

App for Finding a Comedy Improv Team

Bulat Minnakhmetov
Faculty of Computer Science
Higher School of Economics
Moscow, Russia
brminnakhmetov@edu.hse.ru

Abstract

Improv is becoming more popular in Russia. There are more performances, new schools, and tournaments.

Improv requires teamwork. However, it is hard to find a team that will exist for an extended period of time. The current methods for finding a team are going to courses, participating in jams, or asking friends. These approaches are time-consuming, costly, and often fruitless. As a result, many people lose motivation and stop doing improv. This is slowing down the development of the genre.

This project dedicated to developing a service that will help improvisers find a team. Users will be able to search for teammates based on gender, age, location, performance videos, preferred improv styles, experience and practice schedules. It will make it easier to find a like-minded person and avoid unproductive connections. The result of the project will be an mobile app for Android and IOS.

Keywords

Mobile application development, UX design

1 Introduction

In improv the ability to react quickly to what is happening and interact with partners plays a key role. There is no script in improvisation and therefore teamwork is very important.

Comedy improvisation in Russia is attracting more and more people. Schools, courses, tournaments, and performances are opening up. The most successful teams create their own shows.

In order for improv team to exist for a long time, members must share similar values and have same capabilities. Some people want to do it professionally, others are just looking for a hobby. Some people are willing to practice every day, others are not ready to spend more than two hours a week. There are also different tastes regarding what improvisational form to play. There are a lot of them and new ones are being invented.

At present, people attend courses and jams¹ to find a teammate. These events are not aimed at helping people find a team. They require money and time but offer a small chance of meeting a like-minded person. Without a team, a person has to spend money to continue doing improv because jams and courses are the only events where you can play improv without a team. Not everyone has that opportunity.

This work is dedicated to developing a service for helping people find an improv team. The result will be an app for Android and IOS, where an improviser can fill their profile and search for other improvisers using filters based on profile features.

¹An improv jam is an event where improvisers gather to practice improv. There session can be attended by any improviser regardless of their experience.

At the moment, there are no alternative solutions. There are actor databases, but they were designed for casting directors to find actor for a specific role. The search criteria there are focused more on appearance and specific skills, such as playing the piano.

Let's have a brief overview of the article. First, we will discuss the target audience. Second, we will look at existing methods for finding a team. Third, we will go through UI sketches. Finally, we will talk about the tech stack and the backend architecture.

2 Target audience

The target audience of the proposed service are people attending courses and jams at improv schools. Now there are no official statistics on how many people practice improv worldwide. When it comes to Moscow, approximately 2,000 people are doing improv, based on the number of participants in improv school chats.

3 Existing solutions

The current approaches to find an improv team:

(1) Courses

Pros:

- You can play with people and decide for yourself whether you want to play with them in a long run.
- You can meet new people who didn't play improv before.

Cons:

- Expensive. In Moscow 8-lessons course (3 hours per lesson) costs over 30k rubles on average. If your only goal is to find a team and you don't care about learning, this method is inefficient.
- Many people take a improv course to try something new. Improv schools position themselves as a place where you can leave your comfort zone and meet new people. So in the course there might be no people who is ready to commit to something long term as a team.

(2) Jams

Pros:

- Jams also give you the opportunity to play with potential teammates.
- You only pay for one session at a time.
- A jam doesn't have a fixed set of people like a course. So every jam you can meet someone new.

Cons:

- But if you come to a jam more than once, you will notice that it's more or less the same people. One improv school has even made a subscription to attend jams. The subscription is increasing chance that you will meet the same person at next jam session.

- Still expensive. A ticket to a jam in Moscow costs around 1,000 rubles on average. ImproCulture, 2024, MIC, 2024
- (3) Improv-related chats.
- Pros:**
- Wide selection.
- Cons:**
- No way to see how the person plays beforehand.

The first two methods require money and time and offer only a small chance of finding a team. The third method is cheaper and only requires finding a space for a trial practice, but it still time-consuming since there is no way to see how a person plays beforehand. It means multiple attempts will be needed to find the right people.

The high cost of jams and courses becomes an even bigger issue when it comes to young improvisers. They often don't have a lot of money. They might find money to take a course but after that they have to pay again. Because of this, improv schools are mostly attended by middle-aged and older people. Often they just want to relax after work and not pursue improv professionally.

A platform where improvisers could find teammates based on criteria such as gender, age, location, performance videos, preferred improv styles, experience, availability, and practice schedule would solve the problem. When a team is formed, the cost of practice sessions is low since they can practice at someone's home or rent a space such as a dance studio. It's two to three times cheaper than a ticket to jam.

There are no analogs to the proposed solution. There are actor databases, such as backstage.com, Akteram.com or Kino-Teatr.ru, but the search for an actor for a film project and a partner for improvisation are different. The differences are shown in Table 1.

Feature	Actor for Film	Partner for Improv
Role	Follows script	Creates scenes on the spot
Interaction	Structured plan	Supports partner, flexible
Process	Prepared, stable	Unpredictable, mistakes are part

Table 1: Film Actor vs. Improv Partner Comparison

Thus for a film actor accuracy, stability and ability to internalize someone else's words are important, whereas an improviser has to be flexible, forgiving mistakes and able to figure out what to do on the spot.

4 Product design

All business logic is concentrated in two screens shown in Figure 1. The UI designed to satisfy following criteria: it should be clean, simple, distract free and focusing only on import parts, which is finding the right person to play with. It will updated once feedback from first users are collected.

4.1 Profile Screen

In the profile, you will be able to see photo, performance videos, and details such as:

- Name
- Gender



Figure 1: Profile and Search screens

- Age
- Performance videos
- Bio
- Improvisation format
- Training availability
- Goals in improvisation
- Courses taken

You can click the "Contact" button to start a chat with the person. The chat screen is under development.

Videos and bio take precedence over other fields in the profile because it is likely that people already used filters to put constraints on them, whereas videos and bio are new information that you didn't interact with on search page.

4.2 Search Screen

In the search screen, you can search based on the parameters mentioned above. To go to the profile screen, you need to click on the search item on the page. The UI design is using existing solutions (e.g filter chips under a search bar or navbar at the bottom of the screen that makes navigation more intuitive) that you can find in apps like Ozon, Wildberries, Avito and many more. It will make easier for new users to start using the service.

5 Tech Stack

5.1 Client

Kotlin Multiplatform (KMP) JetBrains, 2024 was selected as the technology stack for the cross-platform mobile application because of following reasons:

- (1) **Use Existing Expertise:** The author has experience with **Kotlin** for Android development. It will make development process faster and does not require to learn a new language.

- (2) **Code Sharing:** KMP allows share a code base between Android and iOS.
- (3) **Flexibility:** It enables the use of **native UI frameworks** for Android and iOS when it's necessary.

5.2 Backend

The following technology stack was chosen the backend:

- Backend: **Golang** (Gin)
- Database: **PostgreSQL**
- File Storage: **AWS S3**
- API: **REST API**
- Authentication: **JWT**
- Password Hashing: **bcrypt**

This technology stack was chosen based on performance, scalability, and security and author's existing experitices in **Golang** T. G. Authors, 2024. The **Gin** G. W. F. Authors, 2024 framework was selected for the backend because of its high performance, concurrency support, and low overhead.

PostgreSQL Stonebraker and Rowe, 1986 was chosen for storing user profile data. It's a reliable, scalable and time-proven.

AWS S3 Services, 2024 is used for storing photos and videos because of its scalability, availability, and cost-effectiveness.

A **REST API** Fielding, 2000 was chosen for communication between the mobile client and the backend. It's simple and has extensive support.

When it comes to security, **JWT** Jones et al., 2015 was chosen for authorization and **bcrypt** Provos and Mazières, 1999 for hashing passwords.

6 Architecture

The backend is structured using a **microservices architecture** Francesco et al., 2017 as shown in Figure 2 to make each service responsible for a specific task, which increases fault tolerance of the entire system and make it possible to scale each service independently.

The following is description for each microservice:

6.1 User Service

Responsibilities:

- Registration: Handle new user registration.
- Authentication: Validate credentials and return JWT token.

Endpoints:

- POST /register: Registers a new user.
- POST /login: Authenticates a user and returns a JWT token.

6.2 Profile Service

Responsibilities:

- Get: Allow users to get their and other people profiles.
- Update: Allow users to update their bio, location, and other textual information.
- Media: Handle uploading and linking photos and videos to profile.

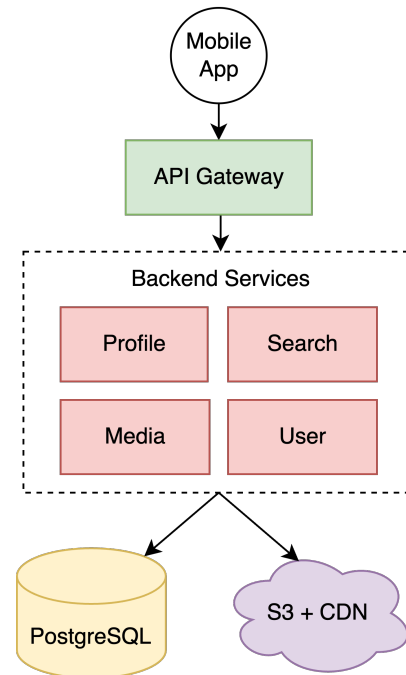


Figure 2: Backend architecture

Endpoints:

- GET /profile: Get the current user's profile.
- PUT /profile: Updates the current user's profile.

6.3 Media Service

Responsibilities:

- Upload files to S3. After upload, the service generates a URL that points to the media file. The CDN caches media, which improves speed and reduce the load on S3.

Endpoints:

- POST /media/upload: Uploads media to S3 and returns the CDN URL.
- GET /media/{media_id}: Returns the CDN URL for existing media.

6.4 Search Service

Responsibilities:

- Search to search profiles based on text fields from profile.

Endpoints:

- GET /search: Performs a search query to find profiles that satisfy certain criteria.

7 Anticipated results

The backend has been deployed, and the apps for both platforms have been published on Google Play and the App Store, respectively, and are available for download.

8 Conclusion

Improv is getting more popular in Russian but it's still difficult to find an improv team. There are approaches as taking a course, going to jams or ask people in improv-related chats but there are ineffective. They require money, time and often don't bring results. Many beginner improvisers lose their motivation and stop doing improv because it's hard to find the right people to create a long term project as a team. If a person does not have a team, in order to practice improv they have to spend a lot of money. The problem is especially important for young improvisers. They don't stable income yet. The described in this paper service attempts to make it easier for improvisers to find teammates. The app will allow users to search for improv partners based on criteria important for making long lasting partnership: goals in improv, performance videos, practice schedules and preferred improv styles.

The app will be available on both mobile platforms Android and IOS. It will allow many people to use the service. It will help them to form teams that lasts and keep doing improv. This project can help the improv community grow and make the art form more accessible to everyone. With more team, number of improv performances will increase. In the long run it will foster the overall development of improv as art form.

References

- ImproCulture. (2024). Improculture jam tickets [Accessed: 2024-02-28]. <https://improculture.ru/jam#jamtickets>
- MIC. (2024). Improv course 7 [Accessed: 2024-02-28]. <https://moscowimprovclub.com/course7>
- JetBrains. (2024). Kotlin multiplatform documentation [Accessed: 2024-02-28]. <https://kotlinlang.org/docs/multiplatform.html>
- Authors, T. G. (2024). Go documentation [Accessed: 2024-02-28]. <https://go.dev/doc/>
- Authors, G. W. F. (2024). Gin documentation [Accessed: 2024-02-28]. <https://gin-gonic.com/docs/>
- Stonebraker, M., & Rowe, L. A. (1986). The design of postgres. *ACM SIGMOD Record*, 15(2), 340–355. <https://doi.org/10.1145/16894.16888>
- Services, A. W. (2024). Amazon simple storage service (s3) [Accessed: 2024-02-28]. <https://aws.amazon.com/s3/>
- Fielding, R. T. (2000, May). *Architectural styles and the design of network-based software architectures* [Doctoral dissertation, University of California, Irvine]. https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf
- Jones, M. B., Bradley, J., & Sakimura, N. (2015, May). Json web token (jwt) [Accessed: 2024-02-28]. <https://doi.org/10.17487/RFC7519>
- Provos, N., & Mazières, D. (1999). A future-adaptable password scheme [Accessed: 2024-02-28]. *Proceedings of the 1999 USENIX Annual Technical Conference*. <https://www.openbsd.org/papers/bcrypt-paper.pdf>
- Francesco, P. D., Malavolta, I., & Lago, P. (2017). Research on architecting microservices: Trends, focus, and potential for industrial adoption. *2017 IEEE International Conference*

on Software Architecture (ICSA), 21–30. <https://doi.org/10.1109/ICSA.2017.24>

Word Count 1794