

Mobile App Development

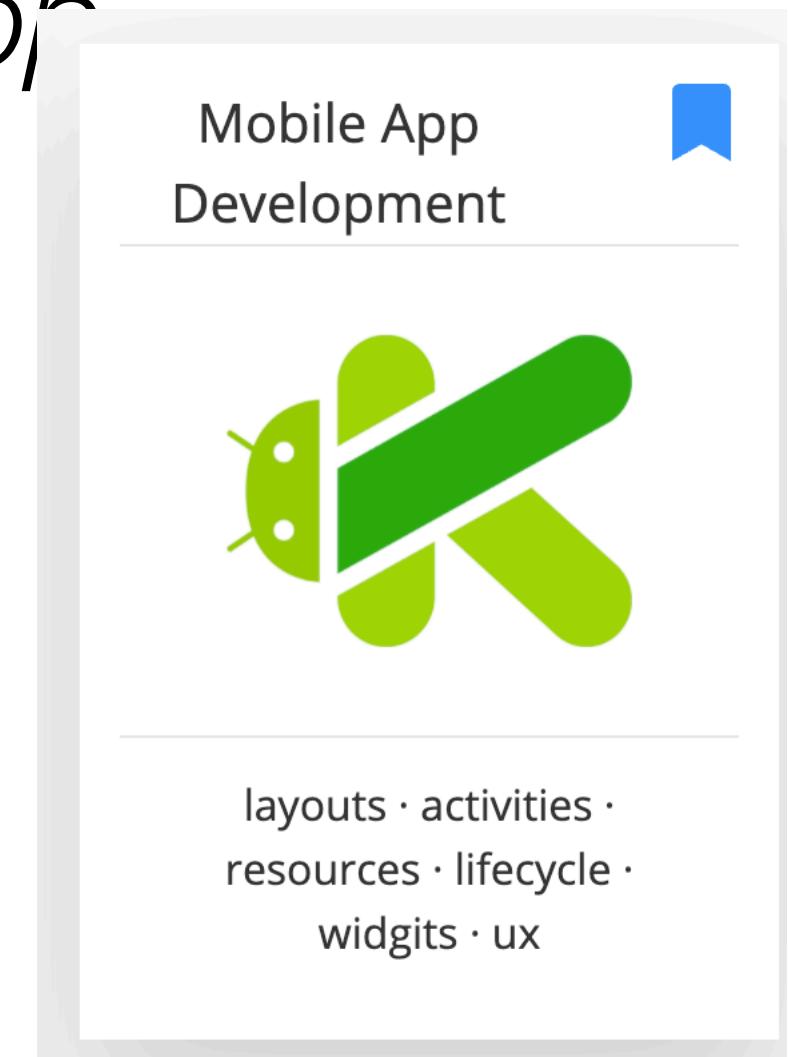
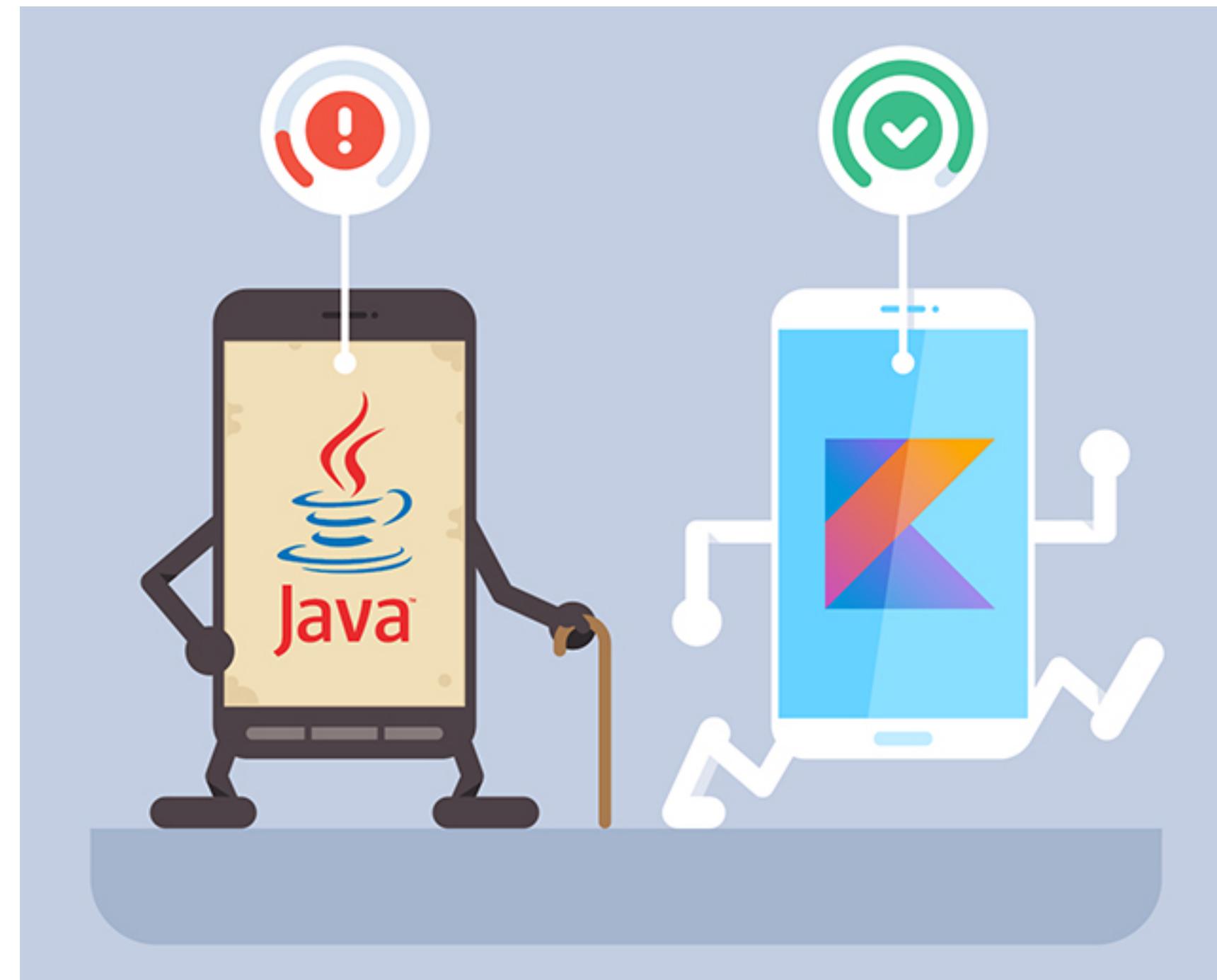


layouts · activities ·
resources · lifecycle ·
widgits · ux

2020

Android Programming with Kotlin

Course Mission: Accelerate into Android App development leveraging the power & expressiveness of Kotlin

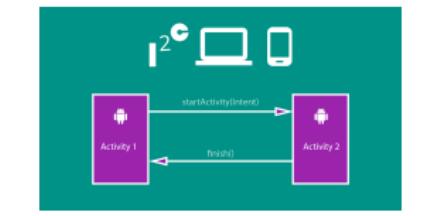


00: Introduction



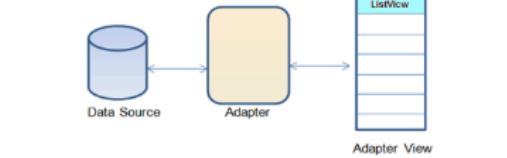
Android is a mobile operating system designed primarily for touchscreen devices

01: Activities



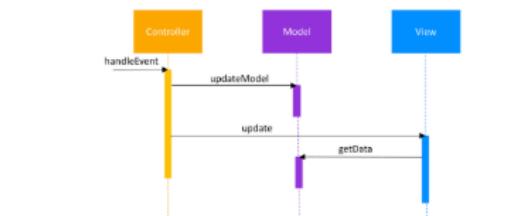
Activities are the fundamental building block of Android applications

02: Adapters



Application models are presented to activities using Adapters.

03: Models



Models capture the essential information realised by the application.

04: Images



Images can be selected and displayed from the device image gallery.

05: Maps



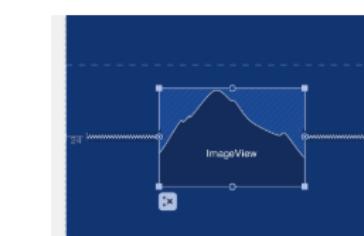
Google Maps can be integrated into an android app

06: Persistence



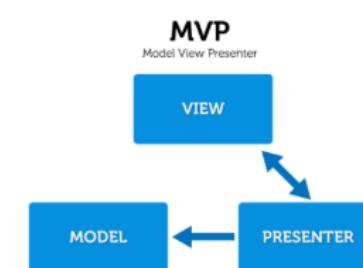
Models can be saved/recovered by the application.

07: Layouts



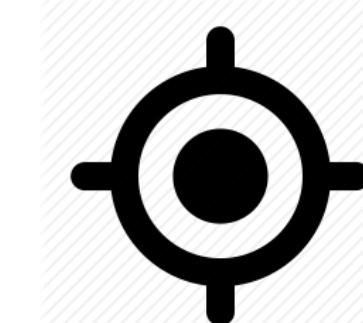
The UX is organised on screen by Layout components.

08: MVP



The Model View Presenter patterns organises the application structure

09: Location



An application can determine and track the device's current location.

10: Rooms



The Room framework is part Android Architecture Components

11: Firebase



Firebase provides Authentication and Storage cloud services

“Building Android applications is both exciting and challenging. The diversity of the applications one can build is fascinating, however their complexity can be overwhelming. Kotlin, a new language fully supported by google, dramatically enhances the power and simplicity of the programmers task.”



Kotlin is:

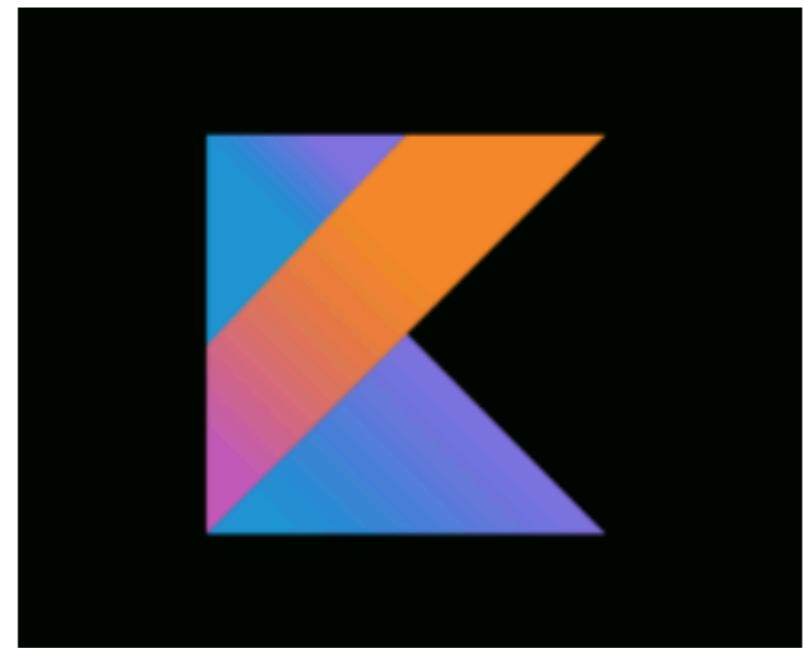
Modern

Secure

Typesafe

Expressive

Concise



Assumptions:
Intermediate level Java or
closely related language
skills

Accelerate into the
fundamentals of Kotlin we
need

Every topic explores
specific Kotlin features *in parallel* to android code

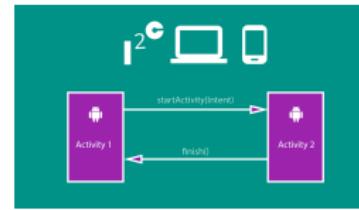
12 Topics

00: Introduction



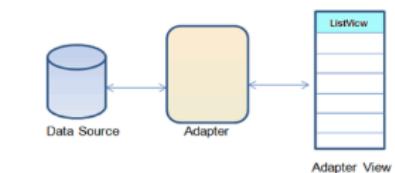
Android is a mobile operating system designed primarily for touchscreen devices

01: Activities



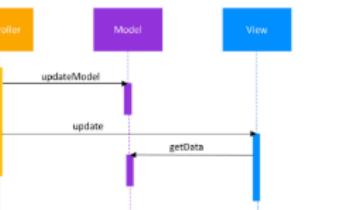
Activities are the fundamental building block of Android applications

02: Adapters



Application models are presented to activities using Adapters.

03: Models



Models capture the essential information realised by the application.

04: Images



Images can be selected and displayed from the device image gallery.

05: Maps



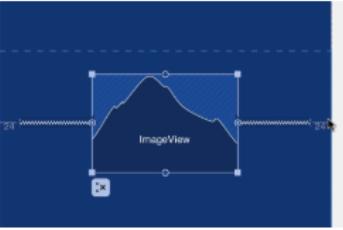
Google Maps can be integrated into an android app

06: Persistence



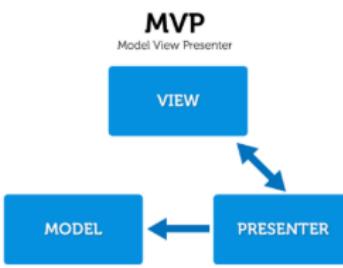
Models can be saved/recovered by the application.

07: Layouts



The UX is organised on screen by Layout components.

08: MVP



The Model View Presenter patterns organises the application structure

09: Location



An application can determine and track the device's current location.

10: Rooms



The Room framework is part Android Architecture Components

11: Firebase



Firebase provides Authentication and Storage cloud services

00: Introduction



HELLO
WORLD



Android is a mobile operating system designed primarily for touchscreen devices

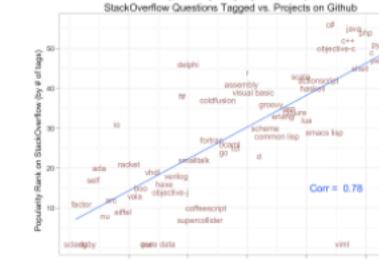
Kotlin Overview 1



Part 1 of our Kotlin Overview focusing on its origins and some important milestones.



Evolution of Kotlin



Programming language heritage, characteristics & type systems



Kotlin Overview 2



Part 2 of our Kotlin Overview taking a brief look at the main features of the language.



Kotlin in Context



A simple algorithm expressed in Java, Groovy, Swift and Kotlin



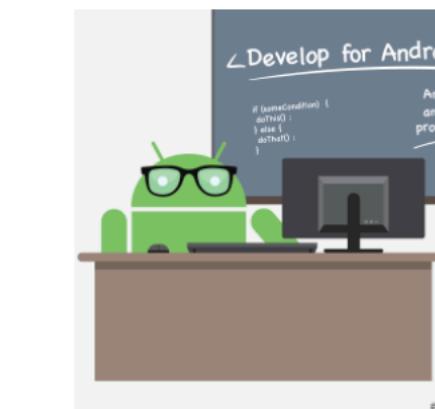
Kotlin Overview 3



Part 3 of our Kotlin Overview takes a quick look at the IDE we'll be using - IntelliJ.



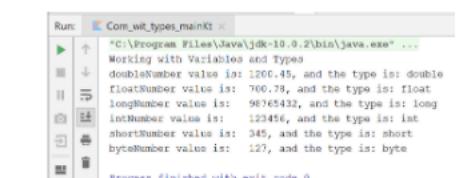
Android Overview 1



Here we take a brief look at Android, exploring its background and main features.



Lab-K00 Intro



Introduction to Kotlin and the IntelliJ IDEA.



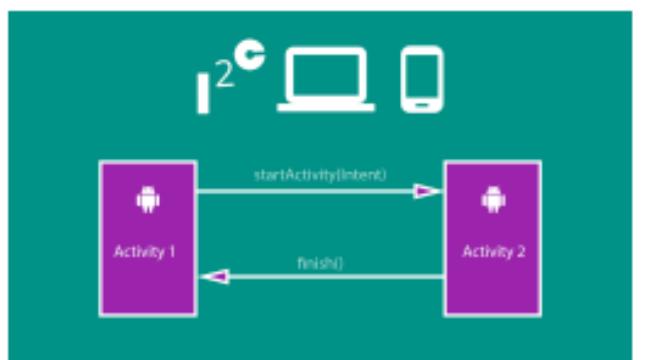
Lab A00 Studio



Download and configure Android Studio. Generate and run a sample application.



01: Activities



Activities are the fundamental building block of Android applications

Kotlin Syntax 1.1



An overview of the Kotlin syntax covering types and variables.



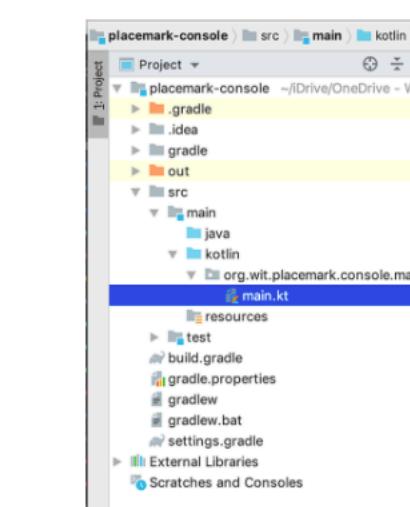
Kotlin Syntax 1.2



Here we take a look at Kotlin functions and control flow.



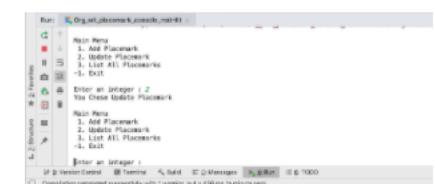
Placemark Console 1.0



A brief look at our Placemark-Console Kotlin Version 1.0



Lab-K01 Placemark-Console.1.0



Introduction to Kotlin and the IntelliJ IDEA through a console version of ****Placemark****.

Android Overview 2



The platform structure and components



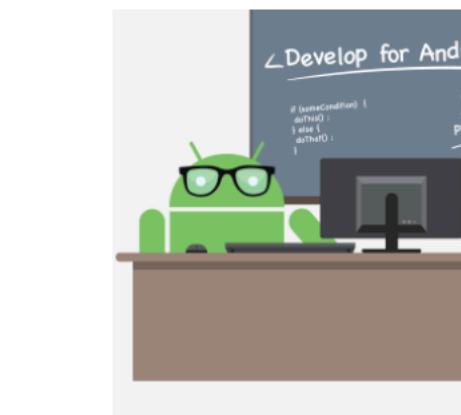
Basic Syntax I



Rapid tour of the basic syntax of Kotlin



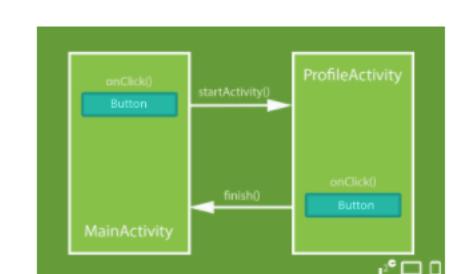
App Activities



A first look at the structure of an Activity in an Android application.



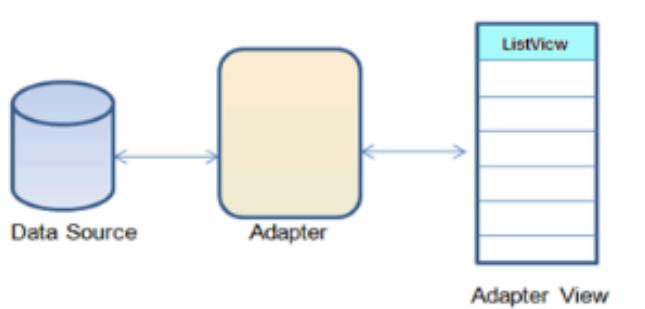
Lab-A01 Activities



Layout the PlacemarkActivity - supporting create/edit of placemarks



02: Adapters



Application models are presented to activities using Adapters.

Kotlin Syntax 1.3



Syntax- Part 1.3

Strings, Ranges and Type Checks in Kotlin are covered here.



Kotlin Syntax 1.4



Syntax- Part 1.4

A quick overview of null safety and comments in Kotlin.



Kotlin Syntax 2.1

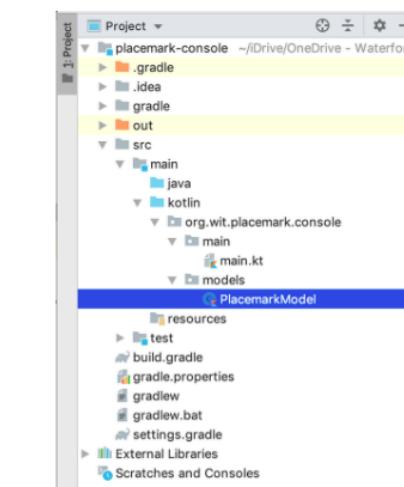


Syntax- Part 2.1

Here we focus on creating different types of Classes in Kotlin.



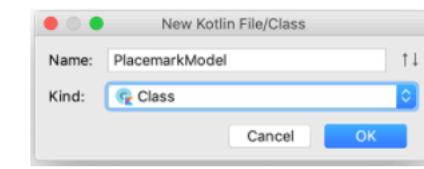
Placemark Console 2.0



A brief look at our Placemark-Console Kotlin Version 2.0



Lab-K02 Placemark-Console.2.0



A continuation of our console version of **Placemark** where we introduce a Model and Collection classes.

Basic Syntax II



More of the basics of Kotlin



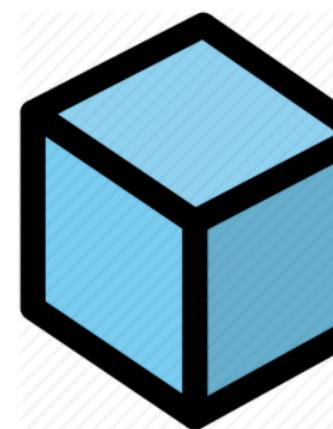
Kotlin Deep Dive



A deep dive into some key Kotlin features: Data Classes & Lambdas



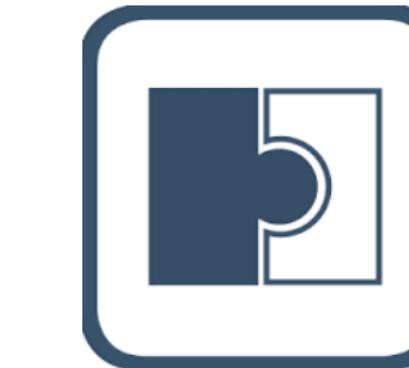
Android Application Object



This class enables a global application object to be defined, accessible from all activities



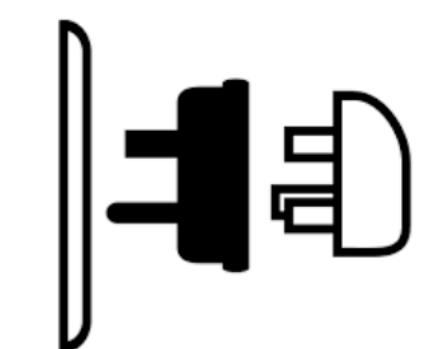
Recycler View + Adapters



Managing lists is facilitated by the Recycler View + Adapters - 2 key patterns in Android development



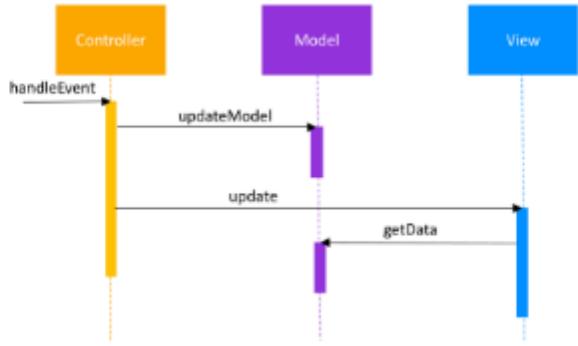
Lab-A02 Adapters



Introduce new activity to display a list of placemarks. Support adding to this list.



03: Models



Models capture the essential information realised by the application.

Kotlin Syntax 2.2



Kotlin Collections are the focus here.



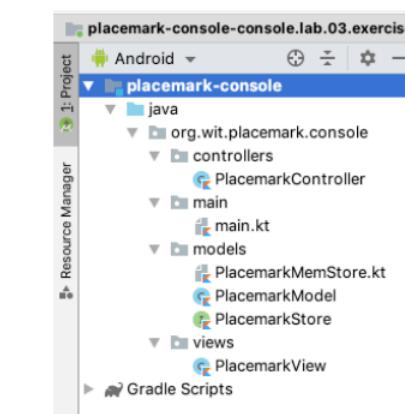
Kotlin Syntax 2.3



A continued focus on collections, classes and functional programming.



Placemark Console 3.0



A brief look at our Placemark-Console Kotlin Version 3.0



Lab-K03 Placemark-Console.3.0



A continuation of our console version of **Placemark** where we introduce a Model-View-Controller Pattern (of Sorts).



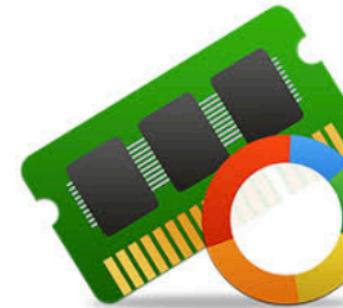
Toolbar



The Android Support Library implements AppCompatActivity + a range of general purpose components, including a Toolbar



Memory Store



Abstract the Placemarks data structure into PlacemarkStore interface + in-memory implementation.



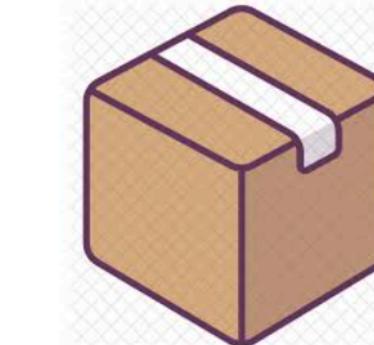
Adapter + Listener



Equip the Adapter with a Listener interface. Use this interface to communicate from the adapter to the host Activity



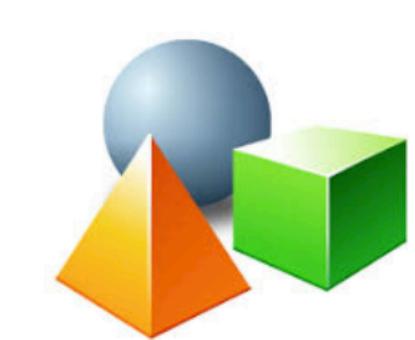
Parcelable



Encapsulate model data for transmission between Activities



Lab-A03 Models



Application object + Model classes to manage placemarks.

04: Images



Images can be selected and displayed from the device image gallery.

JS2K Basics



Basics examples:
JavaScript & Kotlin
equivalents



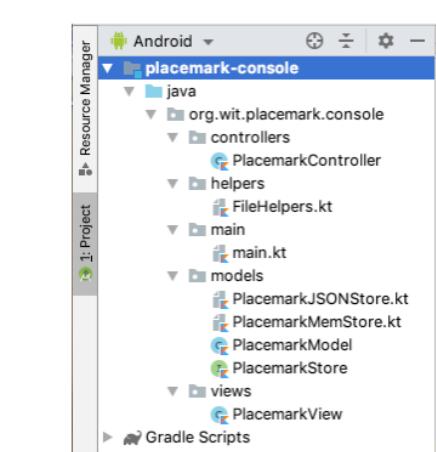
JS2K Collections



Collections and more
examples: JavaScript &
Kotlin equivalents



Placemark Console 4.0



A brief look at our
Placemark-Console Kotlin
Version 4.0



Lab-K04 Placemark- Console.4.0

```
placemarks.json
[{"id": -5289129053945747338, "title": "New York New York", "description": "So Good They Named It Twice"}, {"id": -9125691116741554, "title": "Rings of Kerry", "description": "Some more places in the Kingdom"}, {"id": 3677903504042292304, "title": "Blaa Land", "description": "just another description"}]
```

A continuation of our
console version of
****Placemark**** where we
introduce Persistence to
save our Placemarks to
disk (a JSON file).

Android Platform, Components & Activities



Key features of the
platform, the principle
components + the activity
model



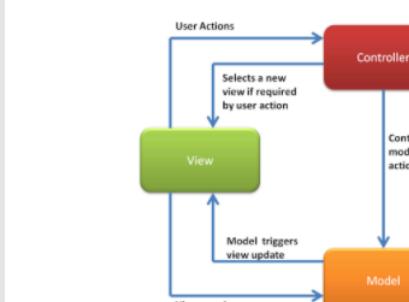
Resources



A general term for all
layout, strings, bitmaps
and other XML artefacts
in an Android Project



Model Updates



Creating and updating a
Placemark. Updating the
placemark list.



Images



Support selecting image
from phone gallery, and
then displaying them in
an activity.



Lab-A04 Images



Allow an image to be
selected from the phones
photos, and stored with
the placemark



05: Maps



Google Maps can be integrated into an android app

Introduction to Concurrency

A diagram illustrating the difference between Concurrency and Parallelism. It shows two processes, Process 1 and Process 2, each with multiple tasks represented by colored circles (green, blue, red). In Process 1, tasks are sequential (green to blue to red), representing Concurrency. In Process 2, tasks are parallel (green, blue, red all occurring simultaneously), representing Parallelism. A vertical arrow labeled 'TIME' indicates the progression of time.

A quick look at some of the concepts underpinning concurrency

[YouTube video](#)

Concurrency & Coroutines

A diagram showing a single thread represented by a blue bar. Inside the thread, there are two green arrows labeled 'FUNCTION A' and 'FUNCTION B'. One arrow has a dashed red line above it, indicating it is 'Suspended'.

A discussion on coroutines in programming

[YouTube video](#)

Coroutines in Kotlin

A smartphone screen displaying a map application. Below the phone is the Kotlin logo. The phone's screen shows a map with several location markers and route lines.

Kotlin coroutine concepts & theory

[YouTube video](#)

Kotlin Coroutine Code

```
Using Kotlin coroutines
```

```
fun position(item: Item) {
    launch(ComposableJob) {
        val post = suspendPost(item)
        processPost(post)
    }
}

suspend fun processPost(item: Item) {
    // ...
}
```

Here we look at how Kotlin implements coroutines

[YouTube video](#)

App Structure

A diagram of the Sydney Opera House with various dashed lines and arrows pointing to different parts, representing the app structure.

Key classes and relationships in the application.

[YouTube video](#)

Map Activity

A screenshot of an Android application titled 'Maps Utils Demo'. The screen shows a map of London with several red location markers. Some markers have numerical counts next to them (e.g., 10+, 20+). The map includes street names like 'Burton Road' and 'London City'. At the bottom, there are standard Android navigation buttons.

Google Map Activity can be inserted into an app via a Wizard from Studio. API Keys must be acquired from google directly.

[YouTube video](#)

Cameras & Markers

An icon of a white house inside a dark grey location marker.

These abstractions enable the map view to be manipulated programmatically, and facilitate direct manipulation by the user.

[YouTube video](#)

Lab-A05 MapActivity

A screenshot of a MapActivity showing a detailed map of a university campus. The map includes buildings labeled 'Building Library', 'Walton Building', and 'Odeon Cinema'. There are also several location markers with labels like 'IT G16', 'IT G18', 'IT G20', 'IT G21', 'IT G22', and 'IT G23'. A legend at the bottom right indicates '0-100m' and '0-500m' scales.

Include a MapActivity, enabling the user to select the location of the placemark

06: Persistence



Models can be saved/recovered by the application.

JetBrains Anko Library - Part 1



A quick look at JetBrains Anko Kotlin Library for Android



JetBrains Anko Library - Part 2



A further look at the Anko Library for Android



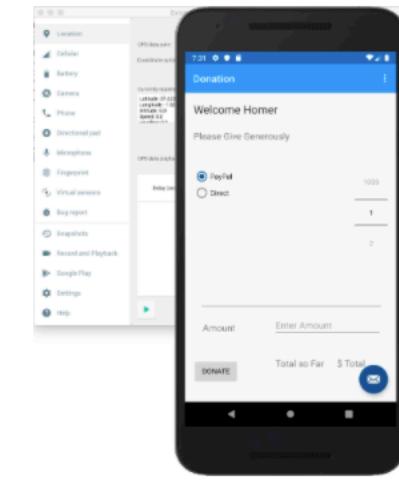
Introducing Donation



A quick look at a new Android App and some Anko Examples



Lab K06 Donation-V1



Introduce a new Case Study Donation



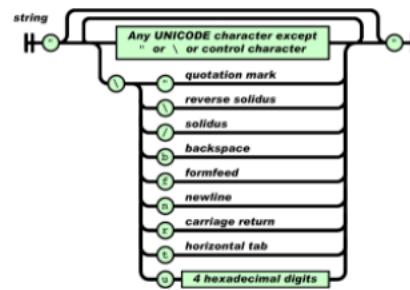
File Formats

```
<?xml version="1.0"  
encoding="UTF-8"?>  
  
<coffee objname="c1">  
  <name> mocha </name>  
  <shop> costa </shop>  
  <price> 2.0 </price>  
  <rating> 3.5</rating>  
  <favourite> 0 </favourite>  
</coffee>
```

When a simple file persistence strategy, CSV, Name/Value, YAML, XML & JSON are all potential candidate formats for file-based storage.



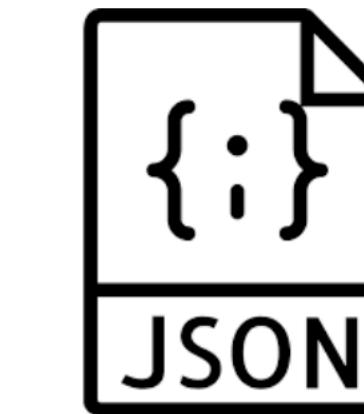
JSON



A detailed look at the syntax of JSON



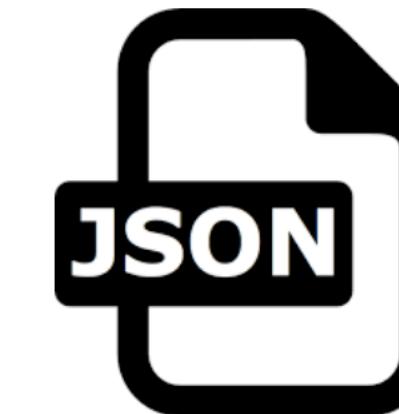
JSON Store



A new PlacemarkStore implementation - PlacemarkJSONStore - to persist placemarks to a JSON file.



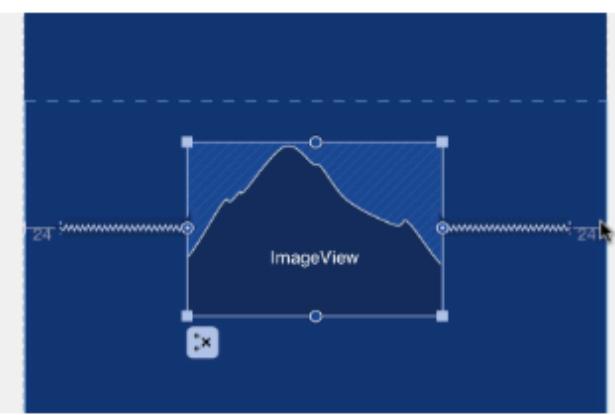
Lab-A06 JSON



Save and restore placemarks from a JSON formatted file



07: Layouts



The UX is organised on screen by Layout components.

Android Fragments

An Introduction to the use and benefits of Fragments in Android

[YouTube](#)

AndroidX Navigation

The AndroidX Navigation Library and Nav Drawer Usage

[YouTube](#)

Donation Code Walkthrough

A look at the use of Navigation and Fragments in Donation with manual setup

[YouTube](#)

DonationX Walkthrough

A look at the use of Navigation and Fragments in Donation with the Navigation Library

[YouTube](#)

Lab K07a Donation-V2

Introduce a Nav Drawer and Fragments into Donation

[YouTube](#)

Constraint Layout

ConstraintLayout allows you to create large and complex layouts with a flat view

[YouTube](#)

Overall Map Activity

Introduce an activity to display all placemarks on a single map.

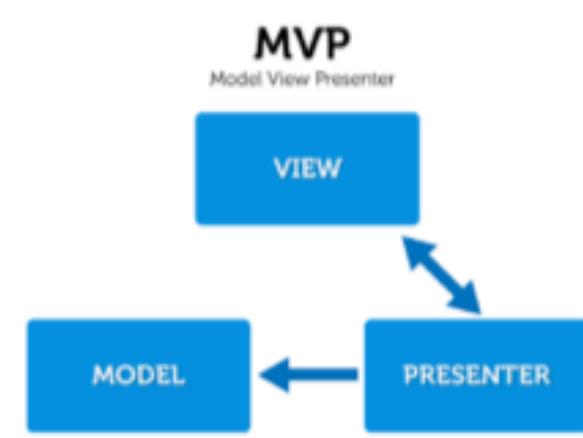
[YouTube](#)

Lab-A07 MapView

Display all placemarks on a map in a new activity



08: MVP



The Model View Presenter patterns organises the application structure

09: Location



An application can determine and track the device's current location.

10: Rooms



The Room framework is part Android Architecture Components

11: Firebase



Firebase provides Authentication and Storage cloud services



Lab A00 Studio



Download and configure
Android Studio. Generate
and run a sample
application.

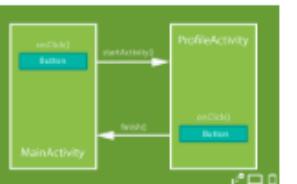


Lab-A04 Images

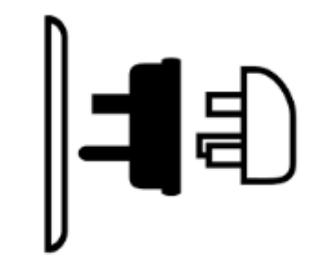


Allow an image to be
selected from the phones
photos, and stored with the
placemark

Lab-A01 Activities



Layout the
PlacemarkActivity -
supporting create/edit of
placemarks



Introduce new activity to
display a list of placemarks.
Support adding to this list.

Lab-A02 Adapters

Lab-A03 Models



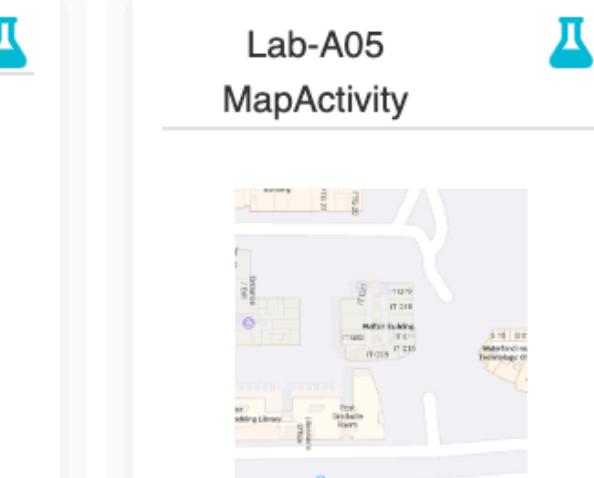
Application object + Model
classes to manage
placemarks.

30 Labs

Lab-A04 Images



Lab-A05 MapActivity

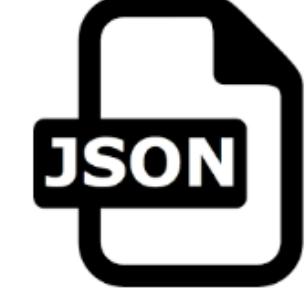


Lab-A08a MVP I



Refactor Activities to use
the Model View Presenter
pattern

Lab-A06 JSON



Lab-A07 MapView



Display all placemarks on a
map in a new activity

Lab-A08b MVP II



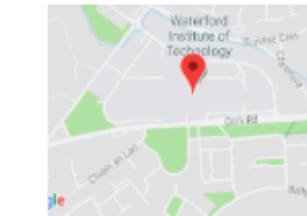
Complete the presenter
pattern imple
introducing B
BasePresen

Lab-A10a AndroidX



Migrate to AndroidX +
manage library versions
more optimally

Lab-A09a Location



When creating a new
placemark, use the current
location as the starting

Lab-A09b Tracking



Extend the location facility
to track location in real time.

Lab-A10b Rooms



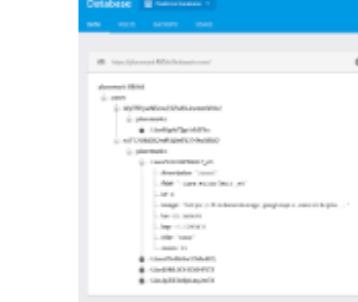
Introduce a new
PlacemarkStore
implementation to persist to
an SQLite database

Lab-A11a Firebase Auth



Authenticate users against
the Firebase Authentication
service

Lab-A11b Firebase Database



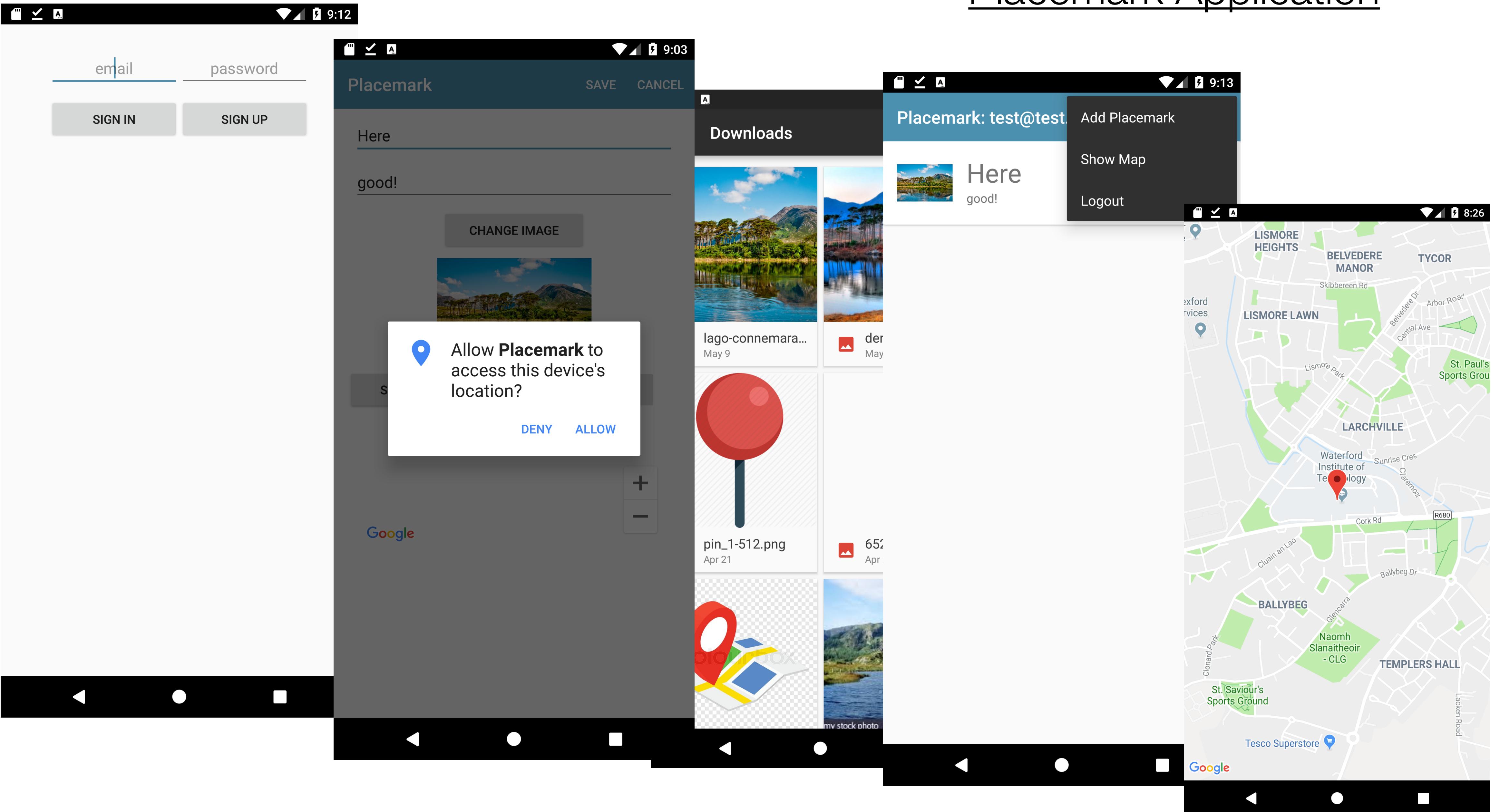
Store placemarks in
Firebase Realtime
Database

Lab-A11c Firebase Storage



Store images in Firebase
Storage

Placemark Application



2020

Semester 2		S	M	T	W	T	F	S	Modules
		1	31	1	2	3	4	5	mobile app dev
September	1	2	6	7	8	9	10	11	mobile app dev
		3	13	14	15	16	17	18	mobile app dev
		4	20	21	22	23	24	25	mobile app dev
		5	27	28	29	30	1	2	
October	reading-week	6	4	5	6	7	8	8	mobile app dev
		7	11	12	13	14	15	16	mobile app dev
		8	18	19	20	21	22	23	mobile app dev
		9	25	26	27	28	29	30	
November	reading-week	10	1	2	3	4	5	6	
		11	8	9	10	11	12	13	mobile app dev
		12	15	16	17	18	19	20	mobile app dev
		13	22	23	24	25	26	27	mobile app dev
December	reading-week	14	29	30	1	2	3	4	
		15	6	7	8	9	10	11	mobile app dev
		16	13	14	15	16	17	18	mobile app dev
		17	20	21	22	23	24	25	
January	18	19	27	28	29	30	31	1	
		20	3	4	5	6	7	8	
February	1	21	28	29	30	31	1	2	
		22	3	4	5	6	7	8	

Mobile App Development



layouts · activities · resources · lifecycle · widgets · ux

Project




proposal · specification · plan · iterations · implementation · report · demo

MONDAY

TUESDAY

WEDNESDAY

THURSDAY

FRIDAY

10:45

10:45

12:15

12:15

Mobile App Development
Webinar
12:15-2:00

2:00

Mobile App Development
Webinar
12:15-2:00

13:45