

Eugene Shuvagin, Android Developer

Number: +375297387026

Telegram: <https://t.me/i30mb1>

Email: [fate.i30mb1@gmail.com](mailto:fate.i30mb1@gmail.com)

[GitHub](#), [Google Play](#), [LinkedIn](#)

Hello! My name is Eugene. Live in Belarus. Graduated from as an engineer. After than i decided to become an Android Developer and now I am working at [T-bank](#) company.

## Work History

### VironIT / Android Developer

*September 2019 – Mart 2022*

- Developed and maintained the Turkish messenger “[BIP](#)” app
- Mentor new team members, co-op students, interns, and peers alike

### Kufar / Android Developer

*Mart 2022 – May 2024*

- Developed and maintained the marketplace “[Kufar](#)” app

### Tinkoff / Android Developer

*May 2024 – Present*

- Developed a Selfie verification component for seamless integration across Tinkoff’s ecosystem, including the main [banking app](#), [SME app](#), [investment platform](#), and POS terminals.

## Skills

### Common Skills

- Java, Kotlin, Git
- Android Architecture Components
- Custom View, Compose
- Data Binding
- RxJava2, Kotlin Coroutines, Kotlin Flow
- Retrofit2
- Dagger2
- JUnit4, Junit5, Mockito, Robolectric
- Gradle
- Design Patterns

### Languages

- English (Upper-Intermediate)
- Russian (Native)
- Chinese (Intermediate)

### Accomplishments

- Successfully completed an internship for 50 students in the company
- Helped over 10000 people answering [Stackoverflow](#) questions
- Writing articles to share knowledge with community on the [Medium](#)/[Habr](#)

## Work Experience

### Projects in T-Bank:

- Implemented a self-contained WebSocket solution (zero-dependency) to bridge T-bank's mobile and web ecosystems, increasing identified user flows by 100% in production
- Built FaceShot SDK across Tinkoff's ecosystem with modular camera/core components (Photo/Video), dynamic ML model updates, and custom hint analytics, achieving 100% production rollout; optimized pipelines (Kotlin DSL, Maven publishing) and benchmarked ML engines (MLKit/TFLite/ONNX) for inference speed/accuracy
- Adapted FaceShot SDK for POS terminals – optimized camera, ML models, and UI for low-power devices, enabling biometric authentication on banking kiosks/ATMs
- Integrating CryptoPro for FSB-compliant biometrics: AES-256 channels + EDS for photo/video samples

**Project name:**

[BiP](#) - Turkish Messenger

**What I have done:**

- Implemented feature of adding chat messages to favorites, including development of a new screen for displaying favorite messages, creating a new table in the database to store the favorite messages, and database migration for integration. Also, added a button in the UI to allow users to mark a message as a favorite
- Implemented an "Archived Chat" feature that allows users to keep a few desired chats hidden in an "Archived Chat Folder". The feature allows users to easily archive and hide desired chats for better organization and privacy, and also allows them to view the archived chats and unarchive them at any time. The feature also included an SQLite database to store the archived chats, allowing for quick and efficient retrieval of hidden conversations
- Implemented a feature that allows users to send video or photo using the original quality instead of a compressed version. This feature uses the device's original resolution and quality settings to send media files, ensuring that the recipients receive the highest quality media possible. This feature also allows users to send large files without having to worry about the file size limitations of traditional compressed files
- Implemented a feature that sends notifications to all members of a group chat when someone exits, changes the group avatar or makes any other changes to the group. This feature keeps users informed and engaged by keeping them up-to-date on the latest changes and activity within their group chats
- Implemented a feature for that displays the file size of media files such as images and videos before they are downloaded, allowing users to make informed decisions about whether to download or not
- Implemented a feature for a messaging app that allows users to copy selected dialog text with the names of the sender and recipient in a short format. This feature allows users to easily copy and share specific conversations or parts of conversations with others
- Implemented a feature that allows users to watch YouTube videos in a separate draggable window and being able to chat and interact with other users. This feature improved the user experience by allowing users to multitask within the app and enjoy a more seamless experience
- Implemented a feature for a messaging app that allows users to change the wallpaper for chats using a standard collection or by loading their own original image. The feature also includes the option to crop and adjust the image to fit the chat background perfectly
- Implemented a feature that integrates Zoom into camera. Allows users to zoom in and out while capturing photos or videos using their device's camera. The feature is integrated with Camera1/Camera2 API which are the standard Android APIs for camera operations

**Additional information:**

- Migrate Java to Kotlin and reduce the use of third-party libraries
- Reduced application crash from 92 to 97% while working on ANR
- Shortened application launch time in 20%

**Project name:**

[Kufar](#) – classified ads for the Belarusian market

**What I have done:**

- Implemented a set of paid promotion services with a complex user flow that allows for effective sales. This feature allows users to purchase and promote their products or services through the app, using a user-friendly interface and a seamless flow. This feature includes targeted advertising, analytics, and reporting tools to help users optimize their campaigns and measure their performance
- Implemented a native widget for payment transactions within the app, replacing the previous implementation of a web-based widget that was accessed through a WebView. Improved the user experience by providing a secure and user-friendly interface for making payments directly within the app, and streamlined the payment process by eliminating the need to navigate to a separate web page
- Implemented a basket feature for e-commerce advertisement that allows users to save their selected products for later purchase or review, and also allows users to purchase multiple products at once. The basket feature also includes an option for users to apply a discount code or voucher for the products, making it easy for users to save money
- Integrated an online booking system for advertisement, along with the ability to view landlord statistics and the option to report any issues. This feature allows users to easily schedule and manage their advertisements, while also providing them with information about their landlords. Additionally, the feature enables users to report any issues they may have with their landlords, increasing transparency and accountability
- Implemented a VIN auto report generation feature that allows users to generate a comprehensive report on a vehicle's history using its VIN number. The report includes information such as accident history, service records, and previous ownership. The feature also allows users to view the report as a PDF or have it sent to their email for easy access and sharing
- Working on the ad management feature (Yandex and Google Ads), I focused on improving the user experience by making the ads more relevant and less intrusive. To accomplish this, I implemented various targeting options such as geographic, demographic and interest-based targeting. This helped to increase the click-through rates and revenue of the apps
- Developed a parser that converts HTML-like text to Spannable text, allowing for the implementation of various text formatting options such as color, bold, and clickability
- Migrated code from RxJava2 to Coroutines Flow to improve the app's performance and stability. The migration improved the ability to handle errors and cancellation of tasks, making the app more robust

**Additional information:**

- A/B/C experiment across all the app
- Reduced application crash from 95 to 98% while working on ANR
- Speed up the project build time for 30% by modifications to build scripts, tuning Dagger2
- Migrated from Groovy DSL scripts to Kotlin DSL and integrated a Version Catalog
- Increase test coverage from 28 to 34% by adding Test for old/new product functionality
- Migrate app to dark theme