|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **УТВЕРЖДЕНО**  **RU.17701729.05.01-01 81 01-1-ЛУ** |  | |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | RU.17701729.05.01-01 ПЗ 01-1 | | **Клиентская часть мобильного приложения для конвертации голоса речи в аудиозаписях**  **Пояснительная записка**  **RU.17701729.05.01-01 81 01-1**  **Листов 55** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2022**

СОДЕРЖАНИЕ

[1. ВВЕДЕНИЕ 4](#_Toc114578533)

[1.1 Наименование программы на русском языке: 4](#_Toc114578534)

[1.2 Наименование программы на английском языке: 4](#_Toc114578535)

[1.3 Область применения программы: 4](#_Toc114578536)

[2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ 5](#_Toc114578537)

[2.1 Документы, на основании которых ведется разработка 5](#_Toc114578538)

[2.2 Наименование темы разработки 5](#_Toc114578539)

[3. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 6](#_Toc114578540)

[3.1 Функциональное назначение 6](#_Toc114578541)

[3.2 Эксплуатационное назначение 6](#_Toc114578542)

[4. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 7](#_Toc114578543)

[4.1 Постановка задачи на разработку программы 7](#_Toc114578544)

[4.2 Описание алгоритма и функционирования программы 7](#_Toc114578545)

[4.2.1 Описание алгоритма программы 7](#_Toc114578546)

[4.2.1.1 Описание схемы обработки аудио 7](#_Toc114578547)

[4.2.2 Обоснование выбора алгоритма решения задачи 11](#_Toc114578548)

[4.3 Описание и обоснование выбора метода организации входных и выходных данных 12](#_Toc114578549)

[4.3.1 Описание метода организации входных и выходных данных 12](#_Toc114578550)

[4.3.2 Обоснование выбора метода организации входных и выходных данных 12](#_Toc114578551)

[4.4 Описание и обоснование выбора организации хранения данных 12](#_Toc114578552)

[4.5 Описание и обоснование выбора состава технических и программных средств 12](#_Toc114578553)

[4.5.1 Требования к составу и параметрам технических средств 13](#_Toc114578554)

[4.5.2 Требования к разрешениям программе на техническом устройстве 13](#_Toc114578555)

[1.1.1 Требования к информативной и программной совместимости 13](#_Toc114578556)

[5. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 14](#_Toc114578557)

[5.1 Ориентировочная экономическая эффективность 14](#_Toc114578558)

[5.2 Предполагаемая потребность 14](#_Toc114578559)

[5.3 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 14](#_Toc114578560)

[5.3.1 Отечественные аналоги 14](#_Toc114578561)

[5.3.2 Зарубежные аналоги 14](#_Toc114578562)

[5.3.3 Преимущества данной программы 14](#_Toc114578563)

[6. СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 15](#_Toc114578564)

[ПРИЛОЖЕНИЕ 1 16](#_Toc114578565)

[ПРИЛОЖЕНИЕ 2 17](#_Toc114578566)

[ПРИЛОЖЕНИЕ 3 19](#_Toc114578567)

1. **ВВЕДЕНИЕ**
   1. **Наименование программы на русском языке:**

«Клиентская часть мобильного приложения для конвертации голоса речи в аудиозаписях»

* 1. **Наименование программы на английском языке:**

«Client Part of Mobile Application for Voice Conversion in Audio Recordings»

* 1. **Область применения программы:**

В настоящее время применение нейросетевых технологий для обработки аудиозаписей набирает популярность как в рамках теоретических исследований, так и при разработке пользовательских приложений. Людям бывает интересно, насколько похоже на оригинальный голос будет звучать аудиозапись, «окрашенная» в этот голос компьютерными средствами. Наше приложение направлено на удовлетворение этого интереса и предоставление пользователям возможности изучить способности нейросетей по изменению аудио, а также создавать собственные аудио с помощью такой обработки.

1. **ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ**
   1. **Документы, на основании которых ведется разработка**

Программа выполняется в рамках темы курсового проекта в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия» Национального исследовательского университета «Высшая школа экономики», факультет компьютерных наук, департамент программной инженерии.

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 "Программная инженерия" и утвержденная академическим руководителем тема курсового проекта.

* 1. **Наименование темы разработки**

**Наименование темы разработки** – «Клиентская часть мобильного приложения для конвертации голоса речи в аудиозаписях».

**Условное обозначение темы разработки** – приложение VoiceVersa.

1. **НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**
   1. **Функциональное назначение**

Приложение предназначено для обработки аудиозаписей, а именно для конвертации голоса аудиозаписи речи в выбранный голос и сопутствующих действий, таких как запись аудио, выбор голоса, локальное хранение аудиозаписей и ассоциированной с ними информации.

Функции, помогающие это реализовать:

1. регистрация и аутентификация аккаунта пользователя;
2. выгрузка, запись, прослушивание, локальное сохранение исходной аудиозаписи;
3. отображение и выбор доступных для конвертации голосов;
4. обработка аудиозаписи в соответствии с выбранным голосом;
5. прослушивание, сохранение и локальное сохранение полученной аудиозаписи.
   1. **Эксплуатационное назначение**

Приложение может быть использовано любым человеком, который хочет конвертировать голос речи в аудиозапись с помощью смартфона на платформе Android. Наша программа поможет это сделать.

1. **ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**
   1. **Постановка задачи на разработку программы**

При постановке задачи были проведены опрос и анализ интересов потенциальных пользователей, и на основании выводов были выделены основные функции, которые нужны в приложении такого рода. Таким образом, задача заключалась в написании программы, которая должна позволять пользователю:

1. Регистрироваться с использованием адреса электронной почты и паролем
2. Авторизоваться через адрес электронной почты с реализацией:
   1. Хранения данных на сервере;
   2. Синхронизации данных одного аккаунта на разных устройствах;
   3. Резервного копирования данных на устройстве для возможности бессетевой работы приложения;
   4. Авторизации на основе прошлых сеансов работы.
3. Входить в приложение без авторизации - в режиме гостя.
4. Добавлять аудиозапись для преобразования путем
   1. записи аудио в приложении;
   2. загрузки существующего файла.
5. Прослушивать добавленную аудиозапись.
6. Сохранять аудиозапись для преобразования на устройство.
7. Прослушивать доступные голоса для преобразования.
8. Выбирать голос для преобразования аудио из доступных.
9. Преобразовывать добавленное аудио в итоговое аудио.
10. Прослушивать полученную аудиозапись.
11. Сохранять полученную аудиозапись на устройство.

Кроме этого, приложение должно позволять авторизованным пользователям:

1. Выходить из своей авторизованной учетной записи.
2. Сохранять и просматривать аудиозапись для преобразования в приложении.
3. Сохранять и просматривать полученную аудиозапись в приложении.
4. Включать и отключать автосохранение обоих видов аудиозаписей в приложении.
5. Отправлять заявку на добавление желаемого голоса в доступные для преобразования путем:
   1. загрузки архива аудиозаписей;
   2. записи аудио в приложении.
   3. **Описание алгоритма и функционирования программы**
      1. Описание алгоритма программы
         1. Описание схемы обработки аудио

Данное приложение направленно на обработку аудиозаписей в методике переноса стиля. Эта схема предполагает несколько элементов: аудиозапись – источник, аудиозапись – голос и аудиозапись – результат.

*Источник* – это аудиозапись, которую создает пользователь, и из которой в результате остается речь – произнесенные элементы, содержание.

*Голос* – это аудиозапись, на которой была натренирована нейросетевая модель, используемая для преобразования аудио. Из голоса в конечный результат переносится стиль – уникальные звуковые характеристики человеческой речи, например тембр.

*Результат* – это цель обработки, создаваемая нейросетевой моделью с помощью комбинации источника и голоса. Это аудиозапись, содержащая речь источника и стиль голоса, то есть не существовавшую до этого комбинацию содержания и формы речи.

* + - 1. Описание общей схемы работы системы

Работа программы разделена на несколько логических блоков, которые идеологически соответствуют экранам в приложении:

1. Вход пользователя в приложение.
2. Обработка аудиозаписи.
3. Просмотр сохраненных аудиозаписей в библиотеке.
4. Изменение настроек приложения.
5. Создание заявки на обработку собственной аудиозаписи как голоса.

Для реализации этих блоков создан интерфейс, который также содержит вспомогательный блок навигации. Хотя блок обработки аудио и является наиболее важным и представляет собой собственно минимальный продукт, связанный с жизненным циклом аудиозаписи при ее преобразовании, остальные части реализуют функции, которые также могут быть интересны пользователю.

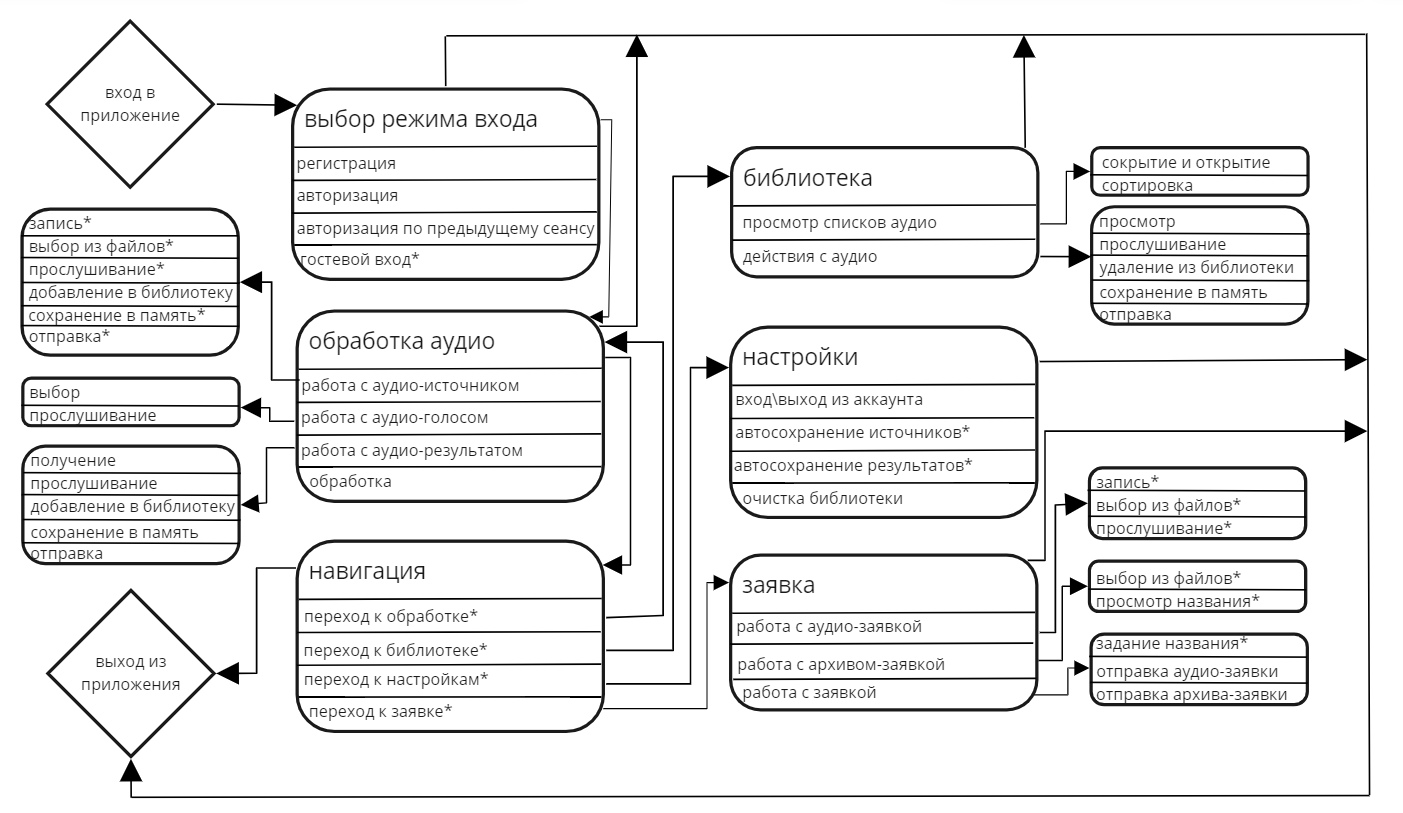


Рисунок 1 – Общая блок-схема программы. Действия, отмеченные символом \* - это действия, доступные пользователю без подключения к серверу.

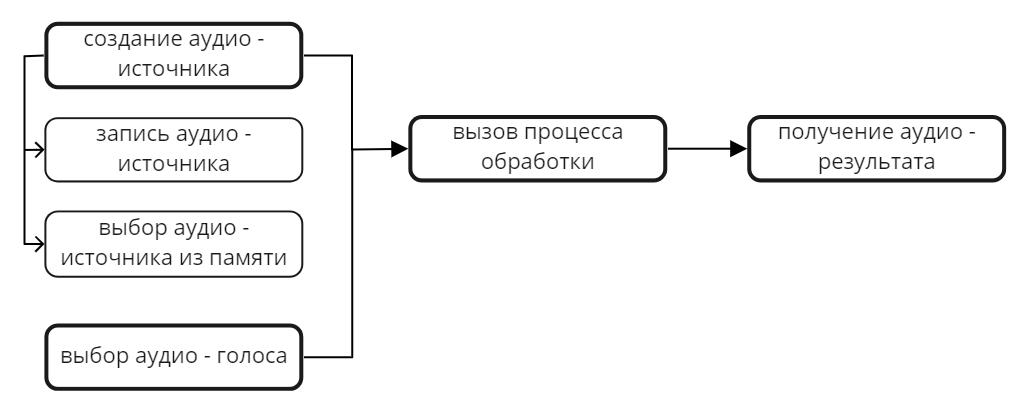


Рисунок 2 – Более подробная схема обработки аудио. Действия в толстых рамках выполняются параллельно по вертикали и последовательно – по горизонтали.

* + - 1. Описание входа пользователя в приложение

Вход пользователя осуществляется с помощью либо регистрации\авторизации через сервер, либо гостевого входа с локальным сохранением данных.

В первом случае пользователь может зарегистрироваться с помощью адреса электронной почты как именем пользователя и введенным паролем. Авторизация происходит через сервер по этим же данным. Также возможен вход при использовании сохраненных с прошлых сеансов авторизованной работы с приложением данных, которые заносятся на устройство в файлы SharedPreferences [10].

Если пользователь не хочет по какой-то причине регистрироваться или входить в учетную запись, возможен гостевой вход без подключения к серверу, но в таком случае функционал приложения ограничен, так как данные хранятся и обрабатываются только локально на устройстве.

* + - 1. Описание обработки аудиозаписи
         1. Описание основного процесса обработки

Создание и обработка аудиозаписи являются основным функциональным элементом этого приложения. Экран, реализующий эти функции, последовательно проводит пользователя через жизненные стадии аудиозаписи для ее обработки.

Сначала создается аудиозапись-источник, которую пользователь либо записывает с помощью микрофона, либо прикрепляет с помощью меню выбора файлов, сохраненных на устройстве.

Далее выбирается голос из списка голосов, получаемых с сервера. Голоса, доступные для обработки – это именно те голоса, в которые можно «окрасить» исходное аудио для получения результата.

После создания источника и выбора голоса становится возможным вызов обработки аудио – «окрашивания» источника в выбранный голос, происходящего на сервере. По ответу от сервера пользователю становится доступен результат обработки исходного аудио. На этом основная обработка аудио заканчивается.

Загрузка голосов, а также непосредственно обработка аудио, ожидаемо, происходит с помощью сервера. По этой причине при бессетевой работе или гостевом входе доступен только функционал, связанный с созданием и сохранением источника.

* + - * 1. Описание дополнительных функций

Для удобства работы с приложением интерфейс также предусматривает остановку, возобновление, повторное включение записи аудио с микрофона, прослушивание и перемотку источника и результата.

Отображения аудиозаписей источника и результата также содержат меню для дальнейших действий с аудио. Аудиозаписи можно сохранить локально на устройстве: сохранение осуществляется в папку загрузок. Также, при условии зарегистрированного входа и подключения к серверу, аудио можно добавить в библиотеку приложения – тогда они будут отображаться на экране библиотеки, сохраняться между запусками приложения на одном устройстве и синхронизироваться на устройствах при входе в один и тот же аккаунт. Кроме того, можно отправить аудиозапись с помощью меню отправки файлов.

Также важно отметить: при отметке параметров автосохранения в настройках как активных аудиозаписи будут загружаться на устройство автоматически. Для удобства пользователя отметить автосохранение как активное или неактивное можно отдельно для источников и результатов.

* + - 1. Описание просмотра сохраненных аудиозаписей в библиотеке

Экран библиотеки помогает пользователю просматривать сохраненные в приложении аудиозаписи, которые были созданы ранее: как источники, так и результаты. Эта опция доступна при авторизованном входе и подключении к серверу, так как библиотека аудио хранится там.

Аудиозаписи разделены на 2 списка: источники и результаты, для более наглядного отображения и понятности для пользователя. Списки можно сворачивать и разворачивать по желанию.

Также возможно отсортировать элементы списка: по названию, дате создания и длительности. Это сделано для большего удобства при просмотре и поиске аудио.

Каждый элемент содержит название аудио, дату создания, длительность и элементы для прослушивания аудио: кнопку включения\выключения проигрывания, перемотки. Также каждому элементу соответствует меню: аудио можно сохранить на устройстве и отправить куда-либо (как на экране обработки аудио), и удалить из библиотеки. В этом случае аудиозапись удаляется также и на сервере из библиотеки пользователя.

Загрузка аудиозаписей библиотеки с сервера происходит в момент открытия экрана, а обновление информации на сервере – при каждом действии изменения статуса аудио и, таким образом, отображение списка аудио всегда остается актуальным.

* + - 1. Описание изменения настроек приложения

Редактирование настроек приложение осуществляется с помощью экрана настроек. Там пользователь может задать параметры для работы приложения.

Первый блок содержит настройки аккаунта: отображение того, с каким именем пользователь в данный момент работает в приложении, а также функции входа и выхода из аккаунта (активны если пользователь не вошел и вошел в аккаунт соответственно). При входе пользователь направляется на авторизацию на первом экране приложения. При выходе осуществляется очистка информации об аккаунте пользователя: стирается имя пользователя, пароль и токен для входа, также очищаются SharedPreferences для входа по предыдущим сеансам работы. Таким образом персональные данные пользователя остаются сохраненными на сервере и исчезают с устройства, кроме загруженных в локальную память элементов.

Настройки автосохранения источников и результатов помогают установить ожидаемое дефолтное поведение по загрузке аудиозаписей: чтобы пользователю не приходилось каждый раз загружать результат, можно выставить флаг автозагрузки, тогда аудиозапись будет добавляться в папку загрузок устройства после создания автоматически. При отключении автосохранения новые аудиозаписи не будут сохраняться автоматически, но аудиозаписи, загруженные до этого, останутся.

* + - 1. Описание создания заявки на обработку собственной аудиозаписи как голоса

Интересной функцией, которая может быть особенно привлекательна для пользователей является возможность отправить на сервер собственную заявку для голоса. Для этого можно использовать новый файл, записанный в приложении, либо архив с аудиозаписями, прикрепляемый через окно выбора файла. Выбранные файлы отправляются на сервер для дальнейшей обработки.

Так как для добавления какой-либо аудиозаписи в список голосов и последующей возможности окрашивать в нее другие аудио необходима перетренировка нейросетевой модели на материале потенциального нового голоса, заявка обрабатывается вручную модератором сервера, и требует существенного времени, поэтому в приложении не предусмотрен никакой немедленный ответ на заявку. Однако, как только обработка заявки будет осуществлена, голос станет доступен в загружаемых с сервера голосах. Важно отметить, что голоса общие для всех пользователей приложения, поэтому обработанный по заявке голос также будет доступен всем пользователям, а не только автору.

* + 1. Обоснование выбора алгоритма решения задачи

За основу для составления алгоритма работы данного приложения были взяты, с одной стороны, потребности потенциальных пользователей относительно подобной программы, и, с другой стороны, используемая схема нейросетевой обработки аудио. Сочетание этих факторов определило то, какие входные данные требуются от пользователя, какие функции должны быть связаны с этими данными.

Выбор метода обработки аудио связан с эффективностью данной схемы и конкретной реализации нейросетевой модели [12]. Так как приложение стремится исследовать возможность применения этих технологий в обывательском использовании и познакомить пользователей с прогрессом в данной области, схема обработки является наиболее существенной. Так, аудио-источник должно формироваться в идеале из максимально чистой записи голоса, голос – выбираться из доступных, на которых была натренирована модель, и результат формируется на их основе.

То же самое относится к алгоритму отправки заявки. Так как требуется аудиозапись, где наиболее слышен голос человека, создание аудио преимущественно осуществляется с помощью микрофона.

При выборе методов реализации остальных функций в приоритете были требования пользователей к системе. Например, просмотр аудиозаписей в библиотеке обусловлен основными функциями, существующими в приложениях, занимающихся отображением аудио: прослушивание, перемотка, загрузка, удаление, отправка.

Алгоритм хранения данных как метод разделения локального скачивания и сохранения в библиотеку обусловлен возможными требованиями пользователей, связанными с хранением аудиозаписей: это может быть слишком требовательно к памяти в случае только локального сохранения. В связи с этим была создана возможность удаленного хранения данных на сервере.

При выборе редактируемых настроек учитывалось то, какие функции могут быть актуальны пользователю. Так, с наличием авторизации в аккаунте может быть полезным и выход из аккаунта с целью обеспечения информационной безопасности.

Вход в приложение был реализован подобным образом с целью обеспечить баланс между простотой использования, то есть легкостью процедуры входа – без каких-либо двухфакторных аутентификаций и т.п. в силу относительной обособленности данного приложения, и безопасностью, так как вне зависимости от простоты решения оно не должно компрометировать личные данные пользователя. Возможность входа на основании предыдущих сеансов была мотивированна главным образом удобством пользователя.

Режим гостевого входа, хотя он и содержит ограниченный функционал, был добавлен с целью обеспечения возможности пользования приложением офлайн – главным образом для удобства пользователя.

* 1. **Описание и обоснование выбора метода организации входных и выходных данных**
     1. Описание метода организации входных и выходных данных

Входными данными программы являются строки, файлы, отправленные пользователем, обработка нажатий на экран, также аудиофайлы в случае записи пользователем исходного аудио.

Выходными данными программы являются обработанная аудиозапись и ее отображение в приложении, а также отображение информации на экране, включая внесенные пользователем данные, сообщения при работе с системой. Для этого используются элементы контроля: кнопки, текстовые поля, сообщения.

* + 1. Обоснование выбора метода организации входных и выходных данных

Описанный выше формат входных данных был выбран в соответствии с требованиями, предъявленными к входным данным в техническом задании, а также в связи с требованиями схемы обработки. Формат аудио mp3 выбран в силу его распространенности (что может быть полезным при, например, выборе источника из файлов, сохраненных на устройстве), легковесности и относительной универсальности. Ввод через интерфейс сделан таким образом, чтобы пользователю было наиболее привычно с ним взаимодействовать.

Выходные данные в части вывода в интерфейс организованы также с целью сделать отображение информации в приложении максимально информативным, при этом не перегружая интерфейс.

* 1. **Описание и обоснование выбора организации хранения данных**

Файлы хранятся в памяти, ассоциированной с приложением на устройстве, в виде mp3 файлов (аудиозаписей) и записей в SharedPreferences. Аудио файлы находятся в папке, доступной пользователю, что может быть полезно для пользователя. Файлы SharedPreferences сохраняются в приватном режиме и недоступны для чтения, благодаря этому можно рассчитывать на надежность их хранения. Так как существует опция загрузки основных данных (библиотеки пользователя) на сервере, проблема нехватки памяти для хранения аудиозаписей не стоит так остро.

* 1. **Описание и обоснование выбора состава технических и программных средств**

Программные и технические средства выбраны на основании минимальных требований, необходимых для работы программы, предназначенной для смартфонов на платформе Android. Платформа Android выбрана в силу ее распространенности и удобства для разработки. В силу того, что приложение предусматривает локальное сохранение данных, необходимо достаточное количество памяти на устройстве.

Так как приложение взаимодействует с сервером для обеспечения большей части функций, приложение должно иметь разрешение на выход в интернет. Кроме этого разрешение на использование микрофона очевидно необходимо для записи аудио.

* + 1. Требования к составу и параметрам технических средств

Для корректной работы программы необходимо выполнение следующих требований к техническим средствам:

1. Смартфон на базе платформы Android;
2. Экран с разрешением 720х1280, 1080х1920 или 2560×1440 пикселей;
3. Не менее 256 Мб оперативной памяти;
4. Не менее 64 Мб встроенной памяти или памяти на SD-карте.
   * 1. Требования к разрешениям программе на техническом устройстве
5. Программа должна иметь разрешение на выход в сеть Интернет.
6. Программа должна иметь разрешение на запись аудио.
7. Программа должна иметь разрешение на чтение и сохранение файлов на устройстве.
   * 1. Требования к информативной и программной совместимости

Для корректной работы программы необходимо выполнение следующих требований к программным средствам:

1. Лицензионная операционная система Android 8.1 или выше;
2. Доступ к сети интернет для установки.
3. **ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**
   1. **Ориентировочная экономическая эффективность**

Расчет экономической эффективности не предусмотрен.

* 1. **Предполагаемая потребность**

Программа ориентирована на эксплуатацию отдельными пользователями, имеющими мобильное устройство и желание заняться конвертацией голоса речи своих аудиозаписей.

* 1. **Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами**

При исследовании предметной области были выделены следующие наиболее популярные аналоги нашего программного продукта:

* + 1. **Отечественные аналоги**

1. Мобильное приложение “ВКонтакте” [1]
2. Мобильное приложение “Телеграм” [2]
   * 1. **Зарубежные аналоги**
3. Мобильное приложение “Narrator’s Voice” [8]
4. Мобильное приложение “Voice FX” [14]
5. Мобильное приложение “Voice Recorder” [15]
6. Нейросетевая модель «StarGANv2-VC»[11]
   * 1. **Преимущества данной программы**

В сравнении с аналогами приложение обладает следующими преимуществами:

1. использование реалистичных голосов;
2. нейросетевая обработка аудиозаписи;
3. широкий набор функций;
4. бесплатность.
5. **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**
6. ВКонтакте [Электронный ресурс] //URL: https://play.google.com/store/apps/details?id=com.vkontakte.android (Дата обращения: 26.01.2022, режим доступа: свободный)
7. Телеграм [Электронный ресурс] //URL: https://play.google.com/store/apps/details?id=org.telegram.messenger (Дата обращения: 26.01.2022, режим доступа: свободный)
8. Babich, Nick. A Comprehensive Guide To Mobile App Design [Электронный ресурс] // URL: <https://www.smashingmagazine.com/2018/02/comprehensive-guide-to-mobile-app-design/> (Дата обращения: 01.04.2022, режим доступа: свободный)
9. Data and file storage overview [Электронный ресурс] // URL: https://developer.android.com/training/data-storage (Дата обращения: 01.04.2021, режим доступа: свободный)
10. MediaMetadataRetriever [Электронный ресурс] // URL: https://developer.android.com/reference/android/media/MediaMetadataRetriever (Дата обращения: 01.05.2022, режим доступа: свободный)
11. MediaPlayer [Электронный ресурс] // URL: https://developer.android.com/reference/android/media/MediaPlayer (Дата обращения: 01.05.2022, режим доступа: свободный)
12. MediaRecorder overview [Электронный ресурс] // URL: https://developer.android.com/guide/topics/media/mediarecorder(Дата обращения: 01.05.2022, режим доступа: свободный)
13. Narrator’s Voice [Электронный ресурс] //URL: https://play.google.com/store/apps/details?id=br.com.escolhatecnologia.vozdonarrador&hl=en\_US&gl=US (Дата обращения: 26.01.2022, режим доступа: свободный)
14. Retrofit 2 — How to Download Files from Server [Электронный ресурс] // URL: https://futurestud.io/tutorials/retrofit-2-how-to-download-files-from-server (Дата обращения: 01.08.2022, режим доступа: свободный)
15. Save key-value data [Электронный ресурс] // URL: https://developer.android.com/training/data-storage/shared-preferences (Дата обращения: 01.09.2022, режим доступа: свободный)
16. StarGANv2-VC [Электронный ресурс] //URL: <https://github.com/yl4579/StarGANv2-VC> (Дата обращения: 26.01.2022, режим доступа: свободный)
17. StarGANv2-VC: A Diverse, Unsupervised, Non-parallel Framework for Natural-Sounding Voice Conversion [Электронный ресурс] //URL: https://arxiv.org/abs/2107.10394 (Дата обращения: 26.01.2022, режим доступа: свободный)
18. Upload file via Multipart POST method in Kotlin [Электронный ресурс] //URL: https://medium.com/@Tarek360/upload-file-via-multipart-post-method-in-kotlin-d43d3f025670 (Дата обращения: 15.08.2022, режим доступа: свободный)
19. Voice FX [Электронный ресурс] //URL: https://play.google.com/store/apps/details?id=com.mobzapp.voicefx&hl=ru&gl=US (Дата обращения: 26.01.2022, режим доступа: свободный)
20. Voice Recorder [Электронный ресурс] //URL: https://play.google.com/store/apps/details?id=com.splendapps.voicerec&hl=ru&gl=US (Дата обращения: 26.01.2022, режим доступа: свободный)

ПРИЛОЖЕНИЕ 1

**ТЕРМИНОЛОГИЯ**

Таблица 1 – Терминология

|  |  |
| --- | --- |
| **Термин** | **Определение** |
| Речь | Аудиозапись с произнесенными вслух словами. Состоит из *текста* – словесной составляющей - и *голоса* - совокупности уникальных звуковых характеристик аудиозаписи. |
| Конвертации голоса речи, перенос стиля речи, окрашивание аудиозаписи | Это процесс обработки аудиозаписей, при котором голос из одной аудиозаписи совмещается с текстом из другой аудиозаписи (с изначально другим голосом), тем самым производя новую аудиозапись с ранее не существовавшей комбинацией голоса и текста |
| Аудиозапись- источник, источник | Речь, которая обрабатывается с целью извлечения из нее текста. |
| Аудиозапись- голос, голос | Речь, которая обрабатывается с целью извлечения из нее голоса. |
| Результат | Речь, получаемая в результате переноса стиля речи. Содержит текст источника и голос голоса. |

ПРИЛОЖЕНИЕ 2

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ**

**Классы пакета model приложения VoiceVersa**

*Таблица 2 - Классы пакета model приложения VoiceVersa*

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| Audio | Класс для репрезентации аудиозаписи в приложении. |
| User | Класс для репрезентации пользователя в приложении. Хранит информацию, относящуюся к текущему сеансу. |

**Классы пакета serverClasses приложения VoiceVersa**

*Таблица 3 - Классы пакета serverClasses приложения VoiceVersa*

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| AudioApiService | Интерфейс для сервиса общения с сервером – описывает запросы к серверу. |
| AudioFromServer | Дата-класс для репрезентации ответа с аудиозаписью (хранимой в библиотеке) от сервера. |
| AudioInfo | Дата-класс, содержащий информацию об аудио с сервера. Вложен в AudioFromServer. |
| AudioListResponse | Дата-класс для ответа от сервера, содержащий массив аузиозаписей с сервера (AudioFromServer или VoiceFromServer). |
| LoginRequest | Дата-класс для запроса к серверу на вход в аккаунт. |
| ResultFromServer | Дата-класс для ответа сервера на запрос об обработке аудио – результат обработки. |
| Token | Дата-класс для токена при общении с сервером. |
| VoiceFromServer | Дата-класс для ответа с аудиозаписью (голосом) от сервера. |

**Классы пакета controller приложения VoiceVersa**

*Таблица 4 – Классы пакета controller приложения VoiceVersa*

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| Controller | Класс – контроллер, осуществляет основную логику вне классов активити, через него происходит обращение к сервису общения с сервером и заполнение данных модели. |

**Классы пакета view приложения VoiceVersa**

*Таблица 5 - Классы пакета view приложения VoiceVersa*

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| AccountActivity | Класс для экрана навигации по приложению, личный кабинет, содержит переходы на экраны обработки аудио, экран библиотеки, экран настроек и экран заявки. |
| AuthActivity | Класс для входа в приложение, отображает элементы для регистрации, аутентификации или гостевого входа пользователя. |
| ProcessActivity | Класс для экрана обработки аудио – записи источника, выбора голоса, преобразования аудио. |
| RequestActivity | Класс для экрана заявки на обработку аудио как голоса – содержит запись и проигрывание аудио, прикрепление выбранного файла, отправка на сервер. |
| SettingsActivity | Класс для экрана настроек – содержит элементы для выхода из аккаунта, очистки библиотеки, настройки автосохранения. |

**Классы пакета view.libraryActivity приложения VoiceVersa**

*Таблица 6 - Классы пакета view.libraryActivity приложения VoiceVersa*

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| AudiosListAdapter | Класс адаптер для списка планов, являющегося частью элемента сворачиваемого списка. Позволяет проигрывать аудио при нажатии на нее. |
| LibraryActivity | Класс для экрана библиотеки – содержит списки сохраненных аудиозаписей. |
| ListForRV | Вспомогательный класс для элемента сворачиваемого списка, содержит название, характеристику свернутости и название способа сортировки. |
| ListViewHolder | Класс холдер для связи объекта списка аудио и разметки для элемента списка аудио, настраивает поля с информацией в соответствии с информацией об аудио. |
| NestedListAdapter | Класс адаптер для сворачиваемых списков – его элементами являются список аудио-источников с заголовком и список аудио-результатов с заголовком, настраивает логику взаимодействия с ними, позволяет сворачивать и разворачивать оба списка. |
| ViewHolder | Класс холдер для связи объекта сворачиваемого списка и разметки элемента списка сворачиваемых списков. |

ПРИЛОЖЕНИЕ 3

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ МЕТОДОВ, ПОЛЕЙ И СВОЙСТВ**

**Описание членов класса Audio**

*Таблица 7 – Описание членов класса Audio*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| init | public | void | Int, String, String, String, Int, Date | | Конструктор класса с параметрами ID, имени, происхождения (источник или результат), url к месту хранения соответствующего аудиофайла, длительности и даты создания, задает значения полей. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| ID | public | Int | | Поле для уникального имени файла на сервере, необходимо для идентификации при работе с сервером. | |
| title | public | String | | Название аудиозаписи. | |
| source | public | String | | Происхождение аудиозаписи – источник, результат или голос – нужно для корректного отображения в списках библиотеки. | |
| url | public | String | | Поле для адреса памяти, где хранится соответствующая объекту физическая аудиозапись. Нужна для получения аудиофайла по объекту. | |
| duration | public | Int | | Длительность аудио. В сеттере осуществляется проверка на то, что она не меньше нуля. | |
| date | public | Date | | Дата создания аудио. Используется при отображении аудио в списках библиотеки для удобства и понимания пользователя. | |

**Описание членов класса User**

*Таблица 8 – Описание членов класса User*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| init | public | void | String, ArrayList<Audio>, ArrayList<Audio> | | Конструктор класса, задает имя пользователя, список аудио – сохраненных в библиотеке и список голосов. |
| toString | public | String | - | | Метод для преобразования объекта пользователя в строку, содержит информацию об основных полях: имени, сохраненным аудио и голосам. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| name | public | String | | Поле для имени пользователя. Нужно для идентификации пользователя внутри сеанса, а также для сохранения информации между сеансами для входа по сохраненным данным. | |
| audios | public | ArrayList<Audio> | | Поле списка сохраненных в библиотеке аудио пользователя. | |
| voices | public | ArrayList<Audio> | | Поле списка голосов, которые будут загружаться и отображаться на экране обработки аудио. | |
| autoSaveRes | public | Boolean | | Поле пользовательской настройки автосохранения результатов обработки аудио. Сохраняется между сеансами работы в приложении. Если оно отмечено как правда, результаты обработки аудио будут сохраняться в папку загрузок устройства автоматически. | |
| autoSaveRec | public | Boolean | | Поле пользовательской настройки автосохранения источников обработки аудио. Сохраняется между сеансами работы в приложении. Если оно отмечено как правда, источники обработки аудио будут сохраняться в папку загрузок устройства автоматически. | |

**Описание членов интерфейса AudioApiService**

*Таблица 9 – Описание членов интерфейса AudioApiService*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | *Назначение* |
| save | public | Call<Any> | MultipartBody.Part, String | Метод для отправки на сервер запроса на сохранения аудиозаписи в библиотеку. Содержит аудиозапись, преобразованную в MultipartBody.Part, и строку – токен для доступа к серверу. Возвращает результат о завершении запроса, null в случае неудачи. |
| delete | public | Call<Any> | Int, String | Метод для отправки на сервер запроса по удалению аудиозаписи из библиотеки. Передается идентификатор аудиозаписи и токен для доступа к серверу. Возвращает результат о завершении запроса, null в случае неудачи. |
| request | public | Call<Any> | MultipartBody.Part, String | Метод для отправки на сервер запроса на обработку заявки - аудиозаписи или архива как голоса. Принимает как параметры отправляемый файл в формате MultipartBody.Part и токен для доступа к серверу. Возвращает результат о завершении запроса, null в случае неудачи. |
| process | public | Call<ResultFromServer> | Int, MultipartBody.Part, String | Метод для отправки на сервер запроса на обработку аудиозаписи. Принимает параметры идентификатора голоса, файл источника и токен для доступа. Возвращает результат обработки в формате ResultFromServer. |
| authorize | public | Call<Token> | LoginRequest | Метод для запроса авторизации существующего пользователя от сервера. Принимает объект класса LoginRequest, содержащий имя пользователя и пароль. Возвращает токен в случае успешной авторизации. |
| signUp | public | Call<Any> | LoginRequest | Метод для запроса регистрации нового пользователя на сервере. Принимает объект класса LoginRequest, содержащий имя пользователя и пароль. Возвращает результат о завершении регистрации. |
| loadLibrary | public | Call<AudioListResponse<AudioFromServer>> | String | Метод для отправки на сервер запроса на загрузку библиотеки. Принимает токен пользователя для доступа к серверу и библиотеке. Возвращает объект класса AudioListResponse с типизирущим параметром аудиозаписи (источника или результата) с сервера. |
| loadVoices | public | Call<AudioListResponse<VoiceFromServer>> | String | Метод для отправки на сервер запроса на загрузку голосов. Принимает токен пользователя для доступа к серверу. Возвращает объект класса AudioListResponse с типизирущим параметром голоса с сервера. |
| downloadFileWithDynamicUrlSync | public | Call<ResponseBody> | String?, String | Метод для отправки на сервер запроса на загрузку файла. Принимает параметры url файла на сервере и токен для доступа к серверу и файлам. Возвращает файл в формате ResponseBody. |

**Описание членов класса AudioFromServer**

*Таблица 10 – Описание членов класса AudioFromServer*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| id | public | Int | Уникальный идентификатор аудио из библиотеки на сервере. Нужен для осуществления операций с аудио, например удаления, загрузки. |
| audio | public | AudioInfo | Поле типа информации об аудио, его наличие обусловлено форматом хранения информации на сервере. |
| url | public | String | Поле для url аудиофайла на сервере. Нужно для скачивания файла. |

**Описание членов класса AudioInfo**

*Таблица 11 – Описание членов класса AudioInfo*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| name | public | String | Поле имени файла. Нужно для последующего отображения аудио, полученного с сервера. |

**Описание членов класса AudioListResponse**

*Таблица 12 – Описание членов класса AudioListResponse*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| results | public | List<T> | Поле, содержащее список объектов. Существование и организация этого класса обусловлены форматом передачи данных с сервера. |

**Описание членов класса LoginRequest**

*Таблица 13 – Описание членов класса LoginRequest*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| username | public | String | Поле имени пользователя. Нужно для авторизации, регистрации на сервере. |
| password | public | String | Поле пароля пользователя. Нужно для авторизации, регистрации на сервере. |

**Описание членов класса ResultFromServer**

*Таблица 14 – Описание членов класса ResultFromServer*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| *url* | *public* | *String* | *Поле url аудиозаписи-результата на сервере. Необходимо для последующего скачивания файла с сервера.* |

**Описание членов класса Token**

*Таблица 15 – Описание членов класса Token*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| token | public | String | Токен для доступа к серверу. |

**Описание членов класса VoiceFromServer**

*Таблица 16 – Описание членов класса VoiceFromServer*

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Назначение* |
| id | public | Int | Уникальный идентификатор голоса на сервере. Нужен для осуществления операций с голосом, например окрашивания в него аудио источника. |
| name | public | String | Поле имени голоса, нужно для отображения голосов, созданных на основании файлов с сервера, в меню выбора голосов на экране обработки. |
| url | public | String | Поле для url голоса на сервере. Нужно для скачивания файла. |

**Описание членов класса Controller**

*Таблица 17 – Описание членов класса* *Controller*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| init | public | void | String | | Конструктор класса с параметром домашней директории – туда будут сохраняться файлы приложения. Задает все остальные поля. |
| readAudioNames | public | ArrayList<String> | String | | Метод для чтения имен аудиофайлов (формата mp3) из заданной в параметрах папки. Возвращает список названий. |
| makeDirectories | public | String | String, String | | Метод создания директории, принимает параметры существующей директории, куда надо поместить новую, и название новой директории. Возвращает абсолютный путь к созданной директории при успешном создании. |
| downloadAudioByURL | public | LiveData<Boolean> | String, String | | Метод создания сервиса для загрузки аудио с сервера. Принимает параметры url файла и путь, куда нужно сохранить загруженное аудио. Вызывает приватный метод загрузки, возвращает объект наблюдаемой информации с типом Boolean – успешность или неуспешность загрузки. |
| serverDownloadAudioByURL | private | void | Call<ResponseBody>, MutableLiveData<Boolean>, String | | Метод вызова сервиса загрузки аудио, принимаемого первым параметром. По выполнении запроса обновляет наблюдаемый объект успешности, принимаемый вторым параметром. |
| writeResponseBodyToDisk | private | Boolean | ResponseBody, String | | Метод записи полученного аудиофайла в память, принимает полученный файл в формате ResponseBody и путь, куда его требуется записать. Возвращает успешность сохранения аудио в память. |
| loadLibrary | public | LiveData<AudioListResponse<AudioFromServer>> | - | | Метод создания сервера для загрузки информации о файлах библиотеки, возвращает объект наблюдаемого списка аудио с сервера. |
| serverLoadLibrary | private | void | Call<AudioListResponse<AudioFromServer>>, MutableLiveData<AudioListResponse<AudioFromServer>> | | Метод вызова сервиса загрузки информации об аудио библиотеки, принимаемого первым параметром. По выполнении запроса обновляет наблюдаемый объект списка аудио с сервера, принимаемый вторым параметром. |
| loadVoices | public | LiveData<AudioListResponse<VoiceFromServer>> | - | | Метод создания сервиса для загрузки информации о файлах голосов, возвращает объект наблюдаемого списка голосов с сервера. |
| serverLoadVoices | private | void | Call<AudioListResponse<VoiceFromServer>>, MutableLiveData<AudioListResponse<VoiceFromServer>> | | Метод вызова сервиса загрузки информации о голосах, принимаемого первым параметром. По выполнении запроса обновляет наблюдаемый объект списка голосов с сервера, принимаемый вторым параметром. |
| deleteAudio | private | LiveData<Any> | Int | | Метод создания сервиса для удаления аудио из библиотеки, принимает идентификатор аудио, возвращает объект наблюдаемого результата успешности. |
| serverDeleteAudio | private | void | Call<Any>, MutableLiveData<Any> | | Метод вызова сервиса удаления аудио из библиотеки, принимаемого первым параметром. По выполнении запроса обновляет наблюдаемый объект успешности запроса, принимаемый вторым параметром. |
| signInOrUp | public | LiveData<String> | String, String, String | | Метод создания сервиса для регистрации нового пользователя или авторизации существующего, принимает имя пользователя, пароль и ключ – либо регистрации, либо авторизации. В зависимости от ключа вызывает serverSignIn или serverSignUp. Метод возвращает наблюдаемый объект токена, полученного в результате авторизации пользователя. |
| serverSignIn | private | void | Call<Token>, MutableLiveData<String> | | Метод вызова сервиса авторизации существующего пользователя, принимаемого первым параметром. По выполнении запроса обновляет наблюдаемый объект токена, принимаемый вторым параметром. |
| serverSignUp | private | void | Call<Token>, MutableLiveData<String>, LoginRequest | | Метод вызова сервиса регистрации нового пользователя, принимаемого первым параметром. По выполнении запроса обновляет наблюдаемый объект токена, принимаемый вторым параметром. Третий параметр – запрос на авторизацию, так как в случае регистрации нового пользователя сначала вызывается регистрация, затем авторизация. |
| process | public | LiveData<ResultFromServer> | Int | | Метод создания сервиса для обработки аудио на сервере, принимает идентификатор голоса, задает параметры сервиса как этот идентификатор и аудио-источник, путь к которому хранится в контроллере. Возвращает наблюдаемый объект результата с сервера, содержащий url, по которому в дальнейшем будет загружен результат. |
| serverProcess | private | void | Call<ResultFromServer>, MutableLiveData<ResultFromServer> | | Метод вызова сервиса обработки аудиозаписи, принимаемого первым параметром. По выполнении запроса обновляет наблюдаемый объект результата с сервера, принимаемый вторым параметром. |
| addToLibrary | public | LiveData<Any> | String | | Метод создания сервиса для сохранения аудио в библиотеку на сервере, принимает путь к аудиозаписи, преобразует считанное оттуда аудио в MultipartBody.Part и вызывает метод save сервиса. Возвращает наблюдаемый объект результата успешности. |
| serverAddToLibrary | private | void | Call<Any>, MutableLiveData<Any> | | Метод вызова сервиса сохранения аудио в библиотеке на сервере, принимаемого первым параметром. По выполнении запроса обновляет наблюдаемый объект результата успешности. |
| sendRequest | public | LiveData<Any> | String | | Метод создания сервиса для отпраки на сервер запроса на обработку файла как голоса. Принимает название запроса, которое автоматически содержит индикатор того, отправляется архив или аудиозапись, формирует на основании этого ключа и пути, сохраненного в контроллере MultipartBody.Part, содержащий требуемый файл, и вызывает метод отправки запроса сервиса. Возвращает наблюдаемый объект результата успешности. |
| serverRequest | private | void | Call<Any>, MutableLiveData<Any> | | Метод вызова сервиса отправки запроса на обработку голоса на сервере, принимаемого первым параметром. По выполнении запроса обновляет наблюдаемый объект результата успешности. |
| downloadAudio | public | void | String, String | | Метод для загрузки файла в папку загрузок локальной памяти устройства. Принимает параметры пути, где файл сохранен в данный момент (недоступная пользователю память) и имя, с которым файл надо сохранить в папку загрузок. Записывает файл по требуемому адресу. |
| deleteLibrary | public | Boolean | - | | Метод очистки библиотеки, вызываемый с экрана настроек. В методе осуществляется итерация по списку аудио библиотеки пользователя, для каждого аудио вызывается метод удаления аудио. Возвращается успешность удаления всех аудио. |
| deleteAudio | public | Boolean | Audio | | Метод удаления аудио из библиотеки. Происходит удаление файла из памяти, удаление объекта аудио из списка аудио пользователя и вызывается метод удаления аудио с сервера. В случае успешности всех операций возвращается булевое значение правды. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| context | public | Context? | | Поле контекста приложения для того, чтобы иметь доступ к функциям, доступным из контекста, изнутри контроллера. | |
| homePath | public | String | | Поле адреса папки приложения в памяти устройства. Задается в конструкторе и используется для сохранения файлов приложения. | |
| requestRecordingPath | public | String | | Путь к файлу- аудиозаписи запроса на обработку голоса. Используется для отправки файла на сервер для обработки. Путь заполняется в конструкторе. | |
| requestArchivePath | public | String | | Путь к файлу- архиву запроса на обработку голоса. Используется для отправки файла на сервер для обработки. Путь заполняется в конструкторе. | |
| recordingPath | public | String | | Путь, по которому в файлах приложения хранится аудио-источник. Это всегда один файл, новые записи перезаписывают этот файл. Это мотивированно экономией памяти, так как сохранять все аудио, которые пользователь записал, но, возможно, не будет использовать, было бы затратно. Используется для доступа к записи – при отображении, обработке, сохранении, добавлении в библиотеку и т.д. Путь заполняется в конструкторе. | |
| resultPath | public | String | | Путь, по которому в файлах приложения хранится аудио-результат. Это всегда один файл, новые результаты перезаписывают этот файл. Это мотивированно экономией памяти, так как сохранять все результаты, которые пользователь получил, но, возможно, не будет использовать, было бы затратно. Используется для доступа к результату – при отображении, сохранении, добавлении в библиотеку и т.д. Путь заполняется в конструкторе. | |
| voicesPath | public | String | | Путь, по которому в файлах приложения находится папка с голосами. Сюда они загружаются с сервера, и отсюда они читаются на экране обработки аудио. Путь заполняется в конструкторе. | |
| savedPath | public | String | | Путь, по которому в файлах приложения находится папка с аудиозаписями библиотеки. Сюда они загружаются с сервера, и отсюда они читаются на экране библиотеки. Путь заполняется в конструкторе. | |
| online | public | Boolean | | Поле-флаг подключения к серверу. Заполняется в конструкторе. В случае неподключенности к серверу пользователь не может пользоваться определенными функциями приложения, поэтому это поле используется для блокировки данных функций. Обновляется при авторизации (или ее отсутствии). | |
| BASE\_URL | private | String | | Объект-компаньон, хранящий IP-адрес сервера. Используется для подключения к серверу. | |
| service | private | AudioApiService? | | Поле для сервиса, реализующего интерфейс AudioApiService, используется для взаимодействия с сервером и отправки запросов, описанных в интерфейсе AudioApiService. | |
| token | public | MutableLiveData<String> | | Поле наблюдаемого объекта -строки для токена, используется при вызове сервиса по авторизации. | |
| result | public | MutableLiveData< ResultFromServer> | | Поле наблюдаемого объекта результата – аудио, получаемого от сервера при вызове метода обработки аудио. | |
| voices | public | MutableLiveData< AudioListResponse<VoiceFromServer>> | | Поле наблюдаемого объекта списка голосов, получаемого от сервера при вызове метода загрузки голосов. | |
| library | public | MutableLiveData< AudioListResponse<VoiceFromServer>> | | Поле наблюдаемого объекта списка аудио в библиотеке, получаемого от сервера при вызове метода загрузки библиотеки. | |
| requestResult | public | MutableLiveData<Any> | | Поле наблюдаемого объекта – результата вызова сервиса по отправке запроса на обработку голоса. | |
| deleteResult | public | MutableLiveData<Any> | | Поле наблюдаемого объекта – результата вызова сервиса по отправке запроса на удаление аудиозаписи из библиотеки. | |
| saveResult | public | MutableLiveData<Any> | | Поле наблюдаемого объекта – результата вызова сервиса по отправке запроса на сохранение аудиозаписи в библиотеке. | |
| downloadBody | public | MutableLiveData<Boolean> | | Поле наблюдаемого булевого объекта – успешности вызова сервиса по отправке запроса на загрузку аудиофайла. | |

**Описание членов класса AccountActivity**

*Таблица 18 – Описание членов класса AccountActivity*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | *Назначение* |
| onCreate | public | void | Bundle? | Метод задания внешнего вида экрана: он содержит кнопки для навигации на остальные экраны. |

**Описание членов класса AuthActivity**

*Таблица 19 – Описание членов класса AuthActivity*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| onCreate | public | void | Bundle? | | Метод задания внешнего вида экрана: он содержит поля для ввода логина и пароля, кнопки для гостевого входа, авторизации и регистрации пользователя. Также настраиваются основные глобальные переменные приложения, например создается контроллер, вызываются методы дальнейшей настройки элементов экрана. |
| checkOnline | private | void | - | | Метод проверки того, подключено ли приложение к серверу. В случае отсутствия подключения блокируются кнопки регистрации и авторизации и выводится сообщение о недоступности функционала. |
| setListeners | private | void | - | | Метод настройки листенеров кнопок: при нажатии кнопок входа с авторизацией вызывается соответствующий метод, при нажатии кнопки гостевого входа – происходит вызывается метод получения локально сохраненных параметров автосохранения, так как они доступны и не зарегистрированному пользователю, состояние контроллера выставляется как офлайн, далее открывается следующий экран. Настраивается нажатие на кнопку показа или скрытия пароля: полю пароля задаются необходимые модификаторы. |
| checkSaved | private | void | - | | Метод проверяет наличие сохраненного ранее сеанса работы: если пользователь на этом устройстве уже входил в аккаунт и не вышел из него, то в SharedPreferences будут сохранены его данные для входа. В случае получения этих данных поля имени пользователя и пароля заполняются ими и вызывается метод авторизации, который в случае успеха автоматически переводит пользователя на следующий экран. |
| notGuest | private | void | String | | Метод для обработки входа не в гостевом режиме: в зависимости от ключа, передаваемого в параметрах, вызывается метод либо регистрации, либо авторизации контроллера. Если запрос прошел успешно, вызывается метод авторизованного входа. Иначе выводится сообщение о неверных данных. |
| proceedAuthorized | private | void | String | | Метод продолжения в авторизованном режиме сохраняет подтвержденные данные авторизации (среди которых есть имя пользователя, пароль и токен, получаемый в параметрах метода) в SharedPreferences, вызывает метод получения сохраненных предпочтений автосохранения, затем вызывает открытие следующего экрана. |
| getAutoSavedPrefs | private | void | SharedPreferences | | Метод позволяет получить сохраненные предпочтения автосохранения пользователя через обращение к объекту пользовательских предпочтений, который задается входным параметром. При получении предпочтений ими заполняются соответствующие поля пользователя. |
| Поля | | | | | |
| Имя | Мод. Доступа | Тип | | Назначение | |
| user | public | User | | Объект класса пользователь, глобальный для всего приложения. Помогает синхронизировать информацию и везде иметь одни и те же списки, в частности аудиозаписей и голосов. | |
| controller | public | Controller | | Объект класса контроллер, глобальный для всего приложения, позволяющий синхронизировать остальную информацию, как например пути сохранения аудио. Также используется для вызова основных методов, относящихся к общей логике приложения и работе с сервером. | |
| loginText | private | EditText | | Поле для объекта разметки редактируемого текста для имени пользователя. | |
| passwordText | private | EditText | | Поле для объекта разметки редактируемого текста для пароля. Имеет модификаторы, позволяющие скрывать и открывать пароль. | |
| signInButton | private | Button | | Поле кнопки авторизации, по нажатию на нее происходит вход авторизованного пользователя. | |
| signUpButton | private | Button | | Поле кнопки регистрации нового пользователя, нажатие на которую вызывает соответствующий процесс | |
| guestButton | private | Button | | Поле кнопки гостевого входа, прикрепленной к соответствующему процессу. | |
| eyeButton | private | Button | | Кнопка глаза, позволяющая настраивать сокрытие и показ пароля. | |
| showingPassword | private | Boolean | | Булевое поле, связанное с кнопкой сокрытия пароля. | |

**Описание членов класса ProcessActivity**

*Таблица 20 – Описание членов класса ProcessActivity*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| onCreate | public | void | Bundle? | | Метод задания внешнего вида экрана: он содержит элементы для операций с аудиозаписями источником, голосом и результатом. Также вызываются методы получения разрешений, настройки листенеров, настройки меню источника и результата, получения голосов с сервера и их настройки на экране. |
| setEnabled | private | void | - | | Метод настройки активности элементов-кнопок экрана: некоторые действия не должны быть доступны в начале, например проигрывание результата до его создания, или же если нет подключения к серверу. |
| getVoices | private | void | - | | Метод вызывает метод контроллера по запросу на скачивание голосов с сервера. Если запрос успешен, вызывается метод сохранения голосов. |
| downloadAllInArray | private | void | AudioListResponse<VoiceFromServer> | | При получении запроса на скачивание голосов сервер возвращает список, в котором вложены url файлов. Данный метод проходится по этому списку, вызывая для каждого голоса метод сервера по скачиванию файла аудио и сохраняя название голоса в список голосов пользователя. Далее по этому списку голоса будут добавлены в меню выбора голоса, однако это происходит после истечения таймера, так как на физическое сохранение аудиофайлов требуется время. |
| setVoices | public | void | - | | В этом методе настраивается спиннер с выбором голоса: в него кладется список названий голосов, заполненный в прошлом методе. |
| getPermissions | private | void | - | | Метод для запроса разрешений пользователя на использование приложением микрофона и доступ к памяти устройства. Эти разрешения необходимы для работы приложения, поэтому их необходимо дать. |
| setListeners | public | void | - | | Метод настраивает листенеры на кнопки начала, приостановки записи источника, а также прикрепления источника из файлов устройства. |
| setMenuListeners | public | void | - | | Метод задает листенеры меню, связанных с источником и результатом обработки: скачивание, добавление и отправка аудио. |
| downloadAudio | public | void | String, String, String, String | | Метод, происходящий при выборе скачивания в меню источника или результата, скачивает аудиозапись в папку загрузок устройства, копируя файл из места его временного хранения с помощью вызова метода контроллера для загрузки аудио. Параметры: путь, по которому хранится файл, путь, по которому его нужно сохранить, строки для вывода сообщения об ошибке или неудаче. |
| addAudioToLibrary | private | void | String, String, String | | Метод, происходящий при выборе добавления в библиотеку в меню источника или результата, содержит параметры пути, по которому хранится файл, который надо добавить, а также строки для сообщения об успехе или неудаче. Вызывается метод контроллера по добавлению аудиозаписи, и также его метод по добавлению ее в библиотеку. |
| shareAudio | private | void | String, String, String | | Метод, происходящий при выборе отправки в меню источника или результата, содержит параметры пути, по которому хранится файл, который надо отправить, а также строки для сообщения об успехе или неудаче. Вызывается интент для отправки файла в какое-либо приложение по выбору пользователя. |
| onClick | public | void | View? | | Метод для реакции на выбор аудиофайла из файлов на устройстве, используется для прикрепления источника из существующих файлов, вызывает интент для выбора файла. |
| onActivityResult | public | void | Int, Int, Intent? | | Метод для обработки выбранного файла с устройства. В случае успешного выбора вызывается метод скачивания файла из выбранного места в память, выделенную приложению для дальнейшей работы. Параметры – код запроса, который выполнялся, код результата – успешности или неуспешности выбора, а также информация, выбранная в интенте. |
| getAudioFromStorage | private | void | Intent?, String | | Метод для сохранения источника, выбранного из существующих файлов в память приложения. Параметры – интент, содержащий путь к выбранному файлу, путь, куда нужно сохранить выбранный файл. Вызывается метод копирования и создания проигрываемого источника, где файл будет загружен в медиа проигрыватель. |
| copy | private | void | File, File | | Метод для копирования файла из директории, указанной первым параметром, в директорию, указанную вторым параметром. Осуществляет копирование через файловые потоки. |
| mediaRecorderInit | private | void | - | | Метод инициализации аудиозаписывающего объекта, задаются такие параметры как кодировка, необходимая для итогового файла, путь его сохранения и устройство записи – микрофон. |
| startRecording | private | void | - | | Метод обработки процесса записи источника, если запись не ведется, то она начинается, иначе она заканчивается и вызывается метод получения проигрываемого источника. |
| pauseRecording | private | void | - | | Метод приостановки записи медиарекордером, если запись аудио идет, иначе вызов метода продолжения записи, если она была остановлена. |
| resumeRecording | private | void | - | | Метод продолжения записи аудио, медиарекордер продолжает запись, меняется отображение кнопки записи. |
| createTimeLabel | public | String | Int | | Метод форматирования времени проигрывания: принимает число секунд и трансформирует это в строку, содержащую это число секунд в формате времени, возвращает эту строку, нужен для отображения длительности аудио и времени, которое аудио играло. |
| playRecBtnClick | public | void | View | | Метод для обработки нажатия на кнопку проигрывания источника: если плеер не играет, то он начинает играть, иначе ставится на паузу. |
| getPlayableRecording | private | void | - | | Метод для создания проигрываемого источника: к аудиозаписи прикрепляется соответствующий медиа проигрыватель, настраивается он, а также полоска с отображением длительности и этапа проигрывания аудио, настраивается реакция на ее перемотку, запускается дополнительный процесс, который обновляет текст, соответствующий текущему времени проигрывания, а также положению линии перемотки. Настраивается доступность кнопок для нажатия, также если включено автосохранение, то источник сохраняется. |
| playResBtnClick | public | void | View | | Метод для обработки нажатия на кнопку проигрывания результата: если плеер не играет, то он начинает играть, иначе ставится на паузу. |
| getPlayableResult | private | void | - | | Метод для создания проигрываемого результата: к аудиозаписи прикрепляется соответствующий медиа проигрыватель, настраивается он, а также полоска с отображением длительности и этапа проигрывания аудио, настраивается реакция на ее перемотку, запускается дополнительный процесс, который обновляет текст, соответствующий текущему времени проигрывания, а также положению линии перемотки. Настраивается доступность кнопок для нажатия, также если включено автосохранение, то результат сораняется. |
| process | public | void | View | | Метод для вызова метода контроллера по обработке источника и выбранного голоса. Если приходит результат, содержащий url для скачивания файла, то вызывается метод контроллера по запросу на скачивание файла, после скачивания вызывается метод получения проигрываемого результата. В случае неудачи на каком-либо этапе выводятся сообщения об ошибке. |
| playVoiceBtnClick | public | void | View | | Метод для обработки нажатия на кнопку проигрывания голоса: если плеер не играет, то он начинает играть, иначе ставится на паузу. |
| getPlayableVoice | private | void | - | | Метод для создания проигрываемого голоса: к аудиозаписи прикрепляется соответствующий медиа проигрыватель, запускается дополнительный процесс, который обновляет текст, соответствующий текущему времени проигрывания. |
| onItemSelected | public | void | AdapterView<\*>?, View?, Int, Long | | Метод выбора элемента из спиннера – помогает выбрать голос из списка голосов. При выборе голоса останавливает проигрыватель, если он играл (в случае выбора голоса после выбора и проигрывания другого голоса), вызывает метод получения проигрываемого голоса для нового выбранного голоса. |
| onNothingSelected | public | void | AdapterView<\*>? | | Метод для обработки ситуации отсутствия выбора, присутствует поскольку этого требует интерфейс, приписанный активити для выбора элемента. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| actionsRec | private | ImageView | | Картинка, представляющая собой кнопку меню для источника: файл можно сохранить, добавить в библиотеку или отправить. | |
| actionsRes | private | ImageView | | Картинка, представляющая собой кнопку меню для результата: файл можно сохранить, добавить в библиотеку или отправить. | |
| startRecBtn | private | Button | | Кнопка для начала записи источника с помощью микрофона. | |
| attachRecBtn | private | Button | | Кнопка для прикрепления источника с помощью выбора из уже существующих на устройстве файлов. | |
| pauseRecBtn | private | Button | | Кнопка для паузы записи аудио. | |
| voiceSpinner | private | Spinner | | Спиннер для выбора голоса, содержит список доступных голосов. | |
| processBtn | private | Button | | Кнопка для начала процесса обработки аудио. | |
| playRecBtn | public | Button | | Кнопка для проигрывания источника. | |
| playResBtn | public | Button | | Кнопка для проигрывания результата. | |
| playVoiceBtn | public | Button | | Кнопка для проигрывания выбранного голоса. | |
| positionRecBar | public | SeekBar | | Полоса для обозначения прогресса в проигрывании источника и перемотки проигрывания. | |
| positionResBar | public | SeekBar | | Полоса для обозначения прогресса в проигрывании результата и перемотки проигрывания. | |
| elapsedTimeLabelRec | public | TextView | | Текст для обозначения прошедшего времени проигрывания источника. | |
| elapsedTimeLabelRes | public | TextView | | Текст для обозначения прошедшего времени проигрывания результата. | |
| totalTimeLabelRec | private | TextView | | Текст для обозначения общей длительности источника. | |
| totalTimeLabelRes | private | TextView | | Текст для обозначения общей длительности результата. | |
| recPlayer | private | MediaPlayer? | | Медиа плеер для источника. | |
| resPlayer | private | MediaPlayer? | | Медиа плеер для результата. | |
| voicePlayer | private | MediaPlayer? | | Медиа плеер для голоса. | |
| totalTime | private | Int | | Поле для значения общей длительности источника. | |
| totalTimeRes | private | Int | | Поле для значения общей длительности результата. | |
| totalTimeVoice | private | Int | | Поле для значения общей длительности выбранного голоса. | |
| processed | private | Boolean | | Поле-флаг для хранения информации о том, произошло ли получение результата. Если поле не равно правде, некоторые элементы экрана недоступны для взаимодействия. | |
| mediaRecorder | private | MediaRecorder? | | Звукозаписывающий объект для записи источника. | |
| state | private | Boolean | | Поле для мониторинга состояния записи источника. | |
| recordingStopped | private | Boolean | | Поле для мониторинга приостановки записи источника. | |
| chosenRec | private | String? | | Поле для хранения пути до выбранного в качестве источника файла. | |
| voice | private | String | | Поле для хранения имени выбранного голоса. | |
| handler | public | Handler | | Хендлер для проигрывания источника: обновляет текст о том, сколько времени прошло с момента проигрывания, также меняет отображение кнопки проигрывания и состояние проигрывания в момент окончания аудио. | |
| handlerRes | public | Handler | | Хендлер для проигрывания результата: обновляет текст о том, сколько времени прошло с момента проигрывания, также меняет отображение кнопки проигрывания и состояние проигрывания в момент окончания аудио. | |
| handlerVoice | public | Handler | | Хендлер для проигрывания голоса: меняет отображение кнопки проигрывания и состояние проигрывания в момент окончания аудио. | |

**Описание членов класса RequestActivity**

*Таблица 21 – Описание членов класса RequestActivity*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| onCreate | public | void | Bundle? | | Метод задания внешнего вида экрана: он содержит элементы для создания данных для заявки, например записи и прикрепления аудио, прослушивания аудио, прикрепления архива, отображения название архива, ввода названия заявки и кнопки отправки. Также вызываются методы настройки активированности элементов, проверки подключения к серверу, настройки листенеров. |
| checkOnline | private | void | - | | Метод проверки того, подключено ли приложение к серверу. В случае отсутствия подключения блокируются кнопки отправки файлов и выводится сообщение о недоступности функционала. |
| mediaRecorderInit | private | void | - | | Метод инициализации аудиозаписывающего объекта, задаются такие параметры как кодировка, необходимая для итогового файла, путь его сохранения и устройство записи – микрофон. |
| onClick | public | void | View? | | Метод для реакции на выбор файла заявки из файлов на устройстве. Обрабатывает как прикрепление аудио, так и прикрепление архива в зависимости от идентификатора элемента, вызвавшего метод. Вызывает интент для выбора файла из памяти. |
| onActivityResult | public | void | Int, Int, Intent? | | Метод для обработки выбранного файла с устройства. В случае успешного выбора вызывается метод скачивания файла из выбранного места в память, выделенную приложению для дальнейшей работы. Параметры – код запроса, который выполнялся (разный для выбора аудио и архива, поэтому есть возможность обработать оба запроса в одном методе), код результата – успешности или неуспешности выбора, а также информация, выбранная в интенте. |
| getFileFromStorage | private | void | Intent?, String | | Метод для сохранения файла для заявки, выбранного из существующих файлов в память приложения. Параметры – интент, содержащий путь к выбранному файлу, путь, куда нужно сохранить выбранный файл. Вызывается метод копирования и, в случае если выбрано аудио, создания проигрываемого аудио, где файл будет загружен в медиа проигрыватель. Настраивается доступность кнопки отправки соответствующего файла. |
| copy | private | void | File, File | | Метод для копирования файла из директории, указанной первым параметром, в директорию, указанную вторым параметром. Осуществляет копирование через файловые потоки. |
| setListeners | private | void | - | | Метод настраивает листенеры на кнопки начала, приостановки записи аудио для заявки, а также прикрепления файла для заявки из файлов устройства. |
| pauseRecording | private | void | - | | Метод приостановки записи медиарекордером, если запись аудио идет, иначе вызов метода продолжения записи, если она была остановлена. |
| startRecording | private | void | - | | Метод обработки процесса записи аудио, если запись не ведется, то она начинается, иначе она заканчивается и вызывается метод получения проигрываемого аудио. |
| resumeRecording | private | void | - | | Метод продолжения записи аудио, медиарекордер продолжает запись, меняется отображение кнопки записи. |
| createTimeLabel | public | String | Int | | Метод форматирования времени проигрывания: принимает число секунд и трансформирует это в строку, содержащую это число секунд в формате времени, возвращает эту строку, нужен для отображения длительности аудио и времени, которое аудио играло. |
| playRecBtnClick | public | void | View | | Метод для обработки нажатия на кнопку проигрывания аудио: если плеер не играет, то он начинает играть, иначе ставится на паузу. |
| getPlayableRecording | private | void | - | | Метод для создания проигрываемого аудио: к аудиозаписи прикрепляется соответствующий медиа проигрыватель, настраивается он, а также полоска с отображением длительности и этапа проигрывания аудио, настраивается реакция на ее перемотку, запускается дополнительный процесс, который обновляет текст, соответствующий текущему времени проигрывания, а также положению линии перемотки. Настраивается доступность кнопок для нажатия. |
| checkName | private | Boolean | - | | Метод для проверки корректности имени заявки, возвращает результат проверки. Вызывается перед отправкой заявки. |
| sendArchive | public | void | View | | Метод вызова отправки заявки с архивом на сервер, перед отправкой проверяется корректность имени и к нему добавляется префикс с флагом архива. |
| sendAudio | public | void | - | | Метод вызова отправки заявки с аудио на сервер, перед отправкой проверяется корректность имени и к нему добавляется префикс с флагом аудио. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| sendAudio | private | Button | | Кнопка для отправки заявки с аудио. | |
| sendArchive | private | Button | | Кнопка для отправки заявки с архивом. | |
| startRecBtn | private | Button | | Кнопка для начала записи аудио для заявки с помощью микрофона. | |
| attachRecBtn | private | Button | | Кнопка для прикрепления аудио для заявки с помощью выбора из уже существующих на устройстве файлов. | |
| pauseRecBtn | private | Button | | Кнопка для паузы записи аудио. | |
| playRecBtn | public | Button | | Кнопка для проигрывания аудио. | |
| positionRecBar | public | SeekBar | | Полоса для обозначения прогресса в проигрывании аудио и перемотки проигрывания. | |
| elapsedTimeLabelRec | public | TextView | | Текст для обозначения прошедшего времени проигрывания аудио. | |
| totalTimeLabelRec | private | TextView | | Текст для обозначения общей длительности аудио. | |
| archiveName | private | TextView | | Текст для отображения названия выбранного архива. | |
| attachArchive | private | Button | | Кнопка для прикрепления архива с помощью выбора из уже существующих на устройстве файлов. | |
| requestName | private | EditText | | Поле для ввода текста названия заявки. Нужно для человекочитаемого отображения заявки при обработке ее на сервере. | |
| mediaRecorder | private | MediaRecorder? | | Звукозаписывающий объект для записи аудио. | |
| recPlayer | private | MediaPlayer? | | Медиа плеер для аудио. | |
| chosenRec | private | String? | | Поле для хранения пути до выбранного в качестве аудио для записи файла. | |
| totalTime | private | Int | | Поле для значения общей длительности аудио. | |
| state | private | Boolean | | Поле для мониторинга состояния записи аудио. | |
| recordingStopped | private | Boolean | | Поле для мониторинга приостановки записи аудио. | |
| handler | public | Handler | | Хендлер для проигрывания аудио: обновляет текст о том, сколько времени прошло с момента проигрывания, также меняет отображение кнопки проигрывания и состояние проигрывания в момент окончания аудио. | |

**Описание членов класса SettingsActivity**

*Таблица 22 – Описание членов класса SettingsActivity*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| onCreate | public | void | Bundle? | | Метод задания внешнего вида экрана: он содержит элементы для настройки приложения, а именно выхода и входа в аккаунт, настроек автосохранения, очистки библиотеки. Также вызываются методы проверки подключенности к серверу, настройки листенеров, обновление экрана, настройка выхода и входа в аккаунт. |
| checkOnline | private | void | - | | Метод проверки того, подключено ли приложение к серверу. В случае отсутствия подключения блокируются кнопки входа и выхода из аккаунта и очистки библиотеки и выводится сообщение о недоступности функционала. |
| setLogInOut | private | void | - | | Метод настройки ответа на нажатие на кнопку входа и выхода. Происходит проверка того, авторизован ли сейчас пользователь. Если нет, то кнопка отображается как «войти в аккаунт» и при нажатии перенаправляет пользователя на экран авторизации. Если же пользователь авторизован, то кнопка отвечает за выход из аккаунта. Выводится диалог о подтверждении действия с предупреждением о последствиях, в случае подтверждения вызывается метод выхода из аккаунта. |
| logOut | private | void | - | | Метод выхода из аккаунта: в сохраненных данных пользователя стираются данные для авторизации, объект пользователя меняется на аналогичный гостевому входу, выводится сообщение о выходе из аккаунта. |
| setListeners | private | void | - | | Метод настраивает листенеры на кнопки автосохранения результатов и источников. При нажатии обновляется соответствующее поле объекта пользователя, кроме этого данные записываются в сохраненные данные пользователя с целью извлечения их потом. |
| setLibraryDelete | private | void | - | | Метод очистки библиотеки. При нажатии на кнопку выводится диалог о подтверждении действия и предупреждением о последствиях, в случае подтверждения вызывается метод очистки библиотеки в контроллере. |
| refresh | private | void | - | | Метод настройки отображения экрана: происходит проверка, находится ли здесь пользователь как гость или как авторизованный пользователь, в зависимости от этого настраивается внешний вид кнопки входа и выхода и текст о статусе пользователя. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| logInOutButton | private | Button | | Кнопка для настройки входа и выхода из аккаунта. Меняет внешний вид и поведение в зависимости от того, авторизован ли пользователь в текущем сеансе. | |
| accountStatus | private | TextView | | Текст, отображающий, авторизован ли пользователь в текущем сеансе, и если да, то с каким именем. | |
| autoSaveRec | private | SwitchCompat | | Кнопка для включения или отключения автосохранения источников. | |
| autoSaveRes | private | SwitchCompat | | Кнопка для включения или отключения автосохранения результатов. | |
| deleteLibrary | private | Button | | Кнопка для удаления библиотеки. | |

**Описание членов класса AudiosListAdapter**

*Таблица 23 – Описание членов класса AudiosListAdapter*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| init | public | void | ArrayList<Audio> | | Конструктор, задающий поле аудиозаписей, отображаемых в данном списке. |
| onCreateViewHolder | public | ListViewHolder | ViewGroup,Int | | Метод для настройки прикрепления холдера отображения для каждого конкретного элемента – создается объект класса ListViewHolder, который связывает аудио в списке аудио с элементом в отображении. |
| onBindViewHolder | public | void | ListViewHolder, Int | | Настройка связывания: вызывается метод связывания элемента с аудиозаписью у холдера. |
| getItemCount | public | Int | - | | Метод получения числа элементов в отображаемом списке, возвращает длину списка аудиозаписей. |
| getItem | private | Audio | Int | | Метод получения элемента – аудиозаписи, находящейся на заданной в параметрах позиции в списке отображаемых аудиозаписей. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| audios | private | ArrayList<Audio> | | Поле для хранения списка отображаемых аудиозаписей. | |

**Описание членов класса LibraryActivity**

*Таблица 24 – Описание членов класса LibraryActivity*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| onCreate | public | void | Bundle? | | Метод задания внешнего вида экрана и основных переменных: вызывается метод получения аудио, задается разметка, вызывается метод обновления экрана с базовой сортировкой списков аудио. |
| getAudios | private | void | - | | Метод для вызова метода контроллера для получения аудиозаписей от сервера. В случае удачного получения списка url вызывается метод загрузки всех аудио. |
| downloadAllFromArray | private | void | AudioListResponse<AudioFromServer> | | Метод загрузки аудио, принимающий на вход ответ сервера, который содержит вложенные объекты аудио с сервера, содержащие url. Для каждого объекта вызывается метод контроллера по загрузке файла через url. Задаются параметры аудиозаписей, такие как происхождение, имя и т.д., аудиозаписи добавляются в список. Поле аудиозаписей в библиотеке объекта пользователя задается полученным списком. |
| setUp | private | void | - | | Метод для настройки отображения элементов: так как экран содержит два списка аудио – результаты и источники, они сами находятся в списке rvList, для которого задается адаптер nestedListAdapter. Задается спиннер с выбором сортировок элементов в списках: их можно сортировать по названию, длительности и дате создания. |
| onItemSelected | public | void | AdapterView<\*>?, View?, Int, Long | | Метод для реализации интерфейса выбора элемента спиннера. Используется для выбора сортировки списков. При выборе вызывается метод обновления внешнего вида экрана с передаваемым параметром выбранной сортировки. |
| onNothingSelected | public | void | AdapterView<\*>? | | В случае отсутствия выбора сортировки сортировка осуществляется по названию аудиозаписей. |
| refresh | private | void | String | | Метод обновления внешнего вида списков настраивает содержимое списка вложенных списков и сортировки объектов внутри них. Принимается на вход строка с выбранной сортировкой. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| nestedListAdapter | private | NestedListAdapter | | Адаптер для списка вложенных списков, связывает элемент – список аудиозаписей, источников либо результатов, с списком аудио. | |
| rvList | private | RecyclerView | | Отображение списка вложенных списков, содержит список сохраненных результатов и список сохраненных источников. | |
| sortSpinner | private | Spinner | | Спиннер с сортировками, позволяющий выбрать в связи с чем сортировать аудиозаписи в библиотеке. | |
| listList | private | ArrayList<ListForRV> | | Список вспомогательных элементов с названиями вложенных списков, значениями их развернутости и сортировками, применяемыми внутри них. | |

**Описание членов класса ListForRV**

*Таблица 25 – Описание членов класса ListForRV*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| init | public | void | String, Boolean, String | | Конструктор, задающий значения полей вспомогательного объекта вложенного списка: название, характеристику развернутости и сортировку. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| name | public | String | | Поле названия списка списков. Например, у списка аудиозаписей-результатов будет равно «Сохраненные результаты». Используется для отображения списка на экране. | |
| expand | public | Boolean | | Поле для статуса отображения списка: он может быть свернут или развернут. | |
| sort | public | String | | Поле для хранения ключа-названия сортировки, применяемой к объектам внутри списка. Нужно для сортировки аудиозаписей по некоторым их параметрам. | |

**Описание членов класса ListViewHolder**

*Таблица 26– Описание членов класса ListViewHolder*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| init | public | void | View, AudiosListAdapter, ArrayList<Audio> | | Конструктор, задающий поля родительского адаптера и списка аудиозаписей. |
| createTimeLabel | public | String | Int | | Метод форматирования времени проигрывания: принимает число секунд и трансформирует это в строку, содержащую это число секунд в формате времени, возвращает эту строку, нужен для отображения длительности аудио и времени, которое аудио играло. |
| bind | public | void | Audio | | Метод связывания отображения элемента списка с конкретным аудио, принимаемым как параметр. Поля отображения, такие как название, длительность и т.д. настраиваются в соответстии с полями аудио. Вызывается метод создания проигрываемого аудио и настройка листенеров для кнопки меню аудио. |
| setMenuListeners | private | void | - | | Метод задает листенеры меню, связанных с аудио в списке аудио: элемент в библиотеке можно удалить из библиотеки, скачать или отправить. |
| deleteAudio | private | void | - | | Метод удаления аудио из библиотеки. Вызывается метод контроллера по удалению аудио из библиотеки, в случае удачного удаления удаляется элемент из локального списка аудио, находящегося в холдере, уведомляется родительский адаптер с целью не отображать удаленный элемент. |
| downloadAudio | private | void | - | | Метод, происходящий при выборе скачивания в меню аудио в библиотеке, скачивает аудиозапись в папку загрузок устройства, с помощью вызова метода контроллера для загрузки аудио. |
| shareAudio | private | void | - | | Метод, происходящий при выборе отправки в меню аудио в библиотеке. Вызывается интент для отправки файла в какое-либо приложение по выбору пользователя. |
| playBtnClick | private | void | View | | Метод для обработки нажатия на кнопку проигрывания аудио в библиотеке: если плеер не играет, то он начинает играть, иначе ставится на паузу. |
| getPlayableAudio | private | void | - | | Метод для создания проигрываемого аудио из элемента в списке библиотеки: к аудиозаписи прикрепляется медиа проигрыватель, настраивается полоска с отображением длительности и этапа проигрывания аудио, настраивается реакция на ее перемотку, запускается процесс, обновляющий текст, соответствующий текущему времени проигрывания, а также положению линии перемотки. Настраивается доступность кнопки проигрывания. |
| setUpSeekBar | private | void | - | | Метод настройки полосы с отображением проигрывания аудио, при изменении положения трекера аудиозапись перематывается. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| parentAdapter | private | AudiosListAdapter | | Поле родительского адаптера. Необходимо хранить для уведомления его контента об изменении списка элементов в случае удаления аудио. | |
| audios | private | ArrayList<Audio> | | Список аудиозаписей в родительском списке. Нужен для изменения в случае удаления аудио из библиотеки для сохранения актуальности информации. | |
| audioPlayer | private | MediaPlayer? | | Медиа плеер для элемента в списке библиотеки. | |
| audio | public | Audio | | Аудиозапись в библиотеке, к которой прикреплен элемент в списке на экране библиотеки. | |
| positionBar | public | SeekBar | | Полоса для обозначения прогресса в проигрывании аудио и перемотки проигрывания. | |
| playBtn | public | Button | | Кнопка для проигрывания аудио. | |
| elapsedTimeLabel | public | TextView | | Текст для обозначения прошедшего времени проигрывания аудио. | |
| totalTime | private | Int | | Поле для значения общей длительности аудио. | |
| totalTimeLabel | private | TextView | | Текст для обозначения общей длительности аудио. | |
| menuButton | private | ImageView | | Кнопка для меню действий с аудио в библиотеке: удаление, загрузка и отправка. | |
| handlerAudio | public | Handler | | Хендлер для проигрывания аудио: обновляет текст о том, сколько времени прошло с момента проигрывания, также меняет отображение кнопки проигрывания и состояние проигрывания в момент окончания аудио. | |

**Описание членов класса NestedListAdapter**

*Таблица 27 – Описание членов класса NestedListAdapter*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| init | public | void | List<ListForRV> | | Конструктор, задающий значение полю списка вспомогательных объектов вложенных списков. |
| onCreateViewHolder | public | ViewHolder | ViewGroup, Int | | Метод создания вью холдера для каждого элемента списка списков, возвращает холдер. |
| onBindViewHolder | public | void | ViewHolder, Int | | Метод для связывания элемента- списка аудио с адаптером и вью холдером. Настраивается соответствие между элементом – отображением и элементом списка вспомогательных элементов. Название, свернутость и сортировка выставляются соответственно прикрепленному объекту. Настраивается поведение кнопки скрытия списка. |
| setUpArray | private | void | ListForRV, Int | | Метод для задания списка аудио внутри элемента списка списков: если название списка соответствует сохраненным записям, то его внутренний список аудио для отображения приравнивается к сохраненным источникам, иначе – результатам. Настраивается связь с сортировкой – если она высталена как поле в вспомогательном элементе и локальном списке, то адаптер списка аудио должен сортировать элементы по соответствующему правилу. |
| getItemCount | public | Int | - | | Метод для получения числа элементов в отображаемом списке, возвращает число элементов в списке списков – в случае данного приложения всегда будет 2 списка, сохраненные источники и результаты. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| listList | private | List<ListForRV> | | Список списков, то есть отображаемых элементов, которые далее заполняются соответственно сохраненными в библиотеку источниками и результатами. | |
| thisAdapter | private | NestedListAdapter? | | Поле для хранения текущего адаптера, нужно для его прикрепления. | |
| adapter | private | AudiosListAdapter? | | Поле для адаптера вложенного списка. Нужно для связи с внутренним списком аудио и настройки его отображения. | |

**Описание членов класса ViewHolder**

*Таблица 28 – Описание членов класса ViewHolder*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | *Аргументы* | | *Назначение* |
| init | public | void | ListItemBinding | | Конструктор, заполняющий переменную связи. |
| **Поля** | | | | | |
| *Имя* | *Мод. Доступа* | *Тип* | | *Назначение* | |
| binding | public | ListItemBinding | | Поле для связи с прикрепленным списком, осуществляется с помощью автоматического байндинга с списком-элементом. | |

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | Новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |