Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский**

**технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

Допускаю к защите

|  |  |
| --- | --- |
| Руководитель |  |
|  | подпись |
|  | В.Л. Аршинский |
|  | И.О. Фамилия |

|  |
| --- |
|  |
| Разработка приложения с использованием объектно-ориентированного подхода |
| наименование темы |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

|  |
| --- |
| Объектно-ориентированное программирование |

|  |
| --- |
| 1.007.00.00 – ПЗ |
| обозначение документа |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | ИСТб-19-2 |  |  |  | И.В. Карбушев |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Нормоконтроль |  |  |  |  |  | В.Л. Аршинский |
|  |  |  |  | подпись |  | И.О. Фамилия |
| Курсовой проект защищен с оценкой | | | |  | | |

Иркутск 2021 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

ЗАДАНИЕ

на курсовое проектирование

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| По курсу | Объектно-ориентированное программирование | | | | | | | | | | | | | | | | | |
| Студенту | Карбушеву И.В. | | | | | | | | | | | | | | | | | |
|  | (фамилия, инициалы) | | | | | | | | | | | | | | | | | |
| Тема проекта | | Разработка приложения с использованием объектно-ориентированного подхода | | | | | | | | | | | | | | | | |
| Исходные данные | | | Разработать Android-приложение для прослушивания музыкальных композиций и интернет-радиоканалов, и создания из них очередей | | | | | | | | | | | | | | | |
|  | | | | | | | | | |  |  |  |  |  |  |  |  |  |
| Графическая часть на | | | | листах. | | | | | | | | | | | | | | |
| Дата выдачи задания | | | | « |  | » |  | | | | | 2021 г. | | | | |  |  |
|  | | | | | | | | | | | | | | | | | | |
| Задание получил | | | | | | | |  | | | | |  | | И.В. Карбушев | | | |
|  | | | | | | | | подпись | | | | |  | | И.О. Фамилия | | | |
|  | | | | | | | | |  | |  | | |  | |  | |  |
| Дата представления проекта руководителю | | | | | | | | | « | |  | | | » | | мая | | 2021 г. |
|  | | | | | | | | | | | | | | | | | | |
| Руководитель курсового проектирования | | | | | | | |  | | | | |  | | В.Л. Аршинский | | | |
|  | | | | | | | | подпись | | | | |  | | И.О. Фамилия | | | |

**Содержание**

[Введение 4](#_Toc73015374)

[1 Анализ задания и описание предметной области 5](#_Toc73015375)

[1.1 Постановка задач 5](#_Toc73015376)

[1.2 Функциональные требования 6](#_Toc73015377)

[1.3 Выбор средств реализации 7](#_Toc73015378)

[2 Проектирование приложения 8](#_Toc73015379)

[2.1 Проектирование интерфейса 8](#_Toc73015380)

[2.2 Проектирование классов 10](#_Toc73015381)

[2.3 Проектирование базы данных 12](#_Toc73015382)

[3 Реализация приложения 14](#_Toc73015383)

[3.1 Спецификация реализованных классов 14](#_Toc73015384)

[4 Тестирование 16](#_Toc73015385)

[4.1 Методика тестирования 16](#_Toc73015386)

[4.2 Разработанные тесты 17](#_Toc73015387)

[4.3 Результаты тестирования 22](#_Toc73015388)

[Заключение 42](#_Toc73015389)

[Список использованных источников 43](#_Toc73015390)

# Введение

Целью курсового проекта является закрепление теоретических знаний полученных в процессе изучения курса «Объектно-ориентированное программирование» и развитие практических навыков применения объектного подхода в ходе решения конкретной практической задачи – разработки приложения с использованием объектно-ориентированного подхода, а также развитие способностей к обоснованному принятию самостоятельных решений в ходе проектной деятельности и получение базового опыта разработки программного обеспечения.

Задача – создать Android-приложение «EarFeeder» - музыкальный плеер с возможностью корректировать очереди прослушивания под себя и добавления своих интернет-радиоканалов, при этом управляя им, как изнутри приложения, так и извне. Все музыкальные композиции должны браться из памяти телефона и записываться в локальную базу данных.

1. Анализ задания и описание предметной области

## Постановка задач

В ходе разработки необходимо решить следующие задачи:

1. Выработать функциональные требования к приложению
2. Выполнить проектирование интерфейса
3. Реализовать проектирование классов
4. Произвести проектирование базы данных
5. Разработать приложение, основываясь на вышеперечисленных пунктах
6. Определить методику тестирования и проверить работу приложения
7. Резюмировать результаты курсового проекта
   1. Функциональные требования

Приложение должно выполнять функции плеера:

1. Работа с музыкальными треками
   1. Сбор всех музыкальных файлов с устройства в один список
   2. Организация взаимодействия пользователя с музыкальными композициями путем нажатия на них, чтобы запустить воспроизведение
   3. Возможность прослушивания композиций в фоновом режиме
   4. Возможность управлять (ставить на паузу, продолжать воспроизведение, переключать вперед и назад) композициями внутри приложения и через меню уведомлений
   5. Возможность добавлять композиции в «избранное» и отдельно прослушивать их
   6. Возможность управления композициями путем перемотки, перемешивания и зацикливания
2. Работа с интернет-радиоканалами
   1. Просмотр списка сохраненных радиоканалов
   2. Добавление нового интернет-радиоканала в список по ссылке
   3. Удаление радио из списка
   4. Выбор и прослушивание радио из списка
   5. Управление воспроизведением радио из приложения и меню уведомлений
3. Организация взаимодействия пользователя и приложения через дополнительные элементы управления
   1. Возврат в приложение при нажатии на меню уведомлений
   2. Удобный переход между экранами через боковое меню навигации
   3. Общий нижний фрагмент приложения для управления композициями
   4. Возможность переходить назад по экранам без специальных функций телефона
4. Работа с очередями композиций
   1. Создание плейлистов
   2. Редактирование плейлистов
   3. Удаление плейлистов
   4. Перемешивание плейлистов
   5. Сохранение плейлистов в локальную базу данных
5. Пользовательские настройки приложения
   1. Возможность выбора цветовой схемы (красный, зеленый, голубой)
   2. Возможность сбросить все сохраненные материалы в приложении (музыкальные композиции с устройства останутся на месте, в списке)
   3. Выбор средств реализации

Были использованы следующие технологии:

* + - 1. Android studio – IDE для разработки приложения под ОС Android
      2. Room – ORM библиотека для создания и управления базы данных на основе SQLite
      3. Gauvark Audio visualizer – библиотека для визуализации музыкальных композиций в виде всплывающих столбцов

# Проектирование приложения

* 1. Проектирование интерфейса

На основе анализа функциональных требований в приложении были выделены следующие экраны:

1. Приветственный экран – всплывает на 2 секунды после включения приложения (если запуск первый или приложение долго не запускалось).
2. Главный экран выбора треков – на нем находится список всех найденных треков на устройстве, композиции можно запустить и прослушать. Так же отсюда можно попасть в любой экран приложения.
3. Экран выбора радио – позволяет запустить и прослушать интернет-радиоканалы, уже хранящиеся в приложении или добавленные пользователем.
4. Экран выбора плейлистов – позволяет просмотреть или создать плейлисты.
5. Экран создания плейлиста – инструмент для создания плейлистов из всех композиций на устройстве.
6. Экран просмотра плейлиста – показывает все композиции из выбранного плейлиста.
7. Экран редактирования плейлиста – возможность добавить в существующий плейлист другие композиции из оставшихся треков на устройстве.
8. Экран выбора любимых песен – отображает композиции, которые были отмечены пользователем, как «любимые».
9. Экран настроек приложения – позволяет выбрать цветовую схему приложения из предложенных трех (красный, зеленый, голубой) и очистить все данные, кроме списка песен с устройства.
10. Детальное окно играющего трека – возможность расширенного управлению текущей композицией: перемотка, перемешивание, повтор, переключение.

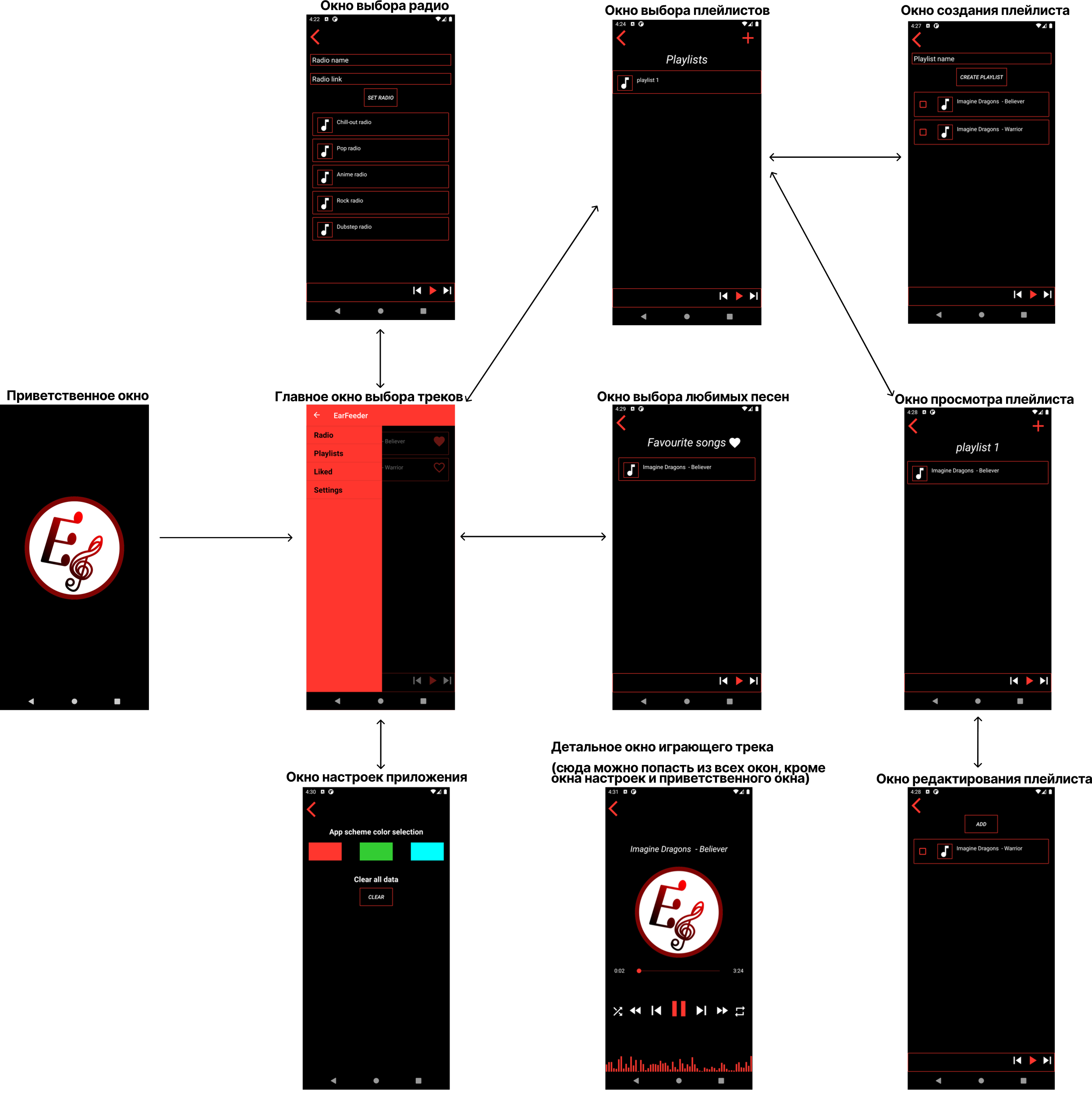


Рисунок 1 – Диаграмма устройства интерфейса приложения

* 1. Проектирование классов

Для работы приложения с базой данных были определены следующие сущности:

1. Track – музыкальная композиция, взятая из хранилища пользователя.
2. Radio – интернет-радиоканал
3. Playlist – список музыкальных композиций, собранных лично пользователем
4. TrackPlaylist – принадлежность музыкальных композиций к плейлистам

Были выделены следующие группы классов:

1. Database – на основе принадлежности к базе данных (сущности, запросы и абстрактный класс, соединяющий все классы-запросы)
2. Service – на основе наследование от класса Service
3. Player – отдельный класс для управления музыкальным плеером
4. App – запускающий класс, содержащий все важные переменные и методы, позволяющие приложению функционировать без лишних переменных, храня единственный экземпляр нужных значений

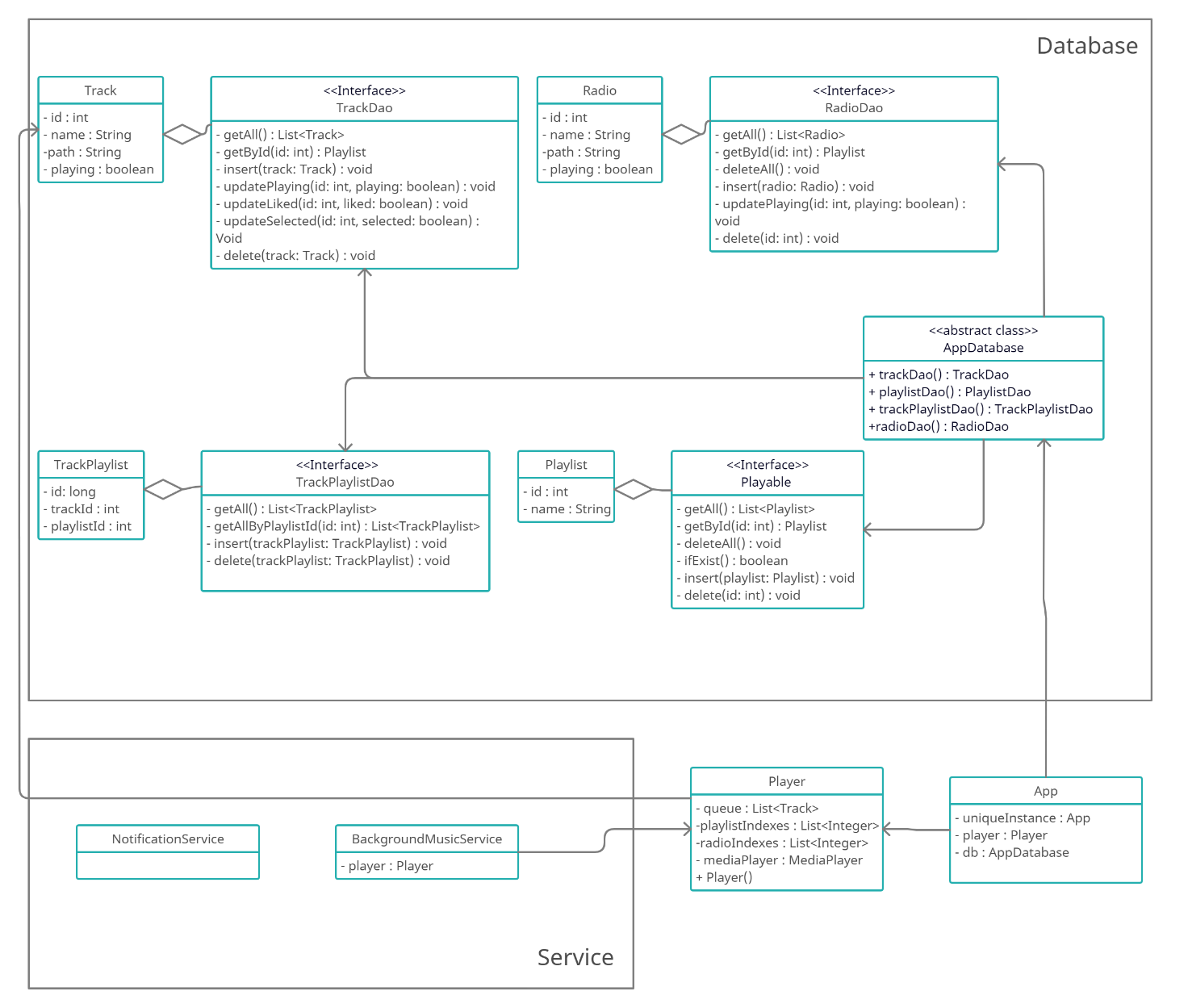


Рисунок 2 – Диаграмма UML, первый фрагмент

Для взаимодействия пользователя с приложением были определены следующие группы классов:

1. Activity – группа экранов приложения
2. Playable – интерфейс, реализующий взаимодействие между плеером и меню уведомлений

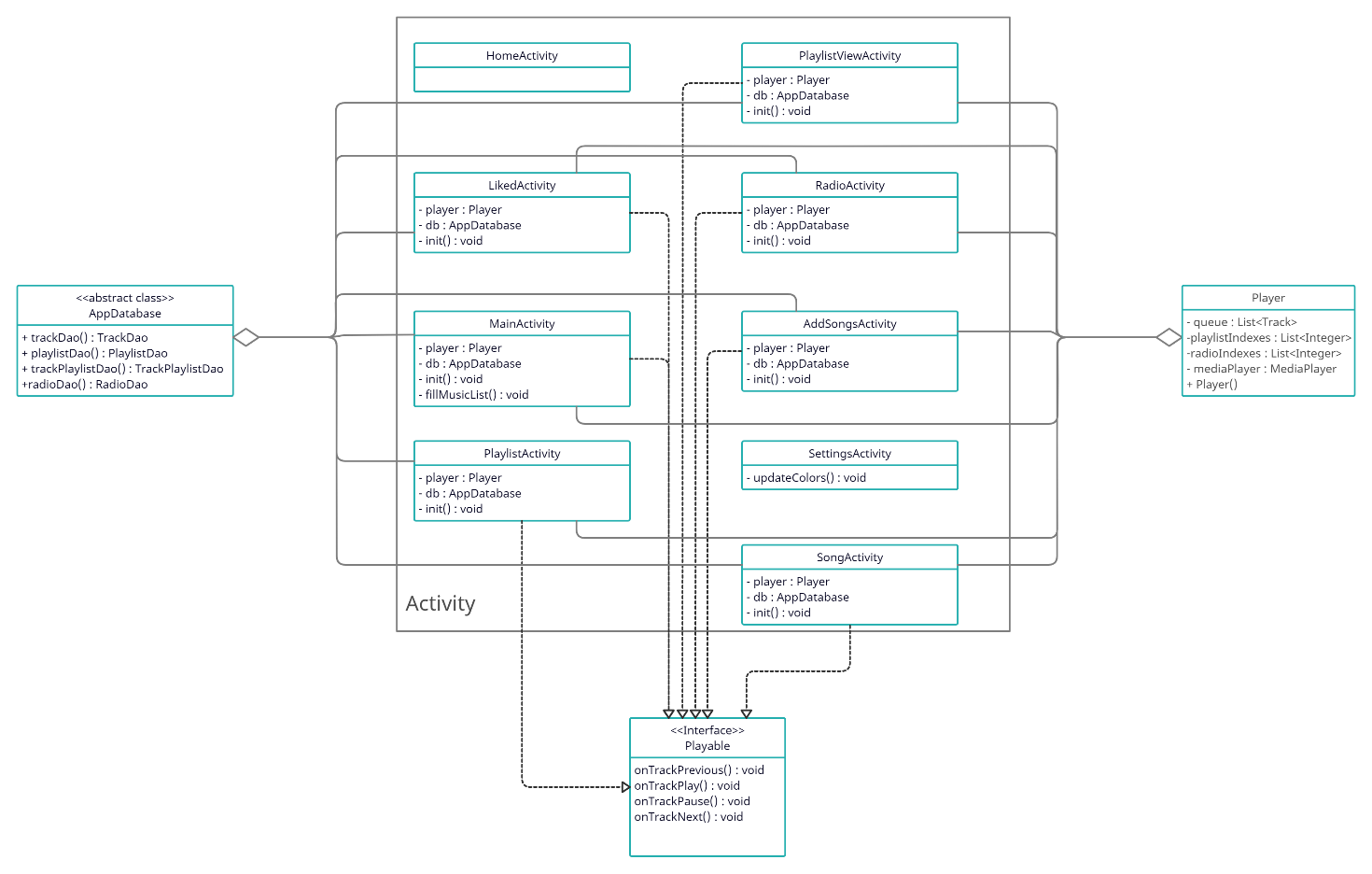


Рисунок 3 – Диаграмма UML, второй фрагмент

* 1. Проектирование базы данных

Для работы с базой данных на основе упомянутых сущностей в проектировании классов были созданы следующие таблицы.

Первичным ключом во всех четырех таблицах является их уникальный идентификатор id, по которому таблицы Track, Playlist и TrackPlaylist могут «общаться» друг с другом.

В таблице TrackPlaylist хранятся id треков и плейлистов, к которым они принадлежат. В то время, как таблица Radio существует сама по себе, хранит только записи о интернет-радиоканалах и не взаимодействует ни с какой другой таблицей.

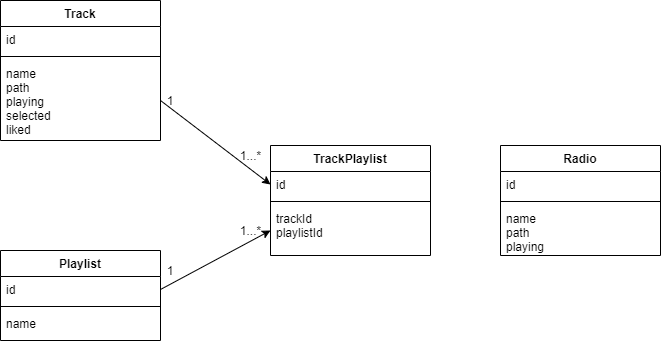


Рисунок 4 - диаграмма классов базы данных

Таблица 1 – Описание таблиц БД

|  |  |  |  |
| --- | --- | --- | --- |
| Имя таблицы | Имя атрибута | Тип | Описание |
| Track | id | int | Идентификатор трека |
| name | string | Название трека |
| path | string | Путь до трека |
| playing | boolean | Играет ли трек сейчас |
| selected | boolean | Выбран ли он в меню выбора треков |
| liked | boolean | Находится ли трек в «любимых» |
| Playlist | id | int | Идентификатор плейлиста |
| name | string | Название плейлиста |

Таблица 1 - Продолжение

|  |  |  |  |
| --- | --- | --- | --- |
| TrackPlaylist | id | long | Идентификатор записи |
| trackId | int | Идентификатор трека |
| playlistId | int | Идентификатор плейлиста, которому принадлежит трек |
| Radio | id | int | Идентификатор радио |
| name | string | Название радио |
| path | string | Ссылка на радио |
| playing | boolean | Играет ли радио сейчас |

**SQL-запросы к БД**

Track:

1. SELECT \* FROM Track
2. SELECT \* FROM Track WHERE id = :id
3. UPDATE Track SET playing = :playing WHERE id = :id
4. UPDATE Track SET liked = :liked WHERE id = :id
5. UPDATE Track SET selected = :selected WHERE id = :id
6. INSERT Track
7. DELETE Track

Playlist:

1. SELECT \* FROM Playlist
2. SELECT \* FROM Playlist WHERE id = :id
3. DELETE FROM Playlist
4. SELECT EXISTS(SELECT \* FROM Playlist)
5. DELETE FROM Playlist WHERE id LIKE :id
6. INSERT Playlist

TrackPlaylist:

1. SELECT \* FROM TrackPlaylist
2. SELECT \* FROM TrackPlaylist WHERE playlistId = :id
3. DELETE FROM TrackPlaylist
4. INSERT TrackPlaylist
5. DELETE TrackPlaylist

Radio:

1. SELECT \* FROM Radio
2. SELECT \* FROM Radio WHERE id = :id
3. DELETE FROM Radio
4. UPDATE Radio SET playing = :playing WHERE id = :id
5. DELETE FROM Radio WHERE id LIKE :id
6. INSERT Radio

# Реализация приложения

* 1. Спецификация реализованных классов

Activity:

1. HomeActivity – приветственный экран с логотипом приложения
2. LikedActivity –экран с любимыми песнями пользователя
3. MainActivity – главный экран со всеми треками с устройства
4. PlaylistActivity –экран с просмотром и возможностью добавления плейлистов
5. PlaylistViewActivity – экран с просмотром песен в выбранном плейлисте
6. RadioActivity – экран с возможностью прослушивания и добавления радио
7. AddSongsActivity – экран с возможность добавления песен в выбранный плейлист
8. CreatingPlaylistActivity – экран с созданием плейлиста
9. SettingsActivity – экран с настройками приложения
10. SongActivity – экран текущей песни с расширенными возможностями управления

Adapter:

1. LikeTrackAdapter – редактирование показа любимых песен в списке
2. PlaylistAdapter – редактирование показа плейлистов в списке
3. TrackAdapter - редактирование показа песен в списке
4. RadioAdapter - редактирование показа радио в списке
5. TrackAdapterSelect – редактирования списка песен с возможностью выбора (checkbox)

ORM (Object-Relational Mapping) – техника, создающая виртуальную объектную БД и добавляющая дополнительный слой абстракции на работу с ней.

DAO (Data Access Object) - интерфейс, отвечающий за передачу запросов в БД и обработку полученных от нее ответов.

Database:

1. Playlist – сущность «плейлист»
2. PlaylistDao – запросы к сущности «плейлист»
3. Radio – сущность «радио»
4. RadioDao – запросы к сущности «радио»
5. Track – сущность «трек»
6. TrackDao – запросы к сущности «трек»
7. TrackPlaylist – таблица связей «треков» и «плейлистов»
8. AppDatabase – набор абстрактных ссылок на классы запросов

Service:

1. BackgroundMusicService – сервис, который организует воспроизведение музыки
2. NotificationService – сервис, который позволяет отображать уведомления
3. NotificationActionService – сервис, который организует отправку действий, которые происходят на уведомлении

Notification:

1. CreateNotification – создание уведомления
2. Playable – интерфейс, позволяющий управлять уведомлением

Fragment:

1. BottomFragment – нижняя панель приложения для управления треками
2. ClearAllFragment – кнопка очистки данных приложения
3. ColorSelectionFragment – панель управления цветовой схемой приложения

LoadingDialog – управление панелью загрузки

RecyclerItemListener – переопределение событий нажатия на RecyclerView

AppColor – организация управления цветовой схемой приложения

Player – все переменные и методы, нужные для воспроизведения музыки

App – запускающий класс для управления и хранения главных состояний приложения, основан на паттерне Singleton

Singleton – паттерн, позволяющий иметь лишь единственный экземпляр некоторого класса через предоставление глобального доступа к этому экземпляру.

# Тестирование

## Методика тестирования

В качестве метода тестирования было выбрано приемочное тестирование.

Целью данного метода является определение готовности продукта, что достигается путем прохода тестовых сценариев и случаев, которые построены на основе спецификации требований к разрабатываемому ПО.

## Разработанные тесты

Таблица 2 – Тесты в главном меню MainActivity

|  |  |  |
| --- | --- | --- |
| № | Совершенное действие | Ожидаемый результат |
| 1 | Запуск трека в главном меню. | Начинает играть нажатая песня, появляется знак того, что эта песня играет, на меню снизу и меню уведомлений в выпадающем верхнем меню телефона знак «запустить» поменяется на знак «пауза». |
| 2 | Нажатие на кнопку «нравится» в форме сердца на песне. | Если песня уже была в «понравившихся», то сердце кнопка станет без заполнения, иначе – заполнится выбранным цветом. Трек добавится в меню «понравившиеся». |
| 3 | Нажатие на кнопку «следующий» на нижней панели приложения. | Песня переключится на следующую в списке, название на панели снизу поменяется. Поменяется иконка играющего трека. |
| 4 | Нажатие на кнопку «следующий» на нижней панели приложения, когда дальше по списку нет песен. | Ничего не произойдет. |
| 5 | Нажатие на кнопку «предыдущий» на нижней панели приложения. | Песня переключится на предыдущую в списке, название на панели снизу поменяется. Поменяется иконка играющего трека. |
| 6 | Нажатие на кнопку «предыдущий» на нижней панели приложения, когда перед текущим треком нет ничего. | Ничего не произойдет. |
| 7 | Нажатие на кнопку «пуск» / «пауза» на нижней панели приложения. | Песня начнет играть/остановится и поменяется изображение на кнопке на нижней панели приложения. |
| 8 | Нажатие на кнопку «пуск» / «пауза», «следующий» или «предыдущий» на нижней панели приложения, когда в начале работы приложения еще не запущен никакой трек. | Ничего не произойдет. |
| 9 | Пункт 3, но нажатие происходит на меню уведомлений. | См пункт 3. |
| 10 | Пункт 4, но нажатие происходит на меню уведомлений. | См пункт 4. |

Таблица 2 - Продолжение

|  |  |  |
| --- | --- | --- |
| 11 | Пункт 5, но нажатие происходит на меню уведомлений. | См пункт 5. |
| 12 | Пункт 6, но нажатие происходит на меню уведомлений. | См пункт 6. |
| 13 | Пункт 7, но нажатие происходит на меню уведомлений. | См пункт 7. |
| 14 | Пункт 8, но нажатие происходит на меню уведомлений. | См пункт 8. |
| 15 | Нажатие на название песни на нижней панели приложения. | Открытие activity с расширенным функционалом управления текущей песни. |
| 16 | Нажатие на название песни на нижней панели приложения, когда в начале работы приложения еще не запущен никакой трек или играет радио. | Ничего не произойдет. |
| 17 | Нажатие на названия других activity в боковом выдвигающемся меню. | Будет произведет переход на соответствующие activity. |

Таблица 3 – Тесты на SongActivity

|  |  |  |
| --- | --- | --- |
| № | Совершенное действие | Ожидаемый результат |
| 18 | Перемещение тайм-кода песни (Seekbar). | Песня окажется на тайм-коде, куда было произведено нажатие. |
| 19 | Перемещение тайм-кода песни во время паузы. | Песня начнет играть с установленного тайм-кода. |
| 20 | Нажатие кнопки «назад». | Действие вернется на главное меню. |
| 21 | Нажатие кнопки «перемотать назад». | Вернет песню на 10 секунд назад. |
| 22 | Нажатие кнопки «перемотать назад», когда от начала песни прошло меньше 10 секунд. | Песня вернется на нулевую секунду. |
| 23 | Нажатие кнопки «перемотать вперед». | Перемотает песню на 10 секунд вперед. |
| 24 | Нажатие кнопки «перемотать вперед», когда до конца песни осталось меньше 10 секунд. | Начнет играть следующая песня. Если дальше нет песен, то воспроизведение окончится. |
| 25 | Нажатие кнопки «предыдущий». | Начнется играть предыдущая песня по списку. |
| 26 | Нажатие кнопки «предыдущий», когда нет предшествующих песен. | Текущая песня возобновится. |

Таблица 3 - Продолжение

|  |  |  |
| --- | --- | --- |
| 27 | Нажатие кнопки «следующий». | Начнется играть следующая песня по списку. |
| 28 | Нажатие кнопки «следующий», когда нет последующих песен. | Воспроизведение окончится. |
| 29 | Нажатие кнопки «повторить». | Если еще не была нажата – песня будет играть на повторе, иначе – песни будут играть по порядку списка. Изменится значок (не закрашенный / закрашенный). |
| 30 | Нажатие кнопки «перемешать». | Песни перемешаются в случайном порядке, приоритет отдается любимым песням. Сразу начнется воспроизведение с первой песни из списка перемешанных. Изменится значок (не закрашенный / закрашенный). |

Таблица 4 – Тесты на Settings

|  |  |  |
| --- | --- | --- |
| № | Совершенное действие | Ожидаемый результат |
| 31 | Нажатие на кнопки цветов. | Переключение цветовой схемы в цвет кнопки. |
| 32 | Нажатие на кнопку «очистить». | Очистятся все плейлисты, любимые песни и радио. |

Таблица 5 – Тесты на LikedActivity

|  |  |  |
| --- | --- | --- |
| № | Совершенное действие | Ожидаемый результат |
| 33 | Нажатие на трек. | См пункт 1. |
| 34 | Нажатие на кнопку «назад». | См пункт 20. |

Таблица 6 – Тесты на PlaylistActivity

|  |  |  |
| --- | --- | --- |
| № | Совершенное действие | Ожидаемый результат |
| 36 | Нажатие на плейлист. | Откроется выбранный плейлист в новом activity. |
| 37 | Нажатие на кнопку «назад». | См пункт 20. |
| 38 | Нажатие на кнопку «добавить». | Откроется CreatePlaylistActivity, где можно будет создать плейлист. |

Таблица 7 – Тесты на CreatePlaylistActivity

|  |  |  |
| --- | --- | --- |
| № | Совершенное действие | Ожидаемый результат |
| 39 | Нажатие на трек. | См пункт 1. |
| 40 | Нажатие на checkbox у трека. | Checkbox нажмется, либо отожмется. |
| 41 | Нажатие на кнопку «создать плейлист». | Все выбранные (checkbox’ом) треки образуют новый плейлист на PlaylistActivity. |
| 42 | Нажатие на кнопку «назад». | Действие вернется на PlaylistActivity. |

Таблица 8 – Тесты на PlaylistViewActivity

|  |  |  |
| --- | --- | --- |
| № | Совершенное действие | Ожидаемый результат |
| 43 | Нажатие на трек. | См пункт 1. |
| 44 | Нажатие на кнопку «назад». | Действие вернется на PlaylistActivity. |
| 45 | Нажатие на кнопку «добавить». | Откроется AddSongsActivity, где можно будет добавить треки в данный плейлист. |

Таблица 9 – Тесты на AddSongsActivity

|  |  |  |
| --- | --- | --- |
| № | Совершенное действие | Ожидаемый результат |
| 46 | Создание activity. | Появятся все песни, не принадлежащие выбранному плейлисту. |
| 47 | Нажатие на трек. | См пункт 1. |
| 48 | Нажатие на checkbox у трека. | Checkbox нажмется, либо отожмется. |
| 49 | Нажатие на кнопку «добавить». | Выбранные треки добавятся в выбранный плейлист. |
| 50 | Нажатие на кнопку «назад». | Действие вернется на PlaylistViewActivity. |

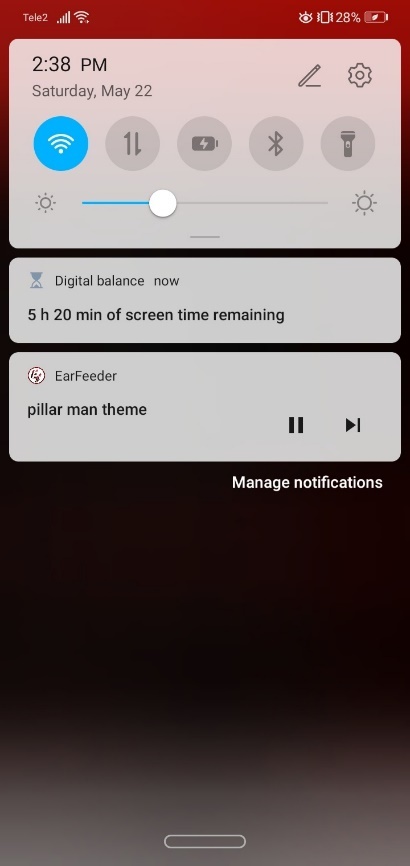
Таблица 10 – Тесты на RadioAcitivty

|  |  |  |
| --- | --- | --- |
| № | Совершенное действие | Ожидаемый результат |
| 51 | Нажатие на радио. | На экране появится колесо загрузки, которое нельзя прервать. Радио начнет воспроизводиться. |
| 52 | Нажатие на радио, когда нет интернета. | Всплывет сообщение «Нет интернет подключения». |
| 53 | Нажатие на кнопку «назад». | См пункт 20. |
| 54 | Нажатие на кнопку «добавить». | На экране появится колесо загрузки, которое нельзя прервать. Радио добавится в список и начнет воспроизводиться. |
| 55 | Нажатие на кнопку «добавить», когда нет интернета. | Всплывет сообщение «Нет интернет подключения». |
| 56 | Нажатие на кнопку «добавить», когда ссылка неверная. | Всплывет сообщение «Неверная ссылка». |

Таблица 10 - Продолжение

|  |  |  |
| --- | --- | --- |
| № | Совершенное действие | Ожидаемый результат |
| 57 | Нажатие на кнопку «старт» / «пауза» на нижней панели приложения, когда играет радио. | Появится загрузка, пока не включится радио. Поменяется иконка воспроизведения. |
| 58 | Нажатие на кнопку «следующий» на нижней панели приложения, когда играет радио. | Появится загрузка, пока радио не переключится на последующее. |
| 59 | Нажатие на кнопку «следующий» на нижней панели приложения, когда дальше нет радио. | Ничего не произойдет. |
| 60 | Нажатие на кнопку «предыдущий» на нижней панели приложения, когда играет радио. | Появится загрузка, пока радио не переключится на предыдущее. |
| 61 | Нажатие на кнопку «предыдущий» на нижней панели приложения, когда перед текущим больше нет радио. | Ничего не произойдет. |

## Результаты тестирования



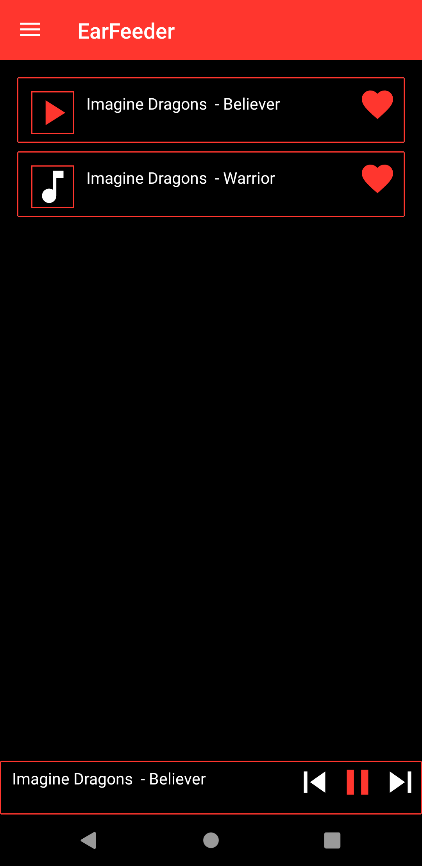


Рисунок 5 - результат теста 1

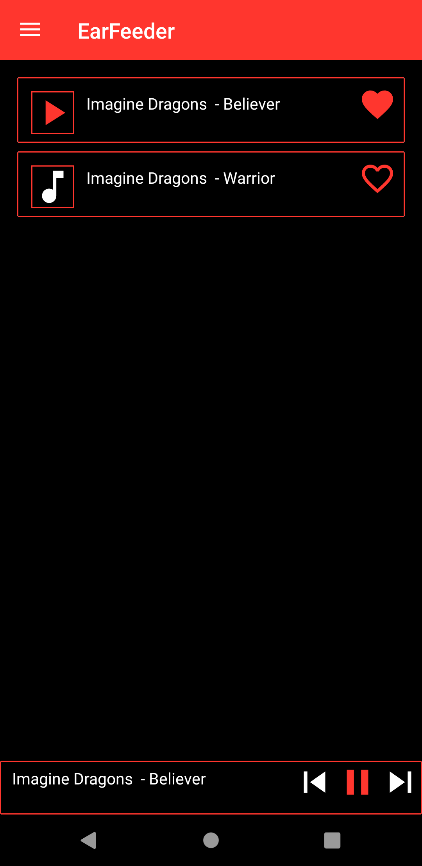
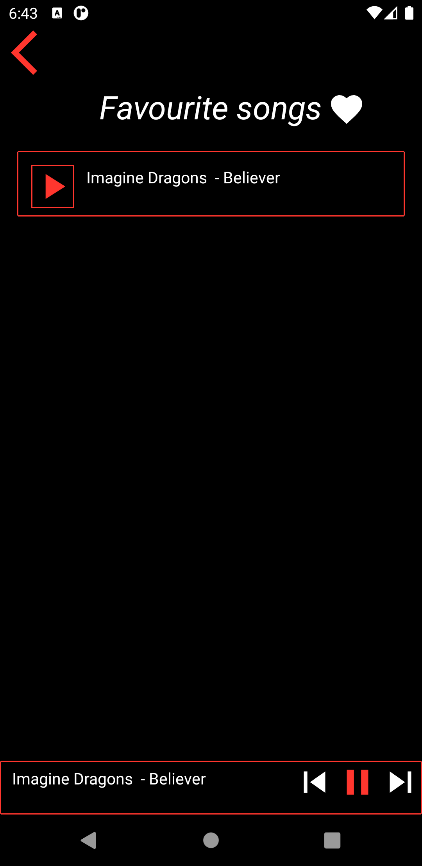


Рисунок 6 – результат теста 2

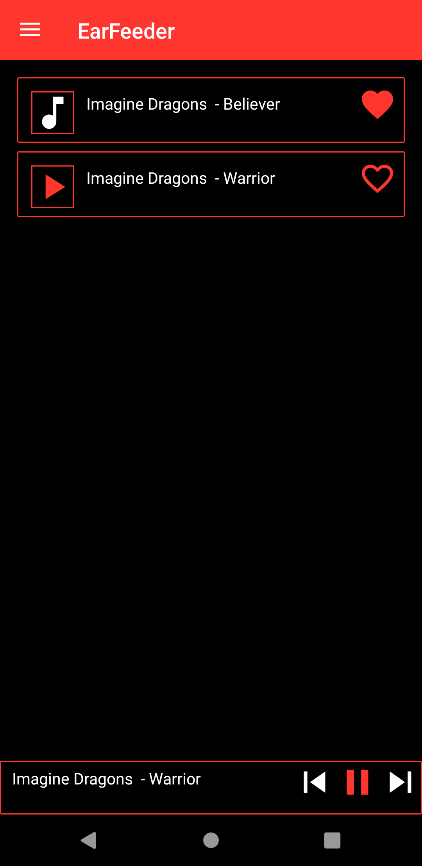
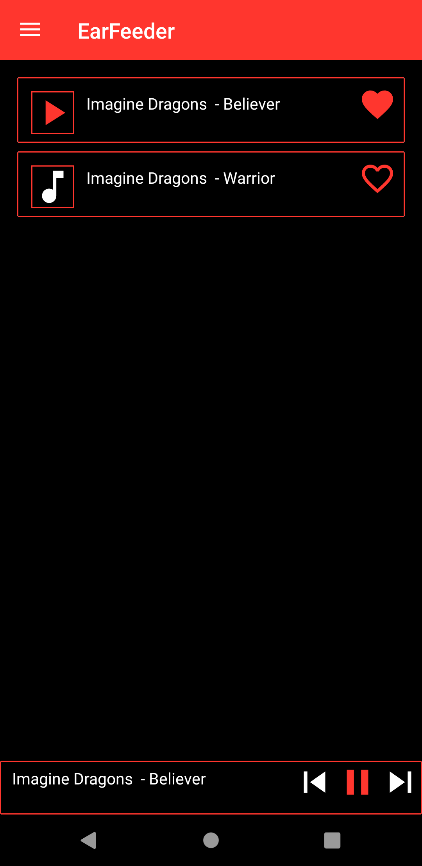


Рисунок 7 – а) результат теста 3; б) результат теста 5

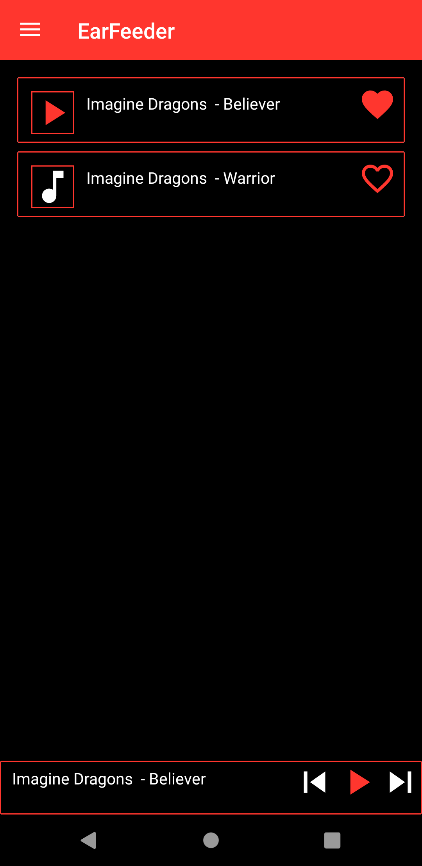
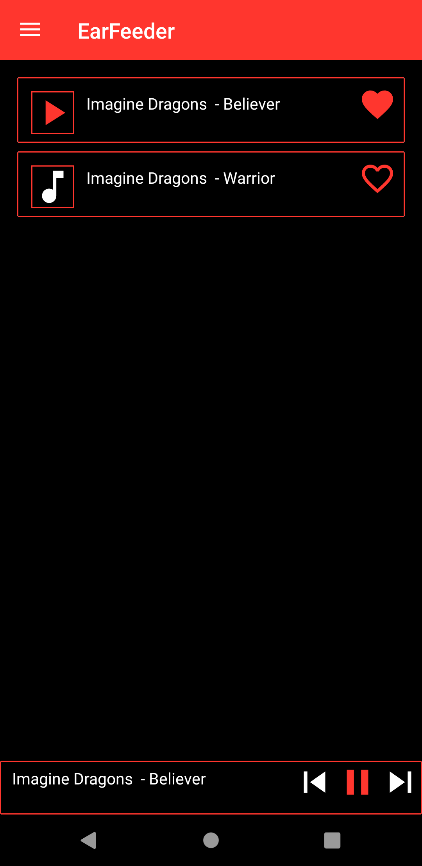


Рисунок 8– результат теста 7

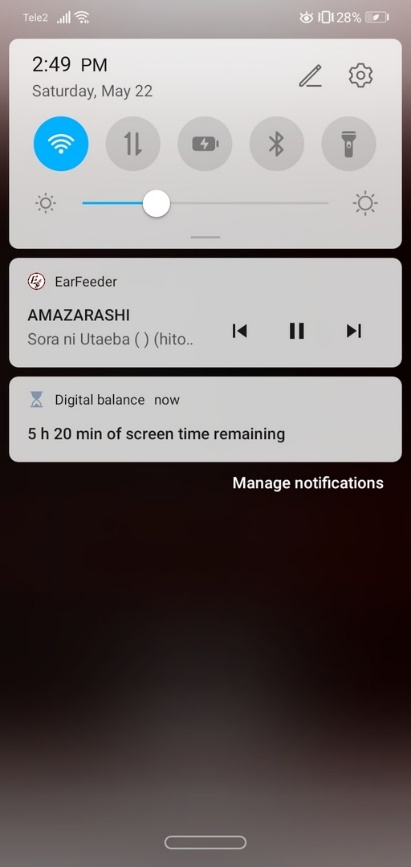
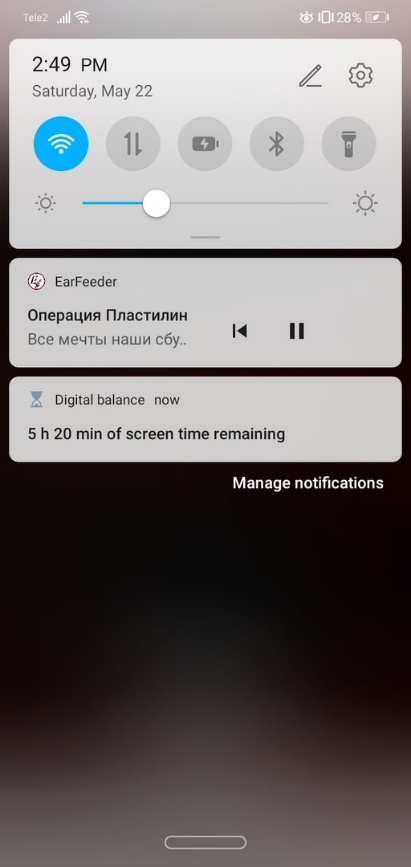


Рисунок 9 – а) результат теста 9; б) результат теста 10

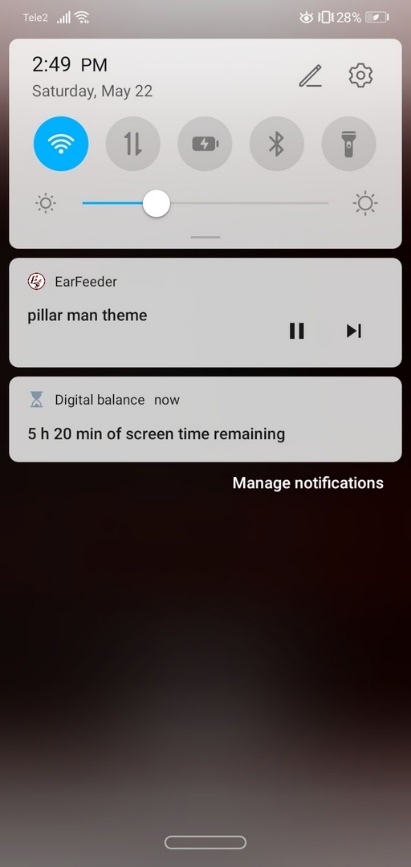
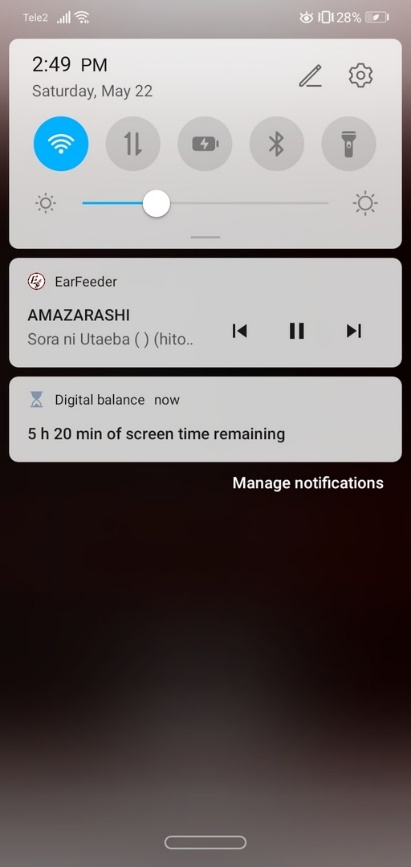


Рисунок 10 – результат тестов 11 и 12

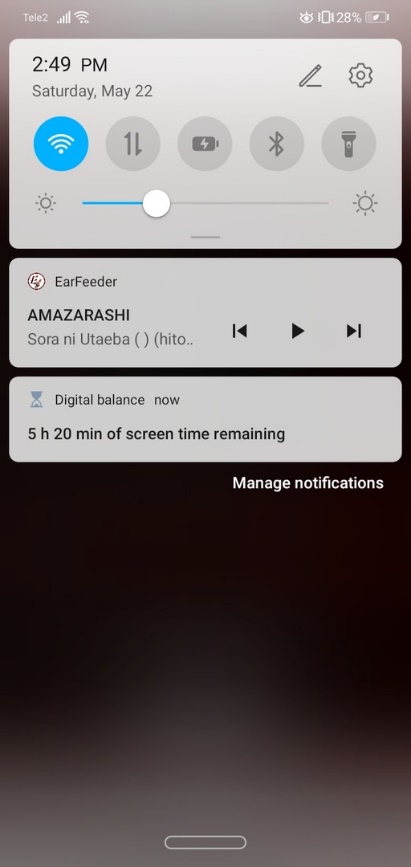
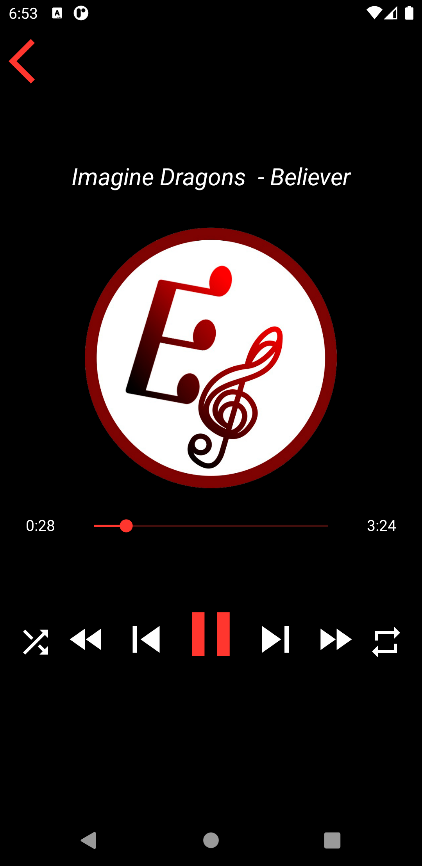


Рисунок 11 – а) результат теста 13; б) результат теста 15

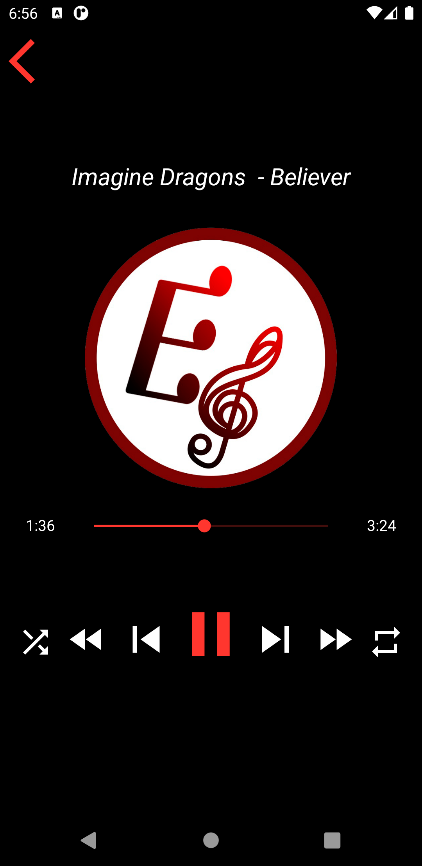


Рисунок 12 – а) результат теста 17; б) результат тестов 18 и 19

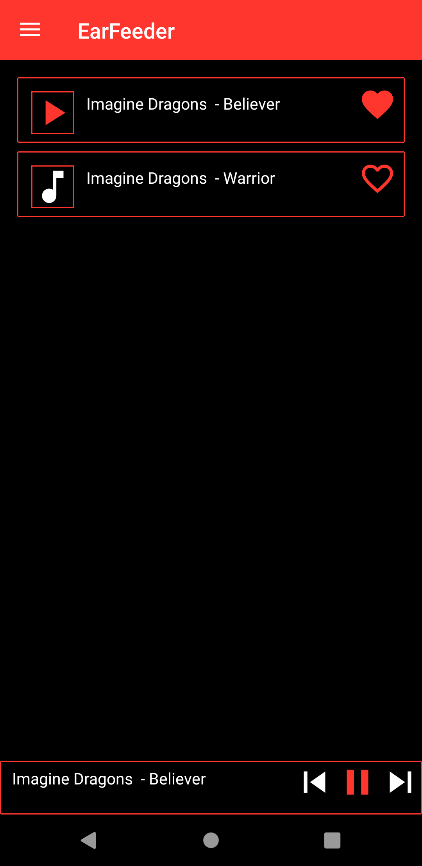
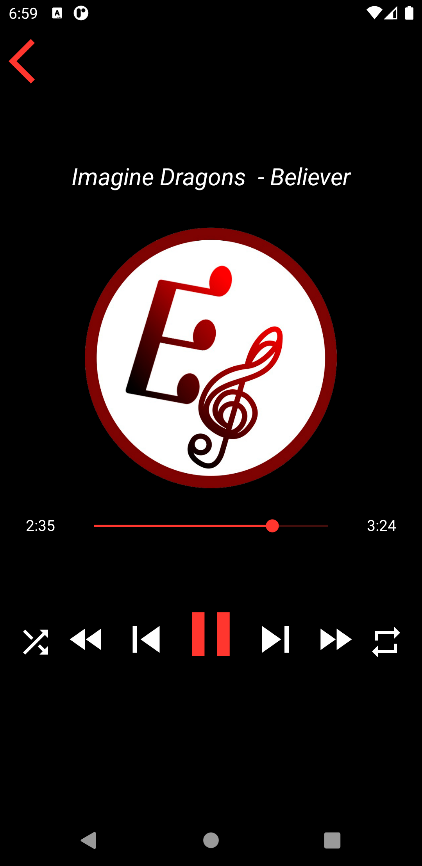


Рисунок 13 – результат теста 20



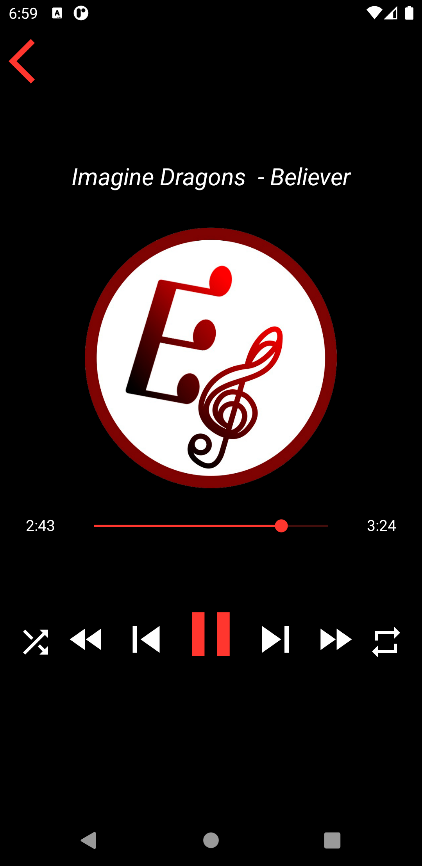


Рисунок 14 – результаты теста 21

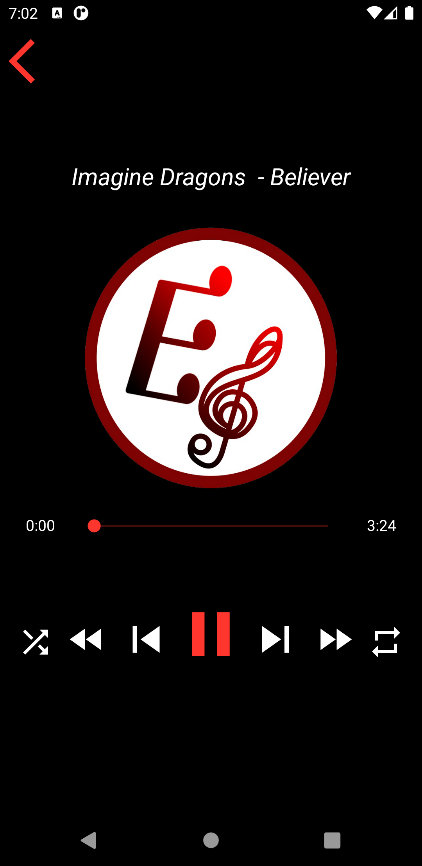
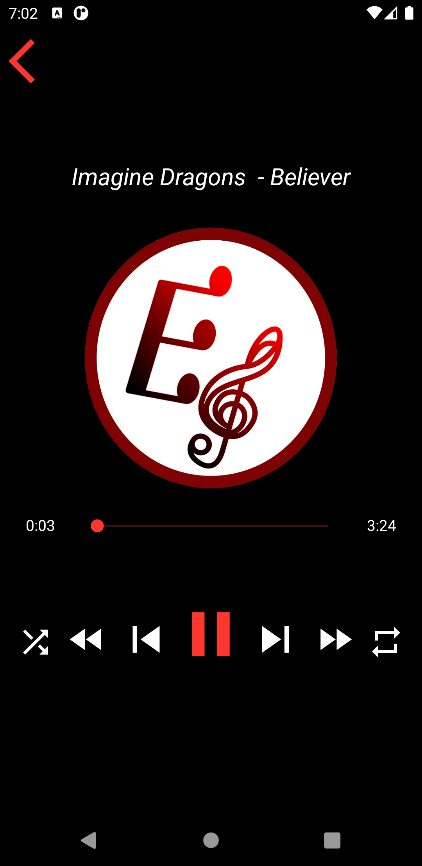


Рисунок 15 – результаты теста 22

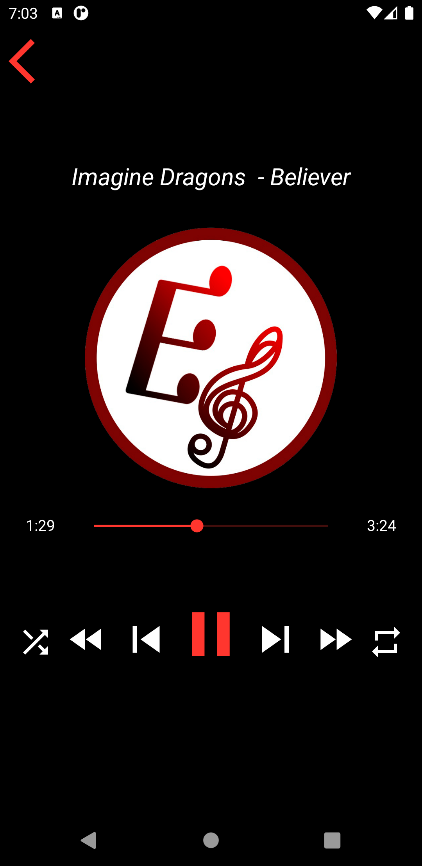
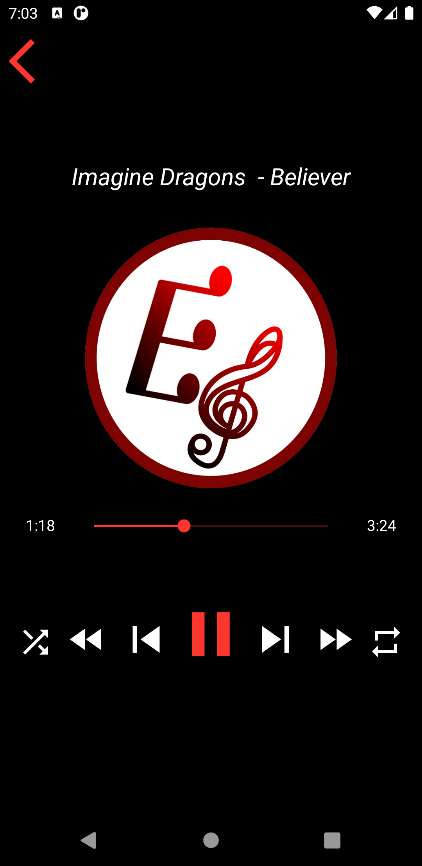


Рисунок 16 – результаты теста 23

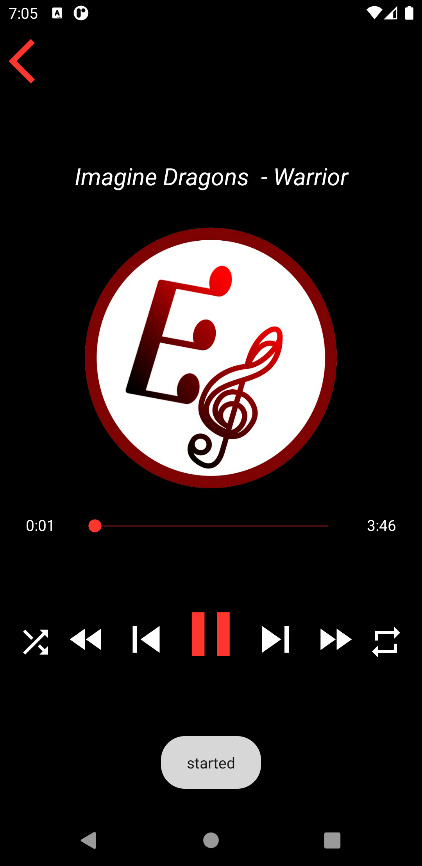
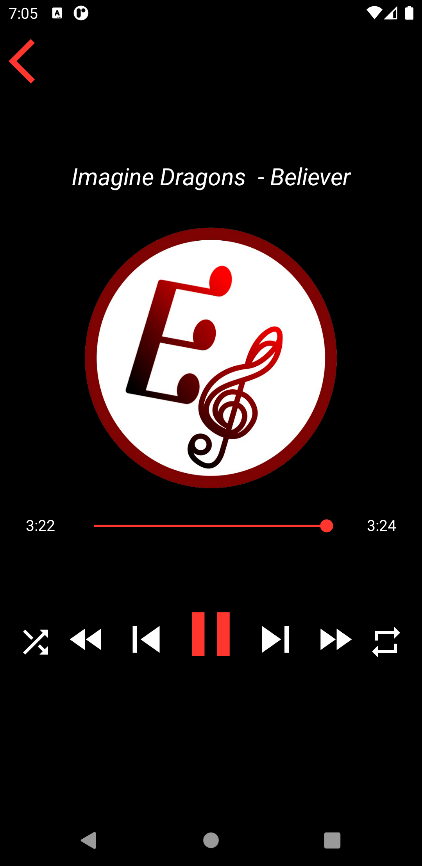


Рисунок 17 – результаты теста 24

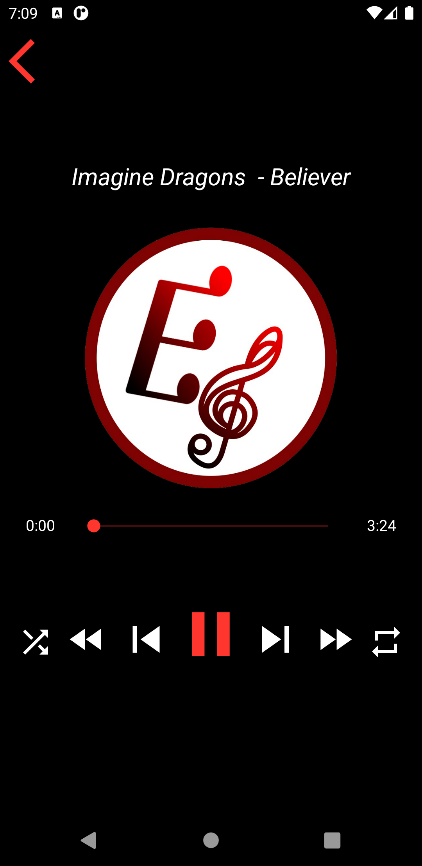
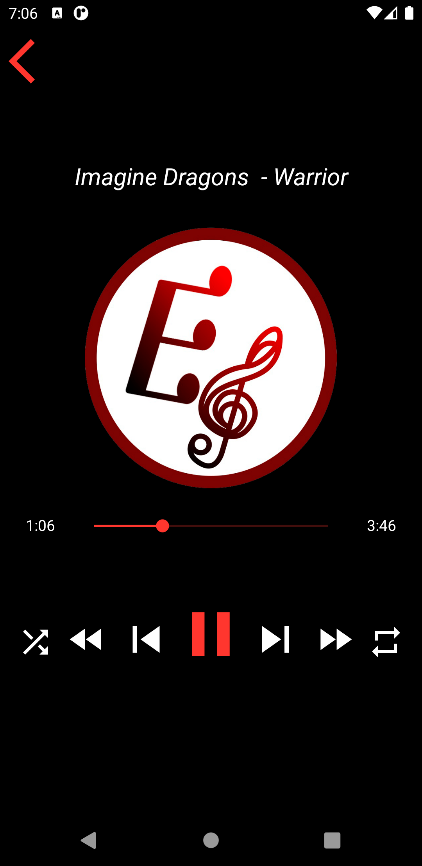


Рисунок 18 – результаты теста 25

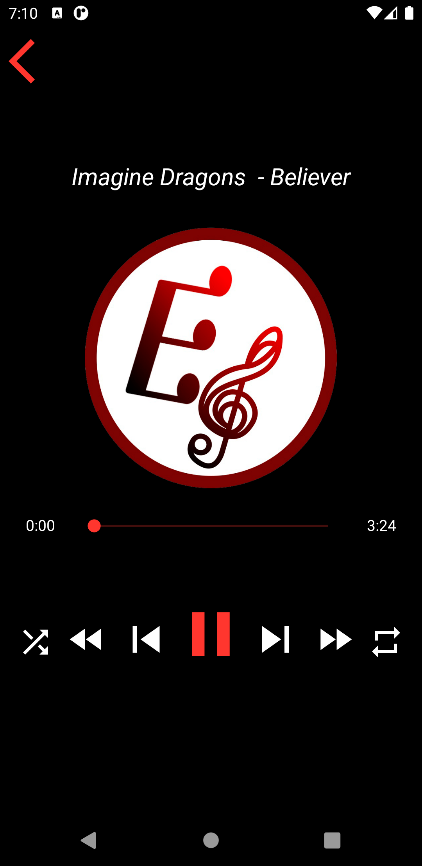
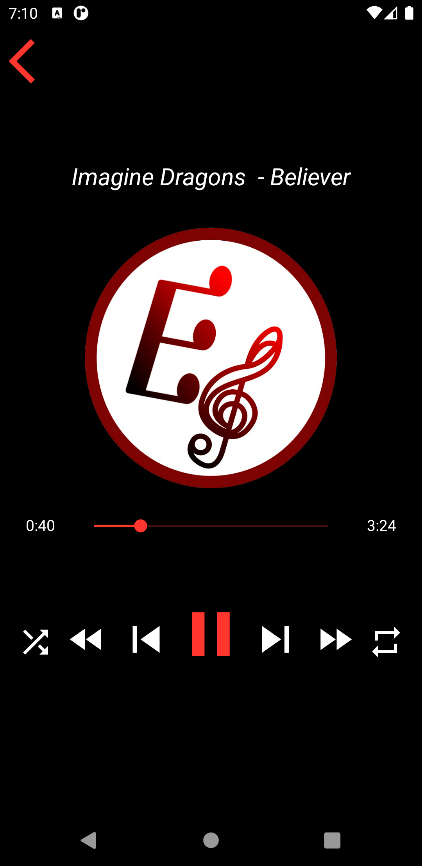


Рисунок 19 – результаты теста 26

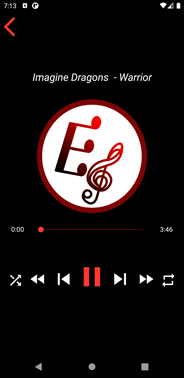
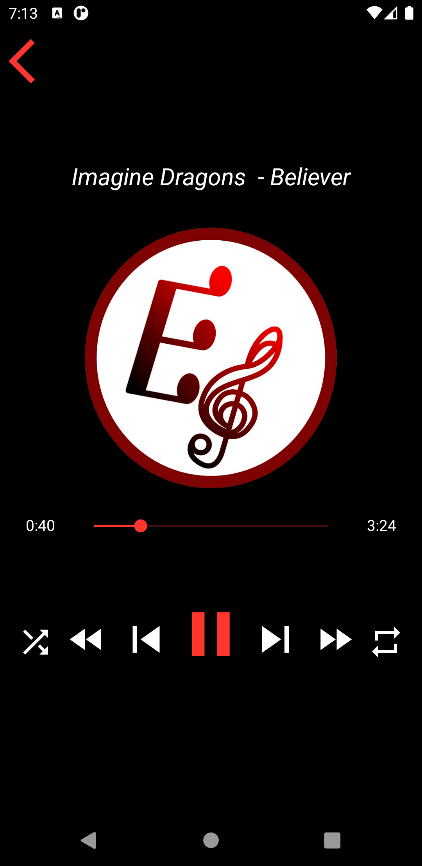


Рисунок 20 – результаты теста 27

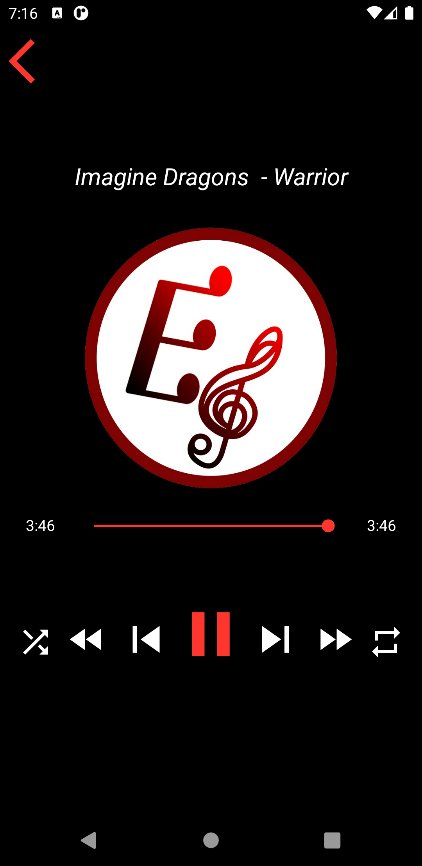
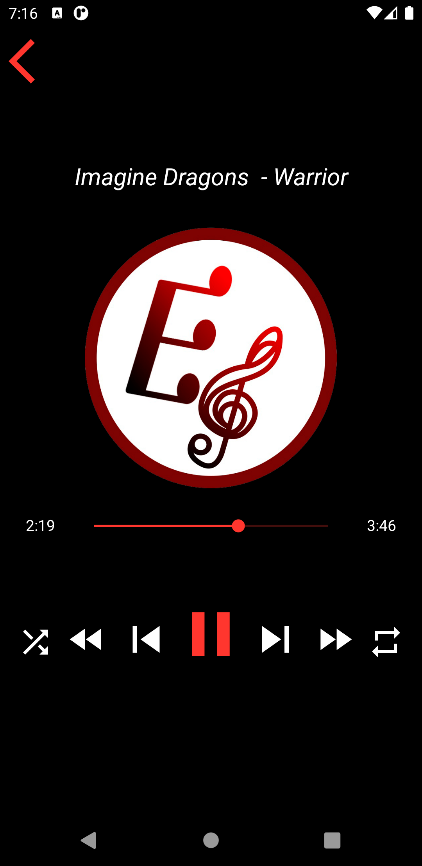


Рисунок 21 – результаты теста 28

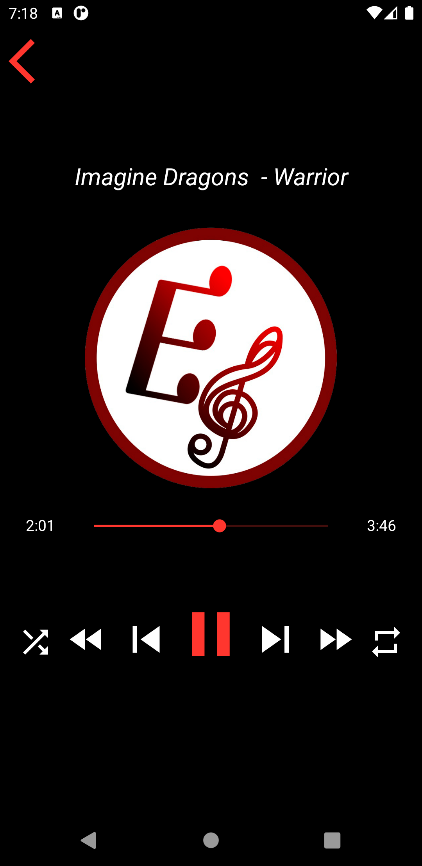
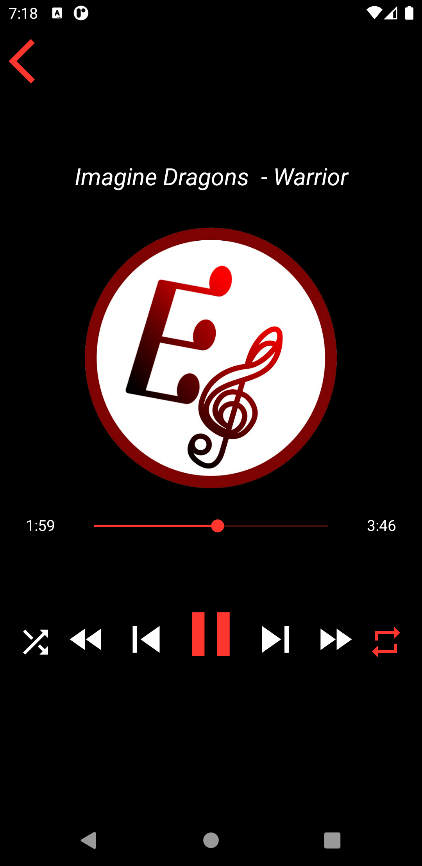


Рисунок 22 – результаты теста 29

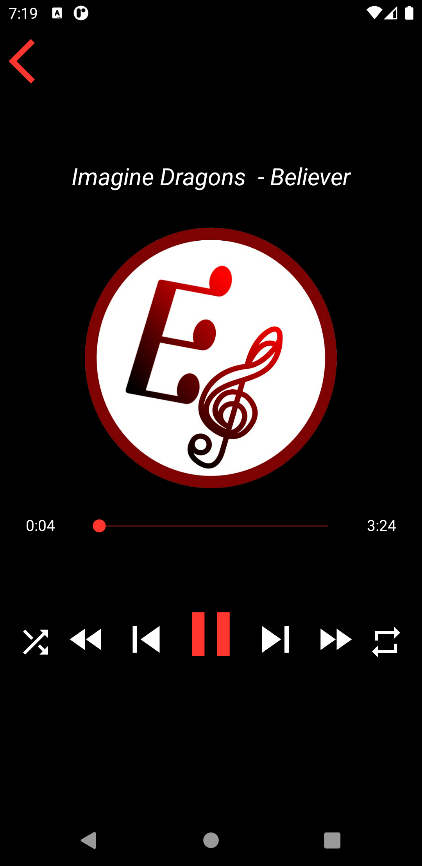
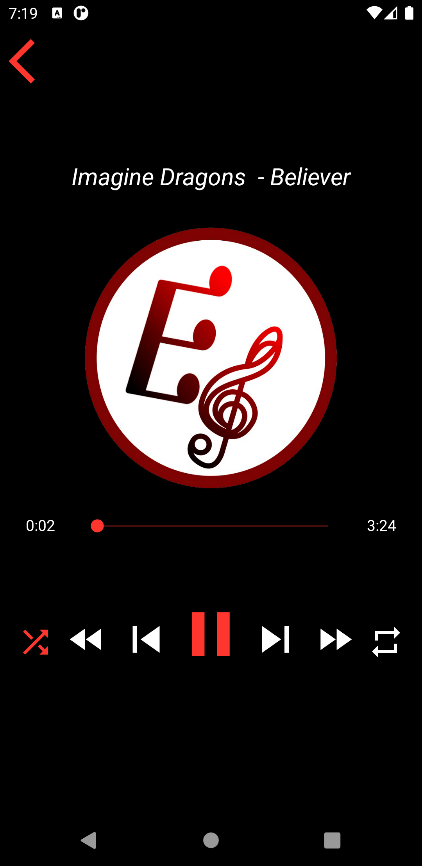


Рисунок 23 – результаты теста 30

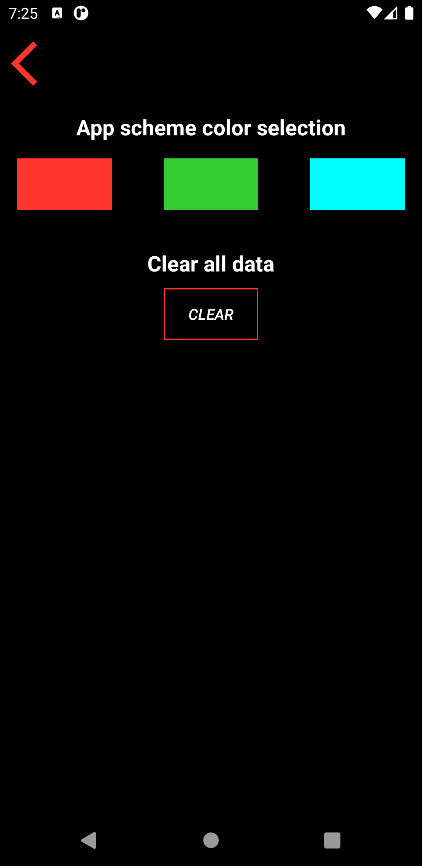
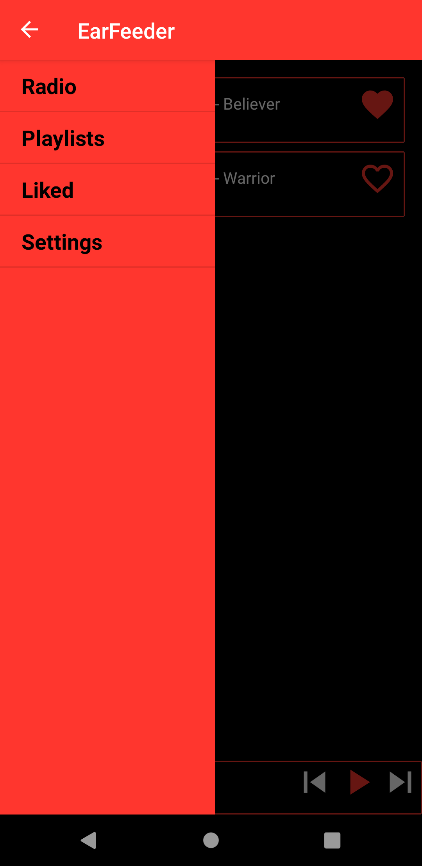


Рисунок 24 – результаты теста 31

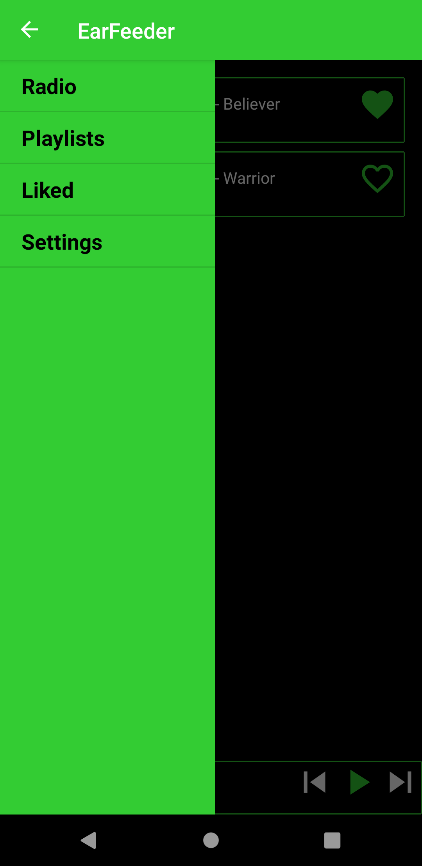
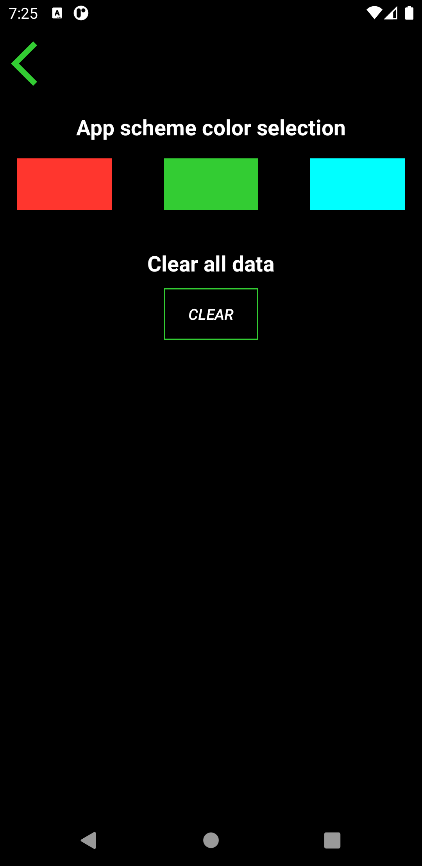


Рисунок 25 – результаты теста 31

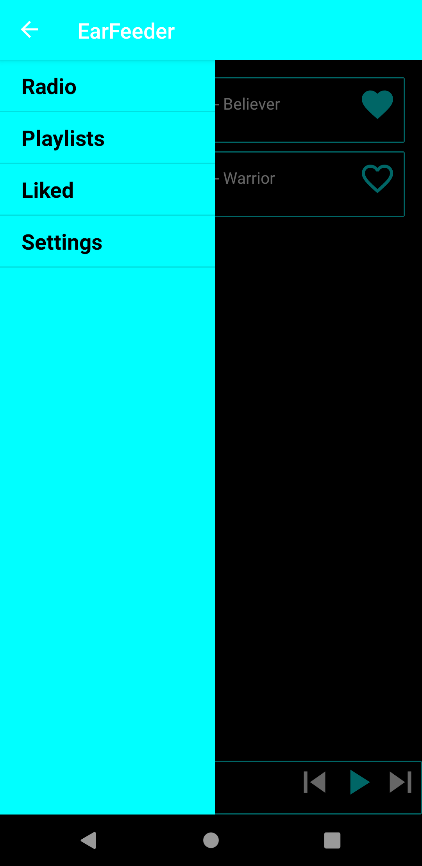
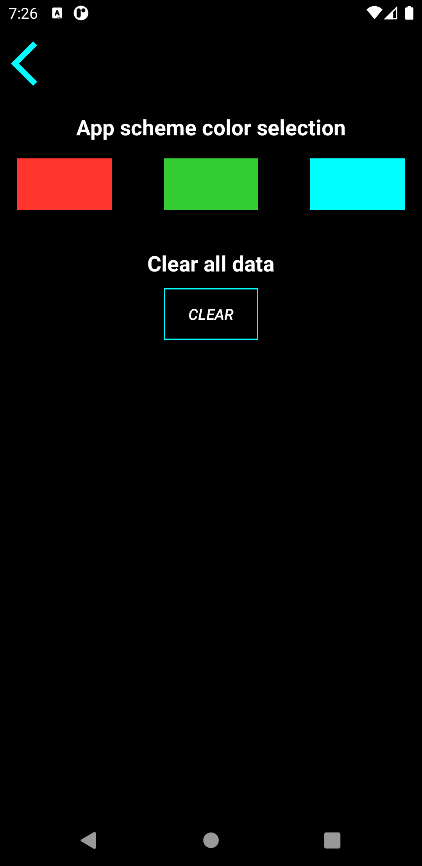


Рисунок 26 – результаты теста 31

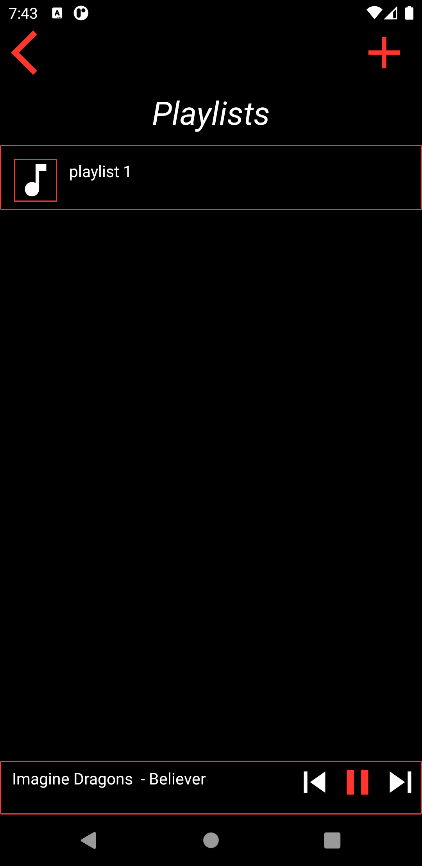
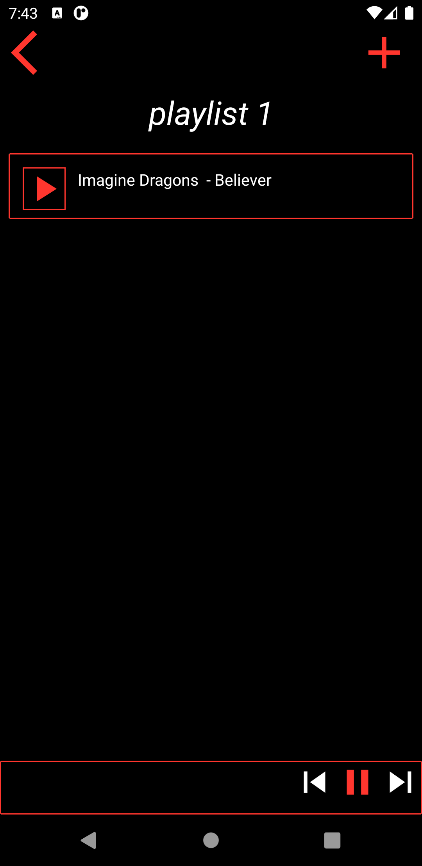


Рисунок 27 – результаты теста 36

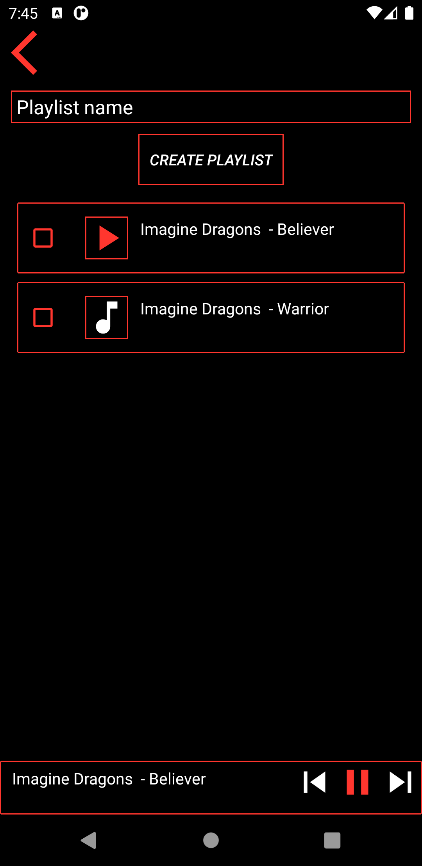


Рисунок 28 – результат теста 37

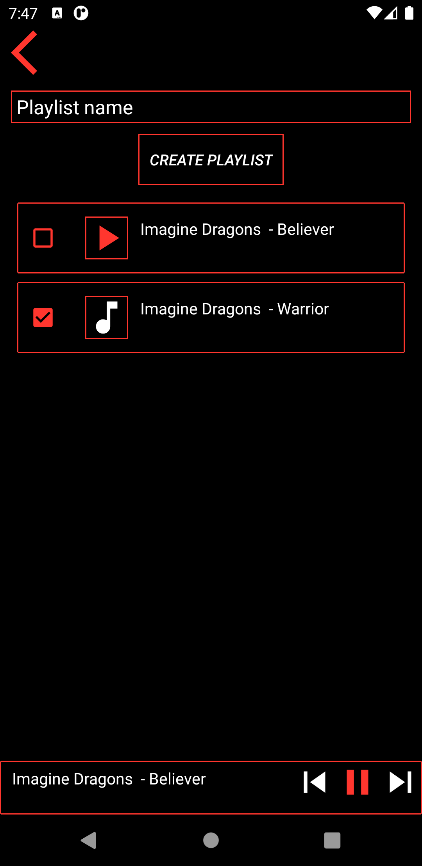


Рисунок 29 – результат теста 40

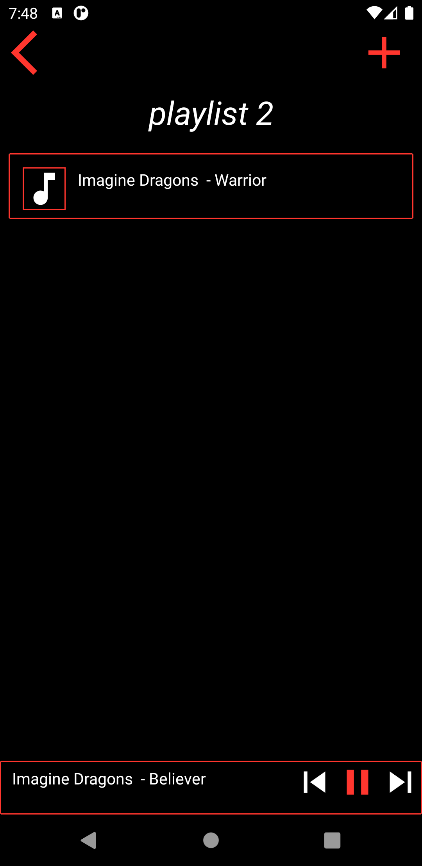
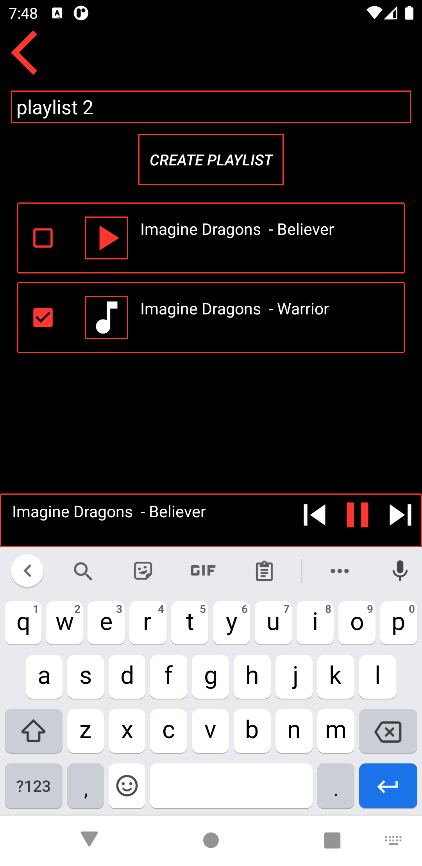


Рисунок 30 – результат теста 41

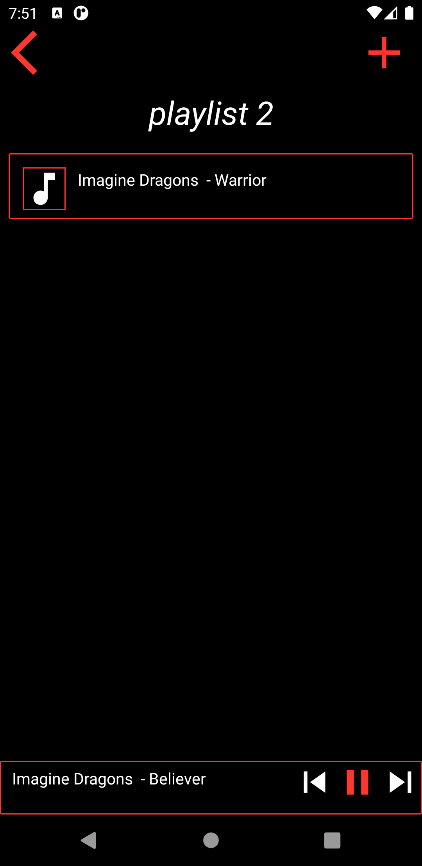
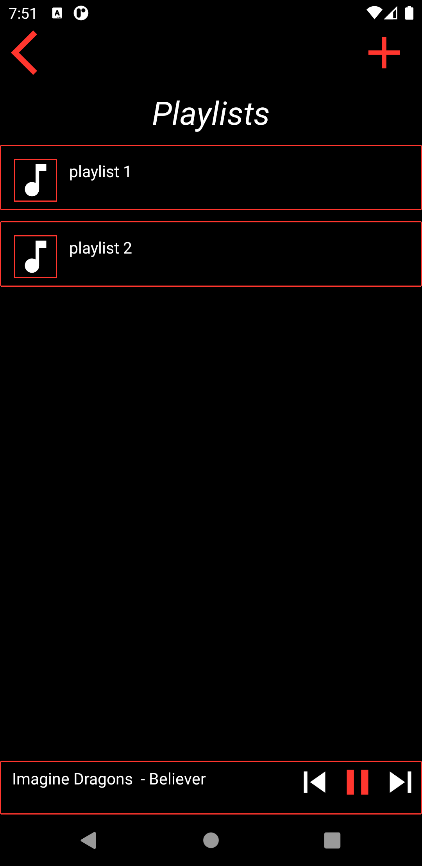


Рисунок 31 – результат теста 44

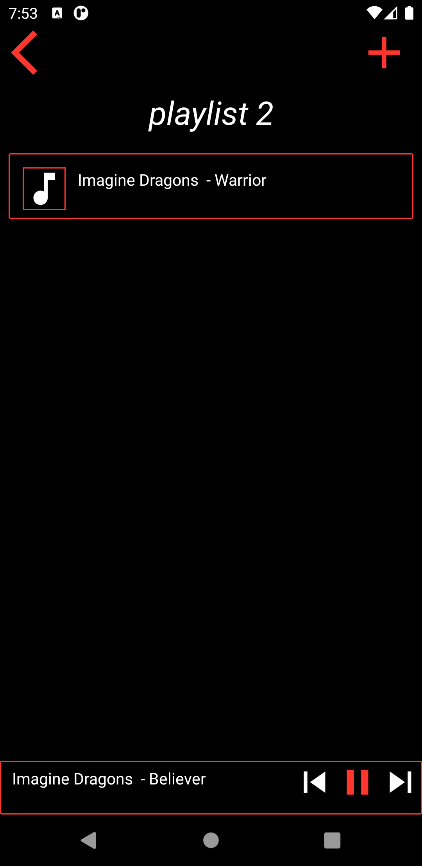
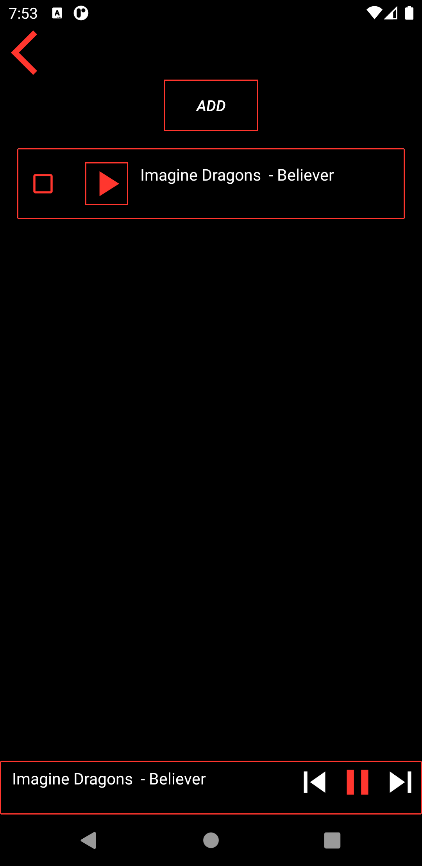


Рисунок 32– результат тестов 45 и 46

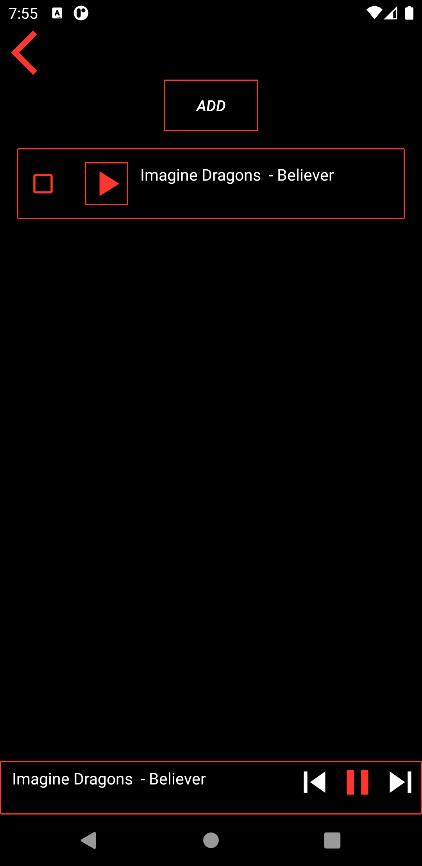
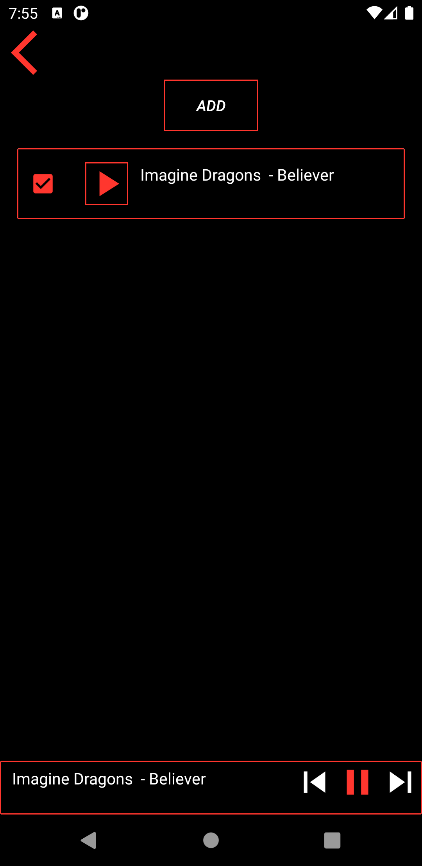


Рисунок 33 – результаты теста 48

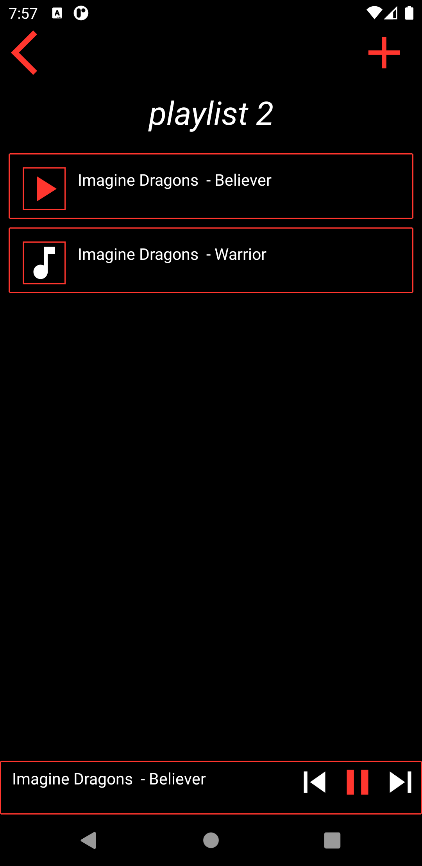
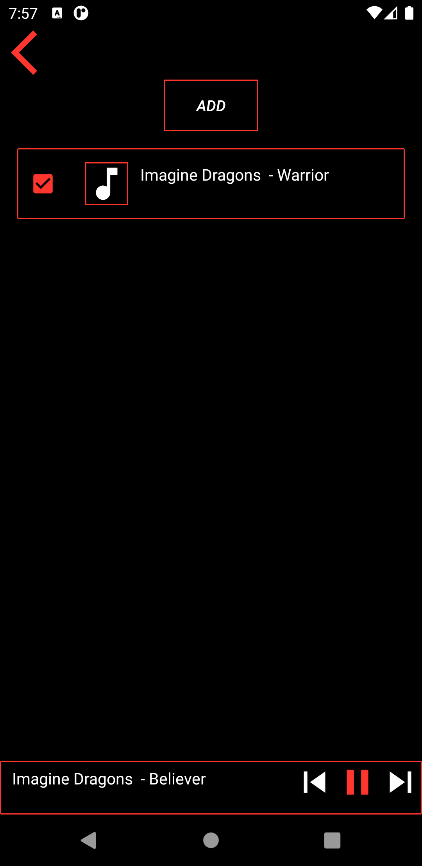


Рисунок 34 – результат теста 49

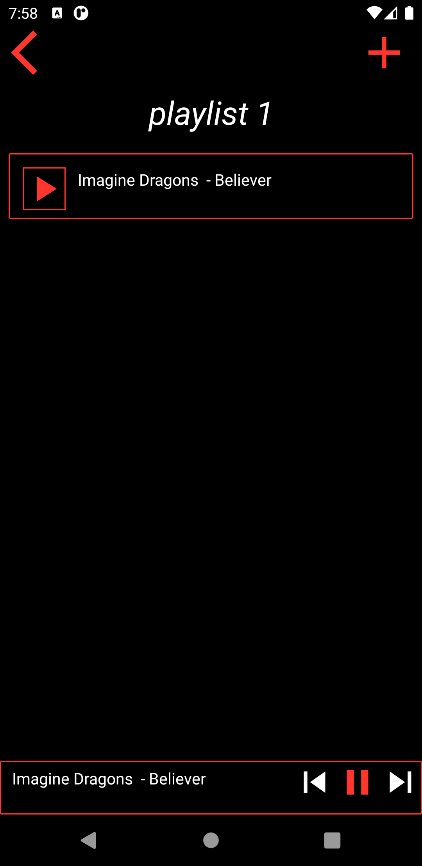
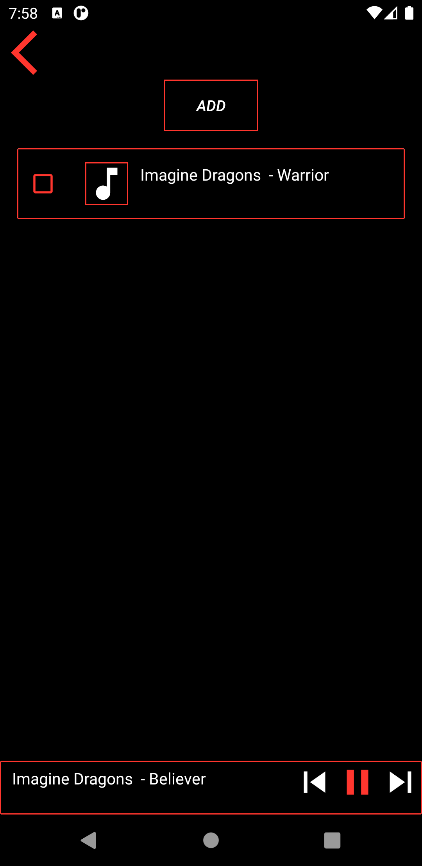


Рисунок 35 – результат теста 50

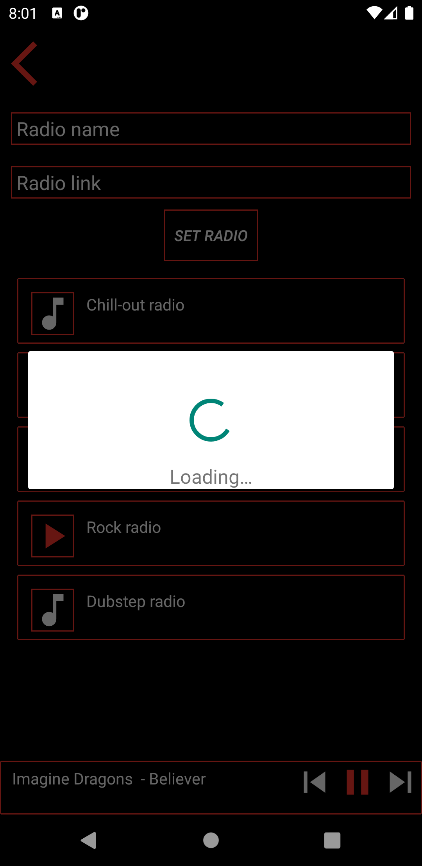
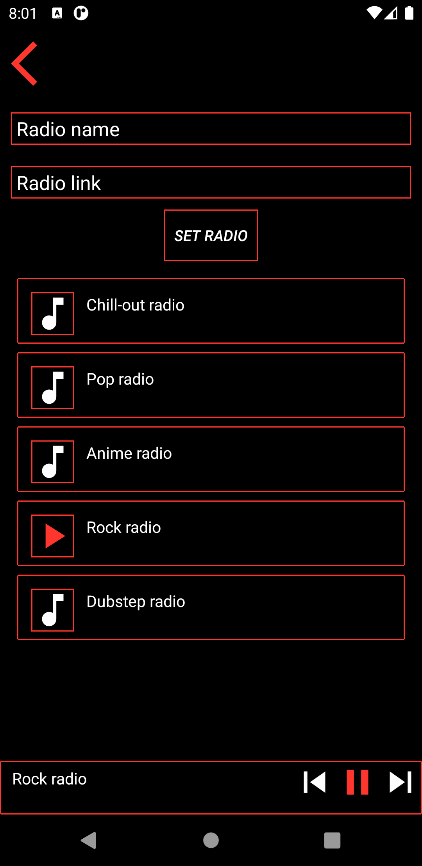


Рисунок 36 – результат теста 51

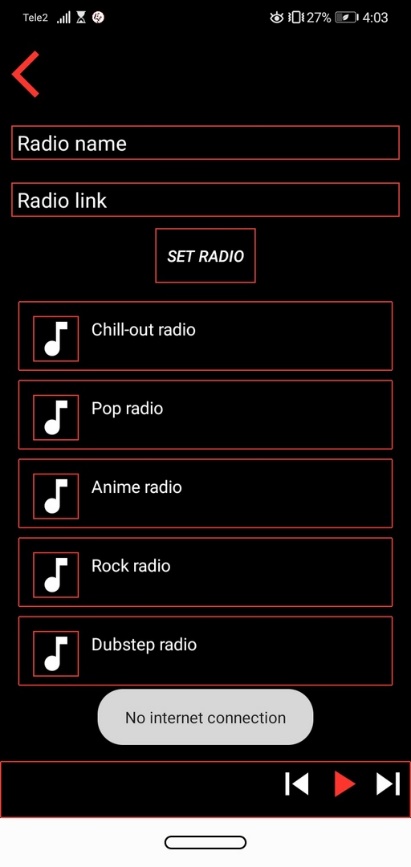


Рисунок 37 – результат теста 52

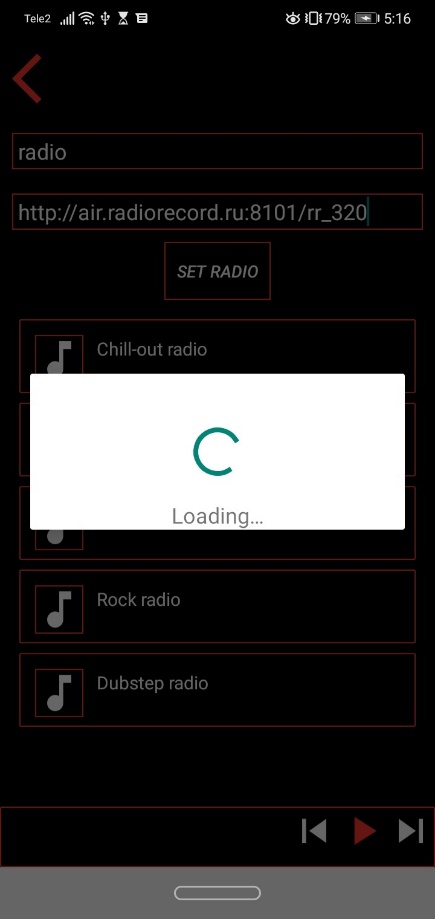
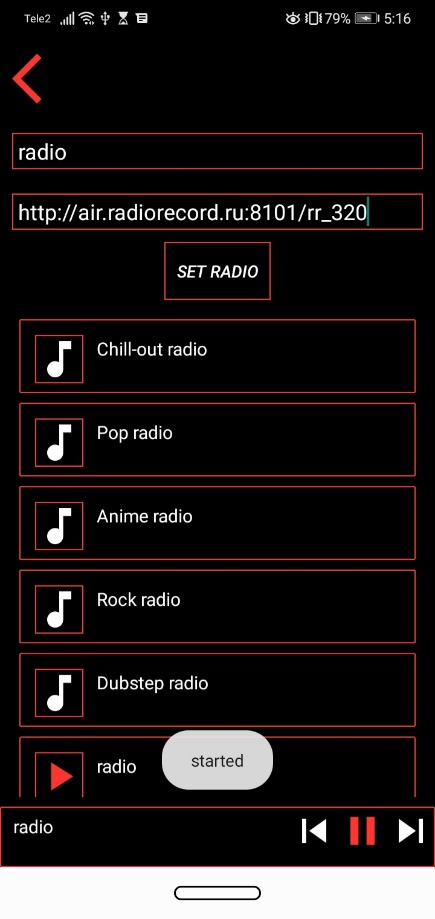


Рисунок 38– результат теста 54

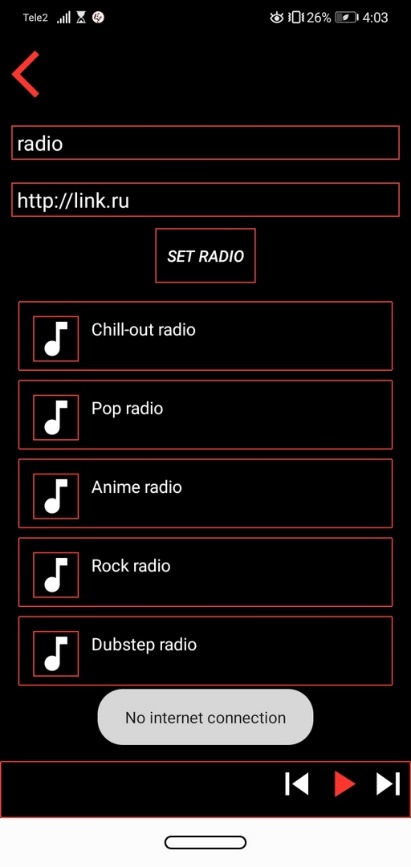


Рисунок 39 – результат теста 55

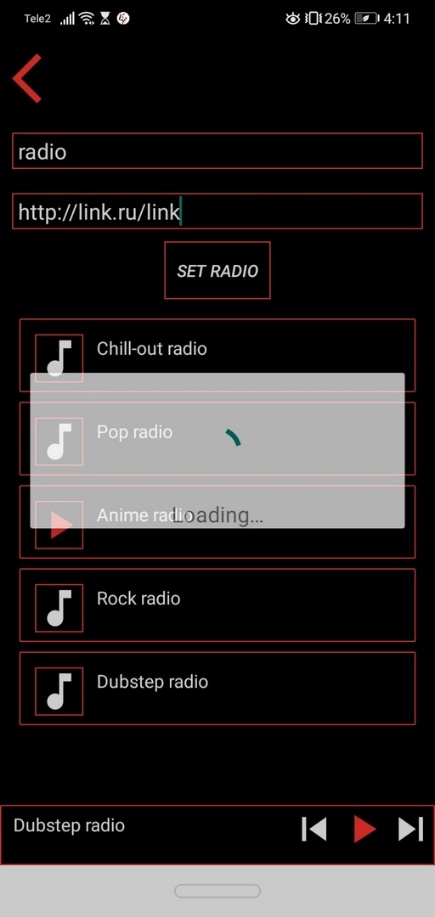
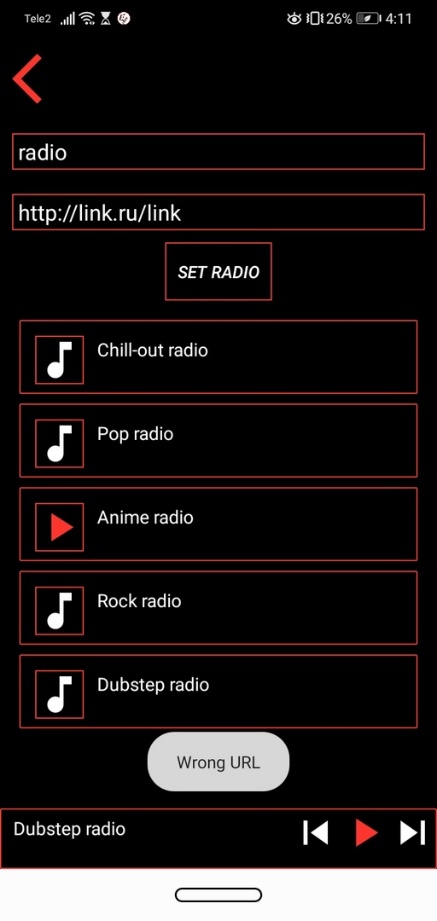


Рисунок 40 – результат теста 56

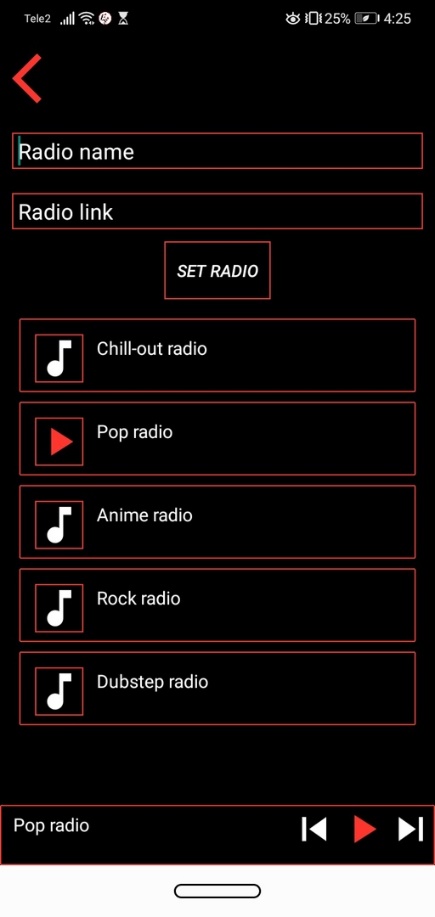
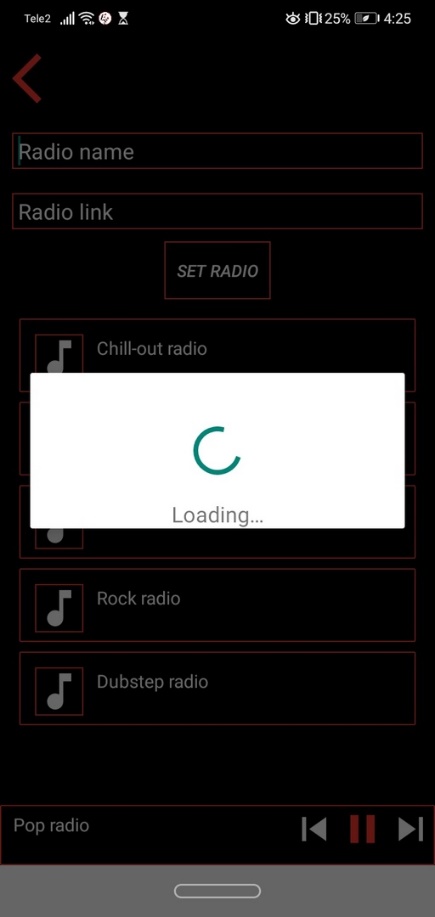
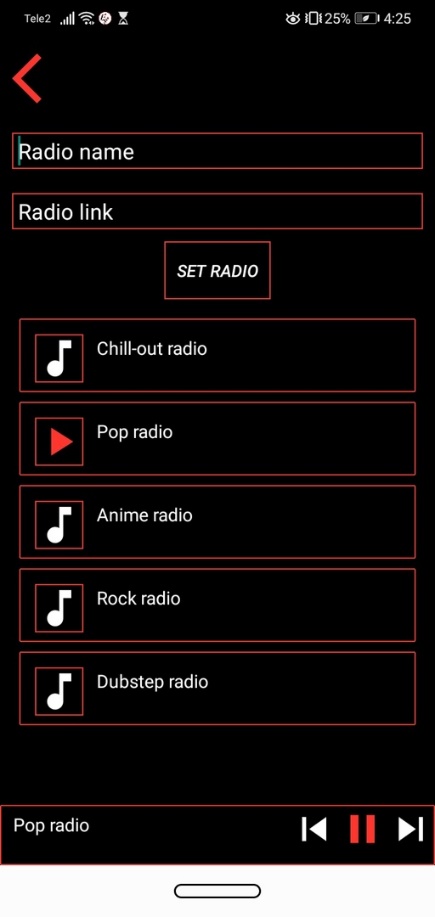
  

Рисунок 41– результат теста 57

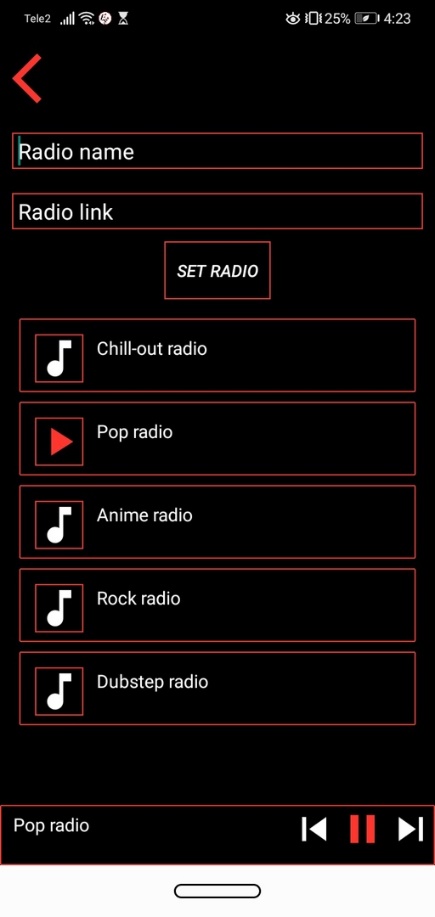
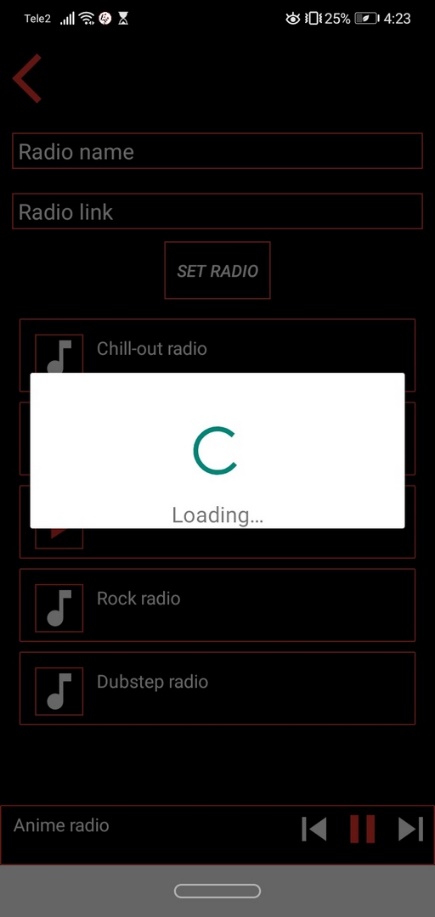
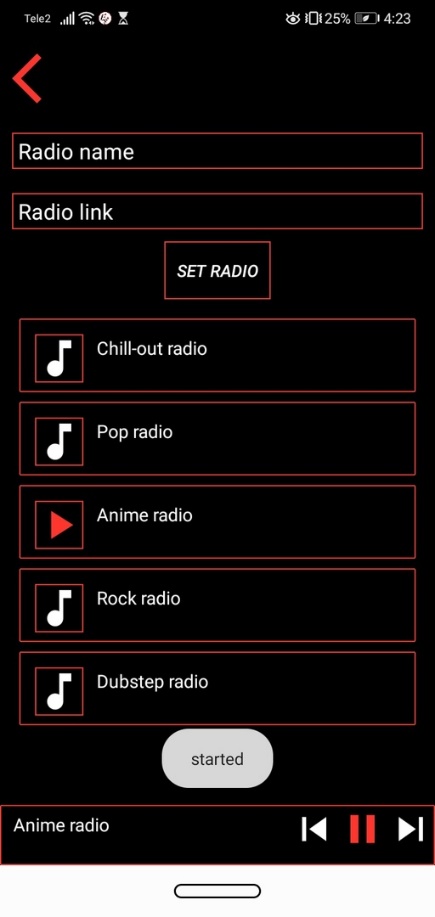
  

Рисунок 42– результат теста 58

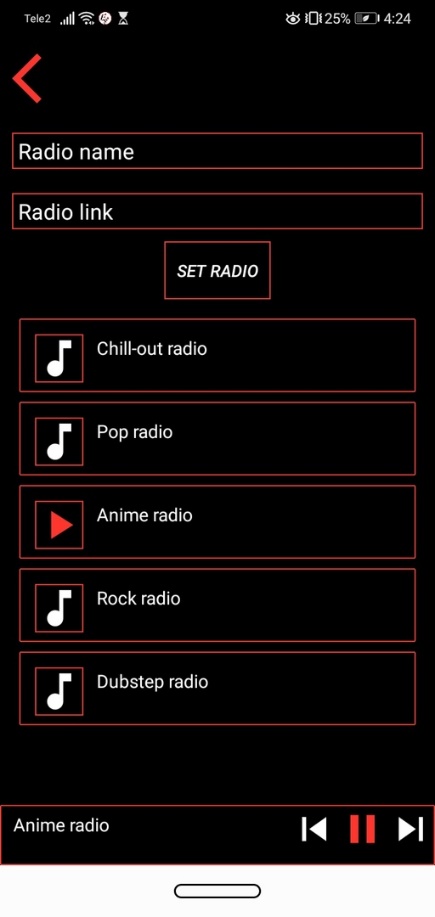
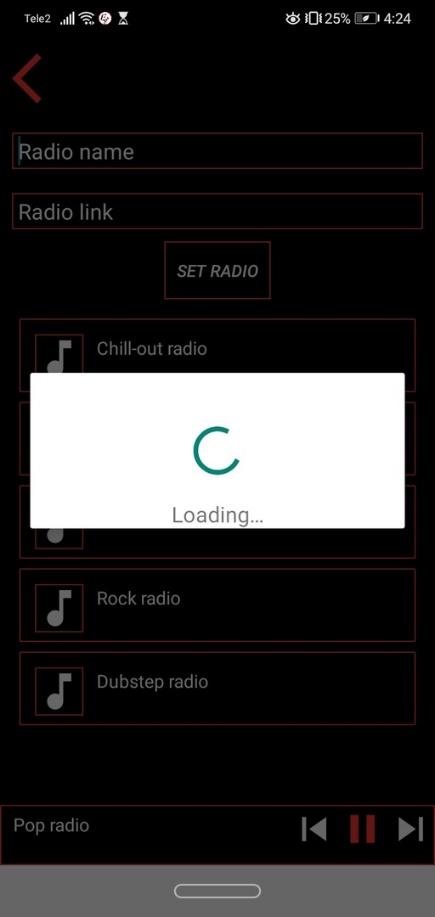
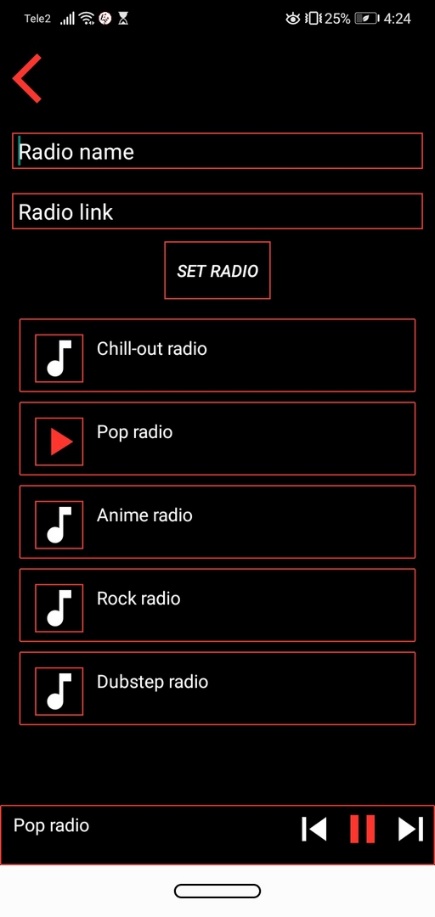
  

Рисунок 43 – результат теста 60

# Заключение

На практике разработки данного приложения развились навыки:

1. Проектирования и применения объектного подхода в реализации системы
2. Проектирования пользовательского интерфейса и компоновке элементов взаимодействия с пользователем
3. Создания и проектирования локальных баз данных с помощью СУБД SQLite
4. Программирования под среду Android (в том числе: работа с внутренними файлами телефона, потоки, сервисы, фрагменты, уведомления и http-запросы)
5. Изучения и применения необходимых видов паттернов программирования (singleton)

Как результат работы:

1. Выработаны функциональные требования приложения
2. Спроектированы интерфейс, классы и локальная база данных
3. Разработано приложение согласно реализованным требованиям и проектированию
4. Приложение было протестировано с помощью приемочного тестирования
5. Составлена документация по всей проделанной работе

Проект находится в общем доступе на сервисе для хостинга GitHub: <https://github.com/MAL1S/MusicPlayer.git>

Цель курсового проекта достигнута.

Список использованных источников

1. Head first. Паттерны программирования. Эрик Фримен, Элизабет Робсон [2020] (дата обращения: 15.04.2021)
2. Сборник ссылок для радио [электронный ресурс] // URL: <https://mc-terra.ru/index.php?topic=7771.0> (дата обращения: 12.04.2021)
3. Start Android [электронный ресурс] // учебник по андроид для начинающих и продвинутых, URL: <https://startandroid.ru/ru/> (дата обращения: 04.04.2021)
4. Java. Экспресс-курс [электронный ресурс] // Сайт Александра Климова [сайт], URL: http :// developer.alexanderklimov . ru /android / java / java. php (дата обращения: 04.04.2021)
5. API Specification for the Java 7 SE. [официальный сайт] URL: http :// docs. oracle. com / javase /7/ docs /api / (дата обращения: 05.04.2021)
6. Отношения классов — от UML к коду [Электронный ресурс] // Хабрахабр: [сайт]. [2006]. URL: http://habrahabr.ru/post/150041/ (дата обращения: 20.05.2021)
7. Что такое ORM [Электронный ресурс]. [2017]. URL: <https://myrusakov.ru/what-is-it-orm-php.html> (дата обращения: 27.05.2021)
8. Data Access Object (DAO) [Электронный ресурс] // Хабрахабр: [сайт]. [2015]. URL: <https://habr.com/ru/post/262243/> (дата обращения: 27.05.2021)