МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное   
учреждение высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

**Разработка приложения Аудиоплеер**

ПРАКТИЧЕСКАЯ РАБОТА

по дисциплине «Программная инженерия»

ЮУрГУ – 09.03.04.20231.308-5692.КР

|  |  |
| --- | --- |
|  | Преподаватель  \_\_\_\_\_\_\_\_\_\_П.Д. Ческидов  Автор работы:  студент группы КЭ-303  \_\_\_\_\_\_\_\_\_\_Д.В. Старостенок  Работа защищена  с оценкой: \_\_\_\_\_\_\_\_\_\_\_  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

Челябинск 2023

Оглавление

[ВВЕДЕНИЕ 3](#_Toc133582482)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc133582483)

[2. ПРОЕКТИРОВАНИЕ 5](#_Toc133582484)

[3. РЕАЛИЗАЦИЯ 6](#_Toc133582485)

[ЗАКЛЮЧЕНИЕ 9](#_Toc133582486)

ВВЕДЕНИЕ

Актуальность:

В настоящее время аудиоплееры являются одними из самых популярных приложений на смартфонах, планшетах и компьютерах. Каждый день миллионы пользователей слушают музыку через различные аудиоплееры. Разработка нового аудиоплеера является важной задачей для программистов и компаний, которые хотят предложить своим пользователям новый и улучшенный функционал, что может привести к увеличению популярности и доходов.

Постановка задачи:

Цель данной работы − разработать современный аудиоплеер, который обладает широким функционалом и может конкурировать с другими популярными аудиоплеерами. Для достижения этой цели требуется выполнить следующие задачи:

* Изучить существующие аудиоплееры, анализировать их функциональность и дизайн, определить основные преимущества и недостатки.
* Спроектировать архитектуру приложения, выбрать язык программирования и технологии для реализации функционала.
* Реализовать основные функции аудиоплеера, такие как проигрывание, пауза, перемотка, изменение громкости, создание плейлистов и т.д.
* Разработать интерфейс пользователя с учетом современных тенденций в дизайне приложений.
* Протестировать приложение на различных устройствах и операционных системах, исправить ошибки и недочеты.

Итоговый результатом является приложение аудиоплеера, которое может быть использовано на различных устройствах и операционных системах.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Данная UML диаграмма (Рис. 1) описывает систему для управления музыкальными композициями и плейлистами, а также для загрузки и удаления плагинов.

В системе существует актер "Пользователь ПК", который может выполнить несколько действий, связанных с прецедентами. Пользователь может прослушивать отдельные композиции, прослушивать плейлисты, создавать, изменять и удалять плейлисты, а также загружать и удалять плагины. Для выполнения этих действий пользователь должен выполнить предварительный поиск композиций, плейлистов и плагинов, что отображено на диаграмме.

Прецедент "Прослушать отдельную композицию" связан с прецедентом "Найти песню", при этом отношение между ними обозначено как <<include>>. Прецеденты "Прослушать плейлист", "Удалить плейлист" и "Изменить плейлист" также связаны с прецедентом "Найти плейлист" через отношение <<include>>. Прецедент "Удалить плагин" связан с прецедентом "Найти плагин" через отношение <<include>>.

Таким образом, данная диаграмма является инструментом для понимания работы системы управления музыкальными композициями и плейлистами, а также для определения необходимых прецедентов для выполнения каждой из функций системы.

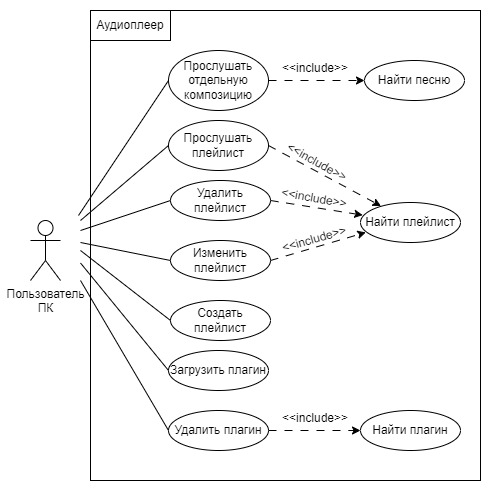


Рис. 1 – UML диаграмма

# ПРОЕКТИРОВАНИЕ

Имеется UML диаграмма классов (Рис. 2). В ней располагаются классы с RUP меткой <<entity>>:

Playlist – Описывающий атрибуты плейлиста, а также реализующий методы связанного с ним интерфейса IPlayable.

Song – Описывающий атрибуты определенной песни, а также реализующий методы связанного с ним интерфейса IPlayable.

Plugin – Описывающий атрибуты плагина.

Также имеются классы с RUP меткой <<control>>:

PlaylistController – Описывающий методы для работы с плейлистам, такие как их создание, редактирование, добавление и удаление песен из него.

SongController – Описывающий методы для работы с песнями, такие как переключение песни, а также методы реализуемые связанным интерфейсом ISearch.

PluginController – Описывающий методы для работы с плагинами, такие как их загрузка и удаление, а также методы реализуемые связанным интерфейсом ISearch.

Далее расположен класс с RUP меткой <<boundary>>:

AudioPlayerForm – Описывающий методы для взаимодействия с контроллерами.

Кроме классов имеются интерфейсы:

ISearch – Который имеет метод поиска.

IPlayable – Который имеет методы для проигрывания, паузы и остановки.

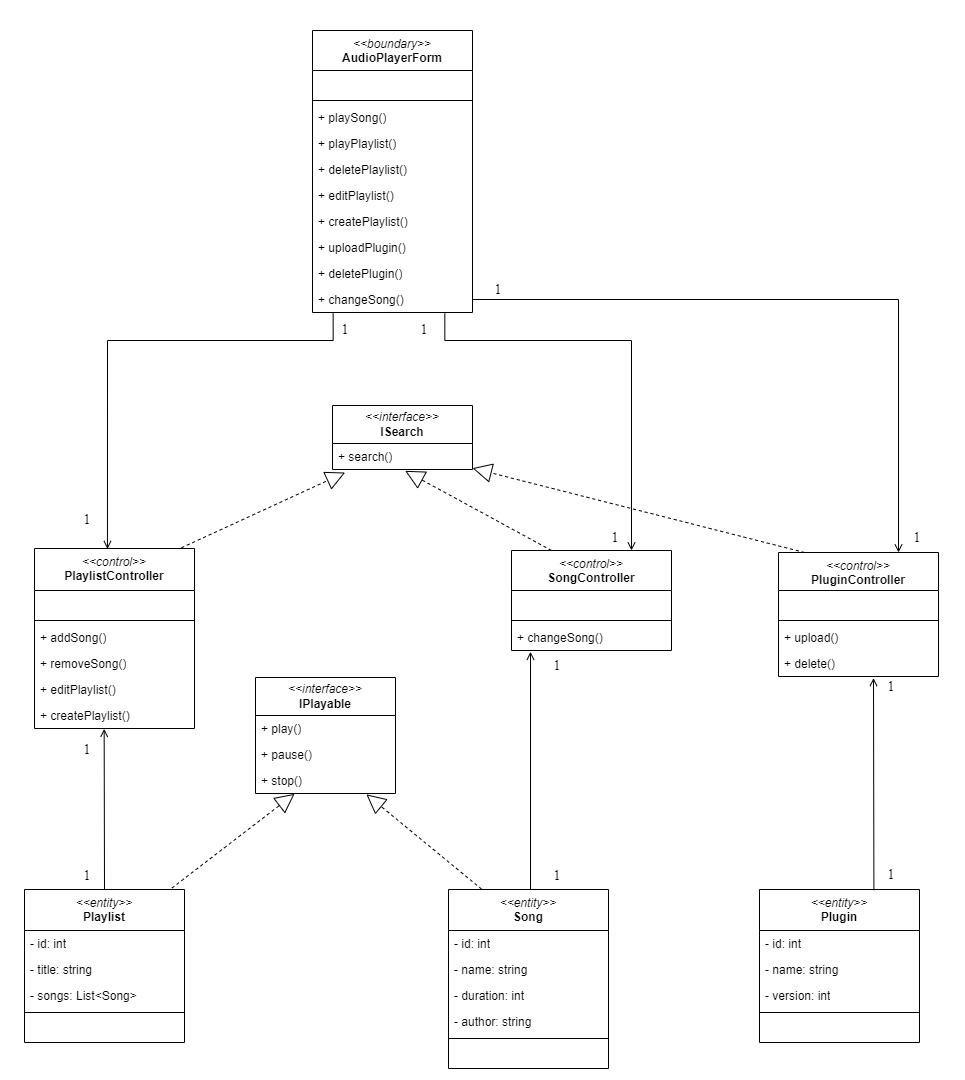


Рис. 2 – UML диаграмма классов

# РЕАЛИЗАЦИЯ

На рисунке 3 изображена диаграмма деятельности. Диаграмма деятельности начинается с узла "start", который означает начало процесса. Далее идет разделение на три параллельные ветки действий с помощью узла "fork".

Первая ветка действий относится к поиску композиции. Пользователь вводит запрос на поиск композиции, затем система ищет композицию. Если композиция найдена, то система проигрывает ее для пользователя. Если композиция не найдена, то происходит переход к следующей ветке действий.

Вторая ветка действий относится к поиску плейлиста. Пользователь вводит запрос на поиск плейлиста, затем система ищет плейлист. Если плейлист найден, то система проигрывает композиции из него для пользователя. Если плейлист не найден, то происходит проверка на необходимость создания нового плейлиста. Если пользователь хочет создать новый плейлист, то система создает его. Если пользователь не хочет создавать новый плейлист, то происходит переход к следующей ветке действий.

Третья ветка действий относится к поиску плагина. Пользователь вводит запрос на поиск плагина, затем система ищет плагин. Если плагин найден, то система удаляет его. Если плагин не найден, то выводится сообщение о том, что поиск не дал результатов.

Диаграмма деятельности заканчивается узлом "stop", который означает завершение процесса.

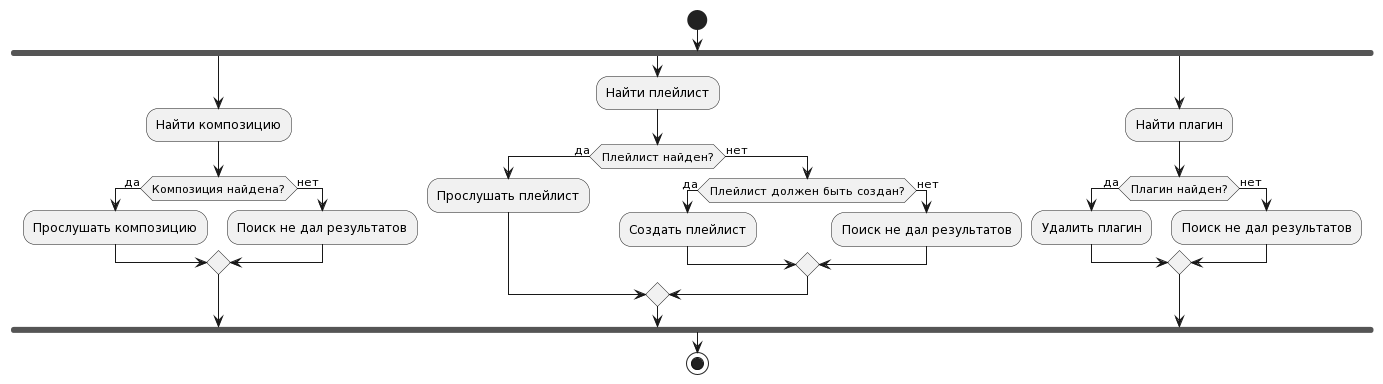


Рис. 3 – Диаграмма деятельности

На рисунке 4 изображена диаграмма последовательности. Диаграмма последовательности описывает три системы взаимодействия пользователя с компьютером: систему управления композициями, систему управления плейлистами и систему управления плагинами.

Система управления композициями находится в первой рамке и включает в себя два объекта: "Поиск композиций" и "Проигрыватель". Пользователь ПК начинает взаимодействие с системой, отправляя запрос "Найти песню" на "Поиск композиций". "Поиск композиций" передает выбранную композицию "Проигрывателю", который в свою очередь проигрывает выбранную композицию. Результат проигрывания композиции возвращается пользователю.

Система управления плейлистами находится во второй рамке и включает в себя два объекта: "Поиск плейлистов" и "Управление плейлистами". Пользователь ПК начинает взаимодействие с системой, отправляя запрос "Найти плейлист" на "Поиск плейлистов". "Поиск плейлистов" передает выбранный плейлист "Управлению плейлистами". Пользователь может прослушать плейлист, выбрав его в "Управлении плейлистами". "Управление плейлистами" передает выбранные композиции "Проигрывателю", который в свою очередь проигрывает их. Результат проигрывания композиций возвращается пользователю. Пользователь может также создать новый плейлист, изменить или удалить существующий плейлист.

Система управления плагинами находится в третьей рамке и включает в себя два объекта: "Поиск плагинов" и "Управление плагинами". Пользователь ПК начинает взаимодействие с системой, отправляя запрос "Найти плагин" на "Поиск плагинов". "Поиск плагинов" передает выбранный плагин "Управлению плагинами". Пользователь может загрузить выбранный плагин в "Управлении плагинами", а также изменить или удалить уже загруженные плагины.

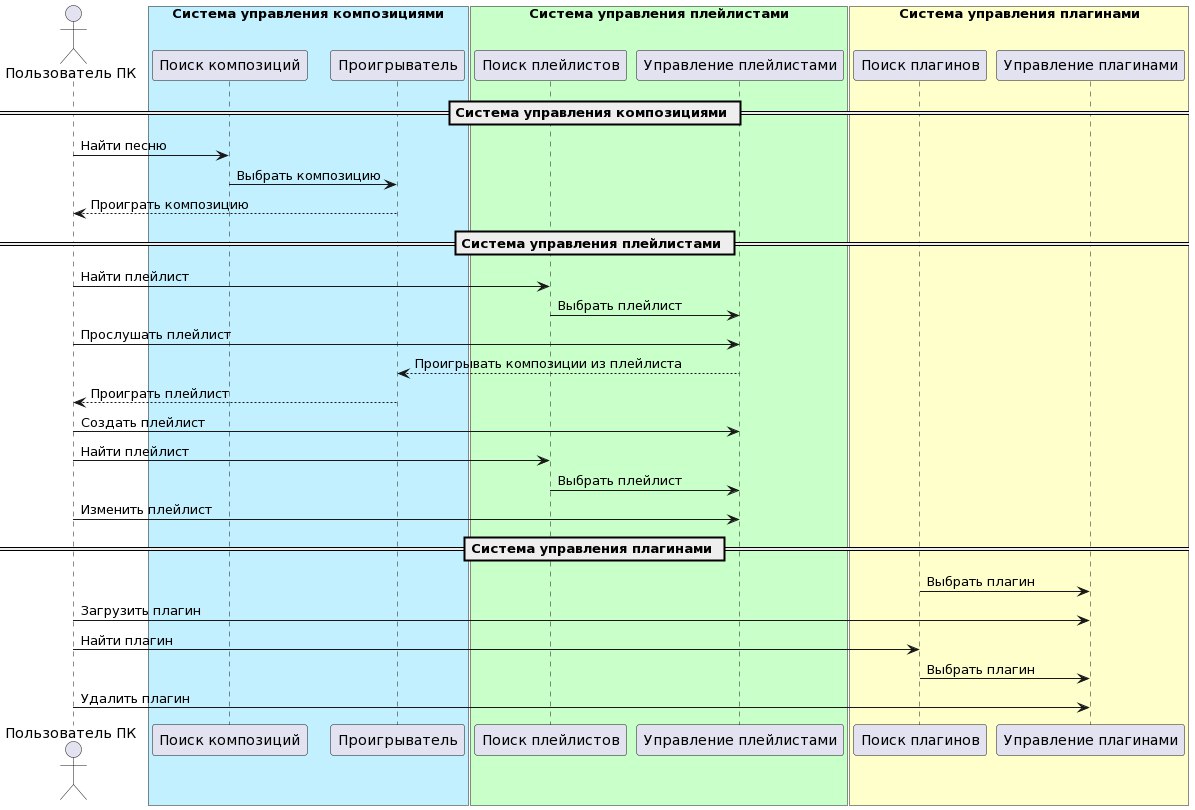


Рис. 4 – Диаграмма последовательности

ЗАКЛЮЧЕНИЕ

В ходе данной работы был разработан современный аудиоплеер, который обладает широким функционалом и может конкурировать с другими популярными аудиоплеерами. Для достижения поставленной цели были выполнены следующие задачи: было проведено исследование существующих аудиоплееров, была спроектирована архитектура приложения, выбраны язык программирования и технологии для реализации функционала, были реализованы основные функции аудиоплеера, был разработан интерфейс пользователя с учетом современных тенденций в дизайне приложений, а также приложение было протестировано на различных устройствах и операционных системах.

Полученный результат является положительным и соответствует поставленной задаче. Разработанный аудиоплеер может быть использован на различных устройствах и операционных системах, что расширяет его потенциальную аудиторию. Было продемонстрировано, что правильный выбор языка программирования и технологий, а также учет современных тенденций в дизайне приложений, способствует разработке современного и функционального аудиоплеера.