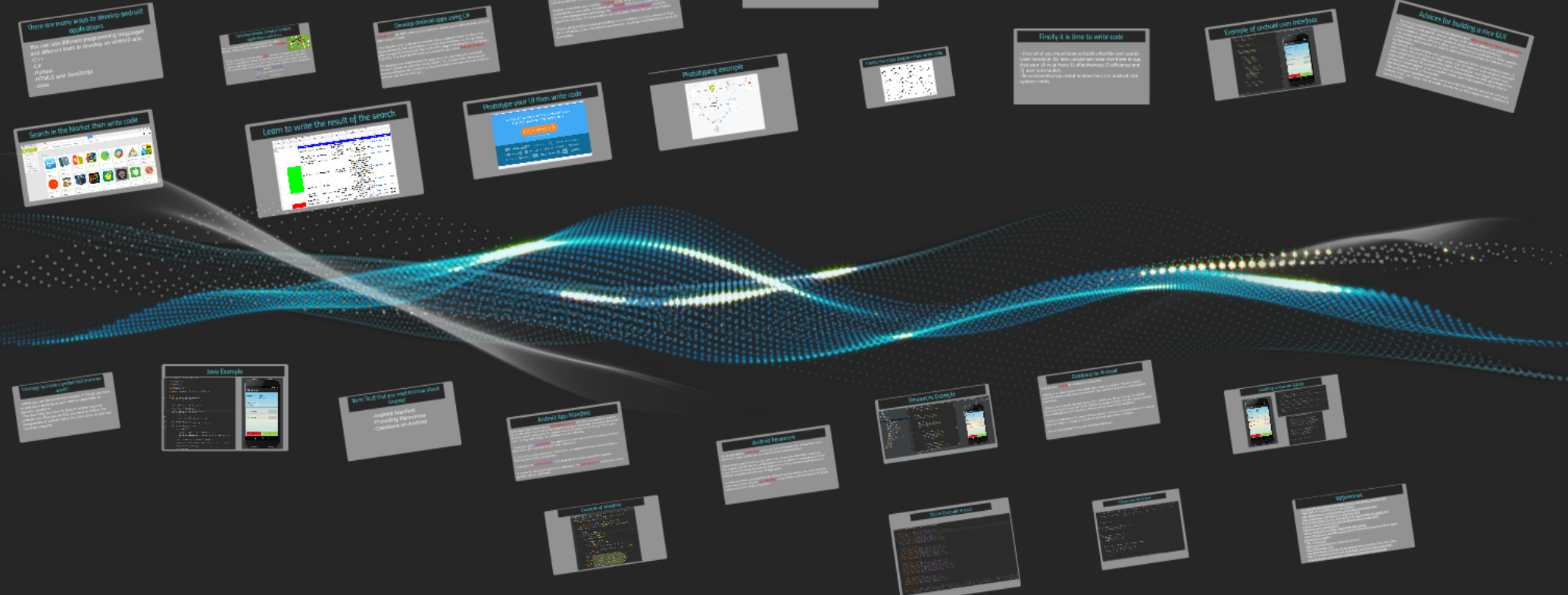
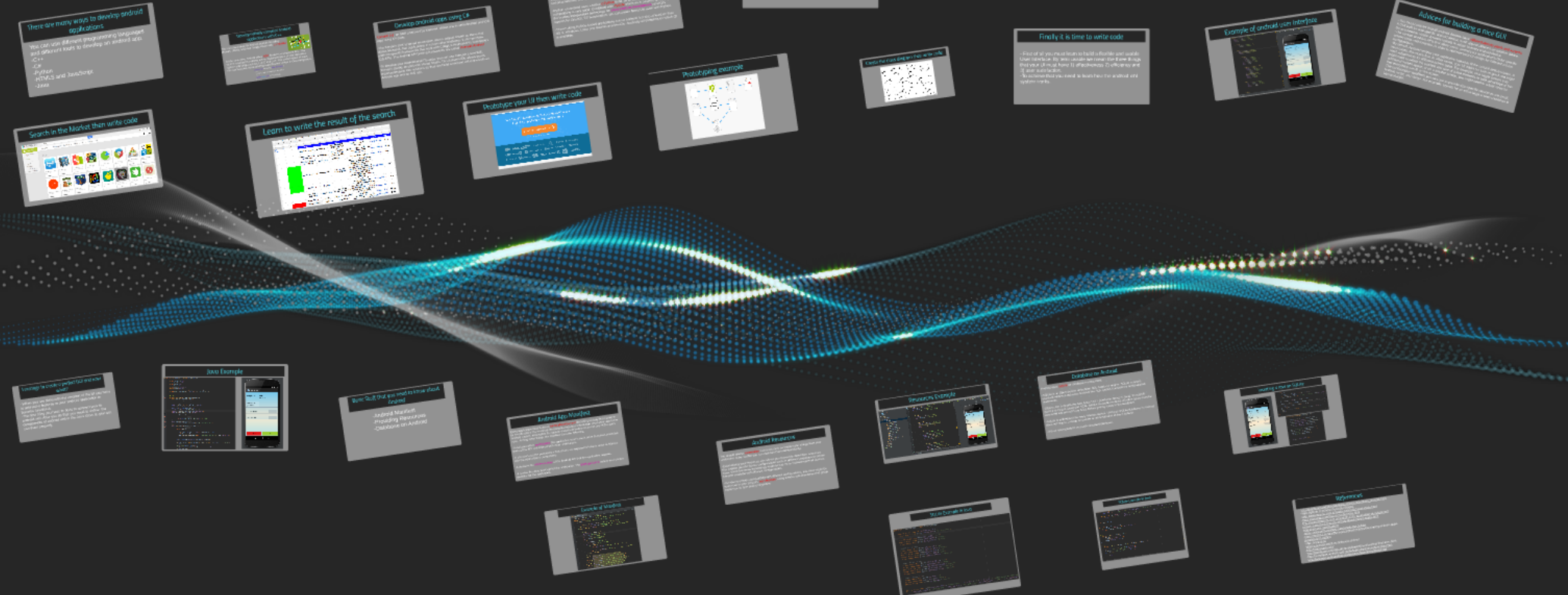


by *Nick Zisis*



by *Nick Zisis*



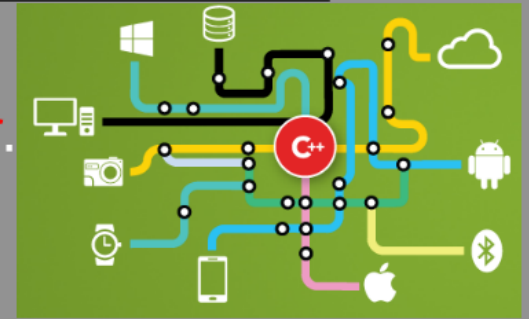
# There are many ways to develop android applications

You can use different programming languages and different tools to develop an android app.

- C++
- C#
- Python
- HTML5 and JavaScript
- Java

# Develop natively compiled Android applications with C++

You can create apps for Android devices including phones, tablets, and now Google Glass with *C++Builder*.



At the same time Android offers *NDK* (Native Development Kit) . The NDK is a toolset that allows you to implement parts of your app using native-code languages such as C and C++. For certain types of apps, this can help you reuse existing code libraries written in those languages.

```
public class MyActivity extends Activity {  
    /**  
     * Native method implemented in C/C++  
     */  
    public native void computeFoo();  
}
```

# Develop android apps using C#

*Xamarin 2.0*, an SDK produced by Xamarin, allows you to write Android and iOS apps using C# code.

-The Xamarin SDK is based on an open source project known as Mono that allows Microsoft .Net applications to run on other platforms. It also provides platform-specific frameworks that work with Google's Android APIs and Apple's iOS APIs. The Android API binding frameworks are called *Xamarin.Android*

-To develop your Android and iOS apps, you can use Xamarin's own IDE, Xamarin Studio, or stick with Visual Studio. The Xamarin IDE allows you to organize projects into solutions so that one build command will make both an Android app and an iOS app.

# DEVELOPING ANDROID APPS COMPLETELY IN PYTHON

While Android already has a good SDK out of the box, being able to use *Python* instead of Java is a big advantage for some developers - it allows for quicker turnaround times, and reuse of Python libraries.

- Python on Android uses a native *CPython* build, so its performance and compatibility is very good. Combined with *PySide* which uses a native Qt build (Qt is the leading independent technology for *cross-platform development*) and Qt's support for OpenGL ES acceleration, you can create fluent UIs even with Python.

- The resulting PySide-based applications run on Android, but also at least on Mac OS X, Windows, Linux and Maemo and MeeGo - basically all platforms on which Qt is available.

# Develop Android Apps with HTML5 and JavaScript

-*PhoneGap* is a free and open source framework that allows you to create mobile apps using web technologies you know and love: **HTML**, **CSS**, and **JavaScript**

-*Ionic* is a powerful HTML5 SDK that helps you build native-feeling mobile apps using web technologies like **HTML**, **CSS**, and **Javascript**.

-Ionic is focused mainly on the look and feel, and UI interaction of your app. That means it isn't a replacement for PhoneGap. Instead, Ionic simply fits in well with these projects in order to simplify one big part of our app: the **front end**.



# Develop Android Apps with Java

The two most famous Android IDE using Java are **Eclipse** and **Android Studio**. Which should i use?

-**Android studio** is made for the Android only while in **Eclipse** you can write code for many languages. This is a feature and a drawback for **Eclipse**.

-**Eclipse** lacks the integration of Git in its system. That doesn't mean, that you can't use Git with **Eclipse**. You can do it, but it will be a long process. **Android Studio** has taken this thing into account. **Android Studio** comes with a Git integration. That means you can make a repository and fetch the data with just a click. That made the **Android Studio** very helpful for the developers.

-Both IDEs work differently in an effort to help you manage and organize your projects. The **Eclipse** uses workspace that is familiar with java developers, on the other hand **Android Studio** introduce the concept of modules. Your app could be one module, a library that you just downloaded can be another and the Ad SDK you are currently integrating could be a third. Each of these modules can have their own Gradle build files and declare their own dependencies.

!!! If you are new to android programming and you want to develop apps with java use **Eclipse** cause is better for newbies, on the other hand if you got some experience use **Android Studio**



Lets start building an android application.  
First of all you need to think an idea

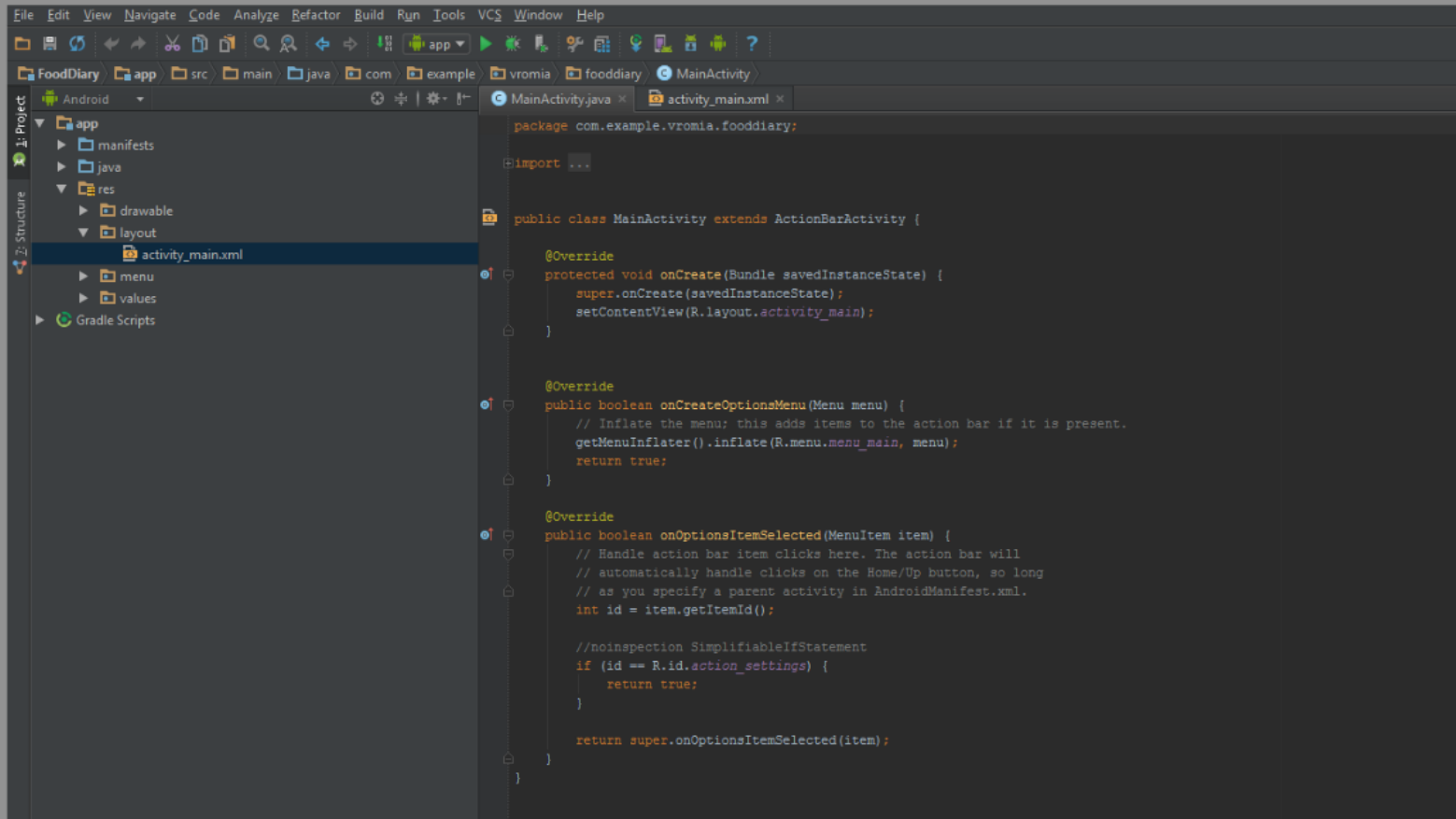


# What IDE should i choose?

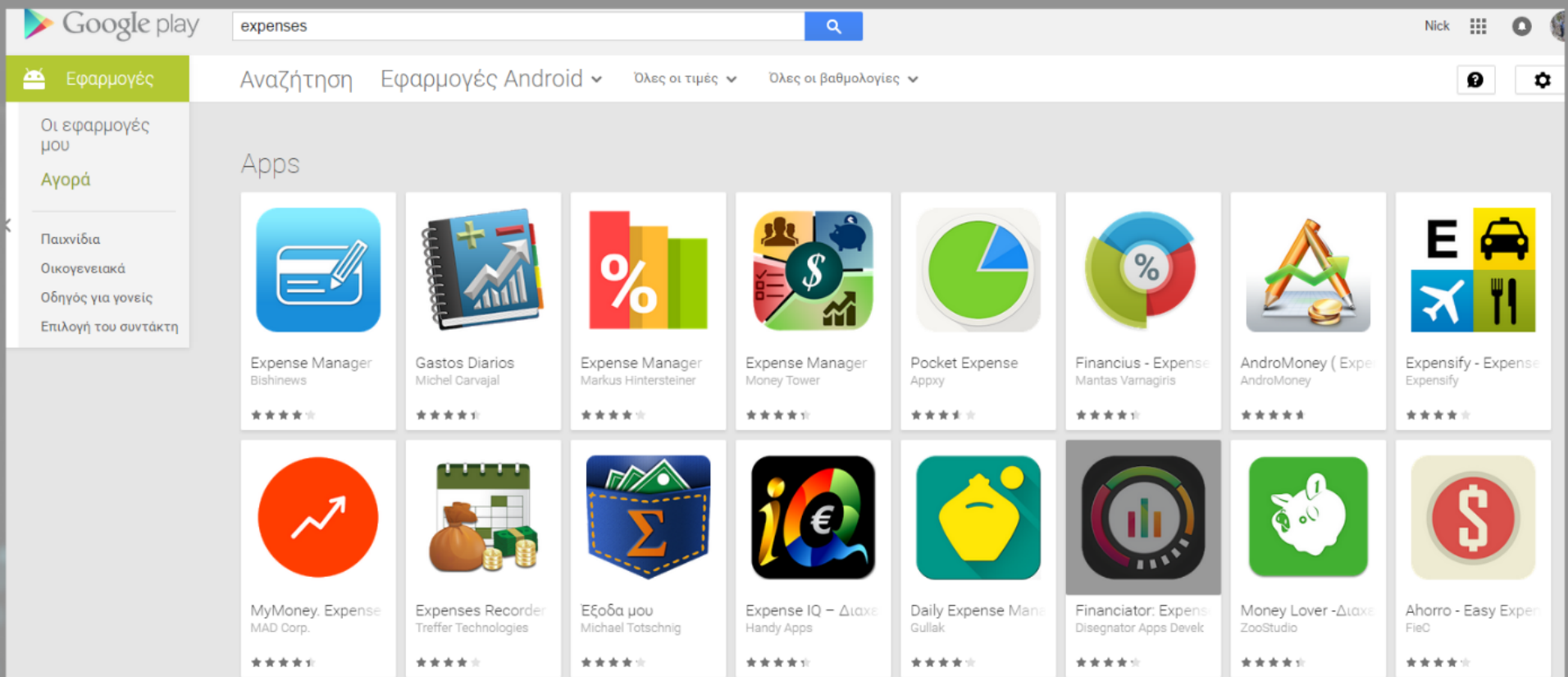
As we described in previous section there are many ways and tools to develop an android application. So it depends totally on you.

If you are comfortable with java use an IDE that uses Java such as Eclipse or Android Studio. On contrast , if you are comfortable with other programming language use the IDE's that this language provides.

# I use Android Studio



# Search in the Market then write code



## Learn to write the result of the search

A	B	C	D	E	F	G	H
		Όνομα	Πεδίο	GUI	Modules	Link	Found by
		Complete Chemistry	Εκπαίδευση	Ικανοποιητικό με πολλές φωτογραφίες Χημείας	Tutorial, Formulas, Quiz, dictionary	etails?id=com.	Zissis
		Organic Chemistry Nomenclature	Εκπαίδευση	Πολύ δυνατό QUI ωραίες photo με ζωντανά χρώματα	Quizzes με διάφορες κατηγορίες που μπορείς να τις συνδυάσεις κιάλας	m/store/apps/d	Zissis
		Functional Groups in Chemistry	Εκπαίδευση	GUI πολύ απλό	Quiz για οργανική χημεία	/apps/details?i	Zissis
		Chemistry Lab Suite	Εκπαίδευση	GUI απλό και περιποιημένο	Επαγγελματικό εργαλείο για μαθητές η επαγγελματίες σε peptide chemistry, mass spectrometry based proteomics and metabolomics, biochemistry and general lab work. Υποστηρίζει 4 modules 1)Solutions & Buffers 2)Proteins 3)Peptide Synthesis 4)Chemicals	re/apps/details	Zissis
		2048 chemistry	Παιχνίδι	-	Το παιχνίδι 2048 προσαρμοσμένο στον περιοδικό πίνακα της Χημείας	/apps/details?i	Zissis
		Chemistry for Engineers	Εκπαίδευση	Δυνατό QUI background χρώμα το μπλε	Καλύπτει 91 θέματα Χημείας τα οποία χωρίζονται σε 5 θέματα. Δίνει εξισώσεις και γραφικές αναπαραστάσεις	s?id=com.faad	Zissis
		Brain chemistry	Παιχνίδι	Μέτριο QUI για game	Surgery game	re/apps/details?	Zissis
		Chemistry News	Ενημέρωση	Πολυ κακό QUI	Περιλαμβάνει ειδήσεις που αφορούν το Πεδίο της Χημείας	/apps/details?i	Zissis

# Prototype your UI then write code

Join 213,323 designers in 204 countries who use  
Fluid UI to prototype their mobile apps

START PROTOTYPING NOW >

Free project with up to 10 screens



DELL

SAMSUNG

AUTODESK



Google

Rhapsody



LinkedIn



MOTOROLA



SAAB

ORACLE

USTREAM

T-Mobile

Symantec



sage

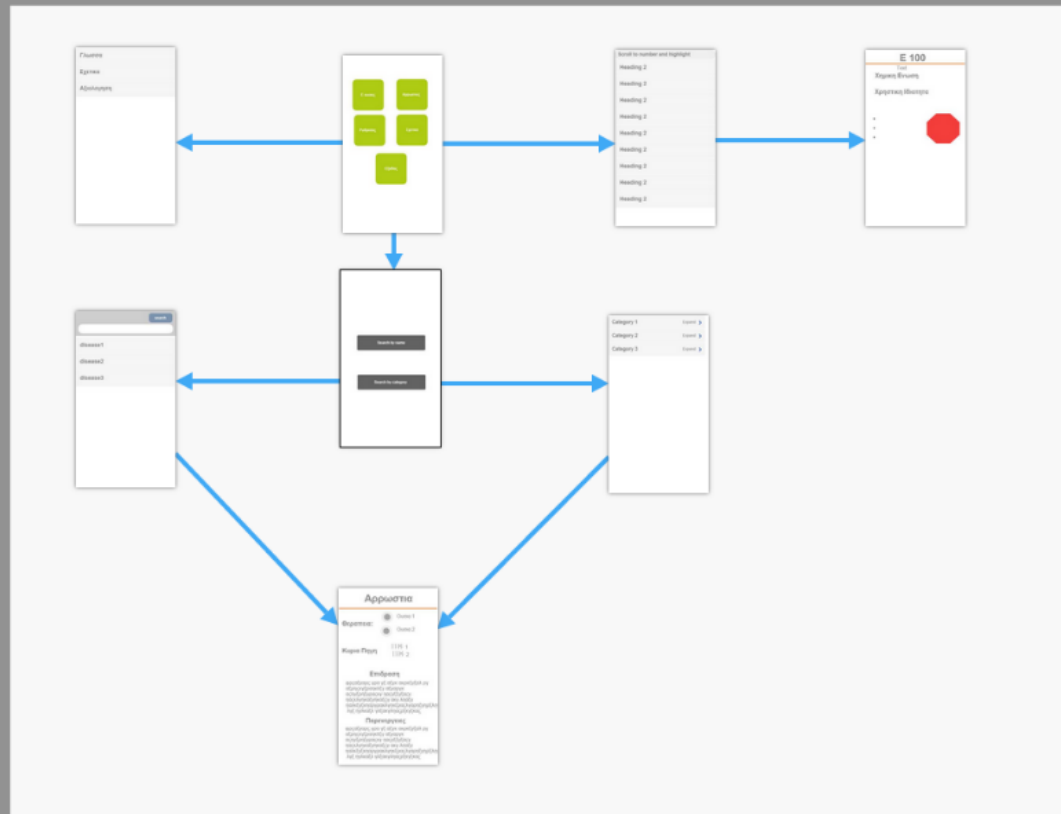
xerox



WELLS FARGO

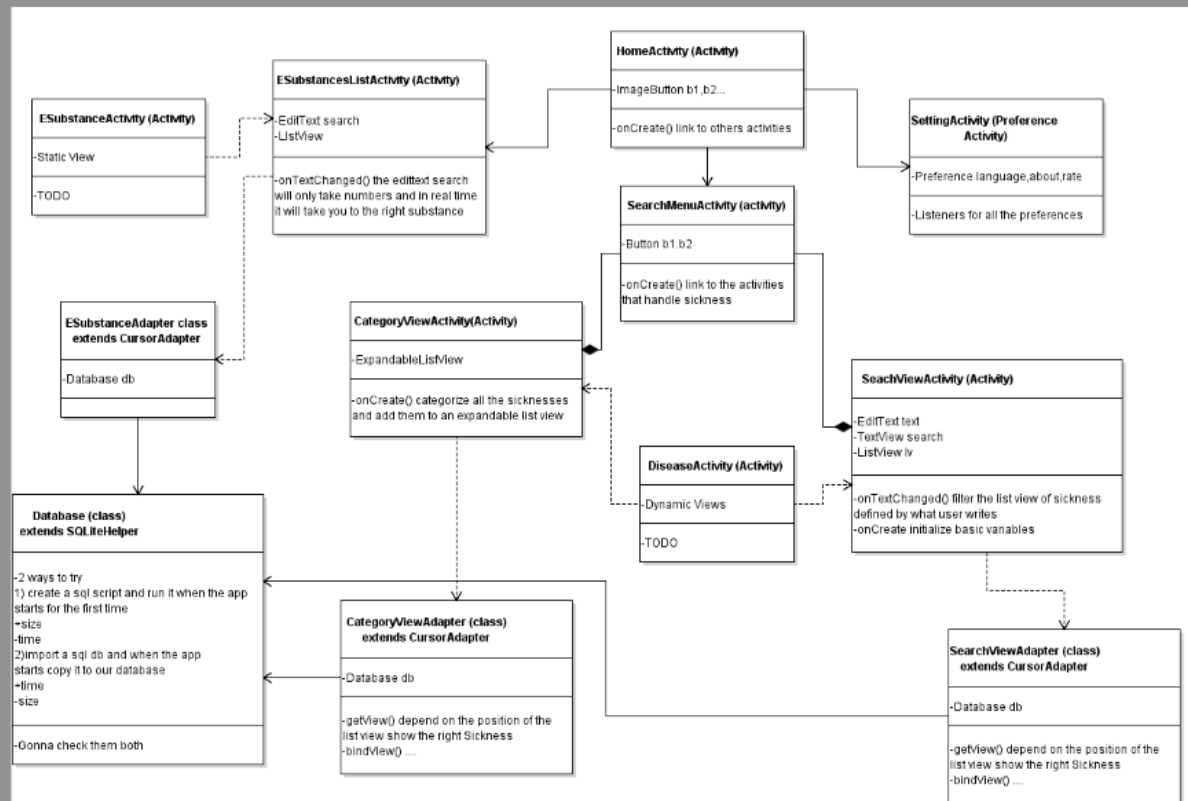
SIEMENS

# Prototyping example





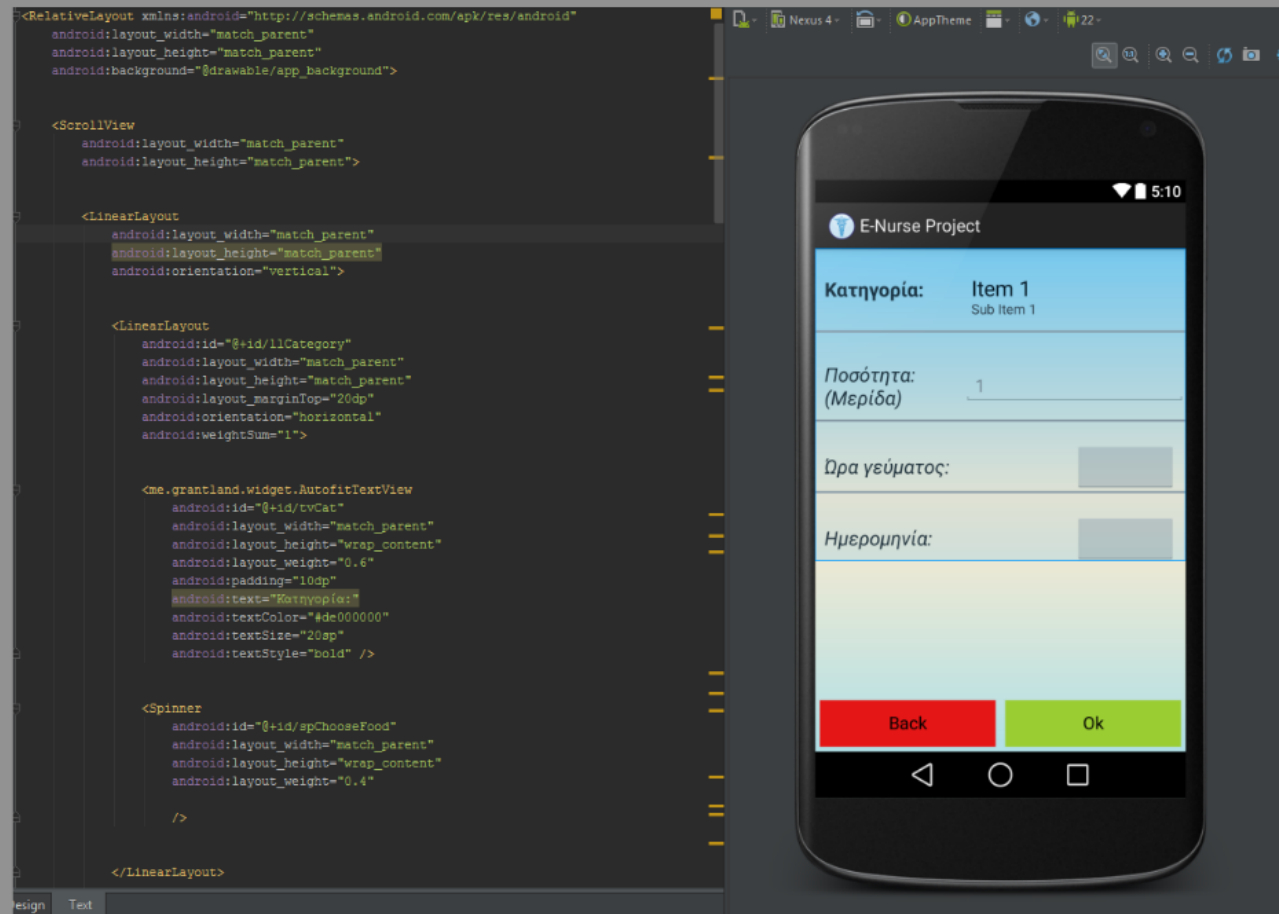
# Create the class diagram then write code



# Finally it is time to write code

- First of all you must learn to build a flexible and usable User Interface. By term usable we mean the three things that your UI must have 1) effectiveness 2) efficiency and 3) user satisfaction.
- To achieve that you need to learn how the android xml system works.

# Example of android user interface



# Advices for building a nice GUI

- You must consider that all android devices have *different density pixels and screens*. A GUI built for one specific android device, it will appear different on another device. So you need to create dynamic layouts to adjust to every android device.
  - The system provides APIs that allow you to control your application's UI for specific screen sizes and densities, in order to optimize your UI design for different screen configurations.
  - By default, Android resizes your application layout to fit the current device screen. In most cases, this works fine. In other cases, your UI might not look as good and might need adjustments for different screen sizes. For example, on a larger screen, you might want to adjust the position and size of some elements to take advantage of the additional screen space, or on a smaller screen, you might need to adjust sizes so that everything can fit on the screen.
- The configuration qualifiers you can use to provide size-specific resources are small, normal, large, and xlarge. For example, layouts for an extra-large screen should go in layout-xlarge/.

# I manage to create a perfect GUI and now what?

- When you are done with the creation of the UI you have to add some features to your android application to become functional.
- The first thing you have to do is to connect java to android xml. After you do that you need to define the components of android xml in the Java class so you can use them properly.

# Java Example

```
public class DietActivity extends FragmentActivity {

    private ImageButton bHour;
    private ImageButton bDate;
    private Button bBack;
    private Button bOk;
    private EditText quantField;
    private Spinner spinner;
    private CalendarDatePickerDialog cdate;//gui for showing date
    private RadialTimePickerDialog timeDialog;//gui for showing date
    private String date, hour;

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

        bHour = (ImageButton) findViewById(R.id.imbtHour);
        bDate = (ImageButton) findViewById(R.id.imbtDate);
        bBack = (Button) findViewById(R.id.btBack);
        bOk = (Button) findViewById(R.id.btOk);
        quantField = (EditText) findViewById(R.id.etQuant);
        spinner = (Spinner) findViewById(R.id.spChooseFood);
        bHour.setImageResource(R.drawable.clock);
        bDate.setImageResource(R.drawable.calendar);

        bHour.setOnClickListener((v) -> {
            timeDialog.show(getSupportFragmentManager(), "Nikos");
        });
        bDate.setOnClickListener((v) -> {
            cdate.show(getSupportFragmentManager(), "Calendar Dialog");
        });
        bBack.setOnClickListener((v) -> { finish(); });
        bOk.setOnClickListener((v) -> {
            double quantity = 1;
            try {
                quantity = Double.valueOf(quantField.getText().toString());
            } catch (NumberFormatException e) {
                Toast.makeText(DietActivity.this, "Plz insert a numeric value", Toast.LENGTH_LONG);
            }

            String foodName = spinner.getSelectedItem().toString();

            HeathDatabase db = new HeathDatabase(DietActivity.this);//instance of current database
            Log.i("Msg", date);
            Log.i("msg", String.valueOf(hour));
            DietItem item = new DietItem(foodName, date, quantity, hour);
            db.InsertDiet(item);
        });
    }
}
```

E-Nurse Project

Κατηγορία: Item 1  
Sub Item 1

Ποσότητα: (Μερίδα) 1

Ωρα γεύματος:

Ημερομηνία:

Back Ok

# Basic Stuff that you need to know about Android

- Android Manifest
- Providing Resources
- Database on Android



# Android App Manifest

Every application must have an *AndroidManifest.xml* file (with precisely that name) in its root directory. The manifest file presents essential information about your app to the Android system, information the system must have before it can run any of the app's code. Among other things, the manifest does the following:

- It declares which **permissions** the application must have in order to access protected parts of the API and interact with other applications.
- It also declares the permissions that others are required to have in order to interact with the application's components.
- It declares the **minimum level** of the Android API that the application requires.
- It names the Java package for the application. The **package name** serves as a unique identifier for the application.

# Example of Manifest

```
AndroidManifest.xml x
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.vromia.e_nurseproject" >

    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/enurse"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Activities.DietActivity" />
        <activity android:name=".Activities.HistoryActivity" />
        <activity android:name=".Activities.LoginActivity" />
        <activity android:name=".Activities.UserDetailsActivity" />
        <activity android:name=".Activities.WorkoutActivity" />
        <activity android:name=".Activities.SettingsActivity" />
        <activity android:name=".Activities.HomeActivity"/>
    </application>
</manifest>
```

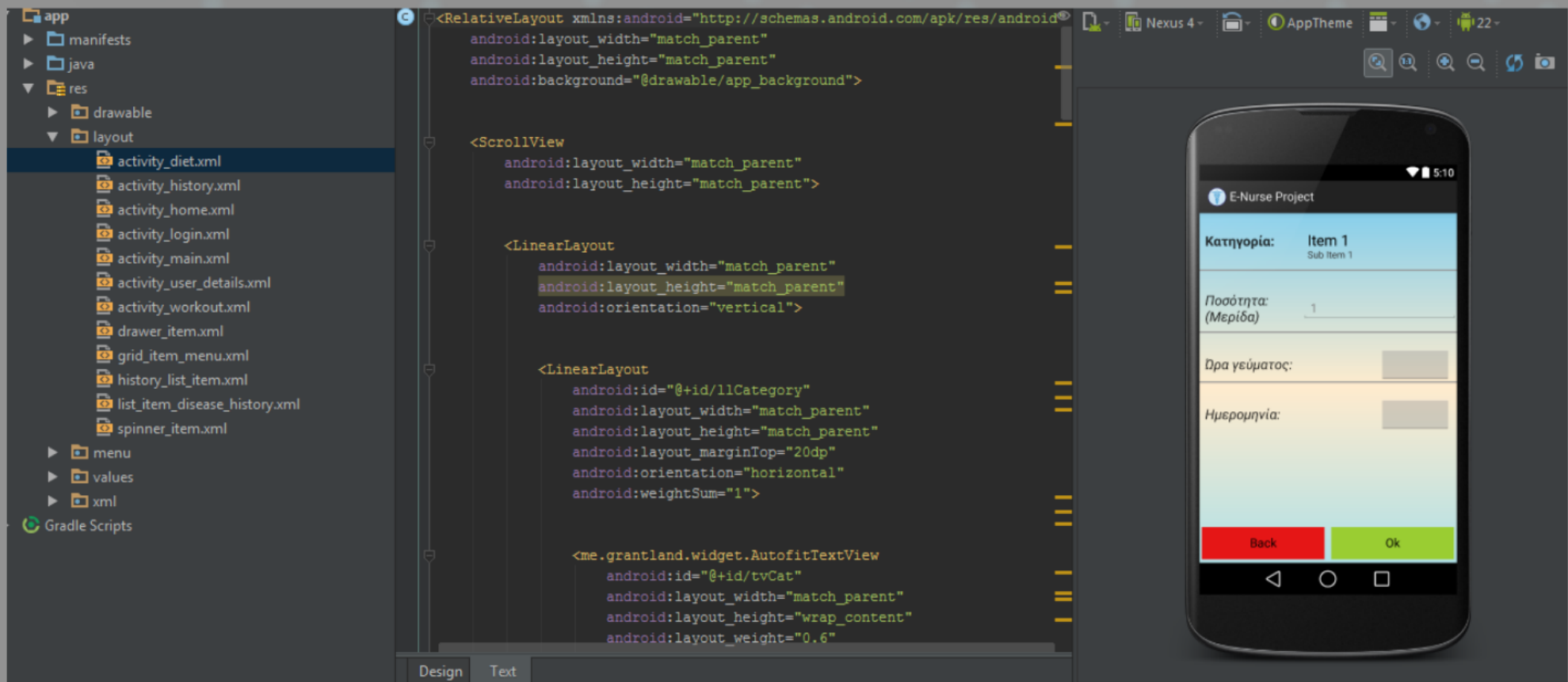
# Android Resources

You should always *externalize* resources such as images and strings from your application code, so that you can maintain them independently.

- Externalizing your resources also allows you to provide alternative resources that support specific device configurations such as different languages or screen sizes, which becomes increasingly important as more Android-powered devices become available with different configurations.

- In order to provide compatibility with different configurations, you must organize resources in your project's *res/ directory*, using various sub-directories that group resources by type and configuration

# Resources Example



# Database on Android

Android uses *SQLite* for database management.

- SQLite is an Open-Source embedded SQL database engine. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements.
- SQLite also supports the data types TEXT (similar to String in Java), INTEGER (similar to long in Java) and REAL (similar to double in Java). All other types must be converted into one of these fields before getting saved in the database.
- SQLite is embedded into every Android device. Using an SQLite database in Android does not require a setup procedure or administration of the database.
- It is an cross platform and self-contained database.

# SQLite Example in Java

```
public class HeathDatabase extends SQLiteOpenHelper {

    private static final int Database_Version = 1;
    private static final String Database_Name = "HealthDatabase";

    private static final String TABLE_DIET = "Diet";
    private static final String TABLE_WORKOUT = "Workout";
    private static final String TABLE_DOCTORS = "Doctors";

    //Table Diet columns
    private static final String KEY_DIET_ID = "_id";
    private static final String KEY_DIET_AMOUNT = "amount";
    private static final String KEY_DIET_DATE = "date";
    private static final String KEY_DIET_TIME = "time";
    private static final String KEY_DIET_CATEGORY = "category";

    //Table Workout columns
    private static final String KEY_WORKOUT_ID = "_id";
    private static final String KEY_WORKOUT_DATE = "date";
    private static final String KEY_WORKOUT_CATEGORY = "category";
    private static final String KEY_WORKOUT_TIME = "workTime";
    private static final String KEY_WORKOUT_PERIOD = "period";

    //Table Doctors columns
    private static final String Key_Did = "id";
    private static final String Key_Dname = "name";
    private static final String Key_Dsurname = "surname";

    private static final String Create_Diet_Table = "CREATE TABLE " + TABLE_DIET + "(" + KEY_DIET_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " + KEY_DIET_CATEGORY + " TEXT NOT NULL," + KEY_DIET_DATE + " TEXT NOT NULL," + KEY_DIET_AMOUNT + " DOUBLE," + KEY_DIET_TIME + "
```



# SQLite Example in Java

```
private static final String Create_Workout_Table = "CREATE TABLE " + TABLE_WORKOUT + "(" + KEY_WORKOUT_ID + " INTEGER PRIMARY KEY AUTO INCREMENT, " + KEY_WORKOUT_CATEGORY + " TEXT NOT NULL," + KEY_WORKOUT_DATE + " TEXT NOT NULL," + KEY_WORKOUT_TIME + " DOUBLE," + KEY_WORKOUT_DURATION + " DOUBLE)";

private static final String Create_Doctor_Table = "CREATE TABLE " + TABLE_DOCTORS + "(" + Key_Did + " INTEGER PRIMARY KEY ," + Key_Dname + " TEXT NOT NULL," + Key_Dsurname + " TEXT NOT NULL" + ")";

private SQLiteDatabase db;

public HealthDatabase(Context context) {
    super(context, Database_Name, null, Database_Version);
    db = this.getWritableDatabase();
}

@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL(Create_Diet_Table);
    db.execSQL(Create_Workout_Table);
    db.execSQL(Create_Doctor_Table);
}

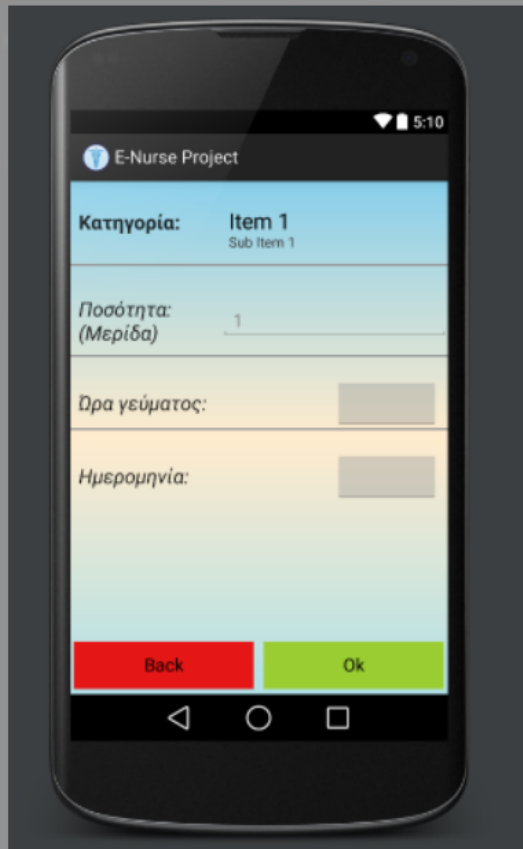
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_DIET);
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_WORKOUT);
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_DOCTORS);

    onCreate(db);
}
```





# Inserting a row on SQLite



```
book.setOnClickListener((v) -> {  
    double quantity = 1;  
    try {  
        quantity = Double.valueOf(quantField.getText().toString());  
    } catch (NumberFormatException e) {  
        Toast.makeText(DietActivity.this, "Plz insert a numeric value", Toast.LENGTH_LONG);  
    }  
    String foodName = spinner.getSelectedItem().toString();  
  
    HeathDatabase db = new HeathDatabase(DietActivity.this); //instance of current database  
  
    DietItem item = new DietItem(foodName, date, quantity, hour);  
    db.InsertDiet(item);  
    db.close();  
    Toast.makeText(DietActivity.this, "Εισαγωγή επιτυχής", Toast.LENGTH_LONG).show();  
    finish();  
});
```

```
public void InsertDiet(DietItem item) {  
  
    ContentValues cv = new ContentValues();  
    cv.put(KEY_DIET_AMOUNT, item.getAmount());  
    cv.put(KEY_DIET_CATEGORY, item.getCategory());  
    cv.put(KEY_DIET_DATE, item.getDate());  
    cv.put(KEY_DIET_TIME, item.getTime());  
  
    Log.i("Category", item.getCategory());  
    Log.i("Amount", item.getAmount() + "");  
    Log.i("Time", item.getTime());  
    Log.i("Date", item.getDate());  
  
    db.insert(TABLE_DIET, null, cv);  
}
```

# References

- [http://developer.android.com/guide/practices/screens\\_support.html](http://developer.android.com/guide/practices/screens_support.html)
- <https://github.com/nzisis/E-Nurse-Project>
- <http://www.vogella.com/tutorials/AndroidSQLite/article.html>
- <https://developer.android.com/ndk/index.html>
- <http://www.infoworld.com/article/2614376/application-development/review--xamarin-2-0-works-mobile-development-magic.html>
- <http://xamarin.com/platform>
- <https://www.embarcadero.com/products/cbuilder>
- <https://ep2013.europython.eu/conference/talks/developing-android-apps-completely-in-python>
- <http://www.qt.io/>
- <http://ionicframework.com/docs/overview/>
- <http://phonegap.com/>
- <http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- <http://developer.android.com/guide/topics/resources/overview.html>
- <http://technocr.com/android-studio-vs-eclipse-the-android-ide/>

by *Nick Zisis*

