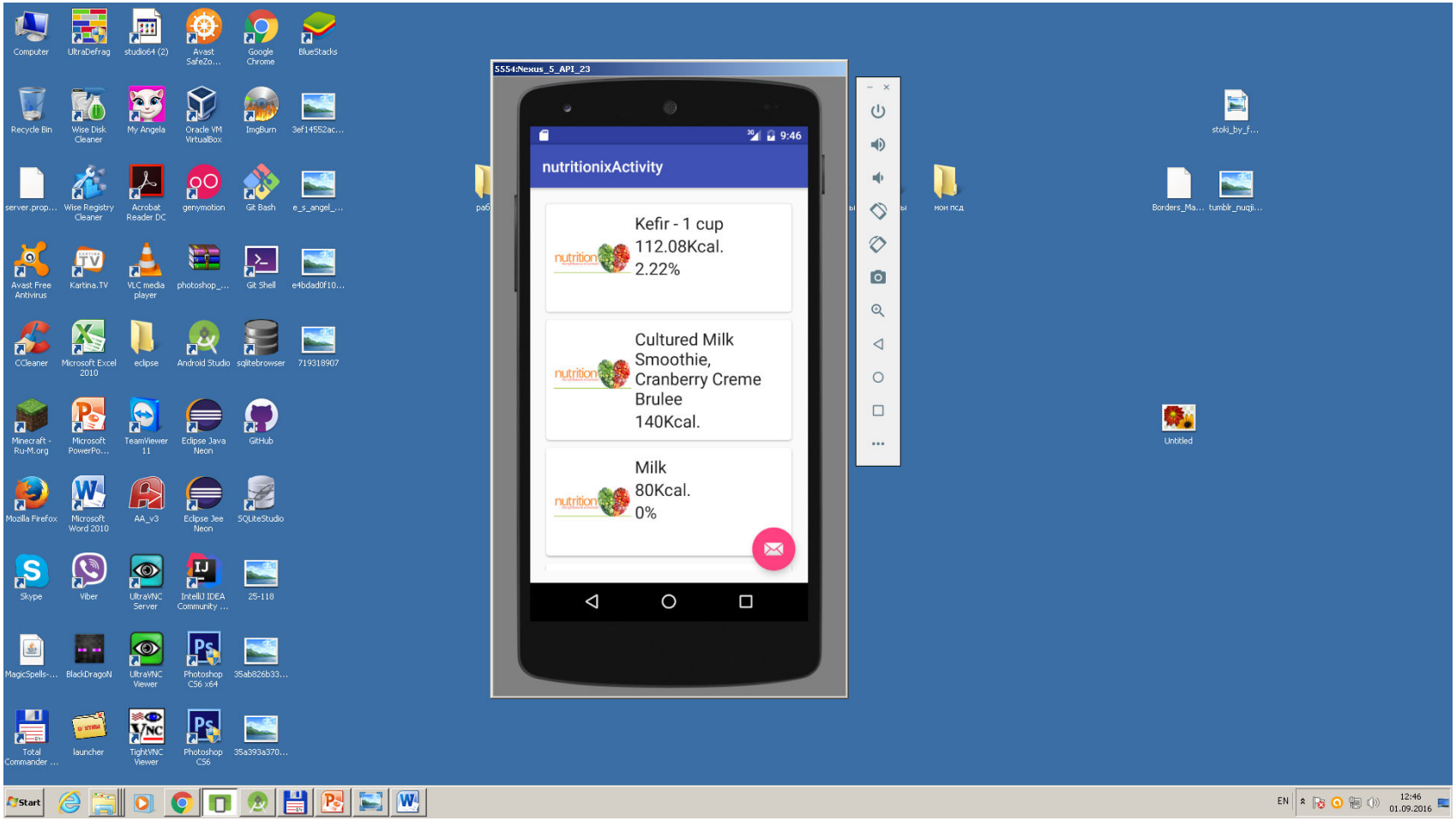
Features concerning JSON and databases of products of food.

- To get information about calories of meal we use Nutrition API of firm Nutritionix that gives free JSON format of nutritons products of food. We use element that named AutoCompliteTextview that uses static array for helping to complite input product that we search in database of Nutritionix.
- We created in Firebase databases of bread, fish, milk, meat and popular products with values of calories for 100 gr and database of physical activities.

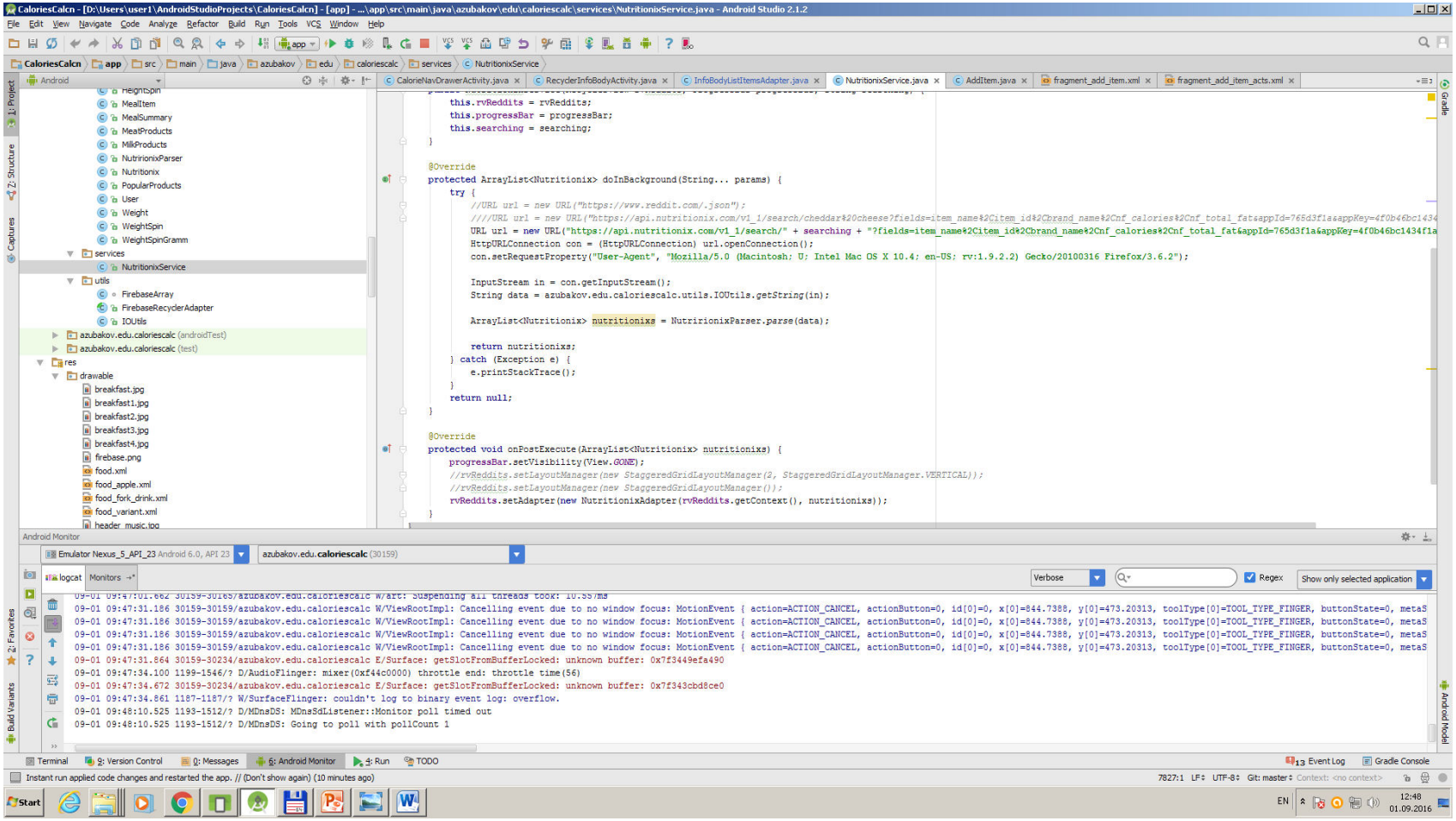
AutoCompleteTextView that complete value of searching product from array in JSON form.



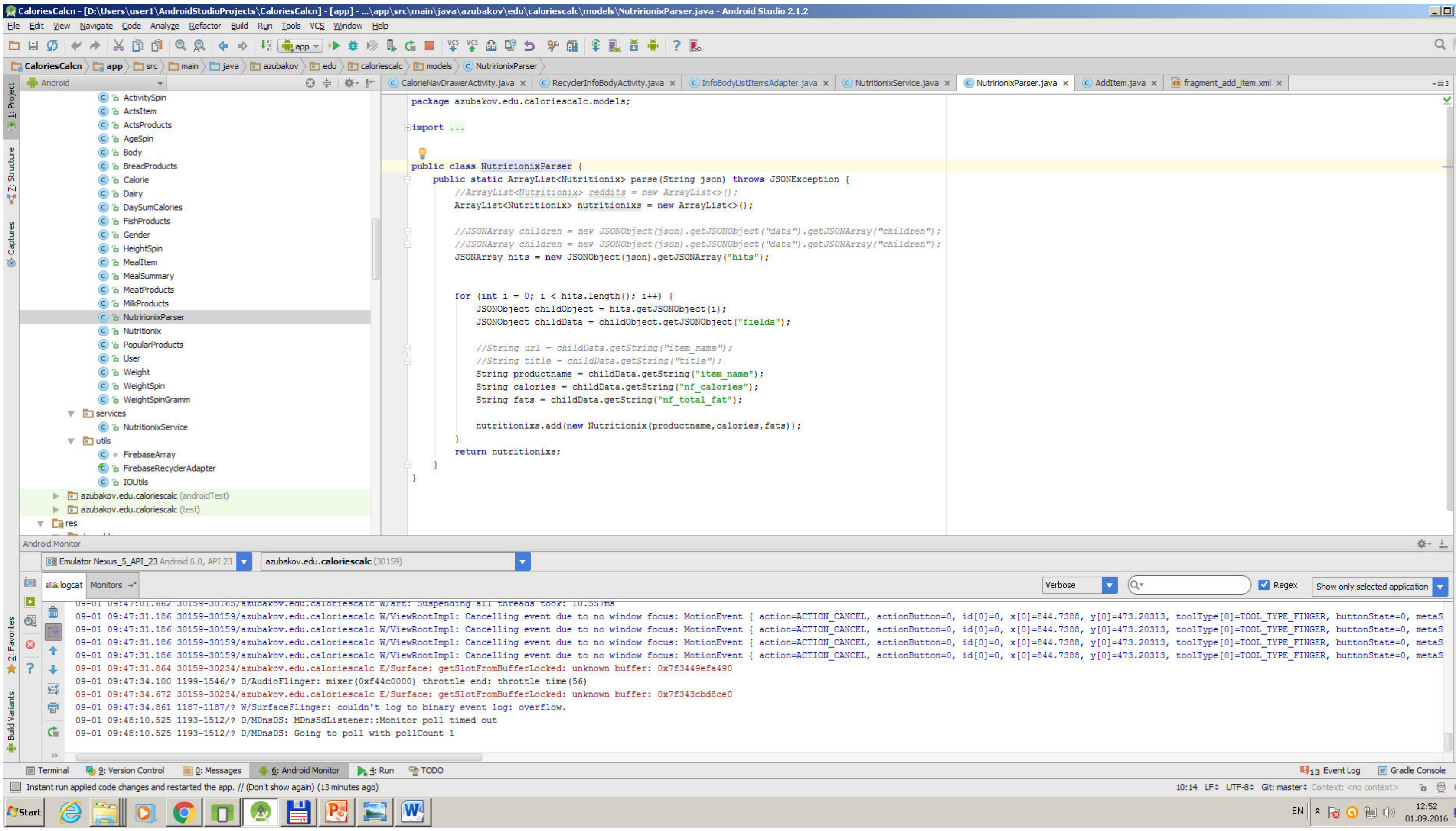
Results of JSON that we got from Nutritionix API.



We use file of service that extends AsyncTask. It use NutritionixParser.



Nutritionixparser that takes data from JSON.

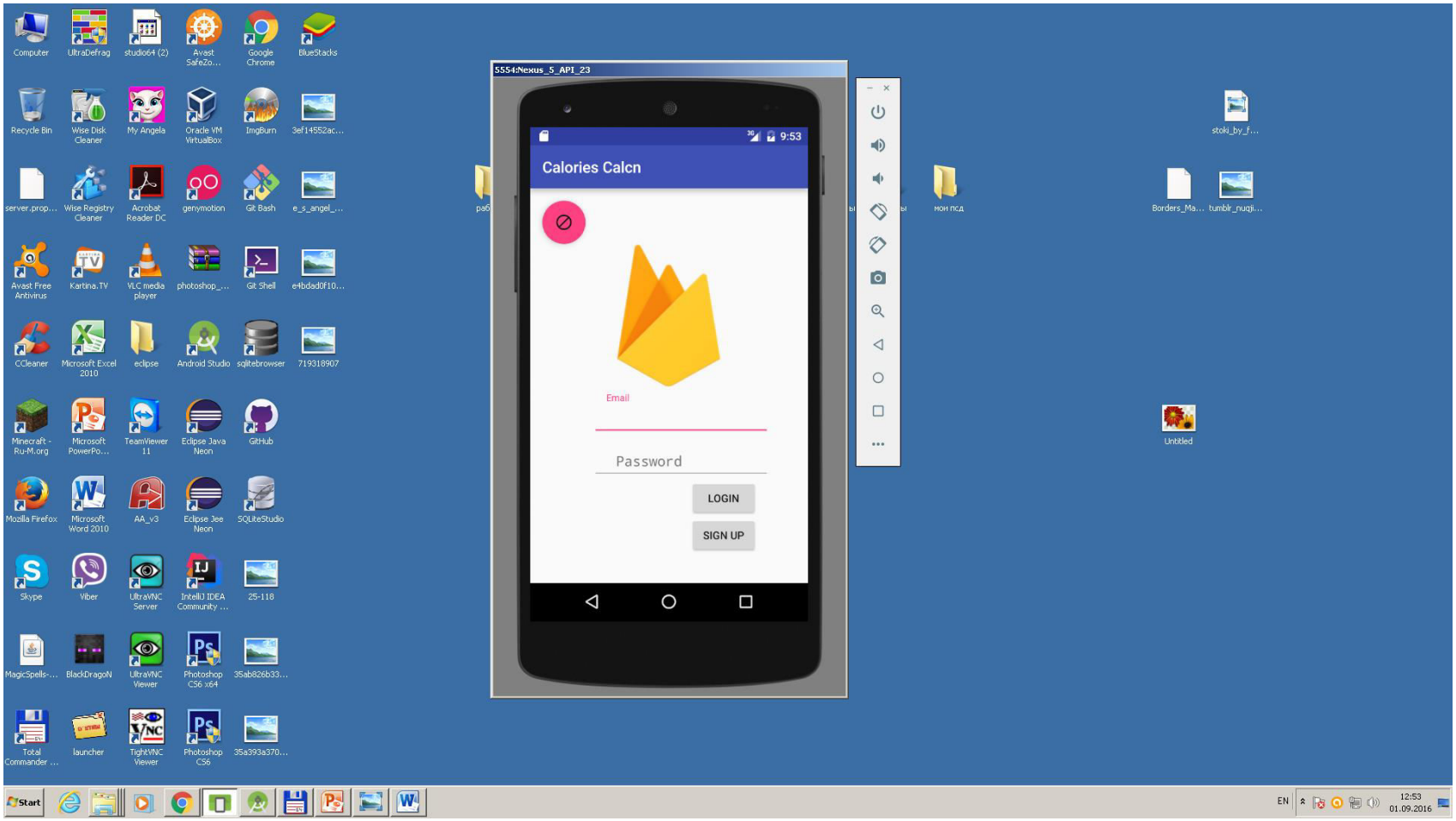


Features of application about login form.

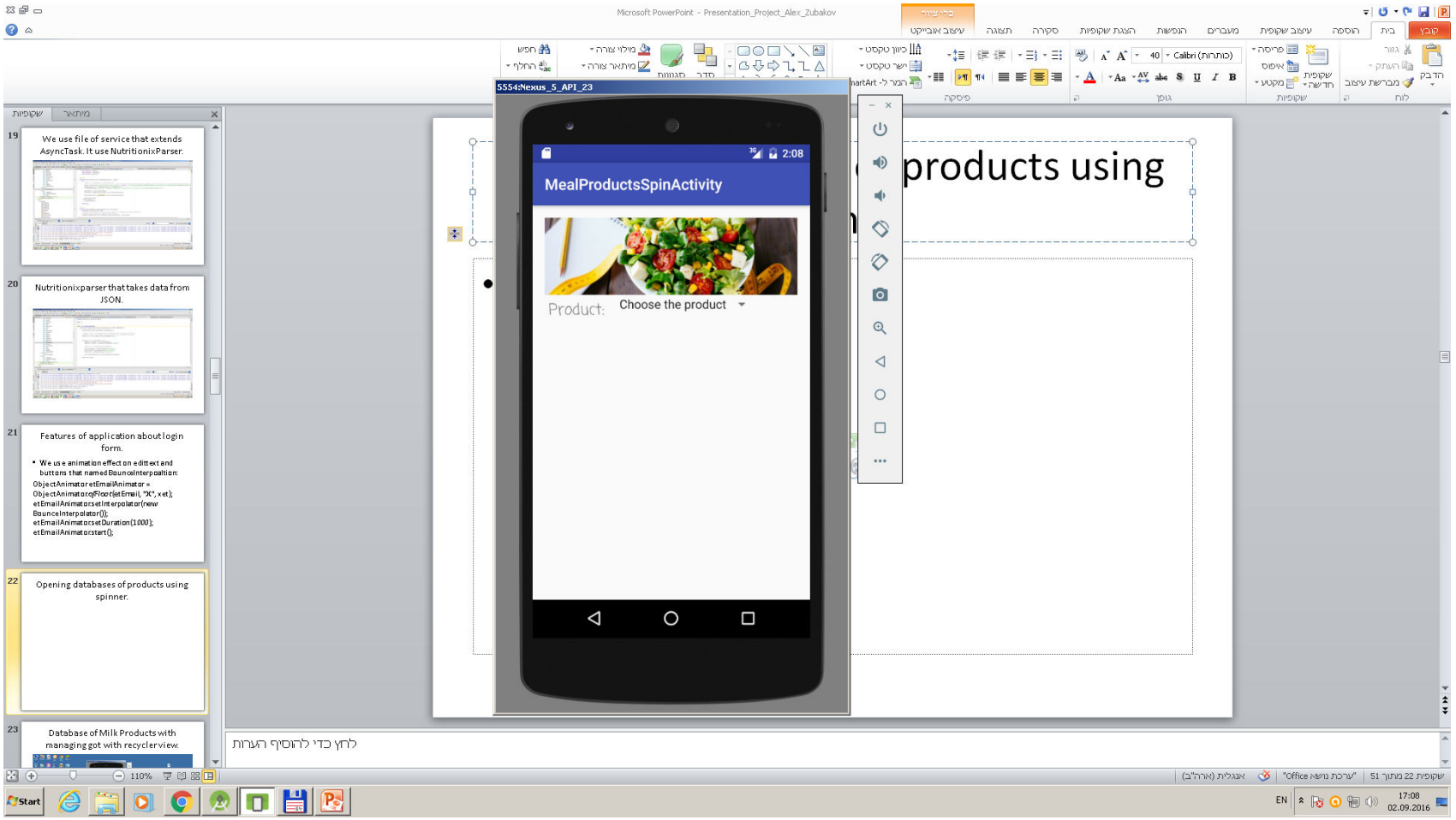
- We use animation effect on edittext and buttons that named BounceInterpoaltion:

```
ObjectAnimator etEmailAnimator =
ObjectAnimator.ofFloat(etEmail, "X", xet);
etEmailAnimator.setInterpolator(new
BounceInterpolator());
etEmailAnimator.setDuration(1000);
etEmailAnimator.start();
```

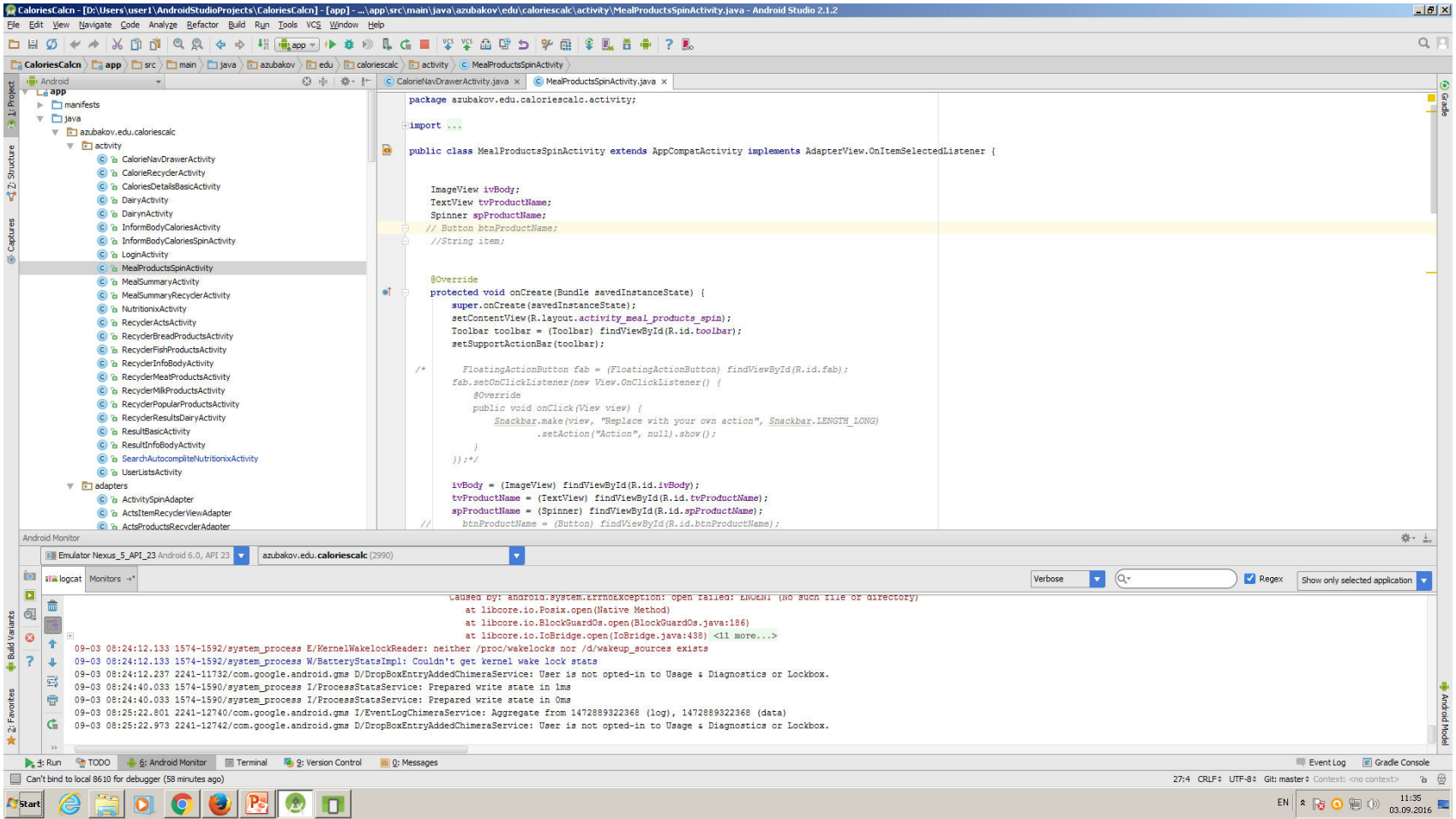
Login form with animation of EditText of Email, Password and two buttons,



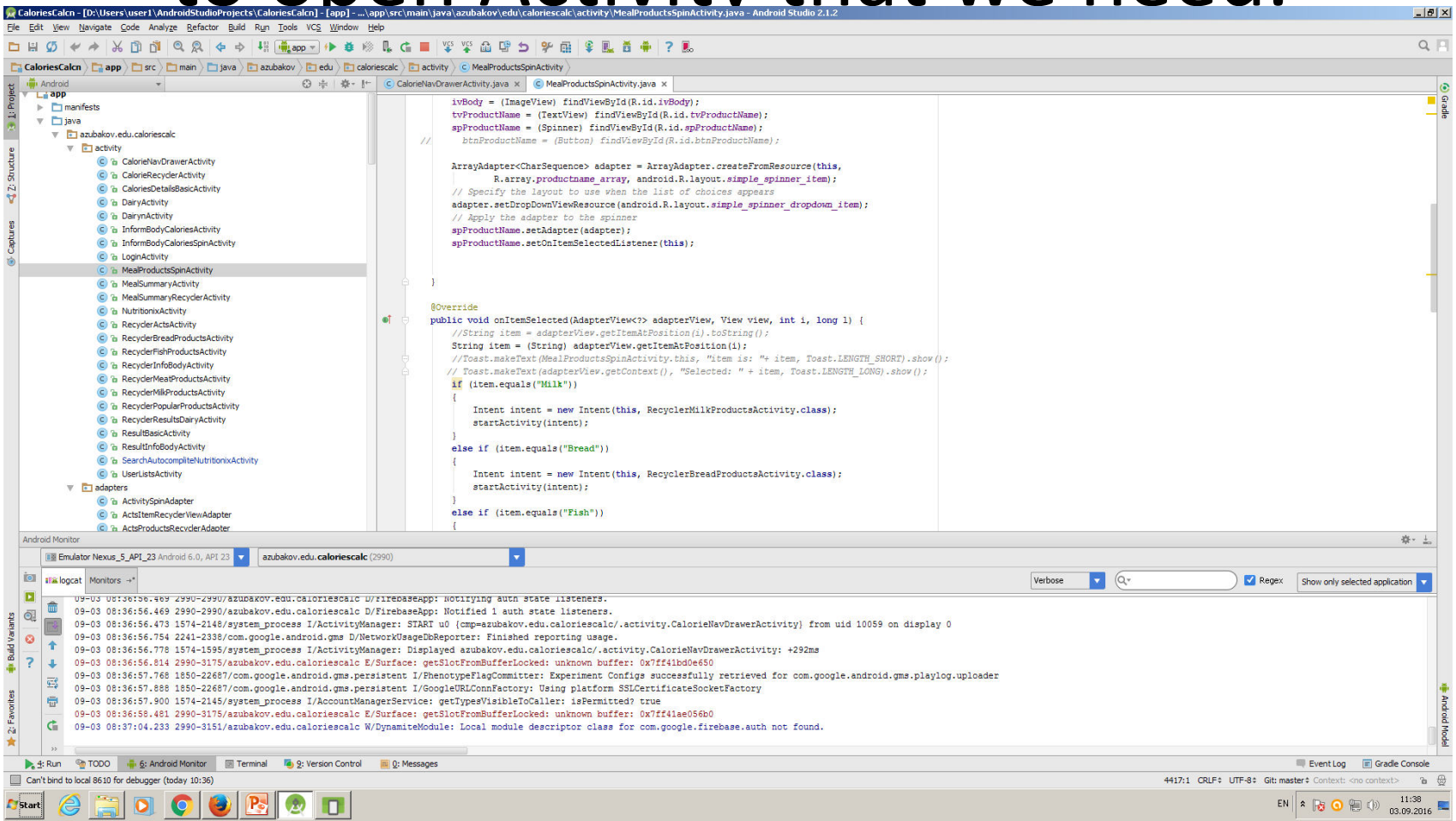
Opening databases of products using spinner.



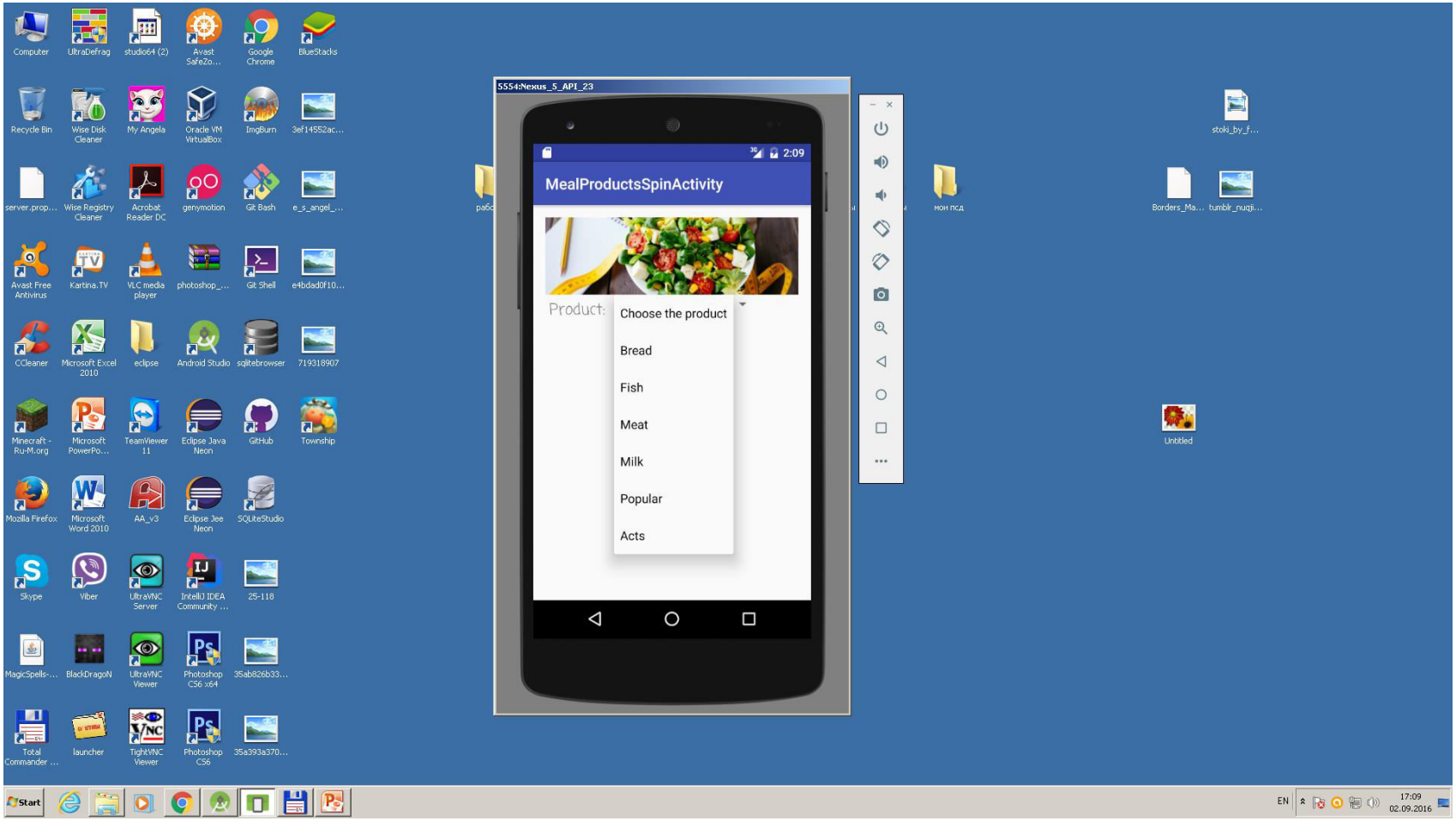
In MealProductSpinActivity we register fields of form and implemets interface AdapterView.OnItemSelectedListener.



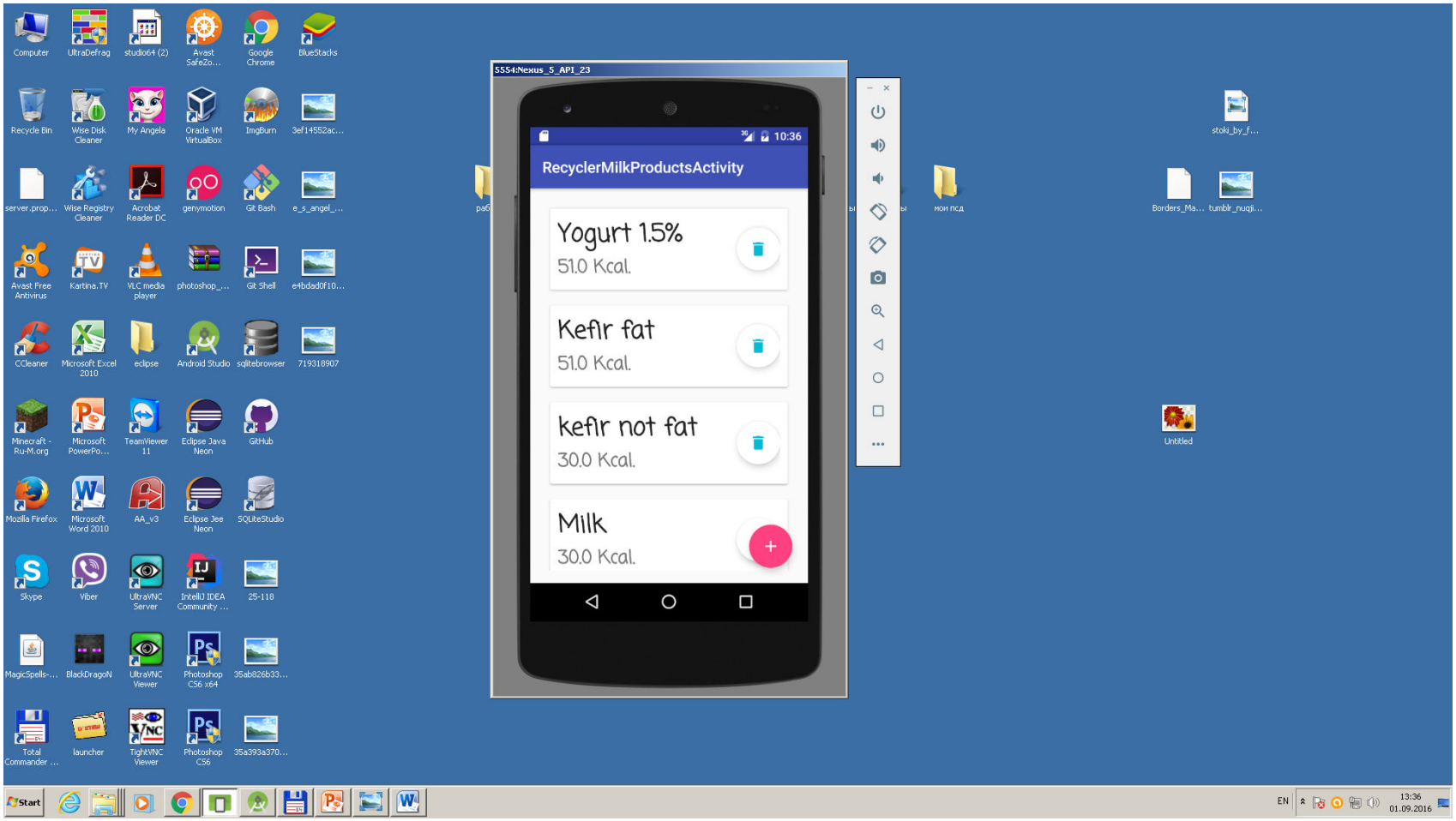
We define adapter using array from strings.xml productname_array. On ItemSeleted we use getItemAtPosition to open Activity that we need.



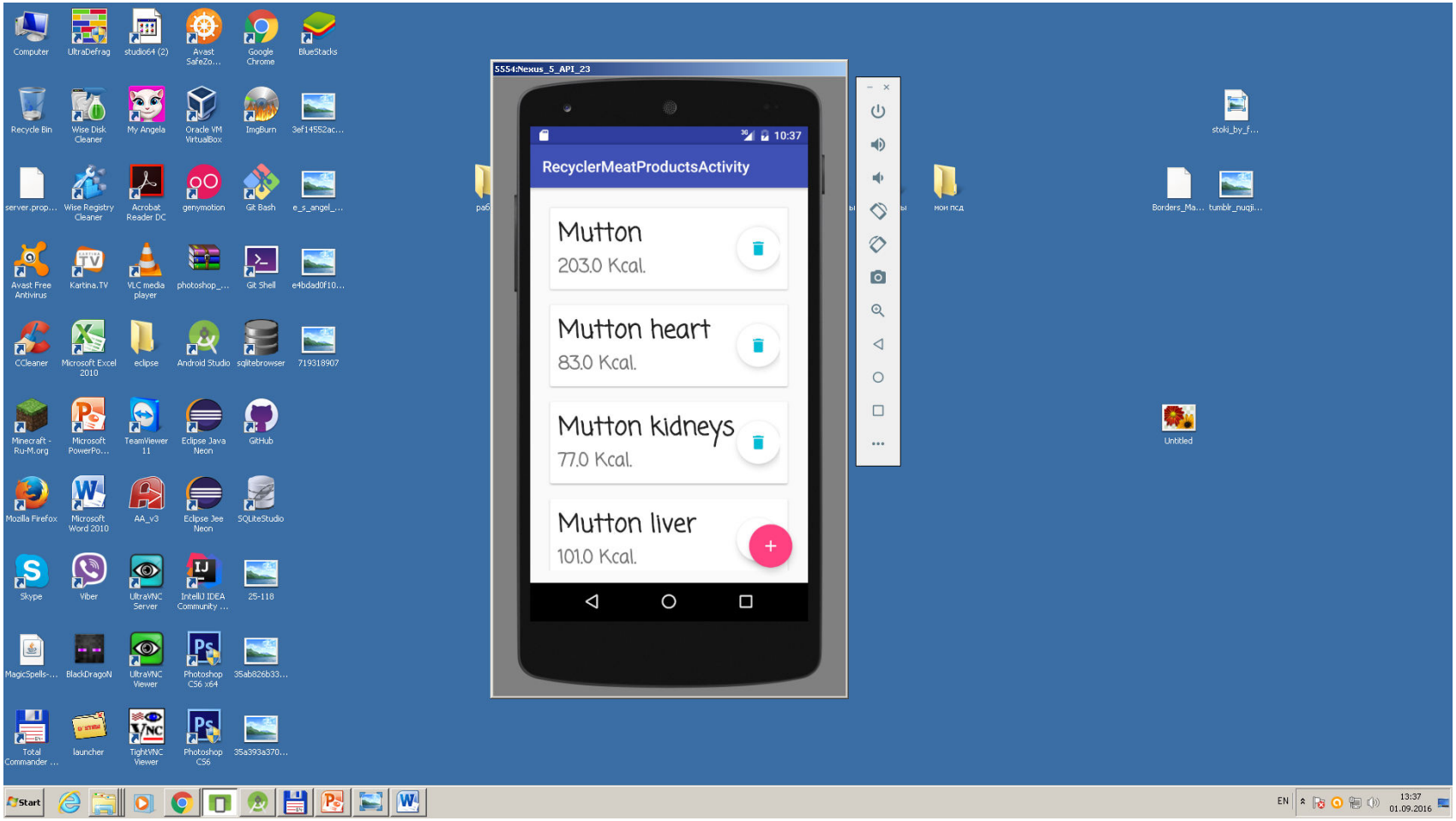
Choosing databases from spinner.



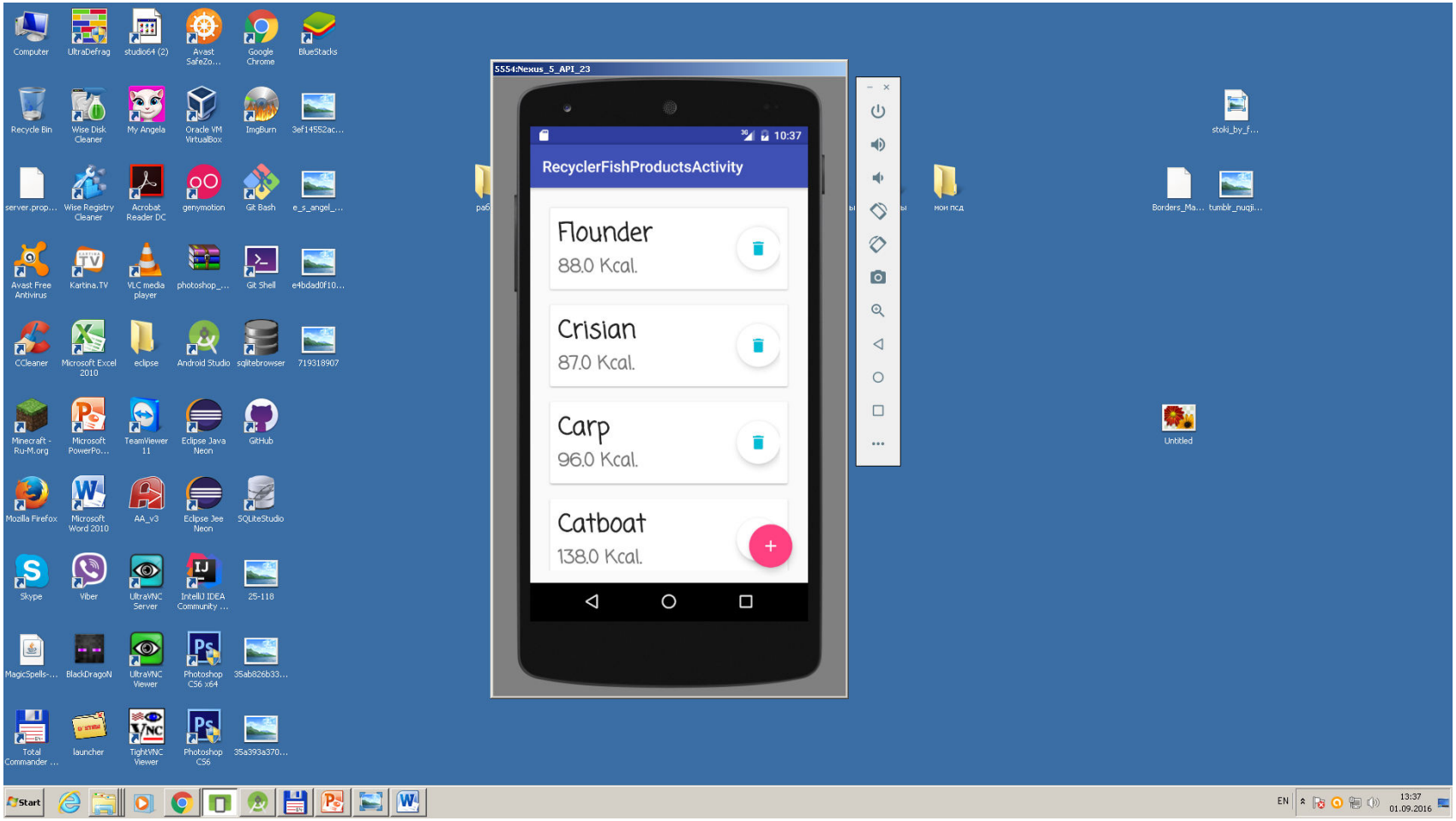
Database of Milk Products.



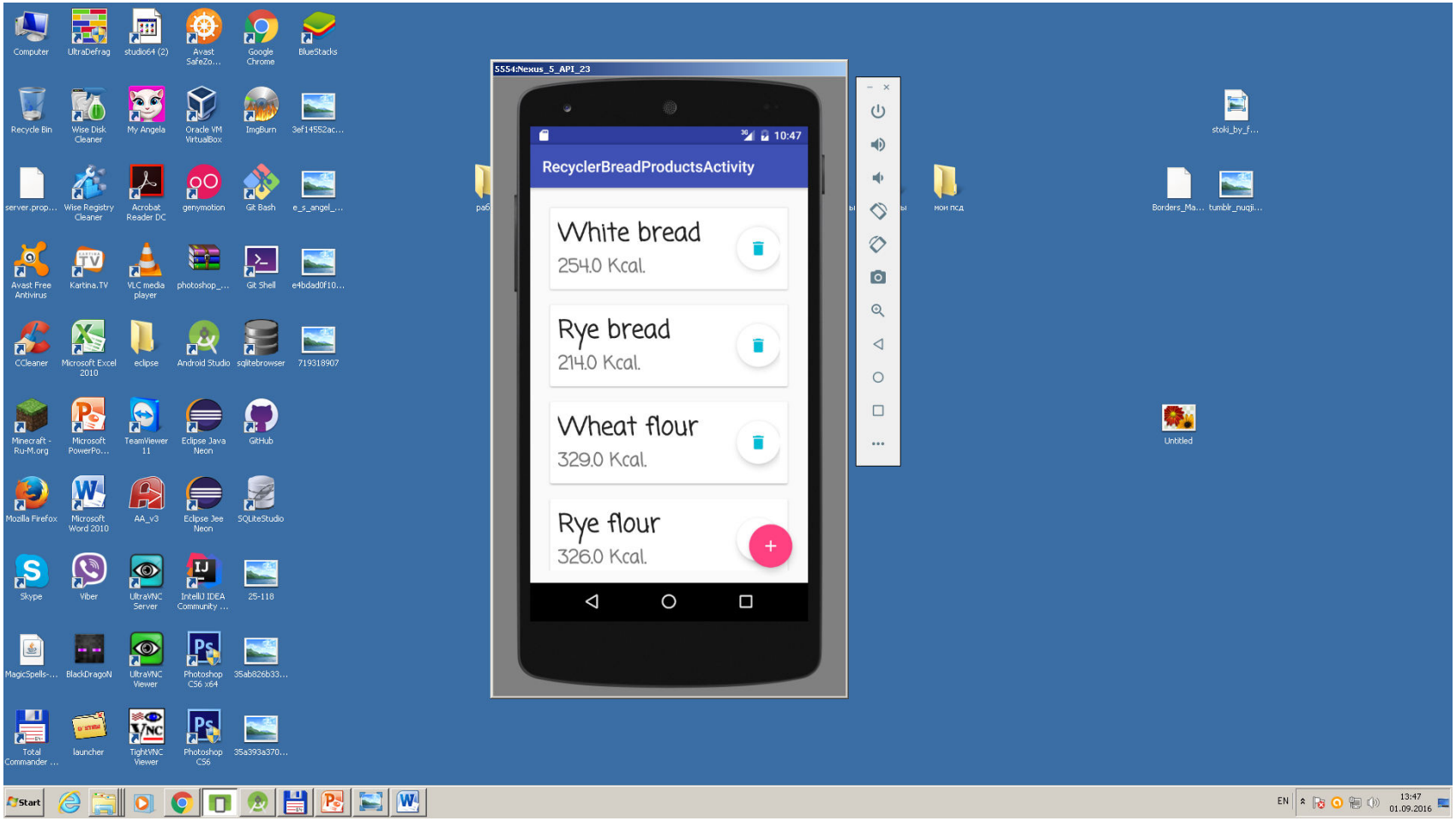
Database of Meat Products.



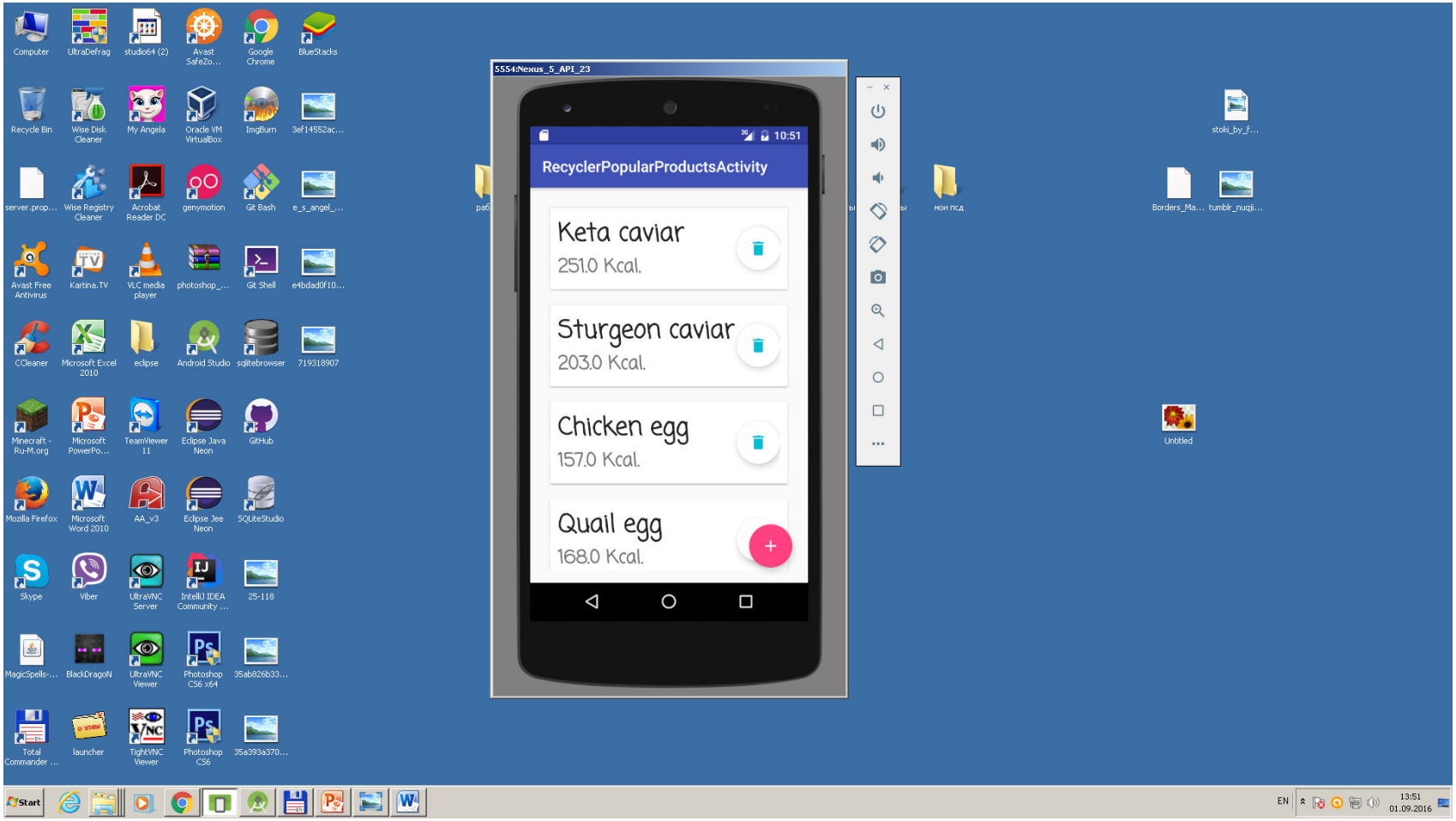
Database of Fish Products.



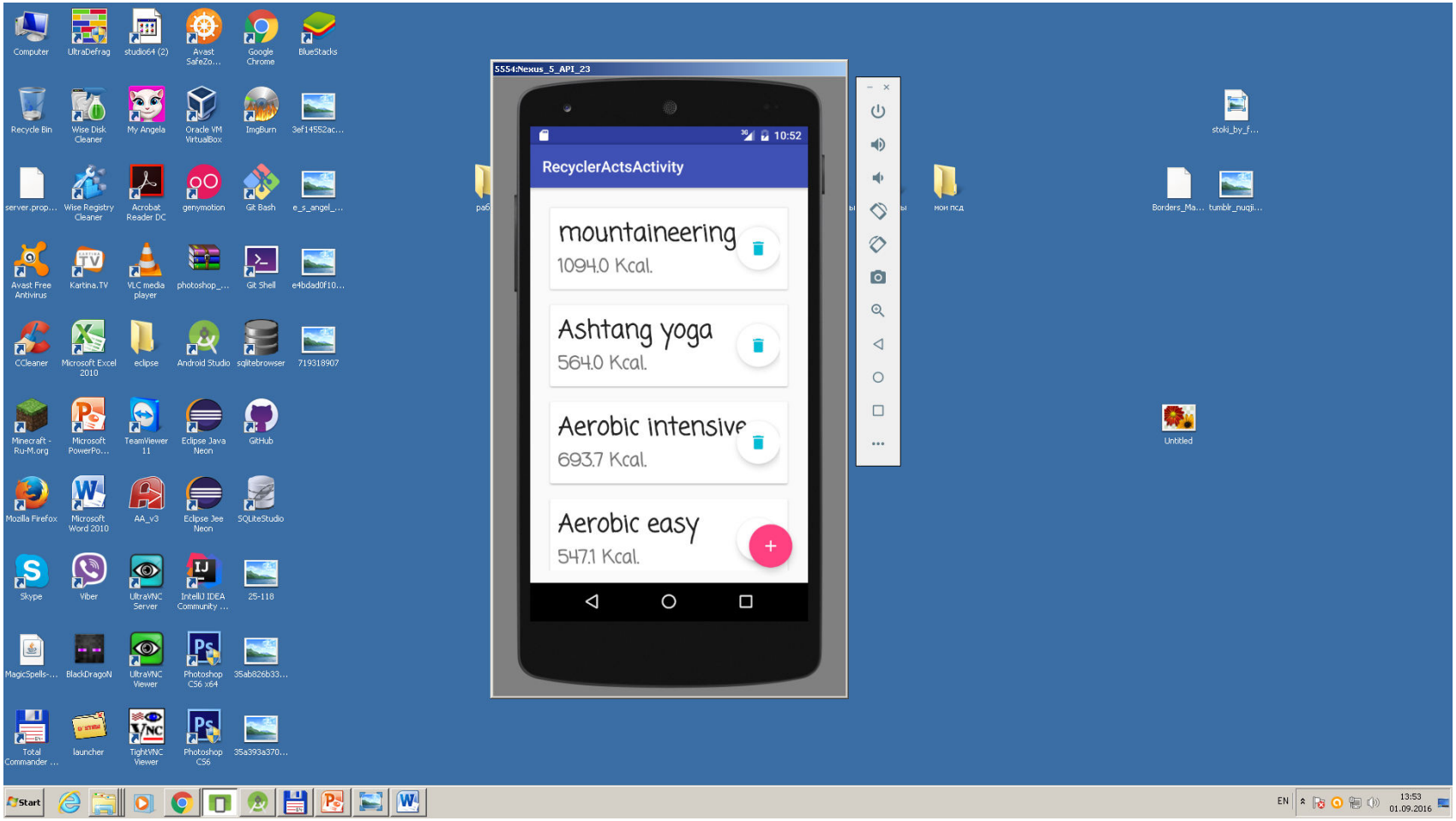
Database of Bread Products.



Database of popular products.



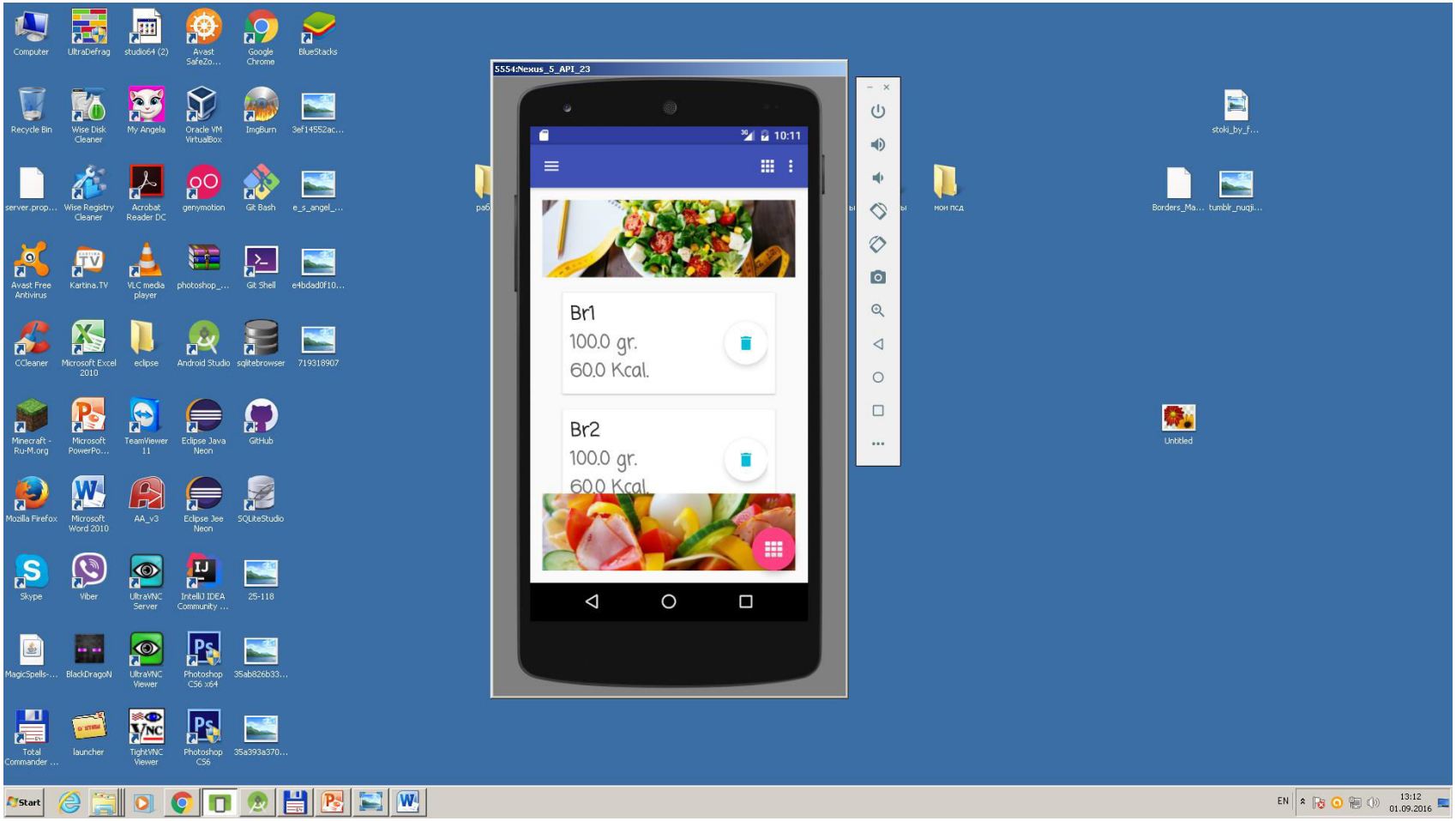
Database of popular products of user with recycler view.



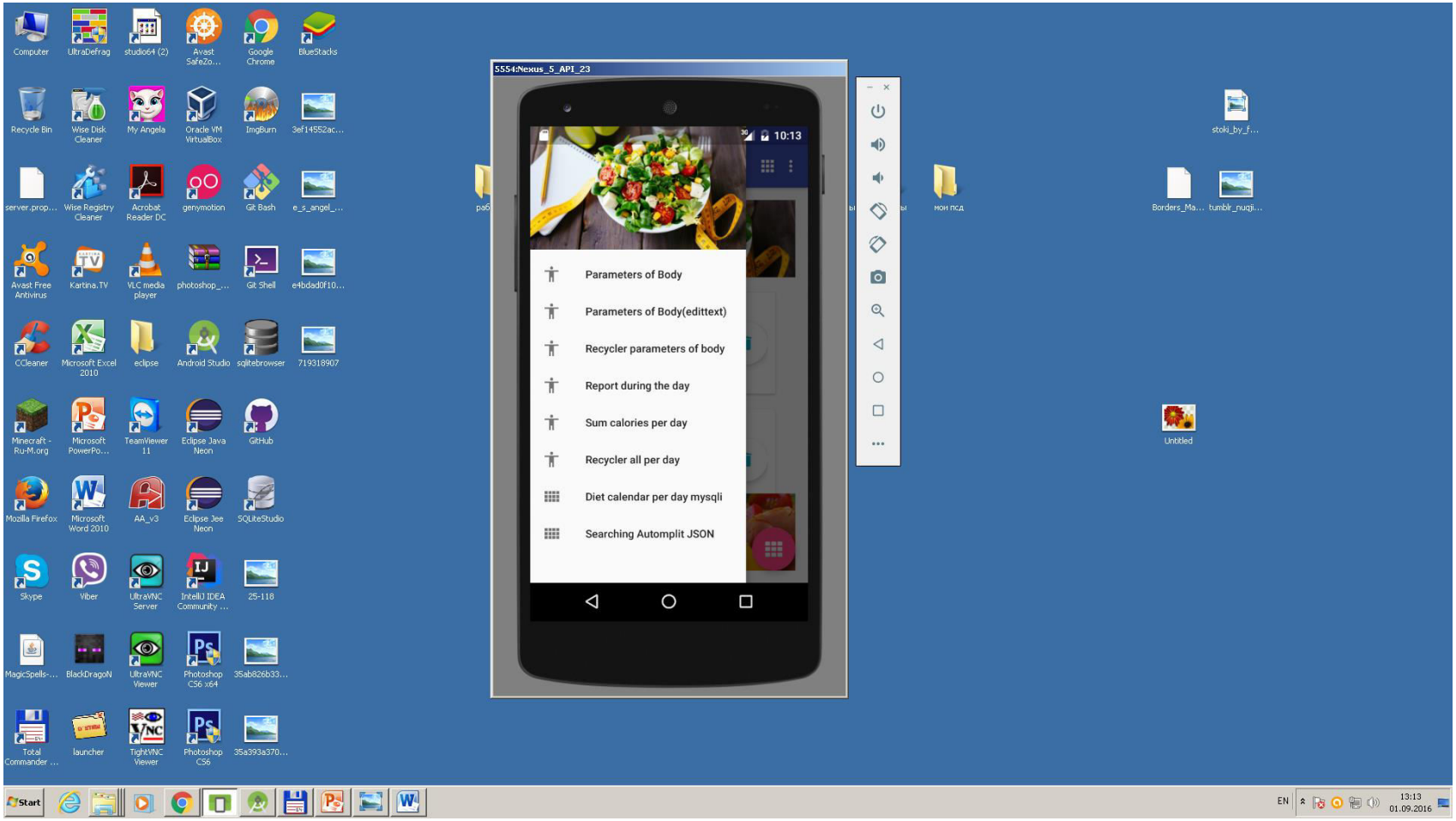
Continuation features of application.

- We use navdrawer with feature of 2 panels from the left and right, using fragments and recycleview elements for showing numbers of calories for breakfast, lunch, dinner, snakes and physical acts in main view of navdrawer.
- For inputting parameters of body we use 2 forms: one of them use spinners with adapters for adding elements to arrays, another input form using TextInputEditText with TextInputLayout.

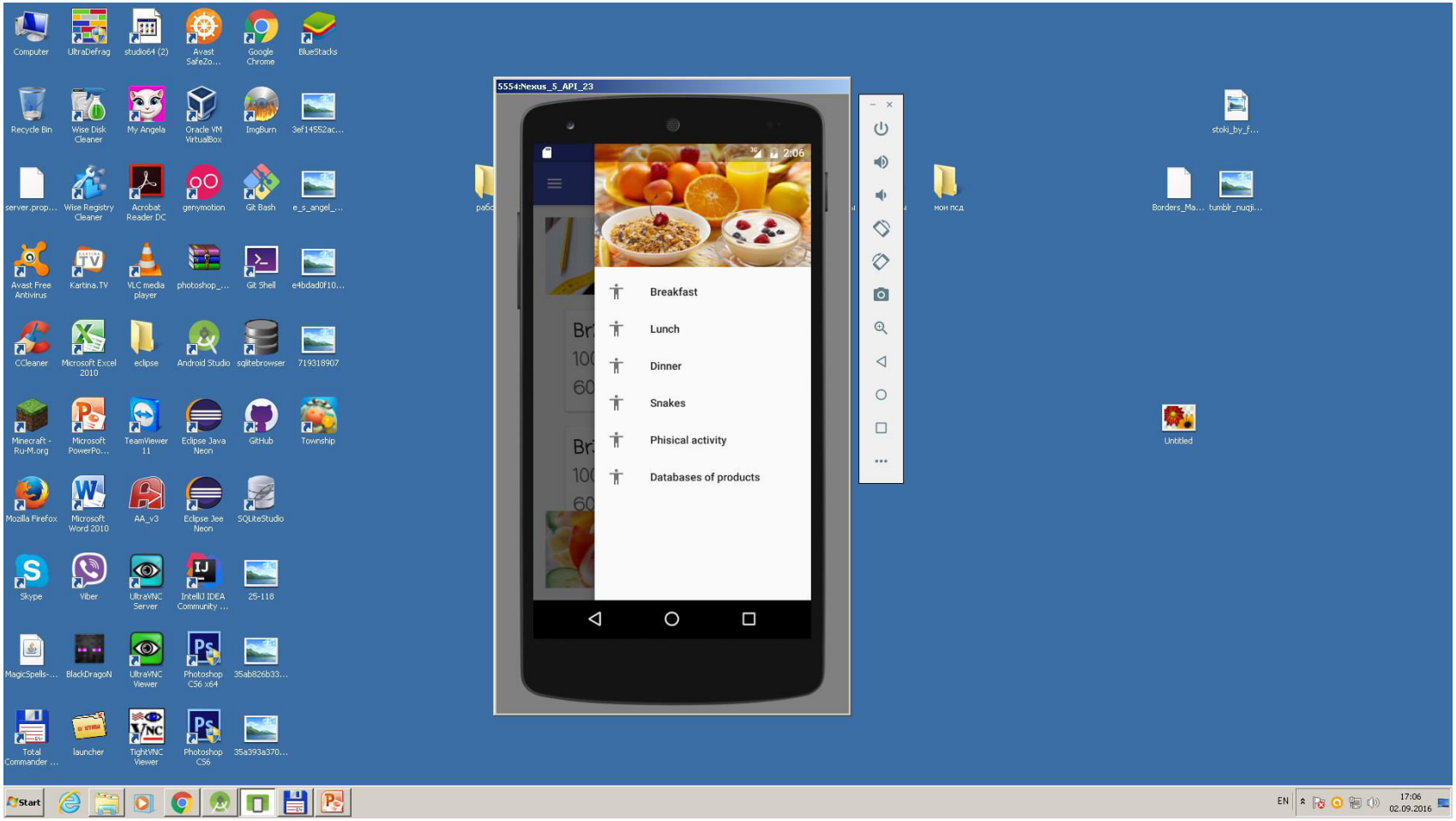
NavDrawing with two panels from the left side and from the right side.



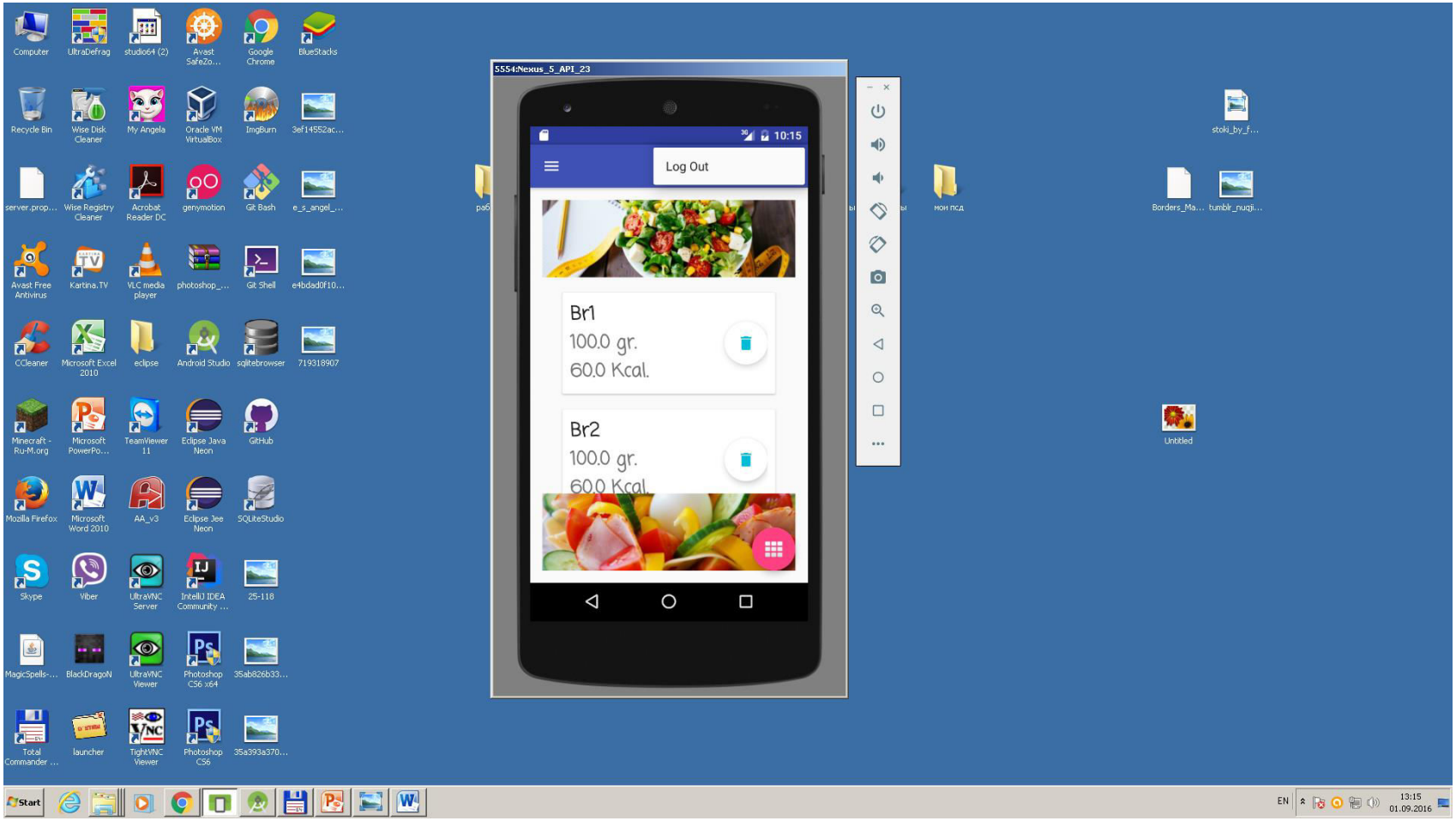
Navdraw with opened left panel.



Navdrawing panel with opened right panel.



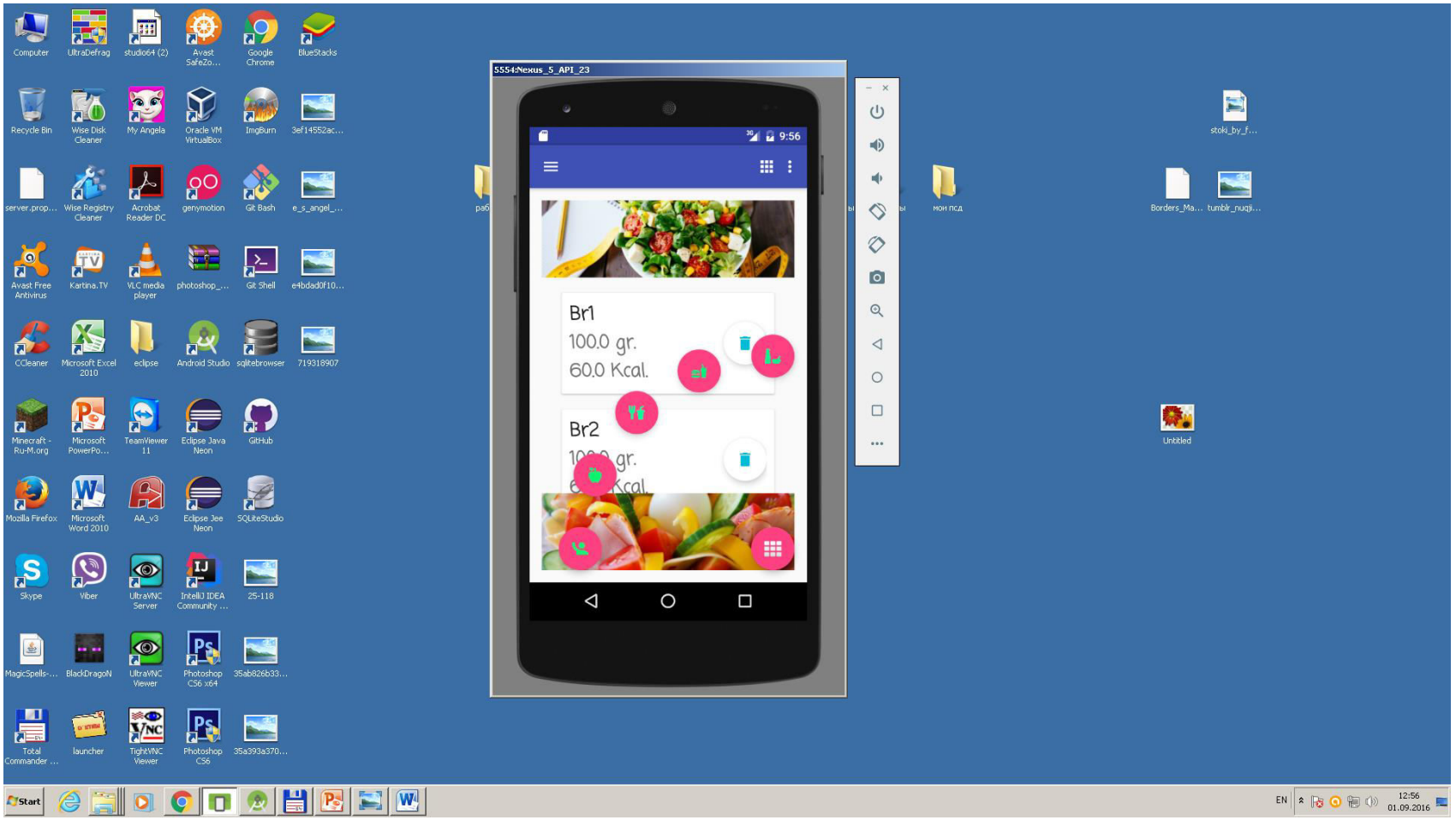
We left menu panel with logout from Firebase.



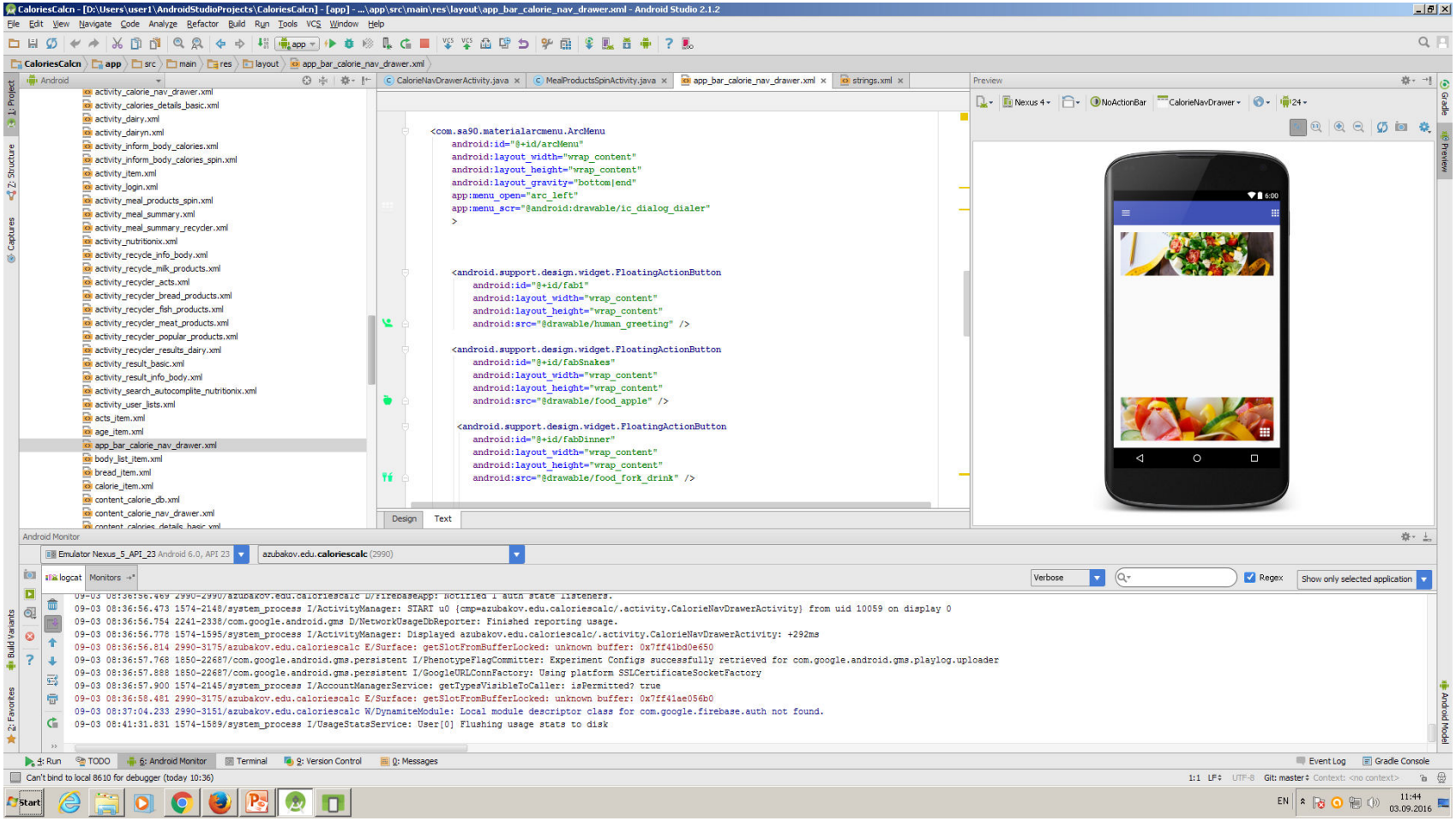
Features of input calories.

- We use element Arcmenu with 5 fab buttons for inputting number of calories during the breakfast, lunch, dinner and Snakes. We use here TextInputEditText with TextInputLayout.
- For inputting Physical Activities we use MaterialTextField with effect of animation with using of icon on opening element.

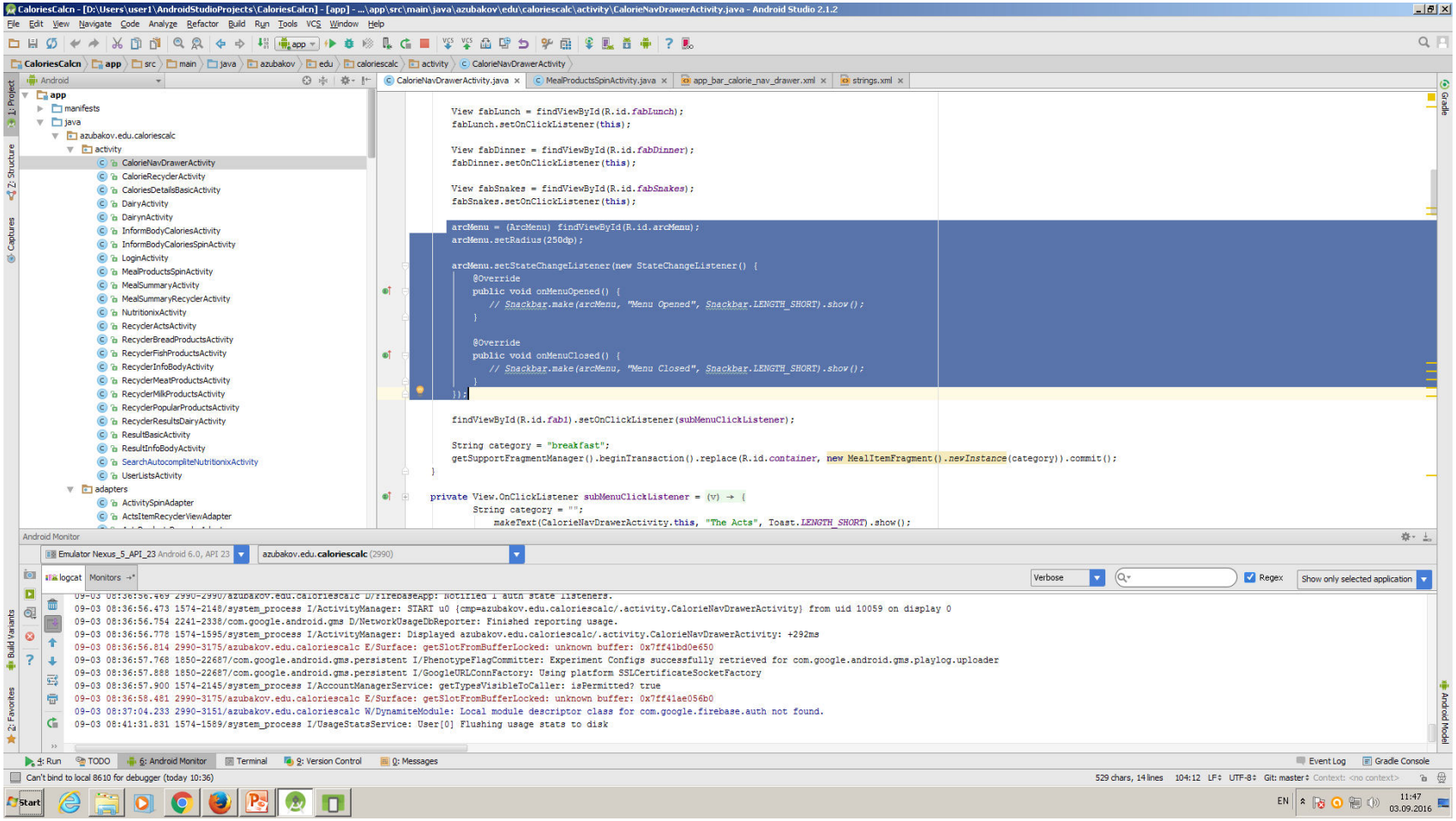
Element arcmenu that opens 5 fab buttons with animation.



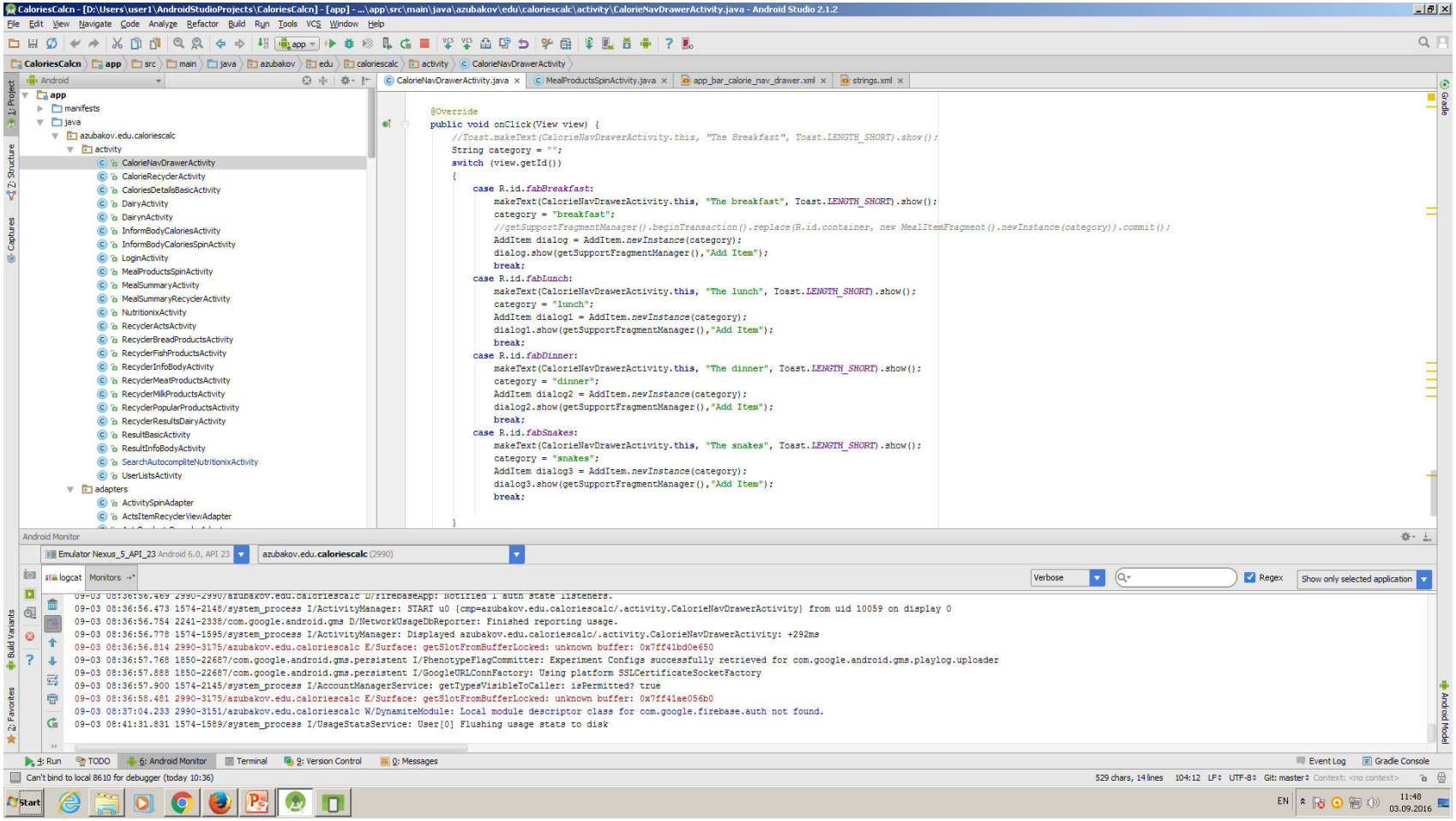
We define element Arcmenu with 5 fab buttons in file app_bar_calorie_nav_drawer.xml.



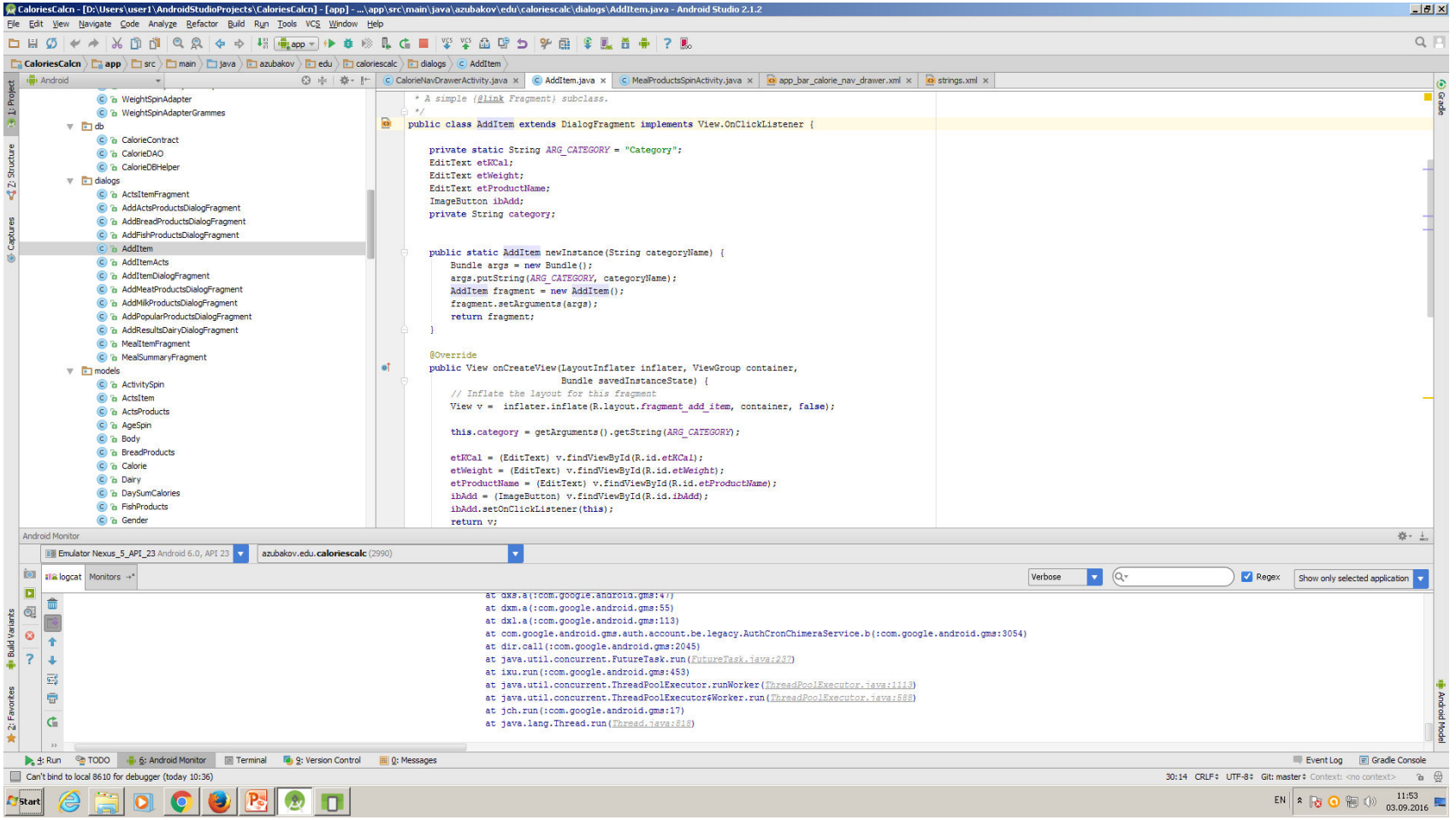
We register and define listener in file CalorieNavDrawerActivity.



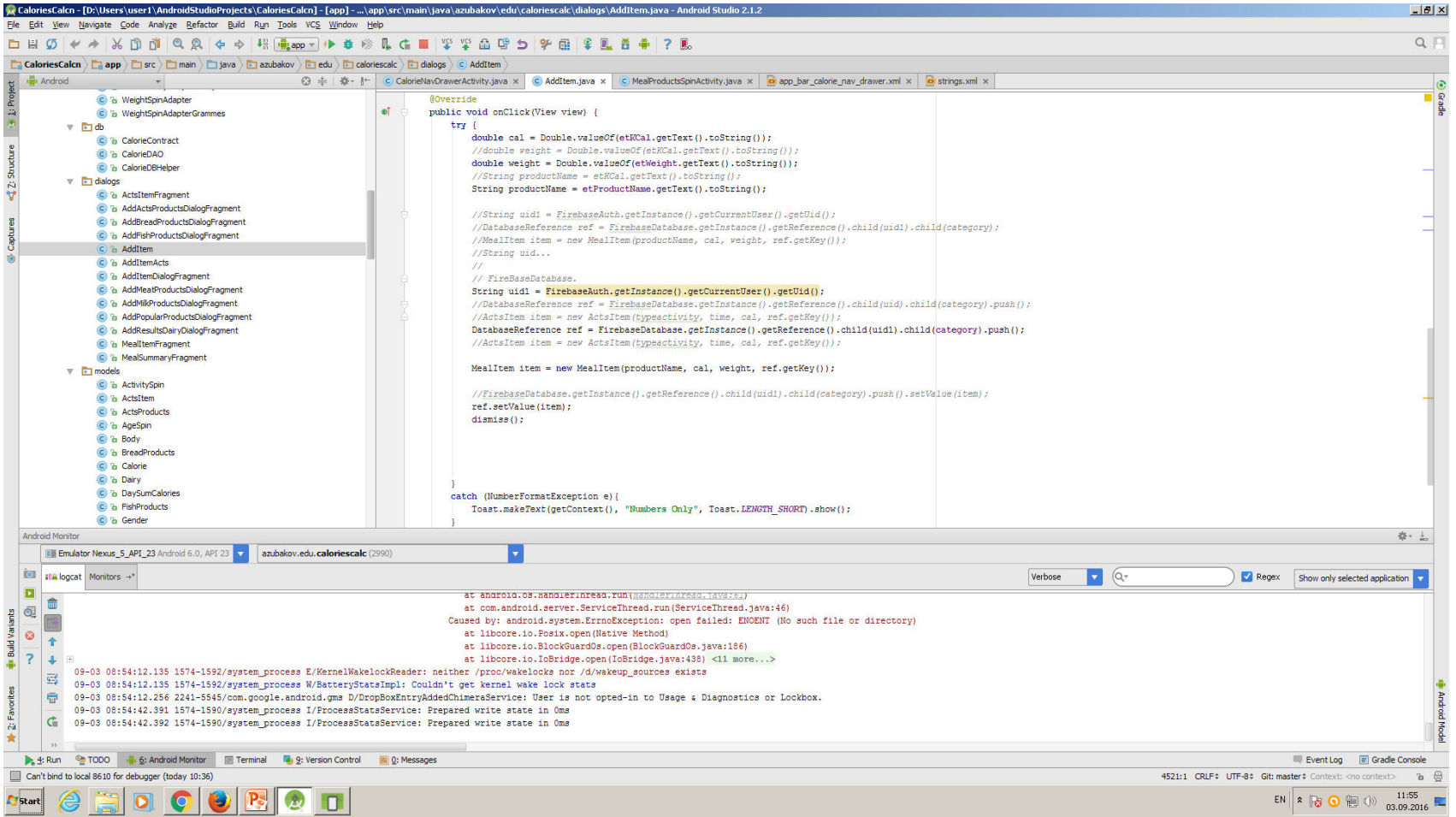
It depends on clicking on fab button we define opening addItem fragment.



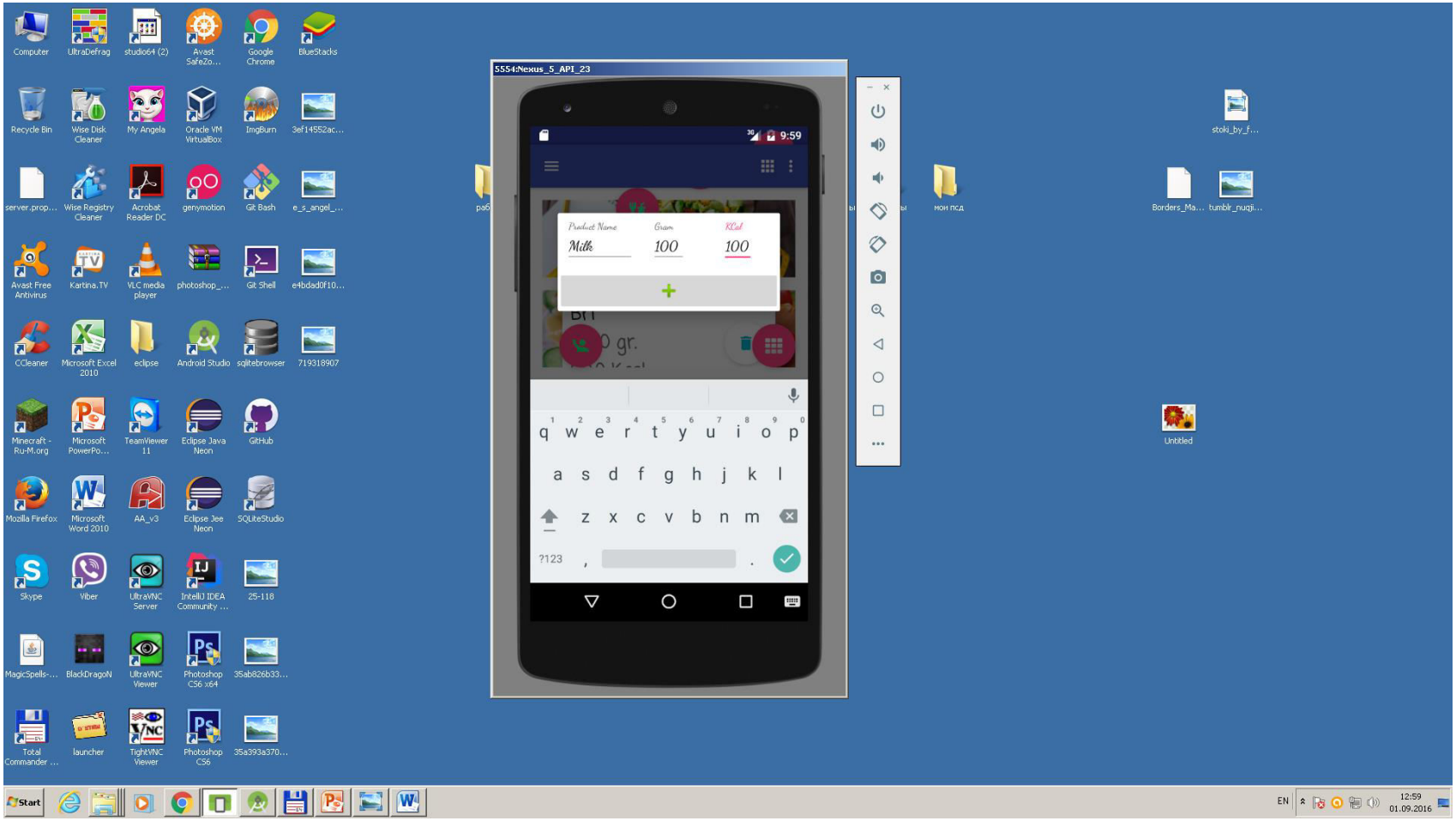
Fragment AddItem for adding and saving information to databse in Firebase.



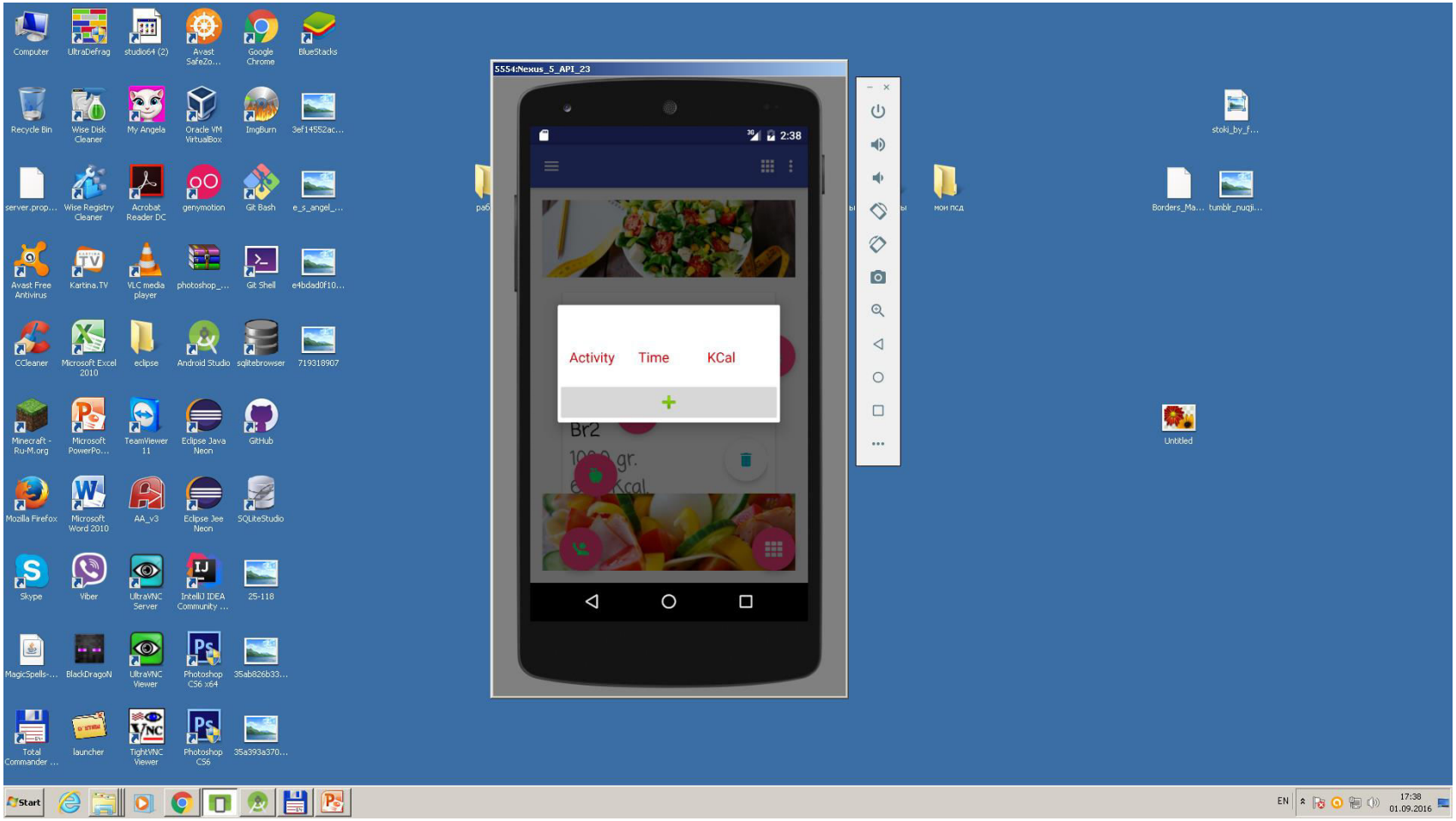
Fragment AddItem – we get reference to database and save information with ref.setValue using model class MenuItem.java



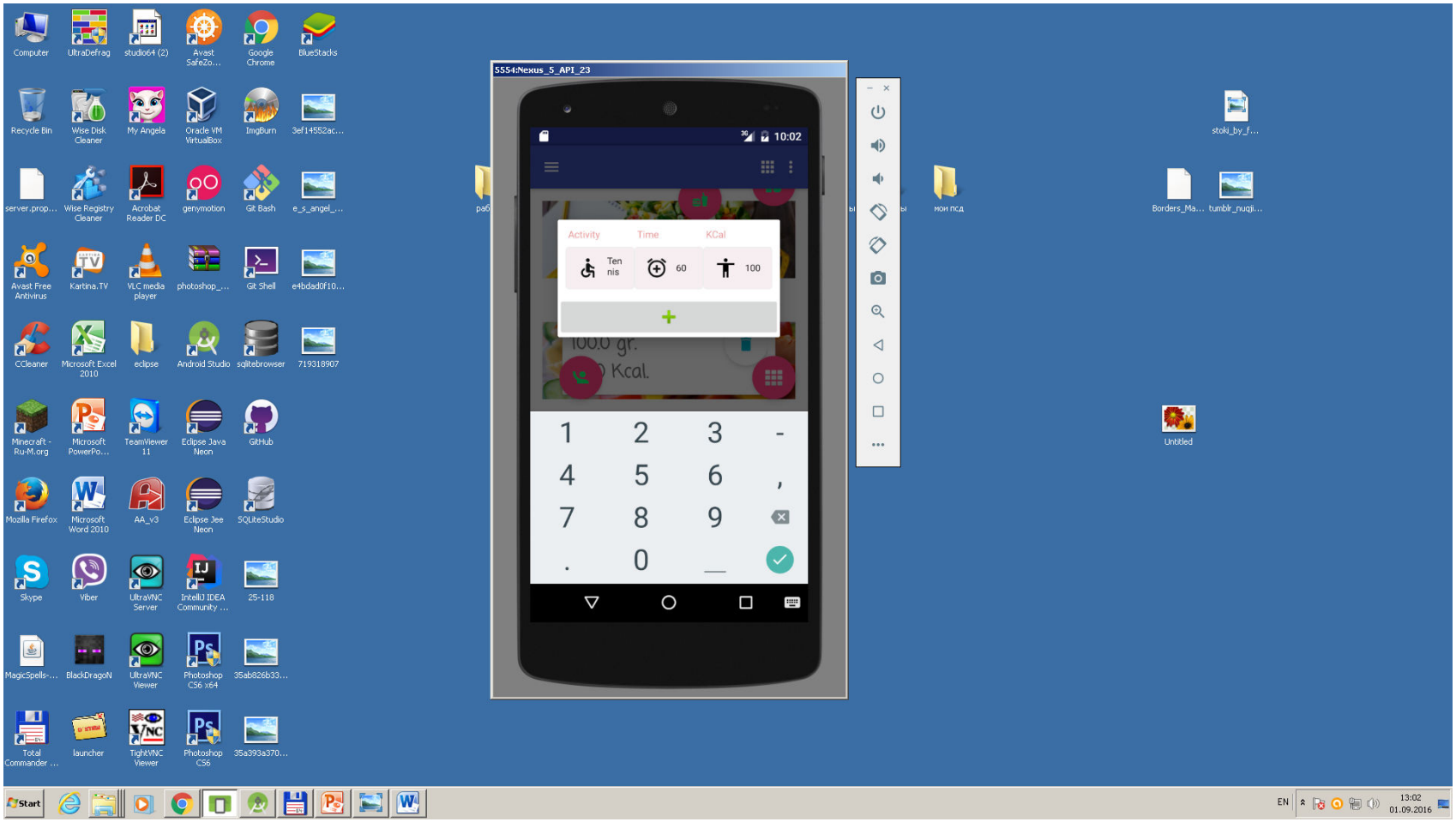
Using TextInputEditText with TextInputLayout.



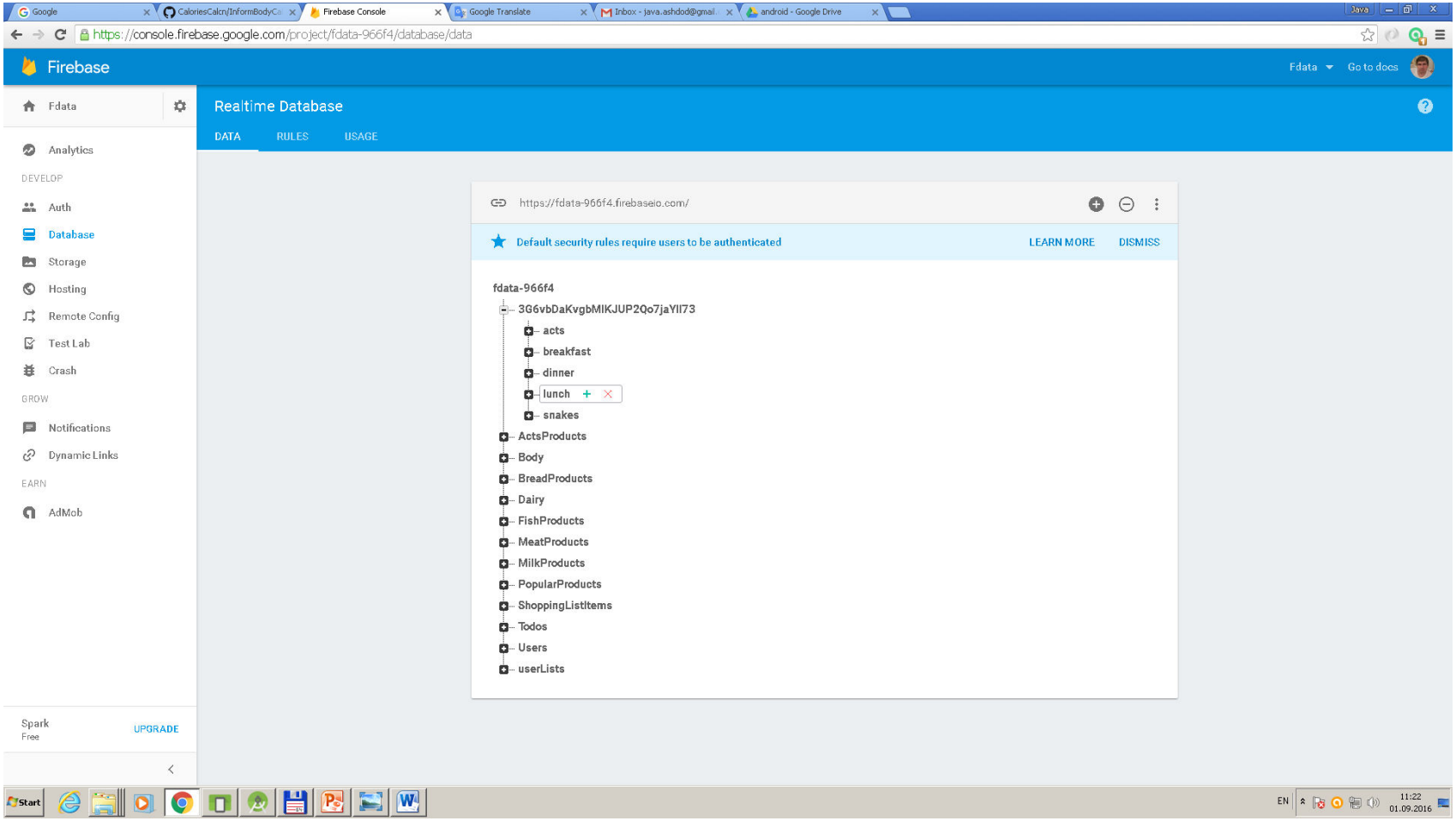
Using MaterialTextField with effect of animation with icon. It is not opend for inputting.



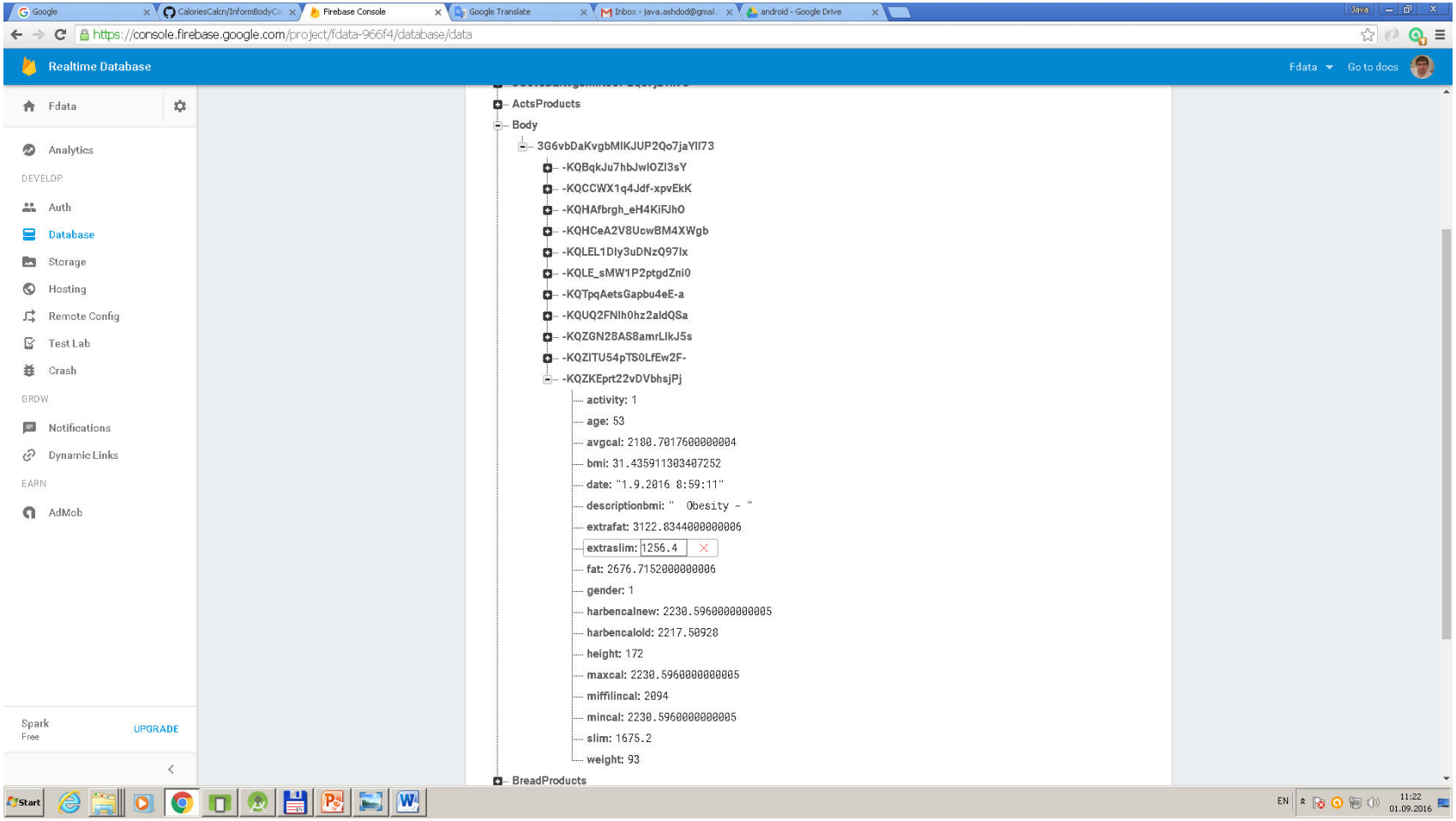
Using MaterialTextField with effect of animation with icon on opening element.



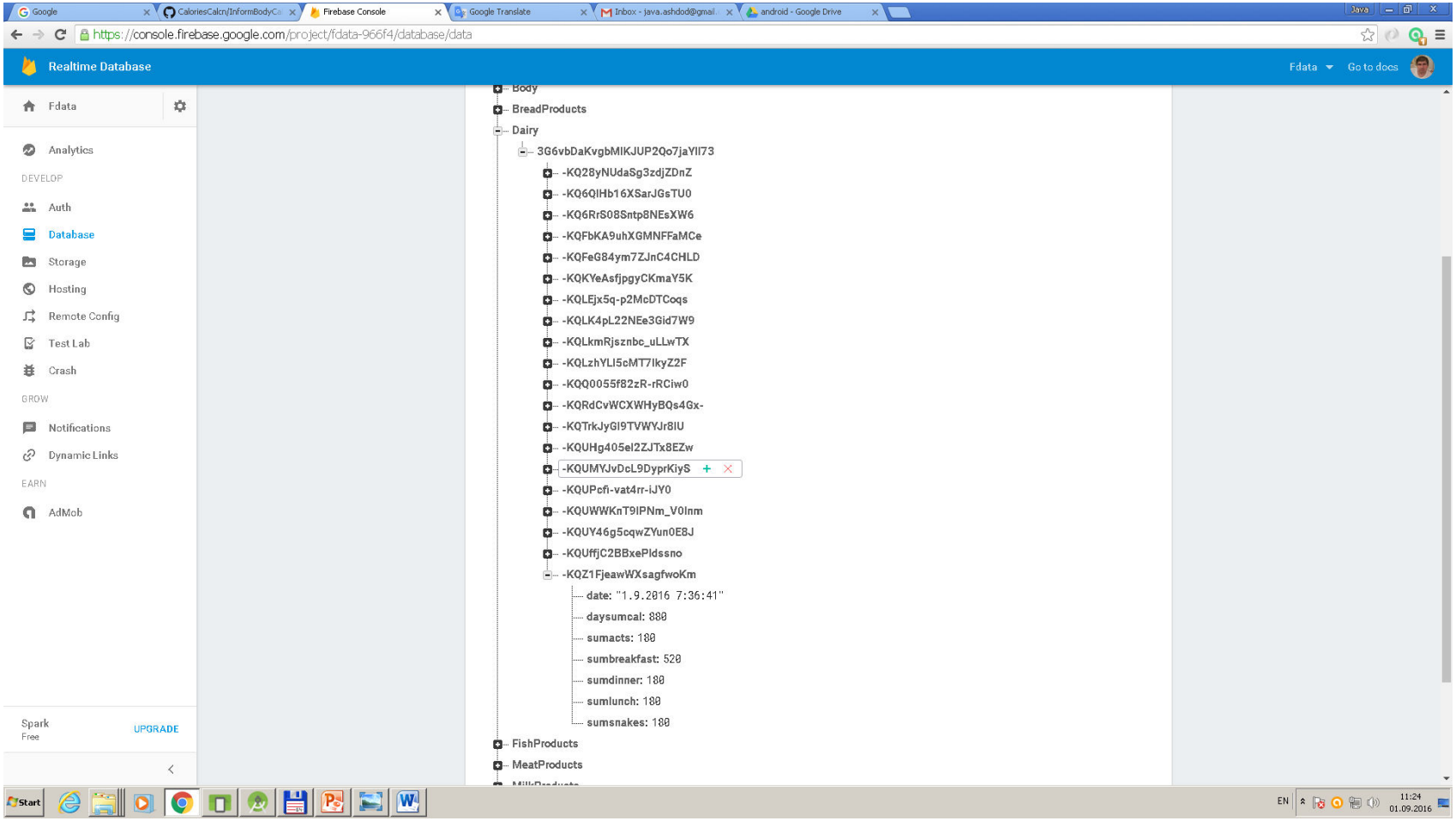
Structure of Firebase.



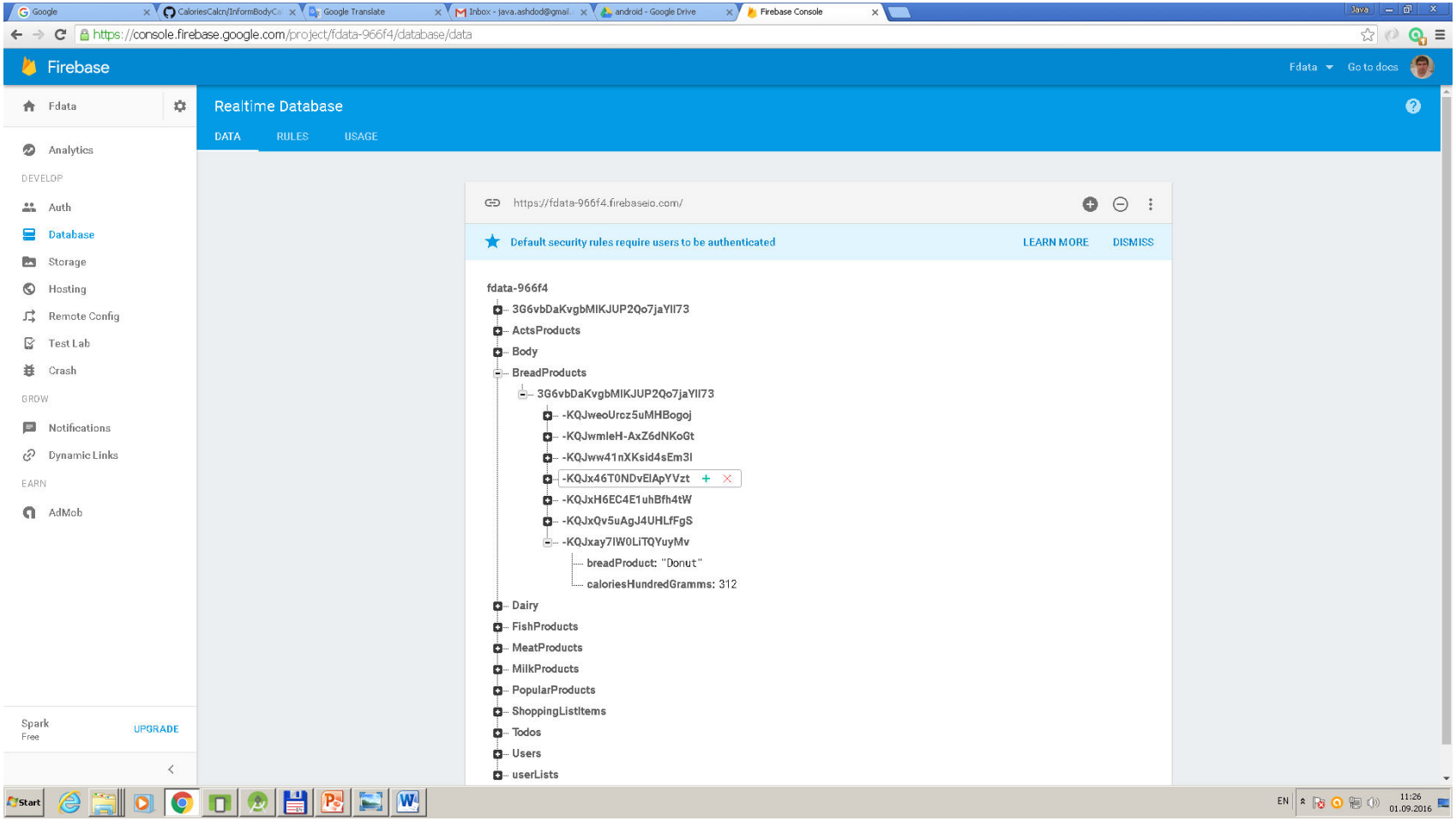
Structure of body's database.



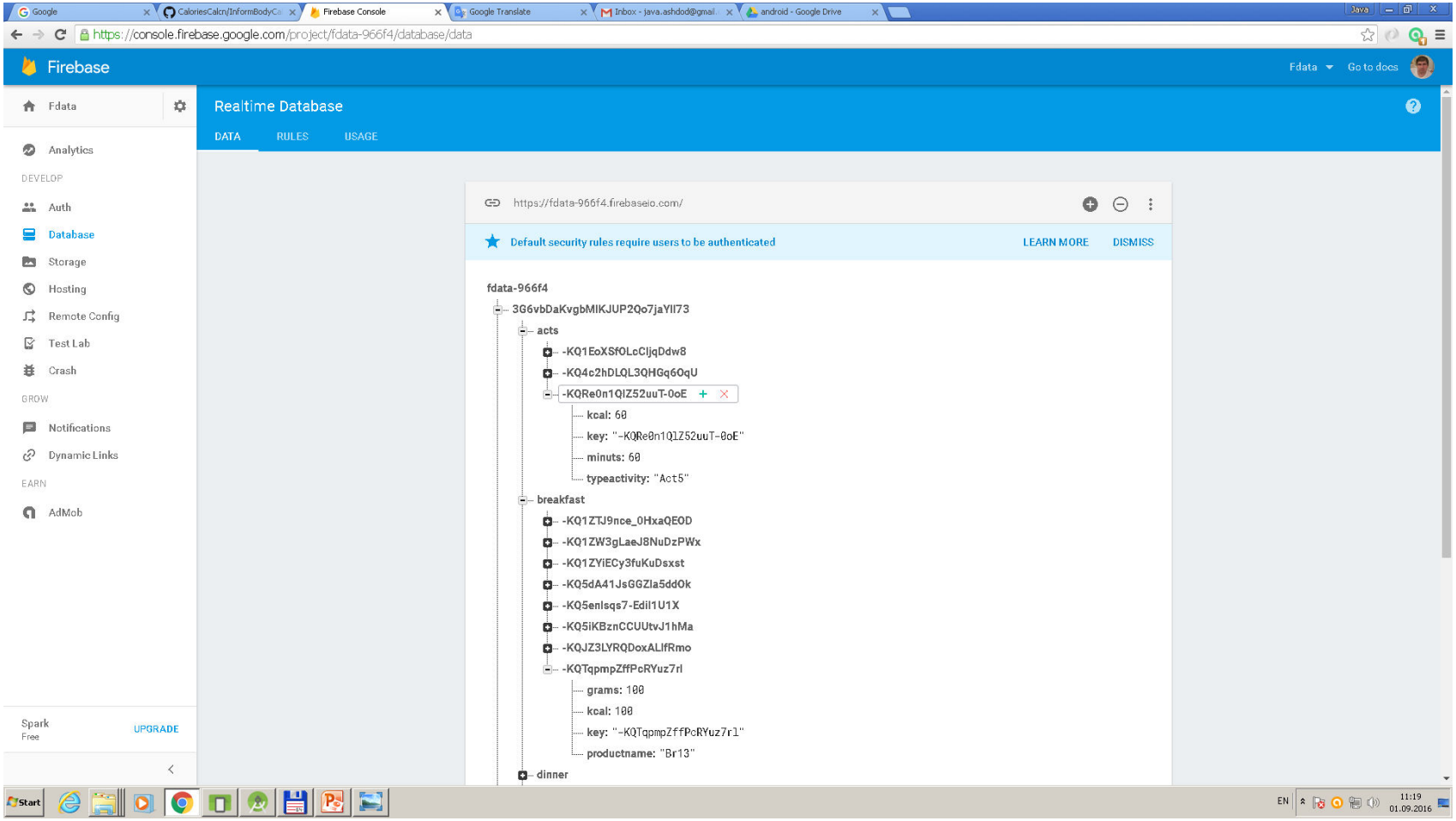
Structure of dairy (history) database.



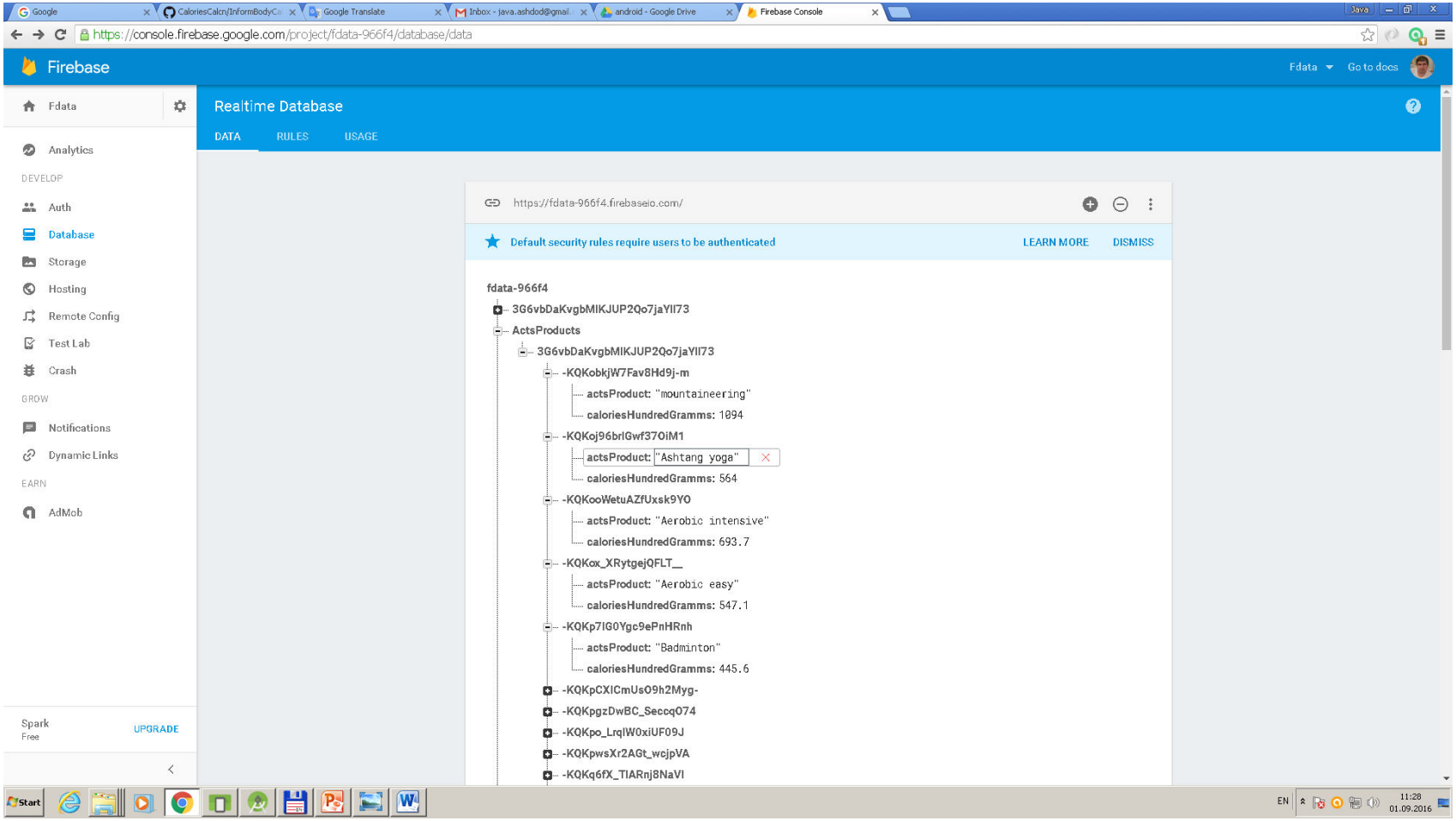
Structure of Breadproducts database.



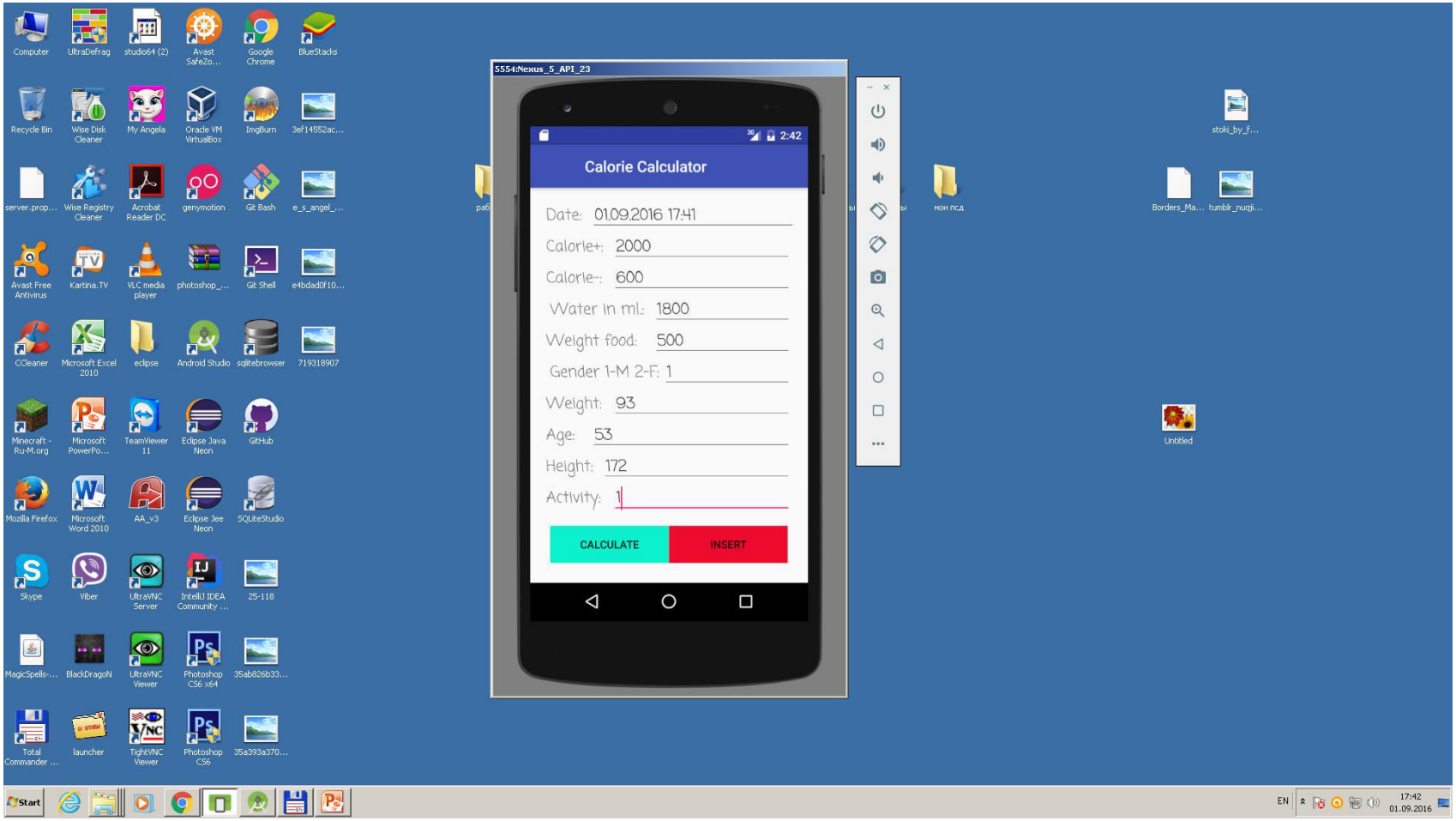
Continuation structure of Firebase.



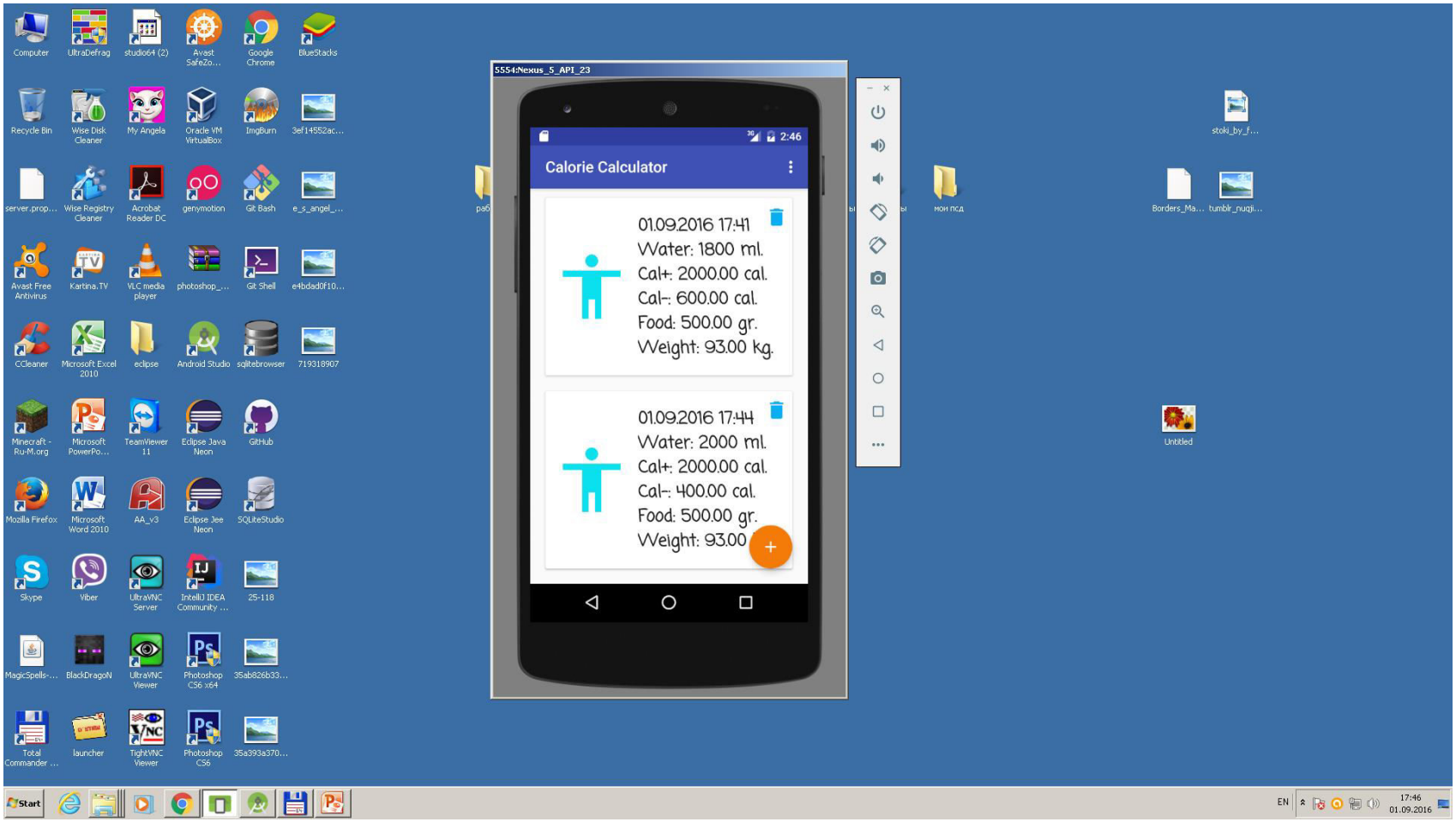
Structure of physical activities database.



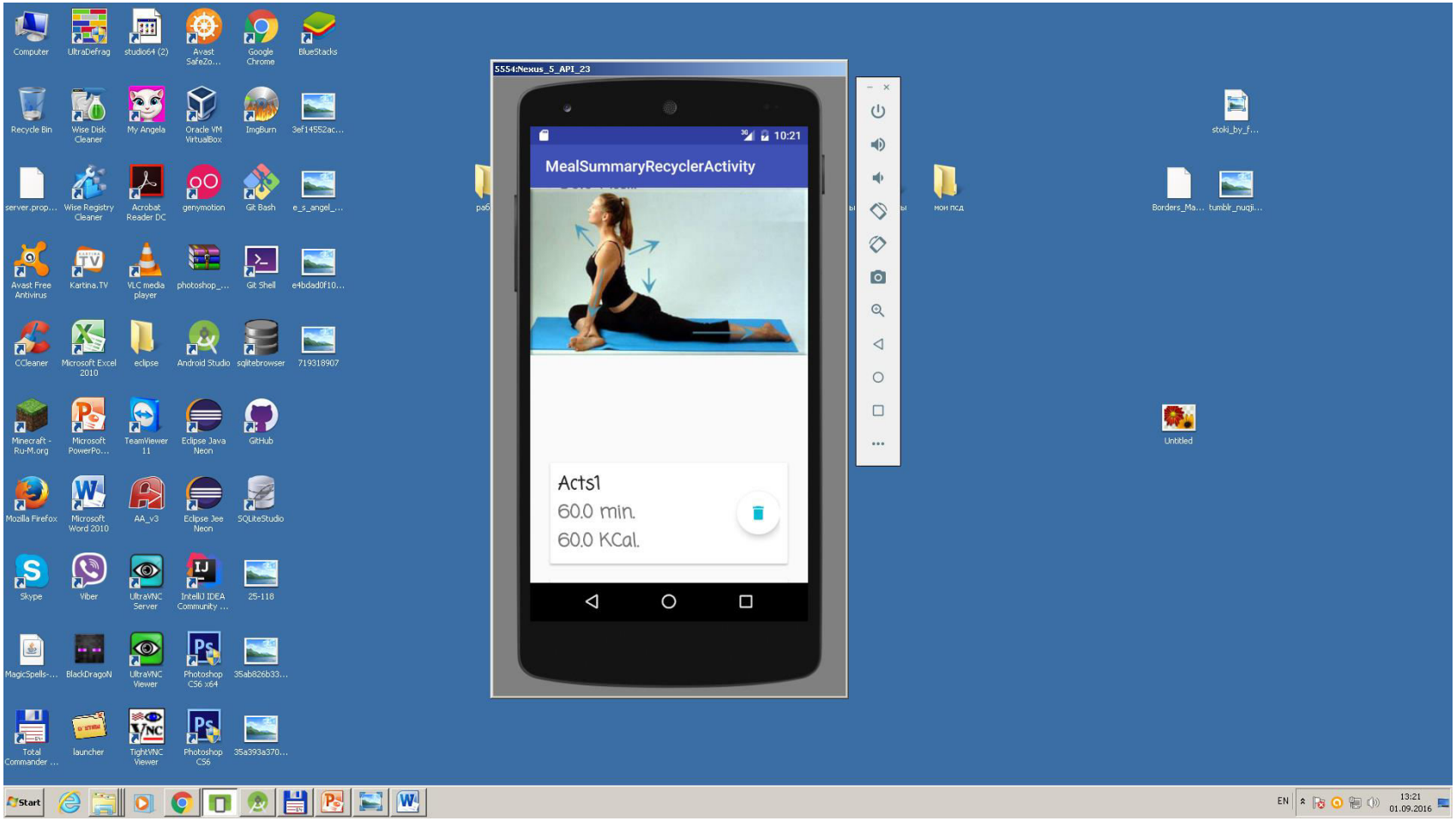
Using of mysqli for calculating calories. We input data and press button Insert.



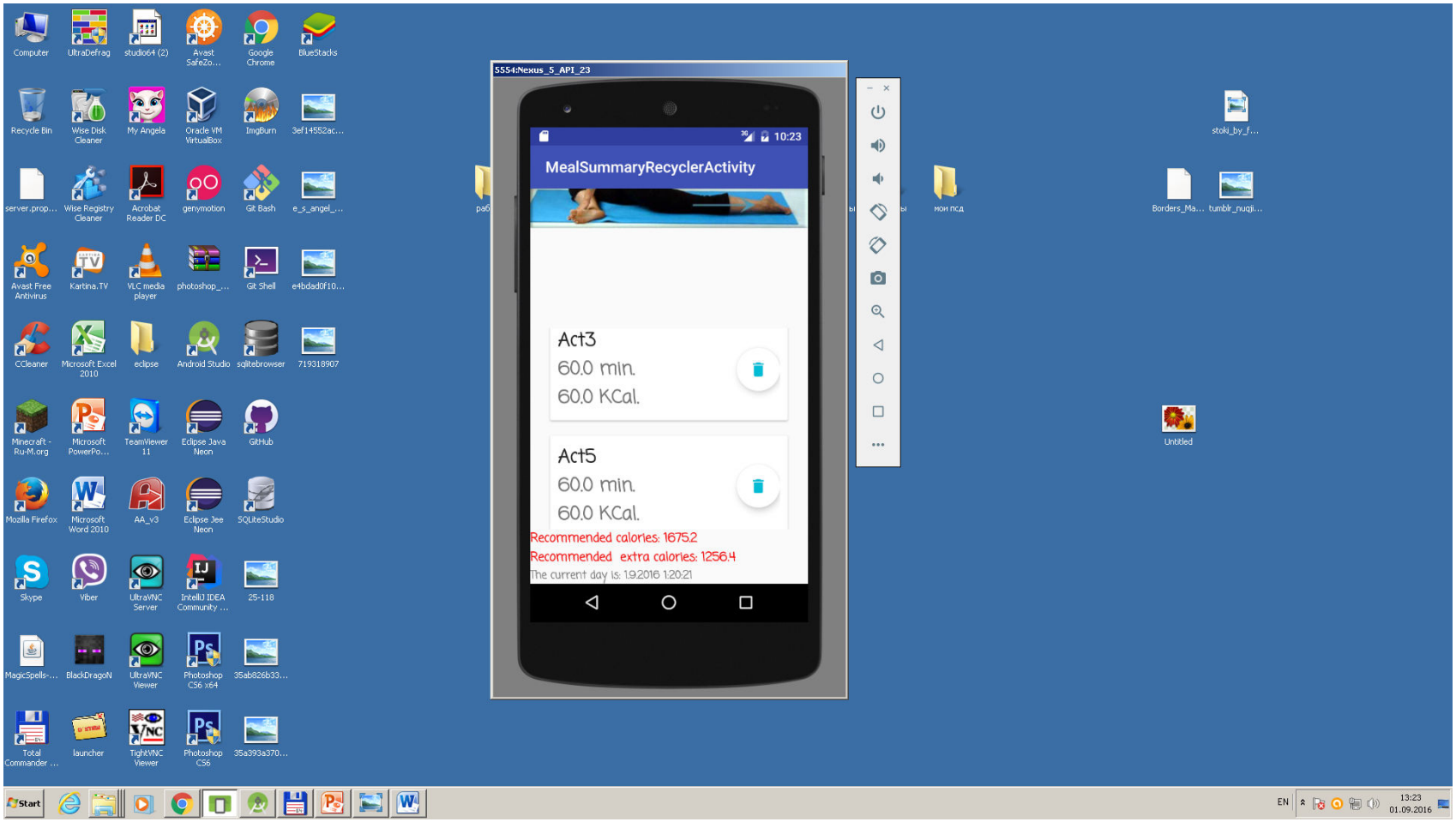
We get information from Local databases. It is for case that something wrong with Internet or cloud Firebase.



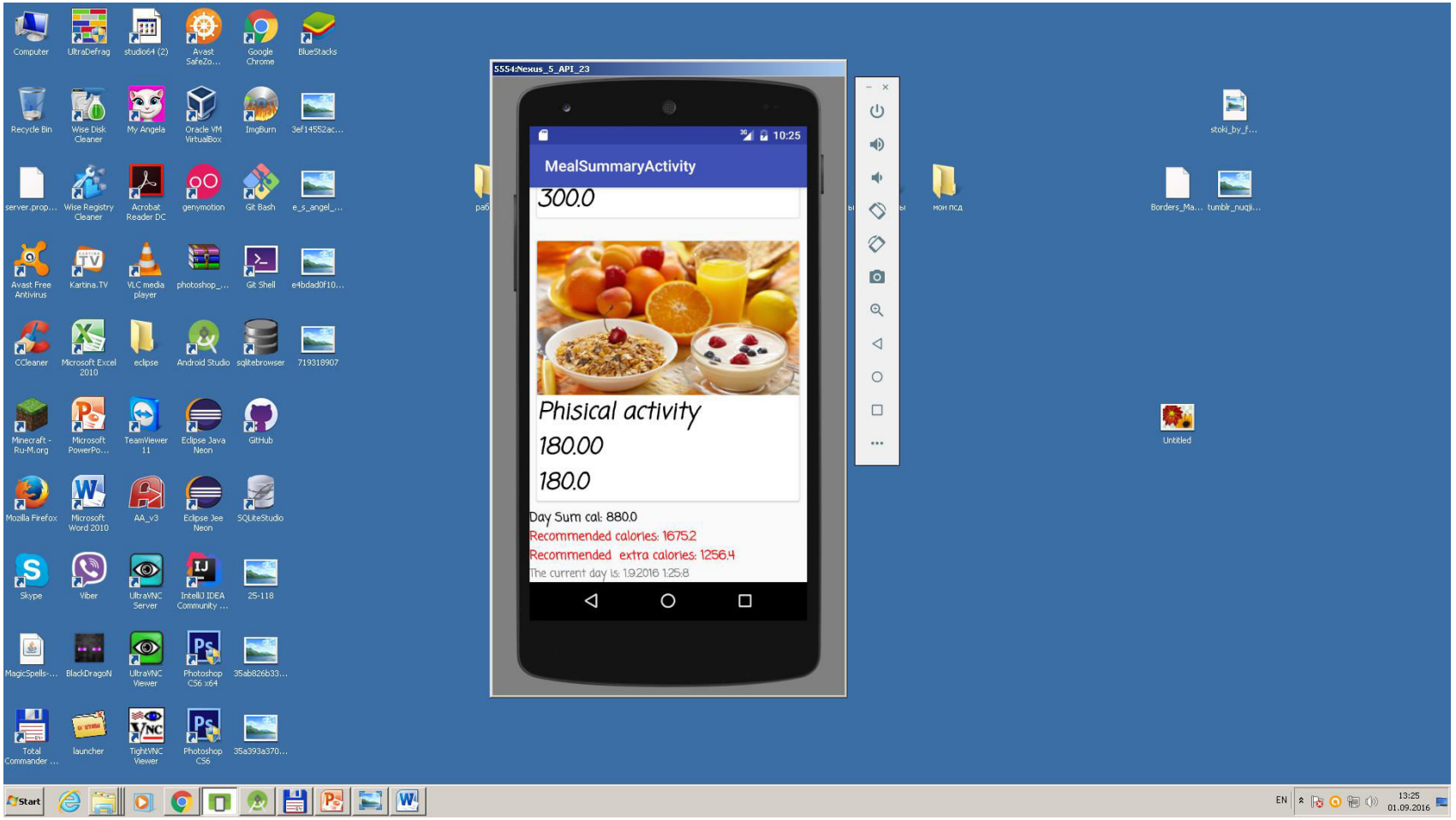
Results: we get recycler view of calories and report during the day.



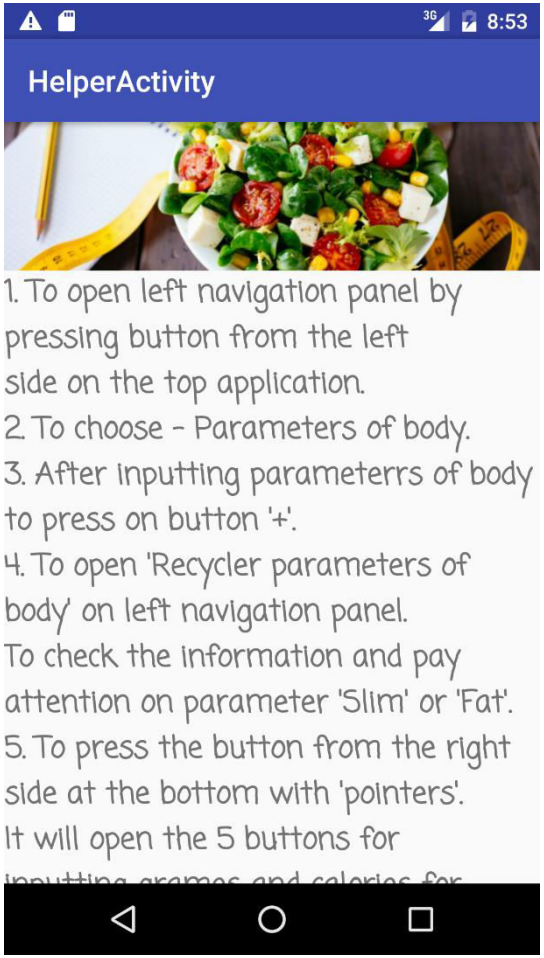
Recycler view with scrolling of nested scrolview element and recommended calories.



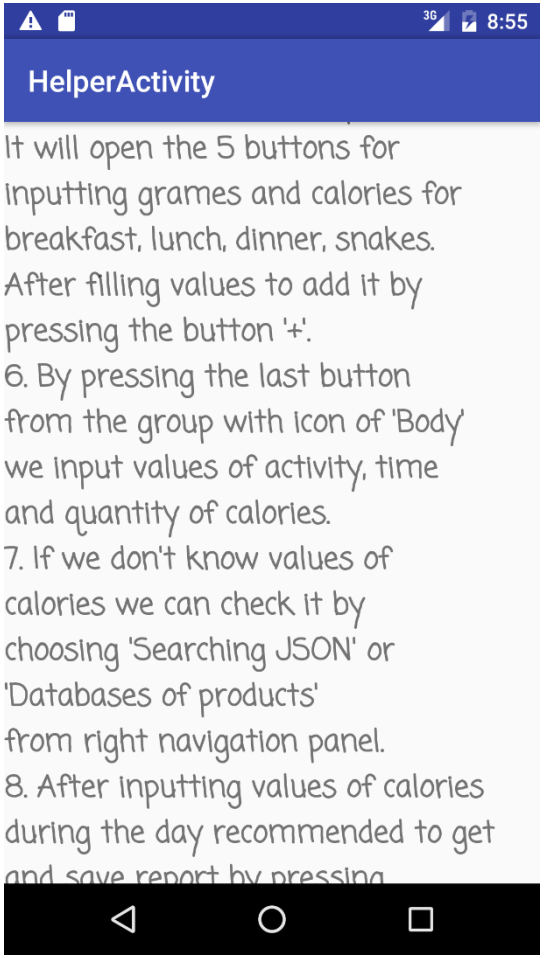
Report according to meals, activity and calculated Day Sum calories compared with recommended calories.



Helper for user.



Helper for user.



Helper for user.

Analyze results.

- So in the end of project we achieve the situation that each person that using the application can slim or gain weight. It will improve the health of person and avoid many diseases.
- The application is situated in github:

https://github.com/azubakov/CaloriesCalcn