

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 обучающего мобильного приложения по хореографии
“Dance life”

Специальность 5B060200 – «Информатика»

Выполнила Кудайбергенова А.Р. Группа ИНФ-15-2

Научный руководитель PhD, доцент Бидахмет Ж. *Ж. Бидахмет*

Консультанты:

по экономической части: к.э.н., доцент *А-У* _____ Бекишева А.И.
« 06 » 05 _____ 2019 г.

по безопасности жизнедеятельности: ст. преп. *Т-Е* _____ Бекбасаров Ш.Ш.
« 01 » 05 _____ 2019 г.

по применению
программного обеспечения: ст. преп. *У-А* _____ Майкотов М.Н.
« 06 » 05 _____ 2019 г.

Нормоконтролер: ст. преп. *А-У* _____ Алимсеитова Ж.К.
« 15 » 05 _____ 2019 г.

Рецензент: зам.дир. по уч. раб. ИИиIT КазНИТУ им. К.И. Сатпаева

_____ Абдолдина Ф.Н.
« ____ » _____ 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 Кудайбергеновой Акгуль Рахатовне

Тема проекта: Разработка обучающего мобильного приложения по хореографии “Dance life”

Утверждена приказом по университету № 53 от «01» 03 2019 г.

Срок сдачи законченного проекта «23» мая 2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Руководство системы менеджмента качества на предприятии; международные стандарты ИСО-9001, данные преддипломной практики.

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

- а) исследование современных операционных систем;
- б) разработка обучающего мобильного приложения «Dance life» по хореографии
- в) технико-экономическое обоснование проекта;
- г) вопросы безопасности жизнедеятельности и охраны труда;

Перечень графического материала (с точным указанием обязательных чертежей): представлены 9 таблиц, 28 иллюстрации.

Основная рекомендуемая литература:

1. Гамма Э., Приемы объектно-ориентированного проектирования // Э.Гамма, Р.Хелм, Р.Джонсон. — СПб: Питер, 2001. — 368 с.
2. UML диаграммы. [Электронный ресурс] // Google Inc. — Электрон. дан. — [Б.м.], 2016 — URL: <https://docplayer.ru/29295497-Uml-diagrammy-diagramma-klassov-2-diagramma-precedentov-22.html>.

3. Бекишева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В0703 - Информационные системы – Алматы: АУЭС; 2013. –24 с.

4. Е. Мананбаева, А.С. Бегимбетова Охрана труда. Методические указания к выполнению расчетно-графических работ для студентов - бакалавров всех специальностей. - Алматы: АУЭС, 2013 - 17 с.

Консультации по работе с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Бекишева А.И.	16.03 - 06.05.19	<i>А.И.</i>
Безопасности жизнедеятельности	Бекбасаров Ш.Ш	02.04 - 01.05.19	<i>Ш.Ш.</i>
Программная часть	Майкотов М.Н.	01.05 - 06.05.19	<i>М.Н.</i>
Нормоконтролер	Алимсеитова Ж.К	03.03.19 - 15.05.19	<i>Ж.К.</i>

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Получение информации о предметной области	01.11.18-20.12.18	<i>Выполнено</i>
Проектирование	21.12.18-20.02.19	<i>Выполнено</i>
Разработка приложения	21.02.19-30.04.19	<i>Выполнено</i>
Тестирование	31.04.19-15.05.19	<i>Выполнено</i>
Внедрение и поддержка	15.05.19-20.05.19	<i>Выполнено</i>

Дата выдачи задания « ____ » _____ 2019 г.

Заведующий кафедрой _____ Т.С. Картбаев

Научный руководитель проекта *Ж. Бидахмет* - Бидахмет Ж.

Задание принял к исполнению студент *А. Кудайбергенова* Кудайбергенова А.Р.

Аннотация

Бесконечное множество новейших технологий, которые были вызваны возрастающим ростом информатизации общества, делает нашу жизнь сложной без быстрого доступа к получению той или иной информации. Сейчас очень сложно представить жизнь современного человека без мобильных устройств. Поэтому на рынке появился новый термин – мобильное приложение.

Целью дипломного проекта является изучение теоретических основ архитектуры мобильных приложений и исследование рынка современных операционных систем, средств разработки, используемых при реализации мобильного приложения, разработка мобильного приложения, позволяющего пользователям обучаться различным видам хореографии с помощью просмотра видео-уроков.

Новизна разработанного приложения состоит в комфортном и практичном для пользователей применении, а также в удобном интерфейсе.

Annotation

The infinite number of new technologies that have been caused by the increasing growth of information society, makes our lives difficult without quick access to information. Now it is very difficult to imagine the life of modern man without mobile devices. Therefore, a new term appeared on the market – mobile application.

The aim of the diploma project is to study the theoretical foundations of the architecture of mobile applications and market research of modern operating systems, development tools used in the implementation of mobile applications, the development of a mobile application that allows users to learn different types of choreography by watching video lessons.

The novelty of the developed application is a comfortable and practical application for users, as well as a convenient interface.

Аңдатпа

Қоғамды ақпараттандырудың өсуімен байланысты көптеген жаңа технологиялар біздің өмірімізді қандай да бір ақпаратты алуға жылдам қол жеткізбей қиын етеді. Қазіргі заманғы адамның өмірін ұялы құрылғыларсыз елестету өте қиын. Сондықтан нарықта жаңа термин – мобильді қосымша пайда болды.

Дипломдық жобаның мақсаты мобильді қосымшалар архитектурасының теориялық негіздерін зерттеу және заманауи операциялық жүйелер нарығын зерттеу, мобильді қосымшаны іске асыру кезінде қолданылатын өңдеу құралдары, бейне-сабақтарды қарау арқылы хореографияның әр түрлі түрлерін үйренуге мүмкіндік беретін мобильді қосымшаны әзірлеу болып табылады.

Әзірленген қосымшаның жаңалығы пайдаланушылар үшін ыңғайлы және практикалық қолданудан, сондай-ақ ыңғайлы интерфейстен тұрады.

Содержание

Введение.....	8
1 Основная часть.....	9
1.1 История операционных систем.....	9
1.2 Статистика используемых операционных систем.....	9
1.3 Операционная система Android.....	11
1.4 Архитектура Android.....	15
1.5 Классификация мобильных приложений.....	17
1.6 Анализ известных решений и программных средств.....	20
2 Программная часть.....	25
2.1 Выбор средств для реализации приложения Android.....	25
2.2 Среда программирования Android Studio.....	26
2.3 Язык программирования Java.....	28
2.4 Эмулятор для разработки приложений Android SDK.....	29
2.5 Пакеты, которые являются частью Android SDK.....	30
2.6 Компоненты пользовательского интерфейса ANDROID.....	34
2.7 Разработка программного обеспечения.....	38
2.8 Создание интерфейса.....	44
2.9 Описание функциональности.....	44
3 Технико-экономическое обоснование проекта.....	50
3.1 Трудоемкость разработки ПП.....	50
3.2 Расчет затрат на разработку ПП.....	51
3.3 Определение возможной (договорной) цены ПП.....	57
3.4 Вывод.....	58
4 Охрана труда и БЖД.....	59
4.1 Описание условий труда в помещении.....	59
4.2 Расчет естественного и искусственного освещения в помещении.....	60
4.2.1 Расчет естественного освещения.....	61
4.2.2 Расчет искусственного освещения.....	63
4.3 Вывод.....	66
Заключение.....	67
Список использованной литературы.....	68
Приложение А.....	69
Приложение Б.....	71
Приложение В.....	95

Введение

Бесконечное множество новейших технологий, которые были вызваны возрастающим ростом информатизации общества, делает нашу жизнь сложной без быстрого доступа к получению той или иной информации.

Ведь помимо своей основной задачи мобильные устройства выполняют важную роль в ежедневной работе, с их помощью можно:

- прочитать файлы;
- проверить электронную почту;
- распечатать различные документы с помощью сетевого принтера.

Поэтому на рынке появился новый термин – мобильное приложение. Отличиями этого сегмента является то, что разработка приложений проводится с учетом особенностей мобильных устройств, таких как:

- различия интерфейса;
- параметры экрана;
- сенсорное управление.

В настоящее время операционная система Android считается одной из наиболее распространенной в сфере мобильных устройств. Android в своей популярности, стоит подчеркнуть, в основном благодарен политике Google, который разрешает производителям мобильных устройств без проблем использовать платформу Android для своего продукта, и открытому исходному коду. Большинство крупнейших компаний-производителей мобильных устройств, таких как Sony, HTC, Huawei, Samsung, применяют платформу Android.

Наряду с этим стоит отметить, что многие компании уже оценили все достоинства такого предложения как разработка приложений и сайтов, а также позаботились о разработке и внедрении подходящего продукта.

В рамках данного дипломного проекта было разработано мобильное приложение, с помощью которого можно обучиться танцам.

В первой главе рассмотрена история первых операционных систем, классификация ОС и мобильных приложений.

Во второй главе описаны средства разработки мобильного приложения, обоснован выбор среды разработки Android Studio, а также разработано само мобильное приложение.

В третьей главе описывается экономическая эффективность программного обеспечения.

В четвертой главе описаны условия труда в помещении, где разрабатывается мобильное приложение.

Актуальность данного проекта можно объяснить интенсивным развитием мобильных приложений, в удобстве и простоте использования пользователем приложения. Было принято решение разработать данное мобильное приложение, позволяющее пользователям, не выходя из дома обучаться танцам, тем более в данном приложении представлены различные категории танцев.

1 Теоретическая часть

1.1 История операционных систем

В 40-х гг. XX века в первых электронных счетных системах не было операционных систем. Электронные устройства этого времени были обусловлены на серию механизированных выключателей или с использованием переключателей на платах. Это специализированные устройства, которые, например, производят баллистические расчеты для войск или печатают контент-тексты после введения налога на карточки. Как изобрели универсальные программируемые компьютеры, были введены машинные языки (состоящие из двоичных цифр 0 и 1 на перфорированной бумажной ленте), которые ускорили процессы программирования.

В начале 50-х гг. XX века компьютер смог выполнить только одну программу за другой. Каждый пользователь использовал компьютер только в продолжение определенного промежутка времени и получил спланированное время с программой и данными об ударах или ударах. Программа загружается в электронно-вычислительную машину (ЭВМ), и машина работает до завершения программы. Чаще всего, программы на передней панели можно настроить посредством переключателей и подсветки панели.

Позже ЭВМ появились с библиотеками программ, которые могли бы быть связаны с пользовательской программой ради облегчения процессов, таких как ввод и вывод, вдобавок создать компьютерный код с легко читаемым программным кодом. Это происхождение современных операционных систем. Правда, они всегда выполняли одну задачу за раз.

Усовершенствованием стал суперкомпьютер Atlas, спроектированный в Манчестерском университете, введенный в эксплуатацию в 1962 году, «явилась для многих настолько узнаваемой современной операционной системой».

Таким образом, в 50-60-х гг. XX века был сформирован и воплощен главный тезис, определяющий функциональность ОС:

- пакетный режим;
- разделение времени и многозадачность;
- разделение полномочий, масштабирование в режиме реального времени;
- файловые структуры и файловые системы.

1.2 Статистика используемых операционных систем

Операционная система (ОС) – интегральное ПО, позволяющее регулировать аппаратные и программные средства компьютера, также предоставляющее основные услуги компьютерным ПО. Всем компьютерным программам, кроме процесса замены в устройствах ПО, требуется операционная система.

Операционная система с разделением времени задумывает задачи в целях практического применения системы, помимо этого включает в себя

программное обеспечение с учетом распределения времени ЦП, хранения, печати и остальных ресурсов [1].

Преобладающей настольной ОС считается Microsoft Windows с долей рынка около 83,3%. Второе место занимает MacOS от Apple Inc. (11,2%), а на Linux приходится третье место (-1,55%).

Среди смартфонов и планшетов, по данным третьего квартала 2016 года, Google Android преобладает с процентной долей в 87,5%, темп роста составляет 10,3% в год, затем последует iOS Apple – 12,1% в год, доля рынка уменьшается на 5,2%, тем временем остальные операционные системы составляют только 0,3%.

Разновидности Linux доминируют в секторах серверов и суперкомпьютеров. Иные специальные классы ОС, такие как встроенные системы и системы реального времени, создаются для многих приложений.

Следует изучить данные статистики использования ОС.

Рейтинг ОС в Казахстане. Казахстан весьма мобильный. Даже если Windows находится на первом месте – 63,85%, все же Android имеет 23,08%, а это составляет $\frac{1}{4}$ часть от рейтинга. Доля iOS здесь также высокая – 7,83% (рисунок 1.1).

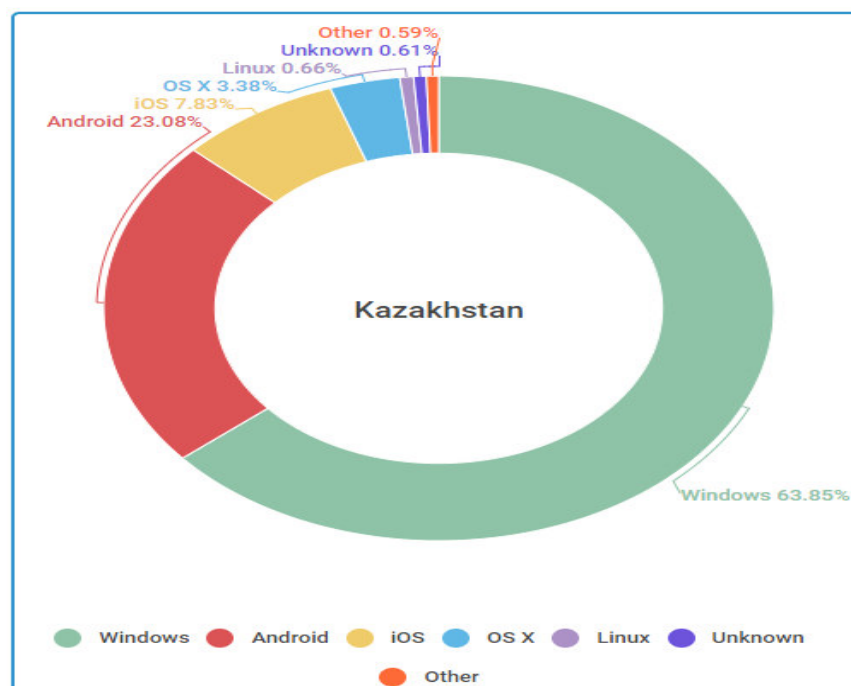


Рисунок 1.1 – ТОП операционных систем в Казахстане

Показатели ОС в мире. Рассматривая относительную статистику настольных и мобильных платформ, на июль 2017 OS Android занимает лидирующее место – 41.24% всех устройств. Второй и третий, следовательно, Windows – 35,24% и iOS - 13,2% (рисунок 1.2).

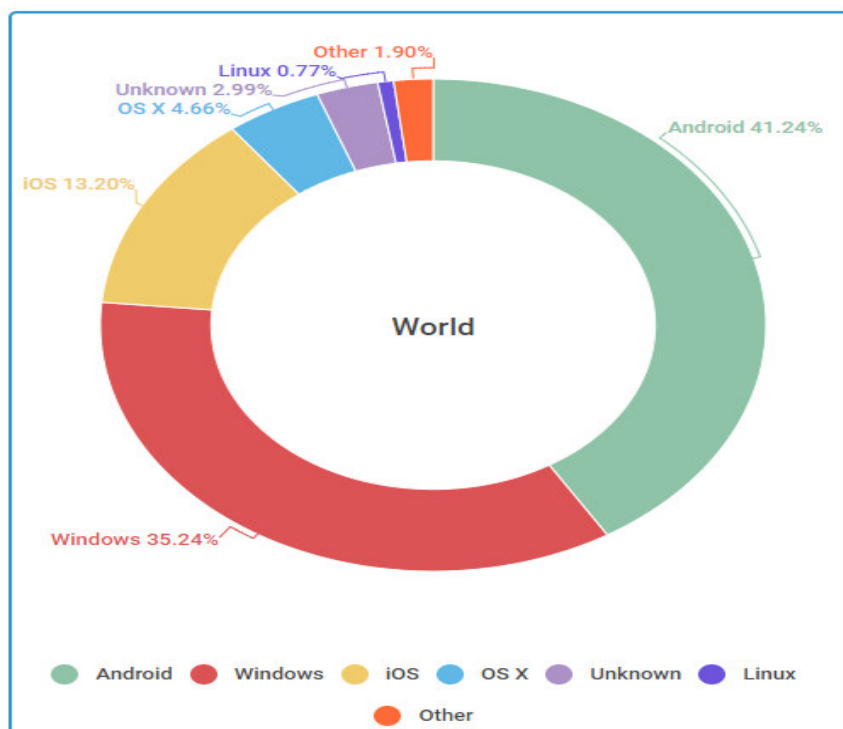


Рисунок 1.2 – ТОП операционных систем в мире

1.3 Операционная система Android

ОС Android – одна из самых популярных и многообещающих операционных систем для разных мобильных устройств [2]. Она предоставляет весьма практичные инструменты и предельную гибкость параметров, что дает любому пользователю смартфона или планшета Android целиком настраивать его в соответствии со своими потребностями. Реализация данной операционной системы стартовала в 2003 году, хотя она стала действительно популярной всего спустя 2 года после приобретения Google. Поворотный момент в истории Android наступил осенью 2008 года. Затем Google представил T-Mobile G1, который стал первым смартфоном с Android. Именно тогда большинство мировых производителей обратили внимание на перспективную операционную систему. Первая версия ОС Android представлена на рисунке 1.3.

Разработчики всегда позиционировали эту операционную систему как систему с открытым исходным кодом. Он дает возможность каждому создавать собственные приложения, игры и другие дополнения в целях улучшения возможностей гаджетов Android (рисунок 1.4). Разработчики сначала подумали обо всем, дабы заставить операционную систему делать все предельно незамедлительно, хоть с «дешевым» оборудованием. Это неоспоримое преимущество, потому что ныне даже люди с самыми скудными денежными возможностями смогут пользоваться главными преимуществами новоизобретенных смартфонов.



Рисунок 1.3 – Первая версия ОС Android

Открытие системы также легко используют производители мобильной электроники, которые производят свои личные многопользовательские интерфейсы, такие как HTC Sense. Поклонникам классического и «чистого» Android стоит обратить внимание на устройство Nexus. Они традиционно первыми получают обновления.

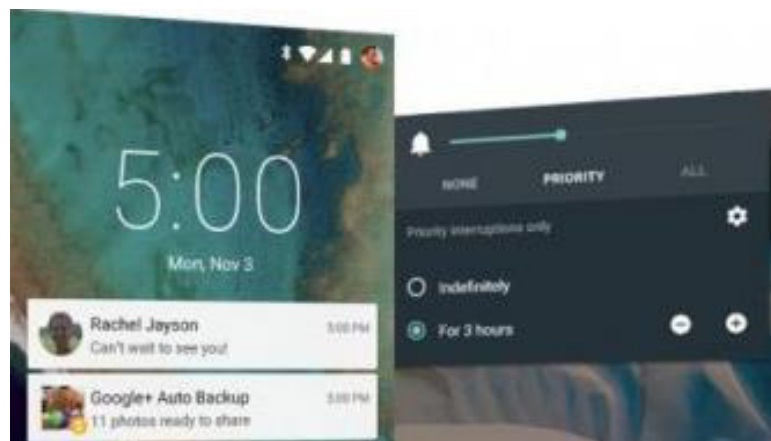


Рисунок 1.4 – Операционная система Android

Время обновления для других устройств обычно задерживается из-за необходимости доработки запатентованных оболочек, основанных на особенностях более новых версий операционной системы.

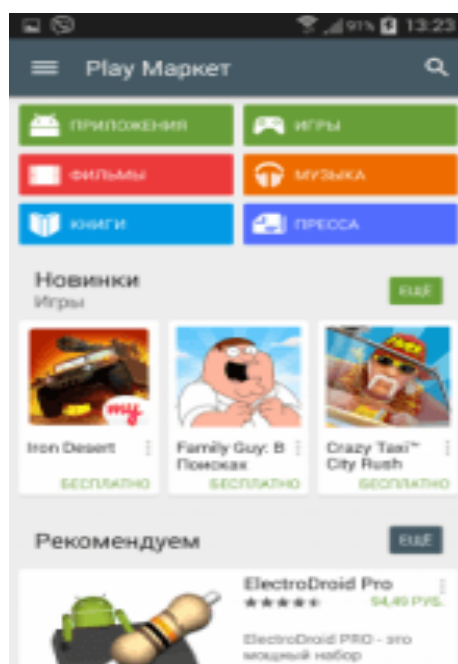


Рисунок 1.5 – Окно Play Store

Основной кладезь игр и приложений – Play Store (рисунок 1.5). Каталоги этого магазина предлагают широкий ассортимент бесплатного и платного контента, предложение которого активно расширяется. Качество игр и приложений «Android» продолжает расти. Google регулярно публикует разные приложения.

В общем, в устройствах Android имеются все функции, которые необходимы современным смартфонам и планшетами в соответствии с их потребностями. Кроме того, функциональность может быть легко расширена с помощью приложений, виджетов или стороннего программного обеспечения.

Если вы хотите купить смартфон на Android, подумайте о том, что большинство функций и программ доступны через Интернет. Если у вас нет доступа к Wi-Fi, рекомендуется установить недорогое соединение для пользователей Интернета либо отключить некоторые функции, требующие доступа к сети.

Опять-таки стоит подчеркнуть ограниченную автономность устройств на Android, особенно представителей предшествующих поколений. Если вы постоянно пользуетесь своим смартфоном или планшетом, вам придется заряжать его каждый день или два раза в день в большинстве случаев. Разработчики усердно работают над устранением данного дефицита и в последнее время на рынки предоставляют интереснейшие устройства с гораздо большей автономностью. Разработчики не остались в стороне. С каждой новейшей версией операционной системы смартфоны потребляют минимум заряда.

Эти сведения позволяют нам выявить основные достоинства и недостатки операционной системы. Неоспоримые достоинства Android:

- открытый программный код, который позволит вам разрабатывать практически все приложения и игры;
- отмена «аппаратных» требований к устройству;
- широкий выбор приложений и занимательных игр на любой вкус;
- абсолютная равнозначность нынешним концепциям функциональности смартфона;
- свобода маневра пользователя. По запросу система и ее интерфейс могут быть легко адаптированы к потребностям клиента;
- знаменитость операционной системы, для большинства современных производителей, что дает возможность пользователю практически с любым бюджетом приобрести многофункциональный и современный смартфон.

Имеются недостатки. Наиболее важными из них:

- Открытый программный код. Да, это можно объяснить, как преимуществами, так и несовершенством. Недочетом является то, что производители мобильных устройств предпочитают свой собственный Shell. Поэтому существуют значительные задержки между официальным выпуском обновления операционной системы и его получением на разных устройствах.
- Относительно низкая автономность. В последнее время был сделан большой шаг в этом направлении, так что очень скоро анекдоты о том, что «Android-смартфоны – самые спортивные, потому что зарядка происходит 2 раза в день», уже не смешная история.

Платформа все время совершенствуется и развивается. Уже сегодня является одним из трех ведущих поставщиков на рынке операционных систем. Большинство экспертов обнаружили, что будущее за Android и неплохое.

Операционная система Android знакома практически всем пользователям мобильных устройств, к ней относились даже восторженные поклонники операционной системы iOS. Операционная система Android разработана для устройств с сенсорным экраном, особенно для планшетов и смартфонов, на основе бесплатного ядра Google от Linux и собственной реализации Google на виртуальной машине Java. Первоначально он был спроектирован компанией Android, Inc., которую далее приобрела Google. Google затем запустил создание Open Handset Alliance (ОНА), который нынче занимается поддержкой и развитием платформы. С Android у вас получится создавать приложения Java, которые управляют устройством через библиотеки, разработанные Google. Android Native Development Kit (NDK) - это набор инструментов для реализации части приложения, разрабатываемой на C, C ++, C #, - позволяет использовать библиотеки и компоненты приложения, написанные на языке «C» и других языках.

Сегодня Android считается одной из самых известных и самых популярных мобильных операционных систем.

Если у вас есть известные смарт-устройства, такие как Samsung, LG, Nexus, Xiaomi либо любое иное устройство, которое не приходит от Apple или Microsoft, вы являетесь постоянным пользователем системы Android.

Операционная система Android была специально разработана для интуитивного управления жестами, и одно из первых коммерческих устройств в этой системе спроектировано в 2008 году.

Сегодня самыми последними и широко используемыми версиями системы являются: Android 7.0 Nougat, Android 8.0 "Oreo" и Android, а десятая версия системы Android Q находится в стадии разработки. Под систему были разработаны, а также находятся в процессе разработки, очень много приложений, которые, непосредственно, можно скачать через Google Play или же с других сторонних ресурсов.

Стоит отметить, что ОС Android является основой многих других операционных систем под кастомные и краудфандинговые девайсы вследствие открытому коду для разработчиков, и она была переведена уже на более 100 языков мира.

По окончании 2016 года, по наблюдениям с различных источников выяснилось, что в среднем доля смартфонов с ОС Android на рынке смартфонов превышает 86% от общего количества проданных смартфонов в мире. Всего за 2016 год было распродано более 344 миллионов устройств, а в 2018 году рынок смартфонов достиг 522 миллиардов долларов.

Хотя и ОС Android разработана специально для смартфонов и планшетов, существуют также версии:

- Android Wear для носимых устройств, в основном это «умные» часы;
- Android Auto для автомобилей;
- Android TV для умных телевизоров.

На базе ОС Android работает большое количество девайсов, которые по количеству устройств превысили основные конкурирующие системы iOS и Windows Mobile.

1.4 Архитектура Android

По мнению разработчиков, платформа Android программиста от ядра, тем самым позволяя ему писать код на Java.

У ОС Android есть ряд важнейших свойств. Первое, эта база данных предоставляет огромный ассортимент API в целях конструирования разных видов приложений и возможность вторично использовать и заменять элементы, предоставляемых приложениями платформы, также косвенными поставщиками. Виртуальная машина Dalvik будет отвечать за запуск приложений на Android. Помимо этого, сервисы многих программистов графической библиотеки для 2D и 3D-приложений придерживаются мультимедийных форматов (Ogg Vorbis, MP3, MPEG-4, H.264, PNG), API с целью доступа к камере, GPS, компас, акселерометр, сенсорный экран, джойстик или клавиатура. Также есть специальный API с целью воспроизведения фоновых звуков, необходимых при разработке игр. Не во всяком устройстве Android целиком имеются данные функции. Естественно, Android не исчерпал список функций, упомянутых здесь. Тем не менее, они являются наиболее важными при разработке игр.

Архитектура Android состоит из множества элементов. Всякий элемент основан на элементах нижнего уровня. На рисунке 1.6 показаны главные элементы Android. Нижняя часть рисунка 1.6 показывает, что ядро Linux предоставляет главные драйверы для аппаратных элементов системы. Помимо этого, ядро отвечает за память, управляет процессами, поддерживает сети и так далее. Среда выполнения Android, которая построена вокруг ядра, ответственна за разработку и внедрение приложений для Android. Каждая программа работает в своем собственном процессе с собственной виртуальной машиной Dalvik.

Dalvik включает программы в формате байт-кода DEX. Файлы Java, содержащие класс, конвертируются в формат DEX при помощи специализированного инструмента DEX, доступного в SDK [3]. Формату DEX приходится значительно мало места в памяти, чем классическому типу файла CLASS, непосредственно приводя к высокому уровню сжатия, разбиению на массивы и комбинации нескольких файлов CLASS.

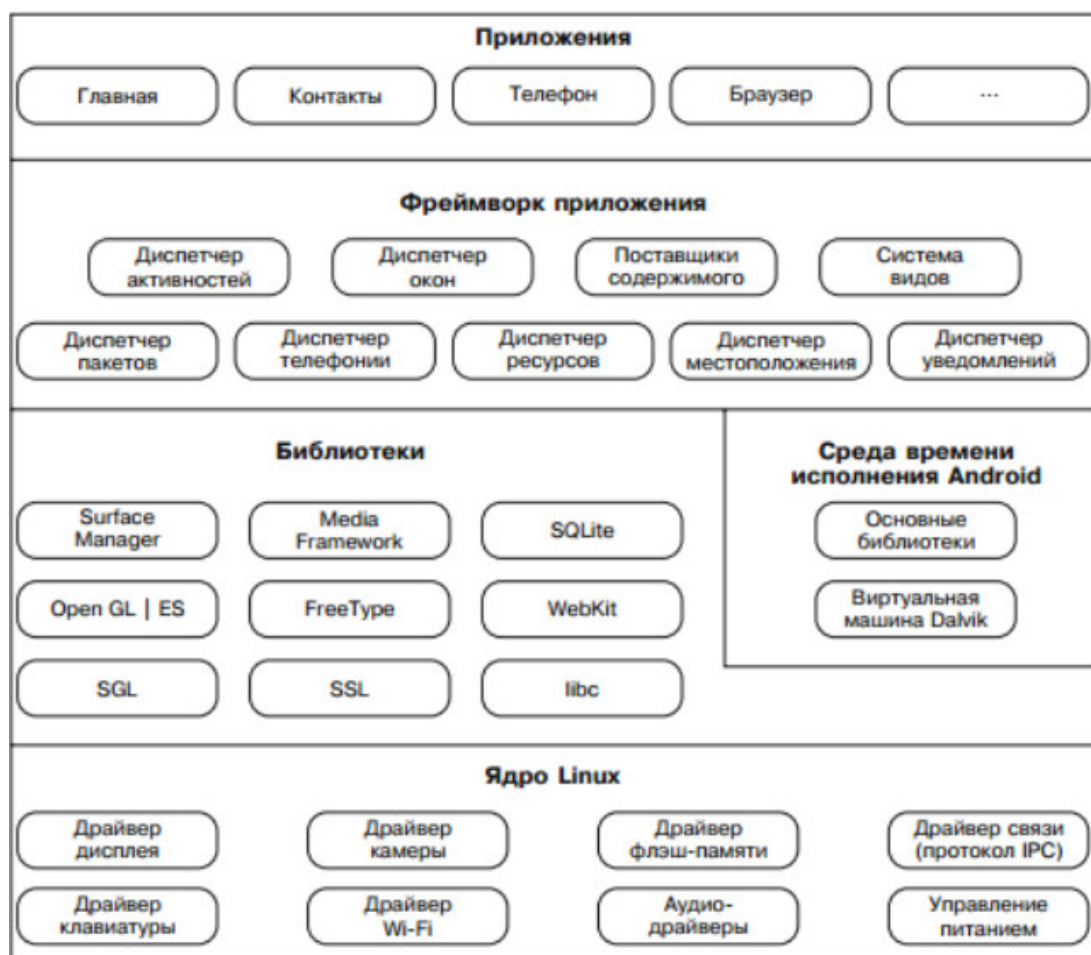


Рисунок 1.6 – Результат обзора архитектуры

Виртуальная машина Dalvik работает с библиотеками ядра, предоставляющие фундаментальные функциональные возможности для

программ Java. В этих библиотеках есть огромный, однако, не полный перечень классов, доступных через Java SE.

До Android 2.2 (Froyo) весь код был приписан. Froyo представляет компилятор JIT Tracking, представляющий часть двоичного кода на лету при компиляции машинного кода. Это существенно увеличивает продуктивность приложений, требующих большей вычислительной мощности. JIT-компилятор использует преимущества намеренно созданных процессоров для сложнейших процессов, таких как операции с плавающей запятой. Это был также сборщик чистых отходов, Dalvik (сборщик отходов, GC). Он использует правило «уведомляй и бери», тем самым приводит разработчиков в тупиковое положение. Однако тщательное изучение позволяет эффективно использовать его при разработке игр. Каждое приложение в оригинале виртуальной машины Dalvik имеет от 16 до 24 МБ ОЗУ. Это следует учитывать при использовании визуальных и звуковых ресурсов.

В приложение к фундаментальным библиотекам, предлагающим некоторые функциональные возможности Java SE, в C / C ++ также есть много чистых библиотек, помогающие заложить базу для использования структуры. Эти системные библиотеки главным образом отвечают за ряд действий (разрисовка графики, воспроизведение звука, доступ к базе данных), которые не очень подходят для Dalvik VM. API использует Java-классы внутри приложения для написания игры.

Фреймворк соединяет системные библиотеки и операционные среды, позволяет получить разрешение к платформе Android. Структура владеет приложениями и предоставляет составную среду, где производится работа. Программисты разрабатывают приложения данной платформе с ассортиментом API Java в таких областях, как реализация пользовательского интерфейса, фоновые службы, уведомления, управление ресурсами, устройство доступа и т. Д. Все важные типы приложений, оснащенные операционной системой Android (почтовый клиент), пишутся через API. Приложения, службы интерфейса или фоновые службы могут работать с иными приложениями. Это соединение позволит использование приложения иных компонентов. Небольшой пример – программа, записывающая и обрабатывающая фотографии. Приложение запросит у системы другой элемент приложения, обеспечивающий эту функцию. Кроме того, первоначальное приложение сможет повторно пользоваться этим компонентом (с камерой, установленной в приложении, или из фотогалереи).

Данный алгоритм удаляет значительную часть бремени от программиста, и помогает настроить поведение многообразия аспектов Android. Для реализации собственного приложения следует рассмотреть инструменты комплекта для разработки программ Android Software Development Kit (SDK).

1.5 Классификация мобильных приложений

Разработка мобильных приложений ныне является одной из самых популярных задач в области информационных технологий. Мобильная разработка направлена на реализацию приложений, позволяющих прогнозировать потребительские предпочтения с целью решения проблем, способы решения которых заранее неизвестны. Приложения теперь анализируют информацию из различных источников одновременно и помогают пользователям принять решения, контролируют процессы и решают иные важнейшие задачи с минимальной затратой времени и средств для пользователя. Это, в свой черед, поможет запустить бизнес-процессы, повысить продуктивность и принимать эффективные решения [4].

Мобильное приложение – специализированная программа для телефона, которая устанавливается на платформе и обладает с заданными функциями и позволяет выполнять определенные действия исходя от решаемых задач. Это своего рода адаптер, который помогает пользователю взаимодействовать с большим количеством информации.

В связи с этим различают:

- приложения-события, предназначенные для трансляции спортивных или иных событий;
- приложения службы, являющиеся аналогами сайтов, отражающих деятельность организаций;
- игры, в том числе развивающие и обучающие;
- интернет магазины, разрабатываемые для покупок в онлайн-режиме;
- промо-приложения, используемые для рекламы различных брендов;
- бизнес-приложения, позволяющие оптимизировать процесс работы организации, обеспечивая доступ к деловой информации и интеграцию с базами данных;
- системные приложения, использующие дополнительные настройки и опции телефона и его программного обеспечения;
- навигационные и поисковые сервисы, применяющие GPS-модуль, что позволяет использовать телефон как полноценный навигатор;
- мультимедийные приложения, расширяющие возможности телефона при работе с видео и аудиоинформацией;
- социальные сети, представляющие собой онлайн-сервисы для общения, распространения информации и организации социальных взаимоотношений;
- контентные приложения и др.

Новая концепция образования повышает значимость мобильных приложений для процесса обучения. В основном это связано с их доступностью, высокой пропускной способностью и привлекательностью для новых информационных технологий. Мобильные образовательные приложения позволяют учащимся совместно выполнять задания, учиться за пределами учебных заведений, имеют равные возможности для участия в

учебном процессе. Применение мобильных приложений в образовании позволяет:

- осуществлять единый контроль уровня знаний студентов;
- упрощение проведения зачетов и контроля;
- ускорять обмен информацией между всеми соучастниками образовательного процесса и облегчить этап взаимодействия преподавателей и учащихся;
- укрепление и усовершенствование учебных занятий;
- применение мобильных устройств для удаленного образования.

Современные мобильные компьютеры или технологии машинного зрения основаны на сравнении и поиске моделей в картинках. Благодаря компьютерному зрению приложения могут считывать штрих-коды, распознавать лица, быстро подсчитывать необходимые объекты и т. д.

Технология компьютерной визуализации широко используется в разработке медицинского программного обеспечения, где очень важно сравнивать большое количество медицинских показателей, включая изображения разных пациентов, для определения диагноза. Технология дополненной реальности – одна из наиболее перспективных областей развития компьютерного зрения (обнаруженный объект дополняется информацией, другим объектом или другими действиями приложения).

Если вы знакомы с языком разметки гипертекста HTML, а также с CSS (каскадными таблицами стилей), вы можете создать простую статическую страницу. С помощью HTML создаются теги содержимого будущего веб-сайта, а внешний вид настраивается с помощью CSS. Для создания более сложных веб-сайтов вам необходимо знание языка программирования PHP-скриптов. Применение различных технологий, которые позволяют сделать приложение более красивым, динамичным и быстро. Такое восстановление займет много времени, как показывает практика, есть много ошибок и работы. Для исправления этих ошибок существует одно такое программное обеспечение, которое облегчает разработку и объединение различных компонентов большого программного проекта, называемого фреймворком. Это программная системная структура (или подсистема), которая может включать в себя утилиты, библиотеки кода, язык сценариев и другое программное обеспечение для облегчения разработки и объединения различных компонентов большого программного проекта. Соединение обычно осуществляется через один API.

Фреймворк определяется как набор конкретных и абстрактных классов, а также определения их взаимосвязи. Определенные классы обычно реализуют взаимоотношения между классами. Абстрактные классы являются точками расширения, где можно использовать или настраивать фреймворки.

Третий самый простой способ создания веб-сайта – это использование системы для обеспечения и организации процесса создания, изменения и управления содержимым страницы (CMS). По сути, CMS является модульной, а управление контентом осуществляется через отдельные модули. В

большинство CMS встроено множество готовых шаблонов, а также редактируемую страницу готовых шаблонов. Для производителей мобильных приложений типология мобильных приложений по структуре важнее. Нативные приложения пишутся на языках программирования для определенной платформы и интегрированы в операционную систему, работают ускоренно и правильно, имеют привилегии перед функциональностью и скоростью использования иных мобильных приложений. Они предоставляют возможность наиболее естественным образом разработать внешний вид и общие положения ПО для заданной платформы. Приложения помимо этого имеют доступ к аппаратной части камеры: камера, микрофон, акселерометр, телефонная книга и т. д., экономят расходы на ресурсы, имеют возможность работать полностью или частично при выключенном интернет-соединении. Конечно, они требуют больших затрат времени и средств для того, чтобы написать приложение, что, в свою очередь, требует от разработчика специальных знаний о среде разработки, а также потому, что каждая платформа имеет свой собственный язык программирования.

Нативные приложения характеризуются своими инструментами и языками программирования. Например, для написания программ под управлением операционной системы Android чаще всего используется AndroidStudio, а для Windows Phone – язык Java для iOS Xcode и Objective-C, а также язык программирования Swift, Visual Studio и C#. Но есть ситуации, в которых необходимо написать приложение за меньшее время, в отличие от нативного, и оно работает на нескольких платформах одновременно.

В этом случае вы выбираете гибридные или мобильные веб-приложения, и для разработки используются различные мобильные кроссплатформенные платформы.

Веб-приложение следует назвать мобильной версией веб-сайта с развитым интерактивным интерфейсом. Их не размещают в магазинах, но для работы пользуются браузером. Производительность работы этих приложений зависит от качества интернет-соединения, а также они невысоки и быстры в реализации, кроссплатформенны, используют стандартные веб-технологии: HTML5, JavaScript и CSS.

Гибридные приложения или генераторы мобильных приложений – выступают посредником среди нативных приложений и веб-приложений, тем самым объединяя в себе кроссплатформенную технологию с возможностью применения программного обеспечения мобильного устройства.

Установленные официальными магазинами, имеют ограниченный доступ к аппаратной части устройства и возможность обновлять независимые данные, функциональность и качество ближе к родным приложениям, но дешевле для них, в зависимости от платформы, используемой разработчиком этого приложения.

1.6 Анализ известных решений и программных средств

В современной глобальной сети Интернет большое развитие получили и были представлены пользователям различные сервисы, которые дают новые возможности для саморазвития человека.

Перед тем как браться за реализацию приложения для обучения хореографии «Dance life» необходимо рассмотреть уже имеющиеся приложения иных танцевальных студий и школ, выявить их достоинства и недостатки.

Во всем мире существуют множество различных стилей танцев, следовательно, обычно список, предлагаемых различных танцевальных программ в школе, может быть весьма большим, либо наоборот, он будет специализироваться на одном виде танца. Любая школа танцев может сама выбрать направление своей деятельности. Есть школы, специализирующиеся только на латинских танцах, таких как мамбо, ча-ча-ча, румба, сальса, меренге, самба. Вторые школы могут преподавать только классические танцы – вальс, танго и другие. Третьи школы предпочитают танцы живота, клубные танцы, ирландские танцы и другие танцевальные направления.

Есть также танцевальные школы, которые предлагают обучение любому танцу, независимо от стиля. Обучение танцам обычно проводится два-три раза в неделю и час. С первого взгляда на приложение, концепция и направление, уровень компетентности руководителей и студентов становятся понятными.

Мобильное приложение «Just Dance Now».

Описание. Сыграйте в любимейшую музыкальную игру в любой точке мира без консоли. Все, что вам понадобится, дабы превратить любую комнату в сумасшедшую танцплощадку, – это экран, который подключен к Интернету, и ваш смартфон в качестве контроллера (рисунок 1.7)!

Вы попадете в фантастический мир Just Dance так легко и увлекательно, а также получите доступ к постоянно пополняемому каталогу из 300 с лишним треков!

В Just Dance Now вы найдете все, что вам нравится – от хитов наших дней до нестареющей классики!

– находится в топе поисковой системы по запросу «Мобильное приложение для танцев»;

– очень удобный и яркий интерфейс;

– возможность подключения ПК, ноутбук, Smart TV или планшет – любое устройство с экраном и выходом в интернет. Все действия будут происходить на большом экране в окне браузера, смартфон же станет контроллером движения.

В нынешнее время, время бурного роста интернет-технологий, есть самые разнообразные способы разработки приложений.

iDance!: приложение для изучения танцев. В приложении доступно 12 танцевальных стилей, которые разделены на различные уровни сложности

(рисунок 1.8). Можно не только повторять движения танцев, известные тренеру, но и создавать свои собственные, смешивая различные элементы.

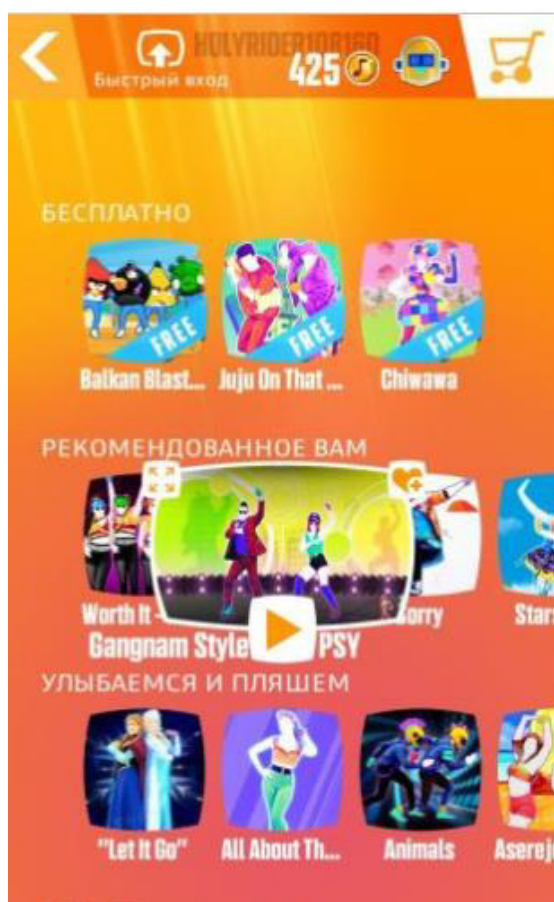


Рисунок 1.7 – Мобильное приложение «Just Dance Now»

Для него вы можете найти имя и отправить друзьям по почте.

Помимо настоящего танца в приложении есть несколько силовых тренировок для разминки. Компания также известна приложением Brain and Nervous System PRO III, которое показывает нервную систему человека до мельчайших деталей. iDance! доступно для скачивания в App Store всего за 3 доллара.

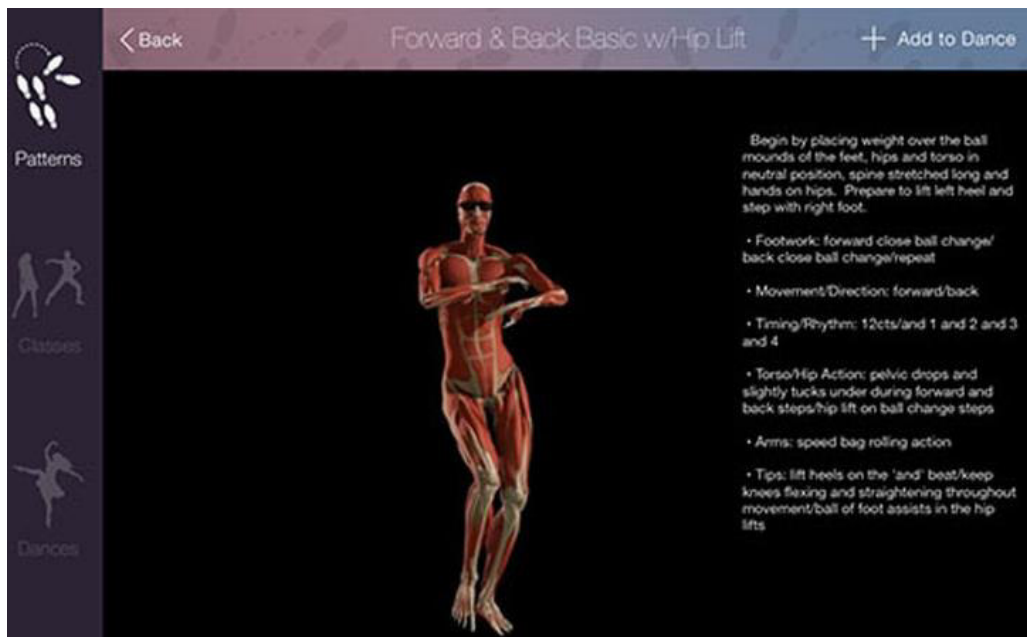


Рисунок 1.8 – iDance!: приложение для изучения танцев

The Dance App. Что ж, это приложение может также заинтересовать профессионалов – оно содержит записи мастер-классов разных хореографов: от Дежана Тубича до Джанель Джинестры (рисунок 1.9). В результате список танцевальных стилей, представленных хореографами, довольно обширен: от современного до хип-хопа. Это не столько обучающие видеоролики, сколько мультимедийное разнообразие мастер-класса - вы сможете почувствовать атмосферу тренинга. Но не думайте, что это будет стоить вам всего бесплатно. Само приложение стоит 33 рубля, и за каждый отдельный мастер-класс взимается дополнительная плата. Доступно в Google Play.

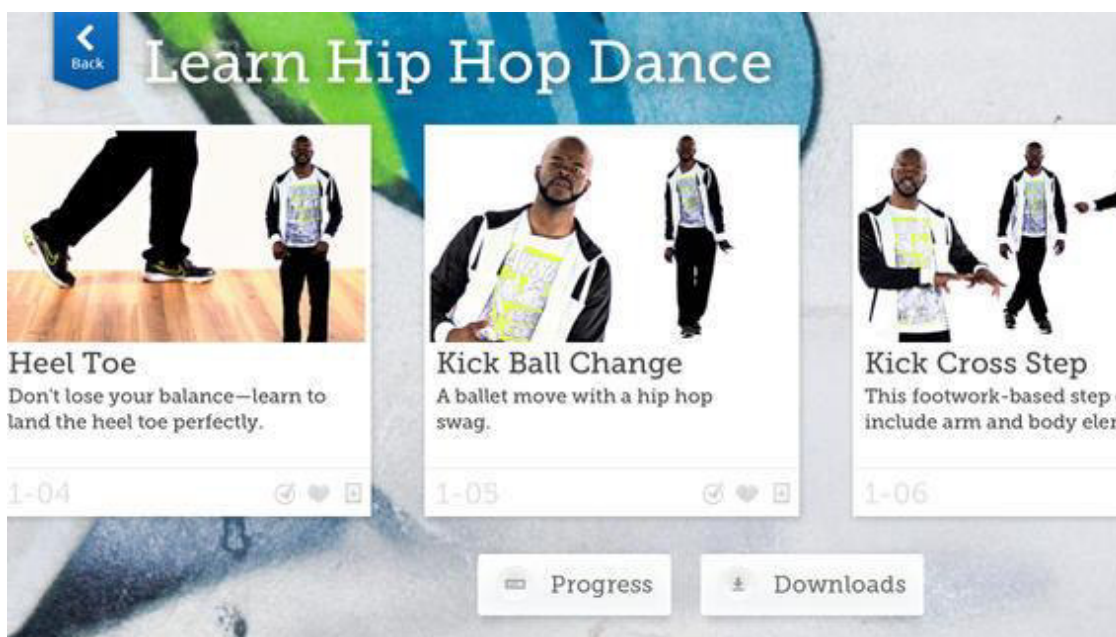


Рисунок 1.9 – The Dance App.

Learn Hip Hop Dance with Brice Johnson. Конечно, в сети много танцевальных событий, и на эту тему есть много мобильных приложений с различными стилями (рисунок 1.10). Но данное приложение требует чувства юмора и качества исполнения. Большой выбор движений и четкое описание того, насколько велик мир хип-хоп танцев.

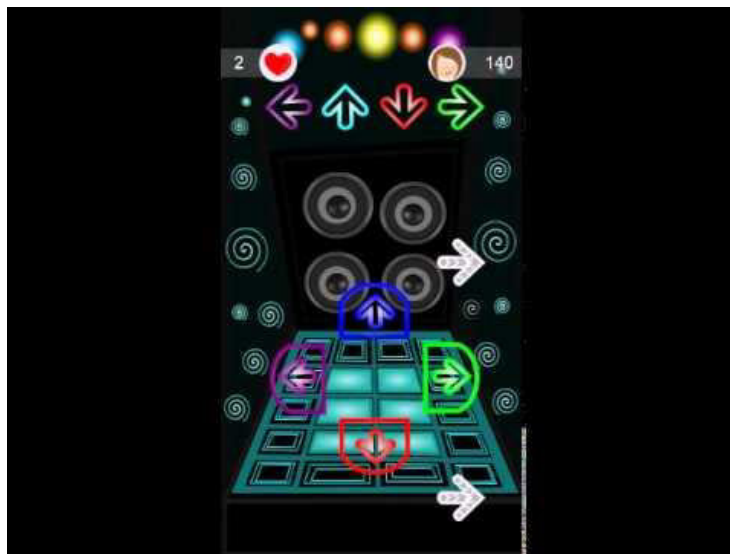


Рисунок 1.10 – Learn Hip Hop Dance with Brice Johnson

Dance Evolution. Что ж, одна из популярнейших танцевальных игр за последние годы, хорошо развита в мобильной версии (рисунок 1.11). Все просто – иди туда, где горит стрелка, машете рукой в направлении, которое светится на экране. С профессиональным танцем – ничего общего с ним, но вы можете повеселиться с друзьями.

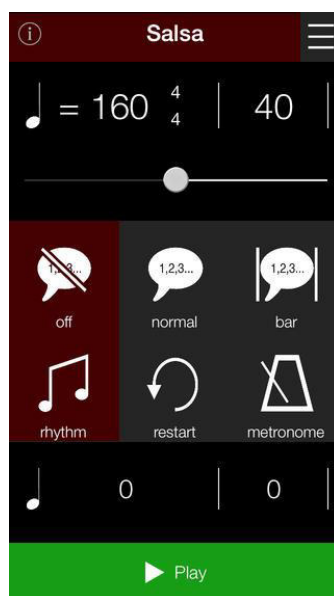


Рисунок 1.11 – Dance Evolution

2 Программная часть

2.1 Выбор средств для реализации приложения Android

Для разработки мобильных приложений под операционную систему Android можно использовать различные средства и среды разработок. Рассмотрим пару таких программных средств [5].

Kotlin. Официально запущенный в Google I / O в мае 2017 года, Google позиционирует его как второй официальный язык программирования для Android и Java, который немного легче понять. Нужно знать Java, чтобы понимать работу на Kotlin, общую структуру языка и его функции. Множество разработчиков рассматривают Kotlin как Java, Shell и Recommended, это первое, что нужно изучить, если вы знакомы с Java Knowledge.

Kotlin совместим с Java и не приводит к потере производительности или увеличению размера файла. Отличие Java заключается в том, что необходим меньший код, так называемая модель кода, которая оптимизирует и более удобочитаема. Создатели смогли избежать нулевой терпимости, и композиция больше не прерывается мелочами как забытый знак “;”

C / C++. Дополнительные языки, также поддерживаются Android Studio и используют Java NDK. Таким образом, вы сможете написать собственные приложения для разработки игр либо других программ ресурсов. Android Studio предлагает поддержку Android NDK (Native Development Kit) для C / C ++, что, в свою очередь, будет значить, что код находится не на виртуальной машине Java, а непосредственно на устройстве, что дает вам больше контроля над системными компонентами, такие как память, датчики, жесты и т. д., а также возможность извлечения максимальных ресурсов устройств Android. Это также будет значить, что вы должны использовать библиотеки, написанные на C или C ++.

Его сложно настроить и не слишком удобно, поэтому рекомендуется писать только те модули программы, для которых нужно мгновенно выполнить сложные операции: обработку и воспроизведение графики, сложного видео и 3D-моделей.

Python. Факт того, что Android не придерживается того, чтобы использовать Python для разработки своих приложений, не значит, что так нельзя. Сторонники данного языка реализовали многообразное количество средств с целью компиляции кода на Python в хорошем состоянии. И то, что имеются различные библиотеки дает возможность им создавать свои собственные интерфейсы, используя ручные линии. Одна из популярных частей Kivy, позволяющая создавать приложения для Play Market на чистом Python.

Basic. Ввиду своей простоты и удобства, Basic считается идеальной отправной точкой для начинающих программистов.

Неважная новость заключается в том, что он не из Android Studio, и не поддерживается для Unity и Xamarin. Хорошая новость заключается в том, что

Basic – специальная разработка для b4a Apps для Android. B4A воплощает концепцию визуального прототипирования колесика, что подразумевает скорость и простоту программирования. Существует множество небольших практических дизайнерских решений, и сообщество быстрого реагирования поможет решить любые проблемы.

Недостатком является то, что Basic-интенсивные игры не должны создаваться и поддерживаться Google из-за неофициального состояния THE count. Это означает, что приложения не соответствуют требованиям дизайна материалов и не могут считаться профессиональными разработчиками с фундаментальными знаниями. И среди языков и сред, представленных в этой статье, этот язык является единственным платным.

Lua (используя Corona SDK). Баланс между простотой изучения Android, разработкой и ощущением контроля со стороны разработчика. Графический движок Corona основан на языке Lua. LUA намного проще, чем Java и Corona SDK. Работать с этим языком просто и приятно. Он поддерживает все свои библиотеки, что позволяет им писать для разных платформ.

Чтобы написать код, вам нужен Notepad ++, а для его запуска без предварительной компиляции необходим эмулятор. После того, как PСA будет создан и программа заработает, вы сможете запустить приложение через онлайн-инструмент. С базовыми навыками программирования, чтобы преодолеть развитие Android от Corona, вы можете сделать это.

Не было никаких ограничений, но были ограничения, которые мешали вам развивать серьезные вещи и получить профессиональный статус. Если вам нужны такие функции в приложении, как внутренние покупки, вы должны заплатить за возможность своей разработки и за использование собственного Android API.

PhoneGap. Если вы уже знаете что-либо о веб-программировании в целом и HTML, CSS и JavaScript в частности, эти знания будут полезны для создания приложений базовой платформы под PhoneGap. Это последний из нашего списка, но не последний по важности, способ покорить развитие платформы Android. PhoneGap позволяет в принципе разрабатывать приложения для Android через веб-разработку. Ваше приложение отображается в WebView, но также, как в Wrapper для мобильных приложений. Для разработчиков PhoneGap подобен мосту для доступа ко встроенным функциям смартфона или планшета, таким как датчик ускорения или камера.

2.2 Среда программирования Android Studio

Несмотря на многообразие средств разработок, было принято решение остановить свой выбор на программном продукте Android Studio.

Программный продукт Android Studio был выбран в качестве среды разработки для приложения Android. Это интегрированная среда разработки (IDE) для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O.

Среда IDE доступна бесплатно с версии 0.1, выпущенной в мае 2013 года, и вступила в фазу бета-тестирования с версии 0.8, выпущенной в июне 2014 года. Первый стабильный выпуск 1.0 был выпущен в декабре 2014 года, тогда как разработка Android (ADT) для Eclipse не проводилась. дольше поддерживается [6].

Android Studio, основанная на программном обеспечении IntelliJ IDEA JetBrains, является официальным инструментом для разработки приложений для Android. Эта среда разработки доступна для Windows, OS X и Linux. Основными особенностями этого региона являются возможности в реальном времени, макеты, а также различные размеры экрана и разрешения. Есть раздел справки. Интегрированные инструменты для улучшения качества приложений. Это легко написать код. Сборные модели для основных схем и компонентов Android. Существуют инструменты эффективности рекламы. Добавлен инструмент для взаимодействия с бета-тестерами. И многое другое.

Формальный язык программирования, поддерживаемый средой разработки Android Studio. Согласно ежегодному исследованию ресурсов Stackoverflow, проведенному в 2018 году, Java один из пяти самых популярных языков программирования.

Первым шагом к освоению Java является установка Android Studio. Это разновидность программного обеспечения под названием IDE (Integrated Development Environment), интегрированная среда разработки или интегрированная среда разработки. Android Studio – это Android SDK, предоставляющий набор инструментов для Android-разработки и все, что вам нужно для надежного получения.

Java относится к большинству официальных документов Google, и поиск бесплатных и платных библиотек и учебников будет легким – они огромны.

Большинство приложений Android, созданных в режиме реального времени, написаны на Java. Это позволило нам реализовать большое количество проектов и в полной мере использовать возможности, предлагаемые системой Android. Кроме того, знание языка Java позволяет корпоративным разработчикам не только создавать новые приложения, но и поддерживать существующие приложения с использованием существующего исходного кода.

К сожалению, сложность Java не позволяет никому программировать. Как объектно-ориентированный язык программирования, он имеет много оригинальности в виде конструкторов классов, включенных в приложения, при использовании системы аварийного завершения и других вещей, которые всегда должны учитываться при разработке. Тем не менее, код Java легко читается и структурируется, особенно если соблюдать уважаемые стандарты его дизайна.

Если вы разрабатываете Java для Android не только для классов Java, которые содержат код, но и для манифестов в XML, которые предоставляют основную информацию о программе, и автоматизированной системе сборки

для команд Gradle, Maven или Ant, каждая из которых написана на Groovy, POM и XML; По умолчанию в проектах используется Gradle, а на начальном этапе изучения Java для обработки файлов, которые вряд ли были записаны в Groovy, это необходимо сделать. XML также часто используется для макета части интерфейса пользователя.

Android Studio, в декабре 2014 года от Google в качестве среды разработки для ОС Android была принята, чтобы улучшать каждый год Android, и разработчики облегчают жизнь. Его функции, такие как визуальный редактор для пользовательского интерфейса и автозавершения кода, помогают упростить процесс разработки.

Особые свойства. С каждой новой версией Android Studio появляются новые функции. В настоящее время доступны следующие функции:

- Редактор макетов: WYSIWYG, функция перетаскивания, предварительный просмотр макетов на нескольких экранах.

- Создание приложений из Gradle.

- Различные типы сборки и генерации нескольких. Apk файлов. Код рефакторинга.

- Статический анализатор кода (Lint) для выявления проблем с производительностью, несовместимости версий и многого другого.

- ProGuard – это интегрированный инструмент для подписи приложений. Модели и компоненты, необходимые для Android.

- Поддержка разработки приложений для Android Wear, Android TV.

- Интегрированная поддержка Google Cloud Platform, интеграция с Google Cloud Messaging и App Engine.

- Android Studio 2.1 поддерживает Android SDK n Preview, что означает, что разработчики смогут создавать приложения для новой программной платформы.

- Новая версия Android Studio 2.1 способна работать с обновленной версией компилятора Jack, а также с улучшенной поддержкой Java 8 и функцией мгновенного запуска.

- Из Platform-tools 23.1.0 Версия для Linux 64-битная.

В Android Studio 3.0 стандарт включает в себя инструменты языка Kotlin на основе JetBrains IDE.

2.3 Язык программирования Java

Java – объектно-ориентированный язык программирования, которую выпустила компания Sun Microsystems в 1995 году, главным образом является основным компонентом платформы Java. В настоящее время разработкой языка занимается компания Oracle, приобретенная компанией Sun Microsystems в 2009 году. Синтаксис языка во многих случаях похож на C/C++. В официальной реализации, Java приложения компилируются в байт-код, который при компиляции интерпретируется виртуальной машиной для конкретной платформы [7].

Язык во многом позаимствовал синтаксис от С и С ++. В основном, взята за основу объектная модель С++, однако она значительно обновлена. Были устранены возможности появления некоторых конфликтных ситуаций, которые могли возникнуть из-за ошибки программиста и облегчен сам процесс разработки объектно-ориентированных программ. Ряд функций, которые должны осуществлять программисты, поручено виртуальной машине.

Прежде всего, Java разрабатывалась как платформа с независимым языком, потому она имеет меньше низкоуровневых возможностей для работы с аппаратным обеспечением. Если это необходимо Java позволяет вызвать подпрограммы, написанные другими языками программирования.

При разработке языка программирования Java изначально было сформировано четыре начальные цели:

- синтаксис языка должен быть простым, объектно-ориентировочным и привычным;
- реализация исполняемых приложений должна быть безотказной и безопасной, а также высокопроизводительной;
- должна сохраниться независимость от архитектуры и портативность;
- язык должен быть динамичным, интерпретируемым и поддерживать мультизадачность.

Под «независимостью от архитектуры» имеется в виду то, что программа, написанная на языке Java, будет работать на любой поддерживаемой аппаратной или системной платформе без изменений в исходном коде и перекомпиляции. Этого можно достичь, компилируя начальный Java код в байт-код, который представляет собой упрощенные машинные команды. Затем программу можно выполнить на любой платформе, которая имеет установленную JVM (виртуальную машину Java), способную интерпретировать байткод в код, учитывающий специфику конкретной ОС и процессора.

Сейчас виртуальные машины Java существуют для большинства процессоров и операционных систем. Основное преимущество использования байт-кода – это портативность. Тем не менее, дополнительные расходы на интерпретацию означает, что интерпретированные программы будут практически всегда работать медленнее, чем скомпилированные в машинный код, и именно поэтому Java получила репутацию медленного языка. Однако, этот разрыв существенно сократился после введения нескольких методов оптимизации в современных реализациях JVM.

2.4 Эмулятор для разработки приложений Android SDK

При разработке приложений используется высокоуровневый интерфейс программирования Java для Android, который позволяет создавать приложения для конечных пользователей Android. Рассмотрим функции эмулятора Android, основные компоненты Android и пакеты, включенные в SDK. Несколько выдержек из кода также предоставляются [8].

Пакет Software Development Kit (SDK) для Android поставляется с Android Studio, плагином, который называется Android Development Toolkit (ADT). С помощью этого инструмента разработки (IDE) вы можете создавать, отлаживать и тестировать приложения Java. Android SDK можно использовать без ADT, вместо использования инструментов вы также можете использовать инструменты командной строки. Эмулятор поддерживает использование обоих подходов, что позволяет запускать, восстанавливать и проверять приложения. 90% разработки приложений может быть завершено даже без использования реального устройства. Полнофункциональный эмулятор для Android воспроизводит наиболее часто изучаемые функции устройства. Функции, которые не могут быть смоделированы в эмуляторе, включают порт USB, работу камеры и видео, имитацию шлема, работу от аккумулятора и технологию Bluetooth.

Эмулятор Android основан на открытой технологии «Имитация» процессора QEMU, разработанной Bellar. Та же технология может эмулировать одну операционную систему на 16 других, независимо от используемого процессора. QEMU обеспечивает эмуляцию уровня процессора.

Если вы используете эмулятор Android, имитируется процессор на основе ARM (Advanced RISC Machine, Advanced Riscmachine). ARM – это 32-разрядная микропроцессорная архитектура, основанная на RISC (компьютер с сокращенным набором инструкций), которая обеспечивает простую конструкцию и повышает производительность за счет сокращения числа контроллеров. Эмулятор использует процессор в моделируемой версии Linux, которая используется в Android.

ARM широко используется в мобильных устройствах и интегрированных электронных устройствах, где важно распределять небольшое количество энергии. Многие мобильные устройства для продажи имеют процессоры с этой архитектурой. Например, Apple Newton основан на процессоре ARM6. Игровые автоматы, Nintendo DS и Game Boy Advance работают на архитектуре ARM версии 4, которая использует около 30 000 транзисторов. Классический Pentium содержит 3,2 миллиона транзисторов.

2.5 Пакеты, которые являются частью Android SDK

Чтобы получить представление о платформе Android, вам необходимо рассмотреть структуру пакетов Java. Поскольку Android SDK отличается от стандартной версии, важно знать, какие пакеты поддерживаются, а какие нет.

Ниже приводится краткое описание важных пакетов, составляющих Android SDK:

– Android.app – реализует модель приложения Android. Среди основных классов есть приложение, которое описывает начальную и конечную семантику, а также ряд классов по явлениям, элементам управления, диалоговым окнам, окнам предупреждений и уведомлениям;

– `Android.Bluetooth` – включает занятия по работе с технологией Bluetooth. Основные классы Bluetooth-адаптер, Bluetooth-устройство, разъем Bluetooth, разъем сервера Bluetooth и `BluetoothClass`. Класс адаптера Bluetooth можно использовать для управления адаптером Bluetooth, установленным на локальном компьютере. Этот адаптер может включать, отключать или запускать процесс обнаружения. Класс устройства Bluetooth – это удаленное устройство Bluetooth, к которому вы можете подключиться. Для связи по Bluetooth между устройствами используются два разъема. Класс Bluetooth – это тип устройства Bluetooth, к которому вы подключены.

– `Android.content` – реализует концепции, связанные с поставщиками контента. Контент-провайдер позволяет обобщать общий доступ и хранение данных. Кроме того, этот пакет реализует основные идеи о намерениях и общих идентификаторах ресурсов (URI) в Android;

– `Android.content.pm`. – предоставляет классы, которые можно использовать вместе с использованием диспетчера пакетов. Он содержит информацию о разрешениях, установленных пакетах, установленных поставщиках, услугах и компонентах, таких как рекламные акции, и установленных приложениях;

– `Android.content.res` – предоставляет доступ к файлам ресурсов, как структурированными, так и неструктурированным. Основные классы `AssetManager` (для неструктурированных ресурсов) и материальных ресурсов;

– `Android.database` – реализует идею абстракции базы данных. Основной интерфейс называется `Cursor`;

– `Android.database.sqlite` – реализует концепцию пакета базы данных Android и использует SQLite в качестве физической базы данных. `SQLiteCursor`, `SQLiteDatabase`, `SQLite Query`, `SQLiteQueryBuilder` и 25 базовых классов `SQLiteStatement`. Однако, прежде всего, необходимо работать с абстрактными классами пакетов баз данных Android;

– `Android.gesture` – этот пакет содержит все классы и интерфейсы, необходимые для работы с пользовательскими жестами. Основные классы `Gesture`, `GestureLibrary`, `GestureOverlayView`, `GestureStore`, `GestureStroke`, `GesturePoint`. Класс жестов – это выбор `GestureStrokes` и `GesturePoints`. Жесты собраны в библиотеке `GestureLibrary`. Библиотека жестов хранится в `GestureStore`. Названия жестов выбираются так, чтобы система могла идентифицировать их как действия;

– `Android.graphics` – содержит класс `Canvas`, `Camera`, `Color`, `Matrix`, `Movie`, `Path`, `Rasterizer`, `Shader`, `SweepGradient` font;

– `Android.graphics.drawable` – предназначенный для работы с протоколами рисования и фоновыми изображениями, он обеспечивает анимационные эффекты при работе с нарисованными от руки объектами;

– `Android.graphics.drawable.shapes` – предназначен для использования в схемах, таких как `ArcShape`, `OvalShape`, `PathShape`, `RectShape` и `RoundRectShape`;

– `Android.hardware` – позволяет использовать так называемые естественные классы, предназначенные для работы с камерой. Класс камеры – это обычная камера, а класс `android.graphics.camera` – это графическая концепция, которая не имеет ничего общего с реальной физической камерой;

– `Android.location` – содержит классы `Address`, `GeoCoder`, `Location`, `LocationManager` и `LocationProvider`. Класс `Address` упрощает обмен XML (`Extensible Address Language`). Геокодер позволяет узнать адрес координат объекта (широта и долгота) и наоборот. Прокат предоставляет информацию о широте и долготе;

– `Android.media` – содержит классы `Media Player`, `Media Recorder`, `Ringtone`, `AudioManager` и `FaceDetector`. Класс `MediaPlayer` предназначен для потоковой передачи аудио и видео. Класс рингтонов используется для воспроизведения коротких звуковых учетных записей, которые можно использовать в рингтонах или уведомлениях. `AudioManager` отвечает за регулировку громкости. `FaceDetector` может использоваться для распознавания лиц на точечном рисунке (растровое изображение);

– `Android.net` – реализует базовую сеть на сокетах API. Основные классы включают `Uri`, `ConnectivityManager`, локальный сокет и локальный `ServerSocket`. Также следует отметить, что Android поддерживает HTTPS-уровень браузера и сетевой уровень. Кроме того, Android поддерживает JavaScript в браузере;

– `Android.net.WiFi` – контролирует соединение Wi-Fi. Основные классы `WifiManager` и `WifiConfiguration`. Класс `WifiManager` отвечает за список настроенных сетей и работает с текущей активной сетью Wi-Fi;

– `Android.OpenGL` – содержит классы поддержки, используемые при выполнении операций OpenGL ES. Основные классы OpenGL ES являются частью другого пакета JSR 239.

– `Android.os` – служба операционной системы, к которой вы обращаетесь с помощью языка Java. Некоторые важные классы – `BatteryManager`, `Binder`, `FileObserver`, `Handler`, `Looper` и `Power Manager`. Класс `Binder` обеспечивает обмен информацией между процессами. `FileObserver` учитывает изменения, внесенные в файлы. Класс обработчика используется для выполнения задач в потоке сообщений, и `Looper` запускает само сообщение потока;

– `Android.preference` – позволяет приложениям предоставлять пользователям возможность управлять своими настройками для этого приложения в единой форме. Основные классы `PreferenceActivity`, `Getpreferencescreen`

– `Android.provider` – содержит ряд предопределенных поставщиков контента, связанных с Android – содержание, контент-провайдеры. Среди контент-провайдеров - Контакты, `MediaStore`, Браузер и Настройки. Этот набор интерфейсов и классов содержит метаданные, описывающие базовую структуру данных;

- `Android.sax` – предоставляет эффективный набор простых XML API (SAX), вспомогательных классов, разработанных для анализа. Основные классы интерфейсов `Element`, `RootElement` и `ElementListener`;
- `Android.speech` – содержит константы для распознавания речи. Этот пакет включен только в версии 1.6 и выше;
- `Android.speech.tts` – обеспечивает поддержку для преобразования текста в речь. Основным классом является `TextToSpeech`. В Android есть механизм Pico TTS (Text to Speech) производства SVOX;
- `Android.telephony` – содержит классы `CellLocation`, `PhoneNumberUtils` и `TelephonyManager`. `TelephonyManager` класс для определения местоположения, из которого был сделан вызов, номер телефона, имя поставщика услуг, тип сети, тип телефона и серийный номер модуля идентификации абонента (Subscriber Identity Module, SIM);
- `Android.telephony.gsm` – позволяет собирать информацию об адресе соты на основе местоположения данных мобильной вышки, но также включает классы, отвечающие за использование SMS-сообщений. Определяет глобальную систему мобильной радиосвязи в названии этого пакета, называемую GSM, в качестве исходного стандарта коротких сообщений (стандарт SMS) (Global Mobile Radio System);
- `Android.telephony.cdma` – поддерживает CDMA-телефонию;
- `Android.text` – содержит классы для обработки текста;
- `Android.text.method` – предоставляет классы для ввода текста в разные элементы управления;
- `Android.text.style` – обеспечивает разнообразие методов обработки текста;
- `Android.Utils` – содержит классы, `DebugUtils`, `TimeUtils` и `Xml`;
- `Android.view` – содержит классы, меню `View`, `ViewGroup`, а также некоторые из процессов слушателей и обратных вызовов;
- `Android.view.animation` – обеспечивает поддержку анимации в структуре промежуточных кадров;
- `Android.view.InputMethod` – реализует ввод-вывод системной архитектуры. Этот пакет содержится только в версиях 1.5 и выше;
- `Android.WebKit` – содержит классы, связанные с веб-браузером. К наиболее важным классам `WebView` относятся `CacheManager` и `CookieManager`;
- `Android.widget` – содержит все классы элементов управления пользовательского интерфейса, которые получены главным образом с точки зрения класса. Основные виджеты – Кнопка, Галочка, хронометр, `AnalogClock`, `DatePicker`, `EditText`, `ListView`, `FrameLayout`, `GridView`, `ImageButton`, `MediaController`, `ProgressBar`, `RadioButton`, `RadioGroup`, `RatingButton`, скроллер, `ScrollView`, `Spinner`, `TabWidget`, `TextView`, `TimePicker`, `VideoView` и `ZoomButton`;

– Com.google.android.maps – содержит классы MapView, MapController и MapActivity, необходимые для работы с Google Maps. Вышеуказанные пакеты очень важны для работы с Android. Из этого списка вы можете получить представление о глубокой структуре платформы Android. В целом API Java Android включает в себя более 40 пакетов и более 700 классов. Однако все эти многочисленные пакеты составляют богатую компьютерную платформу, предназначенную для написания программ для мобильных устройств.

2.6 Компоненты пользовательского интерфейса ANDROID

Пользовательский интерфейс, используемый в операционной системе Android, аналогичен другим полнофункциональным структурам пользовательского интерфейса на локальных компьютерах. Он более современный и асинхронный. Фактически, Android UI Framework принадлежит к четвертому поколению, учитывая первое поколение традиционного интерфейса программирования приложений Microsoft Windows, основанного на C и MFC (классы Microsoft Foundation, библиотеки базовых классов Microsoft на основе C++) – второе считается. В этом случае основанная на Java Uing Swing Framework будет третьим поколением и, как предполагается в ее возможностях проектирования, намного превосходит MFC с точки зрения гибкости. Пользовательские интерфейсы Android UI, JavaFX, Microsoft Silverlight и Mozilla XUL (XUL) включают новый тип пользовательского интерфейса четвертого поколения, который является декларативным пользовательским интерфейсом и поддерживает независимые темы.

Во время программирования пользовательский интерфейс Android использует объявление интерфейса в файлах XML. Затем определение представления XML загружается в приложение с пользовательским интерфейсом, похожим на Windows. Меню приложения также загружается из файла XML. Экраны Android (окна) часто называют действиями, которые охватывают несколько типов, и пользователи должны выполнять логический элемент процесса. Представления являются основными элементами интерфейса Android. Формы могут быть сгруппированы в группы (группы просмотра). Для внутренней организации такого рода они давно используют концепцию известного веба в программировании и взаимодействии пользователя с системой.

Эти составные представления, которые включают представления и группы представлений, работают с определенной логикой компонента пользовательского интерфейса Android.

Одной из ключевых концепций базы данных является управление жизненным циклом Android в Windows Event Activity (Activity Windows). Системные журналы позволяют Android управлять ситуацией, когда пользователь выполняет действия, чтобы скрыть, восстановить, остановить и закрыть окно.

Основные компоненты Android. Фреймворк для пользовательского интерфейса Android, как и другие компоненты Android, основан на новом объекте, намерении. Намерение – это сложное явление, которое объединяет такие идеи, как сообщения, отображаемые в окне, публикации моделей и действия по подписке, межпроцессный обмен информацией и реестры приложений. Ниже приведен пример использования класса Intent для активации или запуска веб-браузера:

```
public static void invokeWebBrowser (Activity activity)
{
    Intent intent = new Intent (Intent. ACTION_VIEW):
    intent. setData (Uri. parse ("http://www.google.com"));
    activity. startActivity (intent);
}
```

В этом примере мы говорим Android открыть соответствующее окно для отображения содержимого веб-сайта. В зависимости от браузера, установленного на вашем мобильном устройстве, Android выберет наиболее подходящий браузер для отображения веб-сайта.

Кроме того, ресурсы Android в основном поддерживаются, зная строки растрового изображения и, по крайней мере, некоторые известные элементы, такие как, например, определение представления на основе XML. Использование ресурсов в этом контексте будет осуществляться по-новому, что сделает работу с ресурсом более легкой, понятной и практичной.

Ниже приведен пример, в котором автоматически генерируется ID ресурса и определяется в файлах XML:

```
public final class R {
    public static final class attr { }
    public static final class drawable {
        public static final int myanimation=0x7f020001;
        public static final int numbers19=0x7f02000e:
    }
    public static final class id {
        public static final int textViewIdl=0x7f080003;
    }
    public static final class layout {
        public static final int frame_animations_layout=0x7f030001;
        public static final int main=0x7f030002;
    }
    public static final class string {
        public static final int hello=0x7f070000;
    }
}
```

Эти классы соответствуют автоматически генерируемому идентификатору или XML-файлу элемента или набора. Если вам нужно использовать эти определения XML вместо использования их

идентификационных данных. Такое посредничество очень полезно при локализации. Другой инновационной концепцией в Android является поставщик контента (контент-провайдер). Поставщик контента относится к абстракции источника данных, который может быть представлен как эмитент и служба REST (State Transfer). База данных основана на абстракции SQLite, что делает эту функцию поставщиков контента мощным инструментом разработки приложений. Было обнаружено, что XML играет важную роль в описании пользовательского интерфейса для Android.

Хотя Android поддерживает диалоговые окна, все они асинхронные. Работа с такими асинхронными диалогами особенно трудна для разработчиков, которые привыкли работать с синхронными модальными диалогами, используемыми в некоторых оконных рамах. Android также поддерживает анимацию – эта функция включена в сотрудничестве со стекком интерфейса пользователя на основе типов объектов. В Android используются анимации двух видов: с построением промежуточной анимации (анимация движения) и покадровой (покадровая анимация). Промежуточная анимация называется изображениями, отображаемыми между основными анимациями. На компьютере эти изображения периодически меняют средние значения и перерисовывают фон. Кадровая анимация состоит из последовательности изображений, которые рисуются через равные промежутки времени один за другим. В Android обе версии анимации используются через функцию напоминания о анимации, интерполяторы и матрицу преобразования. Кроме того, Android может идентифицировать этот тип анимации в файле ресурсов XML.

Графическая библиотека, основанная на этом процессе, поддерживает стандартные преобразования сетки, которые позволяют масштабировать, перемещать и поворачивать изображения. Объект камеры в графической библиотеке поддерживает глубину и проекцию, поэтому 2D-интерфейс может имитировать 3D-эффекты.

В Android доступна трехмерная графика. Причина этого заключается в том, что стандарт OpenGL 1.0 был введен. Как и OpenGL, он основан на польском языке API C. Поскольку программный интерфейс Android SDK основан на Java, OpenGL должен получить к нему доступ через ссылку Java. В Java ME является обязательным для OpenGL ES и уже определено с помощью запроса спецификации Java JSR 239, а для Android OpenGL ES использует тот же тип ссылки Java.

В Android используются новые подходы к представлению информации одним нажатием кнопки, которая доступна через домашнюю страницу. Первая из этих идей называется онлайн-каталогами (онлайн-файлами). Используя оперативные каталоги, коллекция элементов может быть опубликована в папке на главной странице сайта. Содержание этой коллекции варьируется от изменения базы данных до ее базы данных. Новые данные могут быть отправлены на устройство, со съемного носителя или из Интернета.

Вторая идея связана с домашней страницей – это виджет домашнего экрана. Основные виджеты используются для рисования на домашней странице с использованием интерфейса информационных виджетов. Эта информация может периодически изменяться. Примером этих данных может быть множество электронных сообщений, хранящихся на устройстве. Встроенный поиск Android (Integrated Android Search) – это третья идея, связанная с использованием домашней страницы. Этот тип поиска позволяет искать информацию как на устройстве и в Интернете. В Android эта функция не ограничивается только одним поиском и может давать команды с помощью элемента управления поиска.

Кроме того, так называемые поддерживаемые жесты Android, то есть интерпретация движений пальцев пользователя работы с устройством Android позволяет записывать любую последовательность движений пальцев на экране и сохранять их в виде жестов. Затем такие жесты могут использоваться приложениями для определения конкретных действий.

Кроме инструментов Android SDK, есть и другие независимые инновации, которые делают процесс разработки интересной и простой. Некоторые примеры таких явлений – XML / VM, PhoneGap и Titanium. Titanium позволяет использовать HTML технологию в программировании для браузера на основе WebKit Android.

Еще один интересный компонентом Android SDK являются услуги на основе местоположения.

В этом разделе SDK предоставляет разработчикам интерфейсов прикладного программирования, приложение, предназначенное для работы с картами, а также для получения информации в режиме реального времени, связанной с расположением устройства.

Компоненты дисплея и компоненты, связанные с телефонией
Интерфейсы Android API предназначены для работы с аудио-, видео - и телефонными компонентами.

До версии 1.5 на Android можно было только записывать аудио, а видео – нет. Версия 1.5 дала возможность записывать аудио и видео. Это было сделано с помощью MediaRecorder. В версиях Android 2.0 и выше применяется Pico движатель для преобразования текста в речь (TTS).

Интерфейс, используемый в работе Android при преобразования текста в речь, очень проста, а также имеет соответствующий код:

```
TextToSpeech mTTS;  
mTTS. speak (sometextString, TextToSpeech. QUEUE_ADD);  
mTTS. setOnUtteranceCompletedListener (this);  
mTTS. stop ();  
mTTS. shutdown ();  
mTTS. synthesizeToFile (...);  
Еще несколько методов, относящихся к этой сфере:  
playSilence
```

setLanguage
setPitch
setSpeechRate
isSpeaking

Важно отметить, что во всех этих концепциях Android, объединенных в единый файл XML, который определяет пакет приложений, представляет собой файл описания (файл манифеста) (AndroidManifest.Xml).

2.7 Разработка программного обеспечения

Начнем с разработки программного обеспечения с графическим интерфейсом. Это связано со спецификой программирования для Android. Для взаимодействия с элементами окна программный модуль подключается к необходимым элементам с помощью идентификаторов. В этом случае элементы поверхности сначала должны быть созданы в главном окне. Как уже упоминалось, графический интерфейс в объявленных XML-файлах и основным языком разработки приложений для Android – это объектно-ориентированный язык программирования Java. В соответствии с определенной структурой при разработке компонентов приложения будет использоваться язык Java.

На рисунке 2.1 представлен вариант использования разработанного мобильного приложения.

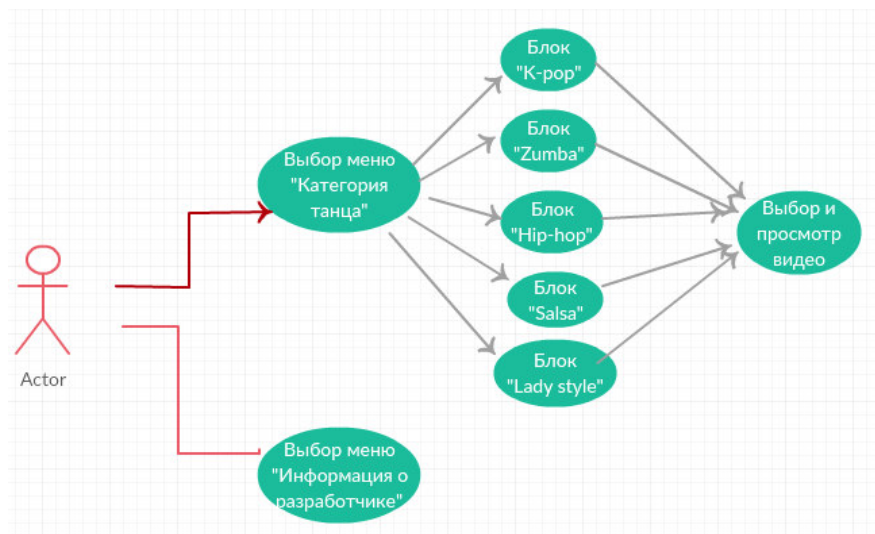


Рисунок 2.1 – Use Case диаграмма (UML)

На рисунке 2.2 представлен внешний вид приложения, запущенного на эмуляторе.

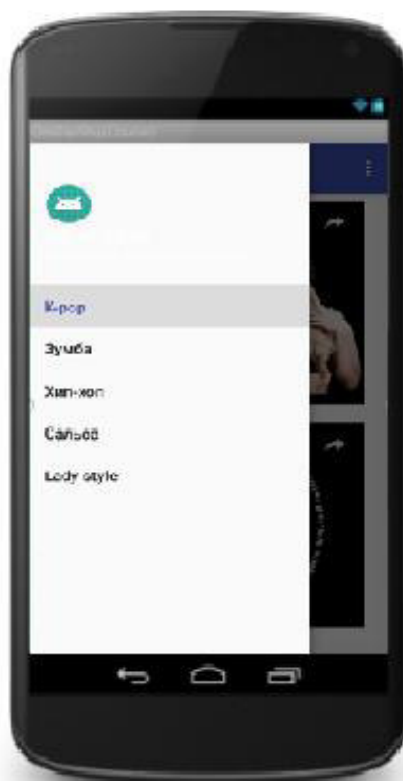


Рисунок 2.2 – Внешний вид мобильного приложения «Dance life»

Разработка мобильных приложений – это длительный и сложный процесс, который можно разделить на две основные стадии:

- создание интерфейса;
- написание функционального кода.

Объект – это прямоугольник в котором указаны подчеркнутые имя объекта и название класса, разделенные двоеточием.

Вниз от каждого объекта тянется пунктирная линия, называемая линией жизни объекта.

Сообщения коорыми обмениваются объекты, изображаются в виде стрелок, направленных от линии жизни одного объекта к линии жизни другого.

Длинные прерывистые полосы на линиях жизни – так обозначают периоды времени, когда объект имеет фокус управления, т.е. выполняет некоторое действие. Диаграмма последовательности использования приложения представлена на рисунке 2.3.

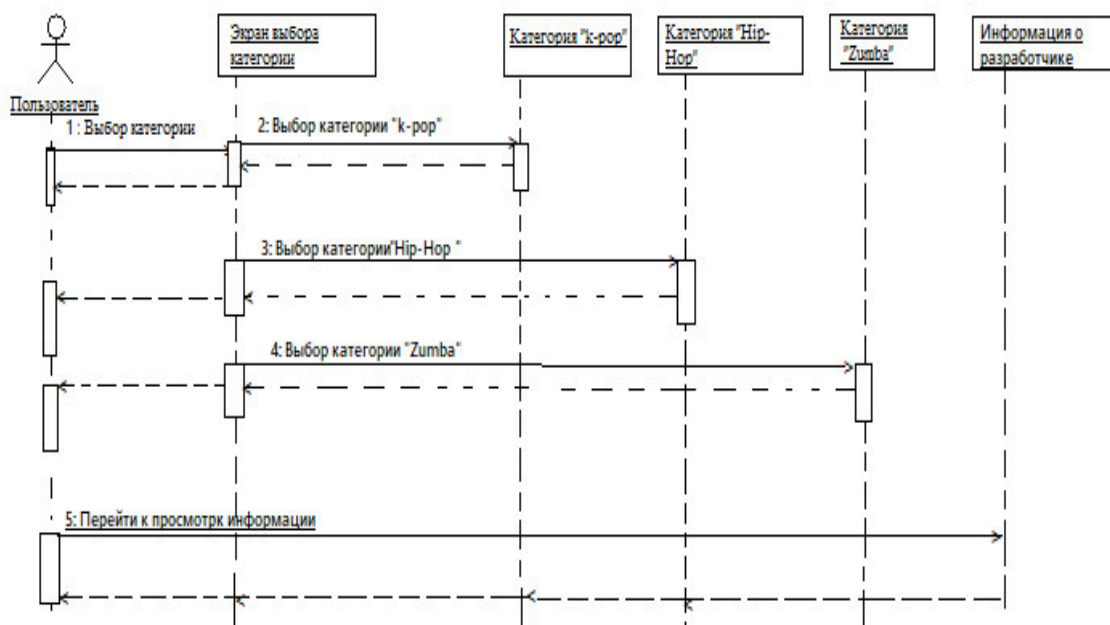


Рисунок 2.3 – Диаграмма последовательности использования приложения

Далее представлена диаграмма деятельности приложения для пользователя (рисунок 2.4). Приведённая диаграмма описывает основные варианты использования разработанного приложения. Из диаграммы видна основная особенность разработанных приложений: пользователь может выбрать только одну категорию и только один видео-урок [9].

Приложение Android разворачивается на устройстве с помощью файла «AndroidManifest.xml». Этот файл содержит необходимую информацию о конфигурации, которая позволит вам правильно установить приложение на ваше устройство. Он также содержит имена классов и типы событий, необходимые для управления приложением, и разрешения, необходимые для его работы. Итак, если приложению требуется доступ к сети, например, для загрузки файла соответствующее разрешение должно быть явно указано в файле манифеста. Это специальное разрешение может иметь много приложений. Эта декларация защиты помогает снизить вероятность повреждения устройства из-за плохо написанного приложения.

Android Software Development Kit (SDK) включает в себя полный набор инструментов для разработки. Он включает в себя отладчик, библиотеки, мобильный эмулятор на основе QEMU, документацию, примеры кода и учебные пособия. В настоящее время поддерживает платформы разработки под управлением операционной системы Linux (любой современный рабочий стол Linux), Mac OS X 10.5.8 или более поздней версии и Windows XP или более поздней версии. В марте 2015 года SDK не будет доступен на Android, но разработка программного обеспечения возможна с помощью специальных приложений Android.

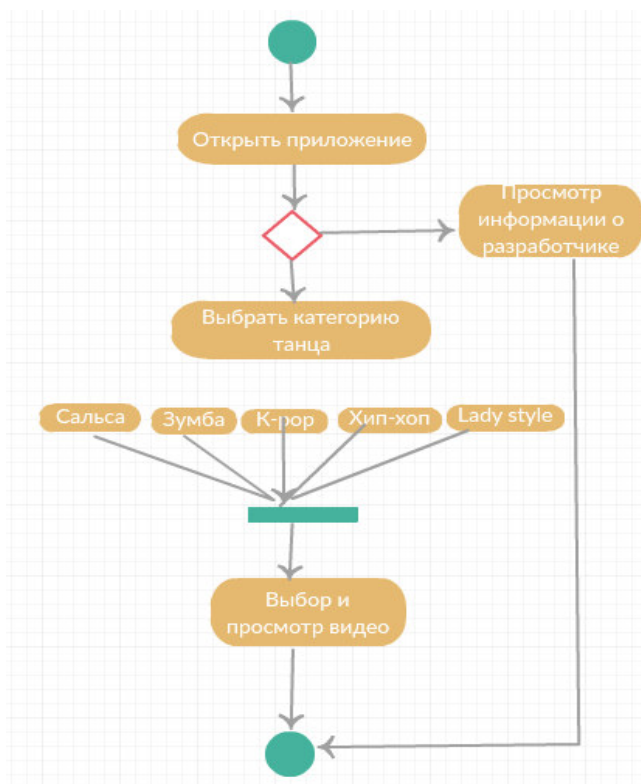


Рисунок 2.4 – Диаграмма деятельности варианта использования приложения

Диаграммы деятельности позволяют моделировать сложный жизненный цикл объекта с переходами из одного состояния (действия) в другое и описывать динамику всех объектов.

При создании проекта среда разработки автоматически генерирует все необходимые файлы и формирует структуру проекта. Основными для работы являются файлы с кодом, которые хранятся в папке «src», и файлы, отвечающие за графическое отображение классов в папке «layout».

В приложении Android пользовательский интерфейс представляет собой один или несколько макетов, которые попеременно отображаются на экране. Layout состоит из серии элементов наследования класса View, организованных в виде древовидной структур [10]. Конкретное наследование может, например, Кнопки, поля ввода текста, области экрана для отображения изображений и т. д. В частности, вы должны выделить наследника класса ViewGroup – это позволяет группировать другие элементы пользовательского интерфейса (например, отображать в виде списка).

Существует два способа создания layout:

- описание дерева элементов в xml-файле;
- программное создание дерева элементов.

При этом layout может являться частью другого layout.

Activity – это компонент приложения, связанный с одним макетом. Отвечает за поведение макета, главным образом описывая, обработчики событий и другую логику, необходимую для выполнения макета. Приложение

состоит из одного или нескольких действий, которые обычно слабо связаны. На рисунке 2.5 показана диаграмма с типовыми Activity.

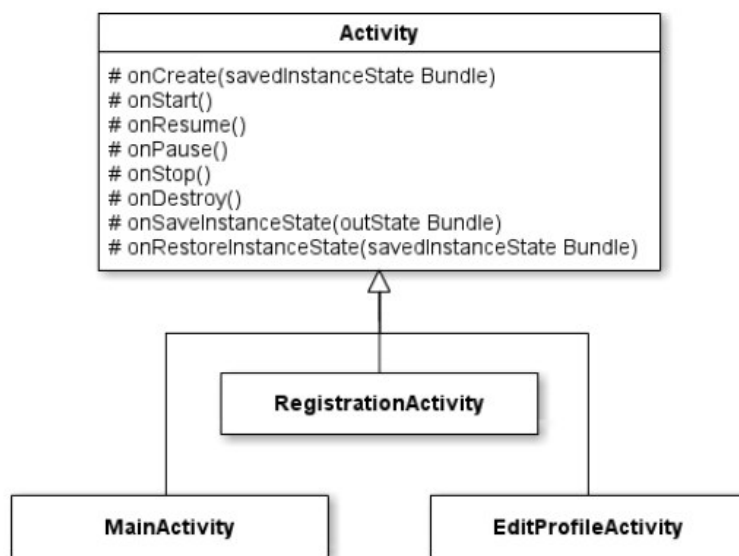


Рисунок 2.5 – Диаграмма с типовыми Activity

Проект Android может состоять из различных модулей. Когда мы создаем проект, создается один модуль – «app» (рисунок 2.6). Модуль имеет три подпапки:

- manifests: хранит файл манифеста «AndroidManifest.xml», который определяет конфигурацию;
- java: хранит файлы кода на языке java, которые структурированы по отдельным пакетам;
- res: содержит используемые в приложении ресурсы.

Отдельный элемент «Gradle Scripts» содержит ряд скриптов «gradle» (как для модуля «app» или других возможных модулей, так и для всего проекта), которые используются при построении приложения.

Собственно, весь код, с которым мы будем работать, располагается внутри этого модуля.

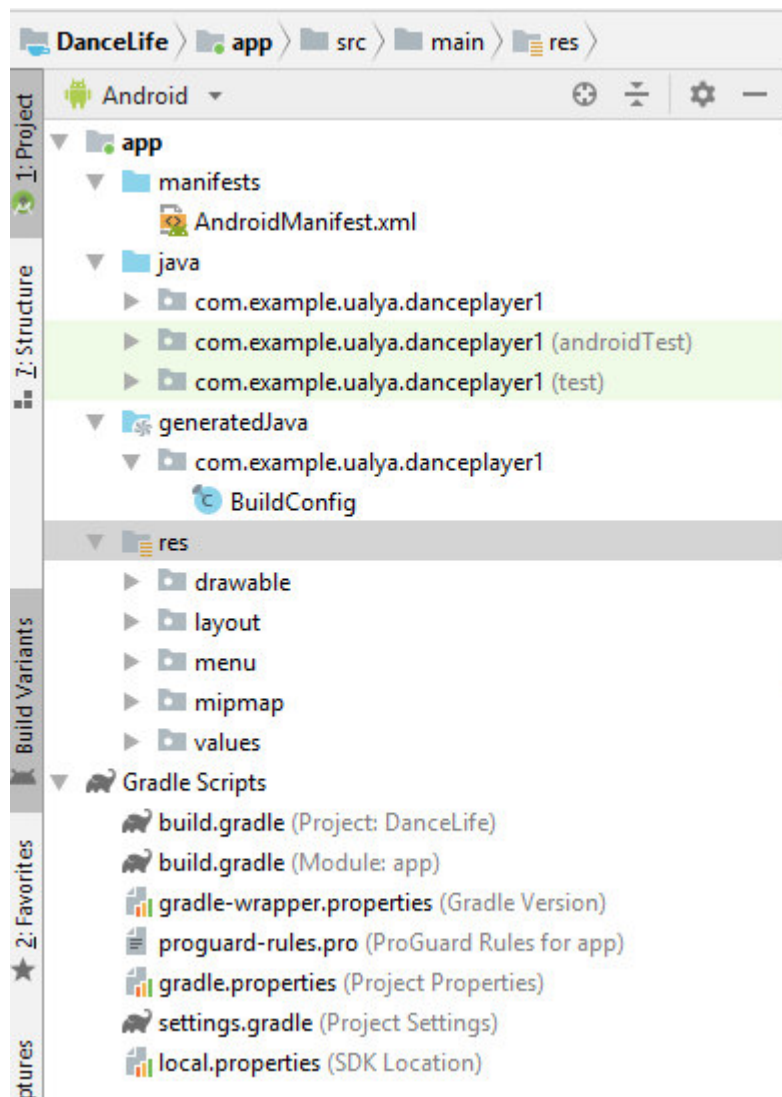


Рисунок 2.6 – Структура проекта «Dance life» в Android Studio

Все модули в проекте описываются файлом «setting.gradle». Он имеет следующее содержимое:

```
include ':app'
```

Файл «build.gradle» содержит информацию, которая используется при построении проекта.

В модуле «app» мы можем увидеть несколько папок и файлов, из которых для нас важными являются:

- каталог «libs» – предназначен для хранения библиотек, используемых приложением;
- каталог «src» – предназначен для хранения исходного кода. Он содержит ряд подкаталогов. Каталоги «androidTest» и «test» предназначены для хранения файлов тестов приложения. А собственно исходные коды располагаются в папке main.

«AndroidManifest.xml» представляет файл манифеста, который описывает фундаментальные характеристики приложения, его конфигурацию и определяет каждый из компонентов данного приложения.

Папка «java» содержит исходные файлы приложения. По нашему проекту в нем находится файл класса «MainActivity», который запускается по умолчанию при старте приложения.

Папка «res» содержит каталоги с ресурсами. В частности, она содержит следующие каталоги:

- папка «drawable» предназначена для хранения изображений, используемых в приложении;

- папка «layout» предназначена для хранения файлов, определяющих графический интерфейс. По умолчанию здесь есть файл activity_main.xml, который определяет интерфейс для единственной в проекте «activity» – «MainActivity»;

- папки «mipmap-xxxx» содержат файлы изображений, которые предназначены для создания иконки приложения при различных разрешениях экрана. И для каждого вида разрешения здесь имеется свой каталог;

- папка «values» хранит различные xml-файлы, содержащие коллекции ресурсов различных данных, которые применяются в приложении.

2.8 Создание интерфейса

Для разработки мобильного интерфейса в Android используется расширяемый язык разметки – Extensible Markup Language – сокращенно XML.

Язык разметки – искусственный язык, который использует набор аннотаций к тексту, предоставляющий инструкции относительно структуры текста или его отражение. Расширяемый язык разметки – предложенный консорциумом World Wide Web (W3C) стандарт построения языков разметки иерархически структурированных данных для обмена между различными приложениями, в частности, через Интернет. Существует, также, упрощенная версия языка разметки – SGML.

В XML документ состоит из текстовых знаков, и пригоден к чтению человеком. Стандарт XML определяет набор базовых лексических и синтаксических правил для построения языка описания информации путем применения простых тегов. Иными словами, предложенный стандарт определяет метаязык, на основе которого путем введения ограничений на структуру и содержание документов определяются специфические, предметно-ориентированные языки разметки данных.

2.9 Описание функциональности

Мобильное приложение “Dance life”, разрабатываемое под операционную систему Android, должно обеспечить пользователям возможность обучаться хореографии.

Самым первым экраном при запуске мобильного приложения появится такой (рисунок 2.7). Это меню выбора категории танца. Здесь нам представлено различные хореографические группы, какой ваша душа пожелает.

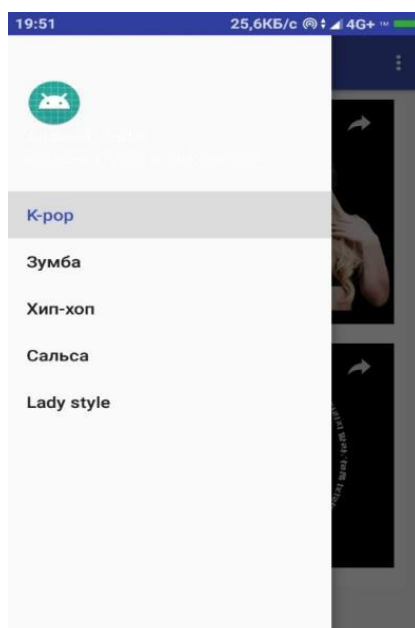


Рисунок 2.7 – Меню выбора категории танца

Если вы выбрали категорию «К-поп», то далее появится экран, в списке которого находятся видео-уроки отсортированные именно по этой категории (рисунок 2.8).

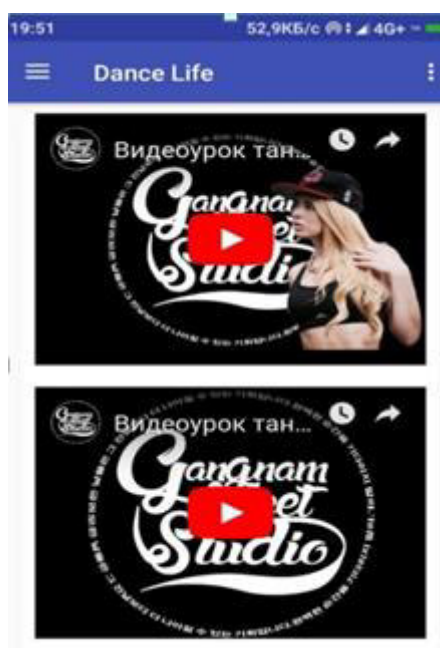


Рисунок 2.8 – Меню видео-уроков из раздела «К-поп»

Далее можно перейти по категории «Зумба», и тогда появится экран, в списке которого находятся видео-уроки отсортированные именно по этой категории (рисунок 2.9 (а, б)).

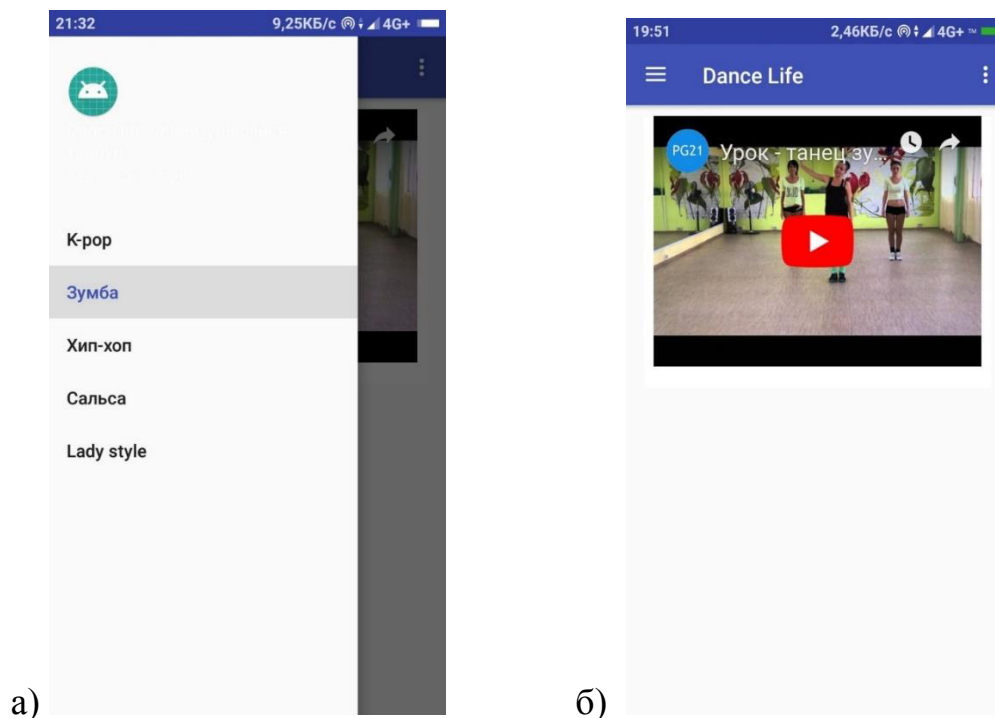


Рисунок 2.9 (а, б) – Меню категории «Зумба»

Далее можно перейти по категории «Хип-хоп», и тогда появится экран, в списке которого находятся видео-уроки отсортированные именно по этой категории (рисунок 2.10 (а, б)).

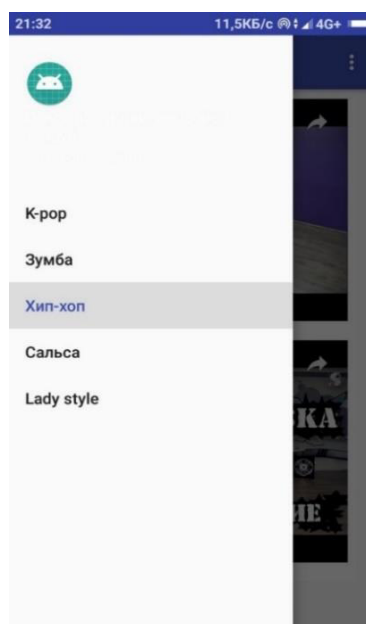


Рисунок 2.10 (а) – Меню категории «Хип-хоп»

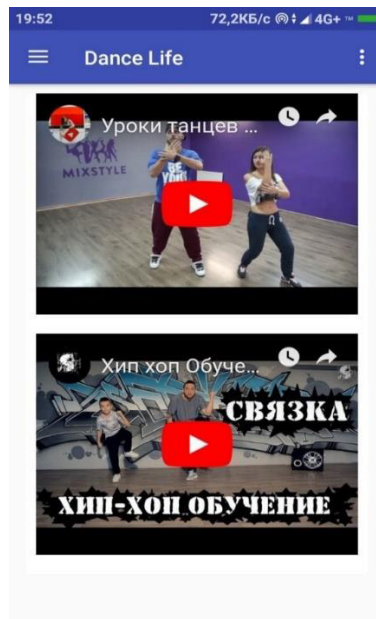


Рисунок 2.10 (б) – Меню категории «Хип-хоп»

Далее можно перейти по категории «Сальса», и тогда появится экран, в списке которого находятся видео-уроки отсортированные именно по этой категории (рисунок 2.11 (а, б)).

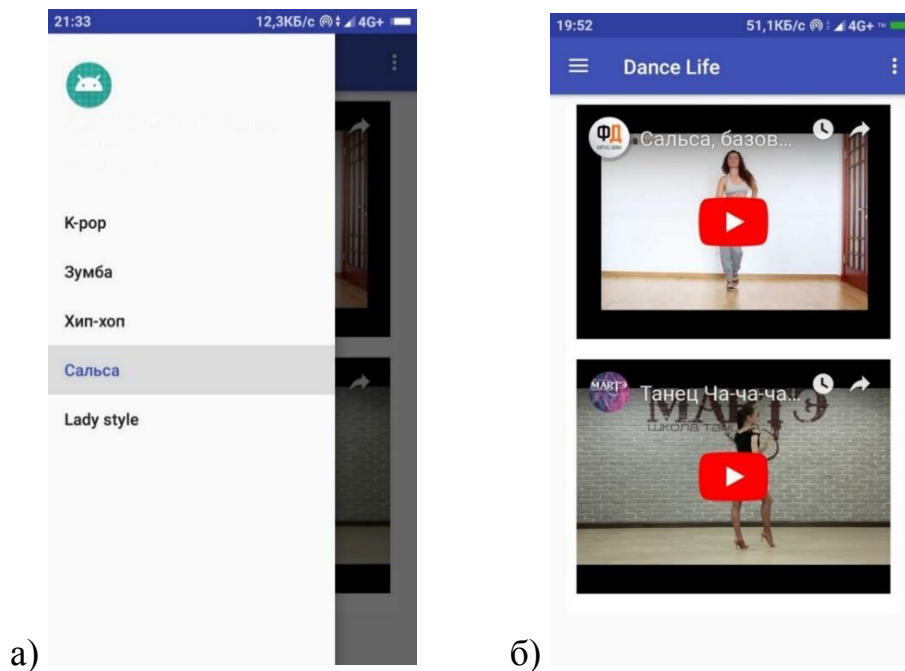


Рисунок 2.11 (а, б) – Меню категории «Сальса»

Далее можно перейти по категории «Lady style», и тогда появится экран, в списке которого находятся видео-уроки отсортированные именно по этой категории (рисунок 2.12 (а, б)).

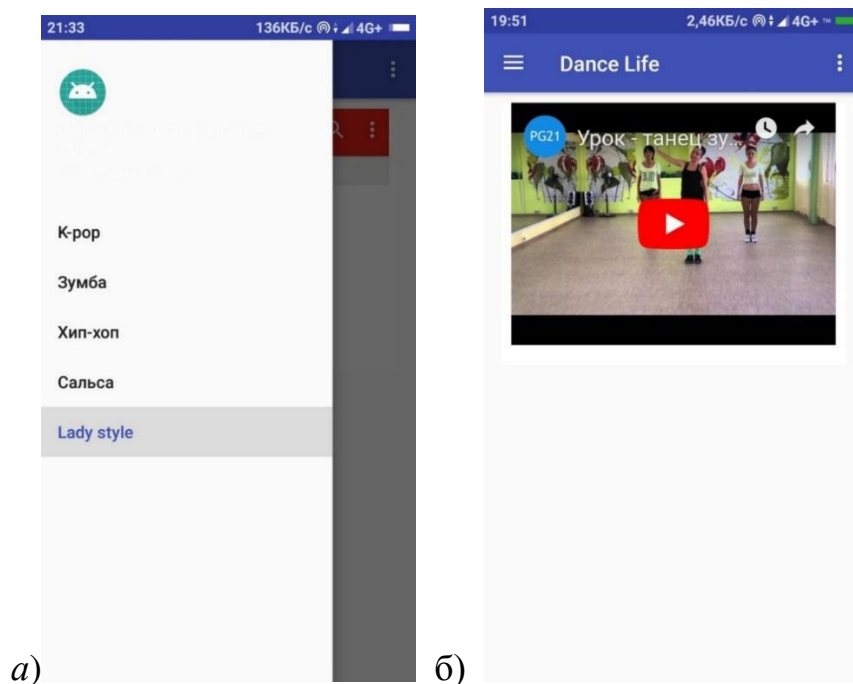


Рисунок 2.12 (а, б) – Меню категории «Lady style»

И самым последним функционирующим экраном данного приложения является меню, где можно просмотреть информацию о разработчике, о программе, о её версии и года выпуска (рисунок 2.13).

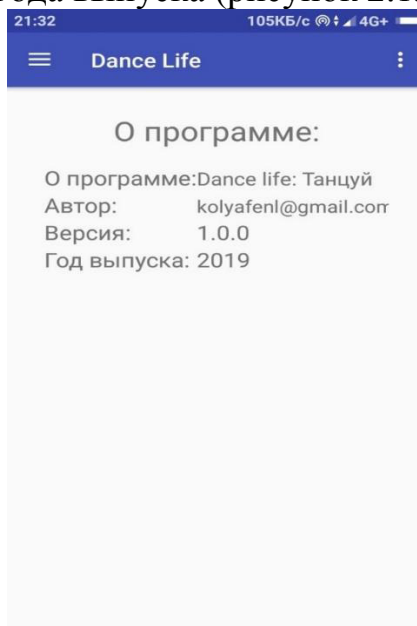


Рисунок 2.13 – Меню «Информация о разработчике»

Пример кода, описывающий разработку меню «Информация о разработчике».

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android=http://schemas.android.com/apk/res/android
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <TextView
        android:id="@+id/textView12"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:paddingBottom="@dimen/activity_vertical_margin"
        android:text="О программе:"
        android:textSize="24sp"/>

    <GridLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="false"
        android:layout_below="@id/textView12">

        <TextView
            android:id="@+id/textView10"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_row="0"
            android:layout_column="0"
            android:text="О программе:"
            android:textSize="18sp" />

        <TextView
            android:id="@+id/textView11"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_row="0"
            android:layout_column="2"
            android:text="Dance life: Танцуй"
            android:textSize="16sp" />
```

В процессе разработки приложения производилось поэтапное тестирование с целью выявления программных ошибок и несоответствий техническому заданию. Для этого нами были созданы эмуляторы смартфона и планшета с разными диагоналями экрана для разных версий Android. Тестируемый программный продукт последовательно запускался на этих эмуляторах, его поведение анализировалось, и при необходимости по результатам анализа вносились изменения в код.

3 Техничко-экономическое обоснование проекта

В данном разделе проводится рассмотрение экономической составляющей реализации проекта.

Себестоимость программного обеспечения включает в себя:

- фонд оплаты труда;
- социальный налог;
- амортизационные отчисления;
- прочие расходы;
- затраты на электроэнергию.

3.1 Трудоемкость разработки ПП

Таблица 3.1 – Распределение работ по этапам и видам и оценка их трудоемкости

Этап разработки ПП	Вид работы на данном этапе	Трудоемкость выполнения
Получение информации о предметной области	Сбор данных о предметной области	20
	Формирование целей и задач проекта	10
	Определение стоимости и сроков разработки ПО	10
Проектирование	Разработка проектных решений по выбору платформы и языка программирования для создания ПО	30
	Назначение требований к интерфейсу приложения	30
Разработка приложения	Построение алгоритма и диаграмм проекта	60
	Построение прототипов системы и ее частей	60
	Написание программы	100
	Доработка программы	20
Тестирование	Проверка программы	20
	Устранение всех недостатков системы и проверка качества	30
Внедрение и поддержка	Установка системы на предприятии	15
	Подготовка документации и руководств по работе с	15

	программой	
Итого трудоемкость выполнения проекта		420

Итого трудоемкость выполнения дипломной работы составит 420 человеко-часов или 53 дней работы или приблизительно 2 месяца работы.

3.2 Расчет затрат на разработку ПП

Определение затрат на разработку ПП производится путем составления соответствующей сметы, которая включает следующие статьи:

- 1) Материальные затраты.
- 2) Затраты на оплату труда.
- 3) Социальный налог.
- 4) Амортизация основных фондов.
- 5) Прочие затраты.

В статью «Материальные затраты» включаются затраты на основные и вспомогательные материалы (бумага, картриджи и другие), энергию, необходимые для разработки ПП [11].

Расчет затрат на материальные ресурсы производится по форме, приведенной в таблице 3.2.

Таблица 3.2 – Затраты на материальные ресурсы

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Бумага	Пачка	1	1200	1200
Картридж	Блок	1	2000	2000
CD-диск	Штука	2	150	300
Итого затраты на материальные ресурсы				3500

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

$$Z_M = \sum_{i=1}^n P_i \times C_i, \quad (3.1)$$

где P_i - расход i -го вида материального ресурса, натуральные единицы;
 C_i - цена за единицу i -го вида материального ресурса, тг;
 i - вид материального ресурса;
 n - количество видов материальных ресурсов.

$$Z_M = (1 * 1200) + (1 * 2000) + (2 * 150) = 3500 \text{ тг}$$

Если для разработки ПП используется электрооборудование, то необходимо рассчитать затраты на электроэнергию по форме, приведенной в таблице 3.3.

Таблица 3.3 – 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	0,5	0,7	345	18,32	1871
Принтер	0,42	0,7	310	18,32	104,8
Мобильный телефон	0,56	0,7	35	18,32	698,2
ИТОГО затраты на электроэнергию					4133,35

$$Z_j = \sum_{i=1}^n M_i \times K_i \times T_i \times C \quad (3.2)$$

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;

C - цена электроэнергии, тг/кВт*ч;

i - вид электрооборудования;

n - количество электрооборудования.

Тариф по городу Алматы для юридических лиц в 2019 году составляет 18,32 тенге за 1 кВт/ч с учетом НДС (согласно данным представленным на официальном сайте ТОО «АлматыЭнергоСбыт»).

$$Z_j = (0,5 * 0,7 * 345 * 18,32) = 2212,14 \text{ тг}$$

$$Z_j = (0,42 * 0,7 * 310 * 18,32) = 1669,68 \text{ тг}$$

$$Z_j = (0,56 * 0,7 * 35 * 18,32) = 251,35 \text{ тг}$$

Общая сумма затрат на электроэнергию:

$$Z_j = 2212,14 + 1669,68 + 251,53 = 4133,35 \text{ тг}$$

В статью «Затраты на оплату труда» включаются расходы по оплате труда всех работников, занятых разработкой ПП (дипломника, руководителей и консультантов дипломной работы, привлеченных лиц).

Затраты на оплату труда рассчитываются по форме, приведенной в таблице 3.4.

Таблица 3.4 – Затраты на оплату труда

Категория работника	Квалификация	Трудоемкость разработки ПП, чел*ч	Часовая ставка, тг/ч	Сумма, тг
Разработчик	Программист	400	850	340000
Руководитель проекта		20	1023	20460
Итого затраты на оплату труда				360460

Общая сумма затрат на оплату труда (Z_{TP}) определяется по формуле:

$$Z_{TP} = \sum_{i=1}^n ЧС_i \times T_i, \quad (3.3)$$

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

T_i - трудоемкость разработки ПП, чел.×ч;

i - категория работника;

n - количество работников, занятых разработкой ПП.

Часовая ставка работника может быть рассчитана по формуле:

$$ЧС_i = \frac{ЗП_i}{ФРВ_i}, \quad (3.4)$$

где $ЗП_i$ - месячная заработная плата i -го работника, тг;

$ФРВ_i$ - месячный фонд рабочего времени i -го работника, час.

Месячный фонд рабочего времени разработчика:

$$ЧС_i = \frac{150000}{176} = 850 \text{ тг}$$

Месячный фонд рабочего времени руководителя проекта:

$$ЧС_i = \frac{180000}{176} = 1023 \text{ тг}$$

Разработчик:

$$З_{тр} = 850 \cdot 400 = 340000 \text{ тг}$$

Руководитель проекта:

$$З_{тр} = 1023 \cdot 20 = 20460 \text{ тг}$$

Общая сумма затрат на оплату труда:

$$З_{тр} = 340000 + 20460 = 360460 \text{ тг}$$

Трудоемкость разработки ПП определяется по данным таблицы 3.1.

В статью «Социальный налог» включается сумма, которая рассчитывается как 9,5% от затрат на оплату труда всех работников ($З_{тр}$), занятых разработкой ПП. При расчете учитываем, что пенсионные отчисления (10% от $З_{тр}$) не облагаются социальным налогом (ставки указаны на 2019 год).

Дополнительная заработная плата:

$$З_{доп} = З_{тр} \cdot 10\% \quad (3.5)$$

$$З_{доп} = 360460 \cdot 0,1 = 36046 \text{ тг}$$

Фонд заработной платы:

$$\Phi_{зп} = З_{тр.о} + З_{доп} \quad (3.6)$$

$$\Phi_{зп} = 360460 + 36046 = 396506 \text{ тг}$$

Расчет социального налога:

$$Н_c = (\Phi_{зп} - ОПВ) \cdot 9,5\% \quad (3.7)$$

где, ОПВ – обязательные пенсионные взносы – 10% от $\Phi_{зп}$.

$$Н_c = (396506 - (396506 \cdot 0.1)) \cdot 0.095 = 33901,26 \text{ тг}$$

В разделе «Амортизация основных фондов» считаем сумму амортизационных отчислений от стоимости оборудования и программного обеспечения (ПО), используемых при разработке ПП. Амортизационные отчисления рассчитываются по форме, приведенной в таблице 3.5.

Таблица 3.5 – Амортизация основных фондов (ОФ)

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Эффективный фонд рабочего времени работы оборудования и ПО, ч/год	Время работы оборудования и ПО для разработки ПП, ч	Сумма, тг
Ноутбук	150000	25	1140	345	11348,6
Принтер	70000	10	1140	310	1903,5
Мобильный телефон	100000	40	1140	35	1228,07
Прогр.обес.	Беспл.				–
ИТОГО амортизация основных фондов					14480

Общая сумма амортизационных отчислений определяется по формуле:

$$Z_{AM} = \sum_{i=1}^n \frac{\Phi_i \times H_{Ai} \times T_{НИРi}}{100 \times T_{ЭФi}}, \quad (3.8)$$

где Φ_i - стоимость i -го ОФ, тг;

H_{Ai} - годовая норма амортизации i -го ОФ, %;

$T_{НИРi}$ - время работы i -го ОФ за весь период разработки ПП, ч;

$T_{ЭФi}$ - эффективный фонд времени работы i -го ОФ за год, ч/год;

i - вид ОФ;

n - количество ОФ.

$$Z_{AM} = \frac{150000 \cdot 25 \cdot 345}{100 \cdot 1140} = 11348,68 \text{ тг}$$

$$Z_{AM} = \frac{70000 \cdot 10 \cdot 310}{100 \cdot 1140} = 1903,5 \text{ тг}$$

$$Z_{AM} = \frac{100000 \cdot 40 \cdot 35}{100 \cdot 1140} = 1228,07 \text{ тг}$$

$$Z_{ам} = 11348,68 + 1903,5 + 1228,07 = 14480,25 \text{ тг}$$

При определении стоимости ОФ необходимо учесть также затраты на доставку и монтаж, установку ПО. Эти затраты могут быть приняты в размере 10-25 % от затрат на приобретение ОФ.

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

$$N_{\text{ф}} = \frac{100}{T_{\text{ж}}}, \quad (3.9)$$

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

$$N_{\text{Ai}} = \frac{100}{4} = 25$$

В статью «Прочие затраты» включаются расходы на арендную плату, включая коммунальные платежи, затраты на лицензирование и сертификацию, расходы на рекламу, канцелярские и прочие хозяйственные расходы (таблица 3.6).

Затраты на арендную плату определяются в зависимости от стоимости аренды 1 кв. м занимаемой площади.

На основании полученных данных по отдельным статьям составляется смета затрат на разработку ПП по форме, приведенной в таблице 3.7.

Таблица 3.7 – Смета затрат на разработку ПП

Статьи затрат	Сумма, тг
1. Материалы	3500
2. Электроэнергия	4133,35
3. Затраты на оплату труда.	360460
4. Отчисления на социальные нужды.	33901,26
5. Амортизация основных фондов.	14480
ИТОГО по смете	382570

Смета затрат на разработку мобильного приложения

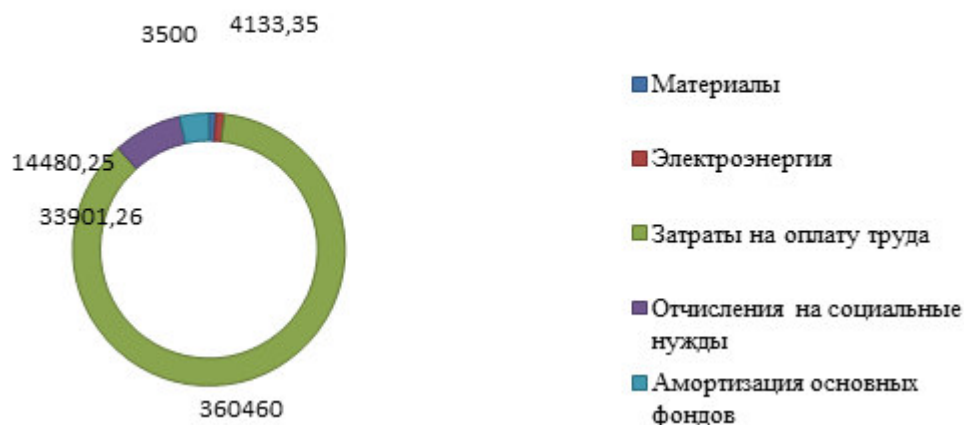


Диаграмма 3.1 – Смета затрат на разработку мобильного приложения

3.3 Определение возможной (договорной) цены ПП

Расчет договорной цены (C_d) :

$$C_d = Z_{НИР} \cdot \left(1 + \frac{P}{100}\right), \quad (3.10)$$

где $Z_{НИР}$ - затраты на разработку ПП (из таблицы 3.7), тг;
 P - средний уровень рентабельности ПП – 20%.

$$C_d = 382570 \cdot (1 + 0.2) = 459090 \text{ тг}$$

На 2019 год ставка НДС установлена в размере 12%.
 Цена реализации с учетом НДС рассчитывается по формуле:

$$C_p = C_d + C_d \cdot \text{НДС}.$$

$$C_p = 459090 + 459090 \cdot 0.12 = 514180 \text{ тг}$$

Далее рассчитаем срок окупаемости проекта результаты расчетов приведены на таблицах 3.8-3.9.

Таблица 3.8 – Расчет окупаемости проекта

Период	Первоначальные затраты	Денежный поток нарастающим итогом
1	382570	50000
2	382570	50000

3	382570	50000
4	382570	50000
5	382570	50000
6	382570	50000
7	382570	50000
8	382570	50000

Таблица 3.9 – Срок окупаемости проекта

Срок окупаемости	8
------------------	---

Рентабельность проекта составит 22 %

3.4 Вывод

В данном разделе произведен анализ разработанного программного продукта с точки зрения актуальности и экономической выгоды его внедрения. Прежде всего, была определена трудоёмкость и сроки выполнения работы по созданию программного продукта. Следующим шагом стал подсчёт затрат, необходимых для реализации и внедрения разработанного мобильного приложения. Завершающим этапом стал расчёт экономического эффекта от внедрения данного программного продукта. Произведённые расчёты и показатели позволяют говорить о целесообразности и экономической эффективности с точки зрения временных затрат от внедрения программного продукта.

Срок разработки данного проекта составил 53 дня или 420 часа.

Себестоимость работ составила – 382570 тенге, в результате расчета, было выявлено, что затраты на разработку и внедрение информационной системы составят 514180. Данный проект окупится приблизительно за 8 месяцев. Рентабельность проекта составит 22 %

4 Охрана труда и БЖД

4.1 Описание условий труда в помещении

Рабочее место – это часть пространства, в котором инженер осуществляет трудовую деятельность, и проводит большую часть рабочего времени, поэтому конструкция моего рабочего места и взаимное расположение всех его элементов соответствует антропометрическим, физическим и психологическим требованиям [12].

Соблюдены следующие основные условия:

- оптимальное размещение оборудования, входящего в состав рабочего места;
- достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения;
- уровень акустического шума не превышает допустимого значения.

В рабочем помещении имеется два рабочих ноутбука, значимого шума от которых, в принципе не имеется. Налажена вентиляционная система в помещении, что является благоприятным фактором для работоспособности.

Единственной проблемой, которая препятствует для повышения уровня труда в помещении с параметрами $L: B: H=5:4:3$, я думаю, является недостаток освещения. Данное помещение имеет одно окно с площадью равной $S=1 \text{ м}^2$ и один светильник типа TLPL228.2x36, в котором применяются две стандартные люминесцентные лампы типа TL-D 58W/54.

Исходя из этих факторов, я решила произвести расчет естественного и искусственного освещения данного офисного помещения.

4.2 Расчет естественного и искусственного освещения в помещении

Условие задачи: Рассчитать необходимую площадь окон для создания нормируемой естественной освещенности в производственном помещении:

- составить схему размещения светильников;
- рассчитать освещенность на рабочем месте точечным методом;
- привести нормативные значения освещенности согласно СНиП РК 2.04-05-2002. Естественное и искусственное освещение. Общие требования;
- произвести реконструкцию системы освещения производственного помещения методом коэффициента использования;
- привести планируемую схему расположения окон светильников;
- сделать выводы и предложения.

Исходные данные:

Тип помещения: Офис

Параметры помещения:

$$L = 5 \text{ м};$$

$$B = 4 \text{ м};$$

$$H = 3 \text{ м};$$

$$h_{ок} = 2 \text{ м};$$

$$S_{ф.о.} = 1 \text{ м}^2$$

Разряд зрительн. работ: IV, б

Коэффициенты отражения:

$$\rho_{\text{пот}} = 70;$$

$$\rho_{\text{стен}} = 50;$$

$$\rho_{\text{пол}} = 30;$$

Высота начала окна $h_{\text{НОК}} = 1 \text{ м};$

Световой пояс: г. Алматы;

$$H_{\text{зд}} = 12;$$

Расстояние до рядом стоящего здания: $P = 6.$

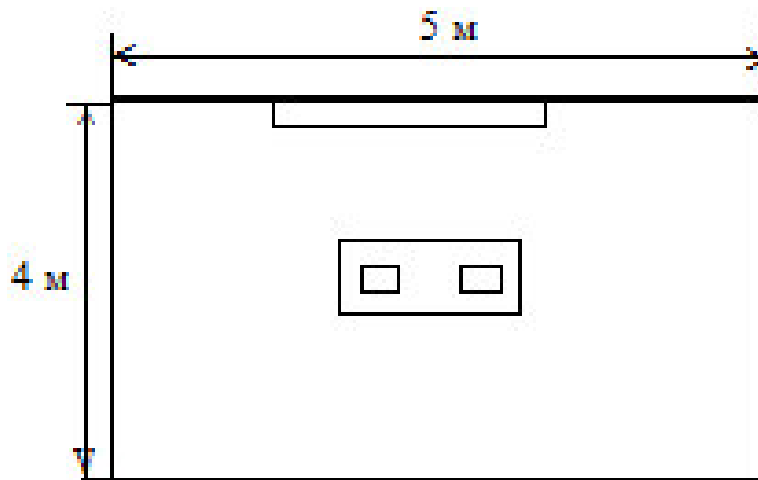


Рисунок 4.1 – Фактическое положение окна и светильника в помещении

4.2.1 Расчет естественного освещения

Расчет осуществляется для установления соответствия условий освещенности на рабочем месте требованиям [13]. Для этого определяем минимальную величину коэффициента естественного освещения e_{\min} по формуле (4.1):

$$e_{\min} = \frac{100 \cdot S_0 \cdot \tau \cdot r}{S_n \cdot \eta \cdot K_{\text{зд}} \cdot K_z} \quad (4.1)$$

Общую площадь окон определяем по формуле (4.2) для бокового освещения:

$$S_0 = \frac{S_n \cdot e_n \cdot \eta_0 \cdot K_{\text{зд}} \cdot K_z}{100 \cdot \tau_0 \cdot r_1}, \quad (4.2)$$

Где S_n – площадь пола помещения, м^2 :

$$S_n = L \cdot B = 5 \cdot 4 = 20 \text{ м}^2$$

e_n – нормированное значение КЕО: $e_n = e_{\text{КЕО}} \cdot m$

e_n – значение КЕО для IV-го разряда б-подразряда зрительной работы:
 $e_{\text{КЕО}} = 0,9$

m – коэффициент светового климата, определяется для ориентации световых проёмов по ресурсам светового климата с ориентацией окон на Ю:
 $m=0.8$

$$e_H = 0,9 * 0.8 = 0,72$$

K_3 – коэффициент запаса по таблице: $K_3 = 1.5$

τ_0 - общий коэффициент светопропускания: $\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4$,

τ_1 - коэффициент светопропускания материала: $\tau_1 = 0.9$

τ_2 - коэффициент, учитывающий потери света в переплетах светопроёма: $\tau_2 = 0,8$

τ_3 - коэффициент, учитывающий потери света в несущих конструкциях: $\tau_3 = 0.9$

τ_4 - коэффициент, учитывающий потери света в солнцезащитных устройствах: $\tau_4 = 0,8$

Тогда

$$\tau_0 = 0.9 * 0.8 * 0.9 * 0,8 = 0.5184$$

η_0 – световая характеристика окон по таблице;

Отношения, необходимые для определения η_0 :

$$\frac{L}{B} = \frac{5}{4} = 1,25$$

$$h_1 = h_{н.ок} + h_{ок} = 2 + 1 = 3 \text{ м,}$$

где h_1 – высота от уровня условной рабочей поверхности до верха окна.

$$\frac{B}{h_1} = \frac{4}{3} = 1.33$$

Таким образом, $\eta_0 = 7.5$

r_1 – коэффициент, учитывающий повышение КЕО при боковом освещении благодаря свету, отраженному от поверхностей помещения и подстилающего слоя, прилегающего к зданию;

$$\frac{P_{\text{пот}} + P_{\text{ст}} + P_{\text{пол}}}{3} = \frac{70 + 50 + 30}{3} = 50\%$$

$$r_1 = 1.1$$

$K_{зд}$ – коэффициент, учитывающий затенение окон противостоящими зданиями по таблице 3.8:

$$\frac{P}{H_{зд}} = \frac{6}{12} = 0.5$$

$$K_{зд} = 1.7$$

Подставим все значения в расчетную формулу:

$$S_0 = \frac{20 * 0,72 * 7.5 * 1.7 * 1.5}{100 * 0.5184 * 1.1} \approx 5 \text{ м}^2.$$

Было предусмотрено одностороннее боковое освещение. Высота оконных проемов составляет 2 м, следовательно, их длина составит 1 м.

Таким образом, фактическая площадь окна оказалась меньше рассчитанной площади окна, что свидетельствует о недостаточности естественного освещения в рабочем помещении.

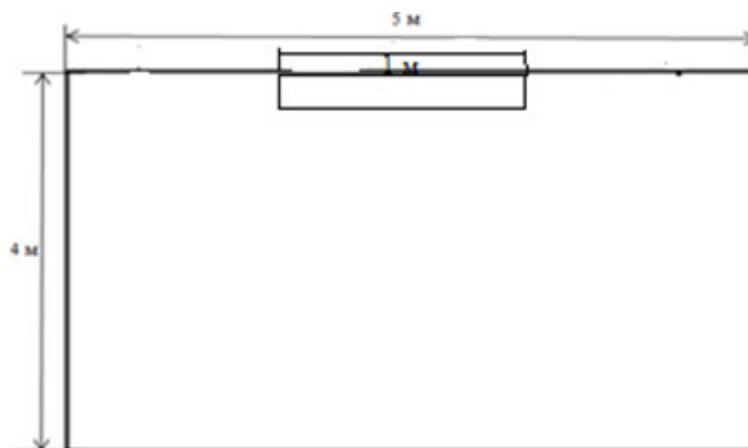


Рисунок 4.2 – Схема помещения при естественном освещении

Теперь находим минимальную величину коэффициента естественного освещения e_{min} , подставив значения в формулу (4.1):

$$e_{min} = \frac{100 * 1 * 0,5184 * 1,1}{20 * 7,5 * 1,7 * 1,5} = 0,14$$

Минимальная величина коэффициента естественного освещения e_{min} получилась меньше нормативного значения КЕО e_n , отсюда, также, можно сделать вывод о недостаточности естественного освещения для данного рабочего места. Далее переходим к расчету искусственного освещения.

4.2.2 Расчет искусственного освещения

Разряд зрительной работы IV (б), поэтому нормируемая освещенность по таблице – 200 лк.

Выбираем светильник типа TLPL228.2x36, в котором применяются две стандартные люминесцентные лампы типа TL-D 58W/54.

Оценка существующей системы искусственного освещения производится путем сравнения фактической освещенности E_{ϕ} производственного помещения с нормативной освещенностью E_n , необходимой для выполнения зрительной работы в данном помещении. Расчет E_{ϕ} производится по формуле 4.3:

$$E_{\phi} = \frac{\Phi * N * n * \eta}{S_n * z * K_3} \quad (4.3)$$

Φ – световой поток одной лампы, $\Phi = 4000$ лм для лампы типа TL-D 58W/54;

N – Количество светильников в помещении;

n – Количество ламп в одном светильнике;

η - коэффициент использования светового потока, % (определяется в зависимости от типов лампы и светильника, коэффициентов отражения стен, потолка и от индекса помещения), для светильника типа TLPL228.2x36 находим $\eta = 0,63$;

S_n – площадь помещения, m^2 ;

$$S_n = L * B = 5 * 4 = 20 \text{ м}^2;$$

z – коэффициент минимальной освещенности, $z = 1,1$ (для люминесцентных ламп $z = 1,1$, для ламп накаливания $z = 1,15$);

K_3 – коэффициент запаса, $K_3 = 1,2$.

Подставив значения в формулу (4.3), получаем,

$$E_{\phi} = \frac{4000 * 1 * 2 * 0,63}{20 * 1,1 * 1,2} = 190,9 \text{ лк}$$

Фактическая освещенность E_{ϕ} производственного помещения получилась меньше нормативной освещенности E_n , поэтому производим реконструкцию помещения, тем самым увеличивая количество светильников в помещении.

Расчет проводим по методу коэффициента. Необходимое количество светильников для создания требуемого нормированного общего освещения в помещении определяется по формуле (4):

$$N = \frac{E_{\min} * S * z * K_3}{\Phi * \eta} \quad (4.4),$$

где $E_{\min} = 200$ лк I, а разряд зрительных работ;

Таким образом, количество светильников равно:

$$N = \frac{200 * 20 * 1,1 * 1,2}{2 * 4000 * 0,63} \approx 2 \text{ шт.}$$

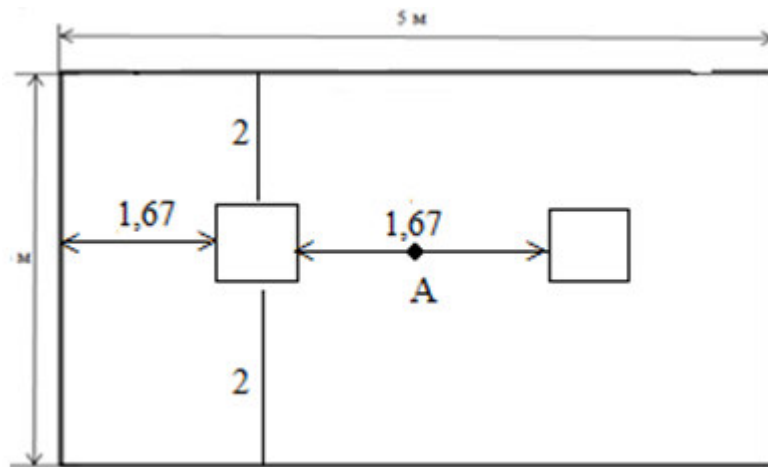


Рисунок 4.3 – Расположение светильников в офисном помещении

Расчет искусственного освещения точечным методом. Размещение светильников в помещении определяется следующими параметрами:

H – высота помещения;

h_c – расстояние светильников от перекрытия (свес);

$h_{рп}$ – высота рабочей поверхности над полом;

Определение расчетной высоты подвеса:

$$h_{расч} = H - (h_{рп} + h_c),$$

$$h_{расч} = 3 - (0,75 + 1) = 1,25 \text{ м}$$

Расстояние между светильниками составит:

$$L_{A,B} = \lambda \cdot h_{расч}, \quad \text{где } \lambda = 1 \div 2 = 0,5 \text{ м}$$

Следовательно,

$$\begin{aligned} L_A &= 2 \text{ м}; & L_B &= 2 \text{ м}; \\ l_A &= 1,67 \text{ м}; & l_B &= 1,67 \text{ м}. \end{aligned}$$

На потолке отмечаем контрольную точку A . Для нее определяем суммарную условную освещенность всех светильников следующим образом:

Находим проекцию расстояния на потолок от точки A до светильника- d_i . Далее определяем угол между потолком и прямой d_i . По этому углу находим условную освещенность. Проверим, выполняется ли условие:

$$E_{г} \geq E_{н}, \quad (4.5)$$

$$\text{где, } E_{\Gamma} = F \cdot \mu \cdot \frac{\sum_{i=1}^m e_{\Gamma_i}}{1000 \cdot K_3}, \quad (4.6)$$

Коэффициент, учитывающий действие равноудаленных светильников:
 $\mu = 1,15$;

$$e_{\Gamma_i} = \frac{I_{\alpha_i} \cos^3(\alpha_i)}{h_{расч}^2}, \quad (4.7)$$

$$\text{где, } \alpha_i = \arctg\left(\frac{d_i}{h}\right),$$

Если внимательно посмотреть на схему расположения светильников, то можно заметить, что расстояние от точки А до светильников является равным.

Лампа 1:

$$d_1 = \sqrt{0,835^2} = 0,835 \text{ м}$$

$$\alpha_1 = \arctg\left(\frac{0,835}{1,25}\right) = 0,589^\circ$$

$$I_{\alpha_1} = 127,2 \text{ кд}$$

$$e_1 = \frac{127,2 * \cos^3(0,835)}{1,25^2} = 30,023 \text{ лк.}$$

$$E_{\Gamma} = \frac{\mu * F_{л}}{1000 * K_3} * \sum E_{\Gamma} = \frac{1,15 * 2 * 4000}{1000 * 1,5} * 40,023 = 207,8 \text{ лк.}$$

Как мы видим, выполняется условие 4.4:

$$E_{\Gamma} > E_{min}.$$

А это значит, что для благоприятного освещения производственного склада потребуется 2 светильника, как и было рассчитано ранее.

4.3 Вывод

Рассмотрев условия труда, я пришла к выводу, что есть недостаток искусственного освещения, из-за которого снижается уровень труда. Сравнив величину коэффициента естественного освещения e_{min} и коэффициента нормативного значения КЕО e_n , мы сделали вывод о достаточности естественного освещения для данного рабочего места. Далее рассчитали

искусственное освещение. Так как фактическая освещенность E_f производственного помещения получилась меньше нормативной освещенности E_n , мы произвели реконструкцию помещения, тем самым увеличивая количество светильников в помещении. Необходимое количество светильников для создания требуемого нормированного общего освещения в помещении равна 2 шт.

Итого, для улучшения труда в офисном помещении необходимо добавить еще один светильник, чтобы было достаточное количество искусственного освещения.

Заключение

Сегодня с развитием сети Интернет большую актуальность получила разработка мобильных приложений.

В течении последних лет показатель, характеризующий уровень спроса на мобильные устройства, постоянно растет.

Такая статистика позволяет сделать вывод о том, что разработка мобильных приложений актуальна и целесообразна.

Главное грамотно оценить, для кого и зачем создается софт. Только полезная разработка получит достойное признание со стороны пользователей.

В данном проекте проведен анализ и обзор существующих инструментальных средств для разработки программного обеспечения для платформы Android, разработано мобильное приложение «Dance life», которое позволяет пользователям обучаться танцам.

Проведенный анализ показал, что платформа Android становится все более популярной как среди разработчиков, так и среди пользователей.

В проекте выполнен обзор официальных средств разработки и средств разработки, представленных сторонними разработчиками. Проведенный анализ показал, что разработка приложений для платформы Android может производиться не только с использованием языка программирования Java, но и с использованием других языков программирования, что делает процесс разработки еще более доступным для большего количества разработчиков. Также в обзоре присутствуют средства разработки, которые ориентированы на начинающих разработчиков или которые могут использоваться в учебных целях. Большинство рассмотренных средств программирования являются либо свободно распространяемыми, либо условно-бесплатными, что тоже повышает популярность платформы Android.

В ходе разработки дипломного проекта были выполнены все поставленные цели и задачи.

Список использованной литературы

1. Статистика ОС [Электронный ресурс] // Google Inc. — Электрон. дан. — [Б.м.], 2017 — URL: <https://marketer.ua/stats-operating-system-2017/> [1].
2. ОС Android [Электронный ресурс] // Google Inc. — Электрон. дан. <http://abrakdabra.ru/enciklopediya/android-os> [2].
3. Архитектура Android [Электронный ресурс] // Google Inc. — Электрон. дан. — [Б.м.], — URL: <https://docplayer.ru/53782926-Diplomnyy-proekt-razrabotka-igrovogo-prilozheniya-pod-operacionnuyu-sistemu-android.html> [3].
4. Классификация мобильных приложений [Электронный ресурс] // Google Inc. — Электрон. дан. — [Б.м.], 2016 — URL: <https://e-koncept.ru/2017/970180.htm> [4].
5. Средства реализации мобильных приложений [Электронный ресурс] // Google In— Электрон. дан.— URL: <http://livetyping.com/ru/blog/na-chem-pishut-prilozheniya-pod-android> [5].
6. Android Studio [Электронный ресурс] // Google Inc. — Электрон. дан.— URL: <https://livetyping.com/ru/blog/na-chem-pishut-prilozheniya-pod-android> [6].
7. Java [Электронный ресурс] // Google Inc. — Электрон. дан.— URL: <http://nado.znate.ru/Java> [7].
8. Android SDK [Электронный ресурс] // Google Inc. — Электрон. дан. — [Б.м.], 2016 — URL: <https://docplayer.ru/53782926-Diplomnyy-proekt-razrabotka-igrovogo-prilozheniya-pod-operacionnuyu-sistemu-android.html> [8].
9. Гамма Э., Приемы объектно-ориентированного проектирования // Э.Гамма, Р.Хелм, Р.Джонсон. — СПб: Питер, 2001. — 368 с. [9]
10. UML диаграммы. [Электронный ресурс] // Google Inc. — Электрон. дан. — [Б.м.], 2016 — URL: [10]
11. Средства разработки приложений [Электронный ресурс] // Google Inc. — Электрон. дан. — [Б.м.], 2016 – URL: <https://livetyping.com/ru/blog/na-chem-pishut-prilozheniya-pod-android>[11]
12. Бекишева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В0703 - Информационные системы – Алматы: АУЭС; 2013. –24 с.[12]
13. С.Е. Мананбаева, А.С. Бегимбетова Охрана труда. Методические указания к выполнению расчетно-графических работ для студентов - бакалавров всех специальностей. - Алматы: АУЭС, 2013 - 17 с.[13]