Mobile Application Development

Radosław Maciaszczyk

The market of mobile devices especially smartphones is growing very fast. Numerous users of these devices causes that the mobile application market is also growing very fast. Universities and colleges have noticed this trend and courses related to mobile applications programming are present in most computer science curricula.

On the market we have a choice between practically two systems, Android and iOS. In Q3 2021, Android is installed on over 72% of devices [1]. Due to greater popularity and lower cost of developing applications, this course concerns programming in Android environment.

Android is an open source operating system for mobile devices and a corresponding open source project led by Google [2]. Android Open Source Project (AOSP) repository offer the information and source code needed to create custom variants of the Android OS, port devices and accessories to the Android platform, and ensure devices meet the compatibility requirements that keep the Android ecosystem a healthy and stable environment for millions of users.

As an open source project, Android's goal is to avoid any central point of failure in which one industry player can restrict or control the innovations of any other player. To that end, Android is a full, production-quality operating system for consumer products, complete with customizable source code that can be ported to nearly any device and public documentation that is available to everyone.

There are not much requirements to start developing applications. Generally you need knowledge of object oriented language, For Android it is (JAVA, KOTLIN).

The freeware Android Studio is used for programming. Development environment is available for multiple platforms including Windows, Mac, Linux, Chrome OS [3].

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as [4]:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices

DOI: https://doi.org/10.11118/978-80-7509-898-6-0083

- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

Android is an operating system that is constantly evolving. This is undoubtedly an advantage, but continuous improvement requires constant development of programming skills. However, it is possible to distinguish the basic range of knowledge needed to develop applications. The following elements are isolated in the course provided by this project:

- System components
- User interaction
- Sensor handling
- Data exchange

The course consists of 45 hours of classes taught in lecture (15h), laboratory (15h) and project (15h).

Lecture:

- Introducing to mobile device and mobile systems.
- Application Fundamentals, components – activities, services, broadcast receivers, content providers
- Component lifecycle – activity, fragments, services
- User Interface, Introduction to Material Design, typography, main component
- Sensors, GNNS – use case of sensors, type of sensors, sensors lifecycle
- Threads and Services, Class – Runnable, Jobs Scheduler, Intent Service, AsyncTask
- Data persistence – Room Database, Using SD Card, SharedPreference class
- Networking, using sockets and HTTP connections
- Google firebase for Android – Cloud Messaging, Cloud firestore,
- Android Performance

Laboratory:

- Configure the Development Environment, Create first program. Debug programs
- Create interactive user interface. Introduction to widgets
- Activities and Intents
- Layouts. Using RecyclerView to display data
- Data persistence
- Sensors and Location
- Services, Notifications
- Networking

Project:

- Introduction to project, project's functions
- Working with own Project
- Documentation
- Presentation project

References

1. https://gs.statcounter.com/os-market-share/mobile/worldwide
2. https://source.android.com/
3. https://developer.android.com/studio
4. https://developer.android.com/studio/intro

Author

dr inż. Radosław Maciaszczyk, West Pomeranian University of Technology in Szczecin, e-mail: Radoslaw.Maciaszczyk@zut.edu.pl