# 2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА

## 2.1 Проектирование архитектуры и описание состояний

**программного средства**

Большая часть приложения, которые используются для аудиоплееров, состоят из множества компонентов. Структура окон приложения описывается в отдельных файлах, отвечающих за графическую часть приложения. Такие файлы создаются с помощью специальных языков программирования, например, *XML*. Это позволяет создавать графические интерфейсы, адаптируемые под различные устройства.

Среднестатистический пользователь часто взаимодействует с графическим интерфейсов программы. Принимая это во внимание, нужно построить для приложения со сложной структурой адаптивный интерфейс, который сможет подстраиваться под запросы конечного пользователя.

Важно создать архитектуру, которая позволит масштабировать приложение, сделает его надёжнее и повысит его производительность для слабых устройств. В связи с этим, выделю следующие важные аспекты архитектуры приложения:

– разделение задач и потребностей приложения;

– построение адаптивного *UI*;

– построение архитектуры приложения, которая будет мало зависеть от конкретного устройства.

Приняв во внимание все факты, изложенные выше, было принято решение использовать парадигму объектно-ориентированного программирования (ООП), что позволит сделать код более структурированным и обеспечить более легкую поддержку кода в будущем для сторонних разработчиков, если таковые появятся. Для разработки архитектуры приложения принято решение использовать архитектурный паттерн проектирования *Model-View-Controller* (*MVC*), который позволит разделить разрабатываемое приложение на слои представлений, которые будут отвечать за графическое отображение интерфейса, слой контроллера, который будет принимать на себя роль управления информацией от модели к представлению и наоборот, и модели, которая будет проводить основную обработку информации приложения. Диаграмма, на которой отображен принцип работы *MVC*, представлена на рисунке 2.1.

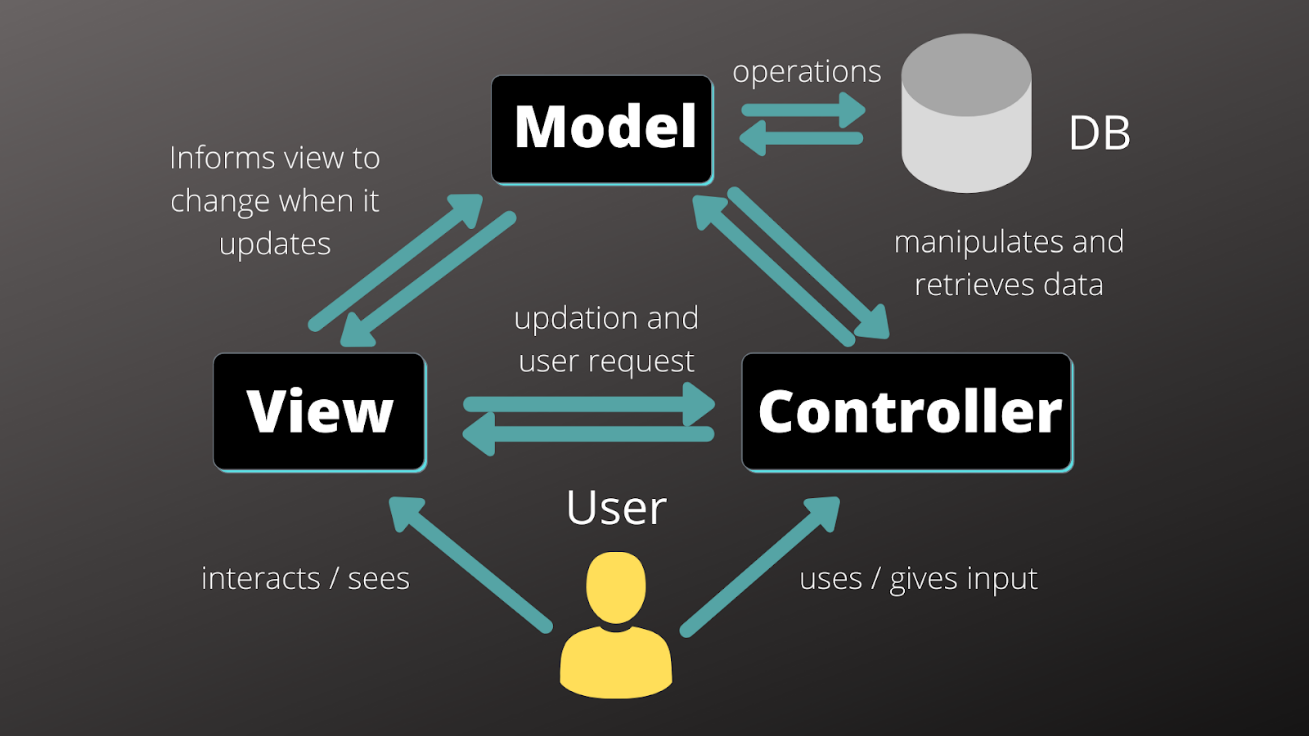


Рисунок 2.1 – Принцип работы *MVC*

Как отображено на диаграмме, шаблон *MVC* состоит из следующих трёх слоёв: представление (*View*), модель (*Model*), контроллер (*Controller*).

Модель хранит в себе данные, позволяет делать запросы в базу данных, позволяет обрабатывать эти данные. Тут находится основная логика нашей программы. Модель не знает, что происходит в контроллере и для чего оно обрабатывает этот запрос, который поступил к ней. Контроллер принимает на себя роль посредника между представление и моделью. Контроллер, можно сказать, реагирует на действия пользователя с графическим интерфейсом, отправляя нужный запрос в модель. Представление отвечает за отображение информации (визуализацию). Одни и те же данные могут представляться различными способами и в различных форматах. Контроллер направляет данные в представление, которое в свою очередь преобразует их в нужный нам графический формат [3].

Задача применения паттерна *MVC* состоит в разделении данных и логики программы от визуализации. За счет разделения архитектуры приложения на слои мы добиваемся более высокой надежности, так как компоненты не будут сильно зависеть от друг друга, получаем возможность редактировать компоненты архитектуры без сильного влияния на остальную часть приложения, а также упрощаем поддержку и тестирование приложения.

Диаграмма состояний – это диаграмма состояний из теории автоматов со стандартизированными условными обозначениями, которая может определять множество систем от компьютерных программ добизнес-процессов [5]. Используются следующие условные обозначения:

– круг, обозначающий начальное состояние

– окружность с маленьким кругом внутри, обозначающая конечное состояние;

– скругленный прямоугольник - обозначает состояние;

– стрелка, обозначающая переход. Название события (если есть), вызывающего переход, отмечается рядом со стрелкой

– толстая горизонтальная линия с либо множеством входящих линий и одной выходящей, либо одной входящей линией и множеством выходящих, что означает объединение и разветвление соответственно.

Опишем состояния программного средства «*lw.sound*» при помощи нотации *UML*. На диаграмме состояний программного средства отображены следующие ключевые состояния: отображение экрана авторизации, отображение главного экрана, отображение окна лайкнувших треков, отображение окна скачанных треков.

Рассмотрим все приведенные состояния последовательно в соответствии с их положением на диаграмме. Начало диаграммы отображается в виде закрашенного круга. Первым состоянием, с которым сталкивается пользователь при запуске приложения, является от отображение главного экрана.

Главный экран предлагает пользователю следующие опции:

– выбрать стандартную папку из которой будут браться треки для проигрывания;

– добавить в стандартную папку новый трек;

– переход на вкладку с лайкнувшими треками (аналогично началу процессу авторизации);

– переход на вкладку со скачанными треками;

– взаимодействие с композицией.

Изначально главный экран открывается с на вкладку с скачанными треками, что позволяет использовать приложение даже без подключения к серверу [6].

 Из данного экрана, в зависимости от выбора пользователя программным средством может быть проведен переход пользователя на другую вкладку, открытие проводника для добавления трека, открытие проводника для выбора стандартной папки, выход из текущего аккаунта или начало проигрывания композиции. Фрагмент диаграммы, на котором отображены данные состояния представлен на рисунке 2.2.

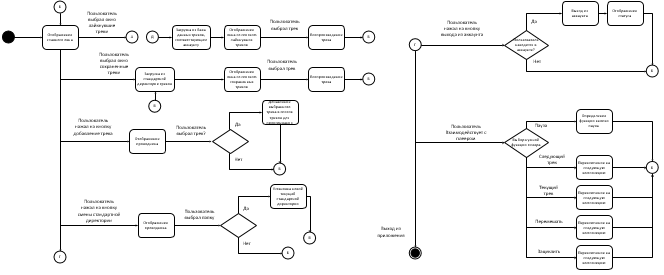


Рисунок 2.2 – Функции главного экрана, предоставляемые пользователю

При выборе опции выхода из аккаунта пользователем, будет запущен процесс по проверке текущего аккаунта. Если на данный момент пользователь вошел в аккаунт, то будет отображено окно с подтверждением выхода, иначе ничего не произойдет, ведь пользователь не вошел в аккаунт, а как следствие ни откуда выходить не нужно. Информация о выбранном аккаунте отображается в левой части экрана приложения. Фрагмент диаграммы, который соответствует данному состоянию приведен на рисунке 2.3.

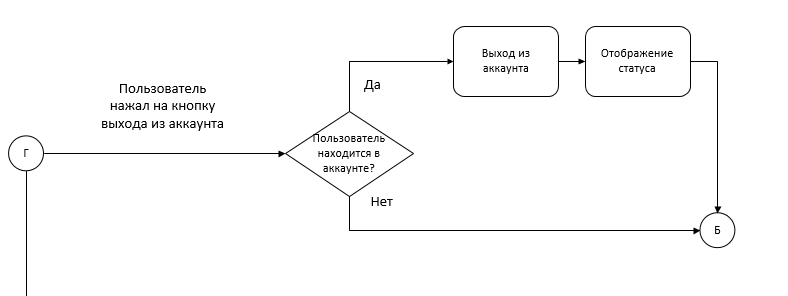


Рисунок 2.3 – Диаграмма состояний при нажатии на кнопку выхода из аккаунта

При нажатии на вкладку лайкнувшие треки произойдет перенаправление на новое окно с формой авторизации, если аккаунт не был выбран, или произойдет импортирование имен треков из базы данных (если есть соединение) в приложение. При нажатии на вкладку скачанные треки программа отобразит все треки, которые соответствуют формату «*.mp3*» и которые находятся в текущей стандартной директории. Фрагмент диаграммы, соответствующий данной части приведен на рисунке 2.4.

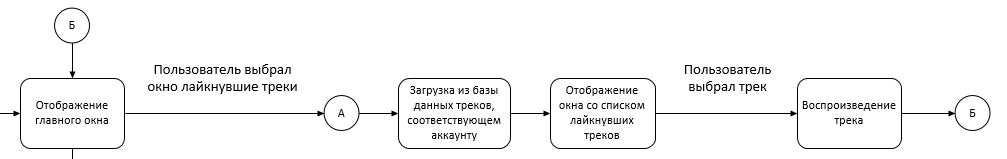


Рисунок 2.4 – Диаграмма состояний при нажатии на кнопку вкладку «*Liked Tracks»*

Так как нам нужно получать информацию от базы данных о композициях пользователя, то нужно установить его личность: войти или зарегистрировать аккаунт. Для входа и регистрации проверяется наличие соединения с сервером, корректность ввода пароля (допустимая длина, допустимые символы), корректность ввода подтверждения пароля (если пользователь хочет зарегистрировать аккаунт). Фрагмент диаграммы, соответствующий данному состоянию приведен на рисунке 2.5.

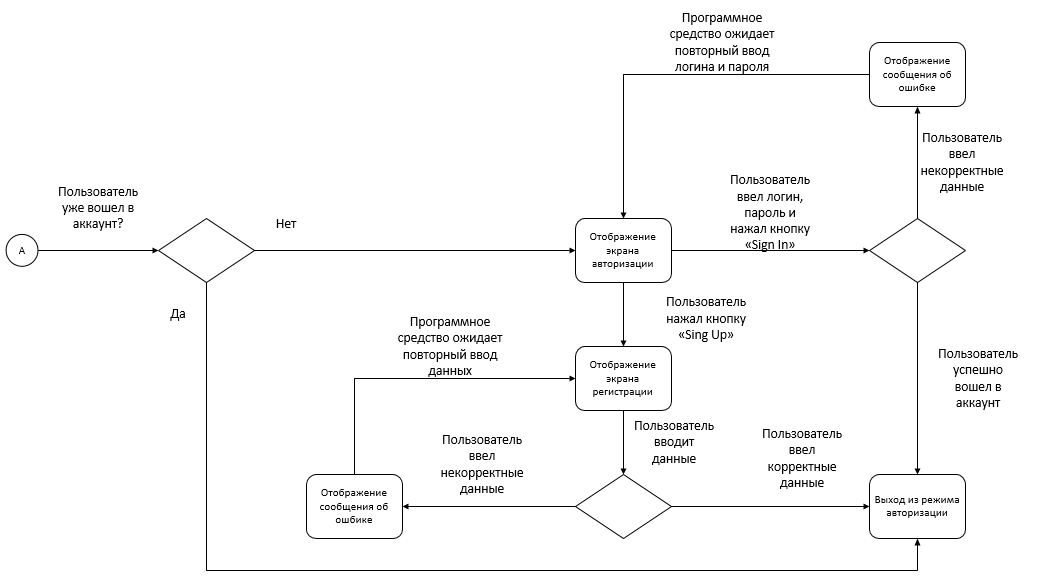


Рисунок 2.5 – Диаграмма состояний для авторизации пользователя

Также на главном экране пользователю предоставляется доступ к плееру по управлению текущим состоянием композиции. Пользователь может перемещать порядок композиций, перейти на предыдущую композицию, остановить текущий звуковой ряд или запустить проигрывание звука, перейти на следующую композицию, поставить повторное воспроизведение текущей композиции. Фрагмент диаграммы, соответствующий данному состоянию приведен на рисунке 2.6.

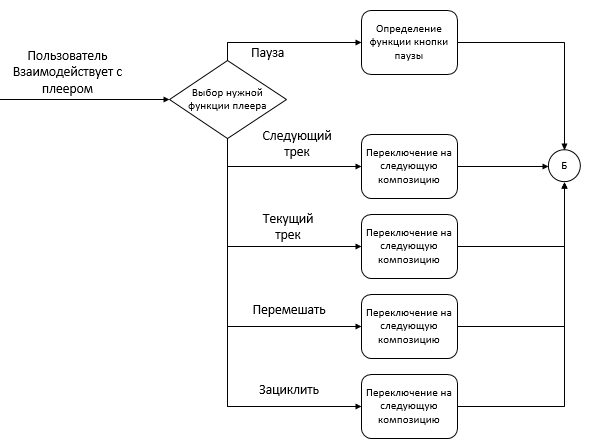


Рисунок 2.6 – Диаграмма состояний плеера

Пользователь может менять текущую стандартную директорию, в которую будут загружаться и из которой будут браться треки для воспроизведения. Диаграмма, соответствующая смене текущей стандартной директории представлена на рисунке 2.7.

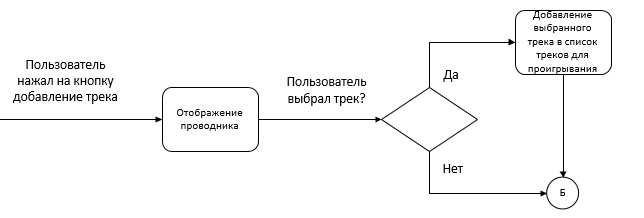


Рисунок 2.7 – Диаграмма состояний смены текущей стандартной директории

Пользователь может добавлять треки в текущую директорию, в которую будут загружаться треки. После того, как трек был добавлен в директорию страница с отображенными треками обновляется. Диаграмма, соответствующая добавлению в текущую директорию представлена на рисунке 2.8.

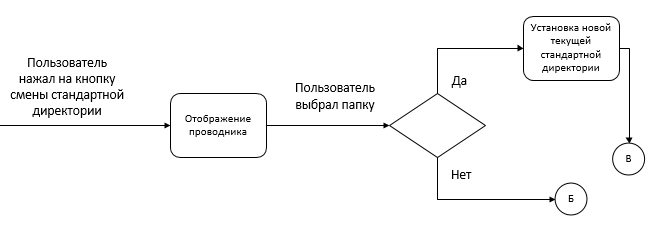


Рисунок 2.8 – Диаграмма состояний добавления в текущую директорию

Таким образом было произведено описание состояний программного средства при помощи нотации *UML* [8].

## Руководство по эксплуатации программного средства.

Начальный экран (см. рисунок 3.1) программного средства предоставляет пользователю следующий функционал:

– взаимодействие со скачанными треками;

– взаимодействие с лайкнувшими треками;

– взаимодействие с аккаунтом.

При нажатии на кнопку «Downloaded Tracks» происходит считывание из стандартной директории всех файлов с расширением «*.mp3*» и отображение их на экран. При повторном нажатии на эту же вкладку происходит обновление списка треков. На рисунке 3.1 приведен пример результата работы данной вкладки.

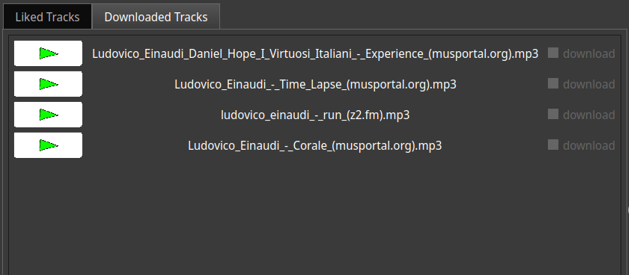


Рисунок 3.1 – Результат нажатия на вкладку «*Downloaded Tracks*»

При нажатии на кнопку «*Liked Tracks*» происходит считывание из базы данных всех файлов, которые относятся к данному пользователю, если пользователь приложения успешно вошел в аккаунт. При повторном нажатии на эту же вкладку происходит обновление списка треков. На рисунке 3.2 приведен пример результата работы данной вкладки.

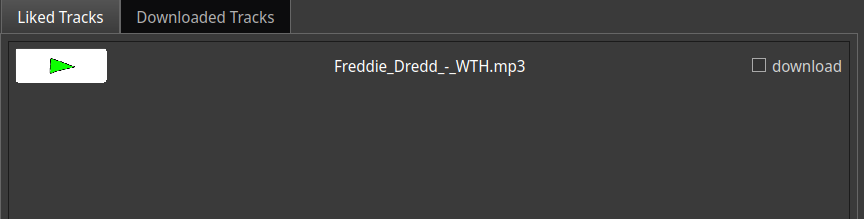


Рисунок 3.2 – Результат нажатия на вкладку «*Liked Tracks*»

Если же текущий пользователь не установлен, то при нажатии на кнопку «*Liked Tracks*» будет предложено войти в аккаунт (см. рисунок 2.10) или, если пользователь выберет опцию регистрации, зарегистрироваться. На этом этапе, если пользователь ввел данные, которые являются корректными, т.е. пароль не содержит запрещенных символов и пароль имеет подходящую длину, то будет произведена попытка подключиться к серверу для получения списка треков, принадлежащих данному пользователю. При неуспешном подключении к серверу, пользователь увидит на экране сообщение об ошибке (см. рисунок 3.3).

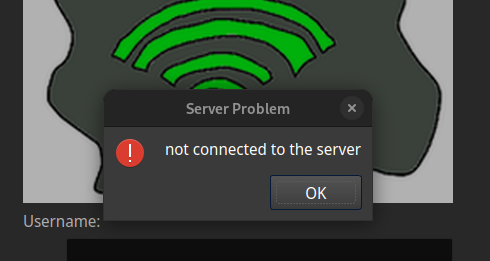


Рисунок 3.3 – Не удалось подключится к серверу

При успешном подключении к серверу будут проверены соответствие логина пользователя и его пароля. Предварительно на стороне клиента будут проверены поля: наличие пустых полей (см. рисунок 3.4), минимальна длина пароля (см. рисунок 3.5), содержание в пароле запрещенных символов (см. рисунок 3.6).

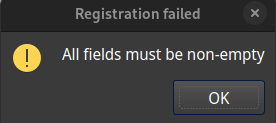


Рисунок 3.4 – Предупреждение о не заполненных полях

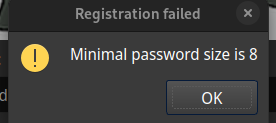


Рисунок 3.5 – Предупреждение о минимальной длине пароля

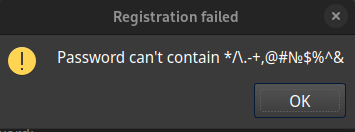


Рисунок 3.6 – Предупреждение запрещенных символах

Помимо данных предупреждений также есть проверка на соответствие поля «*Confirm password*» и поля «*Password*», для того, чтобы пользователь точно ввел правильный пароль при регистрации. Сообщение об ошибке, если это условие не соблюдается, приведено на рисунке 3.7.

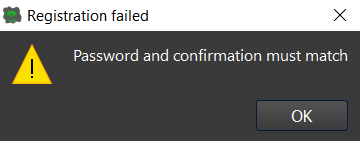


Рисунок 3.7 – Предупреждение запрещенных символах

После успешного входа в аккаунт у пользователя данного приложения отобразиться информация о наименовании аккаунта, в который пользователь вошел. Данная информация появится в левой части экрана: при успешном входе наименование аккаунта окраситься в зеленый цвет (см. рисунок 3.8) – в красный иначе (см. рисунок 3.9).

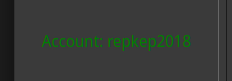


Рисунок 3.8 – Пользователь успешно вошел в аккаунт



Рисунок 3.9 – Пользователь вышел из аккаунта

Также в приложении предусмотрена возможность выхода из аккаунта (см. рисунок 3.10). Если текущий аккаунт не выбран – данная кнопка ничего не делает. Если же текущий пользователь вошел в аккаунт, то при нажатии на выход у пользователя запроситься подтверждение выхода (см. рисунок 3.11).

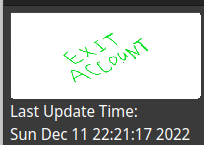


Рисунок 3.10– Кнопка выхода из аккаунта

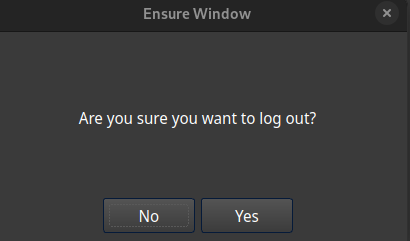


Рисунок 3.11 – Подтверждение выхода из аккаунта

Пользователь может добавлять композиции с устройства. Для этого ему нужно нажать на кнопку, представленную на рисунке 3.12 и находящуюся в левой нижней части экрана. После нажатия на эту кнопку у пользователя откроется проводник, установленный по умолчанию на его компьютере, через который появляется возможность добавлять композиции в текущий плейлист. На рисунке 3.12 данная кнопка обозначена как «*Add From Device*».

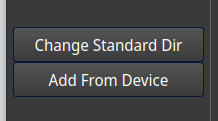


Рисунок 3.12 – Кнопка «*Add From Deviсе»* и кнопка «*Change Standard Dir»*

Скорее всего у пользователя появятся ситуации, при которых было бы удобнее не добавлять один, два, три и т.д. треков вручную, а просто выбрать директорию, в которой они расположены, и установить этот список треков как основной. На рисунке 3.12 данная возможность есть у кнопки «*Change Standard Dir*».

Данное программное средство призвано выполнять роль аудиоплеера, поэтому, как и у любого другого аудиоплеера, у него должна быть секция по управлению текущей композицией, возможность переключаться на следующий и предыдущий треки. Также для обеспечения полного функционала необходимы средства по регулировке положения текущей звуковой дорожки. Для регулировки звука на главной странице в левом углу установлен вертикальный ползунок, при перемещении которого будет меняться громкость текущей звуковой дорожки.

Первый элемент плеера – это слайдер по перемещению позиции в песне. Он представлен на рисунке 3.13. По бокам от него находятся цифры. Слева расположено время, прошедшее сначала композиции, а справа отображено полное время, которое нужно для воспроизведения трека. Также над самим слайдером есть наименование текущей проигрываемой композиции, получаемое по индексу текущего трека в плейлисте. При перемещении ползунка слайдера текущая композиция ставится на паузу, а при остановке перемещения (отпускания мышкой) проигрывание возобновляется. Это сделано для того, чтобы пользователь не слышал быструю перемотку трека, которая по своему звучанию напоминает помехи, при перемещении слайдера.

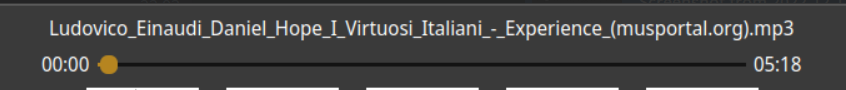


Рисунок 3.13– Слайдер и наименование трека

Следующим ключевым элементом плеера являются кнопки, которые располагаются под слайдером (см. рисунок 3.14). Их всего пять. Первая – это кнопка паузы. У плеера при нажатии на эту кнопку есть три возможных режима работы: пауза текущей композиции, возобновление проигрывания трека, начать проигрывать композиции сначала плейлиста. Последний режим может быть активирован, если до этого ни один трек не проигрывался и слайдер находится в положении, показанном на рисунке 3.15.



Рисунок 3.15 – Основные кнопки по управлению композицией

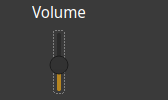


Рисунок 3.16 – Слайдер для изменения громкости звука

Каждая добавленная композиция отображается в меню выбранной вкладке как показано на рисунке 3.23. Чтобы начать проигрывать песню именно с этой композиции нужно навестись на кнопку, что находится левой части той же строки, что и название нужной композиции.

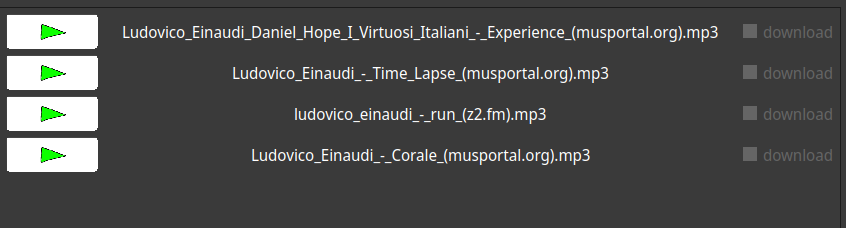


Рисунок 3.17 – Пример отображения строк композиций

Если пользователь решил добавить слишком много треков, т.е. такое количество, которое не умещается на один его экран, то появляется возможность прокручивать список аудиозаписей. (см. рисунок 3.17).

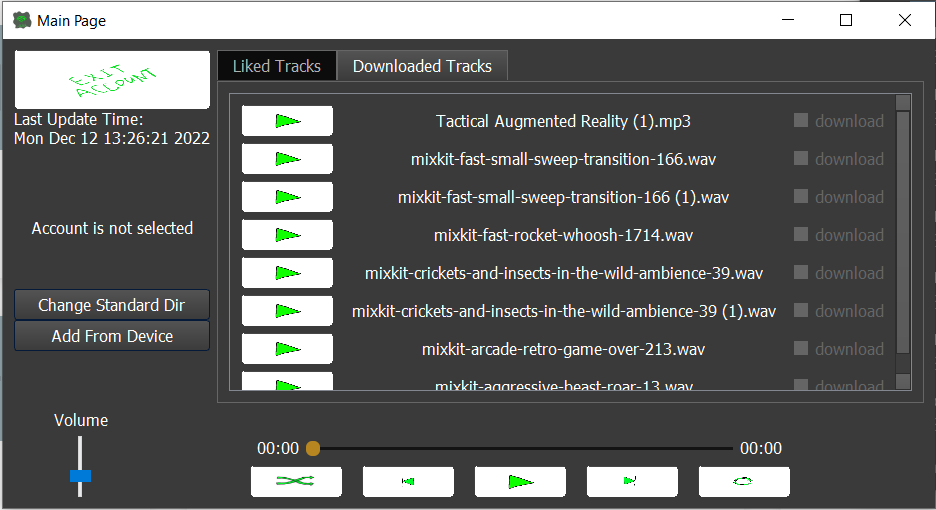


Рисунок 3.17 – Пример появления вертикального слайдера

Если же пользователь добавил трек, у которого название настолько длинное, что не вмещается в открытое окошко плеера, то появляется возможность прокручивать вбок аналогично рисунку 3.17.