COM4501 Android Operating System for Mobile Devices YAHYA BATURAY SARAÇOĞLU – 18290121

App Fit

About the application

The Main Idea:

It is 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:

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();
}</pre>
```

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);
}</pre>
```

DBAdapter.java > The class I created for database adaptation

Creating database table for user:

```
db.execSQL("CREATE TABLE IF NOT EXISTS food (" +
        " 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_thumb VARCHAR," +
        " food_image_a VARCHAR," +
        " food_image_b VARCHAR," +
        " food_image_c VARCHAR," +
        " food_language VARCHAR," +
        " food_notes VARCHAR);");
```

Creating database table for foods

Creating database table for calorie values for eaten foods:

Function to create food categories

Categorizing food

```
public void insertAllCategories() {
    setupInsertToCategories("NULL, 'Bread', '9', '', NULL");
    setupInsertToCategories("NULL, 'Bread', '1', '', NULL");
    setupInsertToCategories("NULL, 'Cereals', '1', '', NULL");
    setupInsertToCategories("NULL, 'Cereals', '1', '', NULL");
    setupInsertToCategories("NULL, 'Frozen bread and rolls', '1', '', NULL");
    setupInsertToCategories("NULL, 'Drinks', '0', '', NULL");
    setupInsertToCategories("NULL, 'Bakang', 'o', '', NULL");
    setupInsertToCategories("NULL, 'Biscuit', 'o', '', NULL");
    setupInsertToCategories("NULL, 'Sodar', '9', '', NULL");

    setupInsertToCategories("NULL, 'Sodar', '9', '', NULL");

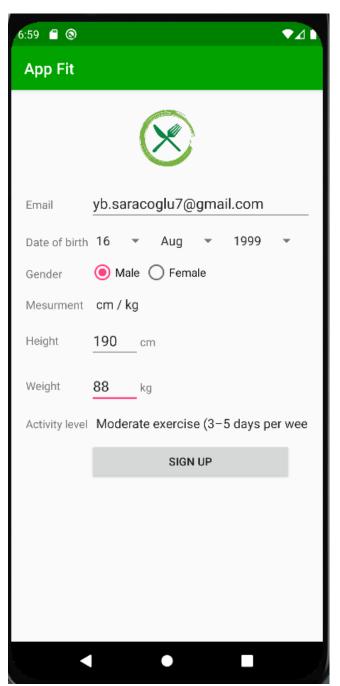
    setupInsertToCategories("NULL, 'Frozen fruits and vegetables', setupInsertToCategories("NULL, 'Canned fruits and vegetables', '1', '', NULL");
    setupInsertToCategories("NULL, 'Canned fruits and vegetables', '60.open();
    db.dapter db = new DBAdapter(context);
    db.open();
    db.new('1d, food_name, food_manufactor_name, food_serving_size_gram, food_serving_values);
    db.close();
}

Function to create food categories:

}
```

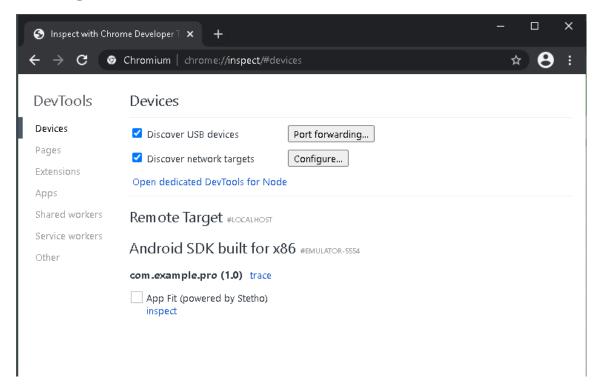
GUI

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.

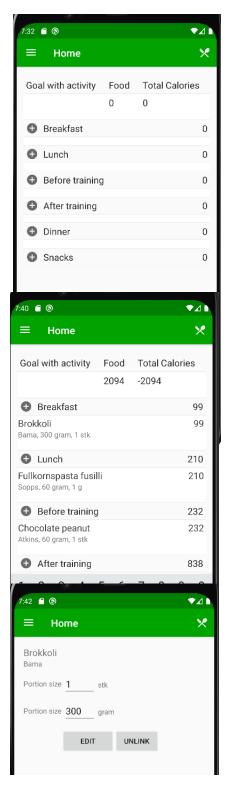
Showing Database with Chromium:



User added after SignUp screen:



Home Screen:

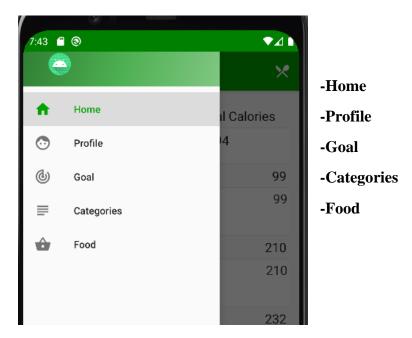


On this screen, we can add the foods we take daily to the appropriate time period.

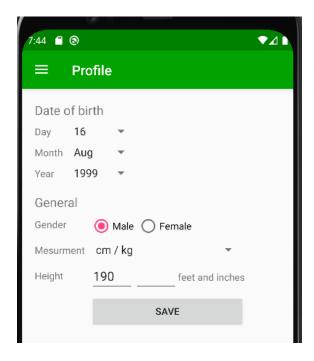
The total amount of calories consumed will be written above.

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

NavBar:



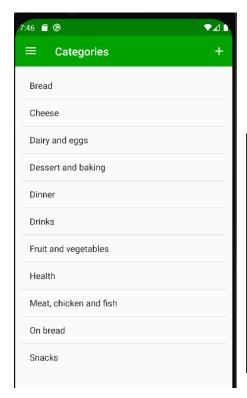
Profile Screen:



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, 'Ereals', '1', '', NULL");
    setupInsertToCategories("NULL, 'Frozen bread and rolls', '1', '', NULL");
    setupInsertToCategories("NULL, 'Crispbread', '1', '', NULL");
    setupInsertToCategories("NULL, 'Baking', '0', '', NULL");
    setupInsertToCategories("NULL, 'Biscuit', '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, 'Vegetables', '11', '', NULL");
    setupInsertToCategories("NULL, 'Canned fruits and vegetables', '11', '', NULL");
}
```

Food Screen:



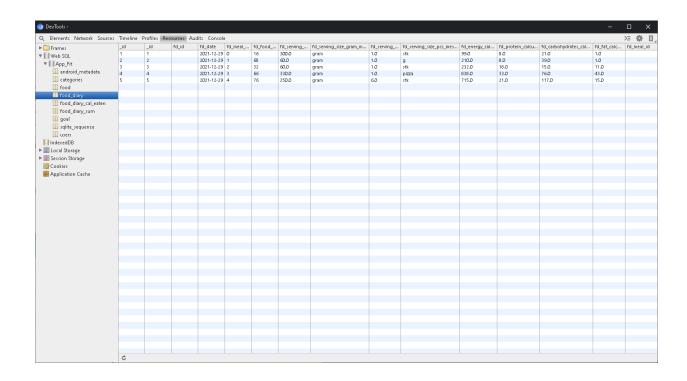
Example foods from database created before:

DevTools -																											-		
્ર Elements Network Source	s Timel	ine Pr	ofiles F	Resources	Audit	Cons	ole																					≻≡	Ф
in Frames	_id	_id	foo	food		f00	f00			foo					foo			foo		foo	foo		food_thumb			food_imag	foo	foo	fac
El Web SQL	1	1		Speltl	Aulie			26.0	gram	1.0	stk	122.0		23.4	1.0	32.0	1.0	6.0	0.0			2	aulie_spelti		aulie_speltl	aulie_speltl			
▼ El App_Fit	2	2		Havre	Axa			60.0	gram		9	389.0	11.4	63.1	7.8	233.0		38.0	5.0			3	axa_havreg		axa_havreg	axa_havreg			
android metadata	3	3		Havre	Axa			0.08	gram	0.08	g	380.0	13.0	61.0	7.0	304.0	10.0	49.0	6.0			3	axa_havreg	axa_havreg	axa_havreg	axa_havreg			
	4	4		Frosn	Hat			0.08			stk	251.0	8.1	50.0	1.5	201.0	6.0	40.0	1.0			4	hatting_fro		hatting_fro	hatting_fro			
ategories	5	6		Frosn Steino	Hvit			70.0	gram	1.0	stk stk	250.0	18.0	25.0	0.5	175.D 156.D	13.D 6.0	18.0	0.0			4	rema 1000		hwita_hjerte rema 1000	hvita_hjert rema 1000			
food	0	6		Fiber	Wasa			10.0	gram gram	1.0	stk	330,0	14.0		5.5	33.0	1,0	4.0	1.0			5			wasa_fiber	wasa fiber			
food_diary	/	,		Sport +	Wasa			16.0	gram	1.0	stk	338.0	10.5		3.5	54.0	2.0	9.0	1.0			2	wasa_fiber	wasa_nber	wasa_riber wasa_sport	wasa_riber			
food_diary_cal_eaten	9	9		Sukker	Dan			100.0	gram	100.0	gram	400.0	0.0	100.0			0.0	100.0	0.0			7	dan sukker		dan_sukker	dan sukker			
food_diary_sum	10	10		Siktet				100.0	gram	100.0	gram	341.0		69.6	1.6		10.0	70.0	2.0			7	moollerens		moollerens	moollerens			
III goal	11	11		Ritz Cr				200.0	gram	1.0	pak	490.0	7.8	61.0	23.0		16.0	122.0	46.0			8	mondelez	mondelez	mondelez	mondelez			
salite sequence	12	12		Batter	Rin			500.0	gram	1.0	boks	50.0	0.4	11.5	0.0	250.0	2.0	58.0	0.0			10	ringnes_ba		ringnes_ba	ringnes ba			
users users	13	13		Frosse				250.0	gram	0.5	pose	26.0	1.9	4.2	0.2	65.0	5.0	11.0	1.0			12	eldorado_fr		eldorado_fr	eldorado f			
	14	14		Fosne	Re			225.0		0.5	pose	27.0	2.8	1.9	0.5	61.0	6.0	4.0	1.0			12	rema 1000		rema 1000	rema 1000			
[IndexedDB	15	15		Røide	Bama			200.0	gram	0.5	pak	62.0	0.7	13.8	0.2	124.0	1.0	28.0	0.0			13	bama_rode		bama_rode	bama rode			
III Local Storage	16	16		Brokkoli				300.0	aram		stk	33.0	2.8	7.0	0.4	99.0	8,0	21.0	1.0			14	bama brok		bama brok	bama brok			
Session Storage	17	17		Gulrot	Bama			44.0	gram	1.0	stk	41.0	0.9	10.0	0.2	18.0	0.0	4.0	0.0			14	bama_gulr	bama_gulr	bama_guir	bama guir			
Cookies	18	18		Isberg	Bama			125.0	aram	0.5	pak	14.0	0.8	2.2	0.1	18.0	1,0	3.0	0.0			14	bama_isber		bama_isber	bama isber			
Application Cache	19	19		Isberg	Bama			50.0	gram	1.0	neve	12.0	0.8	1.5	0.1	6.0	0.0	1.0	0.0			14	bama isber	bama isber	bama isber	bama isber			
Аррисации Сасие	20	20		Meksi	Bama			135.0	gram	0.5	pak	140.0	3.0	3.1	11.2	189.0	4.0	4.0	15.0			14	bama meks	bama meks	bama meks	bama meks			
	21	21		Rød p	Bama			36.0	gram	0.25	stk	30.0	1.0	4.7	0.4	11.0	0.0	2.0	0.0			14	bama_rod	bama_rod	bama_rod	bama_rod			
	22	22		Roma	Bama			0.88	gram	0.5	pak	13.0	1.3	1.6	0.2	11.0	1.0	1.0	0.0			14	bama_roma	bama_roma	bama_roma	bama_roma			
	23	23		Baked	Coop			420.0	gram	1.0	boks	116.0	5.0	19.0	0.5	487.0	21.0	0.08	2.0			15	coop_bake	coop_bake	coop_bake	coop_bake			
	24	24		Kokos	Eld			60.0	gram	0.25	boks	68.0	0.8	3.3	5.7	41.0	0.0	2.0	3.0			15	eldorado_k	eldorado_k	eldorado_k	eldorado_k			
	25	25		Lett k	Eld			60.0	gram	0.25	boks	68.0	0.8	3.3	5.7	41.0	0.0	2.0	3.0			15	eldorado_l	eldorado_l	eldorado_l	eldorado_l			
	26	26		Maisk	Eld			99.0		0.5	boks	76.0	2.3	14.0	1.0	75.0	2.0	14.0	1.0			15	eldorado	eldorado	eldorado	eldorado			
	27	27		Tomat				420.0	gram	1.0	boks	83.0	3.8	14.0	0.6	349.0	16.0	59.0	3.0			15	nora_tomat		nora_tomat	nora_tomat			
	28	28		Bønne	Re			420.0	gram	1.0	boks	95.0	5.3	15.0	0.6	399.0	22.0	63.0	3.0			15	rema_1000	rema_1000	rema_1000	rema_1000			
	29	29		Power	Stro			0.08	gram	1.0	skje	396.0	37.5	39.5	9.6		30.0	32.0	0.8			17	strongr_po	strongr_po	strongr_po	strongr_po			
	30	30		Pure P				70.0		1.0	skje	382.0	43.3	43.2	4.8		30.0	30.0	3.0			17	strongr_pu	strongr_pu	strongr_pu	strongr_pu			
	31	31		MealT	Tec			0.08	gram	1.0	skje	403.0		28.0	9.0	322.0		22.0	7.0			17	tech_nutriti		tech_nutriti	tech_nutriti			
	32	32		Choco				60.0	gram	1.0	stk	386.0	29.4	24.8	19.0		18.0	15.0	11.0			18	atkins_choc		atkins_choc	atkins_cho			
	33	33		Protei	Max			50.0	gram	1.0	stk	328.0		27.0	9.2	164.0	21.0	14.0	5.0			18	maxim_prot		maxim_prot	maxim_prot			
	34	34		YT 1 0	Tine			50.0	gram		stk	427.0	21.0	45.0	17.0	214.0		23.0	9.0			18	tine_yt_1_o		tine_yt_1_o	tine_yt_1_o			
	35 36	35		YT Res	Tine			65.0	gram	1.0	stk	377.0	29.0	48.0	8.3	245.0	19.0	31.0	5.0			18	tine_yt_rest		tine_yt_rest	tine_yt_rest			
		36		Lean	Gy			30.0	gram		skje	410.0		1.4	1.4	123.0		0.0	0.0			19	gymgrossis		gymgrossis	gymgrossis			
	37	37		100%				30.0	gram	1.0	skje	363.0 375.0	78.9 82.5	7.8	2.5	109.0	24.0	2.0	1.0			19	gymgrossis	gymgrossis	gymgrossis	gymgrossis			
	38	38		100% Lean P				30.0		1.0	skje	375.0	80.0	4.0	2.9 5.0	113.0	25.0	1.0	2.0			19	gymgrossis		gymgrossis	gymgrossis			
	40	40		Whey				30.0	gram	1.0	skje skje	376.0	76.0	4.0	8.0		23.0	1.0	2.0			19	gymgrossis	gymgrossis	gymgrossis gymgrossis	gymgrossis gymgrossis			
	41	40		Prozy	Tec			40.0	gram	1.0	skje	368,0	82.5	5.0	1.3		33.0	2.0	1.0			19	gymgrossis tine go mo		tine go mo				
	42	42		Sprek	Tec			28.0		1.0	skje	262.0	71.0	23.3	1.0	73.0	20.0	7.0	0.0			19	tine_go_mo		tine_go_mo				
	43	43		Stiern	Gilde			23.0	gram	1.0	skive	301.0		0.5	27.0	69.D	3.0	0.0	6.0			21	gilde_stjern		gilde_stjern				
	44	44		Kvern	Nor			200.0		0.5	pak	123.0	19.0	0.0	5.0	246.0		0.0	10.0			21	nordfjord		nordfjord	nordfjord			
	45	45		Kvern	Re			200.0	gram	0.5	pak	123.0		0.0	1.9	246.0	38.0	0.0	4.0			21	rema_1000		rema_1000	rema_1000			
	46	46		Striml	Re					0.5		147.0			7.0	294.0		0.0	14.0			21	rema_1000			rema_1000			
	C				A.F				- At most								*****												



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