

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 [BSUIR](#) as an engineer. After that I decided to become an Android Developer and started my career with [VironIT](#) company. Now I am working at [Kufar](#) 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 – Present*

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

## Skills

### Common Skills

- Java, Kotlin, Git
- Android Architecture Components
- Custom View, Compose
- Data Binding
- RxJava2, Kotlin Coroutines, Kotlin Flow
- Retrofit2
- Dagger2
- Dynamic Feature Modules
- JUnit4, Mockito, Truth
- Gradle
- Design Patterns

### Languages

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

### Accomplishments

- Successfully completed an internship for 10 students in the company
- learned to use Adobe Illustrator In order not to waste time waiting for vector icons in the desired resolution from the designer
- Helped over 100 people answering [Stackoverflow](#) questions
- Writing articles to share knowledge with community on the [Medium](#)

## Work Experience

### 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

## **Project name:**

[About Dota](#) - home pet project for Dota 2 fans where I test new technologies before implementing them into a working project. Thanks to this, I see the pros and cons of technologies, I can test the limits of its applicability and do not turn a working project into a zoo of different technologies

## **What I have done:**

- Designed and implemented a feature that parses and stores bio, abilities, tips, talents, last changes, sounds and replicas for 120 heroes in JSON files on the user device. This feature allows Dota2 players to access detailed information about the heroes without having to rely on external sources. The data was parsed from different Dota2 websites and organized in a custom structure for easy access and use within the app. This feature improved the player experience by providing them with a wealth of information about the heroes, and also helped to enhance their overall in-game experience
  - Implemented feature for customizing app notifications with user-downloaded sounds and replicas, improving user engagement and personalization
  - Developed feature that parses and displays latest news from various sources in a WebView within the app. Users can stay updated on current events and easily access news articles by browsing through the in-app WebView, eliminating the need to switch between multiple news sources
  - Implemented feature for live streaming Dota 2 matches from Twitch within the app, including real-time chat functionality and Picture-in-Picture mode. Users can watch their favorite Dota 2 streams and interact with the community in real-time, all while using other features of the app
  - Designed and implemented a feature that parses popular builds for heroes from a public external website on a daily basis. The feature displays the most up-to-date builds for players to use, improving their in-game performance and strategy. The feature ensured that the builds displayed are constantly updated and relevant
  - Implemented a feature that parses and displays tournament games and their outcomes from a public external website. Users can stay informed on the latest competitive scene and view the results of tournaments, keeping them up to date on the current meta and trends
  - Developed feature for in-app purchases that allows users to unlock additional features within the app. This feature allows users to access more content, functionality, and services by purchasing them within the app, providing a more personalized experience and generating revenue for the app
  - Implemented feature that allows users to customize the theme of the app with dynamic options and a changing launch icon. This feature provides a more personalized and visually appealing experience for users, while allowing them to easily identify the app on their device
  - Implemented a feature for remote configuration that allows for the toggling of certain features within the app. This allows for easy and efficient adjustments to the app's functionality, such as enabling or disabling certain features, without requiring an update to be released
  - Developed feature that integrates Google and Yandex ads SDK to show different types of advertisements within the app. This feature allows for more flexible and efficient monetization of the app, as well as providing users with relevant and targeted ads. The feature provides the ability to display various ad formats such as banner, interstitial and video ads. This also allows for more precise targeting and better performance of the ads