МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Кафедра IT-инжиниринг

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой
PhD, доцент
_____ Т.С. Картбаев
«____» _____ 2019 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка мобильного приложения “Music Player” на платформе Android

Специальность 5В060200 – «Информатика»

Выполнил Медет Е.Е. Группа Инф-15-2
Научный руководитель к.ф.м.н., доцент _____ Калижанова А.У.
«27» 05 2019 г.

Консультанты:
по экономической части: к.э.н., доцент _____ А.И. Бекишева
«13» 05 2019 г.

по безопасности жизнедеятельности: ст. преп. _____ Ш.Ш.Бекбасаров
«15» 05 2019 г.

по применению
программного обеспечения: ст. преп. _____ М.Н. Майкотов
«20» 05 2019 г.

Нормоконтролер: ст. преп. _____ Ж.К. Алимсейтова
«30» 05 2019 г.

Рецензент: к.т.н., доцент _____ Ескендирова Д.М.
«____» _____ 2019 г.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В060200 – «Информатика»

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту Медет Еркебулан Ерканатулы

Тема проекта: Разработка мобильного приложения "Music Player" на платформе Android

Утверждена приказом по университету № 33 от «01» 03 2019 г.

Срок сдачи законченного проекта «31» 05 2019 г.

Исходные данные к проекту (эксплуатационно-технические данные, техническое задание): Android Studio, Android SDK, язык программирования Xml.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

а) Изучение области музыкальных приложений для мобильных устройств;

б) Рассмотрение вопросов о разработке программ для операционной системы Android и использовать их для реализации приложения;

в) вопросы безопасности жизнедеятельности и охраны труда;

г) экономическая эффективность работ по стандартизации.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 47 иллюстрации.

Основная рекомендуемая литература:

1. Android Development Tools for Eclipse. [Электронный ресурс] URL: https://marketplace.eclipse.org/content/android-development-tools-eclipse (дата обращения: 12.03.2017).
2. Robolectric. [Электронный ресурс] URL: http://robolectric.org/ (дата обращения: 16.05.2017).

3. Hibernate ORM 5.2.9.Final User Guide. [Электронный ресурс] https://docs.jboss.org/hibernate/orm/current/userguide/html_single/Hibernate_User_Guide.html (дата обращения: 12.04.2017).
4. Mockito. [Электронный ресурс] URL: http://site.mockito.org/ (дата обращения: 16.05.2017).

Консультации по работе с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Бекишева А.И.	01.04.19-15.05.19	
Безопасности жизнедеятельности	Бекбасаров Ш.Ш.	01.04.19-15.05.19	
Программная часть	Майкотов М.М.	01.04.19-15.05.19	
Нормоконтролер	Алимсейтова Ж.К.	01.04.19-13.05.19	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Сбор информации о предметной области проекта	01.11.18-10.12.18	Выполнено
Проектирование	11.12.18-18.02.19	Выполнено
Реализация проекта	19.02.19-15.04.19	Выполнено
Тестирование	16.04.19-15.05.19	Выполнено
Внедрение	16.05.19-20.05.19	Выполнено

Дата выдачи задания «26» октября 2018 г.

Заведующий кафедрой _____ Т.С. Картбаев

Научный руководитель проекта _____ А.У. Калижанова

Задание принял к исполнению студент _____ Е.Е. Медет

Аңдатпа

Дипломдық жобаның мақсаты – мобильді құрылғыдан музыкалық файлдардың негізгі форматын ойнатуға мүмкіндік беретін веб-қосымшаны әзірлеу. Дипломдық жобада сақтау кеңістігін көру және сұрастыру бойынша функциялар, сондай-ақ, mp3 файлдарды қосу, жою және ойнату бойынша операциялар орындалды. Дамыған мобильді қосымшасы ыңғайлы және үнемді.

Қазіргі уақытта Android қолданбасы негізінде музыкалық ойнатқыштар нарықта танымал. Android операциялық жүйесінің түпкілікті дамуы әзірлеушілерге танымал компьютерлік технологияларды игерген білімдермен үйлесетін және мобильді қосымшаны дамытуға үйретететін тамаша платформамен қамтамасыз етеді

Аннотация

Целью данного дипломного проекта является разработка веб-приложения, которое позволит воспроизводить основной формат музыкальных файлов с мобильного устройства. В дипломном проекте реализованы функции для просмотра и запроса пространства хранения данных, а также операции добавления, удаления и воспроизведения mp3 файлов. Разработанное мобильное приложение удобно в использовании и экономично.

Музыкальный плеер на базе приложения Android в настоящее время популярен на рынке. Завершающая разработка операционной системы Android предоставляет разработчикам прекрасную платформу, которая может освоить популярные компьютерные технологии в сочетании с приобретенными знаниями и освоить разработку мобильных приложении.

Annotation

The purpose of this graduation project is to develop a web application that will allow you to play the main format of music files from a mobile device. In the thesis project implemented functions for viewing and requesting storage space, as well as operations for adding, deleting and playing mp3 files. The developed mobile application is convenient to use and economical.

Music player based on the Android application is currently popular in the market. The final development of the Android operating system provides developers with an excellent platform that can master popular computer technologies combined with acquired knowledge and master the development of a mobile application.

Содержание

Введение

С ростом числа людей, имеющих доступ к Интернету через смартфоны и планшеты, разработка мобильных приложений имеет уникальную возможность получить доступ к большому количеству потенциальных потребителей. Согласно интернет-проекту PewResearch, около 67 процентов владельцев смартфонов в США ежедневно используют свои смартфоны для доступа в Интернет. Недавние исследования также показывают, что к 2017 году загрузка приложений увеличится до 200 миллиардов, а последующие доходы от мобильных приложений увеличатся до 63,5 миллиардов долларов. Причина этих исключительных цифр заключается в продолжающемся росте продаж смартфонов и планшетов. Увеличились не только продажи смартфонов и планшетов, но и количество установленных мобильных приложений в геометрической прогрессии. Интернет-проект PewResearch указывает, что примерно у 50 процентов всех пользователей смартфонов установлены мобильные приложения; из этого процента две трети людей являются обычными пользователями мобильных приложений. Эти статистические данные показывают, что мобильные приложения имеют уникальную возможность взаимодействовать с клиентами совершенно нового типа, которые постоянно подключены к Интернету и глобальному коммерческому пространству. По сути, мобильное приложение позволяет вам иметь миллионы новых клиентов в ваших руках. Все, что вам остается сделать, – это разработать эффективное приложение и пожинать плоды своих трудов.

Мобильные приложения имеют преимущество перед веб-решениями, потому что они более трансформируемы, чем веб-, и растут гораздо быстрее. В сфере разработки мобильных приложений доминируют смартфоны, а среди смартфонов 84,7% составляют устройства на базе Android. Мы наблюдаем массовое распространение использования мобильных приложений, особенно в развивающихся странах. Но компаниям нужно будет иметь стратегию разрабо тки мобильных приложений, чтобы их выход на эти рынки происходил довольно быстро и имелся масштабируемый план. Развивающиеся рынки сильно отличаются от развитых, потому что существуют очевидные различия в социальной сфере. Экономическая и психологическая установка.

В этом сценарии необходима правильная стратегия разработки мобильных приложений, включающая следующие пункты: основная причина выбора платформы разработки приложений для Android по сравнению с iOS - более низкая стоимость устройств. Недавно мы видели только появление смартфона на базе Android от Google. Существуют и другие причины выбора устройств, поскольку ОС позволяет пользователю выбрать собственный уровень настройки. Мы также можем упомянуть здесь о многозадачности, предлагаемой некоторыми из этих устройств, по сравнению с устройствами на базе iOS. Более того, мы видим, что эта платформа действительно развивалась намного быстрее по сравнению с конкурентами. Прежде всего, мобильные

приложения на базе Android должны быть запущены в магазине Google Play. Исследования в этом отношении очень важны, так как они направляют процесс разработки приложений с ценными качественными и количественными исследованиями, выясняя причины загрузки приложений. Эти приложения всегда продаются, имея в виду целевую аудиторию.

В пользовательском сообщении в Google Play Store объясняется, что представляет собой приложение, а также дается призыв к действию. Мы должны иметь стратегию разработки итеративного маркетингового приложения, исследуя то, что лучше всего подходит для маркетинга вашего приложения. Не менее важно повысить узнаваемость бренда для мобильных приложений, используя такие ресурсы, как видео, значки приложений, функции и рекламную графику, а также описательные скриншоты.

Локализация приложений – еще один важный аспект маркетинга мобильных приложений и залог их успеха. Локализация поможет повысить органический рост, а также показатели конверсии рекламы. Большинство успешных компаний выбирают локализацию довольно рано на этапе разработки приложения.

В мобильной рекламе наблюдается рост числа приложений. Реклама в приложении - это будущее мобильной рекламы. Реклама в приложении предназначена для предоставления лучших возможностей для целевой аудитории в нужное время. Более того. Маркетинг в приложениях становится более эффективным, поскольку данные доступны и их можно использовать во всех точках экосистемы, что позволяет маркетологу стать более целенаправленным. На этой платформе есть много приложений, которые доступны бесплатно, но расширенные функции доступны за плату пользователя.

В таких густонаселенных странах, как Индия и Китай, бесплатные приложения всегда будут предпочтительнее для увлеченных пользователей по сравнению с платными приложениями. Большинство приложений на Android бесплатны по своей природе. Иногда те же приложения, которые перенесены с iOS на Android, появляются бесплатно. Реклама в приложении работает намного лучше, потому что она основана на данных о местоположении, что полезно для расширения контекста и привлечения клиентов к мобильному приложению.

При создании компании по разработке мобильных приложений необходимо выбрать маркетинговый план, который будет стимулировать пользователей наиболее экономичным способом. Компании-разработчики приложений для Android располагают огромной потенциальной клиентской базой и огромным количеством пользователей смартфонов в развивающихся странах. Тем не менее, решение зависит только от них, будут ли они выбирать эту услугу.

Appinventiv – одна из крупнейших компаний-разработчиков мобильных приложений, которая за два года добилась выдающихся темпов роста. Они много работали в различных сферах, таких как ресторан, здравоохранение,

обучение, развлечения и многое другое. Они верят в предоставление ориентированных на пользователя приложений для своих клиентов, используя передовые технологии.

Android – это операционная система для мобильных телефонов с открытым исходным кодом, выпущенная компанией Google в ноябре 2007 года. Ее появление сломало традиционную закрытую операционную систему для мобильных телефонов. Любой может модифицировать операционную систему мобильного телефона, а также функционировать в соответствии с личными предпочтениями, что также является наиболее привлекательным достоинством Android.

Музыкальный проигрыватель в этом дипломном проекте представляет собой прикладное программное обеспечение на основе Google Android. Приложение Android на мобильных терминалах также полностью нарушило традиционное понимание мобильных терминалов.

И ценить музыку – один из лучших способов облегчить давление в стрессовой жизни современного общества. Таким образом, многие виды мобильных телефонов игроков также разработаны. Тем не менее, многие игроки уделяют внимание внешнему виду и функциям, в то время как ресурсы мобильного телефона пользователя тратятся впустую, такие как большой объем требуемой памяти и ЦП, что доставляет массу неудобств, поскольку одновременно работают несколько программ. Для самых обычных пользователей многие функции бесполезны.

Целью данного дипломного проекта является разработка проигрывател, который может воспроизводить основной формат музыкальных файлов. Реализованы функции для просмотра и запроса пространства хранения данных, а также операции добавления, удаления и воспроизведения. Между тем, это программное обеспечение может воспроизводить, приостанавливать и выбирать песни с последними Btn и следующими Btn в соответствии с требованиями пользователя, а также менять порядок песен и т. д.

Музыкальный плеер на базе приложения Android в настоящее время популярен на рынке. Завершающая разработка операционной системы Android предоставляет разработчикам прекрасную платформу, которая может освоить популярные компьютерные технологии в сочетании с приобретенным и знаниями и освоить разработку мобильных приложении.

1 Обзор проигрывателей Android

1.1 Лучшие музыкальные проигрыватели Android в 2018 году

Одна из самых удивительных вещей об Android – это его гибкость. И это распространяется на приложения музыкального плеера. Буквально сотни приложений в магазине Google Play для аудиофилов на выбор. Эти приложения идеально подходят для тех, кто хочет настроить прослушивание, хранение и другие параметры. Мы сузили список до десяти лучших бесплатных музыкальных плееров для Android в 2018 году. Прежде чем перейти к нашему списку, мы поговорим о встроенном музыкальном проигрывателе в вашем телефоне Android. Музыка Google Play имеет традиционные функции, в которых вы можете создавать плейлисты, перемешивать песни и т. Д. Он также предлагает потоковую передачу в облаке. И что самое приятное, он уже установлен на вашем телефоне. Но зачем вам стороннее приложение? Ну, по большей части, Google Play Music имеет ограниченные возможности. Этого достаточно для любителей обычной музыки, но не для хардкорных. Некоторым людям также нужны такие функции, как эквалайзеры, тегирование файлов, просмотр папок, интерфейс кулера и т.д. Однако эти функции доступны только в сторонних приложениях. Здесь мы перечисляем десять лучших музыкальных проигрывателей в соответствии с их функциями и интерфейсами. Являетесь ли вы случайным слушателем или аудиофилом, вы можете найти тот, который подходит вам лучше всего [2].

1.1.1 Musicolet

Если вам нужен простой в использовании музыкальный проигрыватель с удобными вкладками для доступа к альбомам, спискам воспроизведения и исполнителям, Musicolet – ваш выбор. Он имеет традиционные функции, которые включают поддержку текстов песен, редактор тегов, таймер сна, виджеты и многое другое. Пользователи говорят, что они рекомендуют это приложение с точки зрения его функциональности, которая приносит впечатляющие музыкальные впечатления (Рисунок 1.1).

Другие лучшие и уникальные функции этого приложения, которые мы любим, включают в себя:

- свободный от рекламы;
- легкий;
- позволяет управлять приложением с помощью кнопок на наушнике;
- поддерживает несколько игровых очередей;
- интуитивно понятный графический интерфейс пользователя.

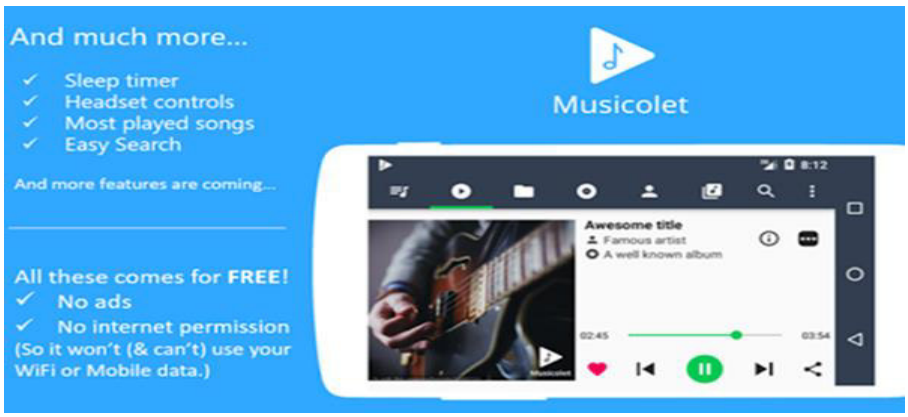

Рисунок 1.1 – Вид музыкального плеера Musicolet

1.1.2 Фонограф

Если вы любите получать приложения с отличными визуальными эффектами, Phonograph подойдет вам по вкусу. Он поставляется с пользовательским интерфейсом, который изменяется в соответствии с цветом содержимого на экране. Вы также можете выбрать понравившуюся тему из движка тем и настроить музыкальный проигрыватель (Рисунок 1.2).

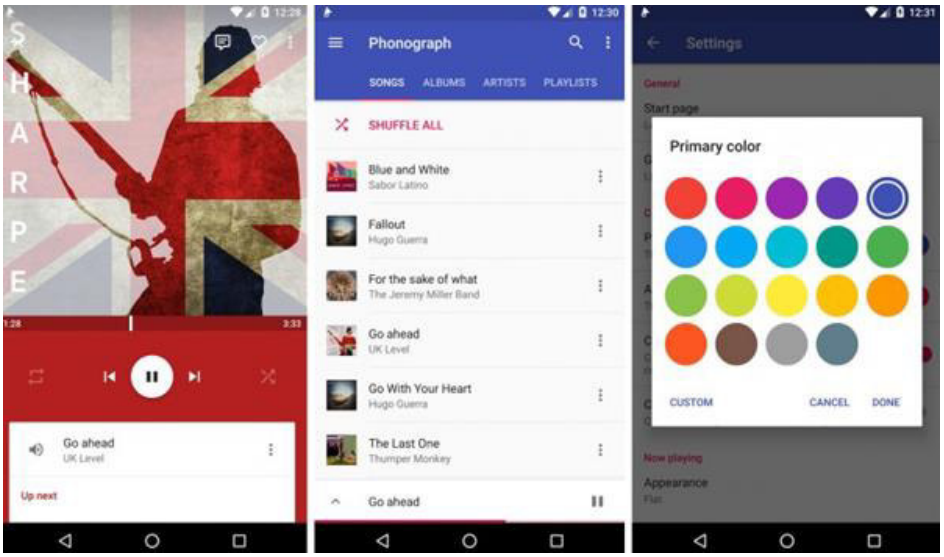

Рисунок 1.2 – Вид музыкального плеера Phonograph

Что нам больше всего нравится в этом лучшем аудиоплеере для Android, так это в том, что он автоматически загружает информацию, которая отсутствует в ваших медиафайлах. Например, название или исполнитель и даже обложки альбомов. Вам также предоставляется возможность этой информации. Мы из тех, кто очень конкретно относится к кредитам, особенно к таким произведениям искусства, как музыка. Вы можете просматривать свою музыку, используя песни, исполнителей, альбомы или плейлисты. Вы также можете сгруппировать их в папки. Фонограф позволяет вам варианты

для легкого доступа и хранения. Если вы достаточно щедры, чтобы пролить немного денег на дополнительные функции. Фонограф имеет дополнительные функции, включая управление экраном блокировки, воспроизведение без пауз и таймер отключения, которые можно приобрести [3].

1.1.3 Пульсар

Pulsar – это предпочтительное приложение для воспроизведения музыки для людей, которым нравятся легкие приложения (и которым не нравится иметь дополнительное место в вашей памяти). Несмотря на то, что это бесплатное и легкое приложение, оно также предлагает множество возможностей - от настройки до простоты использования, поскольку оно не содержит рекламы. Это простое приложение, но вы можете выбрать разные цвета, чтобы персонализировать его интерфейс (Рисунок 1.3).

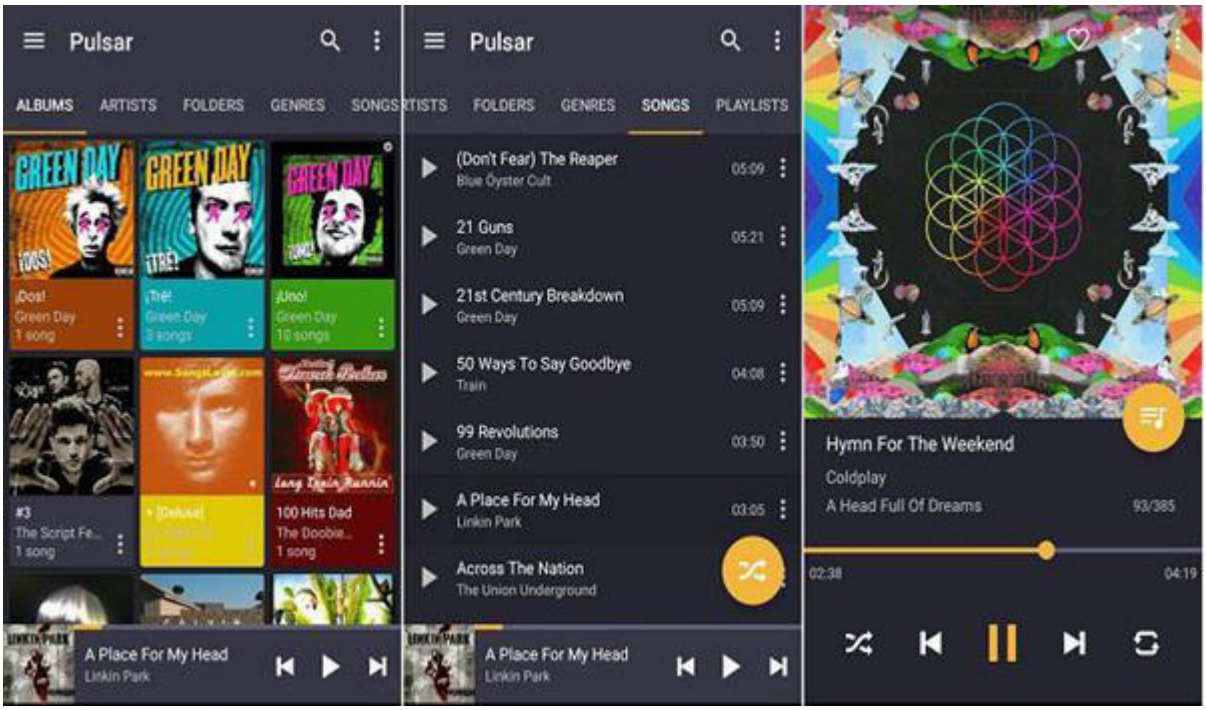

Рисунок 1.3 – Вид музыкального плеера Pulsar

Когда дело доходит до сортировки вашей музыкальной библиотеки, Pulsar позволяет просматривать вашу коллекцию по альбомам, жанрам или исполнителям. Вы также можете создавать папки, если вы предпочитаете сортировать их таким образом. Что нам нравится в Pulsar, так это то, что нам не нужно приобретать его дополнительные функции, которые включают воспроизведение без пауз, поддержку Chrome Cast, виджет домашнего экрана, встроенный редактор тегов. Для аудиофилов в Pro версии доступна опция 5-полосного эквалайзера.

1.1.4 Pi Music Player

Пи не только красива; это также идет с большим количеством особенностей. После установки у вас будет возможность выбрать тему. Хотя вы можете изменить это позже. Его интерфейс довольно прост, но он предлагает столько же функций, что и другие приложения для музыкальных плееров Android. Вы можете воспроизводить музыку из своей библиотеки, используя различные способы сортировки. Если вы хотите играть на исполнителя или для плейлиста. Вы также можете создать папку всех ваших любимых синглов (Рисунок 1.4).

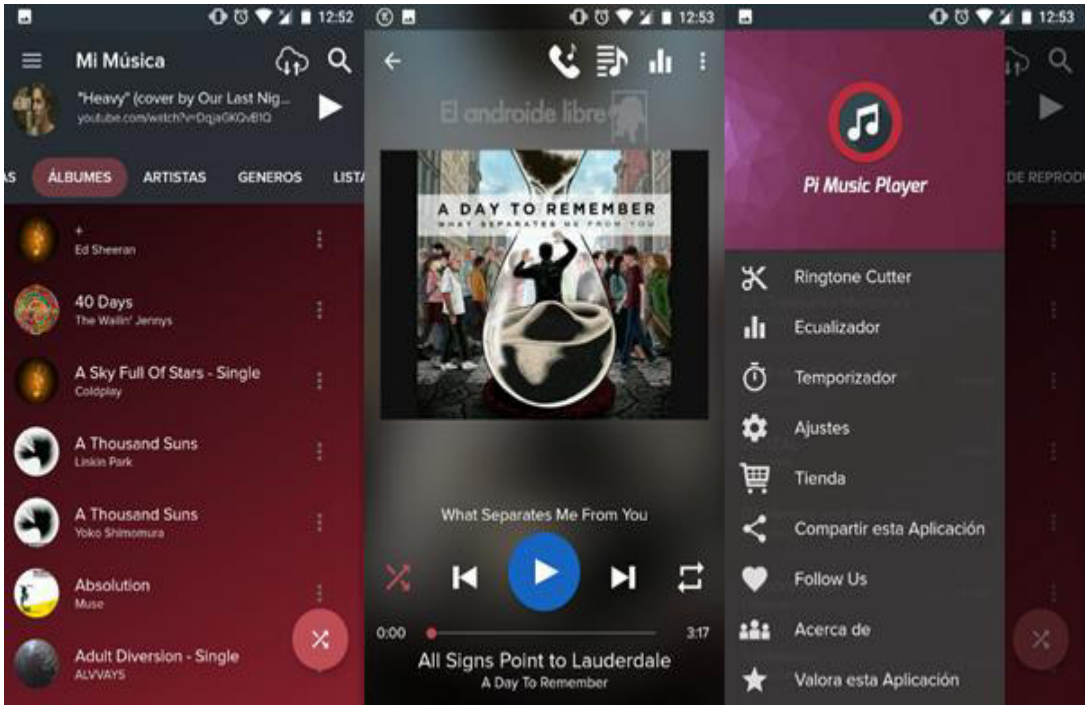

Рисунок 1.4 – Вид музыкального плеера Pi Music Player

Простота этого приложения относится только к его интерфейсу, функция мудрая; он охватывает таймер сна, поддержку виджетов и создание рингтонов. Но что нам действительно нравится в этом приложении, так это легкое приложение, оно уже включает встроенный эквалайзер, который дает вам возможность прослушивать басы, 3D-реверберацию и другие пресеты.

Еще одна интересная особенность этого приложения - оно позволяет вам делиться музыкой с друзьями. Что всегда весело. Слушать музыку с нашими друзьями - это то, что нам нравится. Наша единственная проблема с этим приложением - мы хотим использовать его без рекламы. Существует версия без рекламы, которая доступна для покупки.

Простота этого приложения относится только к его интерфейсу, функция мудрая; он охватывает таймер сна, поддержку виджетов и создание рингтонов. Но что нам действительно нравится в этом приложении, так это легкое приложение, оно уже включает встроенный эквалайзер.

1.1.5 BlackPlayer

Что касается функций, BlackPlayer предлагает одно из самых доступных предложений. Это почти как наивысший уровень возможностей аудиоплеера. Вот некоторые из них (Рисунок 1.5).

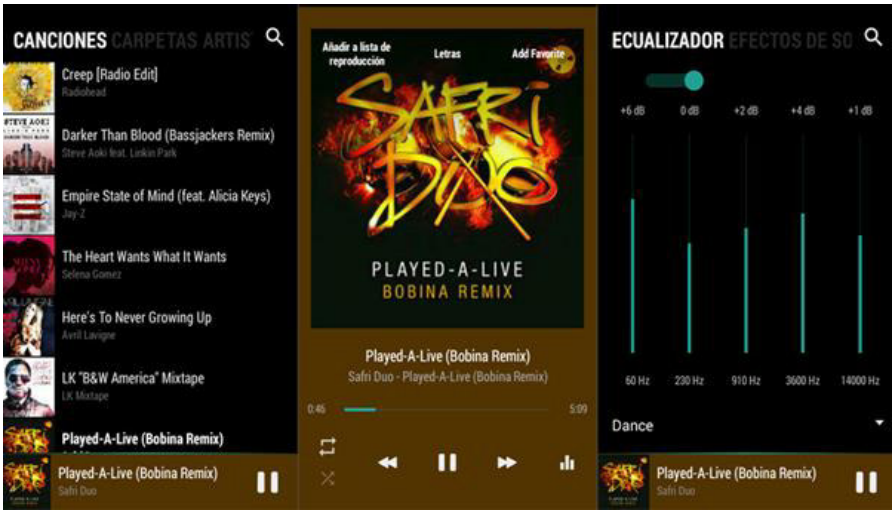

Рисунок 1.5 – Вид музыкального плеера BlackPlayer

Можно настроить свой пользовательский интерфейс, включая цвета и шрифты:

– встроенный полосный эквалайзер;
– усиление басов и виртуализатор;
– виджеты;
– бездорожное воспроизведение;
– редактор тегов ID3;
– таймер сна;
– поддерживает стандартные локальные форматы музыкальных файлов, такие как MP3, WAV и OGG.

С функциями, перечисленными выше, нам почти не нужно приобретать профессиональную версию. Даже бесплатная версия без рекламы [4].

1.1.6 n7player

Если вы ищете самое инновационное приложение для аудиоплеера для Android, n7player - это то, где ваши деньги должны быть. Да, это приложение доступно для покупки, но оно позволяет вам загрузить пробную версию, которая длится 14 дней (Рисунок 1.6).

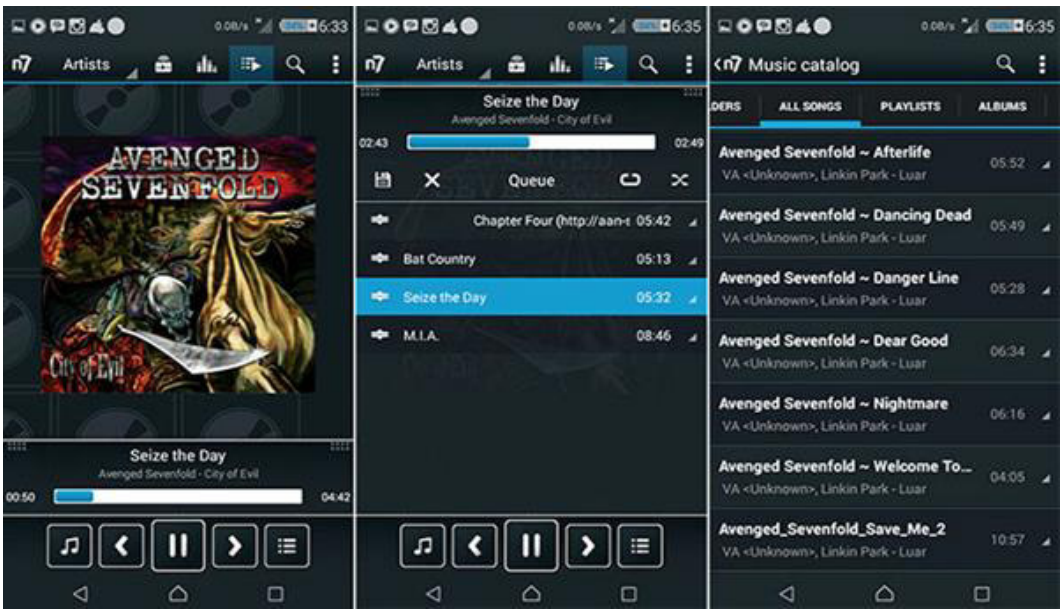

Рисунок 1.6 – Вид музыкального плеера n7player

Самый важный вопрос, стоит ли это приложение наших денег? Ну, это может быть. У него очень стильный интерфейс. Мы увеличиваете ми уменьшаем масштаб, чтобы искать нашу музыку. Мы можем организовать свою библиотеку с различными взглядами. Помимо стильного интерфейса мы получаем продвинутый 10-полосный эквалайзер. Если мы фанаты хардкорной музыки, мы можем настроить музыку так, как хотим ее слушать. Другие функции, которые он включает в себя:

– бездорожное воспроизведение;
– усиление басов и звуковые эффекты виртуализации;
– редактор тегов;
– настраиваемые темы;
– таймер сна;
– виджеты.

1.1.7 MediaMonkey

Еще одно функциональное приложение MediaMonkey. Это приложение не исчерпаны способы, чтобы позволить вам просматривать музыку в вашей библиотеке. Вы можете просматривать по альбомам, аудиокнигам, подкастам, артистам, трекам, жанрам и даже композиторам. Нам даже не нужно беспокоиться о поиске песен в вашей коллекции. Приложение также позволяет редактировать теги песен и добавлять обложку альбома. Функция поиска удобна, так как она точно предсказывает песню, которую вы ищете, и даже отображает исполнителя и название. Бесплатная версия поставляется с пятиполосным эквалайзером со стерео балансом (Рисунок 1.7).

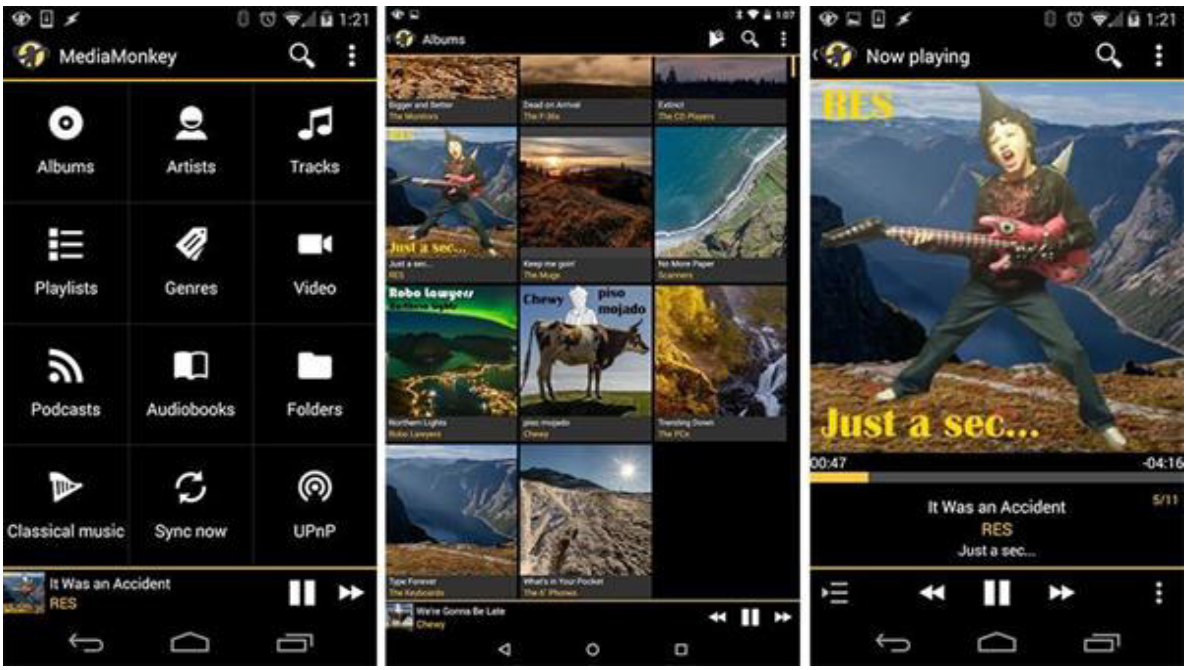

Рисунок 1.7 – Вид музыкального плеера MediaMonkey

Что нам нравится в MediaMonkey, так это поддержка Android Auto. Это означает, что вы можете синхронизировать ваш Android-плеер с MediaMonkey для Windows. Если вам нужно посмотреть на строку поиска песни, вы можете включить ее на панели уведомлений.

1.1.8 Musixmatch

Musixmatch – музыкальный проигрыватель людей, которые любят петь. Это приложение дает вам текст через плавающий виджет текстов песен. Мы даже можем синхронизировать текст в режиме реального времени. Также можно просмотреть тексты песен при использовании Spotify, Youtube, Apple Music, SoundCloud, Google Play Music. По сути, приложение может определить текст любой песни, играющей в фоновом режиме. Нам не нужно будет снова придумывать свою песню к песне.

Приложение загружено с теми же функциями, которые вы можете получить в любом другом приложении музыкального плеера для Android. Другие функции приложения включают в себя поиск треков, используя название, исполнителя. И поскольку он основан на лирике, вы также можете искать, используя одну строку текста (Рисунок 1.8). Излишне говорить, что мы любим это приложение из-за добавленной лирики, потому что мы любим петь вместе с песнями [5].

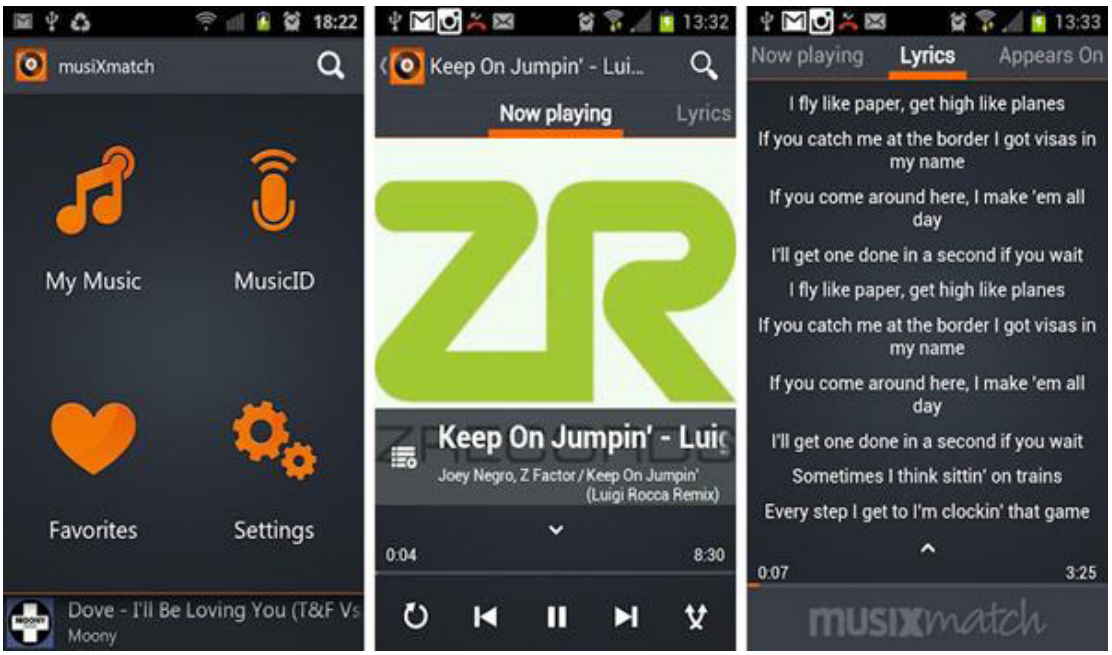

Рисунок 1.8 – Вид музыкального плеера Musixmatch

1.1.9 Rocket Music Player

Еще одно приложение, которое позволяет персонализировать, это Rocket Music Player. Это, пожалуй, самое настраиваемое приложение из тех, что мы перечислили, с более чем 30 темами на выбор.

Мы можем искать музыку, используя различные категории, включая исполнителя, список воспроизведения, композитора, жанр и т. Д. Он также позволяет редактировать теги в соответствии с вашими предпочтениями. Что нам нравится в Rocket Music Player, так это то, что он серьезно относится к безопасности ваших плейлистов. Мы не хотим, чтобы кто-то возлагал руки на наш плейлист! Это приложение предлагает дополнительную функцию блокировки экрана. Также нет необходимости приобретать функцию таймера отключения, поскольку она предварительно установлена вместе с закладками подкастов (Рисунок 1.9).

Он также позволяет редактировать теги в соответствии с вашими предпочтениями. Что нам нравится в Rocket Music Player, так это то, что он серьезно относится к безопасности ваших плейлистов. Мы не хотим, чтобы кто-то возлагал руки на наш плейлист! Это приложение предлагает дополнительную функцию блокировки экрана. Также нет необходимости приобретать функцию таймера отключения, поскольку она предварительно установлена вместе с закладками подкастов

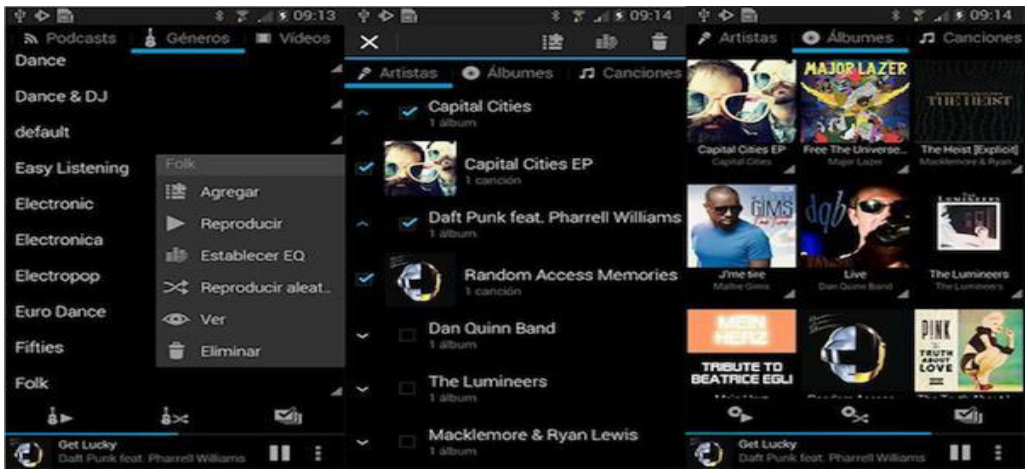

Рисунок 1.9 – Вид музыкального плеера Rocket Music Player

1.1.10 DoubleTwist

DoubleTwist удобен и одновременно и стильный, и функциональный. Это приложение не просто музыкальный проигрыватель Android, потому что оно также поддерживает музыку, подкасты и радио. Пользователи любят, что это приложение может транслировать ваши мелодии на ваши Xbox 360, PS3, колонки Sonos и другие устройства (Рисунок 1.10).

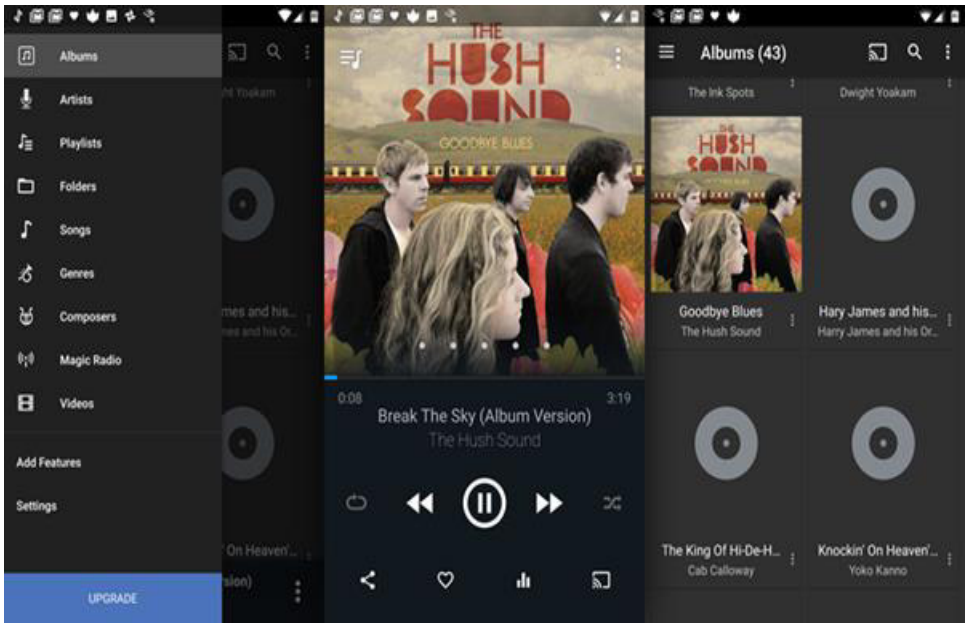

Рисунок 1.10 – Вид музыкального плеера DoubleTwist

Что нам действительно нравится в этом приложении, так это его Magic Radio. Это функция, которая обеспечивает настраиваемое сочетание песен на основе вашей музыкальной библиотеки. Нам нравится слушать музыку, которую мы тщательно выбрали. Он также может синхронизировать плейлисты Windows Media Player с вашего ПК на телефон Android через USB.

1.2 Постановка задачи для дипломного проекта

API – интерфейсы Android Media включают в себя множество расширенных функций, которые позволяют разработчикам создавать мультимедийные возможности. Они включают в себя такие вещи, как ExoPlayer, MediaSession, фокусировка звука, регулировка громкости и множество других потрясающих возможностей, связанных с воспроизведением и управлением мультимедиа.

Цель этого дипломного проекта – познакомиться с API Android MediaPlayer, пройдя через процесс создания очень простого приложения для воспроизведения звука под названием «A Simple MediaPlayer». Как работает MediaPlayer, рассматривая некоторые простые задачи воспроизведения аудио и как они отображаются на конечный автомат MediaPlayer [6].

1.2.1 Основные задачи MediaPlayer

Вот основные задачи, которые MediaPlayer должен выполнить:

– загружаем медиа-файл для воспроизведения. Это делается с помощью методов setDataSource (), prepare () и prepareAsync ();

– начнинаем воспроизведение аудио. Это обрабатывается start ().
Приостановим воспроизведение (после начала воспроизведения). Это обрабатывается pause ();

– остановим воспроизведение и перезагрузите MediaPlayer, чтобы в него можно было загрузить другой медиа-файл. Это обрабатывается с помощью reset ();

– найдем длину песни (в мс). Это обрабатывается getDuration ().
Найдем, какая часть песни играет. Это обрабатывается getCurrentPosition ();

– перейдем к определенной временной позиции (в мс) в песне и начните воспроизведение оттуда. Это обрабатывается seekTo (позиция);

– проверим, воспроизводится ли сейчас аудио. Это обрабатывается isPlaying ().

Узнайте, когда песня закончена, играя. Это выполняется путем присоединения MediaPlayer.OnCompletionListener. Ваш код получит обратный вызов onCompletion () от слушателя. Освободить ресурсы, используемые плеером. Это обрабатывается release (), который освобождает все ресурсы, прикрепленные к игроку. После освобождения игрок больше не может быть использован.

1.2.2 Упрощенный конечный автомат

Вот упрощенное описание конечного автомата MediaPlayer для воспроизведения аудио:

Сначала создаем экземпляр MediaPlayer. Затем загрузите нужный

медиафайл в проигрыватель. Это означает использование метода set Data Source () с параметром, который описывает источник мультимедийного файла. Теперь вам нужно подготовить свой источник данных.

Есть два способа выполнить это:

– используйте Ready(). Этот метод заблокирует вызывающий поток, поэтому его следует использовать только для источников данных, которые хранятся локально на устройстве;

– используйте prepareAsync(). Этот метод подготовит источник данных асинхронно и должен использоваться для удаленных или больших источников данных. Используя этот метод, вы получаете обратный вызов, когда ваши медиафайлы готовы к воспроизведению, и вы можете включить элементы управления воспроизведением мультимедиа в пользовательском интерфейсе вашего приложения на этом этапе. Вы можете использовать MediaPlayer.OnPreparedListener, чтобы получать уведомления, когда проигрыватель загрузил контент для вас, чтобы начать воспроизведение. Вы можете подключить этот слушатель к вашему плееру, используя setOnPreparedListener (слушатель). Теперь, когда ваш звук загружен, вы можете воспроизводить и приостанавливать звук. И вы можете остановить и возобновить воспроизведение.

Воспроизвести - используйте start (), чтобы сообщить плееру о начале воспроизведения. Мы можете спросить плеера, играет ли он, используя isPlaying ().

Пауза - вы можете сделать паузу, только если isPlaying () имеет значение true, поэтому всегда проверяйте это, прежде чем вызывать pause ().

Stop - используйте метод reset () для этого. Он остановит воспроизведение и перезагрузит плеер, чтобы вы могли загрузить в него другие медиа-файлы.

Seek - используйте метод seekTo () со смещением по времени, чтобы переместить воспроизведение звука в нужную позицию. Это необходимо, когда вы позволяете пользователю перемещать воспроизведение в любую позицию на загруженном в данный момент носителе.

Когда вы закончите с экземпляром MediaPlayer, обязательно отпустите его (). После освобождения необходимо создать новый объект MediaPlayer и начать с шага 1, чтобы воспроизвести другой медиафайл.

2 Этапы создания мобильных приложений

Как сделать приложение – важный вопрос для современного технического бизнеса. Мир движется к мобильности все больше и больше. Отрасли промышленности и бизнеса, не имеющие отношения к мобильным технологиям, немного. Следовательно, предприятия и предприниматели ищут новые возможности для роста. Одним из современных вариантов выбора увеличения продаж и количества клиентов является мобильное приложение. Это в руках каждого. Отсюда вытекает следующий вопрос - как создать приложение? В связи с этим, чтобы вывести идею вашего приложения на свет, нужно следовать правильному курсу в процессе разработки мобильного приложения. В этой дипломной работе мы хотели бы поделиться нашим опытом, чтобы помочь вам понять этот процесс и сделать успешную заявку. Сначала краткое резюме этапов, а затем подробное описание бизнеса по созданию приложения.

2.1 Введение в создание мобильного приложения

Говоря о стоимости создания мобильного приложения, мы не можем не рассказать об основах разработки приложений. Все это, конечно, начинается с чьей-то блестящей идеи, а затем проходит различные стадии, пока не будет реализовано. В Интернете вы можете найти множество советов, таких как создание приложения за 10 шагов или 5 ключевых этапов разработки приложения.

И эти этапы / шаги следующие:

Идея и цели: важен не только момент Эврики с отличной идеей для приложения, но и реальная и подробная бизнес-стратегия. Как вы думаете, почему люди будут использовать ваше приложение? Что делает ваше приложение, в основном? Какую проблему поможет решить приложение? Все должно быть ясно.

Исследования: включая общее исследование рынка, исследование конкурентов и особенно исследование пользователей / клиентов. Есть два способа определить вашу целевую аудиторию: кто и сколько. Затем определите ваших главных конкурентов, проанализируйте их продукты, сравните их стратегии. Попробуйте придумать что-то, чего им не хватает.

Каркасы и UX: первые макеты вашего приложения созданы, чтобы понять пользовательский поток и то, что приложение должно делать. Это также называется информационной архитектурой - схемой, отражающей все возможные действия пользователя. Каркас - это грубый шаблон, который вы можете редактировать и исправлять на ранних стадиях. Полезные инструменты – Mockingbot, Axure, Sketchapp и др.

Внутренняя конструкция: или архитектура приложения, для которой вы должны выбрать платформы и набор инструментов, технологический стек, языки программирования, сервисы. Как создать приложение - это вопрос,

который в большинстве случаев предполагает создание серверной части, структуры приложения. Существует два основных способа разработки серверной части: SaaS в качестве серверной части - например, Firebase, AWS Mobile Hub, CloudKit или пользовательский бэкэнд. Последнее предпочтительнее, если вы прогнозируете, что в вашем мобильном приложении появятся новые функции и количество пользователей. Добавление дополнительных функций или их настройка будет проще с собственной серверной системой.

Визуальный дизайн: изготовить несколько вариантов дизайна. Обычно дизайнер создает 3 экрана, то есть три версии визуальной концепции приложения. Все остальные экраны далее основаны на визуальной концепции по вашему выбору. Цвет приложения тоже имеет значение. Большинство основных приложений выполнены в сине-зеленой цветовой гамме (Skype, Twitter, WhatsApp, Shazam).

Кодирование и объединение всего этого. Это самый важный этап, когда ваша команда разработчиков садится и пишет код. Разработчики выполняют задачи под строгим руководством менеджера проекта (практика разработчиков программного обеспечения).

Тестирование: После этого инженеры по обеспечению качества тестируют приложение. Тестирование программного обеспечения играет значительную роль в разработке мобильных приложений, улучшении и очистке конечного продукта. Все ошибки должны быть раздавлены.

Релиз: после того, как все настройки и улучшения сделаны и очищены (и протестированы снова), появляется последняя глава в саге разработки приложений. Проверьте политики отправки в магазине приложений по вашему выбору, чтобы минимизировать время и усилия, чтобы выполнить. Теперь несколько соображений об этих этапах и о том, как сделать мобильное приложение. Самым первым шагом в создании нового приложения является идея. Каждое мобильное приложение начинается с момента Эврика. Рынок мобильных приложений уже полон всех мыслимых типов приложений. Очевидно, что идея вашего приложения должна быть действительно выдающейся, удивительной и захватывающей. И как можно больше полезного для пользователей. Если концепция приложения велика, процесс разработки и все остальные части вашего приложения будут гладкими [7].

Определите идею. Не забывайте, что в таких уникальных приложениях, как игра в дополненную реальность Pokemon Go, изначально было очень мало сторонников. Snapchat также сначала пытался набрать обороты, как «просто еще один мобильный чат». Таких примеров много. Поэтому постарайтесь правильно изложить и представить идею. Будь то простое игровое приложение или мобильное приложение VR, которое помогает ветеранам войны преодолеть ПТСР.

Установите цели. Мобильное приложение, как и любое другое деловое предприятие, должно основываться на решении проблем пользователей.

2.2 Этапы создания мобильного приложения

Создание мобильного приложения не является загадкой в наше время, однако создание успешного мобильного приложения - это процесс, который включает в себя довольно обширное предварительное планирование. За последние несколько лет на рынке приложений произошел стремительный рост, и миллионы приложений появились в двух известных магазинах приложений. В таком случае, когда конкуренция настолько велика, не думаете ли вы, что вы должны быть действительно основательными с основными концепциями построения приложений? Кроме того, вы должны действительно выделять его по функциональности, по своему использованию и по своему дизайну, чтобы привлечь внимание потребителя. И чтобы достичь всего этого, вы должны сделать это правильно с самого первого раза. Так как для приложения очень важно быть на месте в первый раз, поэтому разработчикам необходимо следовать пошаговому процессу создания приложения. В Queppelin мы создаем мобильное приложение для вас в восемь различных этапов, и каждый этап включает в себя несколько этапов. Этот процесс создания приложения, основанный на подробном пошаговом подходе, называется жизненным циклом разработки мобильных приложений. Чтобы понять это далее, жизненный цикл разработки мобильных приложений - это просто представление обычного жизненного цикла разработки программного обеспечения (SDLC), но с точки зрения мобильного устройства. Поэтому без лишних слов давайте рассмотрим восемь этапов цикла разработки мобильных приложений.

2.2.1 Этап 1: предварительное планирование и исследование

Первый этап является наиболее важным, потому что именно на этом этапе вы закладываете необходимую основную работу для того, что последует дальше. На этом этапе очень важно провести серьезные исследования и провести мозговой штурм, прежде чем переходить к следующему этапу. Вам нужно сделать домашнее задание и получить ответы на такие вопросы, как - Какова основная цель этого приложения? Кто целевая аудитория? На какую платформу вы должны ориентироваться в первую очередь? Приложение будет бесплатным или платным?

Получив ответ на все эти вопросы, вы четко знаете, сколько времени потребуется для разработки приложения. Еще одна вещь, которая является обязательной на этом этапе, это анализ конкуренции. Подробно изучите приложение вашего конкурента, чтобы увидеть, какие функции они предлагают. Постарайтесь выяснить, какие функции отсутствуют в их приложении, чтобы вы могли включить его в свое приложение, чтобы выделить его. Как только вы получите всю эту информацию, вам нужно будет определить стоимость и время разработки приложений [8].

2.2.2 Этап 2: Ментальный прототип

Как только вы закончите с исследованием и определите связанные с этим расходы, следующий этап включает в себя подготовку подробного объема работ. Вы должны сделать мысленное прототипирование своего приложения и нарисовать свои идеи в виде эскизов на доске. Это будет первое визуальное представление идей, которые вы собрали на этапе 1, и оно поможет вам раскрыть проблемы юзабилити. Еще одна вещь, которую необходимо сделать на этом этапе, - это узнать мнение соответствующих людей, чтобы понять, что они думают о вашей идее. Обсуждение этого с ними поможет вам выяснить лазейки и позволит вам найти решение для их решения.

2.2.3 Этап 3: Оценка технической осуществимости

Понимания визуальных эффектов недостаточно, поскольку нам необходимо проанализировать, будут ли серверные системы поддерживать функциональность приложения или нет. Чтобы понять, технически осуществима ли идея вашего приложения, вам нужно получить доступ к общедоступным данным, просто используя общедоступные API. Мы также должны сначала определить, для какой платформы вы создаете свое приложение. Создание приложения должно иметь различные требования в зависимости от его платформы (Android / iOS), а также его формата (планшет / смартфон).

2.2.4 Этап 4: Создание прототипа

Рисунок 2.1 – Создание прототипа приложения

Вы не можете определить сенсорный опыт до тех пор, пока вы не

коснетесь приложения и не увидите, как оно работает и работает. Чтобы это произошло, вы должны как можно скорее создать прототип и получить опыт работы с приложением. Это поможет вам понять, идут ли дела в правильном направлении (Рисунок 2.1). На этом этапе вы можете использовать грубые и не исчерпывающие каркасы. Включение заинтересованных сторон в этот процесс и предоставление им возможности прикоснуться к прототипу поможет вам учесть их отзывы и реализовать их в своей работе.

2.2.5 Этап 5: Проектирование и разработка приложения

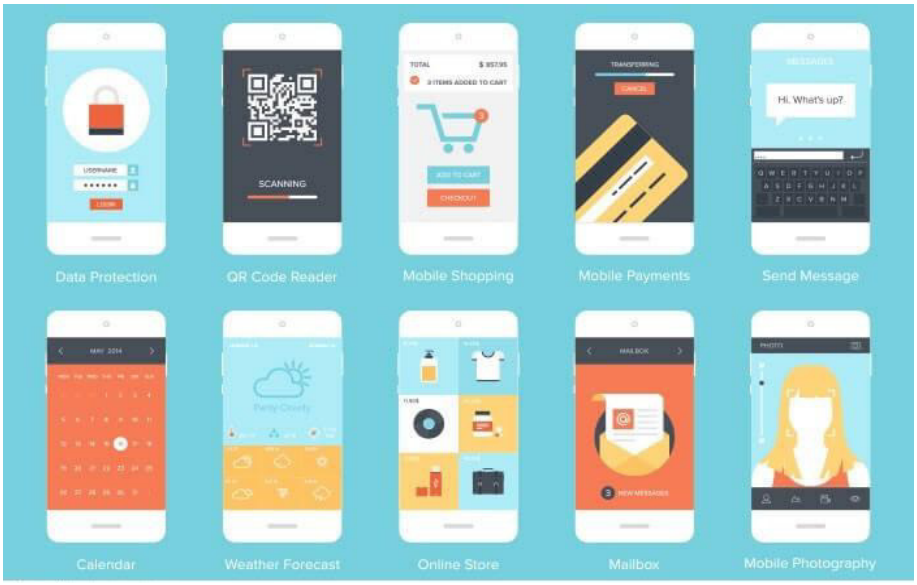

Рисунок 2.2 – Проектирование приложения

Прежде чем перейти к программированию, вы должны разработать приложение. Дизайнер пользовательского интерфейса может создать архитектуру взаимодействия элементов дизайна, а дизайнер пользовательского интерфейса может создать внешний вид вашего приложения. Сам по себе это многоэтапный процесс, а конечные результаты - это визуальные указания и чертежи, которые дают представление о конечном продукте (Рисунок 2.2). Он также информирует вас о том, как взаимодействие должно ощущаться, двигаться и течь [9].

Как только дизайн будет готов, настало время для создания приложения. Гибкая методология является лучшим подходом для разработки мобильных приложений, поскольку она позволяет вносить изменения, добавлять новые функции и продолжать развиваться с изменяющимися тенденциями.

Сам по себе это многоэтапный процесс, а конечные результаты - это визуальные указания и чертежи, которые дают представление о конечном продукте. Он также информирует вас о том, как взаимодействие должно ощущаться, двигаться

2.2.6 Этап 6: Создание приложения с использованием гибких

методологий

Рисунок 2.3 – Использование гибких методологии

Как только дизайн будет готов, настало время для создания приложения. Гибкая методология является лучшим подходом для разработки мобильных приложений, поскольку она позволяет вносить изменения, добавлять новые функции и продолжать развиваться с изменяющимися тенденциями (Рисунок 2.3).

2.2.7 Этап 7: Тестирование мобильного приложения

Рисунок 2.4 – Процесс тестрования

Для фазы 7 нам нужно привлечь целевых пользователей, которые помогут нам протестировать приложение.

Тестирование UAT: для тестирования пользовательского опыта вам нужно передать свое приложение в руки пользователей, на которых вы ориентируетесь, и как только оно пройдет тест UAT, вы поймете, что решение, которое вы предоставляете, действительно работает.

Бета-тестирование. Сделайте свое приложение доступным для бета-тестирования, предоставив возможность открытого тестирования людей. Отзывы этих бета-пользователей помогут вам определить, хорошо ли функционируют ваше приложение в реальной среде (Рисунок 2.4).

2.2.8 Этап 8: Запуск

Как только фаза 7 будет успешной, наше приложение будет готово и будет отправлено в магазины приложений для утверждения. Однако это не конец, так как каждое приложение требует регулярных обновлений и добавления новых функций в течение жизненного цикла разработки мобильного приложения. Цикл разработки начинается снова, как только запускается первая версия приложения. Есть ли на Земле кто-то, кто не любит слушать музыку? Вполне вероятно, что у каждого из нас есть пара исполнителей, чьи треки мы готовы слушать при повторении. Если вы проанализируете эту тенденцию с точки зрения разработчика программного обеспечения Android, вы сможете сделать вывод, что создание приложения для потоковой передачи музыки является невероятно многообещающей областью деятельности, которая может принести большую популярность и значительную прибыль.

2.3. Актуальность создания популярного потокового приложения

В настоящее время продажи цифровых записей растут. По данным IFPI, в 2016 году около 50% всех продаж музыки были цифровыми. В отчете Statista говорится, что количество платных онлайн-подписок в США выросло на 50% в прошлом году - с 20 до 30 миллионов.

Мобильные музыкальные приложения можно отнести к пяти различным типам:

– отслеживание приложений распознавания;
– проигрыватель приложений;
– загрузка приложений;
– настройка создания приложений; а также;
– онлайн потоковые приложения.

В этом дипломном проекте мы обсудим последний тип приложений и узнаем, как создать музыкальное приложение для Android. Прежде чем приступить к описанию процесса разработки музыкального приложения, давайте определимся, какие атрибуты характерны для него. Ориентация профиля и возможности персонализации. Подавляющее большинство современных приложений ориентировано на профили. Это означает, что в рамках своей работы пользователи могут создавать свои собственные профили с указанием некоторой информации о себе (наиболее примитивным вариантом является сочетание уникального псевдонима и аватара). Популярность этого формата приложения обусловлена тем, что почти каждый

стремится продемонстрировать свою уникальность в глазах других, а любимые треки указывают на вкусы пользователя. Чтобы продвинуться в этом еще дальше, многие разработчики предоставляют функциональность, которая позволяет настраивать личный кабинет пользователя (устанавливать индивидуальную тему, добавлять фото–аватар, загружать аудио- и видеозаписи из личного архива и т. д.). Возможность скачивать и хранить музыку. Конечно, любое мультимедийное приложение значительно теряет свою ценность, если оно не обеспечивает возможность загрузки и хранения мультимедиа, поскольку в отсутствие подключения к Интернету пользователи просто не смогут получить доступ к своим любимым трекам. Что касается хранилища, правильная структура будет включать как минимум два способа сохранения - на сервере (облачное хранилище) и на персональном устройстве пользователя.

Возможность интеграции с социальными сетями (Рисунок 2.6). Найдя действительно классный трек, пользователь, вероятно, захочет поделиться им с другими. Самый простой способ сделать это - через учетную запись социальной сети. Поэтому при создании потокового приложения настоятельно рекомендуется подумать о способах его интеграции с известными социальными сетями [10].

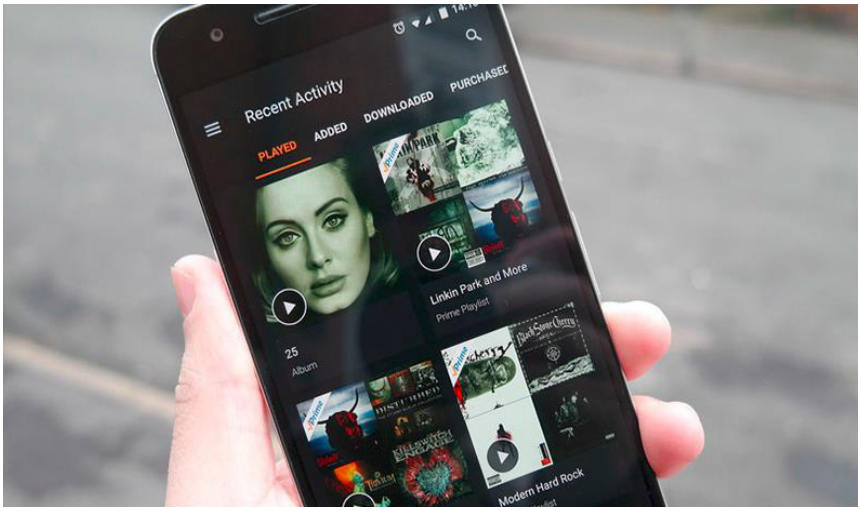

Рисунок 2.5 – Возможность интеграции с социальными сетьями

Возможность общения между пользователями. Вернемся к первому пункту - ориентация профиля и возможности персонализации. Возможность оставлять комментарии позволяет пользователям еще больше персонализировать свой опыт, а возможность искать друзей по интересам также является еще одним шагом в правильном направлении. Кроме того, ваше приложение может включать форумы с темами о музыкальных тенденциях, событиях и артистах.

Уникальные черты. Музыка – настолько популярная тема в разработке программного обеспечения, что, даже создав совершенно новое по

функциональности приложение, вы все равно можете привлечь значительную аудиторию. Тем не менее, мы предлагаем вам продвинуться вперед, создав хотя бы одну уникальную функцию, которая в долгосрочной перспективе послужит примером для многих похожих приложений (как это сделали разработчики уже закрытого проекта Beats Music) с музыкой, основанной на настроения пользователя и т. д.).

Похожий поиск музыки. Самый простой способ увеличить количество пользователей – это максимизировать активность тех, кто уже существует. Это можно сделать очень просто - предлагая треки, похожие на те, которые уже добавлены в плейлист пользователя.

Стильный дизайн. Дизайн является одним из фундаментальных факторов, который напрямую влияет на популярность потокового приложения. Разработав современный интерфейс в соответствии с принципами Material Design и набором стилистических правил, присущих платформе Android, вы, скорее всего, получите действительно модный и загружаемый программный продукт.

2.4 Этапы создания музыкального приложения для Android

Выше мы рассмотрели основные атрибуты потоковых приложений для Android. Теперь давайте коснемся четырех моментов, которые должен учитывать каждый разработчик при принятии решения о создании специального программного решения.

Выбор методов хранения данных. При планировании разработки приложения для потоковой передачи музыки вам нужно будет продумать варианты хранения загруженных мелодий. Чаще всего на смартфонах используется облачное хранилище или локальное кеширование; тем не менее, вы можете использовать небольшую хитрость в своем приложении, прикрепляя профили пользователей к соответствующим страницам в социальных сетях, чтобы получить дополнительную выгоду от подключенных хранилищ.

Создание минимального набора функций для обеспечения жизнеспособности продукта. Подводя итог вышесказанному, мы можем утверждать, что обязательными атрибутами для приложений потоковой передачи музыки являются такие функции, как регистрация на основе профиля пользователя, поиск по нескольким критериям, загрузка и кэширование мультимедиа, автономное воспроизведение и совместное использование в социальных сетях.

Получение музыкальных лицензий. Законодательные последствия - довольно сложный вопрос. Обратите внимание, что основным преимуществом потоковых приложений является возможность прослушивания треков, предпочитаемых другими пользователями, бесплатно, в рамках закона и в Интернете (для подтверждения действительности ваших действий, свяжитесь с SESAC, если вы продвигаете свое приложение в европейских странах и

ASCAP или BMI в США). Тем не менее, если вы планируете реализовать возможность загрузки треков, вам придется беспокоиться о легитимизации библиотеки в вашем приложении.

Выбор метода монетизации. Конечно, конечная цель создания любого приложения - заработать деньги. Поэтому важной частью его развития будет выбор методов монетизации. Что бы вы ни выбрали, не будьте жадными. Платное приложение, которое дополнительно использует рекламные объявления, является практикой, которая обескураживает пользователей. Даже если ваше приложение уникально по функциональности и еще не имеет конкурентов, конкуренция рано или поздно возникнет, и даже такая небольшая разница, как меньшее количество рекламы, может быть достаточной причиной для выбора альтернативного продукта по сравнению с вашим.

2.5 Обзор лучших музыкальных потоковых приложений

Давайте рассмотрим пять лучших приложений для потоковой передачи музыки, созданных для платформ iOS и Android, и концепции которых могут вдохновить вас на создание уникального мобильного решения.

Пандора. Мобильное потоковое приложение Pandora - это уникальный персонализированный опыт и способ прослушивания онлайн-песен.

Функция автоматической генерации музыкальной коллекции на основе индивидуальных вкусов пользователей позволяет пользователям слушать новую музыку без поиска новых треков. Это не так, хотя - Пандора предлагает больше! Вы также можете выбрать любой жанр или группу, которая вам больше всего нравится, и сразу же получить полный список песен, которые могут вас заинтересовать. Кроме того, приобретая премиум-подписку, вы получаете возможность загружать треки и удалять надоедливую рекламу.

Spotify. Стильное мобильное приложение Spotify обеспечивает доступ к вашей любимой музыке из любой точки мира с доступом к Интернету. Пользователи могут выбрать прослушивание произвольных треков и наоборот, а также создавать свои собственные списки воспроизведения, составленные из их любимых исполнителей. Благодаря своей огромной и абсолютно бесплатной платформе для прослушивания мелодий, Spotify получил множество поклонников и приобрел активных пользователей по всему миру.

Радио IHeart. IHeart Radio – это мобильное приложение, которое предоставляет бесплатный доступ к более чем 1500 онлайн-радиостанциям, позволяя поклонникам абсолютно любого музыкального жанра найти то, что им нравится. Это приложение не просто привлекает аудиофилов; с новостями, ток-шоу, интервью - этот список можно продолжить и предложить что-то каждому, кто слушает iHeart Radio.

Звук Облака. Если вы следуете тенденциям, в том числе музыкальным, Sound Cloud - именно то, что вам нужно. Вы можете создавать свои

собственные плейлисты, вдохновленные как вашими вкусами, так и предпочтениями песен ваших друзей или любимых музыкантов. Регистрация в Sound Cloud очень проста и может быть сделана с помощью учетной записи Google или Facebook.

Beats Music. Мобильное приложение Beats Music для iPhone, заблокированное Apple в 2014 году для общего доступа, позволяет пользователям слушать свои любимые треки как в офлайн-режиме, так и в автономном режиме (если пользователь скачал их заранее). Beats Music была одним из фундаментальных программных решений, которые первыми применили автоматическое создание списков воспроизведения на основе настроений пользователей [11].

Стоимость разработки музыкального приложения: оценка этапов разработки. Теперь давайте обсудим стоимость разработки приложения для потоковой передачи музыки. Независимо от сложности, его разработка будет включать в себя следующие этапы:

Инженерный этап. Этот этап включает в себя все, что поможет построить четкое видение будущей реализации программного обеспечения (а именно, выбор подходящих инструментов: фреймворк, SDK, библиотеки и т. Д.). Как правило, разработка программного обеспечения занимает не менее 80 часов.

Кодирование. Очевидно, что написание кода является одним из самых трудоемких этапов при создании приложения для потоковой передачи музыки. Как показывает практика, даже самое примитивное приложение со стандартным набором простых функций и тривиальным дизайном займет не менее 720 часов.

Тестирование. Поскольку ваше приложение предназначено для массового использования, вам придется потратить много времени на тестирование готового продукта. Основываясь на реальных отзывах и комментариях пользователей, вы можете позже выпустить более продвинутое решение, которое гарантированно привлечет больше новых пользователей по сравнению с предыдущими версиями. Сколько будет стоить бета-тестирование, напрямую зависит от того, какую цену вы предлагаете потенциальным тестировщикам за их услуги. Кроме того, вы можете разместить приложение в специальном разделе Google Play. Этот раздел, отличающийся только для тестовых версий приложений, предоставит группе самую подробную информацию о найденных ошибках и требованиях реальных пользователей.

Техническое обслуживание. Для поддержки жизненного цикла выпущенного приложения для потоковой передачи музыки потребуется в среднем 15 часов в месяц. Кроме того, вам придется платить регулярные и текущие лицензионные сборы для доступа к трекам.

В зависимости от сложности приложения и с учетом почасовых ставок разработчиков из стран бывшего СССР, разработка типичного приложения для потоковой передачи музыки может стоить где-то между 24000 и 60000

долларов.

Благодаря разработке музыкального плеера на платформе Android мы получаем четкое представление об общем процессе работы системы. Основная часть музыкального проигрывателя в основном состоит из основного интерфейса, списков воспроизведения, меню, настроек воспроизведения, просмотра файлов и поиска песен. Захватив развитие шести частей, музыкальный проигрыватель имел предварительную шкалу. Основываясь на функции шести категорий, добавьте некоторые другие небольшие функции. Система музыкального плеера реализовала основные функции плеера: воспроизведение, пауза и остановка, увеличение / уменьшение а, регулировка громкости, отображение текста песни, режим воспроизведения, поиск песен, просмотр файлов, запрос к спискам воспроизведения и другие функции. Это развитие связано с популярной технологией разработки мобильных терминалов. Это комбинированное управление языком Java в мобильной платформе с открытым исходным кодом на основе системы Linux + + поддержка базы данных SQLite + файл конфигурации SharePreference. Система реализовала программирование музыкального плеера. Этот дизайн музыкального плеера на основе системы Android требует тщательно продуманного дизайна музыки каркас плеера,
приняв язык Eclipse3.5 + Java в качестве технической поддержки этой системы, с помощью инструментов плагина Android и комбинации версии Android SDK2.1.

2.6 Обзор архитектуры мобильных приложений

Дизайнеры мобильных приложений всегда ищут правильный инструмент, который сделает их дизайн полезным для конечного пользователя. Разработка ценного мобильного приложения требует высочайших навыков, креативности и, конечно, правильных инструментов. Может ли это быть причиной того, что компании вкладывают больше денег в удобные мобильные дизайны? Ответ положителен. Что эти компании ожидают получить от этого смелого шага? Ну, согласно DMI, «через 10 лет инвестиции в 10 000 долларов в компании, ориентированные на дизайн, дадут отдачу на 228% больше, чем те же инвестиции в S & P». Несмотря на то, что по-прежнему важно рисовать каркас и прототип продукта на бумаге, программное обеспечение для разработки приложений делает его менее утомительным и простым для обмена готовыми или незаконченными работами с клиентами, разработчиками и дизайнерами. Но сейчас, когда в мире так много инструментов для проектирования, может быть сложно выбрать лучший инструмент для вашего конкретного типа приложения. Прототипы дают жизнь вашей новой идее. Вам определенно не нужно будет использовать все эти инструменты, только тот, который, по вашему мнению, ваша команда дизайнеров должна перейти на следующий уровень. И последнее, но не менее важное: если вы ищете универсальный инструмент для

мобильных дизайнеров без всяких проблем с кодированием, вы должны попробовать Buildfire.com – он позволит вам с интересом создать мобильное приложение для вашего бизнеса в 3 простых шага: выберите шаблон приложения, настройте приложение, опубликуйте и управляйте.

Задумывались ли вы, какие аспекты вашей жизни не были затронуты возможностями смартфона? Разве мы не зависим от этого инструмента практически от всего, от работы, путешествий, покупок, банковского дела и даже до обучения? Даже предприятия используют мобильность для повышения эффективности своей рабочей силы. Это привело к жесткой конкуренции среди компаний-разработчиков мобильных приложений. В результате многие игроки не могут достичь поставленной цели.

Часто причина неудачи объясняется тем, что компании-разработчики приложений не осознают одну из основ разработки мобильных приложений - архитектуру мобильных приложений. Что такое архитектура мобильных приложений?

Архитектура мобильных приложений – это набор методов и шаблонов, используемых для разработки полностью структурированных мобильных приложений на основе отраслевых и отраслевых стандартов. При разработке архитектуры приложения также учитываются процедуры, которые работают на беспроводном мобильном устройстве, таком как смартфоны и планшеты. Проектирование архитектуры мобильного приложения обычно состоит из нескольких уровней в приложении, включающем следующие уровни:

Уровень представления. Этот уровень включает компоненты пользовательского интерфейса, а также компоненты процесса пользовательского интерфейса.

Бизнес-уровень. Этот уровень состоит из бизнес-объектов, рабочих процессов и бизнес-компонентов.

Уровень данных – компоненты доступа к данным, утилиты данных и сервисные агенты вместе образуют этот уровень.

Элементы, которые необходимо учитывать перед разработкой архитектуры вашего мобильного приложения

Поскольку создание лучшей архитектуры приложений имеет решающее значение для успеха разработки вашего мобильного приложения, вам необходимо помнить о следующих факторах, прежде чем приступить к разработке архитектуры вашего приложения:

Определение типов устройств

Существуют разные категории смартфонов, и вам очень важно оценить тип устройства и его характеристики до выбора конкретной архитектуры приложения (Рисунок 2.6).

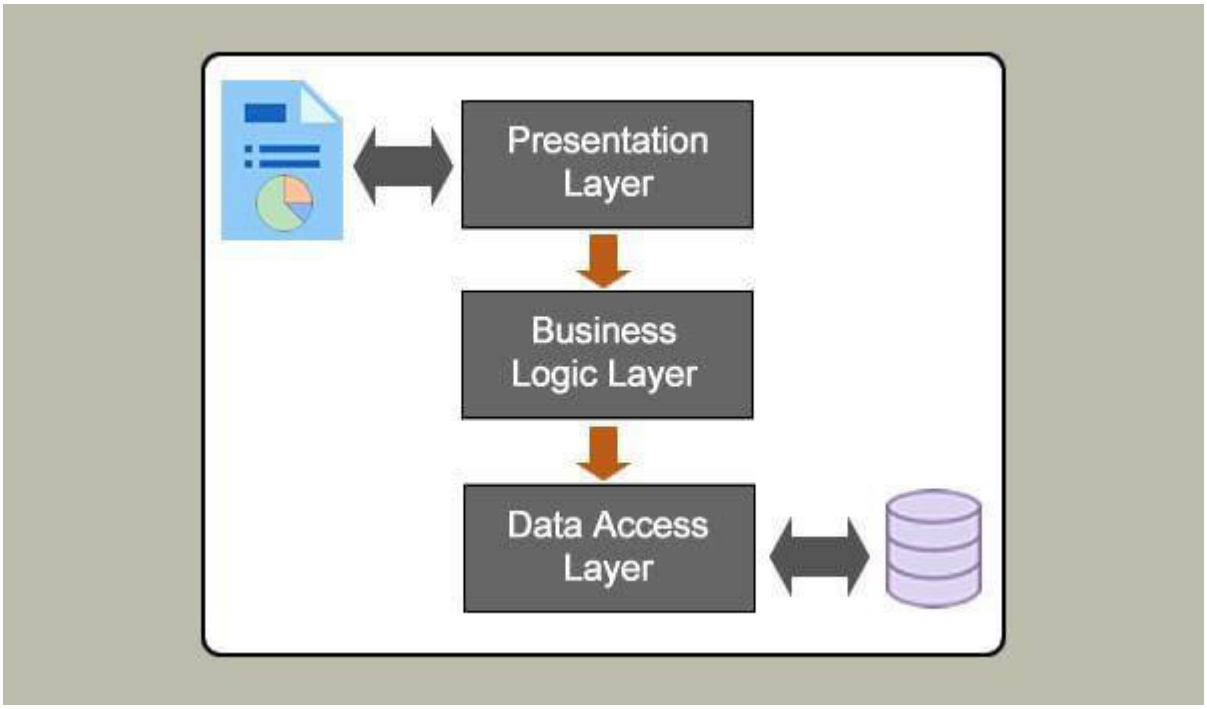

Рисунок 2.6 – Иерархическая структура данных

Вы должны помнить следующие характеристики устройства:

– разрешение экрана;

– размер экрана;

– характеристики процессора;

– объем памяти;

– наличие основы разработки;

– место для хранения.

Почему мы должны определить тип устройства при выборе архитектуры приложения?

Это связано с тем, что функции вашего мобильного приложения могут иметь определенные аппаратные и программные требования.

Учитывая сценарии пропускной способности, чрезвычайно важно помнить о сценариях сети Интернет в тех областях, где процветает наша целевая аудитория. Это связано с тем, что колебания скорости интернета могут негативно повлиять на пользовательский опыт. Наши клиенты не могут быть постоянно подключены к Интернету. Иногда они могут даже иметь прерывистую сеть. Поэтому при создании мобильного приложения вы должны учитывать наихудшие сценарии интернет-сети.

Надо учитывать энергопотребление и скорость при выборе программных протоколов и оборудования для вашего мобильного приложения. Также спроектируйте свой механизм кэширования, механизм доступа к данным и управление состоянием так, чтобы он мог настраиваться на медленное и прерывистое интернет–соединение [12].

Определение пользовательского интерфейса

Когда дело доходит до создания дизайна пользовательского интерфейса

мобильного приложения, вы можете исследовать горизонты творчества с помощью своего особого стиля. Однако вы должны помнить, что интерфейс должен быть максимально простым, чтобы пользователи могли интуитивно взаимодействовать с ним. Кроме того, имейте в виду, что спутанный пользовательский интерфейс может привести к падению вашего приложения.

Выбор правильного метода навигации

Навигация в приложении является очень важным аспектом, который включает в себя как внешний, так и внутренний интерфейс. При выборе метода навигации вам необходимо будет учитывать предпочтения ваших клиентов, а также требования вашего приложения. Это очень важно, так как это окажет огромное влияние на пользовательский опыт. Среди множества различных способов навигации вы должны проанализировать и выбрать наиболее подходящий для нас;

Вот некоторые из популярных:

– сложенная панель навигации;
– контроллер вкладок;
– модальный контроллер;
– единый вид;
– жест-ориентированная навигация;
– прокрутка просмотров;
– поисковая навигация.

Теперь вопрос в том, как бы мы обеспечили соответствие вашего мобильного приложения требованиям вашей целевой аудитории? Что ж, для этого вам нужно будет следовать определенным рекомендациям при создании приложения, чтобы оно работало эффективно в различных сценариях.

Как сохранить технологический процесс вашего мобильного приложения?

Как правило, ваше мобильное приложение может быть структурировано в три разных уровня, как указано выше; мы взглянем на руководящие принципы, которые должны соблюдаться для каждого слоя.

Уровень представления. Основное внимание на этом уровне уделяется тому, как приложение будет представлено конечному пользователю. При разработке этого уровня разработчики приложений должны определить правильный тип клиента, который соответствует инфраструктуре. Следует также помнить об ограничениях развертывания клиента.

Другим предварительным условием для разработки этого слоя является выбор правильного формата данных и внедрение надежной методики проверки данных, чтобы ваше приложение могло быть защищено от неверного ввода данных. Кроме того, разработчики наших мобильных приложений также сосредоточены на отделении бизнес-логики от кода уровня представления.

Бизнес уровень. Ведение журнала, кэширование, проверка, защита и управление исключениями - это различные аспекты, связанные с бизнес-уровнем. Наши разработчики мобильных приложений заявляют, что вы

должны разделить задачи на различные категории, чтобы уменьшить сложность этого слоя. Тем не менее, они говорят, что использование отдельного бизнес-уровня везде, где это допустимо, является идеальным подходом к разработке этого уровня. Для различных функций, таких как сложные бизнес-правила, применение политик, преобразование данных, проверка и т. Д., Вы должны идентифицировать бизнес-уровень.

Уровень доступа к данным. Этот уровень отвечает требованиям приложения и обеспечивает безопасные транзакции данных. Следовательно, вы должны спроектировать этот слой так, чтобы он мог масштабироваться в будущем по мере изменения требований бизнеса. Будучи опытной компанией по разработке мобильных приложений, мы ориентируемся на выбор правильной технологии доступа к данным, чтобы можно было создать безопасный и высокофункциональный уровень.

Вы также можете использовать табличные идентификаторы для визуализации упрощенной структуры проекта для хранения данных и обеспечения возможности перемещения данных через пограничные слои. Все функции доступа к данным инкапсулированы внутри этого уровня, и он управляет всеми соединениями данных, необходимыми мобильному приложению. Более того, он обрабатывает все операции CRUD (создание, чтение, обновление и удаление) и источники данных.

Мы также реализуем подход с наименьшими привилегиями, чтобы защитить любую попытку кражи или повреждения данных путем защиты механизмов доступа к данным (Рисунок 2.7).

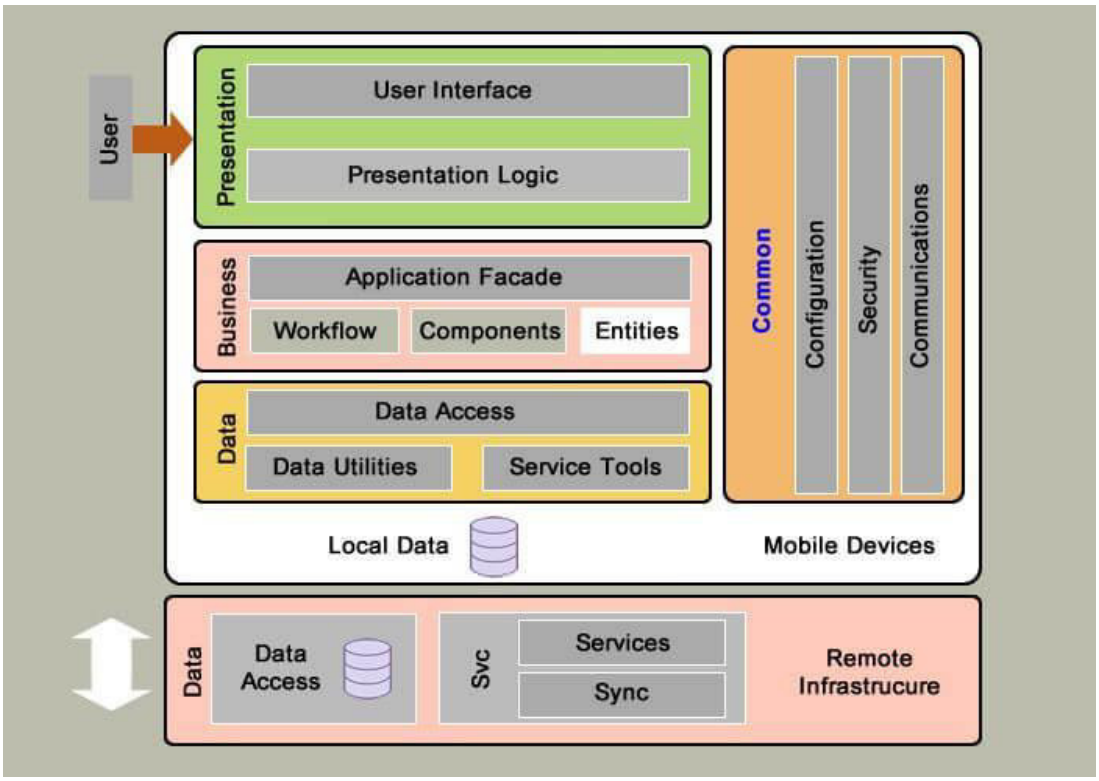

Рисунок 2.7 – Иерархическая структура обработки данных

Как правильно выбрать архитектуру для вашего мобильного приложения?

Мы обслуживаем наших клиентов уже более десяти лет, предоставляя услуги по разработке мобильных приложений. Вот рекомендации нашей команды разработчиков мобильных приложений, которые позволят вам выбрать правильную архитектуру для вашего мобильного приложения:

Если у вас нет бюджетных ограничений, то целесообразно разрабатывать собственные приложения, поскольку они обеспечивают интуитивно понятные функции и производительность. Мы также можем выбрать концепцию «развивайся один раз и беги куда угодно», но придерживайся других подходов. Мы можем либо разработать мобильное веб-приложение, либо гибридное приложение с небольшими усилиями по разработке, либо перейти к кроссплатформенным приложениям с использованием кроссплатформенных сред и обеспечить более богатый опыт при доступе к широкому кругу мобильных устройств.

Если наша целевая аудитория состоит из пользователей iOS и Android, и наша цель – обеспечить лучший пользовательский опыт, то Helios рекомендует разработку собственных приложений. Однако, если нам нужно собрать все другие операционные системы, включая Windows и BlackBerry, кросс-платформенная разработка может быть экономически выгодным вариантом, но удобство использования должно быть поставлено под угрозу. Мы также можем рассмотреть как нативную, так и веб-разработку, чтобы дополнить друг друга и помочь вам лучше привлекать ваших клиентов и сотрудников. В то время как нативные приложения обеспечивают оптимальное взаимодействие с пользователем, веб-приложения обеспечивают видимость вашего бизнеса, связываясь с клиентами со всеми видами устройств. К настоящему времени вы, возможно, уже поняли, насколько важна архитектура мобильных приложений для успешной и прибыльной разработки мобильных приложений. Это не только делает ваше приложение безопасным, но и позволяет масштабировать его с ростом вашего бизнеса. С момента появления смартфона функции воспроизведения музыки и мультимедиа стали неотъемлемой частью одного устройства как смартфона. Это очень удобно, но содержит противоречивые аргументы о качестве звука, поэтому многие пользователи смартфонов используют приложение для воспроизведения музыки. Используя эти музыкальные приложения, люди начинают задумываться о связи между воспроизведением музыки и качеством звука. Однако эти приложения не идеальны, поэтому трудно выбрать хорошее приложение. Этот тезис о преимуществах качества звука приложений музыкального проигрывателя, которые в настоящее время продаются в Android Market через программу RightMark Audio Analyzer, и планирует предложить дизайн системы приложения музыкального проигрывателя для Android, анализируя приложения с учетом недостатков этих приложений.

2.7 Создание музыкального проигрывателя на Android: настройка проекта

Android предоставляет ресурсы для обработки воспроизведения мультимедиа, которые ваши приложения могут использовать для создания интерфейса между пользователем и его музыкальными файлами. В этом уроке мы создадим приложение для основного музыкального плеера для Android. Приложение представит список песен на пользовательском устройстве, чтобы пользователь мог выбирать песни для воспроизведения. Приложение также представит элементы управления для взаимодействия с воспроизведением и продолжит воспроизведение, когда пользователь отходит от приложения, с уведомлением, отображаемым по истечении воспроизведения.

Ищете быстрое решение?

Если вы ищете быстрое решение, на Envato Market есть отличная коллекция шаблонов приложений для Android (Рисунок 2.8). В частности, этот шаблон приложения для Android Music Player – отличный способ начать создавать собственное приложение. "Lite Music" – это шаблон приложения премиум-плеера в Android с простым и элегантным интерфейсом [13].

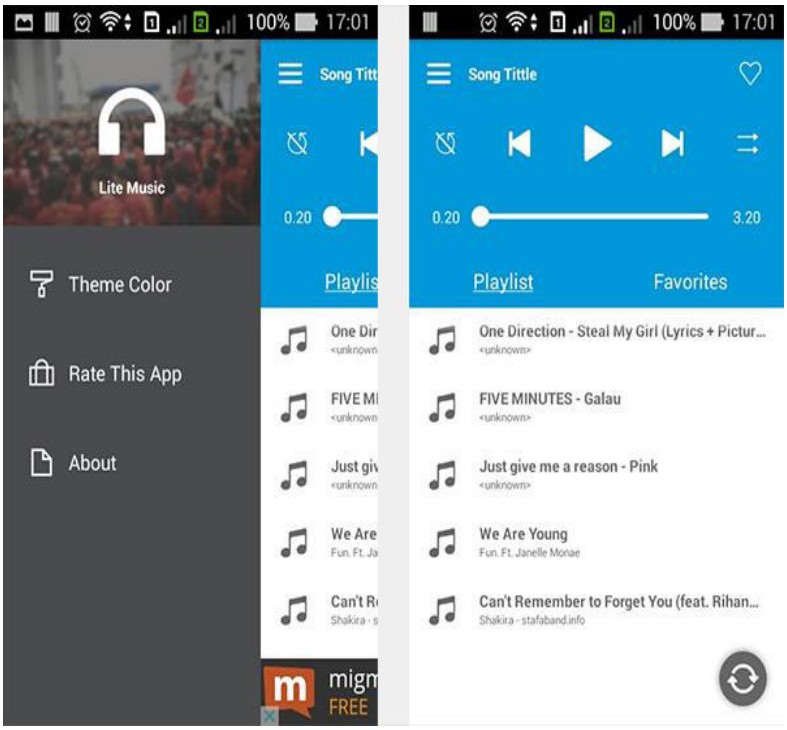

Рисунок 2.8 – Использование шаблонов (образец)

Создание музыкального проигрывателя будет включать использование класса ContentResolver для извлечения треков на устройстве, класса MediaPlayer для воспроизведения звука и класса MediaController для управления воспроизведением. Мы также будем использовать экземпляр

службы для воспроизведения звука, когда пользователь не взаимодействует напрямую с приложением. Мы должны быть в состоянии завершить эту серию, если вы являетесь средним разработчиком Android, поэтому, если мы уже создали несколько приложений, тогда эта серия не должна быть проблемой для вас. Вот предварительный просмотр финального приложения:

В этой части дипломного проекта мы создадим приложение и запросим у аудиоустройства пользовательское устройство, используя классы ContentResolver и Cursor. В следующей части мы будем использовать экземпляр адаптера для представления песен в виде списка, начиная воспроизведение, когда пользователь нажимает на элемент из списка. В последней части этой части мы будем использовать класс MediaController, чтобы предоставить пользователю контроль над воспроизведением, реализовать функции для перехода вперед и назад и включить функцию перемешивания. После этой части мы рассмотрим другие аспекты воспроизведения мультимедиа, которые могут улучшить приложение, такие как обработка аудиофокуса, представление мультимедийных файлов различными способами и воспроизведение потокового мультимедиа.

Платформа Android предоставляет ресурсы для обработки воспроизведения мультимедиа, которые ваши приложения могут использовать для создания интерфейса между пользователем и его музыкальными файлами. В этом уроке мы создадим приложение для основного музыкального плеера для Android. Приложение представит список песен на пользовательском устройстве, чтобы пользователь мог выбирать песни для воспроизведения. Приложение также представит элементы управления для взаимодействия с воспроизведением и продолжит воспроизведение, когда пользователь отходит от приложения, с уведомлением, отображаемым по истечении воспроизведения.

Если мы ищем быстрое решение, на Envato Market есть отличная коллекция шаблонов приложений для Android. В частности, этот шаблон приложения для Android Music Player - отличный способ начать создавать собственное приложение. "Lite Music" - это шаблон приложения премиум-плеера в Android с простым и элегантным интерфейсом.

Создание музыкального проигрывателя будет включать использование класса ContentResolver для извлечения треков на устройстве, класса MediaPlayer для воспроизведения звука и класса MediaController для управления воспроизведением. Мы также будем использовать экземпляр службы для воспроизведения звука, когда пользователь не взаимодействует напрямую с приложением. Вы должны быть в состоянии завершить эту серию, если вы являетесь средним разработчиком Android, поэтому, если вы уже создали несколько приложений, тогда эта серия не должна быть проблемой для вас. Вот предварительный просмотр финального приложения:

2.8 Этапы создания финального проигрывателя Android

В этом пункте мы создадим приложение и запросим у аудиоустройства пользовательское устройство, используя классы ContentResolver и Cursor. В следующей части мы будем использовать экземпляр адаптера для представления песен в виде списка, начиная воспроизведение, когда пользователь нажимает на элемент из списка. В последней части этой серии мы будем использовать класс MediaController, чтобы предоставить пользователю контроль над воспроизведением, реализовать функции для перехода вперед и назад и включить функцию перемешивания. После этой серии мы рассмотрим другие аспекты воспроизведения мультимедиа, которые могут улучшить приложение, такие как обработка аудиофокуса, представление мультимедийных файлов различными способами и воспроизведение потокового мультимедиа.

Android предоставляет множество способов управления воспроизведением аудио / видео файлов и потоков. Один из таких способов - через класс MediaPlayer.

Android предоставляет класс MediaPlayer для доступа к встроенным сервисам медиаплеера, таким как воспроизведение аудио, видео и т. д.

Чтобы использовать MediaPlayer, мы должны вызвать статический метод create () этого класса. Этот метод возвращает экземпляр класса MediaPlayer.

Его синтаксис выглядит следующим образом - MediaPlayer mediaPlayer = MediaPlayer.create (this, R.raw.song);

Второй параметр – это название песни, которую вы хотите воспроизвести. Вы должны создать новую папку в вашем проекте с именем raw и поместить в нее музыкальный файл.

Создав объект Mediaplayer, вы можете вызвать несколько методов для запуска или остановки музыки. Эти методы перечислены ниже:

– mediaPlayer.start ();

– mediaPlayer.pause ().

При вызове метода start () музыка начнет воспроизводиться с самого начала. Если этот метод вызывается снова после метода pause (), музыка начинает воспроизводиться с того места, где она была оставлена, а не с начала.

Чтобы начать музыку с самого начала, вы должны вызвать метод reset (). Его синтаксис приведен ниже. mediaPlayer.reset ();

Помимо метода start и pause, этот класс предоставляет и другие методы для лучшей работы с аудио / видео файлами.

2.9 Переход в режим разработчика в мобильных устройствах

Android предоставляет множество способов управления воспроизведением аудио / видео файлов и потоков. Одним из таких способов

является использование класса MediaPlayer.

Android предоставляет класс MediaPlayer для доступа к встроенным сервисам медиаплеера, таким как воспроизведение аудио, видео и т. Д. Чтобы использовать MediaPlayer, мы должны вызвать статический метод create () этого класса. Этот метод возвращает экземпляр класса MediaPlayer. Его синтаксис выглядит следующим образом –MediaPlayer mediaPlayer = MediaPlayer.create (this, R.raw.song);

Второй параметр – это название песни, которую вы хотите воспроизвести. Вы должны создать новую папку в вашем проекте с именем raw и поместить в нее музыкальный файл.

Создав объект Mediaplayer, вы можете вызвать несколько методов для запуска или остановки музыки. Эти методы перечислены ниже.

mediaPlayer.start ();

mediaPlayer.pause ().

При вызове метода start () музыка начнет воспроизводиться с самого начала. Если этот метод вызывается снова после метода pause (), музыка начинает воспроизводиться с того места, где она была оставлена, а не с начала.

Чтобы начать музыку с самого начала, вы должны вызвать метод reset (). Его синтаксис приведен ниже:

– mediaPlayer.reset ();

Помимо метода start и pause, этот класс предоставляет и другие методы для лучшей работы с аудио / видео файлами. Эти методы перечислены ниже –

Метод и описание:

– isPlaying - этот метод просто возвращает истину / ложь, указывая, что песня воспроизводится или нет;

– seekToposition -этот метод принимает целое число и перемещает песню в эту конкретную позицию на миллисекунду;

– getCurrentPosition -этот метод возвращает текущую позицию песни в миллисекундах;

– getDuration Этот метод возвращает общую продолжительность песни в миллисекундах;

– сброс – этот метод сбрасывает медиаплеер;

– release Этот метод освобождает любой ресурс, связанный с объектом MediaPlayer;

– setVolumefloatleftVolume, floatrightVolume этот метод устанавливает громкость вниз для этого игрока;

– setDataSourceFileDescriptorfd -этот метод устанавливает источник данных аудио / видео файла;

– selectTrackintindex -этот метод принимает целое число и выбирает трек из списка по этому конкретному индексу;

– getTrackInfo -этот метод возвращает массив информации о дорожке.

2.9.1 Описание шагов работы мобильного приложения

Вот пример, демонстрирующий использование класса MediaPlayer. Он создает базовый медиаплеер, который позволяет перематывать, перематывать, воспроизводить и приостанавливать песню.

Чтобы поэкспериментировать с этим примером, вам нужно запустить его на реальном устройстве, чтобы услышать звук.

Описание шагов:

– вы будете использовать Android Studio IDE для создания приложения Android в пакете com.example.sairamkrishna.myapplication;

– измените файл src / MainActivity.java, чтобы добавить код MediaPlayer;

– измените res / layout / activity_main, чтобы добавить соответствующие компоненты XML;

– создайте новую папку в MediaPlayer с именем как raw и поместите в нее музыкальный файл mp3 с именем song.mp3;

– запустите приложение и выберите работающее устройство Android, установите на него приложение и проверьте результаты.

Android SDK предоставляет класс MediaPlayer для доступа к Android во встроенных сервисах медиаплеера, таких как воспроизведение аудио, видео и т. д. В этом руководстве я использую следующие функции этого класса для управления аудиоплеером [14].

Ниже приводится содержимое измененного основного файла активности src / MainActivity.java.

```
MediaPlayer mp = new MediaPlayer();
// Set data source -
setDataSource("/sdcard/path_to_song");
// Play audio
mp.start();
// Pause audio
mp.pause();
// Reset mediaplayer
mp.reset();
// Get song length duration - in milliseconds
mp.getDuration();
// Get current duration - in milliseconds
mp.getCurrentDuration();
// Move song to particular second - used for Forward or Backward
mp.seekTo(positon); // position in milliseconds
// Check if song is playing or not
mp.isPlaying(); // returns true or false
```

Настройка параметров разработчика на устройстве

Приложение «Настройки» на Android содержит экран «Параметры разработчика», который позволяет настраивать поведение системы, помогающее вам профилировать и отлаживать производительность вашего приложения. Например, вы можете включить отладку по USB, захватить отчет об ошибках, включить визуальную обратную связь для касаний, поверхности окон флеш-памяти при их обновлении, использовать графический процессор для рендеринга 2D-графики и многое другое.

2.9.2 Включить параметры разработчика и отладку

Список параметров разработчика может варьироваться в зависимости от версии Android (Рисунок 2.8).

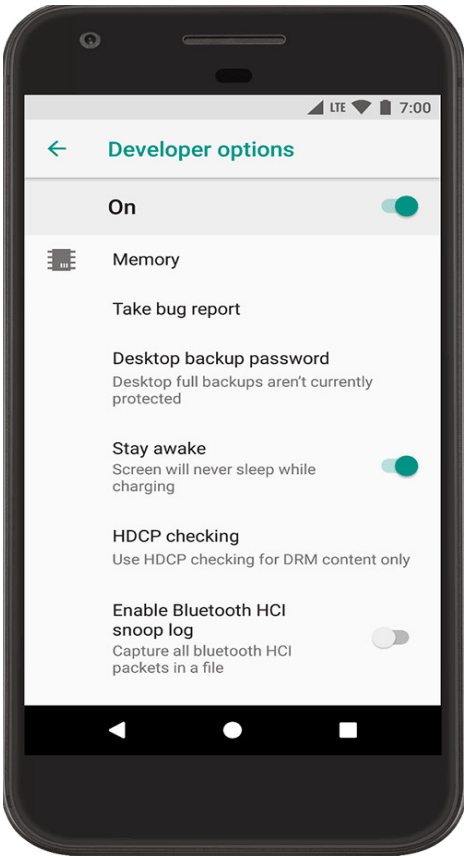

Рисунок 2.9 – Параметры разработчика

На Android 4.1 и ниже, экран параметров разработчика доступен по умолчанию. На Android 4.2 и выше, вы должны включить этот экран следующим образом:

– откройте приложение «Настройки»;
– (только на Android 8.0 или выше) Выберите Система;
– прокрутите вниз и выберите о телефоне;
– прокрутите вниз и коснитесь Номер сборки 7 раз;
– вернитесь к предыдущему экрану, чтобы найти параметры

разработчика внизу.

В верхней части экрана параметров разработчика вы можете включать и выключать параметры (рисунок 1). Вы, вероятно, хотите сохранить это. Если этот параметр отключен, большинство параметров отключено, за исключением тех, которые не требуют связи между устройством и компьютером разработчика.

Далее следует немного прокрутить вниз и включить отладку по USB. Это позволяет Android Studio и другим инструментам SDK распознавать ваше устройство при подключении через USB, поэтому вы можете использовать отладчик и другие инструменты. Остальная часть этой страницы описывает некоторые другие опции, доступные а этом экране. Общие настройки в Android 8.0 и более поздних версиях вы можете нажать плитку «Быстрые настройки разработчика», чтобы добавить выбранные параметры разработчика на панель «Быстрые настройки». После выбора одной или нескольких доступных плиток (рисунок 2) откройте панель «Быстрые настройки» и коснитесь карандаша, чтобы войти в режим редактирования. Затем перетащите плитки разработчика из панели плиток на панель «Быстрые настройки» и снова коснитесь карандаша, чтобы выйти из режима редактирования (Рисунок 2.10).

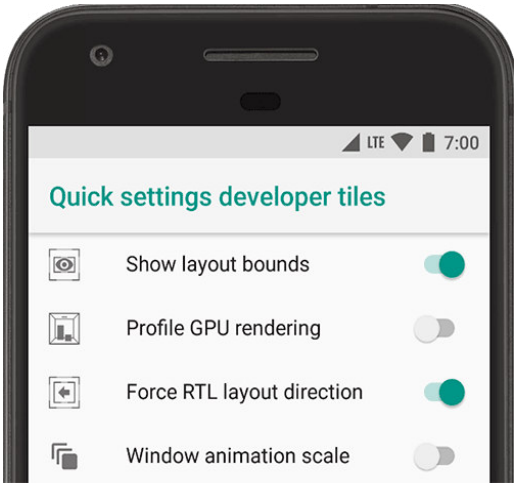

Рисунок 2.10 – Добавить в панель быстрых настроек

Другие общие параметры включают в себя следующее: Память: (на Android 8.0 и выше) Отображение статистики памяти, такой как среднее использование памяти, производительность памяти, общий объем доступной памяти, средний объем используемой памяти, сколько свободной памяти доступно и сколько памяти используется приложениями. Взять отчет об ошибке: Получить копию текущих файлов журнала устройства, чтобы поделиться с кем-то. Когда вы получите уведомление о том, что отчет об ошибке готов, нажмите на уведомление, чтобы поделиться им. Демо-режим системного интерфейса. Упрощает создание чистых снимков экрана благодаря отображению стандартной предварительно настроенной

панели уведомлений, в которой не отображаются уведомления или предупреждения о разряде батареи. Включить демонстрационный режим позволяет изменить внешний вид строки состояния с помощью команд демонстрационного режима adb. Или вы можете использовать Show Demo Mode для скрытия уведомлений и отображения предустановленной строки состояния. Примечание. Команды режима adb demo mode могут работать не на всех устройствах, поскольку они не проверены во время сертификационного тестирования Android. Они проверены только для работы на устройствах Nexus и Pixel. Пароль резервного копирования рабочего стола: задает пароль резервного копирования, чтобы вы могли использовать команды adb для резервного копирования и восстановления приложений и данных устройства под защитой паролем.

«Бодрствовать»: заставляет ваш экран включаться каждый раз, когда вы подключаете его. Включить журнал отслеживания интерфейса хост-контроллера Bluetooth (HCI): захватывает все пакеты Bluetooth HCI в файле, хранящемся по адресу /sdcard/btsnoop_hci.log. Вы можете получить пакеты, а затем использовать программу типа Wireshark для анализа и устранения неполадок в информации (Рисунок 2.11).

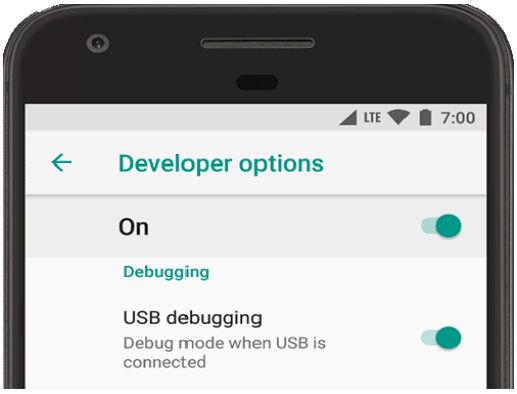

Рисунок 2.11 – Параметры отладки включены

Параметры отладки позволяют настроить отладку на устройстве и установить связь между устройством и компьютером разработчика. Включите отладку по USB (рисунок 3), чтобы ваше устройство Android могло обмениваться данными с вашим компьютером разработки через Android Debug Bridge (adb). Параметр «Ожидание отладчика» будет недоступен до тех пор, пока вы не выберете «Выбор приложения отладки», чтобы выбрать приложение для отладки. Если вы включите «Ожидание отладчика», то выбранное приложение ожидает подключения отладчика, прежде чем оно выполнится. Другие варианты отладки включают следующее:

Постоянно хранить данные журнала на устройстве: выберите тип сообщений журнала, которые вы хотите постоянно хранить на устройстве. Опции отключены, все, кроме радио или только ядра. Выберите приложение для определения местоположения: используйте эту опцию, чтобы подделать

местоположение GPS устройства, чтобы проверить, работает ли ваше приложение так же в других местах. Чтобы использовать эту опцию, загрузите и установите приложение для определения местоположения GPS (Рисунок 2.12).

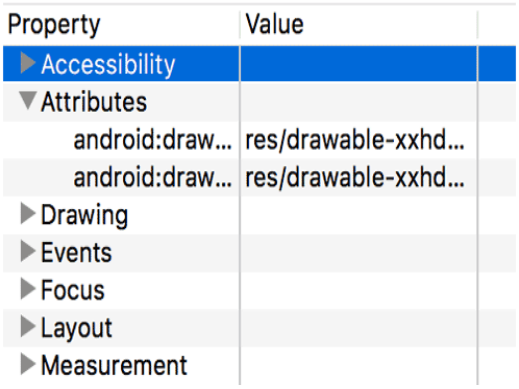

Рисунок 2.12 – Просмотр атрибутов

Включить проверку атрибутов вида: сохраняет информацию об атрибутах вида в переменной-члене mAttributes экземпляра View, чтобы ее можно было использовать для отладки. Вы можете получить доступ к информации об атрибутах через пользовательский интерфейс Layout Inspector, как показано на рисунке 4 (без этого элемент «Атрибуты» недоступен).

Включить отладочные слои графического процессора: доступно на устройствах под управлением Android 9 (уровень API 28) и выше, включите эту опцию, чтобы разрешить загрузку слоев проверки Vulkan из локального хранилища устройства. Чтобы узнать больше, прочитайте слои проверки Vulkan на Android (Рисунок 2.13).

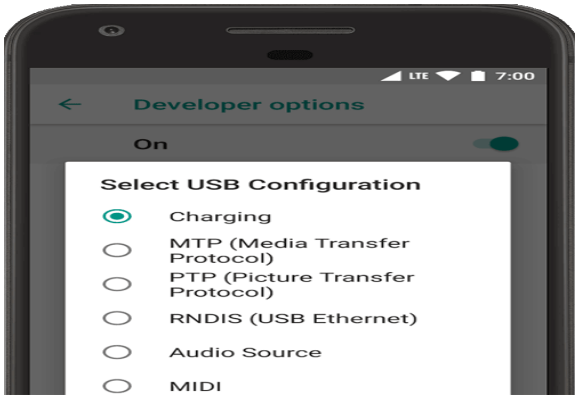

Рисунок 2.13 – Выбор конфигурации USB

Параметры сети предоставляют способы настройки параметров Wi-Fi и DHCP. Нажмите «Выбрать конфигурацию USB», чтобы указать, как компьютер должен идентифицировать устройство. Как показано на рисунке 5, вы можете настроить устройства только для зарядки, для передачи файлов

(MTP), для передачи изображений (PTP), для использования вашего мобильного интернета на ПК (RNDIS) или для передачи аудио или MIDI-файлов. Нажмите «Версия Bluetooth AVRCP» и выберите профильную версию, которую вы хотите использовать для управления всем аудио / видео оборудованием Bluetooth, к которому у вашего устройства есть доступ. Кроме того, для точной настройки воспроизведения звука на устройстве нажмите и установите следующие параметры:

– аудиокодек Bluetooth: отрегулируйте качество звука (кодек) ваших устройств, выбрав один из следующих кодеков;

– SBC: передача данных на устройства вывода звука Bluetooth, такие как наушники и динамики;

– AAC: лучшее качество звука от проводных устройств, чем у MP3 при аналогичных скоростях передачи данных;

– aptX: чистый беспроводной звук в высококачественных смартфонах, колонках, звуковых панелях, наушниках и планшетах;

– aptX HD: потоковая передача высокого разрешения на устройства Bluetooth;

– LDAC: обеспечивает высококачественное прослушивание музыки через беспроводные соединения.

Включить дополнительные кодеки и отключить дополнительные кодеки. Если у вас установлены дополнительные реализации кодеков, используйте эти параметры для их включения и отключения. Диапазон аудиосэмплов Bluetooth: отрегулируйте количество аудиосэмплов в секунду, выбрав частоту дискретизации аудиокодека. Более высокие частоты дискретизации используют больше ресурсов. Аудио биты Bluetooth на семпл: Установите количество битов информации в каждом семпле. Чем выше скорость передачи, тем лучше звук, но файл сэмпла больше. Режим аудио канала Bluetooth: выберите моно или стерео. Кодек Bluetooth Audio LDAC: оптимизируйте звук, чтобы повысить качество звука, сбалансировать качество звука и соединения, повысить качество соединения или использовать адаптивную скорость передачи данных для баланса звука и качества соединения.

В следующем списке описаны другие способы настройки Wi-Fi и DHCP:

– сертификация беспроводного дисплея. Включает расширенные элементы управления конфигурацией и настройки для сертификации беспроводного дисплея в соответствии со спецификациями, указанными в Спецификации дисплея Wi-Fi Alliance Wi-Fi. Сертификация распространяется на Android 4.4 (уровень API 19) и выше;

– включить подробное ведение журнала Wi–Fi. Повышает уровень ведения журнала Wi–Fi для каждой беспроводной сети (SSID), к которой вы подключаетесь, в соответствии с относительным уровнем принимаемого сигнала (RSSI). Для получения дополнительной информации о журналах см. Запись и просмотр журналов с помощью Logcat.

Агрессивная передача обслуживания Wi–Fi на сотовую связь: при

низком уровне сигнала Wi–Fi повышает эффективность передачи данных (Рисунок 2.14).

Рисунок 2.14 – Расположение указателя

Показывать касания, чтобы отображать касания при касании экрана. Под вашим пальцем или стилусом появляется круг, который следует за вами при перемещении по экрану. Касание работает как указатель, когда вы записываете видео на вашем устройстве. Включите указатель местоположения, чтобы показать местоположение указателя (касание) на устройстве с перекрестием. В верхней части экрана появляется полоса для отслеживания координат перекрестия. При перемещении указателя координаты на панели отслеживают расположение перекрестия, и путь указателя отображается на экране [15].

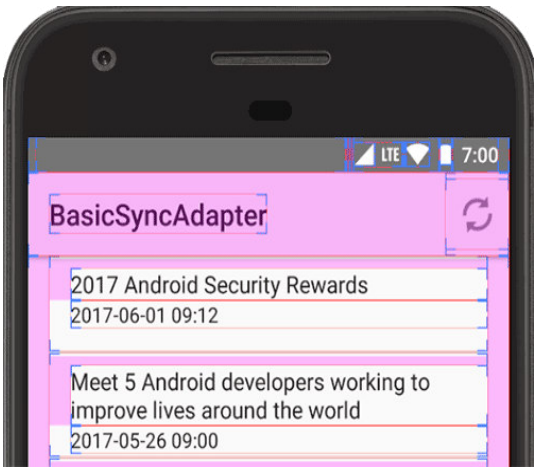

Рисунок 2.15 – Конструкции пользовательского интерфейса

Параметры рисования предоставляют визуальные подсказки о пользовательском интерфейсе приложения и о том, как оно работает. Включите Show Layout Bounds, чтобы показать границы клипа, поля и другие конструкции пользовательского интерфейса на устройстве, как показано на (рисунке 2.15). Другие параметры рисования включают в себя следующее: Принудительное направление макета RTL: принудительно указывает направление макета экрана справа налево (RTL) или слева направо (по

умолчанию). Масштаб анимации окна: устанавливает скорость воспроизведения анимации окна, чтобы вы могли проверить ее производительность на разных скоростях. Чем ниже шкала, тем выше скорость. Масштаб анимации перехода: устанавливает скорость воспроизведения анимации перехода, чтобы вы могли проверить ее производительность на разных скоростях. Чем ниже шкала, тем выше скорость. Имитация вторичных дисплеев: создает вторичный дисплей в качестве наложения на устройстве. Это полезно при поддержке дополнительных дисплеев с API представления. Смотрите вторичные дисплеи (Рисунок 2.16).

Рисунок 2.16 – Цветовое пространство дейтераномалии

Опции рендеринга с аппаратным ускорением позволяют оптимизировать ваше приложение для его целевых аппаратных платформ за счет использования аппаратных опций, таких как графический процессор, аппаратные уровни и мультисэмплирование сглаживания (MSAA). Нажмите «Имитация цветового пространства», чтобы изменить цветовую схему всего пользовательского интерфейса устройства. Опции относятся к типам дальтонизма. Доступны следующие варианты: «Отключено» (без смоделированной цветовой схемы), «Монохромность» (черный, белый и серый), «Дейтераномалия» (красно-зеленый), «Протаномалия» (красно-зеленый) и «Тританомалия» (сине-желтый). «Протаномалия» относится к красно-зеленой дальтонике со слабостью в красных тонах, а дейтераномалия» (показано на рисунке 8) относится к красно-зеленой дальтонике со слабостью

в зеленых тонах. Если вы делаете снимки экрана в смоделированном цветовом пространстве, они выглядят нормальными, как если бы вы не меняли цветовую схему. Вот некоторые другие способы использования аппаратных опций: Установить графический процессор визуализации: изменить графический движок Open GL по умолчанию на графический движок Open GL Skia. Принудительный рендеринг в графическом процессоре: заставляет приложения использовать графический процессор для 2D-рисования, если они были написаны без графического отображения по умолчанию. Показать обновления вида графического процессора: отображает любой экранный элемент, нарисованный с помощью графического процессора. Отладка GPU overdraw: отображает цветовое кодирование на вашем устройстве, чтобы вы могли визуализировать, сколько раз один и тот же пиксель был нарисован в одном кадре. Визуализация показывает, где ваше приложение может выполнять больше рендеринга, чем необходимо. Для получения дополнительной информации см. Визуализация перерисовки графического процессора. Отладка непрямоугольных операций с клипами: отключает область отсечения на холсте для создания необычных (непрямоугольных) областей холста. Обычно область отсечения не позволяет рисовать что-либо за пределами области круглого отсечения. Force 4x MSAA: включает сглаживание нескольких образцов (MSAA) в приложениях Open GL ES 2.0. Отключить наложения HW: использование аппаратного наложения позволяет каждому приложению, отображающему что-либо на экране, использовать меньше вычислительной мощности. Без наложения приложение разделяет видеопамять и должно постоянно проверять наличие столкновений и отсечек, чтобы получить правильное изображение. Проверка использует много вычислительной мощности (Рисунок 2.17).

Рисунок 2.17 – Барное представление

Включите параметр «Отключить маршрутизацию аудио через USB», чтобы отключить автоматическую маршрутизацию на внешние аудиоустройства, подключенные к компьютеру через порт USB.

Автоматическая маршрутизация может мешать работе приложений, поддерживающих USB.

Параметры мониторинга предоставляют визуальную информацию о производительности приложения, такую как операции с длинными потоками и графическим процессором. Нажмите «Профиль GPU Rendering», а затем «На экране» в виде полос, чтобы отобразить профиль визуализации GPU в виде полос (Рисунок 2.18).

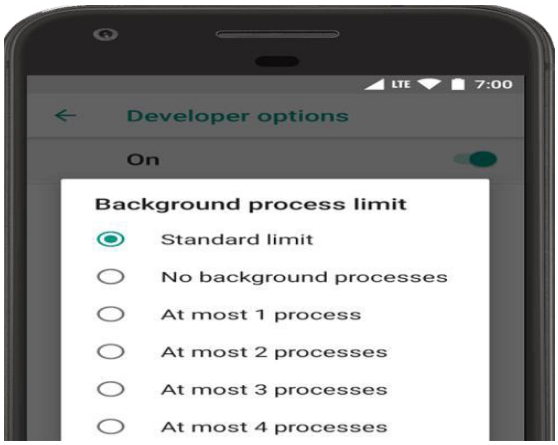

Рисунок 2.18 – Установка пределов фонового процесса

Параметры приложения помогут вам понять, как ваше приложение работает на целевом устройстве. Нажмите Предел фоновых процессов, чтобы установить количество процессов, которые могут выполняться в фоновом режиме за один раз. Возможные настройки показаны на рисунке 10. Нажмите «Сбросить ограничение скорости ShortcutManager во время тестирования, чтобы фоновые приложения могли продолжать вызывать API-интерфейсы ярлыков до тех пор, пока ограничение скорости не будет достигнуто снова (Рисунок 2.19). Для получения дополнительной информации о ярлыках и ограничениях скорости см. ShortcutManager [16].

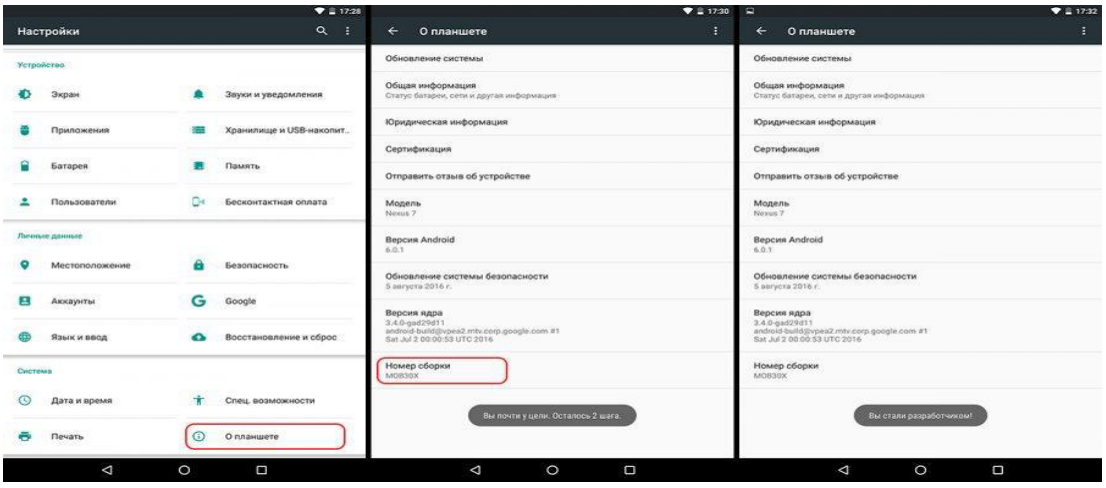

Рисунок 2.19 – Выбор конфигурации мобильных устройств

2.10 Практическая часть: создания музыкального плеера в программе Android Studio

Первая страница мобильного приложения начинается с чтения файлов формата .mp3. Во-первых, доступ к этим файлам будет разрешен пользователем, после доступа файлы будут перечислены как страница мобильного приложения.

XML-файл первой страницы был сохранен в activity_main.xml. Код выглядит следующим образом (Рисунок 2.20):

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
android:padding="10dp">
<ListView
android:layout_width="match_parent"
android:layout_height="match_parent"
android:id="@+id/mySongListView">
</ListView>
</RelativeLayout>
```

Рисунок 2.20 – код activity_main.xml

Рабочий процесс в программе Android Studio (Рисунок 2.21)

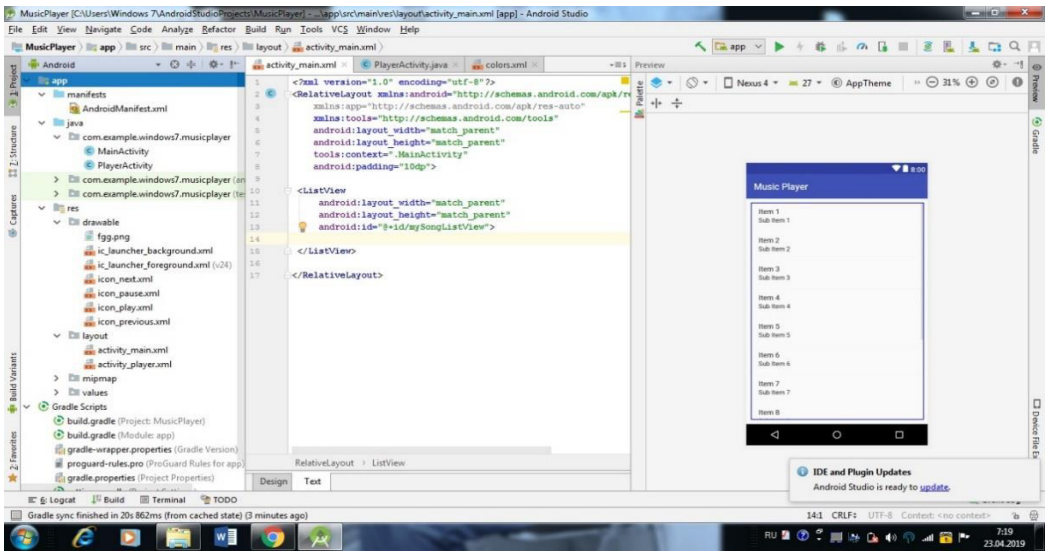

Рисунок 2.21 – Рабочий процесс в программе Android Studio

Код доступа зарезервирован для скриптов Gradle-> Build Gradle (Рисунок 2.22)

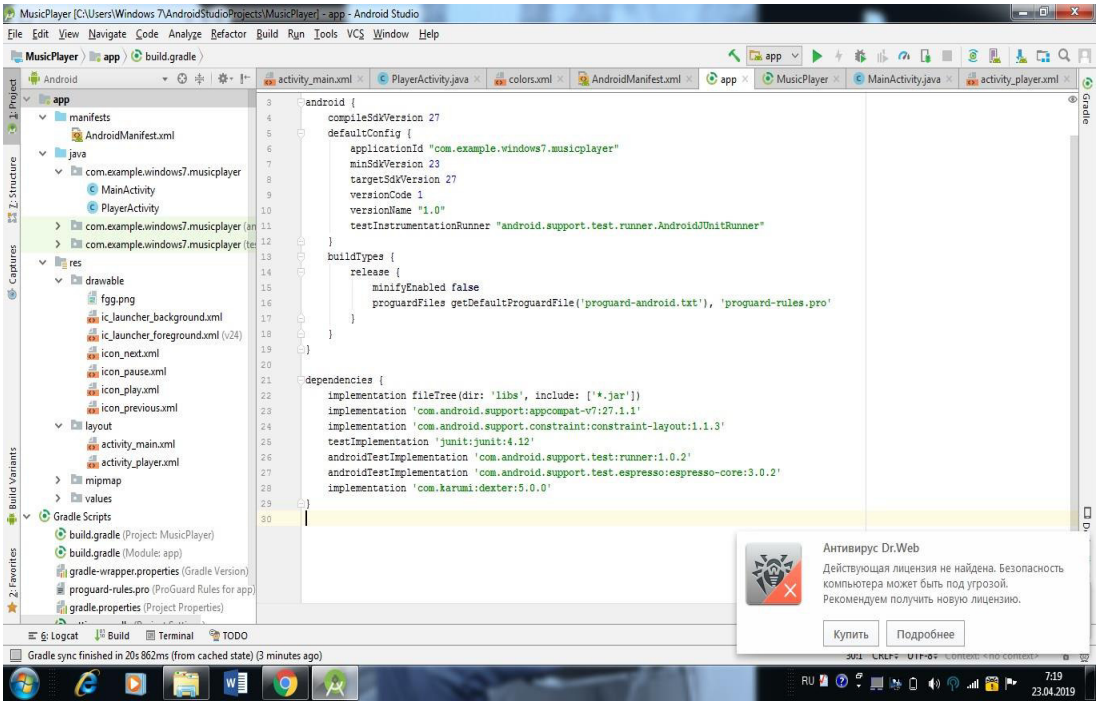

Рисунок 2.22 – Рабочий процесс организации кода доступа

Теперь поговорим о логической части этой страницы: Класс Java является логической частью мобильного приложения. Класс страницы activity_main хранится в файле MainActivity.java, его код выглядит следующим образом (Рисунок 2.23):

```
package com.example.windows7.musicplayer;

import android.Manifest;
import android.content.Intent;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import com.karumi.dexter.Dexter;
import com.karumi.dexter.PermissionToken;
import com.karumi.dexter.listener.PermissionDeniedRespons
import com.karumi.dexter.listener.PermissionGrantedRespon
import com.karumi.dexter.listener.PermissionRequest;
import com.karumi.dexter.listener.single.PermissionListener;

import java.io.File;
import java.util.ArrayList;

public class MainActivity  extends AppCompatActivity
{
```

Рисунок 2.23 – код MainActivity.java

Продолжение данного кода иммет вид (Рисунок 2.24):

```
ListView myListViewForSongs;
String[] items;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
myListViewForSongs = (ListView) findViewById(R.id.mySongListView);
runtimePermission();
}
public void runtimePermission()
{
Dexter.withActivity(this)
.withPermission(Manifest.permission.READ_EXTERNAL_STORAGE)
.withListener(new PermissionListener() {
@Override
public void onPermissionGranted(PermissionGrantedResponse response) {
```

Рисунок 2.24 – код MainActivity.java

Продолжение кода имеет вид (Рисунок 2.25):

```
display();
}
@Override
public void onPermissionDenied(PermissionDeniedResponse response) {
}
@Override
public void onPermissionRationaleShouldBeShown(PermissionRequest permission,
PermissionToken token) {
token.continuePermissionRequest();
}
}) .check();
}
public ArrayList<File> findSong(File file){
ArrayList<File> arrayList = new ArrayList<>();
File[] files = file.listFiles();
for(File singleFile: files){
if(singleFile.isDirectory() && !singleFile.isHidden()) {
arrayList.addAll(findSong(singleFile));
}
else {
if(singleFile.getName().endsWith(".mp3") ||
singleFile.getName().endsWith(".wav")) {
arrayList.add(singleFile);
}
}
```

Рисунок 2.25 – код MainActivity.java

продолжение кода имеет вид (Рисунок 2.26):

```
}
return arrayList;
}
void display() {
final ArrayList<File> mySongs =
findSong(Environment.getExternalStorageDirectory());
items = new String[mySongs.size()];
for (int i=0 ; i<mySongs.size();i++) {
items[i]= mySongs.get(i).getName().toString().replace(".mp3","").replace(".wav",
"");
}
ArrayAdapter<String>myAdapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1,items);
myListViewForSongs.setAdapter(myAdapter);
```

Рисунок 2.26 – код MainActivity.java

продолжение кода имеет вид (Рисунок 2.27):

```
myListViewForSongs.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> adapterView, View view, int i, long l ) {
String songName = myListViewForSongs.getItemAtPosition(i).toString();

startActivity(new Intent(getApplicationContext(),PlayerActivity.class)
.putExtra("songs",mySongs).putExtra("songname", songName).putExtra("pos",i));
}
});
}
}
```

Рисунок 2.27 – код MainActivity.java

После выбора нужного файла открывается следующая страница, и страница выполняется в XML-файле activity_player следующим образом (Рисунок 2.28):

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:orientation="vertical"
android:weightSum="10"
android:layout_height="match_parent"
tools:context=".PlayerActivity">

<LinearLayout
android:layout_width="match_parent"
android:layout_height="350dp"
android:gravity="center"
android:orientation="vertical">

<ImageView
android:layout_width="250dp"
android:layout_height="250dp"
android:src="@drawable/fgg"
/>

<TextView
android:layout_width="match_parent"
```

Рисунок 2.28 – код activity_player

продолжение данного кода имеет вид (Рисунок 2.29):

```
android:layout_height="wrap_content"
android:text="Song name"
android:textAlignment="center"
android:textSize="22sp"
android:layout_marginTop="20dp"
android:singleLine="true"
android:id="@+id/songLable"
android:marqueeRepeatLimit="marquee_forever"
android:ellipsize="marquee"
android:scrollHorizontally="true"
android:textColor="@color/colorPrimary"
/>
</LinearLayout>
```

Рисунок 2.29 – код activity_player

продолжение кода activity player (Рисунок 2.30):

```
android:id="@+id/seekbar"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true"
android:layout_marginBottom="40dp"
/>
<Button
android:id="@+id/pause"
android:layout_width="60dp"
android:layout_height="60dp"
android:layout_centerHorizontal="true"
android:layout_marginHorizontal="5dp"
android:background="@drawable/icon_pause" />
<Button
android:id="@+id/next"
android:layout_width="40dp"
```

Рисунок 2.30 – код activity_player

продолжение кода activity_player (Рисунок 2.31):

```
android:layout_height="40dp"
android:layout_marginTop="15dp"
android:layout_toRightOf="@+id/pause"
android:background="@drawable/icon_next" />
<Button
android:id="@+id/previous"
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_marginTop="15dp"
android:layout_toLeftOf="@+id/pause"
android:background="@drawable/icon_previous"
/> </RelativeLayout>  </LinearLayout>  </LinearLayout>
```

Рисунок 2.31 – код activity player

Рабочий процесс разработки Музыкального плеера. Рабочий процесс именно на платформе Android очень удобно и экономит много времени. Можно писать код в левом углу экрана и одновременно смотреть в правую часть экрана, чтобы видеть какие изменения мы вносим (Рисунок 2.32).

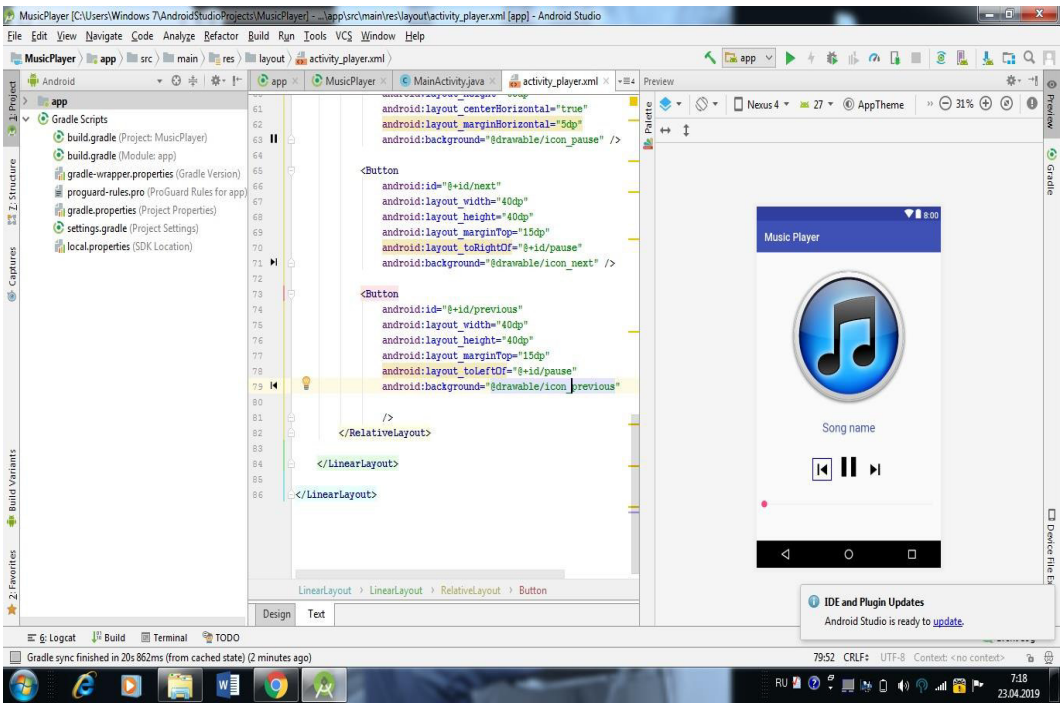

Рисунок 2.32 – Рабочий процесс разработки Музыкального плеера

Логический раздел хранится в файле PlayerActivity.java следующим образом package com.example.windows7.musicplayer (Рисунок 2.33):

```
import android.widget.Button;
import android.widget.SeekBar;
import android.widget.TextView;
import java.io.File;
import java.util.ArrayList;
import static android.graphics.PorterDuff.Mode.SRC_IN;
public class PlayerActivity  extends AppCompatActivity {
Button btn_next,btn_previous,btn_pause;
TextView songTextLabel;
SeekBar songSeekbar;
String sname;
static MediaPlayer myMediaPlayer;
int position;
ArrayList<File> mySongs;
Thread updateseekbar;
```

Рисунок 2.33 – код PlayerActivity.java

продолжение кода PlayerActivity.java (Рисунок 2.34):

```
songSeekbar = (SeekBar)findViewById(R.id.seekbar);
updateseekbar = new Thread(){

@Override
public void run() {
int totalDuration=  myMediaPlayer.getDuration();
int currentposition = 0;
while (currentposition<totalDuration){
try {
sleep(500);
currentposition=myMediaPlayer.getCurrentPosition();
songSeekbar.setProgress(currentposition);
}
catch (InterruptedException e){
e.printStackTrace();
} } } };
if(myMediaPlayer!=null){
```

Рисунок 2.34 – код PlayerActivity.java

Продолжение кода PlayerActivity.java (Рисунок 2.35):

```
myMediaPlayer.stop();
myMediaPlayer.release();
}
Intent i=getIntent();
Bundle bundle=i.getExtras();
mySongs = (ArrayList) bundle.getParcelableArrayList("songs");
sname = mySongs.get(position).getName().toString();
String songName= i.getStringExtra("songname");
songTextLabel.setText(songName);
songTextLabel.setSelected(true);
position = bundle.getInt("pos",0);
Uri u =Uri .parse(mySongs.get(position).toString());
myMediaPlayer  = MediaPlayer.create(getApplicationContext(),u);
myMediaPlayer.start();
songSeekbar.setMax(myMediaPlayer.getDuration());
updateseekbar.start();
songSeekbar.getProgressDrawable().setColorFilter(getResources().getColor(R.color
.colorPrimary), PorterDuff.Mode.MULTIPLY);
songSeekbar.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
@Override
public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser)
{
}
@Override
```

Рисунок 2.35 – код PlayerActivity.java

Продолжение кода PlayerActivity.java (Рисунок 2.36)

```
public void onStartTrackingTouch(SeekBar seekBar) {
}
@Override
public void onStopTrackingTouch(SeekBar seekBar) {
myMediaPlayer.seekTo(seekBar.getProgress());
}
});
btn_pause.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
songSeekbar.setMax(myMediaPlayer.getDuration());
if (myMediaPlayer.isPlaying()) {
btn_pause.setBackgroundResource(R.drawable.icon_play);
myMediaPlayer.pause();
}
else {
btn_pause.setBackgroundResource(R.drawable.icon_pause);
```

Рисунок 2.36 – код PlayerActivity.java

Часть кода myMediaPlayer.start() является заключительным и имеет вид(Рисунок 2.37):

```
@Override
public void onClick(View v) {
myMediaPlayer.stop();
myMediaPlayer.release();
position=((position+1)%mySongs.size());
Uri u=Uri.parse( mySongs.get(position).toString());
myMediaPlayer=MediaPlayer.create(getApplicationContext(),u);
sname=mySongs.get(position).getName().toString();
songTextLabel.setText(sname);
myMediaPlayer.start();
}
});
btn_previous.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
myMediaPlayer.stop();
myMediaPlayer.release();
position=((position-1)<0)?(mySongs.size()-1):(position-1);
Uri u=Uri.parse(mySongs.get(position).toString());
myMediaPlayer=MediaPlayer.create(getApplicationContext(),u);
sname=mySongs.get(position).getName().toString();
songTextLabel.setText(sname);
myMediaPlayer.start();
}
});
```

Рисунок 2.37 – код PlayerActivity.java

Существуют способы, с помощью которых приложения могут принять некоторые характеристики как веб-, так и нативных приложений. В общем, проблемы загрузки в мобильных устройствах требуют эффективных приложений для принятия четко определенных стратегий, таких как:

– использование минимальных HTTP-запросов;

– выполнение максимально возможной обработки на стороне сервера;

– как правило, минимизация данных и медиа-контента.

Когда эти методы будут приняты, появится больше возможностей для того, чтобы сфокусировать разработку под конкретную платформу на создании легких интерфейсов с возможностью обработки на стороне сервера для разных платформ. Эта модель может позволить проектам максимально эффективно использовать ресурсы разработки, при этом обслуживая многопользовательские среды.

Для многих организаций развертывание как собственных, так и веб-приложений по-прежнему считается необходимым. Пользователи по-прежнему используют их обоих, и немного по-разному. Существуют также особые случаи, когда имеет смысл ориентироваться на одну или несколько платформ с помощью выделенных приложений, когда уникальные функции этой платформы имеют повышенную актуальность, например, благодаря высокому уровню поддержки Google Maps на платформе Android. Для разработчиков программного обеспечения доминирующими вопросами, которые следует учитывать при реализации проекта для рынка мобильной связи, темпы изменений на рынке мобильной связи делают стратегическое планирование очень трудным. Должна быть принята гибкая позиция для предоставления мобильных услуг при сохранении как можно более открытых будущих вариантов. В ближайшем будущем и веб-приложения, и нативные приложения будут по-прежнему иметь большое значение. Для потенциальных клиентов идея иметь возможность разработать веб-приложение один раз и развернуть его в многопользовательской среде, является привлекательной. Однако темпы изменений на рынке мобильной связи делают стратегическое планирование очень трудным. Поскольку конечные пользователи используют несколько сред и общаются по постоянно растущему диапазону каналов, мобильный Интернет, скорее всего, сможет достичь всех из них в долгосрочной перспективе. Но учитывая его нынешние ограничения, сосредоточиться исключительно на Интернете будет оставаться сложной задачей. Принято считать, что большинство неотъемлемых преимуществ нативных приложений в конечном итоге будут доступны из веб-приложений. На данный момент веб-приложения еще не могут идеально эмулировать нативные приложения. Нативные приложения взаимодействуют напрямую с операционной системой, а веб-приложения взаимодействуют с браузером, который взаимодействует с ОС. Этот дополнительный слой, через который должны проходить веб-приложения, делает их немного медленнее и грубее, чем нативные приложения.

3. Проектирование

3.1 Диаграмма прецедентов

Диаграмма вариантов использования (англ. use case diagram) в UML — диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне (Рисунок 3.1).

Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе [27].

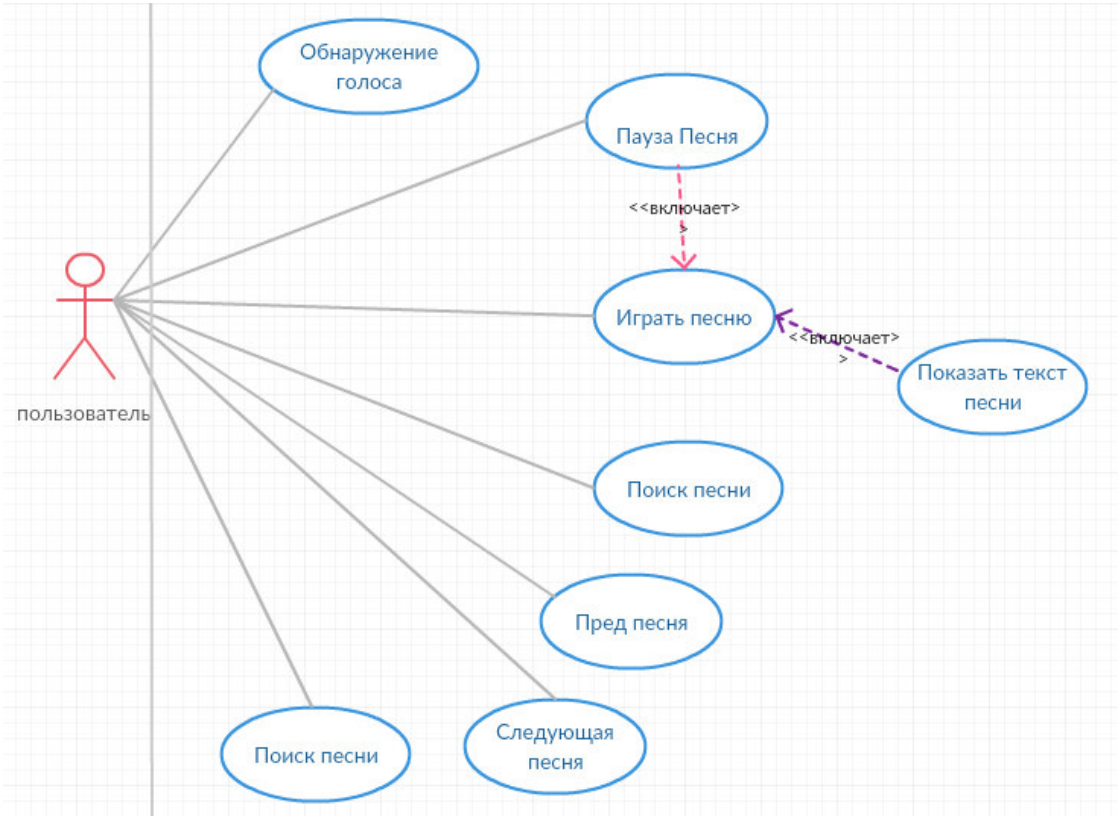

Рисунок 3.1 – Диаграмма прецедентов

3.2 Диаграмма классов

Диаграмма классов (англ. Static Structure diagram)-структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их кооперацией, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования[24].

Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы (классов, типов и т. п.) Она содержит в себе также некоторые элементы поведения (например-операции), однако их динамика должна быть отражена на диаграммах других видов (диаграммах коммуникации, диаграммах состояний). Для удобства восприятия диаграмму классов можно также дополнить представлением пакетов, включая вложенные[24].

При представлении сущностей реального мира разработчику требуется отразить их текущее состояние, их поведение и их взаимные отношения. На каждом этапе осуществляется абстрагирование от маловажных деталей и концепций, которые не относятся к реальности (производительность, инкапсуляция, видимость и т. п.). Классы можно рассматривать с позиции различных уровней. Как правило, их выделяют три основных: аналитический уровень, уровень проектирования и уровень реализации[25]:

на уровне анализа класс содержит в себе только набросок общих контуров системы и работает как логическая концепция предметной области или программного продукта.

на уровне проектирования класс отражает основные проектные решения касательно распределения информации и планируемой функциональности, объединяя в себе сведения о состоянии и операциях.

на уровне реализации класс дорабатывается до такого вида, в каком он максимально удобен для воплощения в выбранной среде разработки; при этом не воспрещается опустить в нём те общие свойства, которые не применяются на выбранном языке программирования.

Диаграмма классов является ключевым элементом в объектно-ориентированном моделировании. На диаграмме классы представлены в рамках, содержащих три компонента:

– в верхней части написано имя класса. Имя класса выравнивается по центру и пишется полужирным шрифтом. Имена классов начинаются с заглавной буквы. Если класс абстрактный — то его имя пишется полужирным курсивом;

– Посередине располагаются поля (атрибуты) класса. Они выровнены по левому краю и начинаются с маленькой буквы;

– Нижняя часть содержит методы класса. Они также выровнены по левому краю и пишутся с маленькой буквы (Рисунок 3.2).

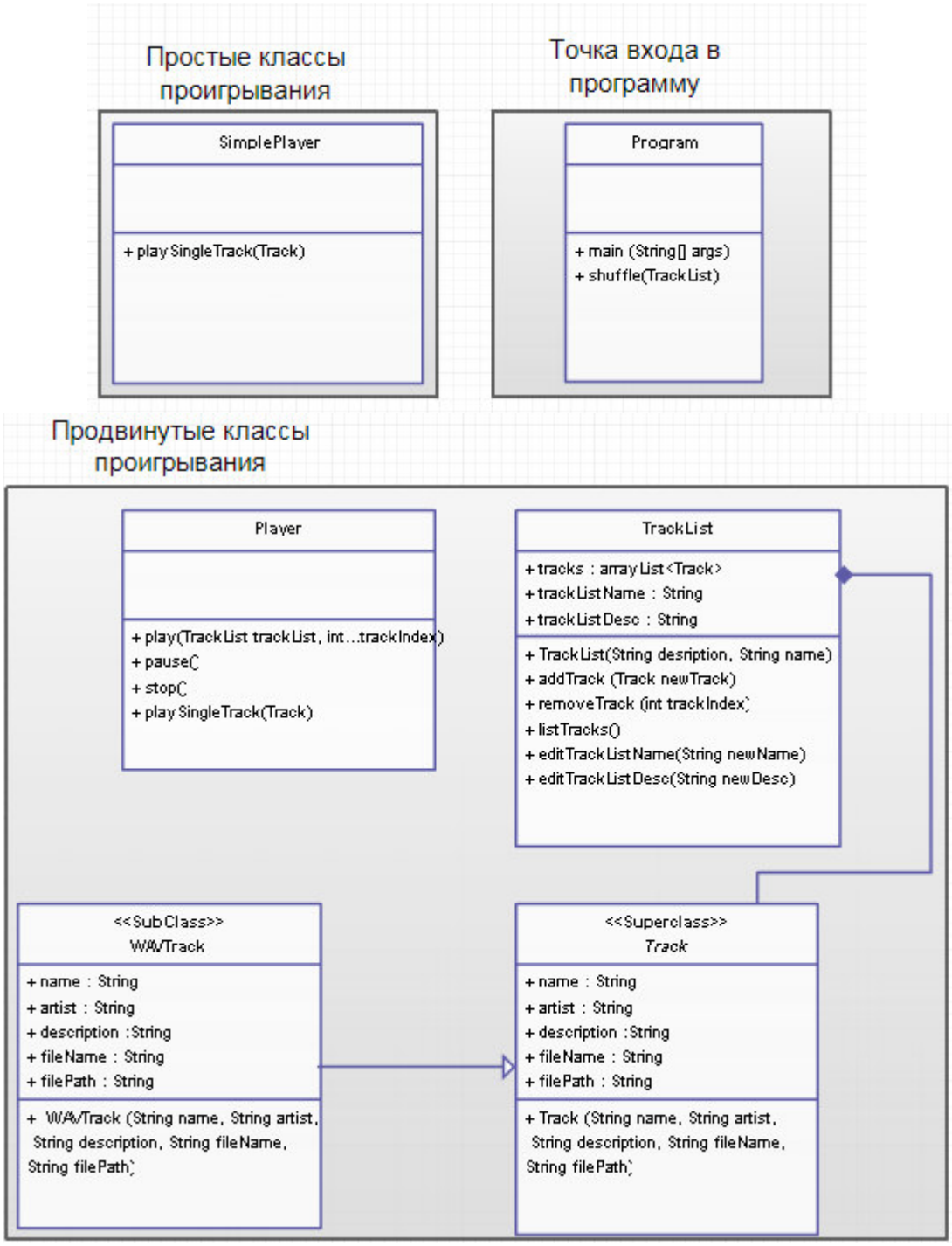

Рисунок 3.2 – Диаграмма классов

3.3 Диаграмма последовательностей

Диаграмма последовательности (англ. sequence diagram) - диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) ИС в рамках какого-либо определённого прецедента (отправка запросов и получение ответов). Используется в языке UML.

Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники с названиями объектов), вертикальные «линии жизни» (англ. lifeline), отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение им определенной функции (прямоугольники на пунктирной «линии жизни»), и стрелки, показывающие обмен сигналами или сообщениями между объектами (Рисунок 3.3).

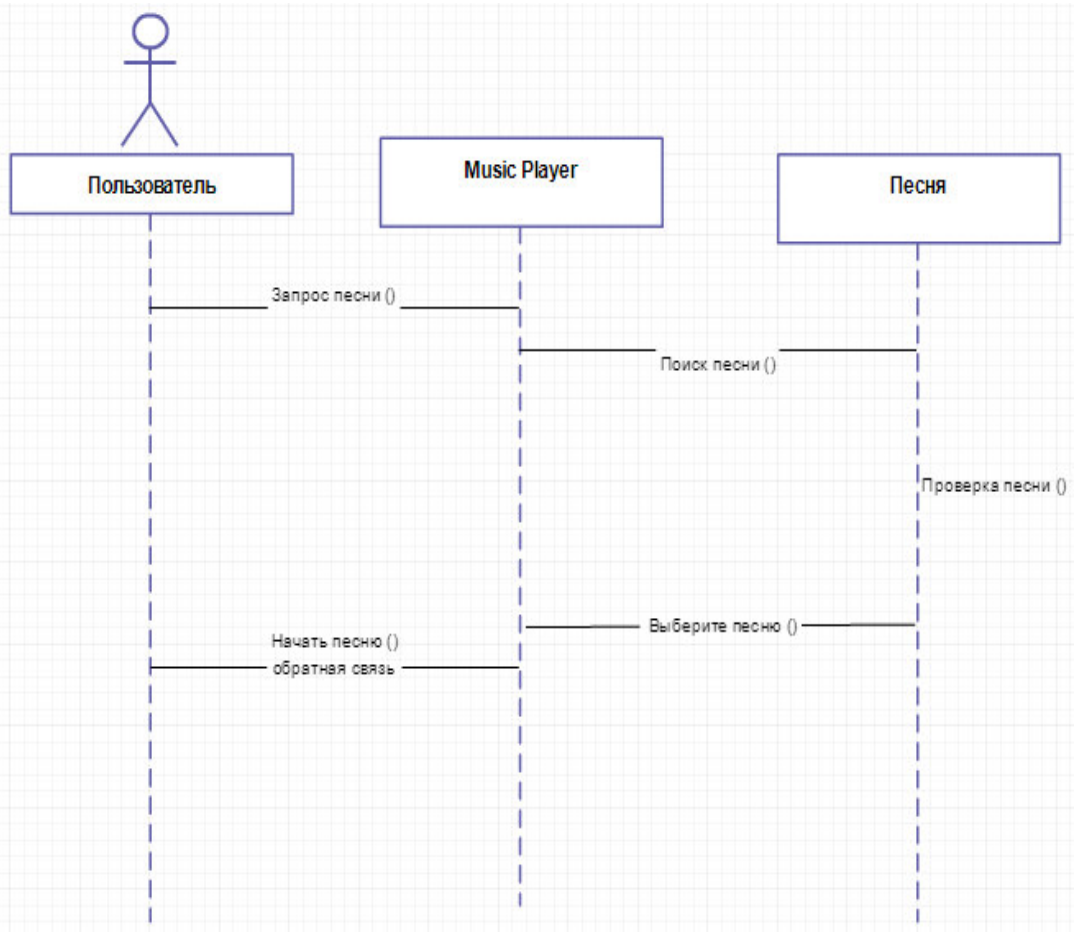

Рисунок 3.3 – Диаграмма последовательностей

3.4 Интерфейс пользователя

Android предоставляет компонент для воспроизведения аудио, это MediaPlayer. MediaPlayer может воспроизводить аудио и видео, с файлами имеющимися на вашем устройстве или из URL. Похоже на другие аудиопроигрыватели, которые вы знаете, MediaPlayer предоставляет методы упрвления звуков (control playback of audio/video) включая старт, пауза, перемотка вперед назад, вы так же можете вызвать MediaPlayer из сервиса. MediaPlayer это компонент не имеющий интерфейс, помогающий вам легко проигрывать музыкальные файлы, но для воспроизведения видео файла вам нужно его скомбинировать с SuffaceView чтобы появились изображения.

Далее простой пример, используем MediaPlayer чтобы воспроизвести файл музыки и некоторые кнопки контроля как старт, остановить, пауза, перемотка вперед и назад (Рисунок 3.4).

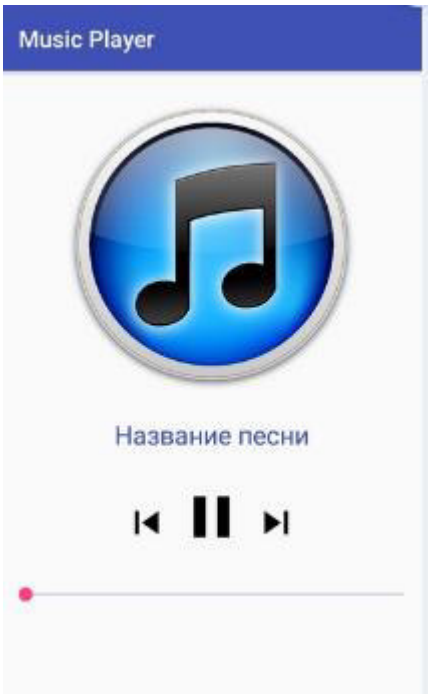

Рисунок 3.4 – Интерфейс пользователя

Первая страница мобильного приложения начинается с чтения файлов формата .mp3. Во-первых, доступ к этим файлам будет разрешен пользователем, после доступа файлы будут перечислены как страница мобильного приложения.

MediaPlayer может воспроизводить аудио и видео, с файлами имеющимися на вашем устройстве или из URL. Похоже на другие аудиопроигрыватели, которые вы знаете, MediaPlayer предоставляет методы упрвления звуков (control playback of audio/video) включая старт, пауза, перемотка вперед назад, вы так же можете вызвать MediaPlayer из сервиса (Рисунок 3.5).

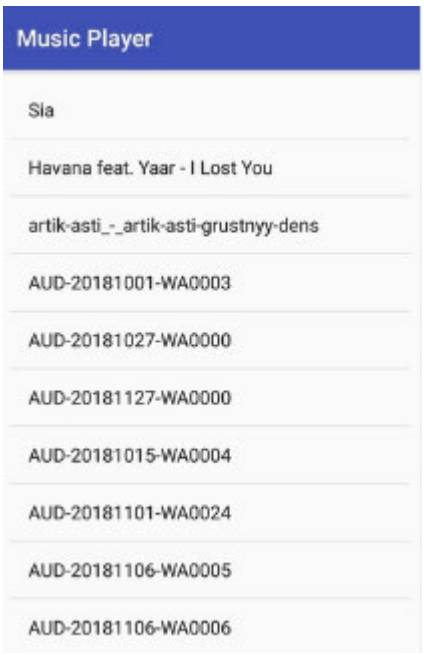

Рисунок 3.5 – Интерфейс пользователя

4 Безопасность жизнедеятельности

4.1 Анализ условий труда

В дипломном проекте я разрабатываю мобильное приложение на платформе Android. Для ее реализаций мне будет достаточно небольшого помещения типа подвала. В помещении есть письменный стол и кресло, а также карманный вай-фай роутер и ноутбук.

Так как помещение находиться в подвале и нету окон, здесь очень тихо. Так же в помещении хорошая вентиляция, тем самым исключаем расчеты на естественное освещение, шум и вентиляцию.

Недостаточность освещения приводит к напряжению зрения, ослабляет внимание, приводит к наступлению преждевременной утомленности. Чрезмерно яркое освещение вызывает ослепление, раздражение и резь в глазах. Неправильное направление света на рабочем месте может создавать резкие тени, блики, дезориентировать работающего. Все эти причины могут привести к несчастному случаю или профзаболеваниям, поэтому столь важен правильный расчет освещенности. Правильно спроектированное и выполненное производственное освещение улучшает условия зрительной работы, снижает утомляемость, способствует повышению производительности труда, благотворно влияет на производственную среду, оказывая положительное психологическое воздействие на работающего, повышает безопасность труда и снижает травматизм. Искусственное освещение применяется при работе в темное время суток и днем, когда не удается обеспечить нормированные значения коэффициента естественного освещения (пасмурная погода, короткий световой день). Освещение, при котором недостаточное по нормам естественное освещение дополняется искусственным, называется совмещенным освещением.

Задание:
– составить существующую схему размещения светильников;
– рассчитать освещенность на рабочем месте точечным методом;
– привести нормативные значения освещенности согласно СНиП РК;
– произвести реконструкцию системы освещения;
– привести планируемую схему расположения светильников;
– сделать выводы и предложения.

Исходные данные
S - площадь помещения, S = 15 м^2;
H - расчетная высота подвеса, h = 2 м;
A - ширина помещения, A = 3 м;
B - длина помещения, B = 5 м.
Тип светильников: ЛБ–40–1

Площадь помещения небольшое S = 15 м^2 примерна рисунке 4.1:

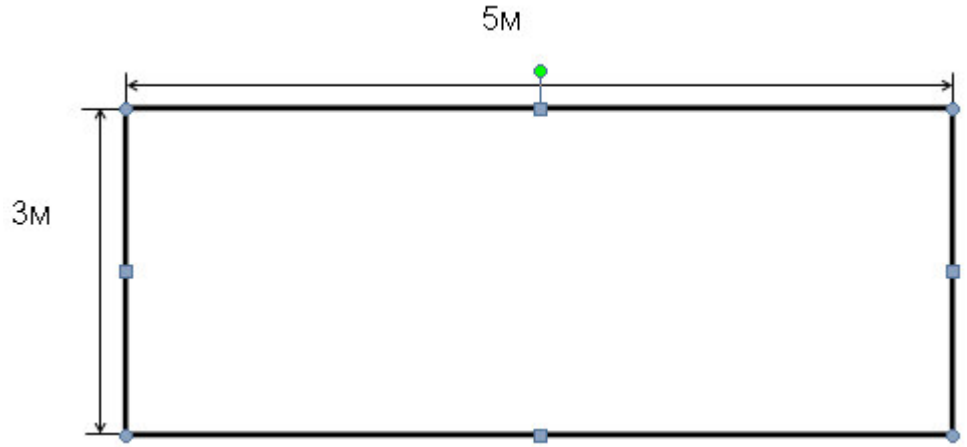

Рисунок 4.1 – Схема помещения

Таблица 4.1 – Технические характеристики лампы ЛБ-40

Наименование	Мощность, Вт	Длина лампы, мм	Световой поток, лм
Лампы люминесцентные ЛБ-40-1	40	600	3000

Рисунок 4.2 – Лампа люминесцентная

Расчет освещенности рабочего места сводится к выбору системы освещения, определению необходимого числа светильников, их типа и размещения. Исходя из этого, рассчитаем параметры искусственного освещения.

Обычно искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп. Будем использовать люминесцентные лампы, которые по сравнению с лампами накаливания имеют ряд существенных преимуществ:

– по спектральному составу света они близки к дневному, естественному свету;

– обладают более высоким КПД (в 1,5-2 раза выше, чем КПД ламп накаливания);

– обладают повышенной светоотдачей (в 3-4 раза выше, чем у ламп накаливания);

– более длительный срок службы.

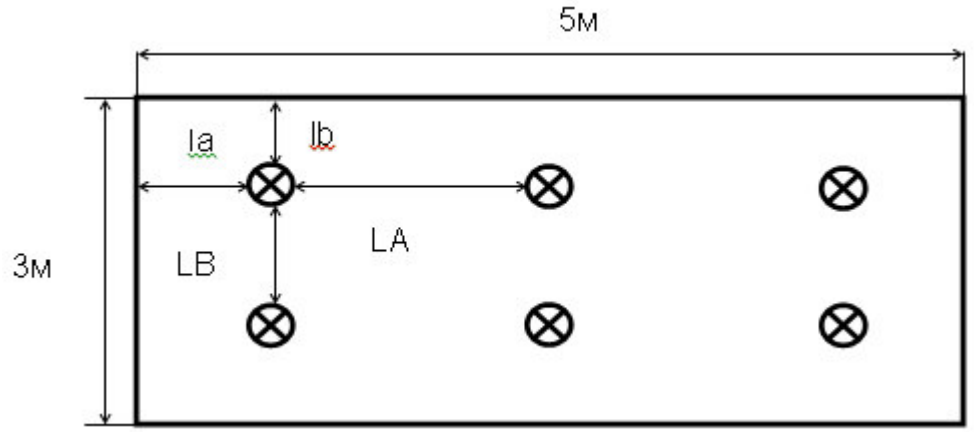

Рисунок 4.3 – Схема расположения светильников

Найдем расстояние между светильниками, учитывая λ=0,5÷1,2.

$$LA=\lambda \cdot hp \quad (4.1)$$

$$LB=\lambda \cdot hp \quad (4.2)$$

$$la=(0,4\div 0,5)\cdot LA \quad (4.3)$$

$$lb=(0,4\div 0,5)\cdot LB \quad (4.4)$$

Учитывая формулу (4.1), получаем, что для нахождения расстояния между светильниками LA, нужно длину помещения разделить на 3:

$$LA=5/3=1.6 \text{ (м)}.$$

Проверим условие коэффициента λ=0,5÷1,2:

$$\lambda=1.6/2=0.8$$

что удовлетворяет.

Учитывая формулу (4.2), получаем, что для нахождения расстояния между светильниками LB, нужно ширину помещения разделить на 2:

$$LB=3/2=1.5 \text{ (м)}.$$

Проверим условие коэффициента λ=0,5÷1,2:

λ=1.5/2=0.7

что удовлетворяет.

Для расчета намечаем контрольную точку А. Необходимо найти d1, d2 – проекции расстояния на потолок между точкой А и соответствующим светильником по формуле (4.5) и (4.6).

$$d_1 = \sqrt{\left(\frac{L_A}{2}\right)^2 + \left(\frac{L_B}{2}\right)^2} \quad (4.5)$$

$$d_1 = \sqrt{\left(\frac{1,6}{2}\right)^2 + \left(\frac{1,5}{2}\right)^2} = \sqrt{1,2} = 1\text{м}$$

$$d_2 = \sqrt{\left(\frac{3 \cdot L_A}{2}\right)^2 + \left(\frac{L_B}{2}\right)^2} \quad (4.6)$$

$$d_2 = \sqrt{\left(\frac{9}{2}\right)^2 + \left(\frac{4}{2}\right)^2} = \sqrt{9,76} = 3,12 \text{ м}$$

Далее определяем угол между высотой потолка и соответствующим отрезком d:

tg α1 = [d1 / hp] (1.8)

tg α1 = [1/3] = 0.33 → α1 = 19°;

cos3 α1 = (0.9455)3 = 0,845;

tg α2 = [d2 / hp] (1.9)

tg α2 = [3,12/3] = 1.04 → α2 = 46°;

cos3 α2 = (0.6947)3 = 0.335;

По этому углу находим силу света от каждого источника

Формула с помощью которой производится перевод люмен в канделы:

Fv= i *2П(1-cos(a))

Составные части формулы:

Fv – световой поток

iv – сила света

a – угол половинной яркости

I1=34 кд (при α = 19°); I2 = 60 кд (при α = 46°).

Освещенность помещения относительно контрольной точки от каждого источника:

$$E_1 = \frac{I_1 \cos^3 \alpha_1}{h_p^2} \quad (4.7)$$

$$E_1 = \frac{34 \cdot 0{,}845}{9} = 3{,}19 \text{ лк};$$

$$E_2 = \frac{I_2 \cos^3 \alpha_2}{h_p^2} \tag{4.8}$$

$$E_2 = \frac{60 \cdot 0{,}335}{9} = 2{,}23 \text{ лк};$$

Суммарная освещенность:

$$E = \frac{\mu \cdot F}{1000 \cdot K_з} \cdot \sum_{i=1}^{8} E_i, \tag{4.9}$$

где μ – коэффициент, учитывающий действие «удаленных» светильников (1,1 ÷ 1,2).

$$E = \frac{1{,}1 \cdot 3000}{1000 \cdot 1{,}5} \cdot (3{,}19 \cdot 4 + 2{,}23 \cdot 4) = 47{,}696 \text{лк}$$

Освещенность на рабочем месте считается недостаточной, поэтому необходимо производить реконструкцию системы освещения производственного помещения.

4.3 Реконструкция системы освещения помещений

Расчет освещения производится для комнаты площадью 15м^2 , ширина которой 3м, длина - 5м. Воспользуемся методом светового потока.

Для определения количества светильников определим световой поток, падающий на поверхность по формуле (4.10):

$$F = \frac{\mathrm{E} \cdot \mathrm{S} \cdot \mathrm{Z} \cdot \mathrm{K}}{\mathrm{n}}, \tag{4.10}$$

где, F – рассчитываемый световой поток, Лм;

E – нормированная минимальная освещенность, Лк (определяется по таблице). Работу программиста, в соответствии с этой таблицей, можно отнести к разряду точных работ, следовательно, минимальная освещенность будет E = 300Лк;

S – площадь освещаемого помещения (в нашем случае S = 15м^2);

Z – отношение средней освещенности к минимальной (обычно принимается равным 1,1…1,2 , пусть Z = 1,1);

K – коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации

(его значение зависит от типа помещения и характера проводимых в нем работ и в нашем случае К = 1,5);

N – коэффициент использования, (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризуемых коэффициентами отражения от стен ($P_С$) и потолка ($P_П$)), значение коэффициентов $P_П$ =70%, $P_С$ =50%, $P_П$ =30%. Значение n определим по таблице коэффициентов использования различных светильников. Для этого (4.1) вычислим индекс помещения по формуле (4.11):

$$i = \frac{L \cdot B}{h_p \cdot (L + B)} \tag{4.11}$$

где, S – площадь помещения, S = 15 м^2;

h – расчетная высота подвеса, h = 2 м;

A – ширина помещения, A = 3 м;

B – длина помещения,B = 5 м.

Подставив значения получим:

$$i = \frac{15}{3 \cdot (3 + 5)} = 0{,}62$$

Зная индекс помещения I , по таблице 4.4 находим n = 0,22

ρ	80	80	70	50	50	30	0
	80	50	50	50	30	30	0
i	30	30	20	10	10	10	0
0,4	0,25	0,10	0,10	0,10	0,06	0,06	0,03
0,6	0,42	0,22	0,21	0,20	0,15	0,15	0,10
0,8	0,54	0,33	0,32	0,30	0,24	0,24	0,18
1	0,62	0,40	0,38	0,35	0,30	0,29	0,23
1,25	0,67	0,47	0,44	0,41	0,35	0,35	0,28
1,5	0,72	0,53	0,49	0,46	0,41	0,40	0,34
2	0,76	0,58	0,54	0,50	0,45	0,44	0,38
2,5	0,82	0,66	0,61	0,56	0,52	0,51	0,45
3	0,85	0,71	0,65	0,59	0,55	0,54	0,48
4	0,87	0,75	0,68	0,62	0,58	0,57	0,52
5	0,90	0,79	0,72	0,65	0,62	0,62	0,56

Рисунок 4.4 – Таблица коэффициентов использования

Подставим все значения в формулу для определения светового потока F:

$$F = \frac{300 \cdot 1{,}5 \cdot 15 \cdot 1{,}1}{0{,}22} = 33750$$

Для освещения выбираем люминесцентные лампы типа ЛБ-40, световой поток которых F = 3000 Лм.

Рассчитаем необходимое количество ламп по формуле (4.12):

$$N = \frac{\mathrm{E} \cdot \mathrm{S} \cdot \mathrm{Z} \cdot \mathrm{K}}{F \cdot \mathrm{n}}, \qquad (4.12)$$

где, N - определяемое число ламп;

F - световой поток, F = 33750 Лм;

$F_л$ - световой поток лампы, $F_л$ = 3000 Лм.

$$\mathrm{N} = \frac{33750}{3000} = 12шт$$

Возьмем другой тип ламп – ЛБ-40-2.

Тогда

$$\mathrm{N} = \frac{33750}{3000 \cdot 2} = 6шт$$

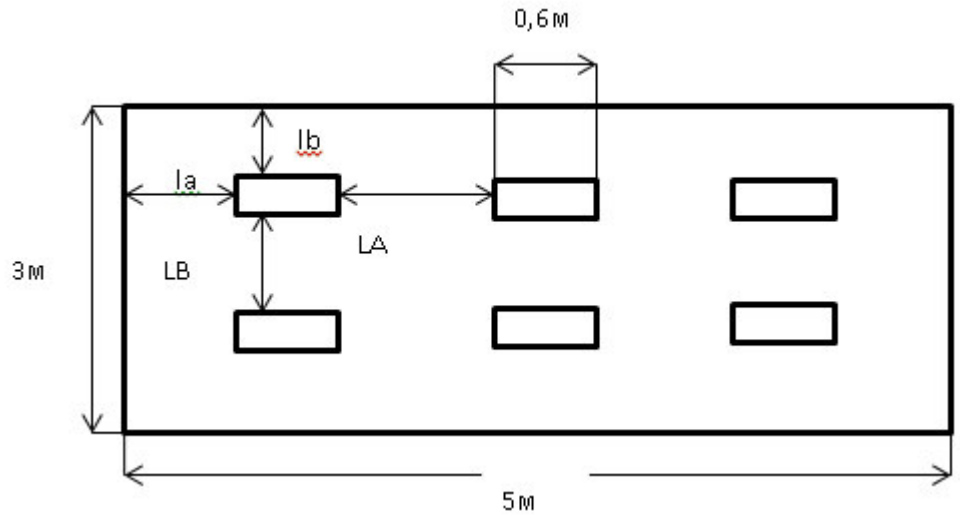

Рисунок 4.5 – Новая схема расположения светильников

Найдем расстояние между светильниками, учитывая λ=0,5÷1,2.

LA=λ· hp=0,5·2=1 (м)

LB=λ· hp=0,5·2=1 (м)

la =(0,4÷0,5)· LA =0,4·1,5=0,6 (м)

lb =(0,4÷0,5)· LB =0,35·1,5=0,5 (м).

Вывод. В данном проекте был проведен расчет искусственного освещения. Расчет освещенности методом коэффициента использования светового потока показал, что количество ламп равное 10 позволяет равномерно освещать помещения с площадью 15 кв м. Для обеспечения необходимой освещенности помещения типа подвал с параметрами 3х5х2 необходимо установить количество светильников типа ЛБ-40-2 до 6 штук.

В данном разделе дипломного проекта были изложены требования к рабочему месту инженера - программиста. Созданные условия должны обеспечивать комфортную работу. На основании изученной литературы по данной проблеме был проведен расчет оптимального освещения производственного помещения. Соблюдение условий, определяющих оптимальную организацию рабочего места инженера - программиста, позволит сохранить хорошую работоспособность в течение всего рабочего дня, повысит как в количественном, так и в качественном отношениях производительность труда программиста, что в свою очередь будет способствовать быстрейшей разработке и отладке программного продукта.

5 Экономическое обоснование разработки проекта

5.1 Трудоемкость разработки программного продукта

Тема дипломного проекта – «Разработка мобильного приложения Music Player на платформе Android».

Целью экономического обоснования дипломного проекта является расчет и анализ затрат, необходимых для создания и реализации информационной системы, изучить затраты на проектирование и функционирование приложения; определить условия и сроки окупаемости затрат. Обосновать рациональность проектирования данного приложения.

В данной главе производится экономический расчет коммерческой стоимости исследования проекта. Расчеты учитывают расходы на создание и внедрение информационной системы.

Для реализации проекта необходимы финансовые, материальные и трудовые ресурсы;

План проектирования и реализации проекта предусматривает 3 этапа в течении 1 месяца (4 недели).

Первый этап:

– постановка задачи;

– разработка содержания работы.

Второй этап:

– создание программного продукта.

Третий этап:

– тестирование;

– откладка;

– внедрение.

Таблица 5.1 – Распределение работ по этапам и видам и оценка их трудоемкости

Этап разработки	Вид работы на данном этапе	Трудоемкость разработки, чел.× ч.
Первый этап	Постановка задачи; Разработка содержания работы	1×45
Второй этап	Создание программного продукта	1×97
Третий этап	Тестирование; Откладка; Внедрение.	1×50
ИТОГО трудоемкость дипломного проекта		192

5.2 Расчет затрат на разработку ПП

Таблица 5.2 – Затраты на программное обеспечение и оборудование

Наименование ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Ноутбук Lenovo Core i5 3320M	шт	1	69000	69000
Лазерный принтер HP Deskjet Ink Advantage 2515	шт	1	42000	42000
Программное обеспечение Android Studio	шт	1	бесплатно	
Итого затраты на программное обеспечение и оборудование				111000

Таблица 5.3 – Материальные затраты

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Бумага	шт	3	1100	3300
Картридж	шт	1	2500	2500
Итого затраты на материальные ресурсы				5800

Общая сумма затрат на материальные ресурсы ($З_м$) определяется по формуле:

$$З_м = \sum_{i=1}^{n} P \times Цi \quad (5.1)$$

где P_i – расход i-го вида материального ресурса, натуральные единицы;

$Ц_i$ – цена за единицу i-го вида материального ресурса, тг;

i – вид материального ресурса;

n – количество видов материальных ресурсов.

$$З_м = \sum_{i=1}^{n} Pi * Цi = ((1100 \times 3) + 2500) = 5800 \text{ тг.}$$

Необходимо рассчитать затраты на электроэнергию по форме, приведенной в таблице 5.4.

Общая сумма затрат на электроэнергию ($З_э$) рассчитывается по формуле:

$$З_э = \sum_{i=1}^{n} \mathbf{M_i} * \mathbf{K_i} * \mathbf{T_i} * Ц \quad (5.2)$$

где M_i – паспортная мощность i-го электрооборудования, кВт;

K_i – коэффициент использования мощности i-го электрооборудования (принимается K_i=0.7¸0.9);

T_i – время работы i-го оборудования за весь период разработки ПП ч;

Ц – цена электроэнергии, тг/кВт×ч;

i – вид электрооборудования;

n – количество электрооборудования.

$$З_{э1} = 0{,}45 \cdot 0{,}7 \cdot 192 \cdot 16{,}53 = 999{,}7 \text{ тг.} \approx 1000 \text{ тг.}$$

$$З_{э2} = 0{,}33 \cdot 0{,}7 \cdot 2 \cdot 16{,}53 = 7{,}6 \text{ тг} \approx 8 \text{ тг.}$$

$$З_{э} = З_{э1+} З_{э2} = 1000 + 8 = 1008 \text{ тг.}$$

Таблица 5.4 – Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки ПП, ч	Цена электроэнергии, тг/кВт*ч	Сумма, тг
Ноутбук Lenovo Core i5 3320M	0,45	0,7	192	16,53	1000
Лазерный принтер HP Deskjet Ink Advantage 2515	0,33	0,7	2	16,53	8
ИТОГО затраты на электроэнергию					1008

5.3 Трудовые ресурсы, задействованные в работе

Для расчёта общей суммы затрат на оплату заработной платы $З_{тр}$ воспользуемся формулой (5.3):

$$З_{тр} = \sum_{i=1}^{n} ЧС_i \cdot Т_i, \quad (5.3)$$

где n – количество разработчиков приложения;

$ЧС_i$ – часовая ставка i-го работника, тг;

T_i – трудоемкость разработки ПП, чел.×ч;

i – категория работника.

Трудоемкость разработки ПП определяется по данным таблицы 5.1.

Часовая ставка инженера-разработчика составляет 900 (тг/ч), трудоемкость разработки – 192 ч.

Результаты расчёта основной заработной платы представлены в таблице 5.5.

Таблица 5.5 – Результаты расчёта затрат основной заработной платы

Наименование содержания работ	Исполнитель	Трудоемкость норма-час	Заработная плата за час, тг/час	Сумма ЗП, тенге
Тех. задание	Разработчик	9	900	8100
Моделирование		36		32400
Программирование		97		87300
Тестирование		34		30600
Внедрение		16		14400
Итого		181		172800

Также необходимо рассчитать отчисления на социальный налог, который составляет 9,5% (согласно статье 485 НК РК) от дохода работника. Социальные отчисления определим по следующей формуле:

$$З_{сзi} = (З_{тр} - З_{ПО}) \cdot \mathbf{0,095} \quad (5.4)$$

где $З_{ПО}$ – пенсионный отчисления, 10% от общего фонда оплаты труда, тенге. Рассчитаем пенсионные отчисления по следующей формуле:

$$З_{ПО} = З_{тр} \cdot \mathbf{0,1} \quad (5.5)$$

Используя формулы 5.5 и 5.9 получаем:

$$З_{ПО} = \mathbf{172800 \cdot 0,1 = 17280} \text{ тенге}$$

$$З_{сзi} = (\mathbf{172800 - 17280}) \cdot \mathbf{0,095 = 14774} \text{ тенге}$$

Сумма годовых амортизационных отчислений определяется по формуле:

$$А=П*Н/100 \quad (5.6)$$

Годовые нормы амортизации ОФ принимаются по налоговому кодексу РК или определяются, исходя из возможного срока полезного использования ОФ:

$$Н_{Ai}=100/Т_{Ni}, \quad (5.7)$$

где T_{Ni} - возможный срок использования i-го ОФ, год.

Таблица 5.6 - Амортизация основных фондов (ОФ)

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Срок полезного использования обор-я и ПО, лет	Сумма амортизации в год, тг	Сумма амортизации в месяц, тг
Ноутбук Lenovo Core i5 3320M	69 000	20	5	13 800	1150
Лазерный принтер HP Deskjet Ink Advantage 2515	42 000	20	5	8 400	700
Программное обеспечение Android Studio	Распространяется бесплатно				
ИТОГО амортизация основных фондов					1850

$$НА_{об}= 100 / 5 = 20;$$

$$А_{н}= (69\ 000 * 20) / 100 = 13\ 800 \text{ тг};$$

$$А_{м}= (42\ 000 * 20) / 100= 8\ 400 \text{ тг};$$

Сумма амортизации за один месяц = А / 12.

Сумма амортизационных отчислений за месяц равна 1850 тг.

В статью «Прочие затраты» включаются расходы на арендную плату, включая коммунальные платежи, канцелярские и прочие хозяйственные расходы.

Стоимость аренды помещения на месяц равна 118 000 тг. (в эту сумму включены коммунальные услуги).

Арендная плата рассчитывается по формуле:

$$АП= Са*S \qquad (5.8)$$

где Са – срок аренды;

S – стоимость аренды за 1 месяц.

$$АП = 118\ 000 * 1 = 118\ 000 \text{ тг}.$$

Расходы на интернет, месячная оплата которого составляет 6300 тг равны:

$$Р_{и}= 6300 * 1= 6\ 300 \text{ тг}.$$

Прочие хозяйственные расходы составляют 5000 тг;

Прочие затраты = 118000 + 6300 + 5000 = 129300 тг.

На основании полученных данных по отдельным статьям в таблице 5.7 приведена смета затрат на разработку ПП

Таблица 5.7 - Смета затрат на разработку ПП

Статьи затрат	Сумма, тг
– Материальные затраты, в том числе:	
– материалы	5800
– электроэнергия	1008
– Затраты на оплату труда.	172800
– Отчисления на социальные нужды.	14774
– Амортизация основных фондов.	1850
– Прочие затраты.	129300
ИТОГО по смете	325532

5.4 Определение возможной (договорной) цены ПП

Величина возможной (договорной) цены ПП должна устанавливаться с учетом эффективности, качества и сроков ее выполнения на уровне, отвечающем экономическим интересам заказчика (потребителя) и исполнителя.

Договорная цена (ЦД) для прикладных ПП рассчитывается по формуле:

$$Ц_д = З_{нир} \times (1 + \frac{Р}{100}) \quad (5.9)$$

где $З_{НИР}$ - затраты на разработку ПП (из таблицы 5.7), тг;

Р - средний уровень рентабельности ПП. %

$$Ц_д = З_{нир} \times (1 + \frac{Р}{100}) = 325532 \times (1 + \frac{20}{100}) = 390638 \text{ тг.}$$

Далее определяется цена реализации с учетом налога на добавленную стоимость (НДС), ставка НДС устанавливается законодательно Налоговым Кодексом РК. На 2019 год ставка НДС установлена в размере 12%.

Цена реализации с учетом НДС рассчитывается по формуле:

$$Цр = Цд + Цд \times НДС \quad (5.10)$$

Цр=Цд + Цд × НДС = 390638 + 390638 *0,12 = 437514,56 тенге.

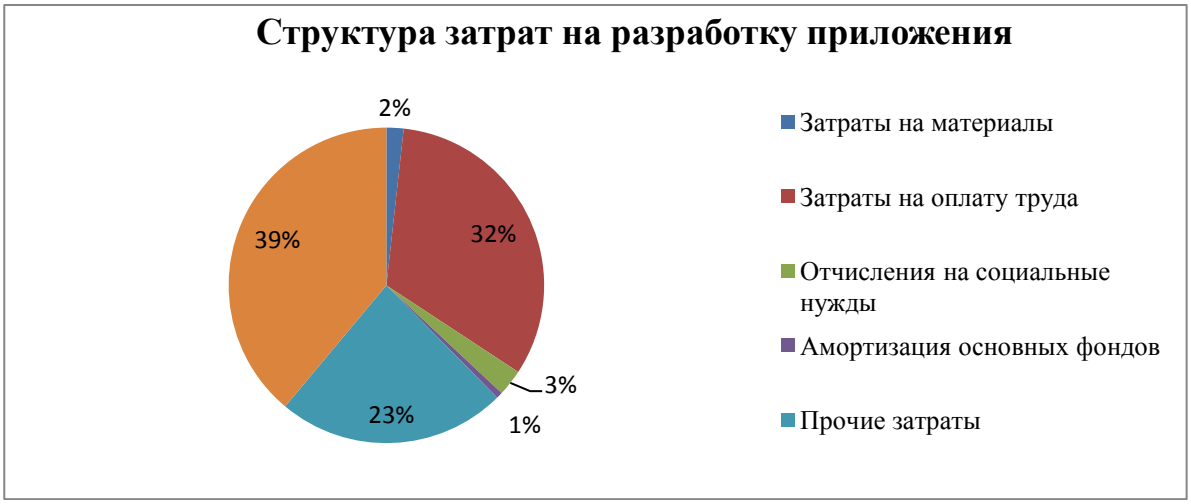

Рисунок 5.1 – Структура затрат на разработку приложения

5.5 Оценка социально – экономических результатов

К социально – экономическим показателям функционирования мобильной информационной системы является обработка таких факторов как:

– качество процесса управления;

– длительность и сроки проектирования программного продукта;

– расходы на реализацию приложения и эксплуатацию программного продукта;

– количеству разработчиков.

Таблица 5.8 – Срок окупаемости

Период	Первоначальные затраты	Денежный поток нарастающим итогом
1	325532	81 000
2	325532	81 000
3	325532	81 000
4	325532	81 000
5	325532	81 000
6	325532	81 000
7	325532	81 000
Срок окупаемости		4 месяцев

Рентабельность составит 20%.

Вывод. Рассчитанную возможную цену ПП необходимо сравнить с ценами на аналогичные разработки.

Суммарные расходы на разработку аналитической информационной системы считаются как сумма фонда заработной плат, затрат на социальное

страхование, накладных расходов и расходов на материалы и комплектующие.

Срок разработки данного проекта составил 30 дня или 192 часа.

Итоговая стоимость разработки мобильного приложения составила 437514,56, в том числе НДС – 46876,56 (12%).

Предполагаемая цена продажи составила 437514,56 тг, прибыль от продажи рекламы составит 81000 тг. в месяц.

Срок окупаемости проекта составит 4 месяцев.

Заключение

Неистовый темп технологических изменений и роста на рынке мобильной связи усложнил для разработчиков стратегическое планирование индивидуального проекта, причем не только с технической точки зрения, но и потому, что доля рынка смартфонов быстро меняется между различными системами.

Музыкальный плеер на базе приложения Android в настоящее время популярен на рынке. Завершающая разработка операционной системы Android предоставляет разработчикам прекрасную платформу, которая может освоить популярные компьютерные технологии в сочетании с приобретенным и знаниями и освоить разработку мобильных приложении. Музыкальный проигрыватель в этом дипломном проекте представляет собой прикладное программное обеспечение на основе Google Android. Приложение Android на мобильных терминалах также полностью нарушило традиционное понимание мобильных терминалов.

Целью данного проекта являлась разработка Android – приложения для проигрывания медиа файлов на смартфонах.

В ходе работы были выполнены следующие задачи:

– изучены особенности разработки приложений для ос Android;

– определены требования к программе;

– спроектирована архитектура приложения;

– разработано мобильное приложение;

– выполнено тестирование.

Список литературы

1 Android Development Tools for Eclipse. [Электронный ресурс] URL: https://marketplace.eclipse.org/content/android-development-tools-eclipse (дата обращения: 12.03.2017).

2 Automating User Interface Tests. [Электронный ресурс] URL: https://developer.android.com/training/testing/ui-testing/index.html (дата обращения: 16.05.2017). 3. Hibernate ORM 5.2.9.Final User Guide. [Электронный ресурс] https://docs.jboss.org/hibernate/orm/current/userguide/html_single/Hibernate_User_Guide.html (дата обращения: 12.04.2017).

3 Java EE 7: Building Web Applications with WebSocket, JavaScript and HTML5. [Электронный ресурс] URL: http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/HomeWebsocket/WebsocketHome.html (дата обращения: 24.04.2017).

4 Mockito. [Электронный ресурс] URL: http://site.mockito.org/ (дата обращения: 16.05.2017). 6. MVP and MVC Architectures in Android. [Электронный ресурс] URL: http://www.techyourchance.com/mvp-mvc-android-1/ (дата обращения: 12.03.2017).

5 Olan M. Unit testing: test early, test often. // Journal of Computing Sciences in Colleges, 2003. – Vol. 19. – No 2. – P. 319–328.

6 Robolectric. [Электронный ресурс] URL: http://robolectric.org/ (дата обращения: 16.05.2017).

7 Spring Security Reference. [Электронный ресурс] URL: https://docs.spring.io/spring-security/site/docs/current/reference/htmlsingle (дата обращения: 23.04.2017).

8 SQLite – a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. [Электронный ресурс] URL: https://sqlite.org (дата обращения: 25.04.2017).

9 SQLite vs MySQL vs PostgreSQL: сравнение систем управления базами данных. [Электронный ресурс] URL http://devacademy.ru/posts/sqlite-vs-mysql-vs-postgresql/ (дата обращения: 12.04.2017).

10 Testing UI for a Single App. [Электронный ресурс] URL: https://developer.android.com/training/testing/ui-testing/espresso-testing.html (дата обращения: 16.05.2017).

11 Web MVC framework. [Электронный ресурс] URL: https://docs.spring.io/spring/docs/current/spring-frameworkreference/html/mvc.html (дата обращения: 23.04.2017).

12 Webber J., Parastatidis S., Robonson I. REST in Practice. – USA: O'Reilly Media, 2010. – 448 p.

13 What is API Level? [Электронный ресурс] URL:http://developer.android.com/guide/topics/manifest/uses-sdkelement.html-#ApiLevels (дата обращения:25.04.2017).

14 What is Object/Relational Mapping? [Электронный ресурс] URL:

http://hibernate.org/orm/what-is-an-orm/ (дата обращения: 12.04.2017).

15 Актуальность изучения иностранных языков в социальноэкономических условиях современной России. [Электронный ресурс] URL: http://www.ibl.ru/konf/140509/61.html (дата обращения: 12.03.2017)

16 Варенина Л. П. Геймификация в образовании. // Историческая и образовательная мысль. 2014. – № 6-2. – С. 314-317.

17 Введение в веб-модель MVC Spring. [Электронный ресурс] URL: https://netbeans.org/kb/docs/web/quickstart-webapps-spring_ru.html (дата обращения: 23.04.2017).

18 Зенина Л.В. Возможности современных компьютерных программ в процессе обучения иностранным языкам в вузе. / Л.В Зенина., Н.А. Каменева. // Вестник Московского государственного гуманитарного университета им. М.А. Шолохова. – Педагогика и пс50

19 Краснова Т.И. Геймификация обучения иностранному языку. // Молодой ученый, 2015. – № 11. – С. 1373-1375.

20 Обзор мобильных приложений для изучения английского языка. [Электронный ресурс] URL:https://habrahabr.ru/post/239985/ (дата обращения: 12.03.2017).

21 Общие сведения о платформе Android. [Электронный ресурс] URL: https://developer.android.com/guide/index.html (дата обращения: 12.03.2017).

22 Основные понятия баз данных. [Электронный ресурс] URL: http://inf.susu.ac.ru/Klinachev/lc_sga_26.htm (дата обращения: 12.04.2017)

23 Сайт Android Studio. [Электронный ресурс] URL: https://developer.android.com/studio/features.html (дата обращения: 12.03.2017).

24 Сайт Anki. [Электронный ресурс] URL: http://ankisrs.net/ (дата обращения: 12.03.2017). 27. Сайт Eclipse. [Электронный ресурс] URL: https://eclipse.org/ (дата обращения: 12.03.2017). 28. Сайт LinguaLeo. [Электронный ресурс] URL: http://lingualeo.com/ru (дата обращения: 12.03.2017).

25 Сайт WebSocket. [Электронный ресурс] URL: https://www.websocket.org/aboutwebsocket.html (дата обращения: 12.03.2017).

26 Сайт Xamarin. [Электронный ресурс] URL: https://www.xamarin.com/ (дата обращения: 12.03.2017).

27 Сервлеты. Введение. [Электронный ресурс] URL: http://www.java2ee.ru/servlets/ (дата обращения: 23.04.2017).

28 Тестирование программного обеспечения - основные понятия и определения. [Электронный ресурс] URL: http://www.protesting.ru/testing/ (дата обращения: 16.05.2017).