Poznań University of Technology

Faculty of Control, Robotics and Electrical Engineering

Institute of Robotics and Machine Intelligence

Division of Control and Industrial Electronics

Mobile application
REST client

Mobile and embedded applications for Internet of Things

Teaching materials for laboratory

Dominik Łuczak, Ph.D.; Adrian Wójcik, M.Sc.

Dominik.Luczak@put.poznan.pl
Adrian.Wojcik@put.poznan.pl

I. Goal

Knowledge

The aim of the course is to familiarize yourself with:

- role of mobile devices in IoT systems,
- development tools for Android OS,
- basics of Java / Kotlin language in the context of mobile app *logic*,
- basic XML tags in the context of mobile app *GUI*.

Skills

The aim of the course is to acquire skills in:

- creating REST client apps for mobile devices,
- implementation of simple GUI of mobile apps,
- implementation of unit tests of your code,
- correct debugging of mobile apps.

Social competences

The aim of the course is to develop proper attitudes:

- strengthening the understanding of the role and application of network communication in IT systems and related security issues,
- strengthening understanding and awareness of the importance of non-technical aspects and effects of the engineer's activities, and the related responsibility for the decisions taken,
- proper technical communication in context of software development,
- choosing the right technology and programming tools for the given problem,
- analyzing technical problem from *user* point of view.

II. Laboratory report

Complete laboratory tasks as per the instructor's presentation. Work alone or in a team of two. **Keep safety rules while working!** Prepare laboratory report documenting and proving the proper execution of tasks. Editorial requirements and a report template are available on the *eKursy* platform. The report is graded in two categories: tasks execution and editorial requirements. Tasks are graded as completed (1 point) or uncompleted (0 points). Compliance with the editorial requirements is graded as a percentage. The report should be sent as a *homework* to the *eKursy* platform by Tuesday, May 4, 2021 by 23:59.

Prepare **single laboratory report** from instructions *L06* and *L07*.

III. Prepare to course

a) Know the safety rules

All information on the laboratory's safety instructions are provided in the laboratory and on Division website [1]. All inaccuracies and questions should be clarified with the instructor. It is required to be familiar with and apply to the regulations.

Attend the class prepared. Knowledge from all previous topics is mandatory.

b) Introduction to Android OS

Android is a mobile operating system based on a modified version of the Linux kernel and other *open source* software, mainly intended for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as Open Handset Alliance [2] and commercially sponsored by Google. The first commercial Android device was launched in 2008.

It is free and open source software. Its source code is known as Android Open Source Project (AOSP) [3], which is licensed under the Apache license [4]. Most Android devices ship with additional software pre-installed, most notably Google Mobile Services (GMS) [5] which includes basic applications such as Google Chrome, the digital distribution platform Google Play and associated Google Play Services development platform. Google has developed several varieties of Android for specific applications, including Wear OS, for wearable devices such as wristwatches, Android TV for TVs, Android Things for smart devices and **Internet of Things**, Android Auto for cars, and Daydream, a virtual reality platform.

Android has been the best-selling smartphone operating system since 2011, and the best-selling tablet operating system since 2013. The current stable version is **Android 11**, released in 2020.

On Android, in addition to the Linux kernel, there are middleware, libraries and APIs written in C, and an application framework, including Java-compatible libraries. Development of the Linux kernel continues independently of Android's other source code projects. In Fig. 1. shows the structure of the Android operating system [6].

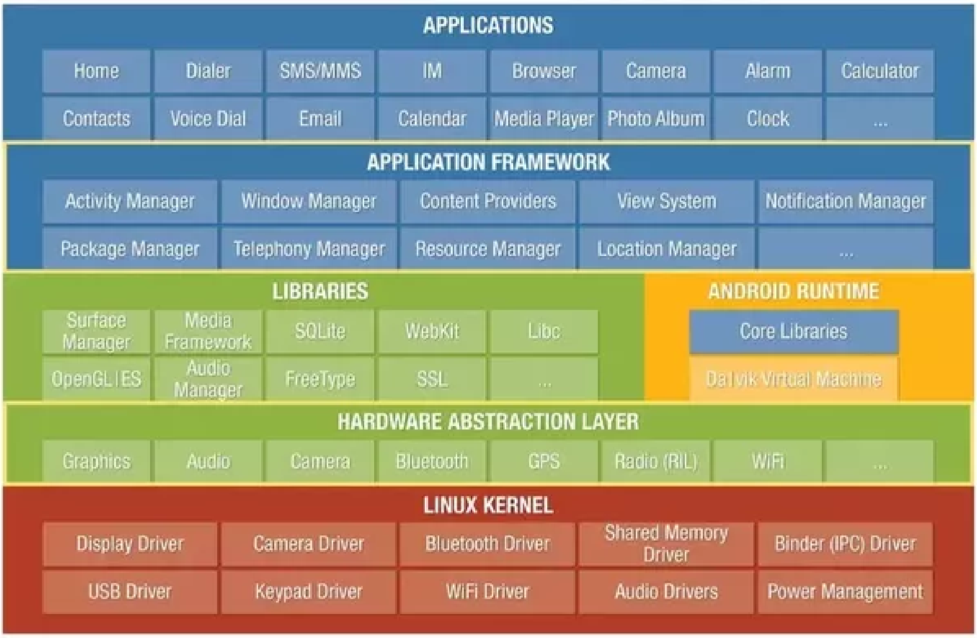

Fig. 1. Android operating system architecture: software layers [6].

c) Introduction to Android Studio IDE

Android Studio is an official integrated development environment (IDE) for Android application development, based on IntelliJ IDEA [7]. Android Studio is available for download **for free** on website developer.android.com for Windows, Mac, Linux and Chrome OS. Android Studio includes a software development kit (SDK) for creating mobile applications in Java [8] and Kotlin [9]. It also supports creating modules in C++ [10]. In Android Studio user can perform all basic stages of software development: code editions, app source code compilations, as well as running and debugging mobile app. It also includes *drag and drop* tools to help you create a graphical user interface and generate an XML file with GUI layout description [11]. By default, the Android Emulator is installed with Android Emulator, which allows you to create a virtual machine with the Android system to run and debug mobile apps.

Android Studio is based on Gradle scripts. Gradle is a *open source* build automation tool that is designed to be flexible enough to create almost any type of software. It is a multi-language software development automation tool. It can control the development process in building, testing, deploying, and publishing software. Supported languages include Java (Kotlin, Groovy, Scala), C and C++, and JavaScript [12]. This means that configuring a project in Android Studio, e.g. adding an external library, will usually be done by editing a `build.gradle` script.

d) Examples

Sample Android Studio projects are available in repository github.com/adrianwojcikpp/AMiWdIP-L. Before performing laboratory tasks, see examples: *Data Grabber* and *LED Display Control*. Both applications are examples of simple REST *clients* for an embedded system web *server*. The *Data Grabber* application is an example of a client *synchronously* downloading measurement data from the server using the GET method. The *LED Display Control* application is an example of a client *asynchronously* sending control data to the server using the POST method.

e) Volley HTTP library

Volley is a library for network communication with HTTP protocol available for the Android platforms in Java and Kotlin. It easily integrates with any parent protocol: supports raw strings, images, and **JSON**. By providing built-in support for needed functions, Volley relieves developer of the burden of writing standard, *boilerplate* code and allows to concentrate on the target application-specific logic [13][14].

NOTE! Use of network communication in the Android mobile app requires appropriate permissions in the manifest file: ACCESS_NETWORK_STATE and INTERNET [15].

IV. Scenario for the class

a) Teaching resources

Hardware	• computer, • mobile device (e.g. Android phone),
Software	• mobile apps IDE (e.g. Android Studio IDE). • mobile device emulator (e.g. Android Virtual Device), • web server with CGI scripts support (e.g. XAMPP), • code editor (e.g. Notepad++, VS Code),

b) Tasks

Familiarize yourself with examples of mobile applications available in GitHub repository. Run and test applications with the use of sample PHP scripts.

1. Create a simple REST client application for a mobile device (smartphone, tablet).
   (a) Application should consist of a single activity containing three interface elements: *output* text view, *input* text view and a button. All elements should be clearly labeled.
   (b) The input text view should be used to specify the URL of the server resource.
   (c) After pushing the button client should request resource from URL.
   (d) The output text view should contain the server's response.

References

1. *Regulations, health and safety instructions* [online]. [N.d.] [visited on 2019-09-30]. Available from: http://zsep.cie.put.poznan.pl/materialy-dydaktyczne/MD/Regulations-health-and-safety-instructions/.
2. *Android Overview | Open Handset Alliance* [online] [visited on 2021-04-12]. Available from: https://www.openhandsetalliance.com/android_overview.html.
3. *Android Open Source Project* [Android Open Source Project] [online] [visited on 2021-04-12]. Available from: https://source.android.com/?hl=pl.
4. *Apache License, Version 2.0* [online] [visited on 2021-04-12]. Available from: https://www.apache.org/licenses/LICENSE-2.0.html.
5. *Android – Usługi Google Mobile* [Android] [online] [visited on 2021-04-12]. Available from: https://www.android.com/intl/pl_pl/gms/.
6. *Platform Architecture* [online] [visited on 2021-04-12]. Available from: https://developer.android.com/guide/platform.
7. *IntelliJ IDEA: The Capable & Ergonomic Java IDE by JetBrains* [JetBrains] [online] [visited on 2021-04-12]. Available from: https://www.jetbrains.com/idea/.
8. *Jak uzyskać oprogramowanie Java dla urządzeń mobilnych?* [Online] [visited on 2021-04-12]. Available from: https://www.java.com/pl/download/help/java_mobile.html?printFriendly=true.
9. *Kotlin and Android* [Android Developers] [online] [visited on 2021-04-12]. Available from: https://developer.android.com/kotlin.
10. *Add C and C++ code to your project* [Android Developers] [online] [visited on 2021-04-12]. Available from: https://developer.android.com/studio/projects/add-native-code.

11. *Layouts* [Android Developers] [online] [visited on 2021-04-12]. Available from: https://developer.android.com/guide/topics/ui/declaring-layout.
12. *Gradle Build Tool* [Gradle] [online] [visited on 2021-04-12]. Available from: https://gradle.org/.
13. *Volley overview* [Android Developers] [online] [visited on 2021-04-12]. Available from: https://developer.android.com/training/volley.
14. *google/volley* [online]. Google, 2021 [visited on 2021-04-12]. Available from: https://github.com/google/volley. original-date: 2017-01-24T17:39:33Z.
15. *App Manifest Overview | Android Developers* [online] [visited on 2021-04-12]. Available from: https://developer.android.com/guide/topics/manifest/manifest-intro.