

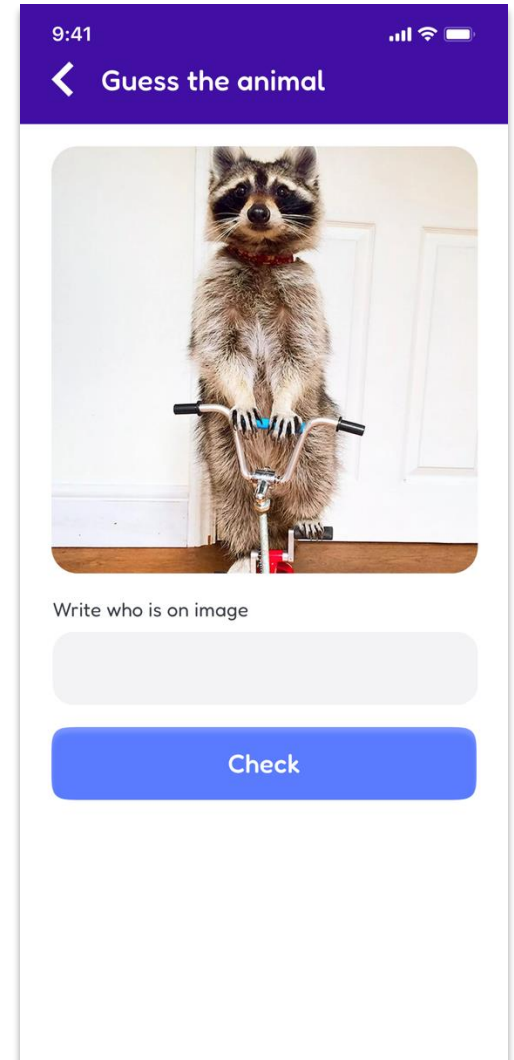
T-Language

Мобильное приложение для изучения языков



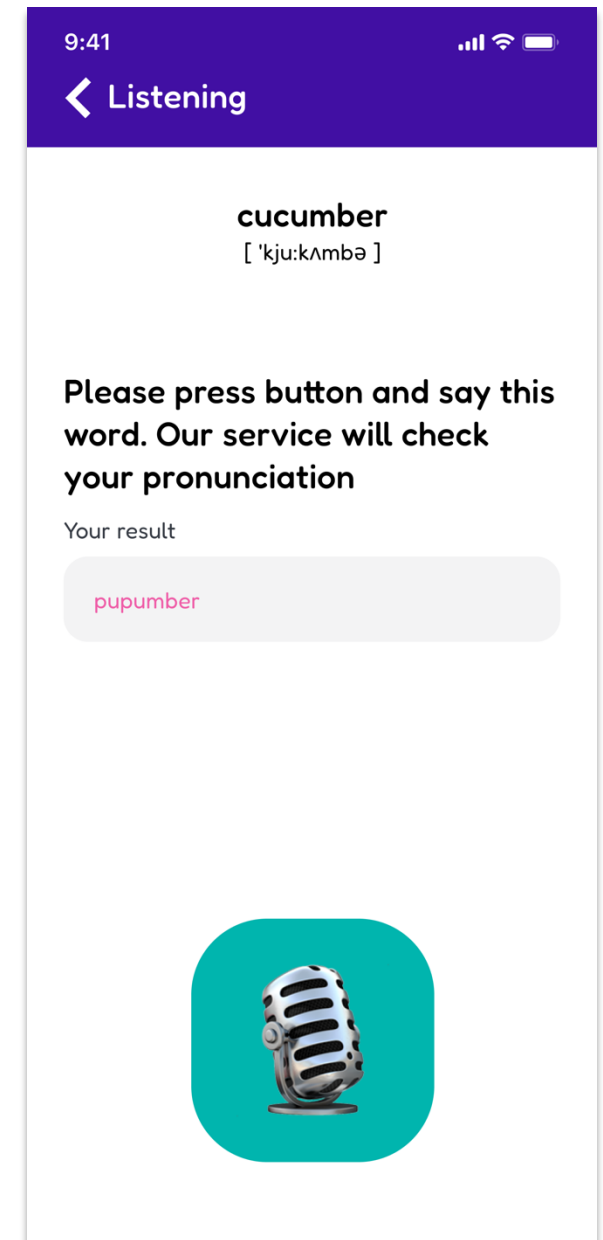
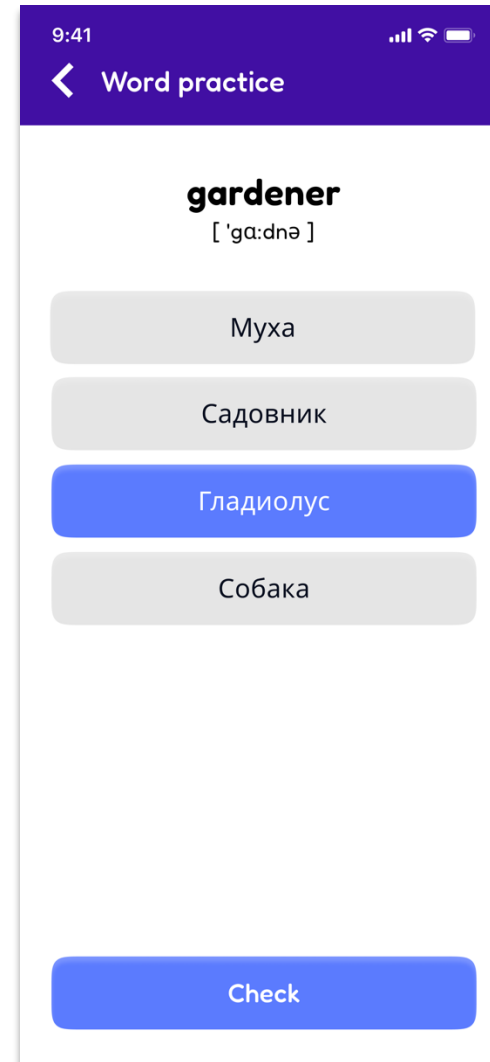
T-Language App

Темирзянов Амир, Республика Татарстан



T-Language: что это?

- Интерактивное приложение для изучения языков
- Геймификация процесса обучения
- Мотивация пользователей через достижения и соревнования




Технический стек

- Язык программирования: Kotlin
- Архитектура: Single Activity + Fragments
- Dependency Injection: Koin
- Платформы: Android 13+
- Бэкенд: Supabase REST API



9:41 📶 🔋

< Login


 Learn at home

For free, join now and start learning

Email Address

Email

Password

..... 

Forgot Password

Login

Not you member? [Signup](#)

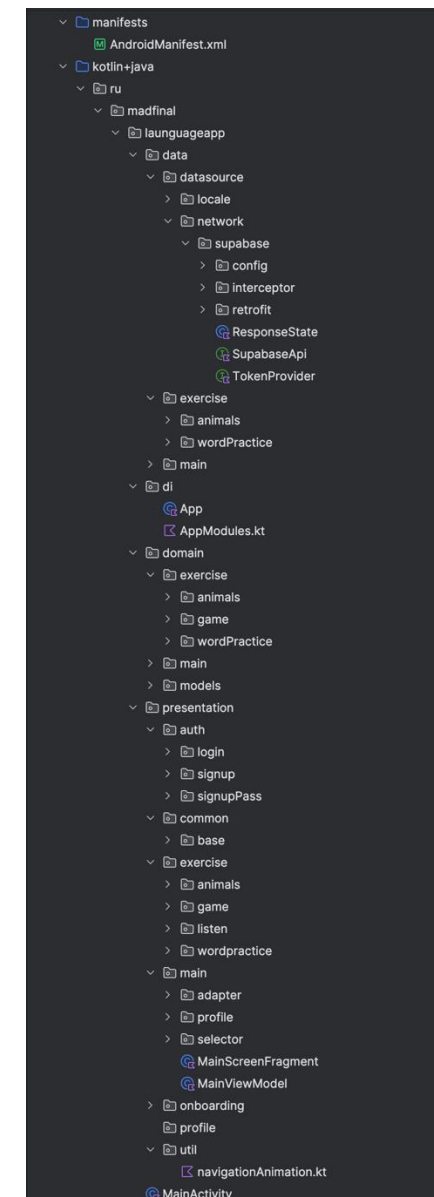
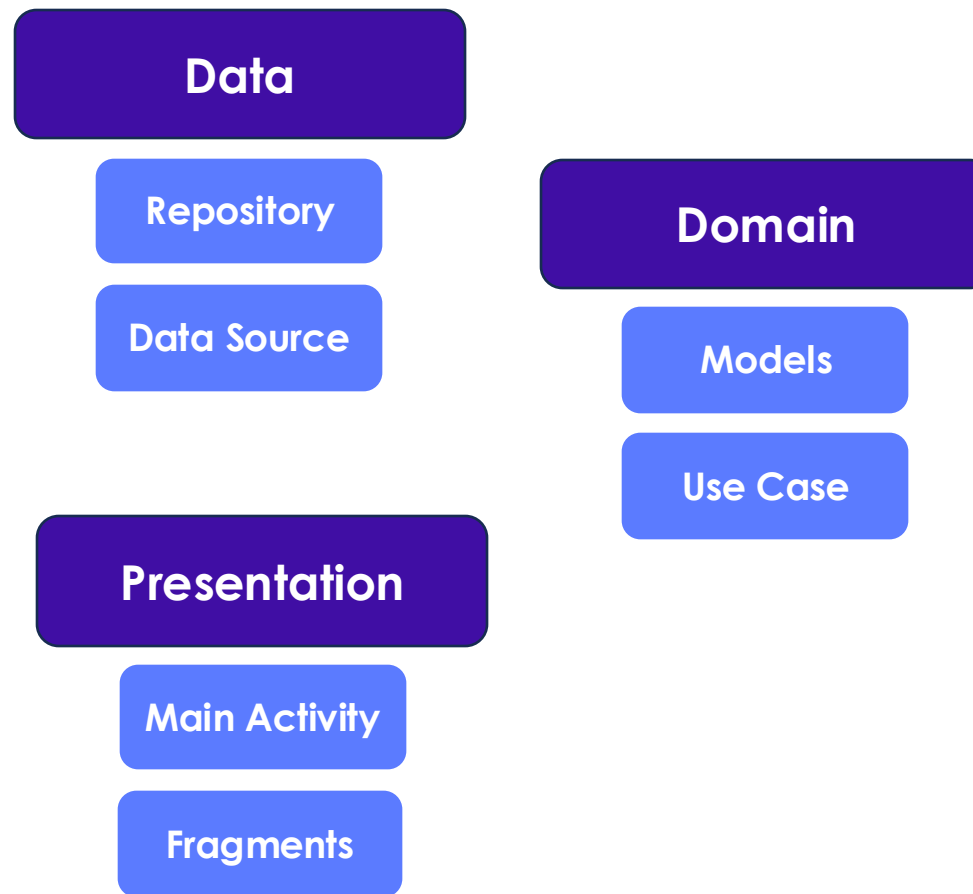
Архитектура приложения

Clean Architecture

Разделение на слои:

- Domain
- Presentation
- Data

MVVM



Supabase: бэкенд

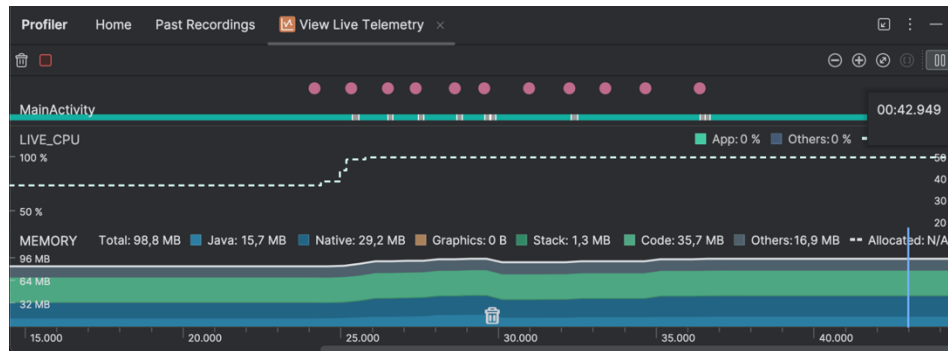
- Авторизация пользователей
- Хранение прогресса обучения
- Управление профилями пользователей
- Интеграция WebSocket для мультиплеера

The screenshot displays the Supabase Studio interface with a dark theme, showing a database schema with 18 tables. Each table is represented by a card with its name, a list of columns, and their data types. Primary keys are indicated by a diamond icon, and foreign keys by a circle with a line. The tables are arranged in a grid-like fashion.

Table Name	Columns
Cart	id (PK), created_at, product, user_id, quantity
Order	id (PK), created_at, product, user_id
Favorite	id (PK), created_at, product, user_id
Product	id (PK), name, image, price, description, gender, category, popular, product_id
exchange	id (PK), created_at, price, name_short, name_full
di_questions	id (PK), created_at, title
fc_images	id (PK), created_at, image
Orders	id (PK), created_at
chats	id (PK), created_at, user_id, name
Notification	id (PK), created_at, title, description
messages	id (PK), created_at, chat_id, text, sender_id, timestamp, is_ai
queue	id (PK), created_at, profile_id, game_id
Promotion	id (PK), title, discount, image
ProductInfo	id (PK), name, image, price, description, gender, category, popular, isFavorite, inCart
Profile	id (PK), created_at, name, surname, address, phone, user_id
profiles	id (PK), created_at, avatar, nickname, points
ra_questions	id (PK), created_at, title, ra_answers
word_practice	id (PK), created_at, english, russian, transcription_english, transcription_russian
exchange_rate	id (PK), created_at, image, country, buy, sell
interest_rate	id (PK), created_at, kind, deposit, rate
language	id (PK), created_at, language
animals	id (PK), created_at, image, name
users	id (PK), created_at, nickname, age, email, password, score, name, lastName, image

Производительность

- Кэширование медиаресурсов
- Оптимизация загрузки данных
- Минимизация сетевых запросов
- Эффективное управление состоянием



T-Language



Конкурент 1



Конкурент 2



Используемые библиотеки

- **Koin** для внедрения зависимостей
- **Retrofit + RxJava** для работы с сетью
- **Room** для локальной базы данных
- **Coroutines** для асинхронного программирования
- **Tensor Flow** для анализа картинок
- **Google Ar** для отображения 3D ЖИВОТНЫХ

```
object SupabaseNetworkModule {
    fun provideOkHttpClient(): OkHttpClient {
        return OkHttpClient.Builder()
            .addInterceptor(HeadersInterceptor())
            .addInterceptor(HttpLoggingInterceptor().apply {
                level = HttpLoggingInterceptor.Level.BODY
            })
            .build()
    }

    fun provideRetrofit(baseUrl: String, client: OkHttpClient): Retrofit {
        return Retrofit.Builder()
            .baseUrl(baseUrl)
            .client(client)
            .addConverterFactory(GsonConverterFactory.create())
            .addCallAdapterFactory(RxJava3CallAdapterFactory.create())
            .build()
    }

    fun provideSupabaseApi(retrofit: Retrofit): SupabaseApi {
        return retrofit.create(SupabaseApi::class.java)
    }
}
```

```
class App : Application() {
    override fun onCreate() {
        super.onCreate()
        startKoin {
            androidContext(this@App)
            modules(appModules)
        }
    }
}
```

```
val repositoryModule = module {
    single<AnimalRepository> { AnimalRepositoryImpl(get(), get()) }
    single<WordPracticeRepository> { WordPracticeRepositoryImpl(get(), get()) }
    single<MainRepository> { MainRepositoryImpl(get(), get()) }
}

val useCaseModule = module {
    factory { GetRandomWordUseCase(get()) }
    factory { GetRandomAnimalUseCase(get()) }
}

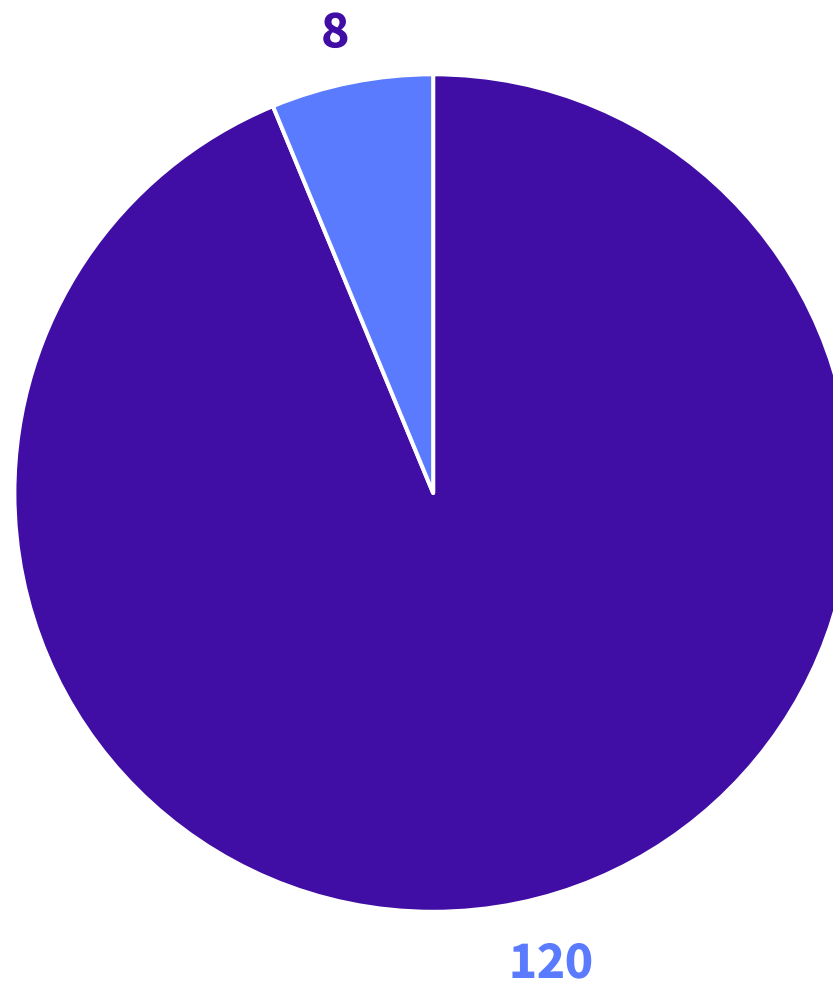
val viewModelModule = module {
    viewModel { MainViewModel(get()) }
    viewModel { WordPracticeViewModel(get()) }
    viewModel { AnimalsViewModel(get(), get()) }
}

val helperModule = module {
    single { TensorFlowHelper(androidContext()) }
}

val appModules = listOf(
    networkModule,
    repositoryModule,
    useCaseModule,
    viewModelModule,
    helperModule
)
```

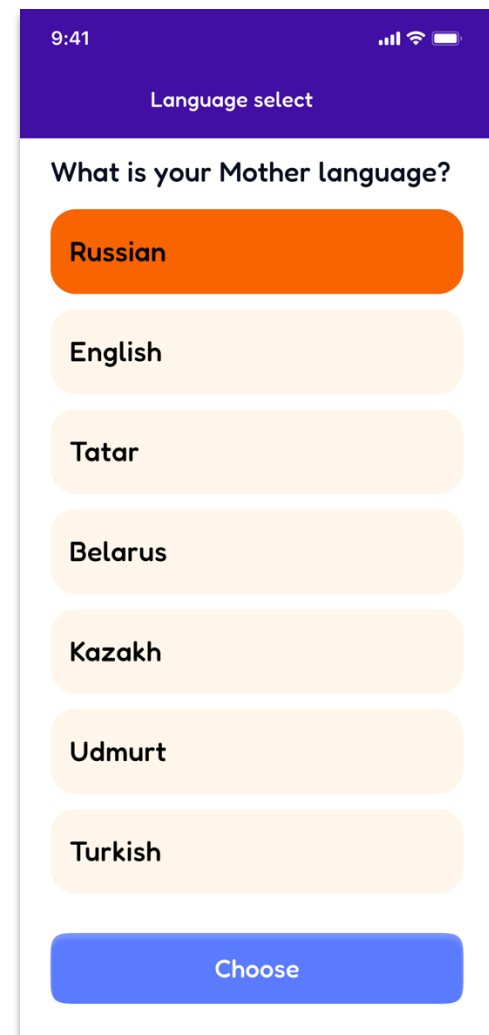
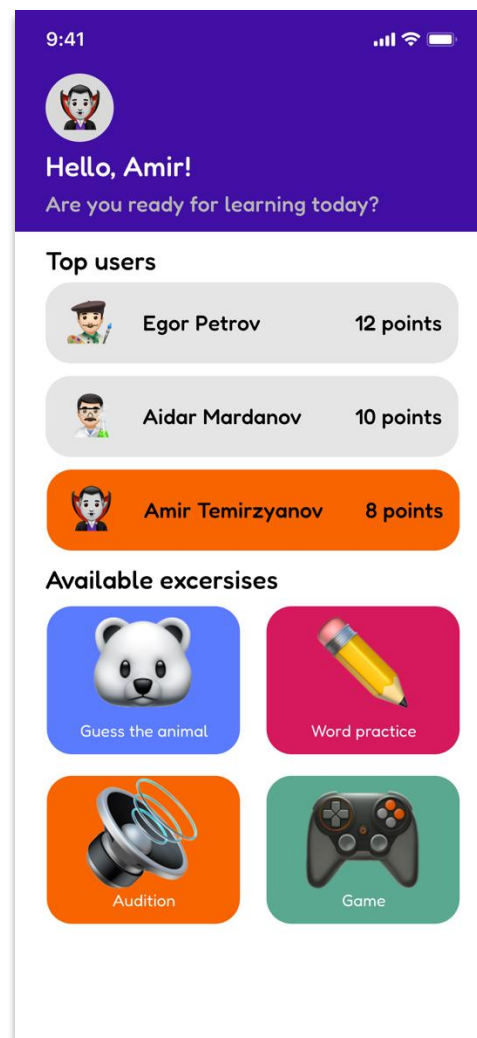
Статистика Tensor Flow

- Верно определено
- Не верно определено



Перспективы развития

- Расширение ML-моделей
- Новые интерактивные упражнения
- Социальные функции обучения
- Поддержка большего количества языков



Thank you for your attention!

Спасибо за внимание!

Игътибарыгыз өчен рәхмәт!

