

Android Development

Lifecycle and Intents

SoftUni Team

Teodor Kostadinov



SoftUni
Foundation

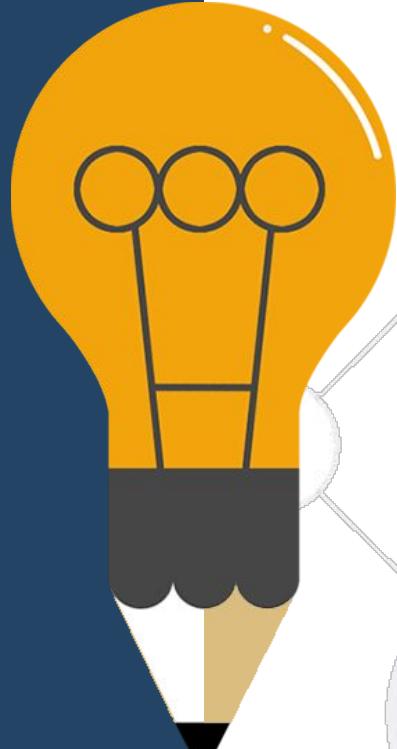


Software University
<http://softuni.bg>



sli.do
#Android

Just a second!



Do you remember what we talked about last time?

Please fill this form: <http://bit.ly/softuni-android-ui>
This will help me understand what you remember
from our previous lecture.

Android Studio Tour

- Welcome to your new favourite IDE
- It iterates over IntelliJ
- Makes Android development as painless as possible
- Let's see it in action!



Demo

Android Studio Overview

Project Files Tour

- A project usually consists of
 - Java/Kotlin files with the main logic of the app
 - XML files for UI and configuration
 - Images/assets



Demo

Project Structure Overview

- Gradle is a build tool
- It knows where your project files are, can compile them and can create an apk
- It is configured from the build.gradle files in your project
- You have two of those
 - One for the whole project
 - One for just your module (by default your module is called app)
 - Most of the settings are in your module build.gradle

Android Manifest

- A configuration file
- Describes your app's screens and services
- Tells the Android OS which screen is the first one to open
- Tells the Android OS if the app can open special file types or do special actions
- Describes what permission your app needs

Activity

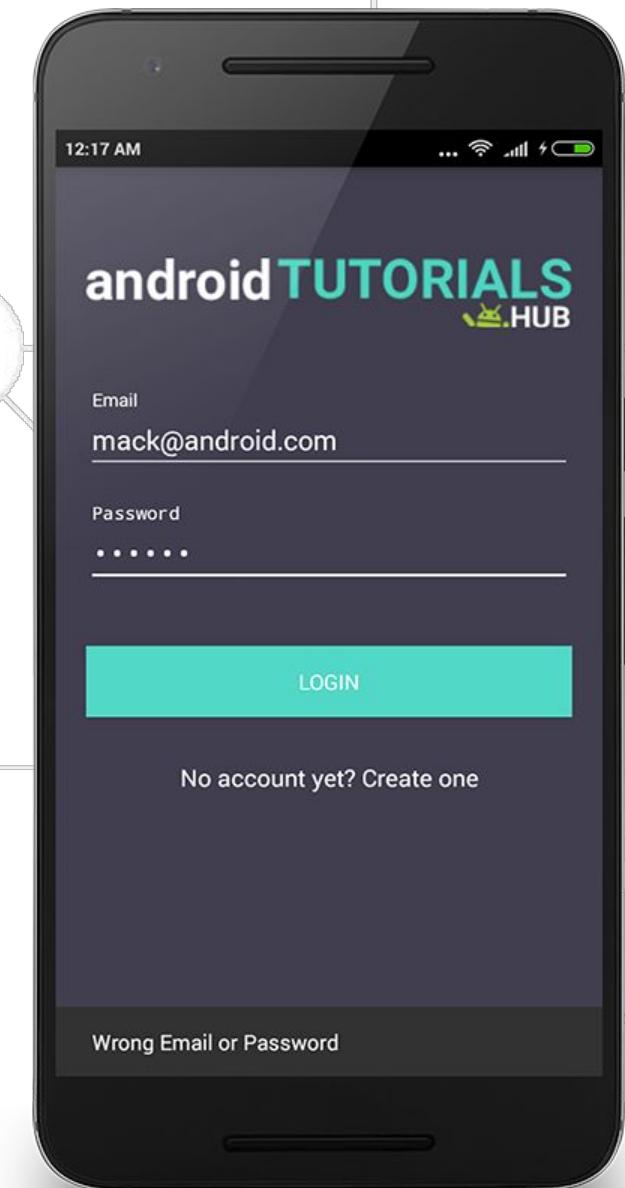
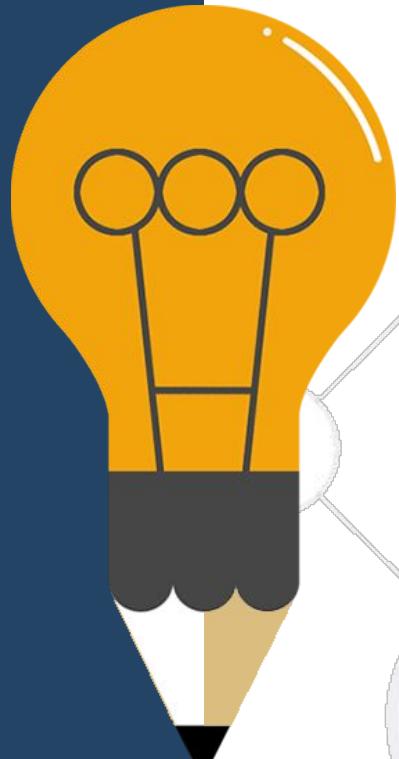
- Before
 - One activity was one screen from the application
 - The main screen logic was written there
- Now
 - Activities are only used as containers
 - Now they don't have UI but only manage sub-activities (fragments)
- A class becomes an Activity when extends
 - Activity
 - AppCompatActivity

Let's do this!

Write the following app:

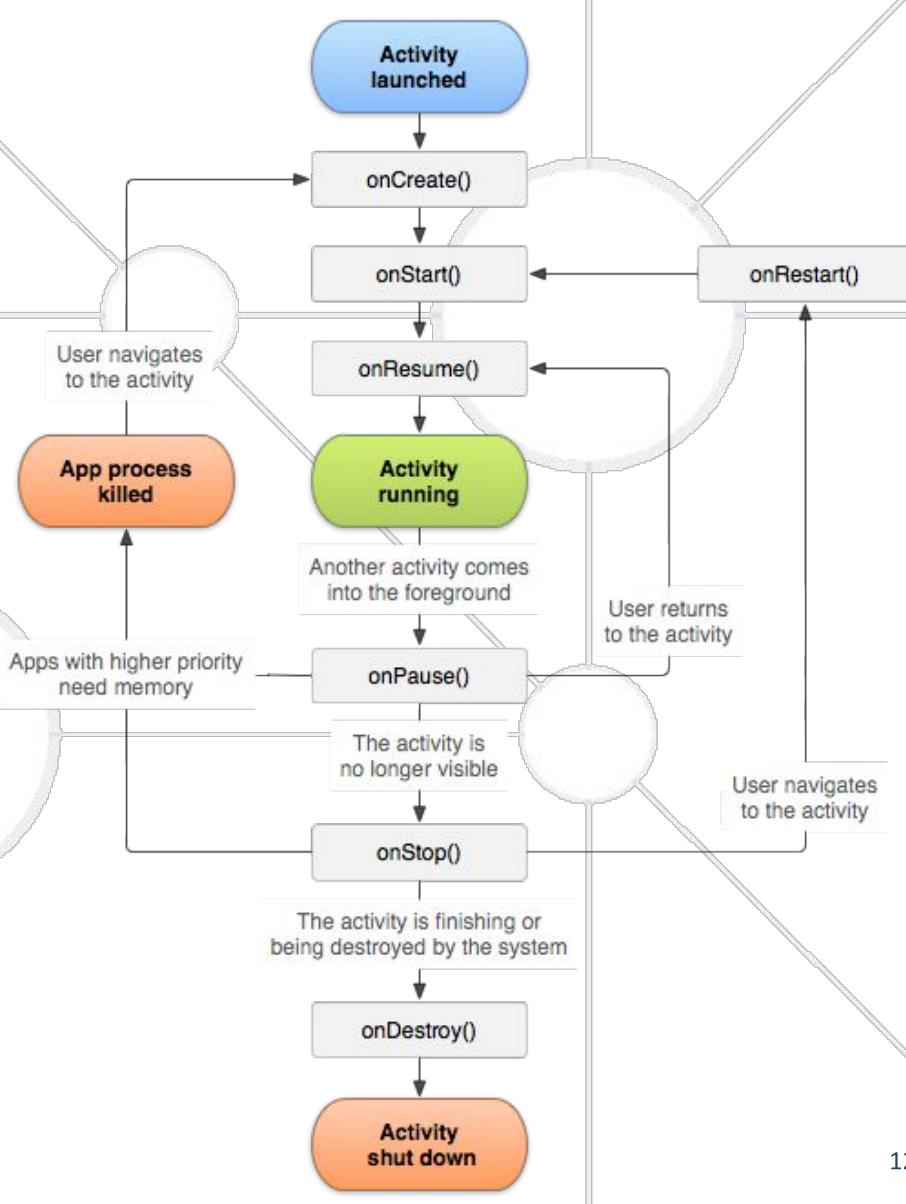
Create a login form. It should have email and password field and a

Login button. All the views should be centered on the screen. There should be a background and the button should be colored.



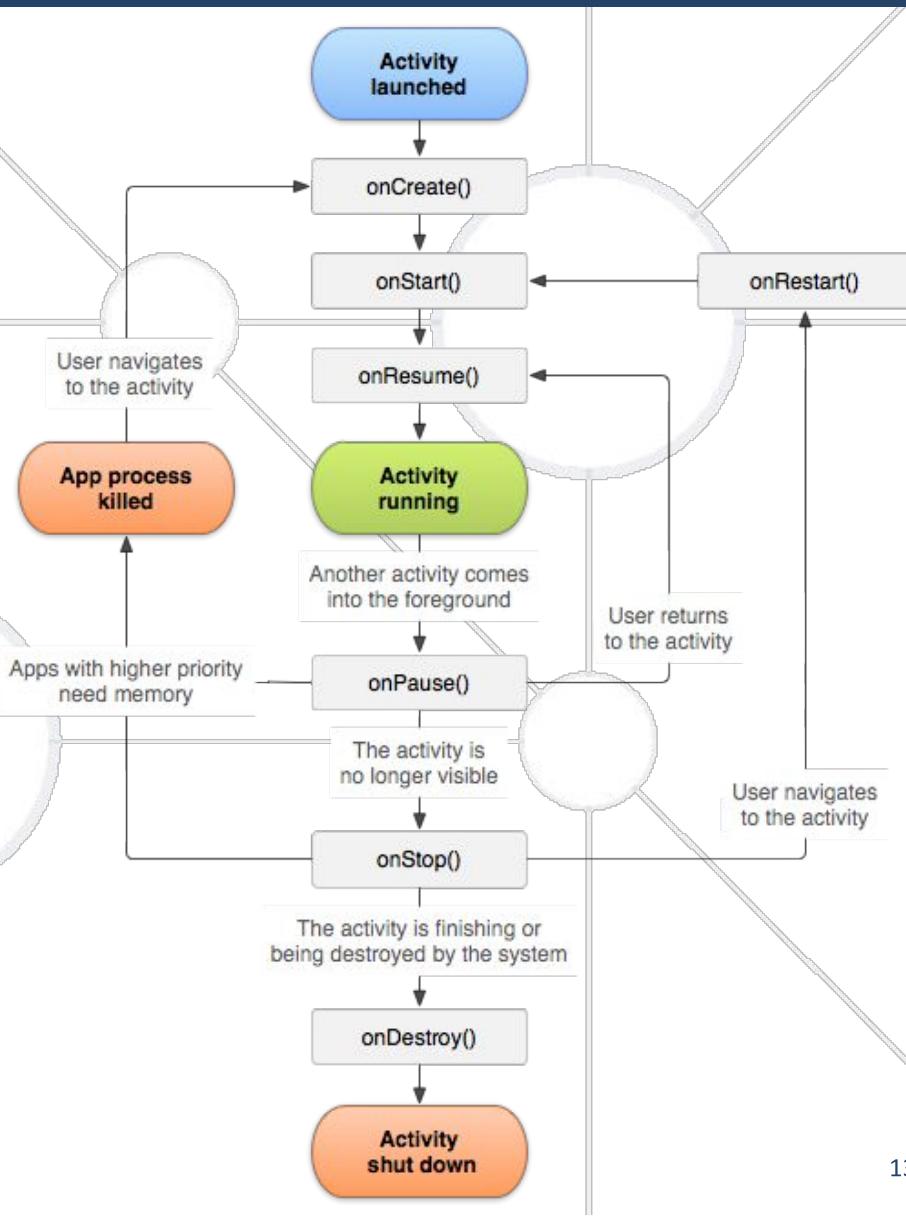
Activity Lifecycle

- Activities have different states
- Some of the states are:
 - Activity is being created
 - Activity is started
 - Activity is resumed
 - Activity is paused
 - Activity is stopped
 - Activity is destroyed
- This is called the lifecycle of an activity
- After each state-change, a method is called, which can be overridden



Activity Lifecycle in Use

- `onCreate()` - field initialization, one-time settings
- `onResume()` - refresh settings, remembered before minimizing screen
- `onPause()` - save important settings and state before screen goes in the background. Called every time something blocks this screen.
- `onDestroy()` - the activity dies permanently. Release all resources here.



Source: <https://developer.android.com/guide/components/activities/activity-lifecycle>

Intents

- Data package
- Can initiate actions
- or can just transfer data
- May have
 - action
 - sender/receiver
 - data

Explicit Intents

- Used to start Activities in your own application
- Have a specific sender and receiver

```
Intent startRegScreen = new Intent(this, RegistrationActivity.class);
startActivity(startRegScreen);
```

```
val startRegScreen = Intent(this, RegistrationActivity::class.java)
startActivity(startRegScreen)
```



Passing Data in Intent

- If you pass simple data, use Extras
- For complicated data, use Bundle

```
Intent startRegScreen = new Intent(this, RegistrationActivity.class);
startRegScreen.putExtra("username", "Pesho");
startActivity(startRegScreen);
```

```
val startRegScreen = Intent(this, RegistrationActivity::class.java)
startRegScreen.putExtra("username", "Pesho")
startActivity(startRegScreen)
```

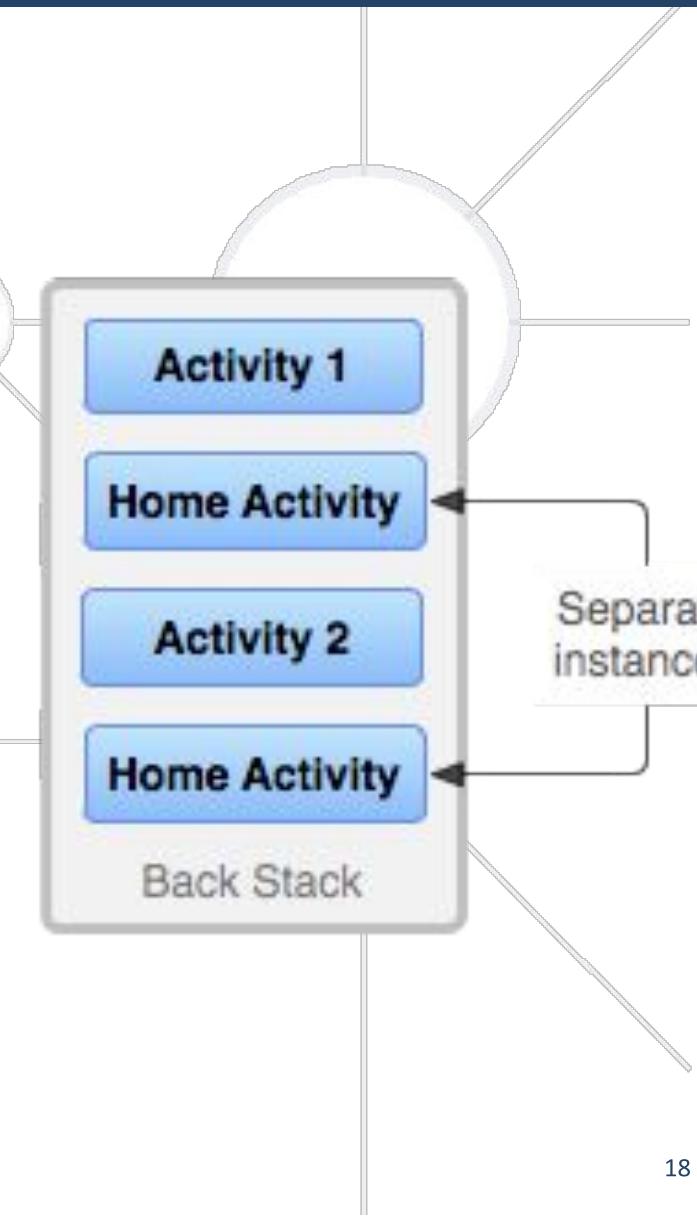




Demo Switching Activities

Backstack

- Each time a new activity is started, it is added to a stack of started activities
- When you close it (press back) the activity is removed from the stack
- The current top activity in the stack becomes visible to the user



- When starting an Activity with an Intent, you can pass some flags to it
- Flags control the way the activity is started
- Examples are
 - `FLAG_ACTIVITY_NO_HISTORY` - the starting activity is not kept in the history stack
 - `FLAG_ACTIVITY_NO_ANIMATION` - prevents the system from applying a transition animation to the activity
 - `FLAG_ACTIVITY_CLEAR_TOP` - clears all activities from the stack

Implicit Intents

- Used to start Activities that you do not own
- You don't have to specify the specific Activity
- Just what action needs to be performed

```
// Create the text message with a string
val sendIntent = Intent().apply {
    action = Intent.ACTION_SEND
    putExtra(Intent.EXTRA_TEXT, textMessage)
    type = "text/plain"
}

// Verify that the intent will resolve to an activity
if (sendIntent.resolveActivity(packageManager) != null) {
    startActivity(sendIntent)
}
```



Kotlin Corner

```
// Create the text message with a string
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT,
textMessage);
sendIntent.setType("text/plain");

// Verify that the intent will resolve to
// an activity
if (
sendIntent.resolveActivity(getApplicationContext()) != null) {
    startActivity(sendIntent);
}
```



Demo External Intents

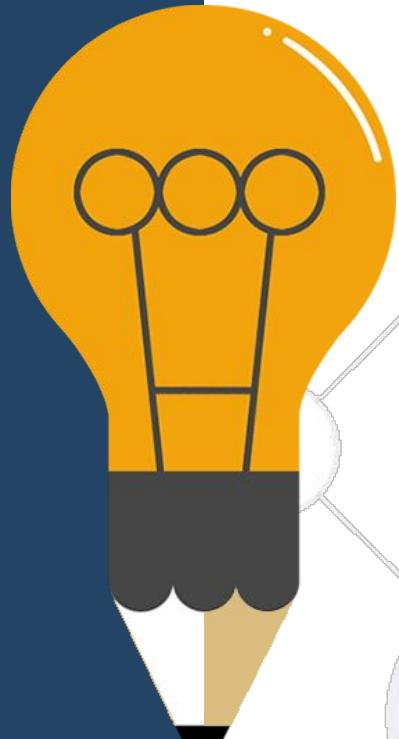
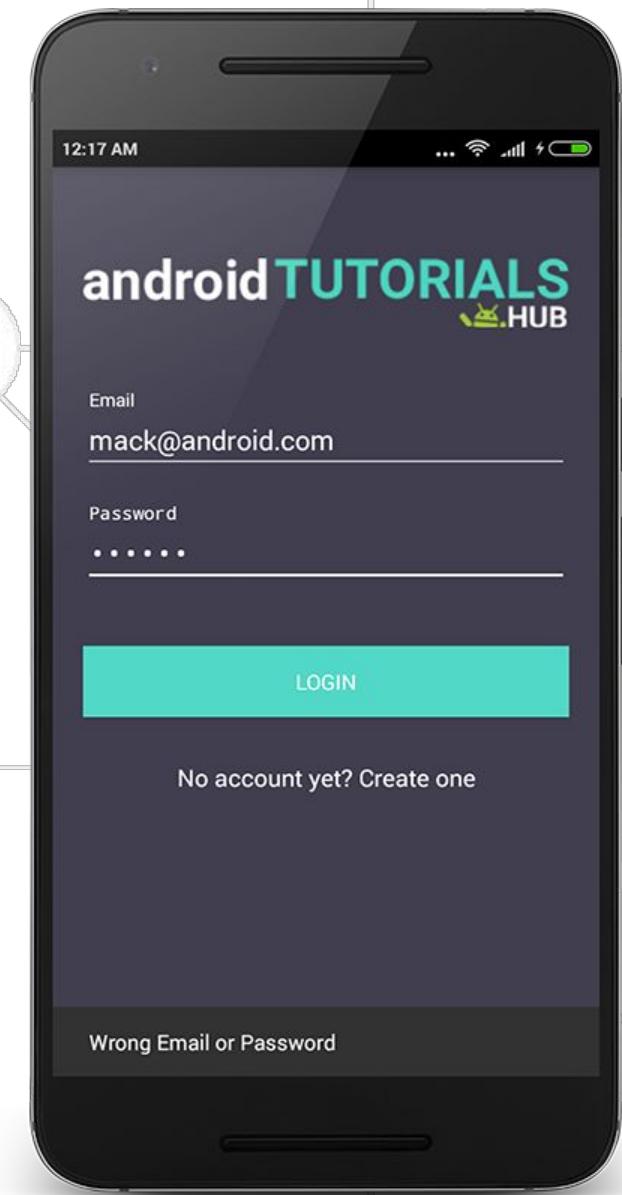
Let's do this!

Add to your app:

Below Login button add create account text. Pressing the text takes you to another screen.

RegisterScreen has:

- email
- password, repeat password
- submit button that sends the data via email
- Contact us button that dials a number
- Has a check for wrong email format and password mismatch



Result from Activity

- Activities can be started with a specific goal
- Usually activities can be started to get a result
- After the activity finishes, the result is returned in an Intent
- A request code is passed when starting such activity, so we know when we get a result, where it comes from

Source: <https://developer.android.com/training/basics/intents/result>

```
const val PICK_CONTACT_REQUEST = 1 // The request code  
...  
private fun pickContact() {  
    Intent(Intent.ACTION_PICK,  
        Uri.parse("content://contacts")).also { pickContactIntent ->  
        pickContactIntent.type = Phone.CONTENT_TYPE // Show  
        user only contacts w/ phone numbers  
        startActivityForResult(pickContactIntent,  
            PICK_CONTACT_REQUEST)  
    }  
}
```

 Kotlin Core

```
static final int PICK_CONTACT_REQUEST = 1;  
...  
private void pickContact() {  
    Intent pickContactIntent = new Intent(  
        Intent.ACTION_PICK,  
        Uri.parse("content://contacts"));  
    pickContactIntent.setType(Phone.CONTENT_TYPE);  
    startActivityForResult(pickContactIntent,  
        PICK_CONTACT_REQUEST);  
}
```

Result from Activity

- The result arrives in the starting Activity
- `onActivityResult` should be overridden

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent) {
    // Check which request we're responding to
    if (requestCode == PICK_CONTACT_REQUEST) {
        // Make sure the request was successful
        if (resultCode == Activity.RESULT_OK) {
            // The user picked a contact.
            // The Intent's data Uri identifies which contact was selected.

            // Do something with the contact here (bigger example below)
        }
    }
}
```

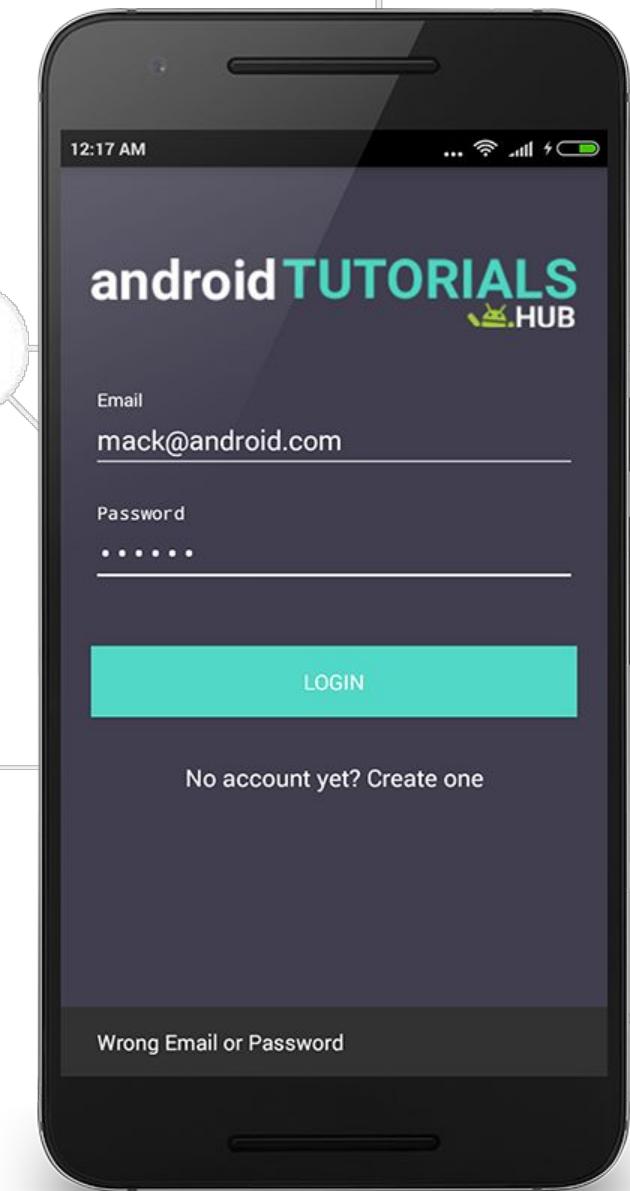
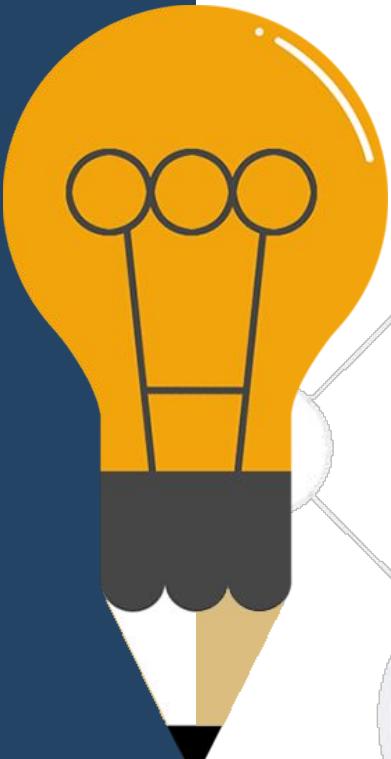
 Kotlin Corner

```
@Override
protected void onActivityResult(int requestCode,
int resultCode, Intent data) {
    // Check which request we're responding to
    if (requestCode == PICK_CONTACT_REQUEST) {
        // Make sure the request was successful
        if (resultCode == RESULT_OK) {
            // The user picked a contact.
        }
    }
}
```

Let's do this!

Add to your app:

- On the Register screen
 - Place for user avatar
 - Button to take picture with camera
 - Picture should be placed on the Register screen



Summary and Resources

Activities are screens. We use Intents to move between screens and pass data.

Resources:

<https://developer.android.com/guide/components/activities/intro-activities>

<https://www.youtube.com/watch?v=wMo2O1oL5BQ&list=PLjsqymUqgpSTXtIngZCXRHEp8-FmDHHfL&index=33>

https://www.youtube.com/watch?v=JOO0IVPL_wA&index=32&list=PLjsqymUqgpSTXtIngZCXRHEp8-FmDHHfL

<https://www.youtube.com/watch?v=y2wFjIR986I&list=PLjsqymUqgpSTXtIngZCXRHEp8-FmDHHfL&index=40>

<https://www.youtube.com/watch?v=mLvGQ64jqWk&index=13&list=PLjsqymUqgpSTXtIngZCXRHEp8-FmDHHfL>



Homework (1)

Вход

Това са три отделни
екрана

Въведете имената си

Име

Вход

При грешни данни, на полетата с
грешни данни се задава
`editText.setError("text");`

Детайли

Клавиатура само с
Години цифри

Адрес

Град

Дата на раждане

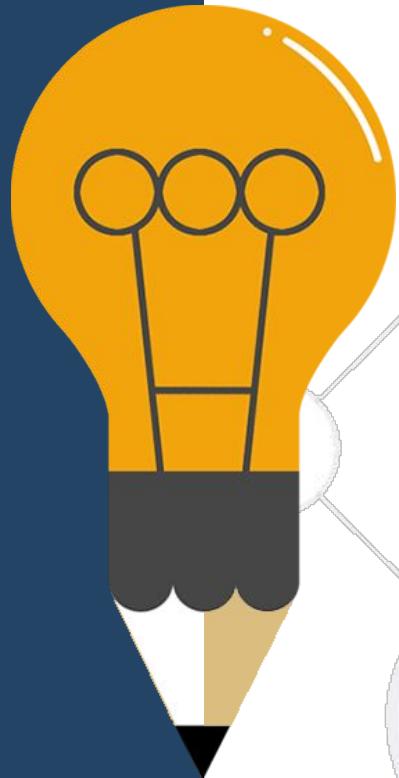
Продължи

Резюме

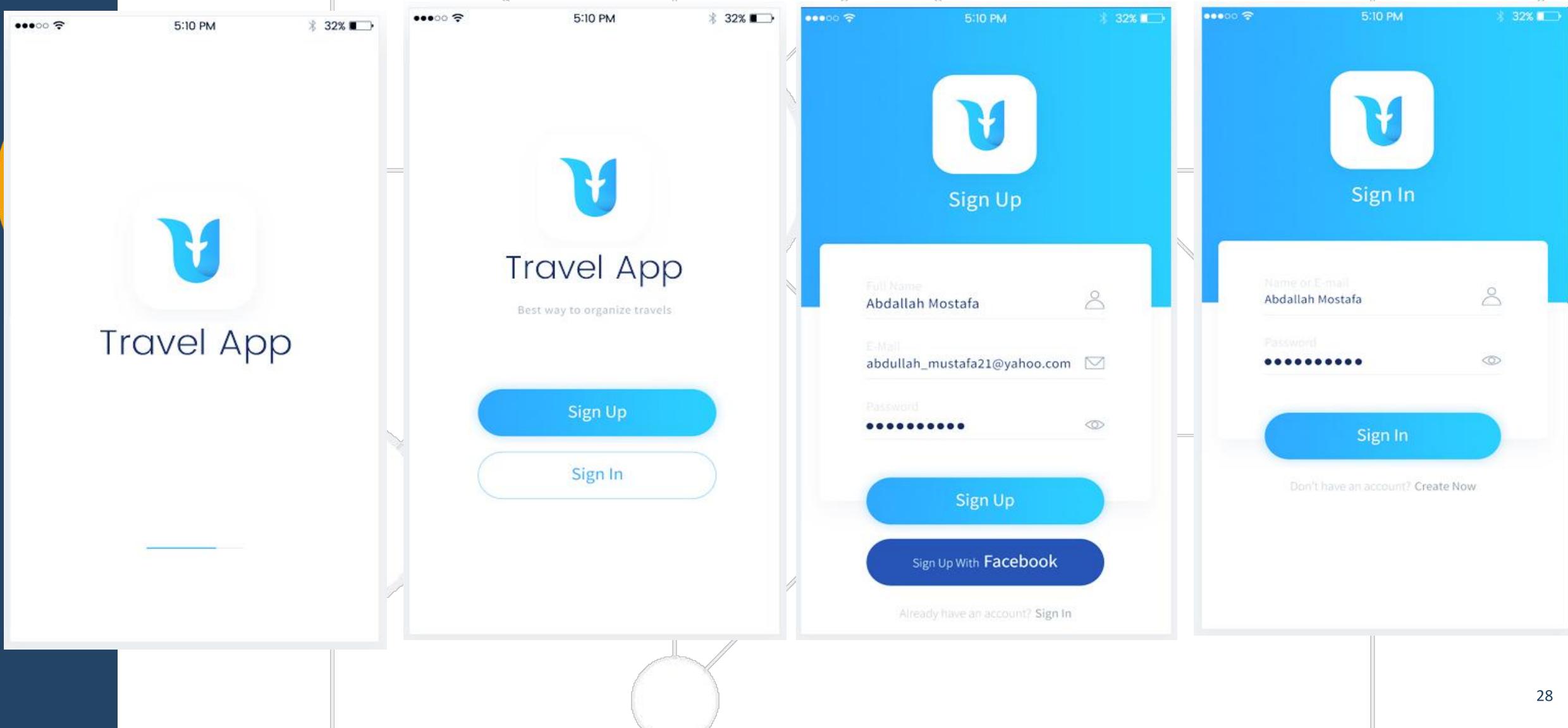
Иван Петров, 56
ул. Кокиче 17,
град Враца

Покажи адреса
на картата

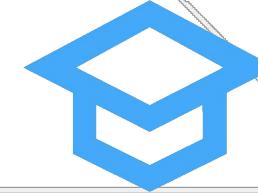
Този бутон отваря
Google Maps с карта
центрирана върху
адреса



Homework (2)



Questions?



SoftUni

Software
University

SoftUni
Svetlina

SoftUni
Creative

SoftUni
Digital

SoftUni
Foundation

SoftUni
Kids



Софтуни диамантени партньори



INDEAVR

Serving the high achievers

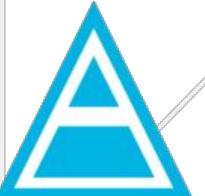


INFRASTICS®



SoftwareGroup
doing it right

NETPEAK



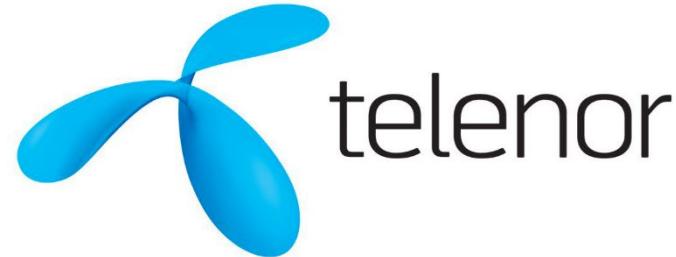
**SUPER
HOSTING
.BG**



Софтуни диамантени партньори



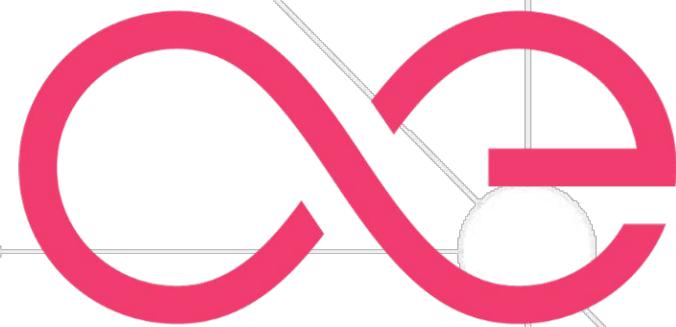
SBTech
we know sports



SmartIT



codexio



æternity

LIEBHERR

Trainings @ Software University (SoftUni)



- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University Foundation
 - <http://softuni.foundation/>
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg

