

Mihaela Mihaljević Jakić

Senior iOS Architect & Developer

Experienced iOS engineer with a deep commitment to modular architecture and elegant code. I specialize in OpenAPI-driven development, building robust, multi-platform Swift applications that scale gracefully. My work is defined by clean structure, extreme attention to testability, and the kind of modularization that makes features easy to extend, reason about, and maintain. I take pride in shipping polished, production-grade software that balances design clarity with engineering precision.

mihaelamj@me.com

+385995491157

Zagreb, Croatia

[LinkedIn](#)

[GitHub](#)

[Website](#)

University of Zagreb, Faculty of Humanities and Social Sciences, M.Sc. in Information and Speech Science

EXPERIENCE

iOLAP (Nov 2022 - Mar 2025) – Senior Senior iOS Developer

ZUMIEZ Stash App

- Worked on the Zumiez Stash app as a Senior iOS Engineer
- Created Swift network client code generator from OpenAPI specs
- Refactored legacy applications and unit-tested critical components
- Implemented a custom lightweight GraphQL client without external dependencies, featuring type-safe request handling, efficient query utilities, custom GraphQL schema parsing, and sophisticated error management
- Designed and implemented a modular filtering system with dynamic UI components, supporting complex data sorting and real-time updates
- <https://www.zumiez.com>
- https://apps.apple.com/app/apple-store/id1163863113?pt=118023521&ct=1m2501_web_header_rbn_v
- | Swift | REST | OpenAPI | GraphQL | UIKit | UI in code | Swift Package Manager | Unit Testing |

AI Chat SDK

- Developed a cross-platform chat SDK for iOS and macOS using Swift and SwiftUI for an AI chatbot
- Implemented real-time messaging, speech recognition, and UI components with modern architecture patterns
- Designed using modular architecture with 10+ local packages handling UI styling, message management, socket communication, and speech recognition
- Built with Combine for reactive programming, protocol-oriented design, and comprehensive testing infrastructure
- Implemented platform-specific components with shared protocols ensuring consistent behavior
- | WebSockets | Swift | UIKit | Artificial Intelligence | Swift Package Manager | macOS | Unit Testing | AppKit |

iRobot App

- Objective-C, Swift, SwiftUI, UIKit, C++
- Added features for the onboarding process
- Automated localization
- | Swift | SwiftUI | Swift Package Manager | Objective-C | UIKit | UI in code | Unit Testing |

Code Weaver (May 2024 - Dec 2024) – Senior Senior iOS Developer

Anonymous Big EU Project

- Refactored client backend OpenAPI specs and built a Swift networking package
- Reorganized and clarified existing OpenAPI specifications to better reflect backend capabilities and client requirements. Ensured consistency and accuracy across endpoints and models
- Created a modular Swift package to encapsulate networking logic, designed for clean separation of concerns, testability, and reuse across projects
- Integrated Swift OpenAPI Generator to automatically generate type-safe API client code from OpenAPI specs, enabling seamless updates and minimizing manual maintenance
- Streamlined the build process to regenerate client code automatically upon spec changes, aligning with CI/CD pipelines and reducing the risk of contract drift
- | Swift | SwiftUI | REST | OpenAPI | Swift Package Manager | Unit Testing | UI Testing | Structured Concurrency |

Wheels Up

- Modernized and refactored legacy Objective-C codebase to Swift
- Improved maintainability and fixed bugs
- Simplified login flow to enhance user experience and reduce authentication friction
- Developed custom UIKit components to improve app performance and user interface
- <https://wheelsup.com>
- <https://itunes.apple.com/us/app/wheels-up/id956615077>
- | Objective-C | Swift | UIKit | UI in code | Push Notifications | REST | AFNetworking | OpenAPI | Unit Testing |

Birch Finance

- Refactored a large iOS app's UI and networking layer using modern Swift
- Implemented MVVM architecture with a clean separation of concerns
- Built a comprehensive networking library with complete test coverage
- Organized code into feature-based modules
- Improved code quality through formatting, linting, and testing infrastructure
- <https://www.birchfinance.com>
- | Swift | REST | UIKit | CoreAnimation | UI in code | Unit Testing | UI Testing |

Budtz Innovation

- Led the refactoring of a legacy iOS app's UI and networking stack using modern Swift paradigms
- Reorganized the codebase into scalable, feature-driven modules to improve team velocity and onboarding
- | Swift | REST | UIKit | UI in code |

Kindergarten (Cancelled)

- iOS app for a Croatian company, for managing kindergartens
- Developed 90% of the app
- Designed REST APIs and implemented them in the app
- Implemented custom UI design code

- | Objective-C | UIKit | UI in code | AFNetworking | AutoLayout | REST | Carthage | Unit Testing | CocoaPods |

Whatt - social network

- iOS app for a social network
- Developed custom expandable TextView with links and tagging
- | Objective-C | UIKit | UI in code | AFNetworking | AutoLayout | REST | Carthage | Unit Testing | CoreAnimation | CocoaPods |

Christian Resources

- Universal iOS app (iPhone and iPad), for a client through oDesk
- Developed the application (100%)
- Created design in Sketch and implemented it in the application, with dynamic UI for iPhone 4, iPhone 5, iPhone 6, iPhone 6 Plus, and iPad
- Used Core Animation, Auto Layout, AFNetworking, remote audio streaming
- Client changed parts of the UI afterward
- <https://itunes.apple.com/us/app/bible-study-tools-christian/id600610494>
- | Objective-C | UIKit | UI in code | AFNetworking | AutoLayout | REST | Carthage | Unit Testing | AVFoundation | CoreGraphics |

ServicePal

- Added custom features using Objective-C, UIKit
- Dynamic Content Framework Developer – designed and implemented a template-based dynamic content creation system that allowed for real-time content generation and updates without requiring app redeployment
- Increased development efficiency by 35%
- <https://servicepal.com>
- | Objective-C | UI in code | UIKit | CoreData | REST | Unit Testing |

Breck World

- Built app from scratch in Swift 4+
- Built UI in code using UIKit and Auto Layout
- Implemented networking layer using URLSession
- ARKit integration
- | Swift | UIKit | UI in code | CoreData | AutoLayout | REST | ARKit |

Birthdayrama App

- A fun way to share birthday wishes with friends and family
- Refactored the entire app
- Created a form factory for the app which uses many screens with input fields
- Used the latest Auto Layout best practices
- <https://www.producthunt.com/products/birthdayrama>
- | Swift | UIKit | UI in code | CoreData | AutoLayout |

Huxly App

- Newsreader reimaged
- Added many features and new screens to the app
- Login / Sign up / Forgot password screens (dynamic form screens)
- Demographics screens (dynamic radio control screens using Core Animation)
- Home and dynamic filter screens
- Sharing functionality
- | Swift | UIKit | UI in code | CoreData | AutoLayout | REST | CocoaPods |

Purch (Mar 2015 - Oct 2018) – Mid Mid iOS Developer

Consumr App

- Adapted existing client application for iOS 8+
- Performed detailed gap analysis for the application, including iOS frontend and REST API backend differences
- Implemented custom UI design code (from design sheets)
- REST API implementation
- Used Core Animation, Auto Layout, AFNetworking, Push Notifications
- <https://x.com/purchxapp>
- | Objective-C | UIKit | UI in code | Push Notifications | CoreAnimation | REST | Unit Testing | AutoLayout | Barcodes & QRcodes |

Shopsavvy App

- App for shopping and barcode scanning
- Developer in Swift, UIKit, Auto Layout
- Worked on parts of the app including barcode scanning, QR code scanning and creation, and general bug fixing
- <https://itunes.apple.com/us/app/shop-savvy-barcode-scanner/id338828953>

- | Swift | UIKit | UI in code | CoreAnimation | REST | Unit Testing | AutoLayout | Barcodes & QRcodes | REST | Unit Testing | Swift Package Manager |

QR Code Reader and Scanner App

- Objective-C, custom UI in code
- QR code scanning and creation
- Added custom sharing and feedback functionality
- <https://itunes.apple.com/us/app/qr-code-reader-and-scanner/id388175979>
- | Objective-C | UIKit | UI in code | CoreAnimation | REST | Unit Testing | AutoLayout | Barcodes & QRcodes |

Undabot (Sep 2013 - Jun 2014) – Junior Junior iOS Developer

Tito

- iOS (iPad) book application about the life of Josip Broz Tito
- Objective-C, CocoaPods
- Implemented custom UI in code (from design sheets)
- CoreText custom page layouts
- | Objective-C | UIKit | UI in code | CoreText | CoreGraphics | CoreAnimation | Carthage | In-App Purchase |

wogibtswas.at

- wogibtswas.at, Austria's biggest "what's on offer" portal
- Objective-C, AFNetworking, REST services, CocoaPods
- Custom UI design in code
- <https://itunes.apple.com/de/app/wogibtswas.at-aktionen-angebote/id771962700>
- <https://www.wogibtswas.at>
- | Objective-C | UIKit | UI in code | AFNetworking | AutoLayout | REST | Carthage |

Blade Soho App

- Custom app for one of the leading London hair salons
- Objective-C, AFNetworking, REST services, CocoaPods
- Implemented custom UI in code (from design sheets)
- <https://www.bladesoho.co.uk>
- | Objective-C | UIKit | UI in code | AFNetworking | AutoLayout | REST | Carthage | Unit Testing |

Coachlette App

- Custom app for personal trainers and coaches
- Objective-C, AFNetworking, REST services, CocoaPods
- Implemented custom UI in code
- | Objective-C | UIKit | UI in code | AFNetworking | AutoLayout | REST | Carthage | Unit Testing |

SKILLS

- | AFNetworking | ARKit | AVFoundation | AppKit | Artificial Intelligence | AutoLayout | Barcodes & QRcodes | Carthage | CocoaPods | CoreAnimation | CoreData | CoreGraphics | CoreText | GraphQL | In-App Purchase | Objective-C | OpenAPI | Push Notifications | REST | Structured Concurrency | Swift | Swift Package Manager | SwiftUI | UI Testing | UI in code | UIKit | Unit Testing | WebSockets | macOS |