Vahid Golnegari

Senior Android Developer

Istanbul, Turkey • v.golnegari@gmail.com • https://www.linkedin.com/in/vahid-golnegari • GitHub •English(Upper-Intermediate)

I am an Android Developer with over **6 years** of experience. My journey in Android development began with **Java** and transitioned to **Kotlin**, and from **XML** to **Jetpack Compose**. I have a deep understanding of all phases of the **SDLC** and a strong foundation in **UI/UX** design and **Agile methodologies**

SKILLS

Languages: Kotlin, Java
Teamwork: Jira, Git, Agile, Swagger
Architecture: MVVM, MVI, MVP
Tools: Android SDK, Jetpack Compose, Jetpack Components, Kotlin Coroutines, RxJava, Retrofit, Socket.Io, Apollo, Dependency Injection (Dagger, Hilt), Room DataBase, DataStore, Navigation Component, FirebasePushNotification, Jenkins, SonarQube, Ktlint,
Concepts: DataStructure & Algorithms, S.O.L.I.D Principle, Design Patterns, Clean architecture, Modular Architecture
Tests: JUnit, Mockito, Robolectric, Espresso

EXPERIENCE

Podin (+500 Downloads), Baloot (+100k Downloads) | Senior Android Developer | (Fully Remote) | Jan 2022 - June 2023

Podin is an application centered around cryptocurrency that facilitates automatic savings for users, including small amounts, through features like direct debits, round-ups, and recurring payments.

- Leveraged **Jetpack Compose** to significantly accelerate UI development, fostering the creation of **modular UI components**. This approach not only enhances design reusability but also empowers seamless composition across the entire application, resulting in significant time and efficiency gains and increasing our development process speed up to %50 faster than before.
- **Orchestrated** the transition to a **modular architecture**, radically transforming our development approach. By segregating the application into **core and feature modules**, we streamlined teamwork, **accelerated feature development**, and established a robust foundation. This modular framework, encompassing critical components in the **core module** and **specific functionalities** in feature modules, propelled our project to new heights. Additionally, the build process was sped up and the build time was reduced by 15%.
- Employed the **MVI architecture** to meticulously **segregate UI**, **business logic**, and **state management**. This strategic implementation not only **enhances code maintainability** but also paves the way for effortless **scalability**, showcasing a proactive approach to future-proofing the application.
- Leveraged **Kotlin Coroutines** to ensure **smooth user interactions** and **efficient data flow** throughout the application.
- Applied **Unit Tests** to enhance **code reliability**, facilitate early **bug detection**, and streamline **refactoring processes**.
- Developed real-time capabilities using **Socket.IO** to subscribe and respond to real-time changes.
- Executed **dependency injection** in Android applications using **Hilt**, streamlining the development process and enhancing **code maintainability**. My proficiency in these technologies has enabled me to create **robust** and **scalable Android applications** while collaborating seamlessly with development teams to meet project objectives
- Managed **app crash monitoring** and resolution using **Sentry** and **Firebase Crashlytics** in various projects. Proficient in **error log analysis** and **collaborating with development teams** to enhance **app stability**. increasing **crash-free** up to %90.
- **Integrated** SonarQube to fortify **code quality** and boost security in our projects. **SonarQube** diligently monitored and pinpointed **code issues**, including **bugs**, **vulnerabilities**, and **technical debt**, allowing for proactive solutions. This tool, combined with cultivating a culture of code excellence and collaboration among team members, led to our **software solutions** becoming **more reliable**, **secure**, and **user-friendly**

Baloot, with over 100,000 downloads, is a stock-centric micro-saving app. It assists users in effortlessly saving, even small amounts, through automated processes like scheduled withdrawals, spare-change accumulation, and periodic contributions, all while focusing on stock investments.

- Orchestrated seamless collaboration among designers, product managers, and developers, ensuring timely delivery of high-quality features; accelerated time-to-market by 20% and increased customer satisfaction by 15%
- Actively **participated** in **code reviews**, offering **constructive feedback** to **optimize code quality** while adhering to best practices.
- Managed the **Kotlin code base** according to **Android best practices**, ensuring rigorous **testing**, **bug fixing**, and continuous enhancements to boost **performance**, **architecture**, **design**, and overall **code quality**, resulting in a remarkable **97% crash-free** rate for the application.
- Skillfully adapted **daily plans** to align with **Sprint goals**, effectively executing **2-week Sprints**, engaging in discussions about previous sprints, and proactively fostering **teamwork improvement** during retrospectives.
- Successfully **Generated unit tests** to enhance the **testability** of all **application layers** through a seamless **migration from MVVM to MVI** architecture.
- Designed a **flexible querying system**: Leveraged **GraphQL**'s flexible querying capabilities to empower users with the ability to request data from multiple sources in a single query, **reducing the number of round-trips** to the server.

- Combined **Dagger2** for streamlined dependency injection, enhancing **code modularity** and **testability**. By automating dependency injection, we eliminated **boilerplate code**, ensuring a more maintainable and scalable codebase. Dagger2 **optimized** our development process, improving our application's architecture and overall efficiency.
- introduced **Ktlint** to enforce consistent **code formatting** and style in our Kotlin project, enhancing **code quality** and readability. This initiative catalyzed our **development workflow**, reduced review cycles, and improved team collaboration, ultimately leading to a higher-quality product.

GAP MESSENGER (+1M DOWNLOADS) | Android developer | REMOTE | Dec 2018 - Feb 2022

Gap Messenger is a cross-platform totally encrypted cloud-based messenger.

- **Conducted** comprehensive **research** and **code review** of **Telegram source codes** to identify the most effective approach for implementing the floating avatar in group chat.
- **Investigated** Telegram source codes to identify the **optimal solution** for creating a user profile similar to Telegram, enhancing the **visual appeal** of the user's profile page.
- **Analyzed Telegram source codes** to determine the **best approach** for implementing the SwipToReply feature, enabling users to respond to messages promptly.
- **Designed and Created an Audio and voice player** using the **Clean MVVM architecture**, ensuring improved **code organization**, **layer separation**, f**aster development**, and enhanced **testing capabilities**.
- **Optimized** message history (server requests) to **minimize redundant requests** and prevent the loading of duplicate messages.

EDUCATION

UNIVERSITY OF BIRJAND **Birjan, Iran**
Bachelor of Software Engineering **Sep 2009 - Mar 2015**