

Санкт-Петербургский государственный университет

Кафедра системного программирования  
Группа 22.Б11-мм

---

**Разработка мобильного приложения для развития  
музыкальных навыков**

*Зайцев Дмитрий Сергеевич*

ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ

Научный руководитель:  
старший преподаватель СПбГУ Сартасов С. Ю.

Санкт-Петербург  
2023

# Оглавление

<b>Введение</b>	<b>4</b>
<b>1 Задачи</b>	<b>5</b>
<b>2 Обзор</b>	<b>6</b>
2.1 Анализ существующих решений . . . . .	6
2.1.1 Perfect ear . . . . .	6
2.1.2 Functional Ear Trainer . . . . .	7
2.1.3 My Ear Trainer . . . . .	7
2.1.4 Результаты анализа . . . . .	8
2.2 Требования к приложению . . . . .	9
2.3 Используемые технологии . . . . .	9
<b>3 Реализация</b>	<b>10</b>
3.1 Архитектура . . . . .	10
3.2 Интерфейс приложения . . . . .	11
<b>4 Апробация</b>	<b>13</b>
<b>5 Заключение</b>	<b>14</b>

# Введение

Музыка всегда была одним из самых популярных видов искусств. Способность выражать мысли и чувства с помощью звука никогда не переставала интересовать человечество. Множество людей с самыми разными целями и подходами к обучению выбирали и продолжают выбирать музыку как объект для своего изучения. В наше время интерес к ней всё так же силен [1].

По результатам проведённых исследований [2], можно сделать вывод о том, что обучение музыке с ранних лет жизни способствует развитию у ребёнка: памяти, моторики, внимания, навыков коммуникации, способности различать сложные эмоции и контролировать их, навыков владения языком и речью. Безусловно, это говорит о полезности прививания любви к данному виду творчества.

Изучение музыки - очень трудоёмкий процесс, требующий от человека не только желания погрузиться в новую область знаний, но и регулярных занятий, постоянной работы над ошибками, наличия рядом опытного преподавателя. Одним из важнейших элементов обучения является развитие *музыкального слуха*, т.е. способности анализировать музыку без прямого чтения нот. Этот навык необходим как ученикам музыкальных школ для успешной сдачи экзаменов, так и людям, изучающим музыку самостоятельно.

Однако, несмотря на стремительное развитие технологий, на данный момент существует не так много эффективных решений для развития этих навыков, объединяющих в себе удобство и практичность.

Данная работа посвящена разработке мобильного приложения, которое сделает процесс изучения музыки увлекательным, познавательным и вдохновляющим для всех, кто хочет погрузиться в мир звуков и мелодий.

# Задачи

**Целью** работы является создание мобильного приложения, которое смогло бы облегчить процесс формирования способности определения интервалов на слух.

В ходе работы были поставлены следующие **задачи**:

## **Осенний семестр**

1. обзор существующих решений,
2. разработка архитектуры,
3. создание приложения с одним обучающим режимом.

## **Весенний семестр**

1. реализация новых режимов,
2. внедрение метрик для оценки прогресса пользователя и ведения статистики,
3. добавление возможности использовать музыкальный инструмент для тренировок,
4. оценка результатов, тестирование, размещение приложения в магазинах.

# Обзор

## 2.1 Анализ существующих решений

Перед началом работы над приложением был проведён анализ существующих решений. Целью было выявление их преимуществ и недостатков, особенно важно было подчеркнуть удачные идеи, которые были бы полезны для пользователя.

В процессе сбора информации были рассмотрены наиболее информативные отзывы от пользователей *Google Play* на несколько приложений с наибольшим числом скачиваний. Ниже приведены результаты оценки таких приложений, как *Perfect Ear*(рис.1), *Functional Ear Trainer*(рис.2), *My Ear Trainer* (рис.3).

### 2.1.1 Perfect ear<sup>1</sup>

#### Достоинства:

- + Есть раздел с теорией, интегрированный в обучение;
- + Большое количество упражнений;
- + Оценка и поощрение достижений пользователя, дополнительная мотивация к занятиям;
- + Возможность просмотра статистики и выявления слабых мест.

#### Недостатки:

- Возможны баги, при которых результат пользователя оценивается некорректно;
- Интерфейс не уведомляет пользователя об ошибках;
- Интерфейс ориентирован только на знающих терминологию людей;
- Почти полностью отсутствует возможность настройки упражнений.

---

<sup>1</sup>Perfect ear (Google Play) — URL: <https://play.google.com/store/apps/details?id=com.evilduck.musiciankit>

## 2.1.2 Functional Ear Trainer<sup>2</sup>

### Достоинства:

- + Есть режим с полной настройкой упражнения;
- + Интерфейс адаптирован под всех пользователей (разные форматы отображения информации);
- + Большое количество настроек как интерфейса, так и всего остального.

### Недостатки:

- Музыкальные обозначения не подходят для всех пользователей (в России они другие);
- Упражнения не учитывают предыдущие попытки пользователя, возможны повторения;
- Из-за платной подписки урезан основной контент.

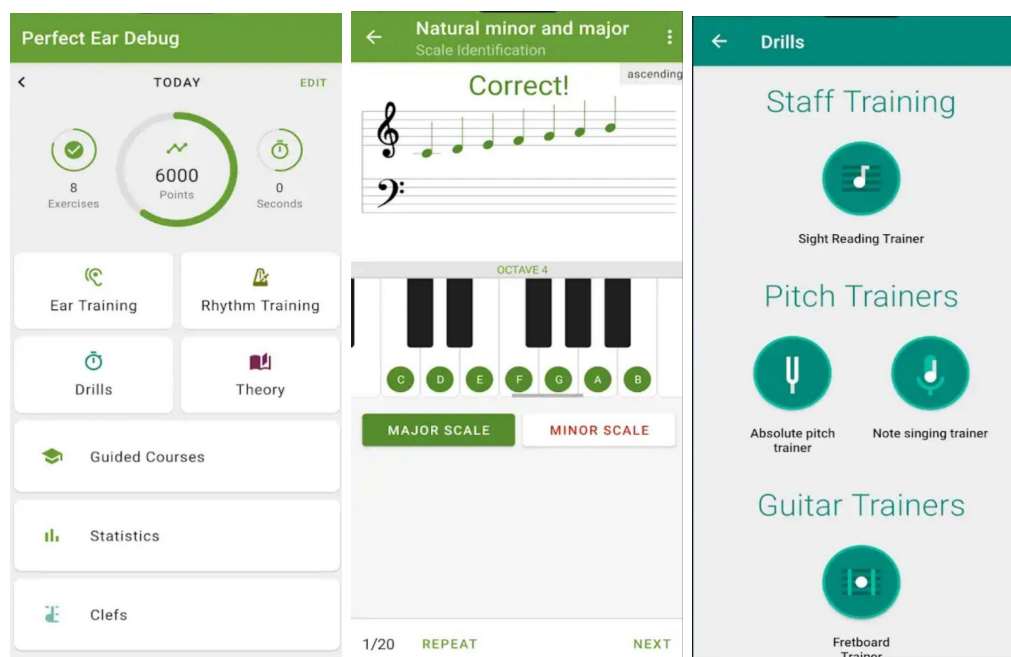
## 2.1.3 My Ear Trainer<sup>3</sup>

### Достоинства:

- + Уникальный режим - подбор целой мелодии на слух;
- + Большое разнообразие упражнений и выбор сложности для каждого;
- + Есть курсы с введением в теорию.

### Недостатки:

- Почти нет настроек интерфейса под разных пользователей;
- Нет поддержки некоторых языков и их обозначений.



(a) Главный экран.

(b) Упражнения.

(c) Доступные задания.

**Рис. 1:** Приложение Perfect Ear.

<sup>2</sup>Functional Ear Trainer (Google Play) — URL: <https://play.google.com/store/apps/details?id=com.kaizen9.fet.android>

<sup>3</sup>My Ear Trainer (Google Play) — URL: <https://play.google.com/store/apps/details?id=com.myapps.eartraining>

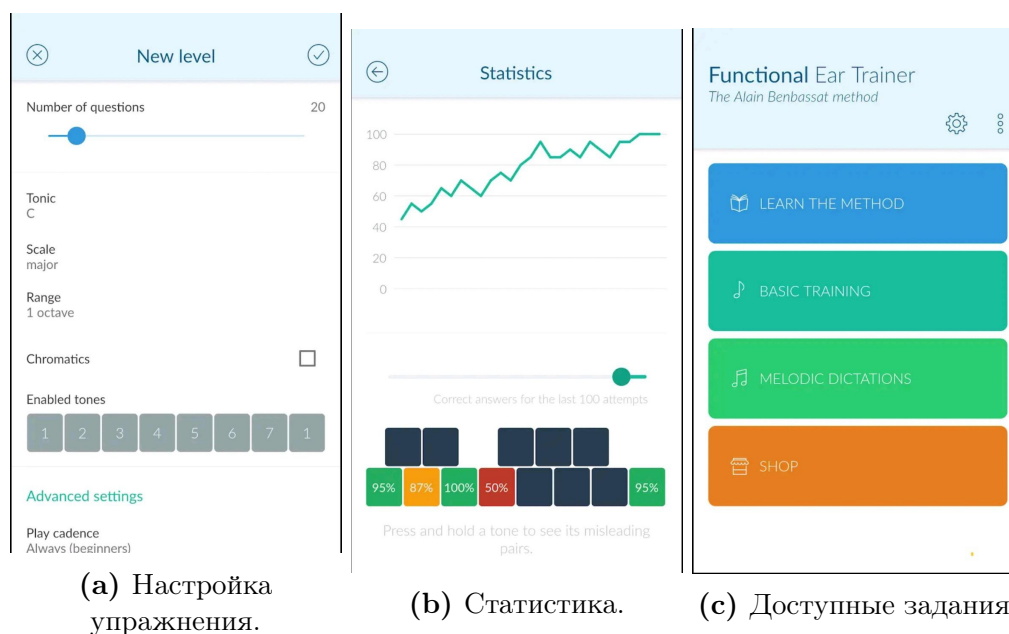


Рис. 2: Приложение Functional Ear Trainer.

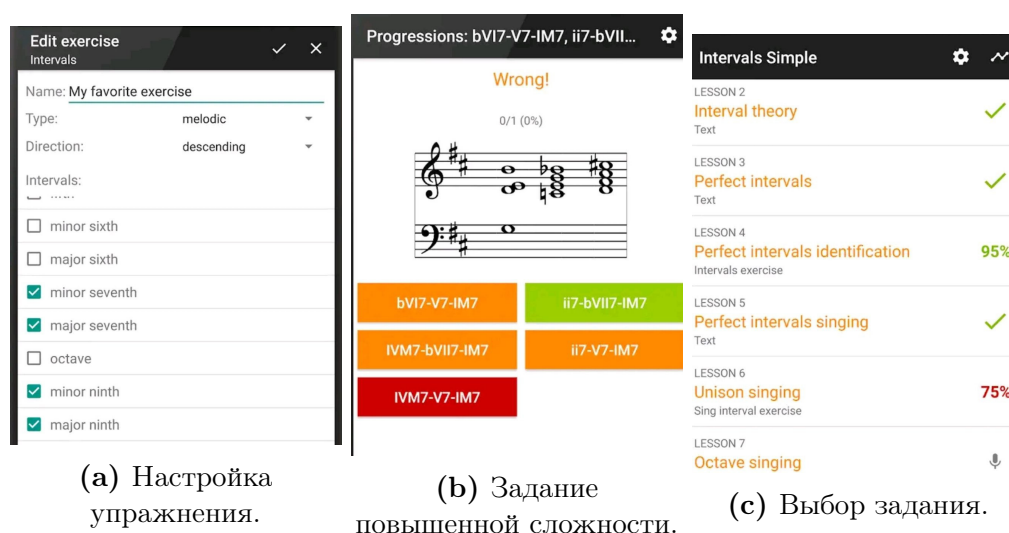


Рис. 3: Приложение Functional Ear Trainer.

## 2.1.4 Результаты анализа

Исходя из приведённого выше обзора, можно сделать вывод, что на данный момент уже существует несколько решений, предоставляющих пользователям возможность обучаться музыке, используя современные технологии. Это говорит о присутствии реальной потребности в приложениях, помогающих в освоении музыкальных навыков.

В то же время, ни одно из них полностью не решает проблемы. Почти в каждом приложении основной контент доступен лишь по подписке, интерфейс не адаптирован под пользователей с разными знаниями, иногда допускаются грубые ошибки при подборе или проверке заданий, доступно очень маленькое количество настроек как пользовательского интерфейса, так и самих упражнений. Более того, для русских пользователей, желающих учиться по принятым в нашей стране обозначениям, круг выбора сводится всего до нескольких вариантов.

Однако, при разработке представленных аналогов было реализовано множество хороших решений, которые следовало бы особо отметить: большое разнообразие различных упражнений с разделением на уровни сложности, есть возможность осваивать теорию (словари с

терминами, статьи) и параллельно закреплять знания на практике, почти во всех приложениях ведётся анализ достижений пользователя и сбор статистики. Также одна из интересных функций – режим, где возможна полная настройка занятия (представлено лишь в *Functional Ear Trainer*, но по отзывам многим это нужно).

## 2.2 Требования к приложению

Исходя из результатов проведённого анализа, были сформулированы требования. Обучающее приложение должно:

1. запускаться на устройствах под управлением ОС Android;
2. быть удобным в использовании как для людей, владеющих музыкальной терминологией и нотной грамотой, так и для новичков;
3. оценивать результаты пользователя, поощряя его за новые свершения, например возможность просмотра статистики, получения “достижений” (как вариант);
4. поддерживать возможность создавать расписание занятий и получать уведомления в запланированное для них время;
5. позволять использовать реальный инструмент как способ ввода информации для прохождения заданий;
6. иметь разнообразные упражнения для развития слуха с возможностью настройки сложности;
7. поддерживать разные языки и обозначения.

## 2.3 Используемые технологии

Основными критериями выбора языка программирования были: возможность запускать код на телефонах под управлением Android, наличие удобных инструментов для мобильной разработки (как библиотек и платформ, так и IDE) и удобный синтаксис. Исходя из этих критериев, для написания приложения был выбран **Kotlin**.

Kotlin на данный момент является одним из самых быстроразвивающихся языков программирования [3]. Кроме этого, он исполняется в *JVM* и имеет совместимость с Java-кодом, что невероятно полезно, так как за всё время их существования, накопилось огромное количество библиотек, платформ и прочих готовых решений для часто встречающихся проблем. Ещё одним преимуществом является поддержка Kotlin в одной из самых популярных IDE для мобильной разработки - *Android Studio*, ведь она позволяет значительно ускорить все этапы разработки продукта. Наличие у Kotlin понятного и удобного синтаксиса стало решающей причиной выбрать этот язык для разработки.

В качестве системы сборки проекта был выбран **Gradle**, в основном из-за быстрой сборки многомодульных приложений и удобстве в настройке.

Фреймворк **JUnit5**, предназначенный для написания тестов под JVM, является одним из самых популярных решений. Он приятен в использовании и имеет хорошую документацию, за что и был выбран.

Одним из самых важных аспектов разработки ПО является создание практичного и красивого пользователю интерфейса. Для этих целей идеально подходит платформа **Jetpack Compose**. Он позволяет создавать интерфейс с помощью коротких конструкций кода на Kotlin, имеет низкий порог вхождения, но при этом даёт возможность создавать современный интерфейс.



# Реализация

## 3.1 Архитектура

Ещё на этапе планирования архитектуры приложения было принято решение выделить для логики, связанной с музыкой, отдельный модуль. На данный момент все классы, представляющие собой ключевые музыкальные абстракции (такие как: нота, мелодия, длительность и т.д.), находятся в модуле **MusicLib**. Во втором модуле (**App**) содержатся UI-компоненты и некоторые классы, позволяющие связывать пользовательский интерфейс и классы из MusicLib. Благодаря такому разделению появляется возможность гораздо проще проводить тестирование и логически отделять разные абстракции друг от друга.

Для представления главных музыкальных абстракций на данный момент существует несколько классов: *Interval*, *NoteRange*, *Melody*, *Note*, *Pause* и *MelodyNote*. Для хранения мелодии применяется класс *Melody*, а для её построения необходима последовательность из нот (*MelodyNote*), либо музыкальных пауз. При этом важным является тот факт, что для представления ноты было принято решение сделать сразу два класса, главное их отличие в том, что *Note* представляет собой просто звук определённой высоты, а *MelodyNote* содержит дополнительные свойства, позволяющие более детально управлять звуком. Классы *NoteRange* и *Interval* представляют собой соответственно диапазон нот (например, для фортепиано) и музыкальный интервал (расстояние между звуками). Кроме этого, существует также множество классов и перечислений для описания различных музыкальных свойств.

Модуль App содержит реализацию классов, связывающих описанные выше классы и интерфейс, доступный пользователю. На данный момент реализована возможность создавать свои виртуальные инструменты с помощью наследования от класса *AbstractInstrument* и возможность извлекать из них звуки соответствующих инструментов с помощью *MelodyPlayer*.

Было реализованно также и несколько ключевых элементов интерфейса. Например, класс *PianoKeyboard* позволяет отрисовывать на экране клавиатуру переданных размеров и с указанным диапазоном клавиш. Этот элемент полезен как для пользователей, не имеющих реального инструмента, так и для интерфейса некоторых заданий.

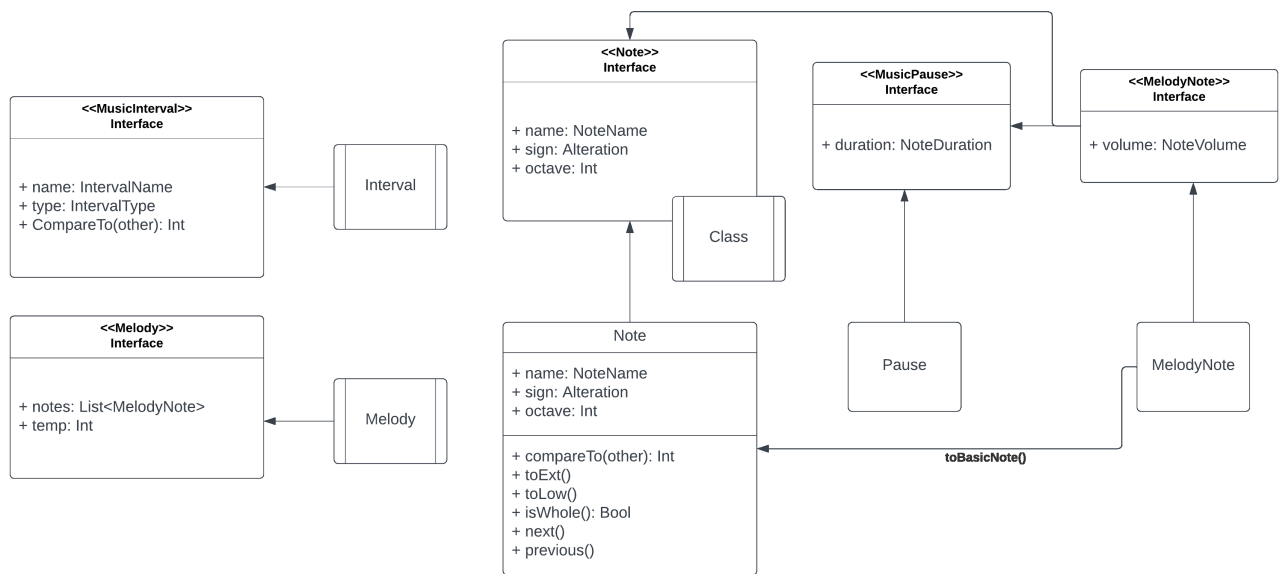


Рис. 1: UML-диаграмма библиотеки.

## 3.2 Интерфейс приложения

Главными требованиями к пользовательскому интерфейсу были: удобство в использовании для людей, не знакомых со сложными обозначениями и терминами, современный дизайн, поддержка разных языков. Важно было учесть, что основные пользователи приложения (учащиеся музыкальных школ) также требуют особого подхода, так как сложный и перегруженный деталями интерфейс может отвлекать ребёнка и затруднять понимание материала.

Цветовая палитра приложения (рис. 2) была специально подобрана так, чтобы не отвлекать от упражнений. Цвета приглушены, но при этом хорошо сочетаются друг с другом.

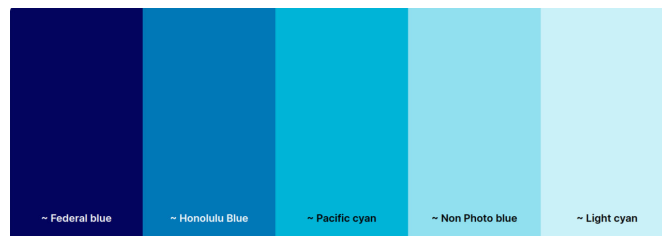
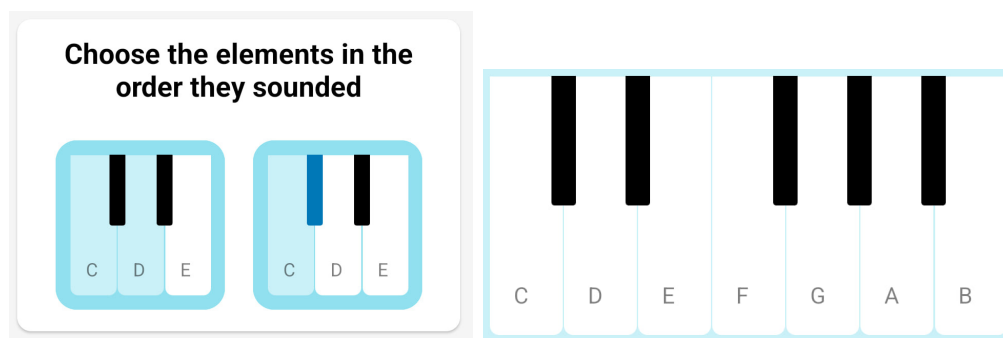


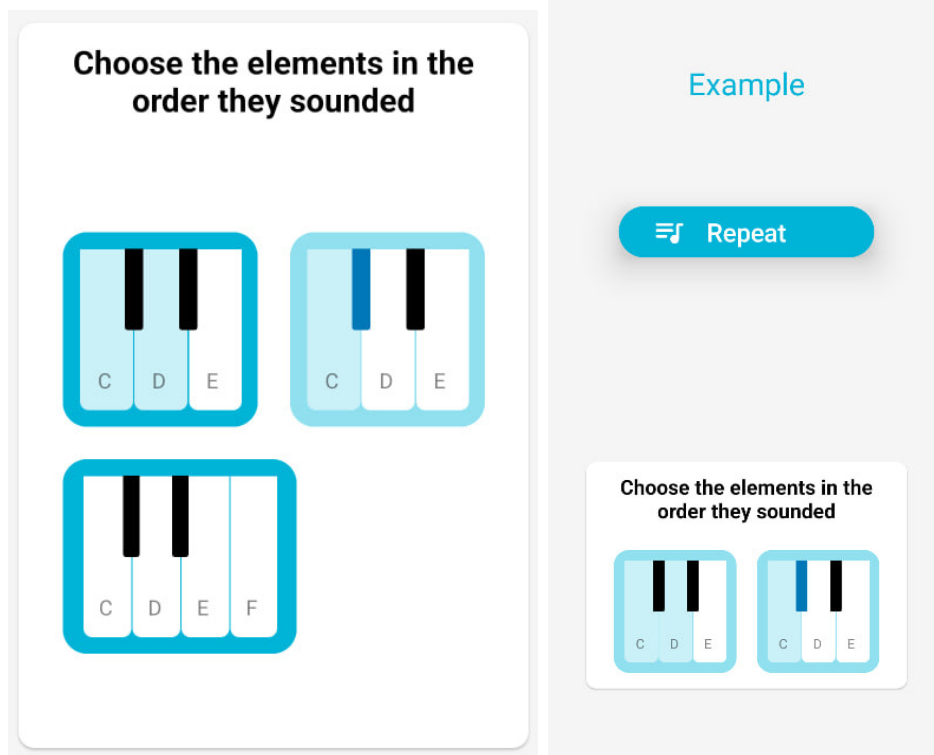
Рис. 2: Цветовая палитра приложения

Формы элементов интерфейса также подбирались исходя из предположений о том, что обилие углов и деталей будет отвлекать и слишком бросаться в глаза. Было создано несколько дополнительных форм и анимаций для специальных элементов, так как подходящих в наборе Jetpack Compose найти не удалось.



(a) Piano Checkbox.

(b) Piano Keyboard.



(c) Piano Checkbox (с выбранными вариантами).

(d) Примерный вид упражнения.

**Рис. 3:** Основные элементы пользовательского интерфейса.

# Апробация

Для проверки удобства использования приложения был выбран специальный опросник - System Usability Scale [4]. В опросе приняли участие трое студентов, закончивших музыкальную школу. Приложение было оценено на 93,0 балла, что свидетельствует о его простоте и удобстве в освоении. Также были выявлены некоторые недостатки системы, которые в будущем будут устранены.

# Заключение

За осенний семестр в рамках учебной практики были решены следующие задачи:

- Выполнен обзор существующих решений.
- Собраны требования для приложения.
- Изучены необходимые для его реализации технологии.
- Разработана архитектура приложения.
- Созданы некоторые элементы пользовательского интерфейса.
- Реализован первый обучающий режим.

Подробнее с результатами работы можно ознакомиться в репозитории GitHub <sup>1</sup>.

---

<sup>1</sup>Репозиторий проекта — URL: <https://github.com/d-zaytsev/android-app>

# Литература

1. *Thamprasert Yada*. Network analysis of relationship in hobbies interest among 50 countries and the changes from COVID-19: Ph.D. thesis / Chiang Mai: Graduate School, Chiang Mai University. — 2023.
2. Music interventions and child development: A critical review and further directions / Elisabeth Dumont, Elena V Syurina, Frans JM Feron, Susan van Hooren // *Frontiers in psychology*. — 2017. — Vol. 8. — P. 1694.
3. *Github*. The top programming languages. — URL: <https://octoverse.github.com/2022/top-programming-languages>.
4. *Brooke John*. Sus: a “quick and dirty” usability // *Usability evaluation in industry*. — 1996. — Vol. 189, no. 3. — Pp. 189–194.