

IOS Developer

Alexander Rakhimov

\$2000

I can recommend myself as a proactive mobile software engineer. I will constantly suggest improvements. I am a team player and have good team-lead skills. I can quickly decompose tasks and suggest optimal solutions. Independent. I know how to find balance between perfectionism and speed of development. I will improve your developers team and the team as a whole.

Contacts

Email: alexanderrkhmv@gmail.com

Phone: +995 551 14 64 57

Linkedin: <https://www.linkedin.com/in/alexander-rakhimov-026374263/>

Website: <https://alejandrorkhmv.github.io/>

Languages

Russian - NS

English - Intermediate

Experience

12.2021 - 10.2022

QA Engineer **Winvestor**

Expertise

IOS SDK, Swift

UIKit, SwiftUI, SnapKit, Storyboards/Xib, Anchors/VFL

URLSession/AF/Async Await

Design Patterns, MVC/MVP/MVVM/VIP/VIPER, SOLID

CoreData, UserDefaults

CocoaPods/SPM, DI - Swinject

NSThread/GCD/OperationQueue, ARC

Unit/UI tests TDD

Combine

Skills

1. According to the plan, I studied the syntax of the Swift language. I took information on SwiftBook.org and YouTube.

I took a course on algorithms and data structures on SwiftBook.

I wrote games that you can play on the console - tic-tac-toe, checkers, chess, sea battle.

2. Then I studied UIKit. I started working with Storyboards - the result of the "Time Calculator" is on my GitHub <https://github.com/AlejandroRkhmv/MyProjects>

3. I learned to work without using Storyboards and apply the MVC architecture - the result is an application that shows the time. I use UserDefaults in it to save the font color that the user has selected. I have published the application in the AppStore by the link <https://apps.apple.com/ge/app/whattimeisit/id1672860073>

4. I learned how to make custom views and work with the API. I took a Networking course on SwiftBook. The result was an application that interacts with the SpaceX API <https://github.com/AlejandroRkhmv/SpaceXAstronauts>

5. I studied the architecture of MVP + Router + AssemblyBuilder. The result is an application that works with the NASA API and shows a picture with a description to the date that the user enters. In addition, I took a course on using CoreData and used CoreData in this application to compile a table of images that the user uploaded: <https://github.com/AlejandroRkhmv/APOD>

6. I finished courses on Multithreading, GSD, NSOperationQueue.

7. I worked with the VIPER architecture. The result was an application that downloads images using GoogleAPI. In this application I also use WKWebView and Cache. <https://github.com/AlejandroRkhmv/imageSearchEngine>

8. I studied DesignPatterns, SOLID principles. I try to apply the acquired knowledge when it is needed. I understood DI and studied the information on the Swinject framework, but have not used it yet.

9. When working on these applications, I actively used UITableViewControllers and UICollectionViewControllers. And also, I have honed the skills of using closures, delegates and notification centers to transfer various data between controllers.
10. I learned how to make up code. I prefer to use Anchors and SnapKit. But I can work with NSLayoutConstraint and Visual Format Language. You can also use Storyboards, but I've heard that it's not very convenient in team development.
11. I completed a course on writing tests as well as on test driven development.
12. I worked with CocoaPods dependency manager.
13. I completed 3 courses on SwiftUI. I liked it more than UIKit. I can draw pictures of any complexity, | make custom tab bars and views. Now I'm mastering the skill of using SwiftUI and the MWM + Coordinator architecture to write my next application, which will also receive data from the network. Among others, I will use frameworks such as WeatherKit and CoreLocation in it.
14. I studied the VIP architecture (Clean Swift) and I also plan to write an application using this architecture for a portfolio.
15. After the 13th and 14th points, I plan to take courses on Combine. I have already touched on it in a slight way and tried it, but not so confidently yet.
16. I can understand someone else's code, I am diligent, purposeful.