VITALI OLSHEVSKI

Software Developer, Android Mobile Applications

Minsk, Belarus job@olshevski.dev olshevski Russian, English

Developing Android apps since 2011. I enjoy writing code that makes sense, creating perfectly aligned UIs and getting to the bottom of how things work.

SKILLS

Languages
Java, Kotlin, C/C++

Libraries/APIs
Android Jetpack, Compose, Coroutines/Flow, Dagger/Anvil/Hilt, Koin, RxJava 2, Retrofit/OkHttp, Gson, Moshi, Glide, Firebase APIs, Camera/Camera2 API, JNI, Gradle, fastlane, Kotlin Symbol Processing

Testing
Junit, Espresso, Robotium, Robolectic, Mockito, Kotest

Databases
SQLite, SQLCipher, Room, Realm

Limited experience with
ExoPlayer, OpenGL ES, OpenCV

EXPERIENCE

Touchlane, July 2018 – present
On-demand project-based work as a Senior Android Developer

Panda Systems, April 2017 – August 2017
Worked on a startup project in the position of a Lead Android Developer

Softeq Development, December 2011 – January 2016
Took part in numerous projects for a variety of well-known customers, such as Nvidia, Nike and Blizzard. Initially started working as an intern, but through years got enough experience to establish oneself as a technical lead.

Burstly, March 2011 – January 2012
Worked as a Junior Android Developer on advertising and monetization SDK and administrated CI

EDUCATION

Belarusian State University, 2007 – 2012
Applied Mathematics and Computer Science, Bachelor's Degree

FEATURED PROJECTS

Compose Navigation Reimagined, January 2022 – March 2022

Personal open-source project. The library serves as a type-safe alternative to the official Navigation Component for Compose:

https://github.com/olshevski/compose-navigation-reimagined

NatureID, September 2020 – February 2021

Plant identification application with a built-in wiki, watering schedules and social media features.

https://play.google.com/store/apps/details?id=plant.identification.flower.tree.leaf.identifier.identify.cat...

Responsibilities: general development of new features, refactoring and optimization of crucial app parts, such as networking and image processing

Tech stack: Kotlin, Coroutines, JNI, C/C++, Retrofit/OkHttp, Moshi, Glide, Room, Koin

OmaMobiili, July 2018 – April 2019

Mobile banking application for the Finnish bank OmaSP. The source code of the existing application was transferred from a previous developer in a poor condition.

https://play.google.com/store/apps/details?id=fi.omasp.mobile.bank

Responsibilities:

- refactoring of the legacy codebase and creation of a new supportable architecture
- reimplementation of the whole UI layer according to the new design
- reimplementation of many app features from scratch

I also led a small local team of Android developers and served as a representative in communication with the remote management team.

Tech stack: Java, Retrofit/OkHttp, Gson, Dagger, RxJava 2, Android Data Binding

Supercan, April 2017 – August 2017

Startup messaging application where users could request registered influencers, celebrities and experts for paid personalized answers in a text or video form.

I led the app development from the very start of the project and implemented a large part of the overall functionality.

Tech stack: Java, Retrofit/OkHttp, Gson, Realm, Camera2 API, ExoPlayer, Firebase Cloud Messaging

SHIELD Camera, December 2014 – January 2016

Built-in camera application for Nvidia Shield series of tablets. AOSP Camera fork with lots of new proprietary features.

Responsibilities:

- refactoring of the original AOSP Camera source code and migration to Camera2 API
- implementation of various features and integration of proprietary libraries and APIs
- functional testing, stress testing, stabilization, performance and memory optimization
- team leading and communication with Nvidia managers and developers

Tech stack: Java, JNI, C/C++, Camera/Camera2 API

Camera Awesome, December 2012 – October 2014

Feature-rich camera application. It supported a wide range of devices, as well as being pre-installed on Nvidia Shield series of tablets as the default camera application.

Responsibilities:

- UI implementation with support of different screen-sizes
- implementation of photo editing, live-preview photo and video filters (based on OpenGL shaders)
- extensive stabilization process due to a large number of supported devices

Tech stack: Java, JNI, C/C++, OpenGL ES, EGL, GLSL, Camera/Camera2 API