

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество

«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Кафедра Телекоммуникационных систем

«ДОПУЩЕН К ЗАЩИТЕ»

Зав.кафедрой \_\_\_\_\_

(ученая степень, звание, Ф.И.О.)

« \_\_\_\_\_ »

201 \_\_\_\_\_ г.

(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка музыкального приложения и  
оценка его применимости в различных технологиях  
беспроводной связи

Специальность 5В071900-Радиотехника, электроника и телекоммуникации

Выполнила Мусаев Руслан Ерболович Группа МЭС-12-5

(Ф.И.О.)

Научный руководитель \_\_\_\_\_

(ученая степень, звание, Ф.И.О.)

Консультанты:

по экономической части:

доцент кафедры ЭОиУТ, Боканова Г.И.

(ученая степень, звание, Ф.И.О.)

« 17 » мая

2016 г.

(подпись)

по безопасности жизнедеятельности:

ст. преп. Беримбетова А.С.

(ученая степень, звание, Ф.И.О.)

« 3 » мая

2016 г.

(подпись)

по применению вычислительной техники:

Ж.Т.Н., ст. преподаватель каф. ТКС; Ерениов Н.Ч.

(ученая степень, звание, Ф.И.О.)

« 31 »

мая

2016 г.

(подпись)

(ученая степень, звание, Ф.И.О.)

« \_\_\_\_\_ »

201 \_\_\_\_\_ г.

(подпись)

Нормоконтролер:

Денисова Г.Д.

(ученая степень, звание, Ф.И.О.)

« 6 »

июня

2016 г.

(подпись)

Рецензент:

Ж.Т.Н., доцент Литовский М.А.

(ученая степень, звание, Ф.И.О.)

« 6 »

06

2016 г.

(подпись)

Алматы 2016 г.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Факультет Радиотехники и связи

Кафедра Телекоммуникационных систем

Специальность 5B071900 - Радиотехника, электроника и телекоммуникации

**ЗАДАНИЕ**

на выполнение дипломного проекта

Студенту Мусаев Рустем Ерболович

(Ф.И.О.)

Тема проекта Разработка мультимедийного приложения и оценка его применимости в различных технологиях беспроводной связи

Утверждена приказом по университету № \_\_\_\_\_ от «\_\_» \_\_\_\_\_ 201\_\_ г.

Срок сдачи законченного проекта «\_\_» \_\_\_\_\_ 201\_\_ г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Центральная спектральная отдачи для LTE  $R_{DL}=3,43 \text{ бит/с/Гц}$ ,  $R_{UL}=1,829 \text{ бит/с/Гц}$ ; Ширина канала LTE  $W=10 \text{ МГц}$ ; Число секторов eNB - 3; Размерность кластера  $N_k=3$ ; Число каналов трафика в одном радиоканале  $N_{ch}=1$ ; Средний трафик одного абонента в месяце  $T_f=30 \text{ Гбайт/мес}$ ; Коэффициент для городской местности  $q_f=2$ ; Число ЧНН в день,  $N_{chd}=7$ ; Число дней в месяце  $N_d=30$

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

- 1) Обзор стандарта GSM и LTE
- 2) Возможности становления технологии LTE
- 3) Обзор главных характеристик технологии LTE
- 4) Радиомониторинг сети LTE
- 5) Радиочастотный диалог технологии LTE
- 6) Безопасность и охрана труда
- 7) Расчет показателей экономической эффективности проекта




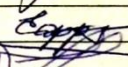
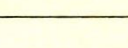
Перечень графического материала (с точным указанием обязательных чертежей):

Итоговая структура стандарта LTE  
 Устройства нового поколения со встроенной поддержкой и примерами их применения  
 Внешний вид эмулятора Android x86 с установленной в нем операционной системой Gingerbread  
 Модель предметной области  
 Диаграмма вариантов использования  
 Диаграмма коопераций для пользователя  
 Структура файлов проекта  
 Зависимость допустимой нагрузки в секторе от числа каналов в графике и вероятности блокировки

Основная рекомендуемая литература:

1. Блог на сайте о разработке под Android на сайте <http://www.habrahabr.ru/blogs/android-development/>.
2. Гельтор А.А. Технологии LTE мобильной передачи данных: учебное пособие. - СПб.: Изд-во Политехн. ун-та 2011. - 188 с.
3. Гольдштейн Б.С., Соколов Н.А., Яновский Г.Г. Сети связи: Учебник для ВУЗов. - СПб.: БХВ - Петербург, 2010. - 225 с.
4. Тихвинский В.О., Терентьев С.В., Юрчук А.Б. Сети мобильной связи LTE: технология и архитектура - М.: Эко-Трендз, 2010. - 176 с.

Консультанты по проекту с указанием относящихся к ним разделов

| Раздел            | Консультант     | Сроки            | Подпись   |
|-------------------|-----------------|------------------|---|
| ВМД               | Великелова А.С. | - 3.05.16        |  |
| Бизнес план       | Боханова Г.И.   | 01.04 - 25.05.16 |  |
| Вспог. техника    | Воронцов И.И.   | 01.03 - 31.05.16 |  |
| Техническая часть | Лазарев И.И.    | 01.03 - 31.05.16 |  |
|                   |                 |                  |   |
|                   |                 |                  |   |

### График подготовки дипломного проекта

[illegible]

Дата выдачи задания «    »    201    г.

Заведующий кафедрой \_\_\_\_\_ (подпись) \_\_\_\_\_ (Ф.И.О.)

Научный руководитель  
Проекта

Задание принял к  
исполнению студент

## **Аңдатпа**

Дипломдық жоба «AUPET Player» тақырыбында Android платформасында мобильдік қосымшасы құрылған, ол музыкалық библиотекаға кіруге өте жақсы құрал болады.

MP3 плеер қосымшасына музыканы жүктеудің екі тәсілі бар. Құрылған қосымша сайттан немесе дербес компьютерден музыканы жүктеуге мүмкіндік береді. Көптеген MP3 плеерлер қосымшаларын компакт-дисктан музыканы MP3 форматына конвертация жасау үшін программалық қамтама арқылы қол жеткізеді.

Сізге тек керегі интернет желісі қосылған ДК, сонымен қатар тиісті техникалық талаптар.

«AUPET Player» қосымшасы жақсы дыбыстың сапасын ұсынады және үлкен көлемдегі әндерді сақтауға мүмкіндік береді.

## **Аннотация**

Дипломный проект на тему «AUPET Player» посвящен разработке мобильного приложения на платформе Android, которое станет отличным инструментом постоянного доступа к музыкальной библиотеке.

Существует два способа загрузки музыки в приложение MP3 плеер. Разработанное приложение позволяет загрузить музыку с персонального компьютера или скачать ее с сайта. Большинство приложений MP3 плееров поставляются с программным обеспечением, которое нужно для конвертации музыки с компакт-дисков в формат MP3. Всё, что вам нужно – подключение к сети интернет и ПК, соответствующий техническим требованиям.

Приложение «AUPET Player» предлагает хорошее качество звука.

Приложение способно хранить большой объем песен, даже базовые модели могут хранить сотни композиций, чем больше объем памяти (обычно измеряется в Гигабайтах), тем больше треков может хранить приложение

## **Abstract**

Diploma project entitled «AUPET Player» dedicated to developing applications for the platform Android, which will be an excellent tool for sustainable access to the music library.

There are two ways to download music in the MP3 player application. The developed application allows you to download music from a PC or download it from the site. Most applications MP3 players come with software that you want to convert music from a CD into MP3 format. All you need - to connect to the Internet and the PC, the appropriate technical requirements.

Appendix «AUPET Player» offers good sound quality. The application can store a large amount of songs, even basic models can store hundreds of songs, the more memory (usually measured in gigabytes), the more tracks can store application.



## Содержание

|  |    |
|--|----|
| Введение   | 8  |
| 1 Ключевые технические свойства технологии GSM и LTE   | 10 |
| 1.1 Обзор стандарта GSM и LTE  | 9  |
| 1.2 Возможности становления технологии LTE   | 12 |
| 1.3 Обзор главных характеристик технологии LTE   | 14 |
| 1.4 Радиоинтерфейс сети LTE  | 16 |
| 1.5 Радиочастотный диапазон технологии LTE   | 17 |
| 1.6 Взаимодействие технологии LTE со стандартом GSM  | 22 |
| 1.7 Описание существующих платформ   | 9  |
| 1.8 Общее описание платформы Android   | 12 |
| 1.9 Google Play  | 17 |
| 1.10 Выбор средств разработки и тестирования   | 22 |
| 2 Проектирование   | 23 |
| 2.1 Язык UML   | 23 |
| 2.2 Rational Rose – лидер среди CASE - средств   | 24 |
| 2.3 Модель предметной области  | 23 |
| 2.4 Диаграмма вариантов использования  | 24 |
| 2.5 Диаграмма взаимодействий   | 28 |
| 3 Разработка мобильного приложения   | 37 |
| 3.1 Цель и рамки проекта   | 37 |
| 3.2 Создание приложения в Android SDK  | 39 |
| 3.3 Реализация приложения  | 41 |
| 3.4 Тестирование приложения  | 44 |
| 4 Расчетная часть  | 51 |
| 4.1 Расчет пропускной способности сети LTE   | 51 |
| 4.2 Расчет пропускной способности сети 3G  | 54 |
| 5 Безопасность и охрана труда  | 67 |
| 5.1 Анализ опасных и вредных производственных факторов при работе с<br>вычислительной техникой | 67 |
| 5.2 Обеспечение электробезопасности  | 68 |
| 5.3 Общие правила пожарной безопасности  | 73 |
| 5.4 Мероприятия по защите объекта при ЧС техногенного и природного<br>характера                | 73 |
| 6 Расчет показателей экономической эффективности проекта                                       | 67 |
| 6.1 Расчет затрат на разработку и внедрения Android-приложения                                 | 67 |
| 6.2 Расчет затрат на комплекс технических средств  | 67 |
| 6.3 Расчет стоимости часа машинного времени  | 68 |
| 6.4 Расчет затрат на разработку технического задания   | 68 |
| 6.5 Расчет затрат на разработку и создание приложения  | 69 |
| 6.6 Расчет затрат на написание и отладку программы   | 70 |
| Заключение   | 81 |

|   |    |
|---|----|
| Перечень сокращений                     | 82 |
| Список литературы                       | 83 |
| Приложение А Справочный материал        | 84 |
| Приложение Б Листинг программы          | 86 |
| Приложение В Пользовательский интерфейс | 87 |

## Введение

На сегодняшний день становление сети телекоммуникаций Республики Казахстан характеризуется большими возможностями. Стратегической задачей развития национальной телекоммуникационной информационной инфраструктуры является обеспечение казахстанского общества средствами и услугами связи высокого качества в необходимом объеме и по доступным ценам. Немаловажным является тот факт, что сегодня множество людей пользуются смартфонами. В современном мире популярность смартфонов растет с каждым днем. Смартфон является умным телефоном, т.е. телефон с широкими возможностями. Основные задачи современного смартфона – дополненная работой с интернет – ресурсами, прослушиванием музыки, фото-видео съемки, использованием игр и приложений. Смартфон - это мобильный телефон, работающий на операционной системе (ОС) открытого типа.

В эти дни на рынке доступны несколько мобильных платформ, стремительную известность среди которых на сегодняшний день приобретает iOS и Android. Виды становления рынка мобильных технологий довольно благоприятны, так как меняется не столько сам рынок, сколько потребитель. Нынешнему поколению владельцев мобильных устройств уже недостаточно просто общаться, им нужен постоянный доступ в глобальную сеть, посредством которой возможно не только находить подходящую информацию, но и находиться в контакте с друзьями, давать информацию о своем местонахождении и многое другое. В данный момент исследование мобильного приложения считается актуальностью.

Android – операционная система для планшетных, мобильных, цифровых, компьютерных проигрывателей, цифровых фоторамок, наручных часов, нетбуков и смартбуков, созданная на ядре Linux. Основным языком для разработки служит Java. Массовое использование платформы Android объясняется открытостью операционной системы. Мобильные устройства под управлением Android работают во всех диапазонах связи, они оснащены WiFi, Bluetooth, GPS, функциональными камерами и качественными дисплеями. Самое главное преимущество ОС Android – огромное количество приложений.

3G (от англ. *third generation* — «третье поколение»), мобильная связь 3 поколения — набор услуг, которые объединяют как высокоскоростной мобильный доступ с услугами сети Интернет, так и технологию радиосвязи, которая создаёт канал передачи данных.

Возможности в сотовых сетях 3G в первую очередь будут ориентированы на мультимедиа возможности сегодняшних телефонов. Например, видео телефония и многое другое. 3G поколение сетей основано на CDMA-сетях, только с более широким каналом, который имеет более высокую пропускную способность.

Переход к сетям 4G позволяет качественно изменить нынешнюю ситуацию в области мобильной связи.



**Целью работы** является то, что нужно разработать музыкальное приложение «AUPET Player» на платформе Android и оценить его применимость в различных технологиях беспроводной связи.

**Для реализации этой цели были поставлены следующие задачи:**

- ознакомление с литературой в этой области;
- разработка музыкального приложения на платформе Android;
- создать проект в языке UML (Unified Modeling Language) в программной среде Rational Rose;
- устроить расчеты и прописать выводы.

В дипломной работе, в котором предполагается разработать музыкальное приложение и оценить его применимость в различных технологиях беспроводных сетей связи. Целью данного дипломного проекта является разработка музыкального приложения на платформе Android.

В дипломной работе будут приняты на вооружение 2 варианта беспроводных сетей: рассчитать пропускную способность для GSM и LTE. Новинка работы заключается в том, что на данный момент количество музыкальных приложений с качественной обработкой звука в Play Market ограничено. Поэтому изучение темы относительно разработки и модернизации данных приложений еще не изучена и может быть применена в дальнейшем при изучении данного вопроса.

## **1. Ключевые технические свойства технологии GSM и LTE**

### **1.1 Обзор стандарта GSM и LTE**

Основным изъяном сетей сотовой взаимосвязи стереотипа GSM на сегодняшний день является низкая скорость передачи данных (предел 9,6 кбит/с). Разработка этого процесса далека от идеала - для передачи данных абоненту дается 1 голосовой канал, а биллинг выполняется отталкиваясь от времени соединения (кроме того по тарифам, слишком мало отличающимся от речевых).

К началу 90-х GSM была передана под управление European Telecommunication Standards Institute (ETSI). Платное введение разработанной технологии началось с середины 1991 г., со временем 1 из видов этого стандарта становилось популярнее и в Северной Америке, а аббревиатуру GSM начали объяснять, как Global System for Mobile. В Европе для восходящего и нисходящего потоков соединения с мобильным терминалом используются спектры частот 890-915 МГц и 935-960 МГц. Доступ выполняется по комбинированной методике TDMA/FDMA.

В частотном диапазоне шириной 25 МГц находится 124 несущих, разделенных промежутками в 200 кГц. К тому же предоставление данных выполняется с разделением по времени. Информация упаковывается в фреймы TDMA, состоящие из 8 составляющих продолжительностью в пределах 0,577 мс, так-называемые кратковременные слоты. Текстура данных, помещенная в масштабах такового слота, именуется (пакет). Следовательно, 1 несущая обслуживает сходу некоторое количество закономерных каналов, любому из их отводится конкретное число слотов. Каналы разделяются на присвоенные, доступ к которым возможно будет иметь исключительно 1 точный мобильный терминал, и совместные, или же правящие, легкодоступные всем приборам в режиме ожидания. Предоставление данных выполняется через Traffic Channel (TCH). Он относится к 1 группе и использует в качестве минимальной структурной единицы мульти фрейм продолжительностью 120 мс, состоящий из 26 фреймов. Для упрощения электроники терминалов мульти фреймы восходящего и нисходящего потоков передаются методично и разделены во времени паузой, равной 3 слотам. Из этого можно сделать вывод невозможность полнодуплексного соединения в сетях GSM. Согласно тех. спецификациям, распознают последующие типы каналов TCH: голосовые - 14,4 Kbps, данных - 9,6 Kbps, 4,8 Kbps и 2,4 Kbps, и еще C BCH – Cell Broadcast Channel. Скорости 14,4 Kbps для данных, получается, достичь исключительно с помощью удаления заголовочной структуры блоков TCH и специальных алгоритмов коррекции ошибок. Так как GSM является цифровой сетью, для передачи не голосовых данных не требуется дополнительного модема.

Само собой, необходимость мобильных пользователей в больших скоростях передачи данных, внезапный подъем пользования мобильного трафика, «нелегкого» контента, выход в свет и бурный подъем продаж

современных видов мобильных приспособлений (телефонов новейшей разработки, планшетных ПК и так далее) будут способствовать операторам к введению LTE-A как наиболее совершенной технологии универсального доступа в среднесрочной возможности. Возможно, навряд ли стоит ждать быстрого введения LTE-A, так как операторы сетей LTE на сегодня наиболее заинтересованы в быстром возврате сделанных инвестиций.

Исследование технологии LTE как эталона официально стартовала в конце 2004 года. Перед исследователями встал вопрос о выборе технологии физического значения [2], которая бы дала высшую скорость передачи данных. Были предложены 2 варианта: W-CDMA, уже применяемую в сетях HSPA, и OFDM – новенькая разработка радио интерфейса. После проделанных работ было решено применять технологию OFDM (Orthogonal frequency – division multiplexing) – мультиплексирование с ортогональным частотным разделением каналов.

В начале мая 2006 года в масштабах плана 3GPP была сделана 1-ая спецификация на радио интерфейс E-UTRA (Evolved UMTS Terrestrial Radio Access) [3]. Эта спецификация вошла в базу 3GPP Release 7. В декабре 2008 года была подтверждена версия стереотипов 3GPP Release 8, которая отмечала строительные и высоко функциональные притязания к системам LTE. Посреди 2009 года были замечены 1-ые искусные системы на базе LTE. В конце 2009 года шведская телекоммуникационная фирма TeliaSonera, в сочетании с Ericsson объявила о запуске первой в мире коммерческой сети в Стокгольме и Осло.

На сегодня сети стандарта LTE работают в наиболее чем 80 государствах мира и их количество быстро повышается.

В Казахстане возведение сетей стереотипа LTE остановлено проблемами в распределении частотного ресурса компаниям-операторам мобильной взаимосвязи. 25 декабря 2012 г. фирма «Алтел» запустила первую в Казахстане сеть LTE в городе Алматы.

## **1.2 Возможности становления технологии LTE**

Сравнительно услуг, то здесь разработчики стереотипа помимо обычной телефонной взаимосвязи пользователю GSM предоставляются разнообразные предложения передачи данных. Абоненты GSM обладают возможностями воплощать в жизнь обмен данными с абонентами ISDN, повседневных телефонных сетей, сетей с коммутацией пакетов и сетей взаимосвязи с коммутацией каналов, используя различные методы и протоколы доступа, к примеру X.25. Вероятна передача факсимильных известий, реализуемых при применении надлежащего адаптера для факс-аппарата. HGSN, которой не было



в старых аналоговых системах, считается двунаправленным предоставление кратких известий SMS (Short Message Service) - до 160 б, передаваемых в режиме с переходным сбережением данных.

В «цифре» получилось воплотить добавочные полномочия, которые неприступны в аналоговых стереотипах предшествующего поколения. Как правило данное относится к качеству звучания голоса собеседника (качество передачи и кодировки речи), аутентификации абонента и автоматическому роумингу. Кроме всего этого, это:

- Использование SIM-карты для обеспечения доступа к каналу и предложениям взаимосвязи;
- шифрование передаваемых известий;
- замкнутый от прослушивания радио интерфейс;
- аутентификация абонента и идентификация абонентского оборудования по криптографическим методам;
- внедрение служб коротких известий, передаваемых по каналам сигнализации;
- автоматический роуминг абонентов разных сетей GSM в государственном и международном масштабе;
- межсетевой роуминг абонентов GSM с абонентами сетей стереотипов DCS1800, PCS1900, DECT, также со спутниковой системой индивидуальной радиосвязи Globalstar.

Стереотип LTE представляет из себя владеющий большой эластичностью эфирный интерфейс [4]. Вид сети носит название E-UTRAN – Evolved Universal Terrestrial Radio Access Network (развивающаяся многоцелевая наземная сеть радиодоступа). Далее приведены главные характеристики технологии LTE.

1. Технология разностороннего доступа:
  - прямой канал (Downlink – DL) –OFDMA;
  - обратный канал (Uplink – UL) –SC-FDMA;
2. Рабочий диапазон частот: 450 МГц; 700 МГц; 800 МГц; 1800 ; 2,1 ГГц; 2,4 - 2,5 ГГц; 2,6 - 2,7 ГГц;
3. Битовая скорость:
  - непосредственный канал (DL) MIMO 2TX×2RX: 100 - 300 Мбит/с;
  - обратный канал (UL): 50 - 172,8 Мбит/с.
4. Ширина полосы радиоканала: 1,4 - 20 МГц.
5. Радиус ячейки: 5 – 30 км.
6. Емкость ячейки (число обслуживаемых абонентов):
  - наиболее 200 пользователей при полосе 5 МГц;
  - наиболее четырехсот пользователей при полосе более 5 МГц.
7. Мобильность: скорость движения до 250 км/ч.
8. Параметры MIMO:
  - прямой канал (DL): 2TX×2RX, 4TX×4RX;
  - обратный канал (UL): 2TX×2RX.

9. Значение задержки (latency): 5мс.
10. Спектральная эффективность: 5 бит/сек/Гц.
11. Поддерживаемые типы модуляции:
  - прямой канал (DL): 64 QAM, QPSK, 16QAM;
  - обратный канал (UL): QPSK, 16QAM.
12. Дуплексное разделение каналов: FDD, TDD.

### 1.3 Обзор главных характеристик технологии LTE

Структура сети LTE изобретена так, чтобы гарантировать поддержку пакетного трафика с «бесшовной» мобильностью, наименьшими задержками доставки пакетов и максимальными признаками свойства профилактики. Основной целью создателей стереотипа LTE были наиболее вероятное облегчение текстуры сети и исключение дублирующих функций сетевых протоколов, отличительных для системы 3GUMTS.

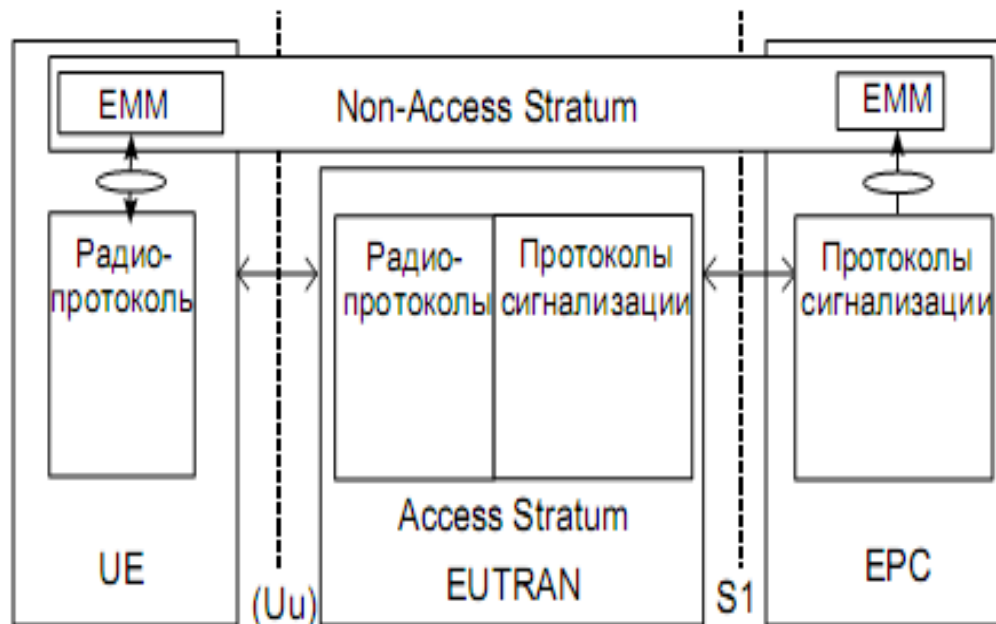


Рисунок 1.1- Итоговая структура стандарта LTE

В архитектуре стандарта LTE все сетевое взаимодействие происходит между 2-мя узлами: базисной станцией (eNB) и блоком управления мобильностью (MME), что включает сетевой шлюз GW (Gateway).

На физическом уровне сеть LTE включает в себя 2 компонента: сети радиодоступа E-UTRAN и базисной сети SAE (System Architecture Evolution).

номера абонентским терминалам, обеспечивает безопасность сети, проводит проверку подлинность известий абонентов и правит роумингом.

Закономерная часть UPE (User Plane Entity) отвечает за передачу этих юзеров согласно протоколам плоскости юзера U-plane. Составляющую UPE выполняет следующие функции: сжатие заголовков IP- протоколов, шифрование потоков этих, терминацию пакетов этих.

Зодчество базисной сети SAE представляет в себе пакетный PS-домен системы LTE, который дает как голосовые, но и всю совокупность IP-услуг на базе технологий пакетной коммутации этих. В базу базисной сети SAE положена концепция «все через IP» и то событие, собственно доступ к ней обладает возможность исполняться как через сети радиодоступа 2 и третьего поколений (UTRAN/GERAN), но и через сети не-3GPP (WiMAX, Wi-Fi), а аналогично через сети, использующие проводные IP-технологии (ADSL , FTTH).

#### **1.4 Радиointерфейс сети LTE**

Радиointерфейс сети LTEE-UTRAN поддерживает два способа дуплексного разнесения каналов: частотный FDD и временной TDD [5]. Работоспособность сетей LTE имеет возможность исполняться в частотных интервалах с разной шириной. Сигналы нисходящего и восходящего направлений могут занимать полосы от 1,4 до 20 МГц исходя из численности энергичных ресурсных блоков. Предоставление информация в восходящем и нисходящем инструкциях организована в кадрах, функционирование которых одинакова 10 мс. Кадры делятся на более мелкие временные структуры – слоты.

В режиме с частотным разнесением FDD кадр делятся на 20 слотов, нумеруемые ни от какого до 19-го, любой из которых имеет продолжительность 0,5 мс. В режиме FDD временной ресурс около кадра разделен пополам для передачи в других инструкциях. Физические каналы в режиме FDD в других инструкциях имеют непрерывный дуплексный разнос. Режим временного разнесения каналов TDD включает в себя асинхронную природу. Предоставление этих в режиме TDD случается незамедлительно в двух направлениях в одном диапазоне частот.

Отличием радиointерфейса в полосы «вниз» сети E-UTRAN считается внедрение технологии многочисленного доступа OFDMA – мультиплексирование с ортогональным частотным разделением. Главным из ключевых целей применения технологии OFDMA является борьба с помехами, стимулированных многолучевым распространением сигнала, ибо OFDM-сигнал рассматривается как большое количество медленно модулируемых узкополосных сигналов, а не как 1 живо модулируемый широкополосный сигнал. Внедрение OFDM базируется на формировании многочастотного сигнала, состоящего из тысячи подносящих частот. При создании OFDM-сигнала поток поочередных информационных знаков продолжительностью  $T_i/N$  разбивается на блоки, имеющие  $N$  знаков;  $T_i$  –

продолжительность 1-го знака. Блок поочередных информационных знаков меняется в блок параллельных знаков, в каком любой информационный знак подходит конкретной частоте многочастотного сигнала.

LTE носит характер закономерного продолжения развития технологий



UMTS и CDMA, направленного на удовлетворение возрастающего спроса в скорости передачи информации в мобильной сети. В концепции скорость соединения в LTE сети может достигать 326,3 Мбит/сек в сторону абонентов и 172,7 Мбит/с в сторону базисных станций. На сегодня LTE может быть беспроводной сетью 4-го поколения с коммутацией массива данных.

В технологии LTE наиболее хорошо отражен частотный диапазон, различающийся минимальными значениями задержки при передаче и увеличенной емкостью. Благодарность данному существует значительное увеличение скорости передачи информации и прирост свойства предоставляемого сервиса. К примеру, в больших населенных пунктах Казахстана LTE очень востребовано.

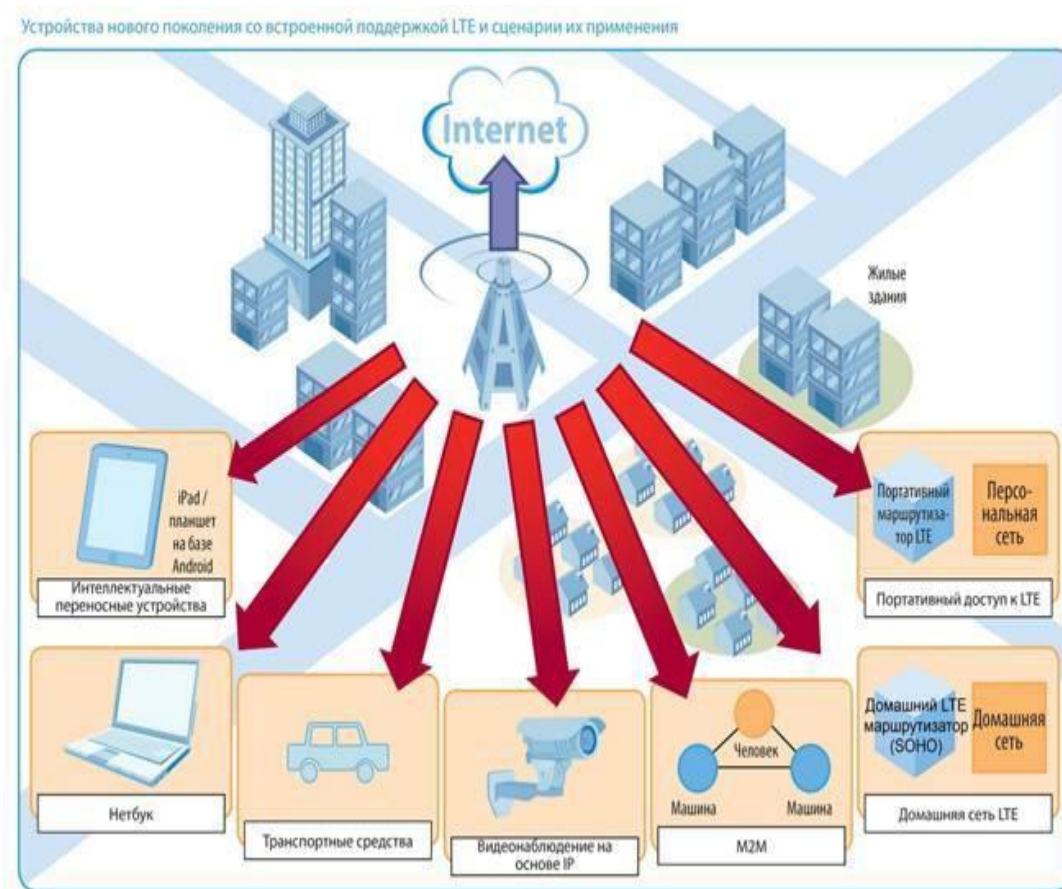


Рисунок 1.2 - Устройства нового поколения со встроенной поддержкой и сценарии их применения

Каковы последствия отдачи беспроводного доступа современной разработки?

На сегодня в роли единственного варианта с целью замены проводного доступа в крупную сеть готов стать LTE. Безусловно, ключевым фактором, определяющим интересную сделку некоторой предложения, считается ценовая политика. Не так давно в прошлом медленный и дорогой беспроводной сеть интернет считался роскошью не для любого юзера

мобильной взаимосвязи, и из-за скудного выбора абоненту приходилось выкладывать немного 10-ов тенге за Мб GPRS этих. На сегодняшний день технологии LTE свидетелем насладиться стремительным беспроводным доступом в отсутствие убытка денежному бюджету и собственным желаниям. В соответствии с этим GSA (Global Mobile Suppliers Association), на начало IV квартала 2012 г. выше 350 операторов вложились в сети LTE (на 41% больше, нежели в 2011 г.). В 48 государствах 105 операторов запустили платную использование сети LTE, даже 10 сеток в не парном диапазоне (TDD). Согласно мониторингу GSA, до конца 2012 г. число коммерческих сеток LTE достигло 159 в 68 государствах, а к концу 2013 г. их стало уже 195 в 72 государствах. При всем при этом специалисты фирмы Strategy Analytics считают, что количество LTE-подключений в текущем году увеличится в 9 раз и достигнет 90 млн, а кто-то говорит и о 150 млн. Согласитесь, что это огромное количество. Драйвером же подъема абонентской базы будут LTE-смартфоны, которые начинают энергично выходить за границы рынков Соединенных Штатов, Южной Кореи и Стране восходящего солнца. В следствии с этим отраслевые предвестники полагают, что к 2017 г. на планете будет выше 1 миллиардов LTE-подключений. К тому же, в фирме AT&T ожидают, что к концу 2014 г. во всем мире LTE-сетями будут пользоваться уже около 300 млн человек.

А настоящий уровень проникновения сотовой взаимосвязи во всем мире к концу 2012 г. составил только лишь 45%, вместе с тем число интенсивных SIM- карт приблизится к 6 миллиардов, собственно подходит вторжению в 85%, рассказывается в исследовании Ассоциации GSM (GSMA)

### **1.5 Радиочастотный диапазон технологии LTE**

Рабочими группами Партнерского плана 3GPP и ETSI в технических спецификациях для LTE определены 17 полос радиочастот для режима частотного дуплекса FDD и 8 полос для режима временного дуплекса TDD, которые показаны в таблице 1 приложения А.

Из таблицы заметно, что спектры, сделанные для становления сеток LTE, теснее освоены либо осваиваются в Казахстане для работы сеток мобильной взаимосвязи и беспроводного доступа различных технологий. Благодаря тому, творение в Казахстане LTE-сетей сопрягается с проблемами с выбором и получением разрешения на применение частотного спектра. Таким образом, грядущее внедрение сетей LTE в Казахстане объединено с необходимостью изменения применения радиочастотного диапазона на базе государственных операций его освобождения и перепланирования.

В Казахстане распределены полосы частот для многообещающих радио технологий, включая LTE, так:

- 1) LTE FDD в рабочем спектре частот band3:

- полоса частот передачи в направлении «к абоненту» — 1860...1880
  - полоса частот передачи в направлении «от абонента» — 1765...1785
- Сопоставимость со стереотипом 3GPP Release9;

2) LTE (3GPPEUTRA) в рабочем диапазоне частот Band20:

полоса частот передачи в направлении (к абоненту) (downlink) 832-862

полоса частот передачи в направлении «от абонента» (uplink) 791-821

Рабочая частота сетей 4GLTE в Казахстане: 800-1900 МГц.

## 1.6 Взаимодействие стандарта LTE со стандартом GSM

Поддержка мобильности абонентского терминала при его движении из зоны профилактики одной сети в зону профилактики иной – считается весомой задачей, образующейся при содействии сети LTE с сетями мобильной взаимосвязи стереотипов 3GPP (UMTS/GSM/HSPA). Взаимодействие сети LTE с сетями 3GPP содержится в обеспечивание дискретной мобильности (роуминга) и обеспечивания постоянной мобильной взаимосвязи (хэндовера). Главными интерфейсами взаимодействия сети LTE с сетями 3GPP считаются интерфейсы S3, S4 и S12. Эти интерфейсы гарантируют взаимодействие закономерного составляющего управления мобильностью MME и шлюза S- GW сети LTE с гарантийным узлом SGSN сетей 3G при помощи туннельного протокола GTP (GPRS Tunnelling Protocol). Протокол GTP уготован для передачи этих плоскости управления (протокол GTP-C) и для передачи этих плоскости пользователя (протокол GTP-U). В критериях роуминга шлюз S- GW визитной сети взаимодействует со шлюзом.

P-GW (шлюз взаимодействия с пакетными сетями) бытовой сети.

Взаимодействие сети LTE с иными 3GPP для предложения классических услуг телефонии исполняется при помощи как классической технологии коммутации каналов (TDM), но и технологии коммутации пакетов на базе сервисной подсистемы IMS.

Хэндовер между сетью LTE и иной сетью 3GPP при претворении в жизнь голосового вызова случается при помощи взаимодействия закономерного составляющего MME с сервером MSC по интерфейсу Sv в случае вызовов из сети LTE в обычный домен коммутации каналов (CS-домен); и при помощи взаимодействия закономерного составляющего MME с узлом SGSN по интерфейсу S3 в случае голосового вызова из сети LTE в домен коммутации пакетов (PS-домен).

Взаимодействие сети LTE с сетями не-3GPP делится на взаимодействие с сетями с гарантированной защищенностью – «верными» и взаимодействие с сетями с негарантированной сохранностью – «не внушающими доверие». В виде «верных» сетей имеют все шансы выступать присоединенные сети иных стереотипов (cdma 2000, WiMAX), в виде «не внушающих доверие» -



общественные IP-сети Веба. Взаимодействие сети LTE с «верными» сетями стереотипов не-3GPP выполняется при помощи шлюза P-GW, взаимодействие с «не внушающими доверие» сетями – посредством шлюза ePDG.

С учетом концепции возведения базисной сети EPC «все через IP» мобильность абонентского терминала при содействии сети LTE с сетями не-3GPP базирована на протоколах управления мобильностью в IP-сетях:

- протоколы управления мобильностью на базе хостов - HBM (Host Based Mobility) – MIPv4, DSMIPv6;
- протоколы управления мобильностью на базе сети – NBM (Network Based Mobility) – PMIPv6.

Идентификация абонентского терминала по IP-адресу и маршрутизация осуществляется так же как в IP-сетях.

Главными различиями технологии LTE от предыдущих технологий мобильной связи 2G и 3G считаются:

- организация связи, как голосовой, но и передачи этих по IP-протоколу;
- высочайшие скорости передачи этих;
- упрощенное зодчество сети.

### **1.7 Описание существующих платформ**

Наличие операционной системы (ОС) - основная особенность, отличающая смартфон от обычного мобильного телефона. При выборе конкретного варианта смартфона или коммуникатора ОС часто становится определяющим фактором.

Наиболее популярные ОС и платформы для смартфонов:

Android - платформа для смартфонов, с открытым исходным кодом, разрабатываемая ОНА (группа компаний во главе с Google). Платформа разработана на Linux.- большой популярности не получили, но тем не менее является перспективным направлением. Смартфоны на базе Linux пользуются спросом в основном в Азии - современная мобильная платформа, созданная компанией Samsung.

### **1.8 Общее описание платформы Android**

Android – это самая популярная на сегодня операционная система для мобильных устройств. По данным на август 2013, ОС Android установлена на 79% устройств мобильного рынка [2].

Платформу Android поддерживают многие разработчики мобильных устройств, например Acer, HTC Corporation, Samsung, Sony Corporation и другие. Для производителей, не поддерживающую эту платформу, существуют сборки Android, предназначенные для установки на их устройства.

В состав платформы Android входят операционная система на базе ядра Linux и среда разработки с открытыми исходными кодами [3].

### 1.8.1 История Android

Платформа Android считается частью группы Open Handset Alliance, поставившей себе цель создать более совершенный мобильный телефон. Данная группа во главе с Google включает операторов мобильных сетей, производителей мобильных устройств и компонентов, разработчиков программных решений и поставщиков услуг, а также маркетинговые компании. В какой то мере разработки программного обеспечения Android находится в самом центре мира открытого ПО.

Датой рождения мобильной платформы Android можем считать 23 сентября 2008 года. Именно в тот день вышла первая версия ОС – Android 1.0 с кодовым именем Apple Pie.

Компания Google только за первый год существования своей мобильной платформы обновила ее трижды. Версии Android 1.1 Banana Bread, 1.5 Cupcake и 1.6 Donut принесли массу нововведений: запись и воспроизведение видео, анимацию переключения окон, многоязычный голосовой поиск, поддержку разрешения экрана 800x480 точек и др.

Приемлемый «товарный» вид платформа Android получила с обновлением до версии 2.0, а после до 2.1 с одинаковым кодовым именем Eclair. Можно было использовать несколько аккаунтов Google, а стандартный веб-браузер получил поддержку HTML5.

В 2010 году Google вышла версия Android 2.2 Froyo, в которой увеличилось скорость работы приложений, включающих в себя JIT-компиляцию, и была создана поддержка Adobe Flash. Обновление до Froyo получили все вышеупомянутые смартфоны с 1 ГГц процессором.

В конце 2010 года разработали версию Android 2.3 названным Gingerbread, которая до весны 2013 года считалась самой популярной среди пользователей. Она включала в себя усовершенствованный интерфейс, поддержка HD-разрешения экрана и переход на файловую систему Ext4.

В начале 2011 года Google показали новую разработанную исключительно для планшетных ПК версию Android – 3.0 Honeycomb. Оказалось что на планшетах Honeycomb и была лучше, чем растянутый смартфонный интерфейс Gingerbread. Но были и проблемы: к примеру, встроенный кардридер планшетов начал работать только после обновления до Android 3.1.

Последней сугубо планшетной версией Android стала 3.2, которую получили все популярные планшеты того времени: Motorola Xoom, Acer Iconia Tab, ASUS Eee Pad Transformer, Samsung Galaxy Tab 10.1, Lenovo ThinkPad Tablet, Sony Tablet P и др.

В конце 2011 года свершилось слияние смартфонной и планшетной версий Android – обновленная 4.0 Ice Cream Sandwich одинаково хорошо подходила для обоих классов устройств. Еще одним важным новшеством ICS стала поддержка двух новых процессорных архитектур: к ARM добавились Intel x86 и MIPS.

В конце 2012 года параллельно с выходом Android 4.2 состоялся показ сразу двух новых гаджетов Google: смартфона LG Nexus 4 и планшета Samsung Nexus

10. Доработанная операционная система дала возможность переключаться между пользовательскими профилями, что важно в первую очередь для планшетных ПК, которыми поочередно пользуются все члены семьи.

Пятый день рождения мобильной платформы Android ее фанаты празднуют, имея на руках свежую версию 4.3 и зная имя следующей – 4.4 Kit Kat. За прошедшее время Android превратилась из гадкого утенка, какой она была до версии Gingerbread, в прекрасного лебедя.

Компании же Google можно гордиться тем, что ее операционная система установлена на более чем 1 млрд. мобильных гаджетов в мире.

### **1.8.2 Основные преимущества и недостатки Android**

Android обладает рядом преимуществ, которые дают ему возможность быть более интересным для пользователя, чем другие платформы:

- большинство приложений равноправны – и встроенные и установленные пользователем. А выбор программы по умолчанию осуществляется простым нажатием кнопки настройки. Такая гибкость практически недоступна другим ОС;

- безопасность данных. Следовательно, все приложения открываются в отдельной области памяти и на своей виртуальной машине;

- программные обеспечения и игры для **Android** работают с минимальной скоростью и при этом потребляют большой объем оперативной памяти;

- частично открытый исходный код. Как известно, операционная система Android очень похожа на браузер Google Chrome;

- информативные и анимированные виджеты – мобильная ОС от Google ввела стандарты на виджеты, которые фактически отсутствовали до появления Android. На сегодняшний день, почти все современные приложения для этой платформы имеют свой виджет для удобного использования в одно нажатие с рабочего стола;

- сервисы Google и синхронизация – большое количество облачных сервисов компании Google уже переустановлены и работают лучше, чем на любой другой платформе. Синхронизация с браузера ПК с браузером телефона при использовании Google Chrome для Android;

- недорого и функционально - Мобильная операционная система Android – приемлемый в цене способ соприкоснуться с самыми современными облачными технологиями. Цены смартфонов Android порой начинается от 100\$, что, конечно же, выделяет его от конкурентов;

- современный магазин приложений Google Play Market – Android имеет собственный быстродействующий магазин приложений, где возможно выбрать для себя любой из сотен тысяч приложений, которые расширяют функционал смартфона. Работать с магазином Google Play не сложно, а для того чтобы начать — достаточно всего лишь завести себе почту от Google. Никаких утомительных регистраций и прочих проблем. Google Play – это современно, удобно и быстро;

- разные бесплатные приложения. Бесплатные даже те, которые на других платформах стоят денег.



- смартфон как носитель файлов – файловая структура Android частично открыта – всегда можно использовать телефон как устройство для хранения любых файлов;

- поддержка карт памяти – практически все смартфоны на базе Android поддерживают Micro SD карты.

К недостаткам операционной системы Android можно отнести:

- основным минусом, который преследовал Android это тормоза, лаги и разные зависания. Эта проблема частично остается на бюджетных телефонах, но никак не на новинках;

- большое потребление интернет-трафика. Даже если не пользоваться электронной почтой, браузером, клиентами социальных сетей и прочими приложениями, деньги со счета все равно исправно списываются. Android в каком-то смысле живет своей жизнью - исправно отправляет отчеты о проделанной работе команде разработчиков. Эти данные тщательно анализируются и позволяют быстро вносить изменения в готовящиеся обновления. Передачу этой информации можно отключить в настройках, но не многие пользователи знают об этом;

- из-за того, что смартфонов на Android очень много, обновления к ним делаются Google, а потом переделывается под оболочку компаниями - изготовителями смартфонов. И обновления получают только дорогие модели;

- из-за того, что каждый может написать приложение для данной ОС и выложить его в магазин приложений, их не проверяют и поэтому в Google Play много разного и даже вирусного ПО.

## **1.9 GooglePlay**

Google Play (предыдущее название – Android Market) – магазин приложений компании Google, позволяющий владельцам устройств с операционной системой Android устанавливать и приобретать различные приложения.

Учётная запись разработчика, которая даёт возможность публиковать приложения, стоит \$25, то есть около 800 рублей. Платные приложения могут публиковать разработчики не из всех стран. Google Play стал результатом ребрендинга портала Android Market 6 марта 2012 года [3].

С момента ребрендинга из Android Market в Google Play и до сих пор использовать все возможности сервиса могут только жители США, но вскоре Google обещает значительно увеличить список стран, которые могут использовать возможности Google Play в полную меру, ведь недавно Google Play подвергся некоторым изменениям.

В Google Play можно найти много полезных и разнообразных приложений. В магазине присутствуют платные и бесплатные приложения. 31 октября 2012 года компания Google объявила, что количество приложений достигло 700 000 приложений, и количество скачиваний достигло 25 миллиардов раз [8].

Все приложения, установленные с Google Play, привязаны к аккаунту Google. Поэтому при покупке нового смартфона или планшета, легко можно перенести

их на новое устройство. При этом для установки приложений даже необязательно иметь под рукой мобильное устройство. Все можно сделать удаленно на веб-сайте Google Play с помощью ноутбука или ПК.

## **1.10 Выбор средств разработки и тестирования**

Выбор устройств-коммуникаторов в качестве аппаратной основы для реализации проекта и AndroidOS в качестве операционной системы в значительной степени обусловил и то, какой инструментарий будет использоваться для разработки и тестирования приложений.

### **1.10.1 Язык Java**

Система разрабатывалась на объектно-ориентированном языке Java, разработанном компанией Sun Microsystems, которая в данный момент приобретена корпорацией Oracle. Выбор данного языка был сделан по ряду соображений.

Программы на языке Java транслируются в байт-код, выполняемый виртуальной машиной (Java Virtual Mashine) – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинство данного способа выполнения программ заключается в полной независимости байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. На коммуникаторах с операционной системой Android такую функцию выполняет виртуальная машина Dalvik.

Полный контроль исполнения программы виртуальной машиной выливается в организацию гибкой системы безопасности. Любые операции, которые превышают установленные полномочия программы, вызывают коллизии при работе с данными) вызывают немедленное прерывание, которое может быть соответственно обработано.

К недостаткам концепции виртуальной машины можно отнести то, что исполнение байт-кода виртуальной машиной в некоторой степени снижает производительность программ и алгоритмов, реализованных на языке Java. Но и влияние этих особенностей сведено к минимуму, благодаря применению технологии трансляции байт-кода в машинный код непосредственно во время работы программы (JIT-технология) и широкому использованию платформенно-ориентированного кода в стандартных библиотеках.

Система Java создавалась объектно-ориентированной с самого начала. Объектно-ориентированная парадигма во многом наиболее удобна при создании программного обеспечения.

Нужно отметить, что существует возможность разрабатывать программы и на C/C++ (с помощью Native Development Kit), и на Basic (с помощью Simple) и с использованием других языков. Также можно создавать собственные программы с помощью конструкторов приложений, таких как App Inventor.

### 1.10.2 Java Development Kit

Для разработки программ на языке Java необходимо специальное программное обеспечение.

Java Development Kit (JDK) – бесплатно распространяемый корпорацией Oracle Corporation комплект разработчика приложений на языке Java, включающий в себя стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java (JRE) [3].

Стоит отметить такой важный факт, что в состав JDK не входит интегрированная среда разработки, которая устанавливается отдельно.

Все современные интегрированные среды разработки приложений на Java, такие, как NetBeans IDE, Sun Java Studio Creator, IntelliJ IDEA, Borland JBuilder, Eclipse, опираются на сервисы, предоставляемые JDK. Большинство из них для компиляции Java-программ используют компилятор из комплекта JDK. Поэтому эти среды разработки либо включают в комплект поставки одну из версий JDK, либо требуют для своей работы предварительной инсталляции JDK на машине разработчика.

### 1.10.3 Android SDK

Android SDK (Software Development Kit, далее SDK) используется для разработки приложений для Android. Он состоит из широкого набора инструментов, документации, утилит и примеров. В него также включены Java-библиотеки, необходимые для создания приложений для Android и содержащие API для фреймворка приложения. В качестве средства разработки поддерживаются все основные операционные системы.

К основным возможностям SDK можно отнести:

- отладчик, способный отлаживать приложения, запущенные на реальном устройстве или эмуляторе;
- профиль памяти и производительности, помогающий обнаружить утечки памяти и найти неэффективный код;
- эмулятор устройства, основанный на QEMU (виртуальной машине с открытым кодом, эмулирующей различные аппаратные платформы), он довольно точен, хотя не всегда быстр;
- утилиты командной строки для связи с устройствами;
- скрипты и утилиты для создания пакетов и развертывания приложений.

SDK может быть интегрирован в IntelliJIdea – удобную, популярную и функциональную среду разработки (IDE) для Java. Эта интеграция достигается с помощью плагина Android Development Tools (далее ADT), добавляющего новые возможности в IntelliJIdea для создания проектов для Android, их исполнения и отладки в эмуляторе или на устройстве, создания пакетов для их развертывания на GooglePlay. SDK и плагин ADT для Eclipse постоянно обновляются, в них добавляются новые возможности. Поэтому регулярно их обновлять – хорошая идея. Любой хороший SDK должна сопровождать исчерпывающая документация. Android SDK не исключение – помимо документации с ним

поставляется много примеров. Кроме того, найти руководство разработчика и полное описание API для всех модулей фреймворка приложения можно на официальном сайте.

Операционная система Android может работать в устройствах множества форм-факторов с разными размерами экрана, аппаратными возможностями и функциями. Как правило, на устройстве установлено много приложений (API Android), а также аппаратных компонентов, включая датчики, GPS, SD-карту памяти и сенсорный дисплей с особыми размерами.

Разработчики Android, которые создают, тестируют и отлаживают приложения на многочисленных устройствах, нуждаются в различных API Android (например, Gingerbread, Honeycomb) и инструментах, упрощающих эти задачи.

В Android SDK реализована поддержка эмуляции устройств. Эмулятор достаточно гибок в настройке и содержит множество программных и аппаратных параметров. Для настройки эмулятора разработчики могут использовать виртуальное устройство – Android VirtualDevice (далее AVD). AVD представляет собой набор конфигурационных файлов, в которых задаются различные программные и аппаратные параметры устройства Android. Эмулятор Android использует эти конфигурации AVD для настройки и запуска соответствующего виртуального образа Android.

Конфигурация AVD обычно включает в себя:

- аппаратный профиль, содержащий все возможности устройства (например, наличие камеры, датчиков);
- системный образ, который эмулятор будет использовать для данного AVD (целевой уровень API, например, Gingerbread или Honeycomb);
- образ данных, который представляет выделенное дисковое пространство для пользовательской информации, настроек и SD-карты;
- прочие настройки, в том числе стиль оформления эмулятора, размеры экрана и объем SD-карты.

Прежде чем использовать эмулятор Android, необходимо создать конфигурацию AVD, в которой будут заданы программные и аппаратные настройки, а также целевой уровень API.

Для этого в Android SDK имеется инструмент под названием `android`. С ним можно работать двумя способами: через командную строку или с помощью графического интерфейса.

Инструмент `android` находится в папке `tools` установочного каталога SDK. Например, в операционной системе Linux он расположен в `android-sdk-linux_x86/tools/android`

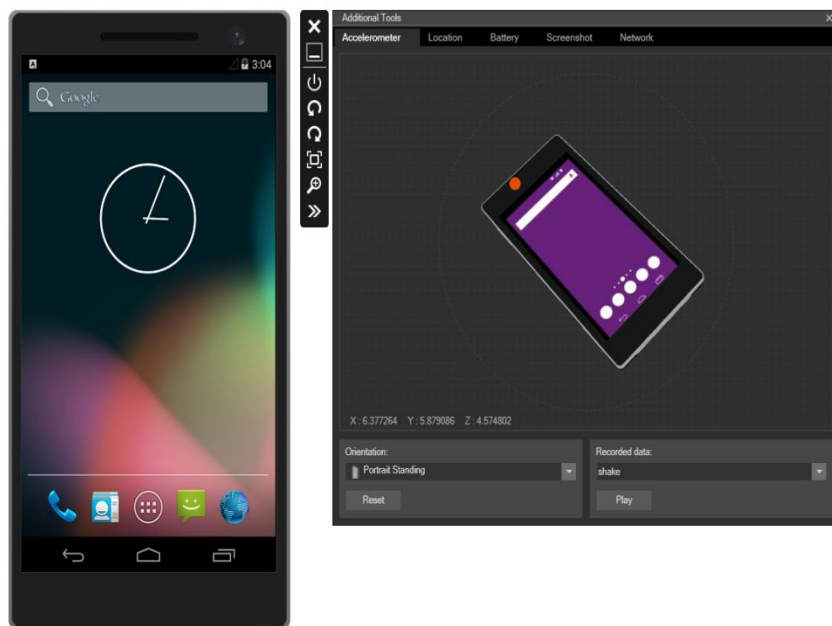


Рисунок 1.3 – Внешний вид эмулятора Android x86 с запущенной в нем операционной системой Gingerbread

Прежде чем использовать эмулятор Android, необходимо создать конфигурацию AVD, в которой будут заданы программные и аппаратные настройки, а также целевой уровень API.

Для этого в Android SDK имеется инструмент под названием android. С ним можно работать двумя способами: через командную строку или с помощью графического интерфейса.

Инструмент android находится в папке tools установочного каталога SDK. Например, в операционной системе Linux он расположен в android-sdk-linux\_x86/tools/android

## 2 Проектирование

Для описания бизнес-процессов используется язык UML (Unified Modeling Language) в программной среде Rational Rose.

### 2.1 Язык UML

UML – это унифицированный графический язык моделирования для описания, визуализации, проектирования и документирования ОО систем [11]. UML призван поддерживать процесс моделирования программного средства на основе объектно-ориентированного подхода, организовывать взаимосвязь концептуальных и программных понятий, отражать проблемы масштабирования сложных систем. Модели на UML используются на всех этапах жизненного цикла программного средства, начиная с бизнес-анализа и заканчивая сопровождением системы. Разные организации могут применять UML по своему усмотрению в зависимости от своих проблемных областей и используемых технологий.



Унифицированный язык моделирования UML завоевал широкое признание в качестве стандартного отраслевого языка для определения, визуализации, создания и документирования артефактов программных систем. Он упрощает сложный процесс проектирования программного обеспечения путем создания «чертежа» для построения системы.

## **2.2 Rational Rose – лидер среди CASE-средств**

Мощный толчок CASE-средства получили в пору внедрения объектно-ориентированной технологии разработки программного обеспечения. Старые технологии разработки программ «сверху вниз» уже не могли справиться с все усложняющимися, труднообозримыми программными комплексами.

Сегодня Rational Rose лидирует среди других CASE-средств, и не случайно. То, что этот пакет позволяет создавать сложные программные системы от замысла до создания исходного кода, привлекает не только проектировщиков систем, но и программистов-разработчиков. За рубежом по причине сильной конкуренции между фирмами-разработчиками программ ни один, даже небольшой программный проект, не обходится без применения CASE-средств. Уже более 50 тысяч больших и маленьких компаний по всему миру используют Rational Rose для разработки программных систем. Это такие известные компании как NASA, Boeing, Lockheed Martin, Honey-well, NBC, Reuters, AT&T и другие.

### **2.2.1 Достоинства Rational Rose**

Одним из достоинств этого программного продукта является возможность использования UML-диаграмм. В распоряжение проектировщика система Rational Rose представляет следующие типы диаграмм, последовательное создание которых позволяет получить полное представление о всей проектируемой системе и об отдельных ее компонентах:

- Use Case diagram (диаграмма вариантов использования);
- Deployment diagram (диаграмма топологий);
- Statechart diagram (диаграмма состояний);
- Activity diagram (диаграмма активности);
- Interaction diagram (диаграмма взаимодействий);
- sequence diagram (диаграмма последовательности);
- collaboration diagram (диаграмма сотрудничества);
- Class diagram (диаграмма классов);
- Component diagram (диаграмма компонентов).

### **2.2.2 Моделирование предметной области**

Моделирование предметной области включает:

- моделирование бизнес-процессов;
- спецификацию бизнес-правил;
- учет внешних сущностей, имеющих отношение к бизнесу;
- моделирование бизнес-сущностей предметной области.

Моделирование бизнес-процессов предполагает исследование внутренних и внешних компонентов бизнеса и их взаимосвязи. В UML для документирования полученных данных используется бизнес-модель.

Моделирование бизнес-процессов является первым этапом унифицированного процесса разработки, результатом которого является установление контекста для остальной части проекта. Во время проектирования системы, моделирование бизнес-процессов помогает не забывать об основных побудительных причинах построения системы.

В основе концепции моделирования бизнес-процессов входят такие понятия, как: бизнес-роль, сотрудник, Бизнес-вариант использования, диаграмма Бизнес-вариантов использования, Бизнес-сущность и диаграмма Деятельности.

Бизнес-роль исполняет человек или подразделение, внешнее по отношению к организации, но взаимодействующие с ней.

Сотрудник – роль внутри организации (не должность). Один человек может играть несколько ролей, занимая одну должность.

### **2.2.3 Диаграмма Бизнес-вариантов использования**

На диаграммах этого типа показаны Бизнес-варианты использования, бизнес-роли и сотрудники организации, а также отношения между ними. Это дает полную модель хозяйственной деятельности организации и описание внешних по отношению к ней сущностей. Диаграмма Бизнес-вариантов использования описывает зону действия организации, то есть ее окружение и границы.

Диаграмма предназначена для быстрого изучения высокоуровневых отношений в бизнесе без показа всех подробностей.

### **2.2.4 Диаграмма деятельности**

Диаграмма деятельности – это способ графического изображения модели рабочих потоков для вариантов использования. Диаграмма показывает шаги (этапы) рабочего потока, точки принятия решений в этом потоке, ответственности для каждого этапа и объекты, влияющие на рабочий поток.

Основными элементами диаграммы деятельности являются:

- дорожки, показывающие сферы ответственности сотрудников, участвующих в Бизнес-варианте использования;
- деятельности (activity), демонстрирующие этапы рабочих потоков;
- действия (actions), показывающие шаги в пределах каждой деятельности. Действие может происходить при входе в деятельность, во время ее исполнения, при выходе из деятельности или при возникновении определенного события.
- бизнес-объекты, являющиеся сущностями, воздействующими на рабочий поток. На эти объекты воздействует рабочий поток, в результате чего объекты изменяют свое состояние;
- переходы, показывающие движение рабочего потока между деятельностями;

- точки принятия решений, демонстрирующие решения, которые необходимо принять в пределах рабочего потока;
- синхронизации, иллюстрирующие два или более одновременных этапа в пределах рабочего потока;
- исходное состояние, определяющее начало рабочего потока;
- конечное состояние, определяющее завершение рабочего потока.

### 2.3 Модель предметной области

С помощью CASE-средств Rational Rose составлена модель предметной области (рисунок 2.1).

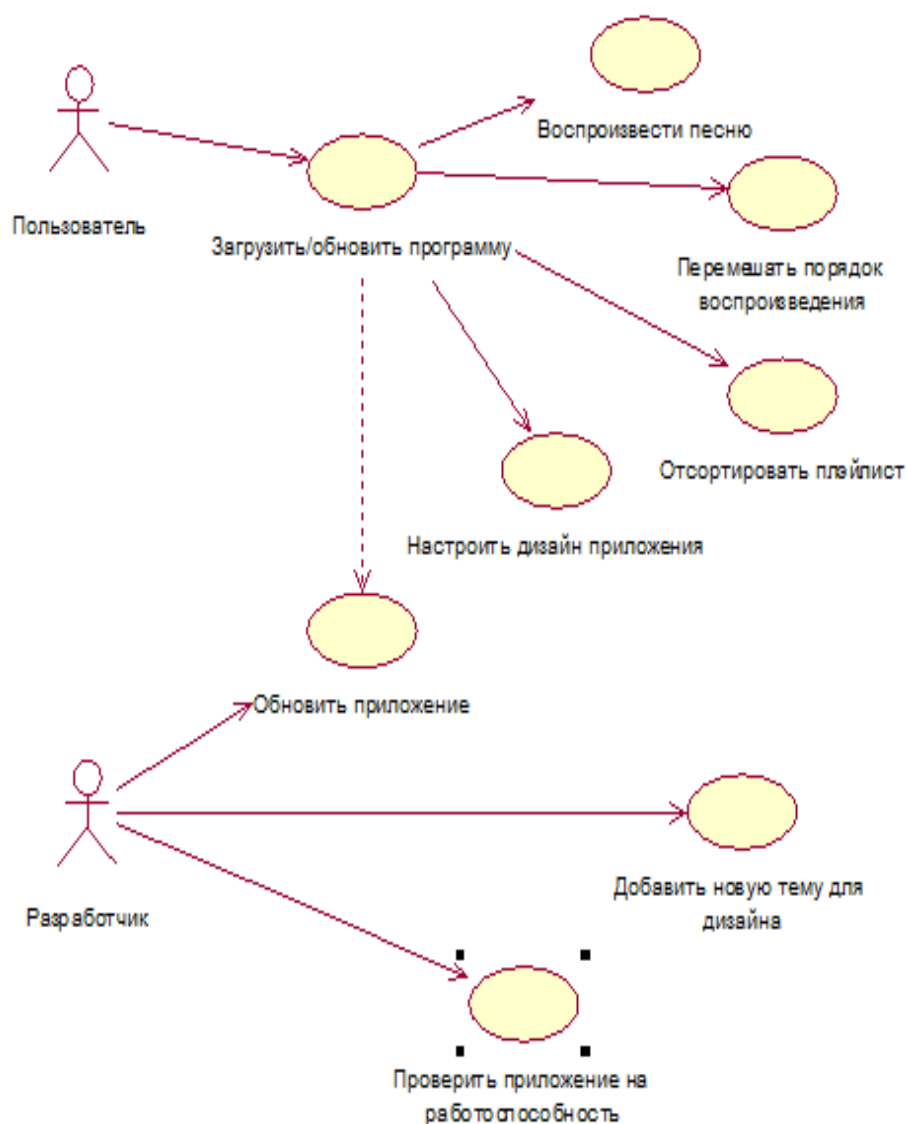


Рисунок 2.1 – Модель предметной области

Администратор добавляет дисциплину в программный код приложения.

После добавления дисциплин, Администратор добавляет новую учебную литературу или удаляет старую литературу.

Измененное приложение Администратор обновляет в Google Play.

Клиент устанавливает или обновляет измененное Администратором приложение через магазин приложений GooglePlay.

Открыв установленное или обновленное приложение, Клиент выбирает нужную дисциплину.

В выбранной дисциплине Клиент просматривает список литературы для загрузки.

Загруженная Клиентом литература открывается в установленном по выбору программе для чтения файлов в формате PDF.

## **2.4 Диаграмма вариантов использования**

Вариант использования представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом). Вариант использования описывает типичное взаимодействие между пользователем и системой. Например, два типичных варианта использования обычного текстового процессора – «сделать некоторый текст полужирным» и «создать индекс». Даже на таком простом примере можно выделить ряд свойств варианта использования: он охватывает некоторую очевидную для пользователей функцию, может быть как небольшим, так и достаточно крупным и решает для пользователя некоторую дискретную задачу. В простейшем случае вариант использования определяется в процессе обсуждения с пользователем тех функций, которые он хотел бы реализовать.

Действующее лицо (actor) – это роль, которую пользователь играет по отношению к системе. Действующие лица представляют собой роли, а не конкретных людей или наименования работ. Несмотря на то, что на диаграммах вариантов использования они изображаются в виде стилизованных человеческих фигурок, действующее лицо может также быть внешней системой, которой необходима некоторая информация от данной системы (например, Система учета). Показывать на диаграмме действующих лиц системы следует только в том случае, когда им действительно необходимы некоторые варианты использования.

Все варианты использования, так или иначе, связаны с внешними требованиями к функциональности системы. Если Системе учета требуется файл, то это требование должно быть удовлетворено. Варианты использования всегда следует анализировать вместе с действующими лицами системы, определяя при этом реальные задачи пользователей и рассматривая альтернативные способы решения этих задач.

Действующие лица могут играть различные роли по отношению к варианту использования. Они могут пользоваться его результатами или могут сами

непосредственно в нем участвовать. Значимость различных ролей действующего лица зависит от того, каким образом используются его связи.

Хорошим источником для идентификации вариантов использования служат внешние события. Следует начать с перечисления всех событий, происходящих во внешнем мире, на которые система должна каким-то образом реагировать. Какое-либо конкретное событие может повлечь за собой реакцию системы, не требующую вмешательства пользователей, или, наоборот, вызвать чисто пользовательскую реакцию. Идентификация событий, на которые необходимо реагировать, помогает выделить варианты использования.

Варианты использования являются необходимым средством на стадии формирования требований к ПО. Каждый вариант использования – это потенциальное требование к системе, и пока оно не выявлено, невозможно запланировать его реализацию.

На рисунке 2.2 показана диаграмма вариантов использования.

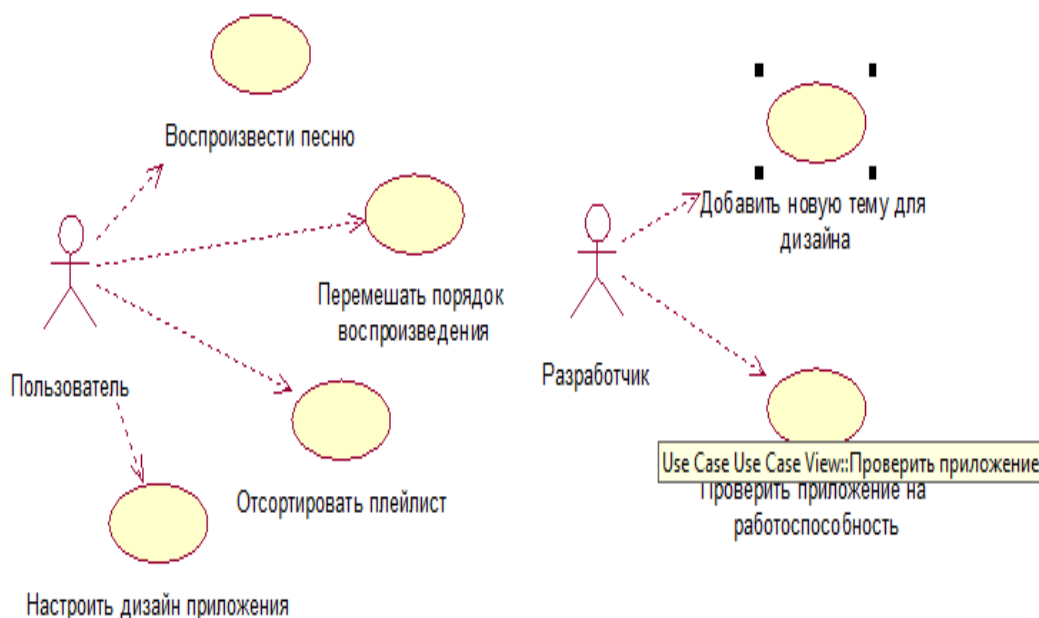


Рисунок 2.2 – Диаграмма вариантов использования

## 2.5 Диаграммы взаимодействий

Диаграммы последовательностей и диаграммы кооперации относятся к диаграммам взаимодействия. Они отражают характер взаимодействия объектов между собой, которое необходимо для выполнения варианта использования, операции или другой поведенческой компоненты.

Диаграммы последовательностей показывают обмен сообщениями между объектами, упорядоченными в виде временной последовательности.

Диаграммы кооперации выделяют отношения между объектами, по которым происходит обмен сообщениями.

На диаграммах взаимодействий показывают связи, включающие множество объектов и отношений между ними, в том числе сообщения, которыми объекты



обмениваются. При этом диаграмма последовательностей акцентирует внимание на временной упорядоченности сообщений, а диаграмма кооперации – на структурной организации посылающих и принимающих сообщения объектов.

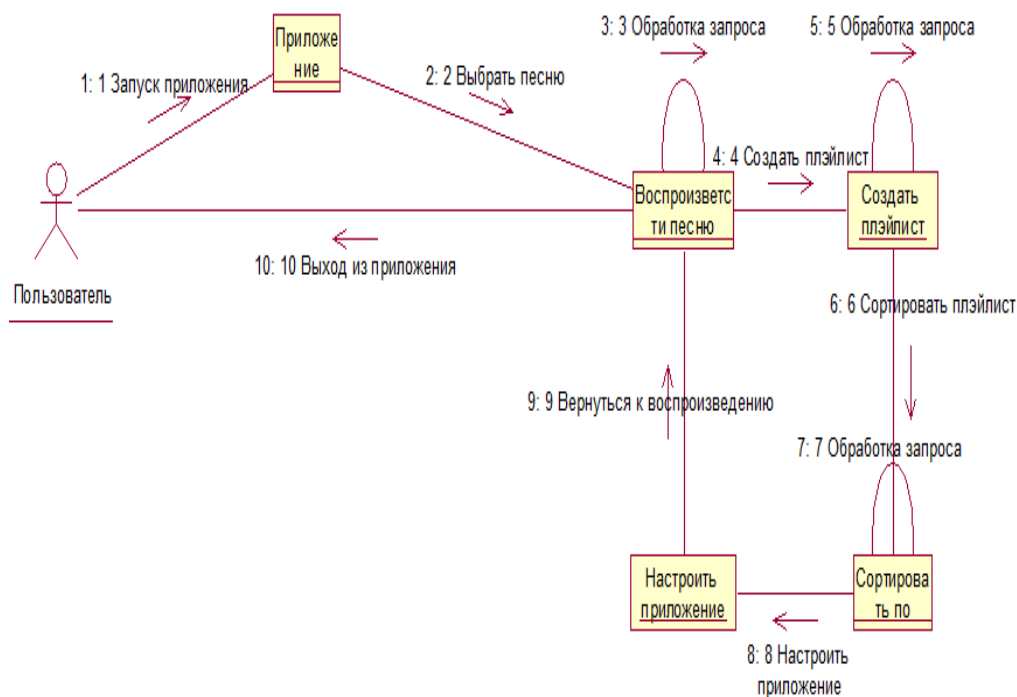


Рисунок 2.3 – Диаграмма коопераций для пользователя

### 3 Разработка мобильного приложения

#### 3.1 Цель и рамки проекта

Необходимо разработать приложение – музыкальный проигрыватель для платформы Android. Приложение должно содержать 7 оконные формы:

- плейлисты;
- недавние;
- исполнители;
- альбомы;
- треки;
- жанры;
- фоновый экран

Проект разрабатывается с целью воспроизведение аудиозаписей различных форматов для пользователей любого возраста.

При запуске приложения пользователям, открывается окно треки со списком аудиозаписей. При выборе аудиозаписи открывается фоновое окно.

Предварительно, функциональные требования, требования к данным

приложения проектируются на языке UML –Rational Rose.

Приложение должно сортировать аудиозаписи по жанрам , альбомам и исполнителям . Далее должен воспроизвести аудиозапись выбранный пользователем.

Срок разработки системы 1 месяц и 3 недели. С февраля 2016 года до 22 марта 2016 года. Срок делится на 3 этапа: разработка проекта, разработка приложения, тестирование и отладка.

### **3.2 Создание приложения в Android SDK**

При создании проекта нужно указать его свойства: Project Name (название проекта), Build Target (версия сборки), Application Name (название приложения), Package Name (название пакета), Create Activity (форма), и Min SDK Version (минимально допустимая версия SDK).

В поле «версия сборки» указывается, какой API будет использоваться в проекте. От выбора зависит набор поддерживаемых функций и версии ОС, на которых приложение сможет работать.

Название пакета определяет уникальное название проекта среди всех программ в ОС Android. Название пакета обычно состоит из стандартного названия компании и названия приложения, например «com.myscompany.appname».

Форма (Activity) представляет собой отдельное окно с элементами пользовательского интерфейса. Данное окно реализует некоторую функциональность приложения и может обрабатывать различные события системы. Управление формами в системе осуществляется при помощи стека форм. Запущенная форма помещается в вершину стека, предыдущая открытая форма автоматически смещается на один уровень стека ниже, при этом вся информация о предыдущей форме сохраняется. Каждая форма имеет четыре состояния функционирования:

- запущенное или активное – форма отображается на переднем плане, т.е. активна;
- приостановленное – форма не имеет фокуса, но при этом видна на экране, т.е. частично перекрывается другой формой;
- остановленное – форма полностью перекрыта другой формой, т.е. не видна пользователю;
- завершенное – форма, если она находилась в состоянии «приостановленное» или «остановленное», может быть выгружена из памяти, или может быть завершен процесс, создавший форму. При повторном запуске формы, ее предыдущее состояние восстановится.

Для формы существует три вида циклов:

- полный цикл – форма проходит через все этапы своей жизни от запуска до завершения;
- цикл видимости – форма видна пользователю на экране, но не активна, то есть пользователь не может взаимодействовать с ней;

– цикл активности – форма является активной, и пользователь может взаимодействовать с ней.

На рисунке 3.1 показана структура файлов проекта тестовой программы для работы с базой данных.

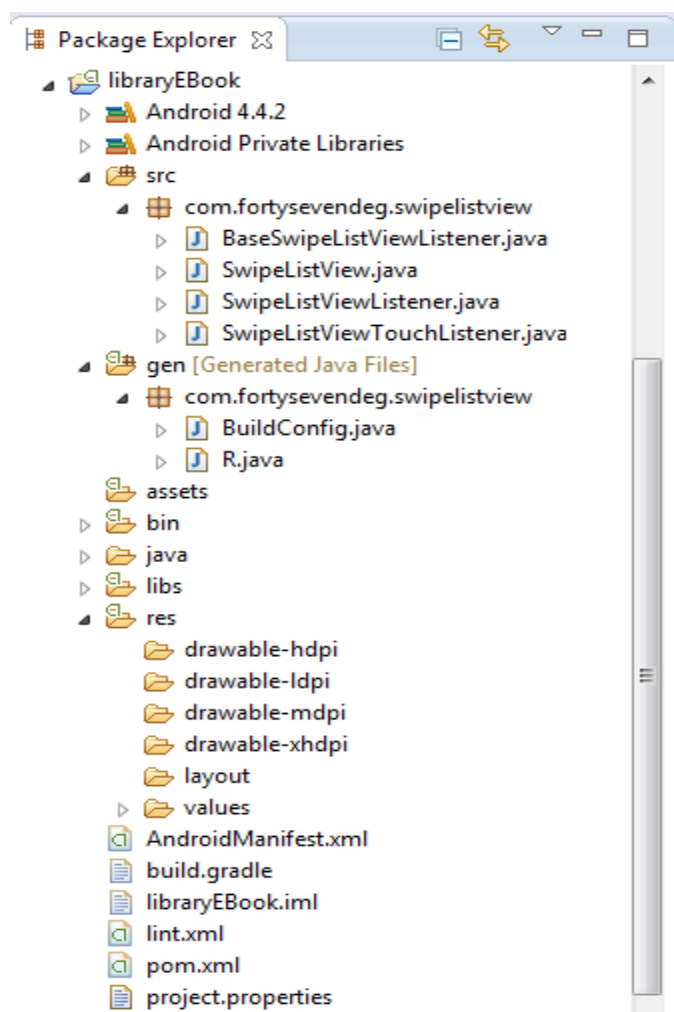


Рисунок 3.1 – Структура файлов проекта

Основными файлами проекта при разработке являются:

- файл «/gen/R.java» – это специальный сгенерированный класс, посредством которого осуществляется доступ к ресурсам приложения;
- папка «/res/drawable-dpi/» содержит все графические файлы, используемые в приложении, для разных разрешений экрана;
- папка «/res/layout/» содержит файлы, в которых в формате XML описывается внешний вид форм;
- файл «/res/layout/main.xml» содержит макет главной формы приложения;
- папка «/res/values/» содержит файлы, содержащие общие константы для всего приложения;
- файл «/res/values/SwipeListView\_attrs.xml» – содержит все атрибуты, используемые в классе SwipeListView;

- папка «/src/» содержит файлы, содержащие исходный код программы на языке Java;
- файлы «/src/BaseSwipeListViewListener.java», «/src/SwipeListView.java», «/src/SwipeListViewListener.java», «/src/SwipeListViewTouchListener.java» – содержат ответные действия для свайпа и являются библиотеками;
- файл «Android Manifest.xml», в котором перечисляются общие свойства проекта, а также все формы, входящие в проект.

### 3.3 Реализация приложения

Программа была реализована на языке Java в среде разработки Eclipse с помощью инструментов для разработки Android SDK. Разработка программы производилась на основе изученных методов разработки приложений для ОС Android. Разработанный проект содержит файлы с исходным кодом на языке Java, представленные на рисунке 3.2, и файлы с описанием интерфейса программы на языке XML, представленные на рисунке 3.3.

Каждый файл с исходным кодом содержит реализацию отдельного класса программы. Названия классов соответствуют названиям файлов, в которых они расположены.

Классы «Base Swipe List View Listener», «Swipe List View.java», «Swipe List View Listener», «Swipe List View Touch Listener» наследуются от базового класса Activity, входящего в комплект средств разработки Android SDK, и реализуют формы приложения. В них реализуются алгоритмы взаимодействия пользователя с элементами интерфейса приложения. Вид интерфейса описан в XML файле.

В pom.xml описано действие приложения для работы с интернетом.

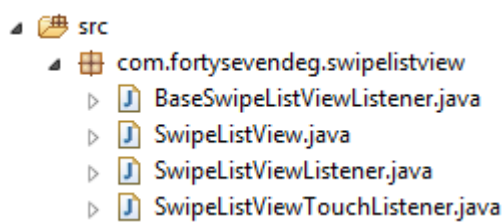


Рисунок 3.2 – Файлы с исходным кодом

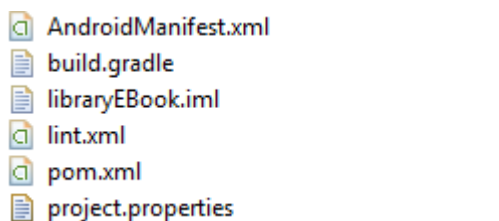


Рисунок 3.3 – Файлы .xml

Интерфейс разработанной программы представлен в приложении А.

Запуск и отладка приложения производилась на эмуляторе, входящем в состав комплект средств разработки Android SDK, а также на реальном мобильном устройстве с операционной системой Android версии 4.3.

### **3.4 Тестирование приложения**

Для запуска приложения, заходим в меню телефона и запускаем значок указанный на рисунке (рисунок 3.4).

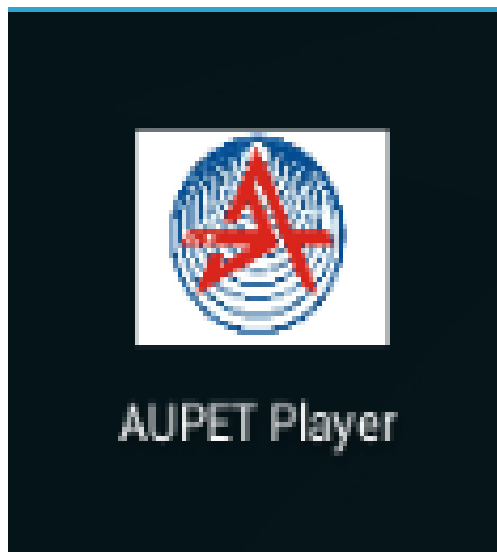


Рисунок 3.4 – Заставка приложения

При запуске программы открывается окно приложения с заставкой, предоставленной на рисунке (рисунок 3.5)



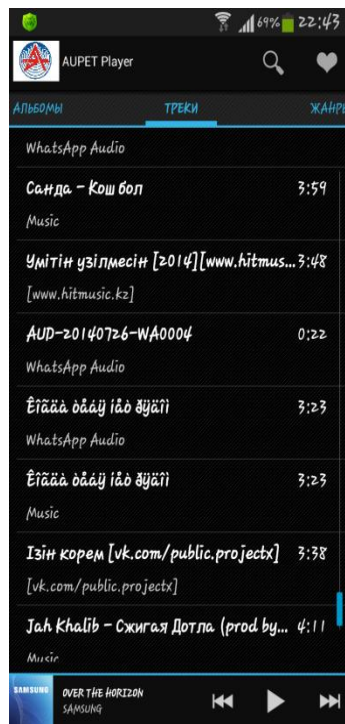


Рисунок 3.5 – Начальное меню приложения

Первая вкладка это жанры, в котором можно выбрать вид композиции (рисунок 3.6)

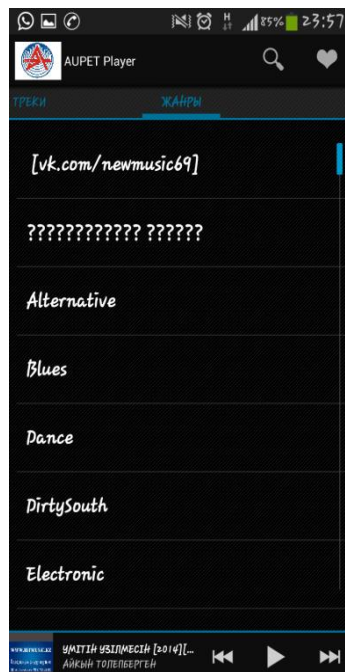


Рисунок 3.6 – Вкладка жанры

Вторая вкладка это альбомы, в котором можно выбрать предпочитаемый альбом (рисунок 3.7)

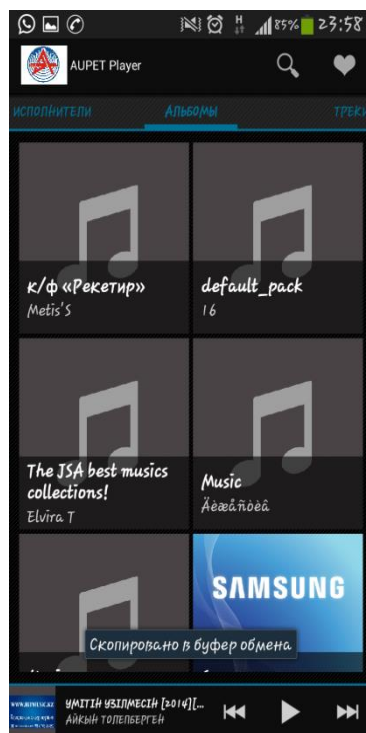


Рисунок 3.7 – Вкладка альбом

Третья вкладка исполнители, в котором можно выбрать один из указанных исполнителей (рисунок 3.8)

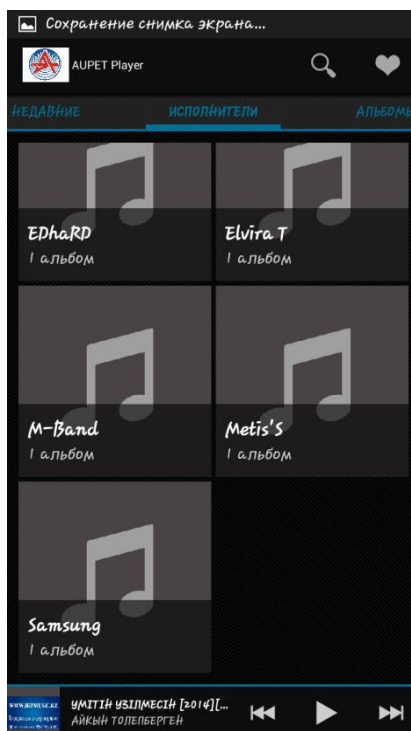


Рисунок 3.8 – Вкладка исполнители

Четвертая вкладка показывает недавние проигранные треки (рисунок 3.9)



Рисунок 3.9 – Вкладка недавние

Пятая вкладка плейлисты, в котором указаны: избранное, недавно добавленные, Quicklist (быстрый доступ к треку) (рисунок 3.10)

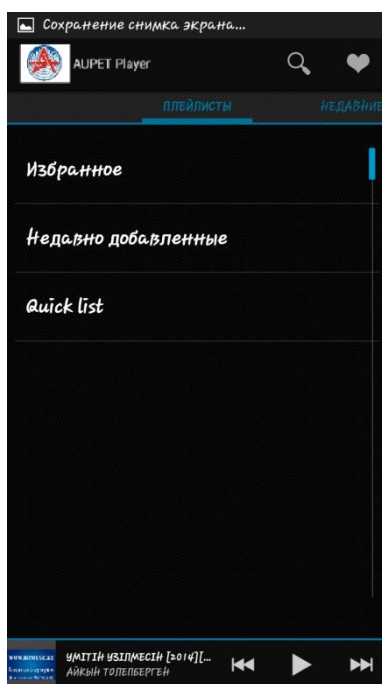


Рисунок 3.10 – Вкладка плейлисты

## 4. Расчетная часть

### 4.1 Расчет пропускной способности сети LTE

В ходе планирования радиосетей LTE наличествует ряд различий от процесса планирования иных технологий беспроводного радиодоступа. Основное различие – данное применение новейшего на подобии многостанционного доступа на базе технологии OFDM [6], благодаря чему бывают замечены свежие понятия и меняются методы проектирования. Процесс планирования радиосети состоит из 2-ух шагов:

- составление наибольшей площади покрытия;
- обеспечение требуемой емкости.

Проектирование радиосети LTE станет производиться в отдаленном микрорайоне, а данное означает, что плотность абонентов станет низка и базисные станции обязаны устанавливаться на максимальном удалении между собой имея цель приккрыть любой eNB как возможно немалую землю. В этой связи надо собрать сообразный частотный спектр. При таком варианте надо руководствоваться правилом, что чем ниже частота, тем далее распространение радиосигнала. Частотный диапазон 791 – 862 МГц вполне подойдет для выполнения этой задачи. Тип дуплекса выберем частотный – FDD.

Пропускную способность, либо емкость, сети расценивают, базирясь на средних значениях спектральной отдачи соты в явных критериях.

Спектральная отдача систем мобильной взаимосвязи являет из себя признак, вычисляемый как отношение скорости передачи этих на 1 Гц используемой полосы частот (бит/с/Гц). Спектральная отдача считается признаком производительности применения частотного ресурса, также определяет скорость передачи инфы в установленной полосе частот.

Спектральная отдача имеет возможность рассчитываться как отношение скорости передачи этих всех абонентов сети в явной географической области (соте, зоне) на 1 Гц полосы частот (бит/с/Гц/сота), также как отношение предельной пропускной возможности сети к ширине полосы 1-го частотного канала.

Центральная спектральная отдача для сети LTE, ширина полосы частот которой одинакова 20 МГц, для частотного вида дуплекса.

Таблица 4.1 - Средняя спектральная эффективность для сети LTE

| Линия | Схема MIMO | Средняя спектральная эффективность(бит/с/Гц) |
|-------|------------|--|
| UL    | 1x2        | 1,254  |
|       | 1x4        | 1,829  |
| DL    | 2x2        | 2,93   |
|       | 4x2        | 3,43   |
|       | 4x4        | 4,48   |

Для системы FDD центральная пропускная способность 1 раздела eNB быть может получена методом прямого умножения ширины канала на спектральную отдачу канала:

$$R = S \cdot W \quad (4.1)$$

где  $S$  – центральная спектральная отдача, бит/с/Гц;  $R_{DL}=3,43$  бит/с/Гц;  $R_{UL}=1,829$  бит/с/Гц/  
 $W$  – ширина канала, МГц;  $W = 10$  МГц.

Для полосы DL:

$$R_{DL} = 3,43 \cdot 10 = 34,3 \text{ Мбит/с.}$$

Для полосы UL:

$$R_{UL} = 1,829 \cdot 10 = 18,29 \text{ Мбит/с.}$$

Центральная пропускная способность базовой станции  $R_{eNB}$  рассчитывается методом умножения пропускной способности одного сектора на количество секторов базовой станции; число секторов eNB примем равное 3, тогда:

$$R_{eNB} = R_{\frac{DL}{UL}} \cdot 3 \quad (4.1.2)$$

Для полосы DL:

$$R_{eNB.DL} = 34,3 \cdot 3 = 102,9 \text{ Мбит/с.}$$

Для полосы UL:

$$R_{eNB.UL} = 18,29 \cdot 3 = 54,87 \text{ Мбит/с.}$$

Последующим шагом станет определение численности сот в намечаемой сети LTE.

Для расчета количества сот в сети нужно было найти единое количество каналов, выделяемых для развертывания проектируемой сети LTE. Единое количество каналов  $N_k$  рассчитывается по формуле:



$$N_k = \left( \frac{\Delta f_{\Sigma}}{\Delta f_k} \right) \quad (4.1.3)$$

где  $\Delta f_{\Sigma}$  - полоса частот, выделенная для работы сети и одинаковая 71 МГц;

$\Delta f_k$  – полоса частот 1-го радиоканала; под радиоканалом в сетях LTE ориентируется это понятие как ресурсный блок РБ, который имеет ширину 180 кГц,  $\Delta f_k = 180 \text{ кГц}$ .

$$N_k = \frac{71000}{180} \approx 395 \text{ каналов}$$

Далее определим число каналов  $N_{\text{к.сек}}$ , которое необходимо использовать для обслуживания абонентов в одном секторе одной соты:

$$N_{\text{к.сек}} = \left( \frac{N_k}{(N_{\text{кл}} \cdot M_{\text{сек}})} \right) \quad (4.1.4)$$

где  $N_k$  – суммарное число каналов;

$N_{\text{кл}}$  – размерность кластера, выбираемое с учетом количества секторов eNB, примем равным 3;

$M_{\text{сек}}$  – количество секторов eNB, принятое 3.

$$N_{\text{к.сек}} = \left( \frac{395}{(3 \cdot 3)} \right) \approx 43 \text{ канала}$$

Далее определим число каналов трафика в одном секторе одной соты  $N_{\text{кт.сек}}$ . Число каналов трафика рассчитывается по формуле:

$$N_{\text{кт.сек}} = N_{\text{км1}} \cdot N_{\text{к.сек}} \quad (4.1.5)$$

где  $N_{\text{км1}}$  – число каналов трафика в одном радиоканале, определяемое стандартом радиодоступа (для OFDMA  $N_{\text{мл}} = 1...3$ ); для сети LTE выберем  $N_{\text{км1}} = 1$ .

$$N_{\text{кт.сек}} = 1 \cdot 43 \approx 43 \text{ канала}$$

Согласно с моделью Эрланга, представленной повторяющий вид графика

на рисунке 3.1, определим возможную нагрузку в секторе одной соты  $A_{сек}$  при возможном значении вероятности блокировки одинаковой 1% и рассчитанным выше значением  $N_{кт.сек.}$ . Определим, что  $A_{сек} = 50$  Эрл.

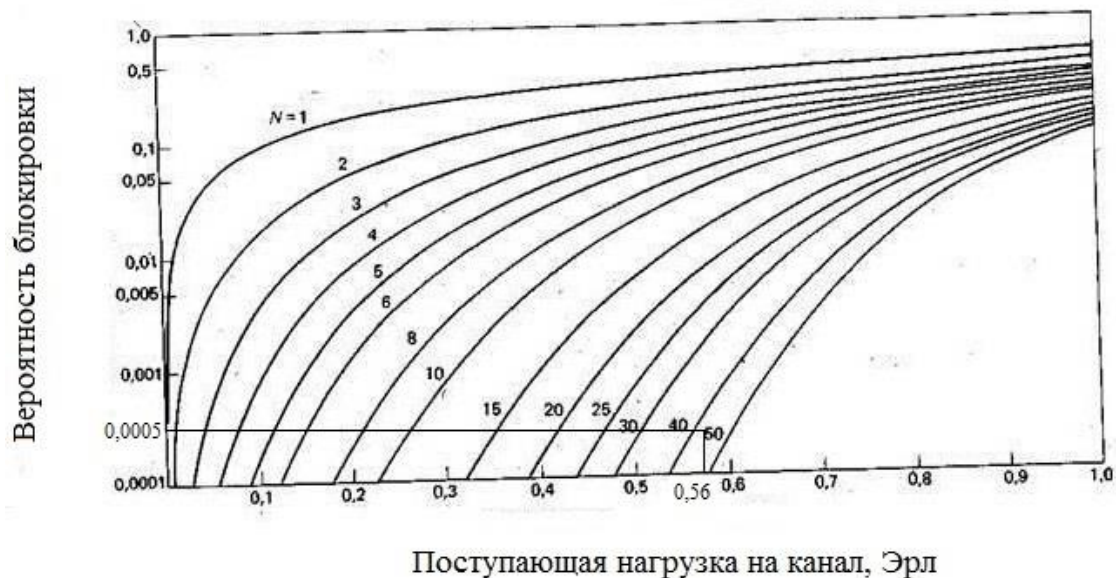


Рисунок 4.1 - Зависимость допустимой нагрузки в секторе от числа каналов трафика и вероятности блокировки

Количество пользователей, которые будут обслуживаться одной eNB, определяется по данной формуле:

$$N_{аб.eNB} = M_{сек} \cdot \left( \frac{A_{сек}}{A_1} \right) \quad (4.1.6)$$

где  $A_1$  – центральная по всем видам трафика абонентская перегрузка от 1-го абонента; значение  $A_1$  имеет возможность сочинять (0,04...0,2) Эрл. Т.к. проектируемая сеть намечается употребляться для скоростного размена информацией, то значение  $A_1$  примем одинаковым 0,2 Эрл. Следовательно:

$$N_{аб.eNB} = 3 \cdot \left( \frac{50}{0,2} \right) \approx 750 \text{ абонентов}$$

Число базовых станций eNB в проектируемой сети LTE найдем по формуле:

$$N_{eNB} = \left( \frac{N_{аб}}{N_{аб.eNB}} \right) + 1 \quad (4.1.7)$$

где  $N_{аб}$  – количество потенциальных абонентов. Количество потенциальных абонентов определим как 10% от общего числа жителей. Общее число жителей мкр. Саина, города Актау, по данным последней переписи населения 2010 года составляет примерно 20000 человек. Таким образом, количество потенциальных абонентов составит 2000 человек, тогда:

$$N_{eNB} = \left( \frac{2000}{750} \right) + 1 \approx 3,6 = 4eNB$$

Среднюю планируемую пропускную способность  $R_N$  для сети, которую проектируем определим путем умножения количества eNB на среднюю пропускную способность eNB. Формула примет вид:

$$R_N = (R_{eNB.DL} + R_{eNB.UL}) \cdot N_{eNB} \quad (4.1.8)$$

$$R_N = (102,9 + 54,87) \cdot 3,6 = 567,972 \text{ Мбит/с.}$$

Далее дадим проверочную оценку емкости проектируемой сети и сравним с рассчитанной. Определим усредненный трафик одного абонента в ЧНН:

$$R_{m.чнн} = \frac{T_m \cdot q}{N_{чнн} \cdot N_d} \quad (4.1.9)$$

где  $T_T$  – средний трафик одного абонента в месяц,  $T_T = 30$  Гбайт/мес;

$q$  – Коэффициент для городской местности,  $q = 2$ ;

$N_{чнн}$  – число ЧНН в день,  $N_{чнн} = 7$ ;

$N_d$  – число дней в месяце,  $N_d = 30$ .

$$R_{m.чнн} = \frac{30 \cdot 2}{7 \cdot 30} \approx 0,28 \frac{\text{Мбит}}{\text{с}}$$

Определим общий трафик проектируемой сети в ЧНН  $R_{общ./чнн}$  по формуле:

$$R_{общ./чнн} = R_{m.чнн} \cdot N_{акт.аб}, \quad (4.1.10)$$

где  $N_{a.m.аб}$  – число активных абонентов в сети; определим число активных абонентов в сети как 70% от общего числа потенциальных абонентов  $N_{аб}$ , то есть  $N_{a.m.аб} = 1400$  абонентов.

$$R_{общ./чнн} = 0,28 \cdot 1400 = 392 \frac{\text{Мбит}}{\text{с}}$$

Таким образом,  $R_N > R_{общ./чнн}$ . Это условие показывает, что проектируемая сеть не будет подвергаться перегрузкам в ЧНН.

## 4.2 Расчет пропускной способности сети 3G

В ходе планирования радиосетей 3G наличествует ряд различий от процесса планирования иных технологий беспроводного радиодоступа. Процесс планирования радиосети состоит из 2-ух шагов:

- составление наибольшей площади покрытия;
- обеспечение требуемой емкости.

Проектирование радиосети 3G станет производиться в отдаленном микрорайоне, а данное означает, что плотность абонентов станет низка и базисные станции обязаны устанавливаться на максимальном удалении между собой имея цель прикрыть любой eNB как возможно немалую землю. В этой связи надо собрать сообразный частотный спектр. При таком варианте надо руководствоваться правилом, что чем ниже частота, тем далее распространение радиосигнала. Частотный диапазон 791 – 862 МГц вполне подойдет для выполнения этой задачи. Тип дуплекса выберем частотный – FDD.

Пропускную способность, либо емкость, сети расценивают, базирясь на средних значениях спектральной отдачи соты в явных критериях.

Спектральная отдача систем мобильной связи является из себя признаком, вычисляемый как отношение скорости передачи этих на 1 Гц используемой полосы частот (бит/с/Гц). Спектральная отдача считается признаком производительности применения частотного ресурса, также определяет скорость передачи инфы в установленной полосе частот.

Спектральная отдача имеет возможность рассчитываться как отношение скорости передачи этих всех абонентов сети в явной географической области (соте, зоне) на 1 Гц полосы частот (бит/с/Гц/сота), также как отношение предельной пропускной возможности сети к ширине полосы 1-го частотного канала.

Центральная спектральная отдача для сети UMTS, ширина полосы частот которой 5 МГц, для частотного вида дуплекса.

Таблица 4.1 - Средняя спектральная эффективность для сети UMTS

| Стандарт         | Ширина полосы частот, МГц | Спектральная эффективность системы, бит/с/Гц/сота |
|------------------|---------------------------|---|
| UMTS<br>Релиз 99 | 5                         | 0,5   |

Для системы FDD центральная пропускная способность 1 раздела eNB быть может получена методом прямого умножения ширины канала на спектральную отдачу канала:

$$R = S \cdot W \quad (4.2.1)$$

где  $S$  – центральная спектральная отдача, бит/с/Гц;  $W$  – ширина канала, МГц;  $W = 5$  МГц.

Для полосы DL:

$$R_{DL} = 2,048 \cdot 5 = 10,24 \text{ Мбит/с.}$$

Для полосы UL:

$$R_{UL} = 0,5 \cdot 5 = 2,5 \text{ Мбит/с.}$$

Центральная пропускная способность базовой станции  $R_{eNB}$  рассчитывается методом умножения пропускной способности одного сектора на количество секторов базовой станции; число секторов eNB примем равное 3, тогда:

$$R_{eNB} = R_{\frac{DL}{UL}} \cdot 3 \quad (4.2.2)$$

Для полосы DL:

$$R_{eNB.DL} = 10,24 \cdot 3 = 30,72 \text{ Мбит/с.}$$

Для полосы UL:

$$R_{eNB.UL} = 2,5 \cdot 3 = 7,5 \text{ Мбит/с.}$$

Последующим шагом станет определение численности сот в намечаемой сети.

Для расчета количества сот в сети нужно было найти единое количество каналов, выделяемых для развертывания проектируемой сети 3 G. Единое количество каналов  $N_k$  рассчитывается по формуле:

$$N_k = \left( \frac{\Delta f_{\Sigma}}{\Delta f_k} \right) \quad (4.2.3)$$

где  $\Delta f_{\Sigma}$  - полоса частот, выделенная для работы сети и одинаковая 3,84 МГц;

$\Delta f_k$  – полоса частот 1-го радиоканала; под радиоканалом в сетях 3 G ориентируется это понятие как ресурсный блок РБ, который имеет ширину 15 кГц,  $\Delta f_k = 15$ кГц.

$$N_k = \frac{3840}{15} = 256 \text{ каналов}$$

Далее определим число каналов  $N_{к.сек}$ , которое необходимо использовать для обслуживания абонентов в одном секторе одной соты:

$$N_{к.сек} = \left( \frac{N_k}{(N_{кл} \cdot M_{сек})} \right) \quad (4.2.4)$$

где  $N_k$  – суммарное число каналов;

$N_{кл}$  – размерность кластера, выбираемое с учетом количества секторов eNB, примем равным 3;

$M_{сек}$  – количество секторов eNB, принятое 3.

$$N_{к.сек} = \left( \frac{256}{(3 \cdot 3)} \right) \approx 28 \text{ канала}$$

Далее определим число каналов трафика в одном секторе одной соты  $N_{км.сек}$ . Число каналов трафика рассчитывается по формуле:

$$N_{км.сек} = N_{км1} \cdot N_{к.сек} \quad (4.2.5)$$

где  $N_{км1}$  – число каналов трафика в одном радиоканале, определяемое стандартом радиодоступа ( $N_{м1} = 1...3$ ); для сети 3 G выберем  $N_{км1} = 1$ .



$$N_{\text{кт.сек}} = 1 \cdot 28 \approx 28 \text{ канала}$$

Согласно с моделью Эрланга, представленной повторяющий вид графика на рисунке 3.1, определим возможную нагрузку в секторе одной соты Асек при возможном значении вероятности блокировки одинаковой 1% и рассчитанным выше значением  $N_{\text{кт.сек}}$ . Определим, что  $A_{\text{сек}} = 50 \text{ Эрл}$ .



Рисунок 4.2 - Зависимость допустимой нагрузки в секторе от числа каналов трафика и вероятности блокировки

Количество пользователей, которые будут обслуживаться одной eNB, определяется по данной формуле:

$$N_{\text{аб.eNB}} = M_{\text{сек}} \cdot \left( \frac{A_{\text{сек}}}{A_1} \right) \quad (4.2.6)$$

где  $A_1$  – центральная по всем видам трафика абонентская перегрузка от 1-го абонента; значение  $A_1$  имеет возможность сочинять (0,04...0,2) Эрл. Т.к. проектируемая сеть намечается употребляться для скоростного размена информацией, то значение  $A_1$  примем одинаковым 0,2 Эрл. Следовательно:

$$N_{аб.eNB} = 3 \cdot \left( \frac{30}{0,2} \right) = 450 \text{ абонентов}$$

Число базовых станций eNB в проектируемой сети 3 G найдем по формуле:

$$N_{eNB} = \left( \frac{N_{аб}}{N_{аб.eNB}} \right) + 1 \quad (4.2.7)$$

где  $N_{аб}$  – количество потенциальных абонентов. Количество потенциальных абонентов определим как 10% от общего числа жителей. Общее число жителей мкр. Саина, города Актау, по данным последней переписи населения 2010 года составляет примерно 20000 человек. Таким образом, количество потенциальных абонентов составит 2000 человек, тогда:

$$N_{eNB} = \left( \frac{2000}{450} \right) + 1 \approx 5,4 = 5 \text{ eNB}$$

Среднюю планируемую пропускную способность  $R_N$  для сети, которую проектируем определим путем умножения количества eNB на среднюю пропускную способность eNB. Формула примет вид:

$$R_N = (R_{eNB.DL} + R_{eNB.UL}) \cdot N_{eNB} \quad (4.2.8)$$

$$R_N = (30,72 + 7,5) \cdot 5,4 = 206,388 \text{ (Мбит/с)}.$$

Далее дадим проверочную оценку емкости проектируемой сети и сравним с рассчитанной. Определим усредненный трафик одного абонента в ЧНН:

$$R_{m.ЧНН} = \frac{T_m \cdot q}{N_{ЧНН} \cdot N_d} \quad (4.2.9)$$

где  $T_T$ – средний трафик одного абонента в месяц,  $T_T = 30$  Гбайт/мес;  
 $q$ – Коэффициент для городской местности,  $q = 2$ ;  
 $N_{ЧНН}$ – число ЧНН в день,  $N_{ЧНН} = 7$ ;

$N_d$  – число дней в месяце,  $N_d = 30$ .

$$R_{m.чнн} = \frac{30 \cdot 2}{7 \cdot 30} \approx 0,28 \frac{\text{Мбит}}{\text{с}}$$

Определим общий трафик проектируемой сети в ЧНН  $R_{общ./чнн}$  по формуле:

$$R_{общ./чнн} = R_{m.чнн} \cdot N_{акт.аб}, \quad (4.2.10)$$

где  $N_{акт.аб}$  – число активных абонентов в сети; определим число активных абонентов в сети как 70% от общего числа потенциальных абонентов  $N_{аб}$ , то есть  $N_{акт.аб} = 1400$  абонентов.

$$R_{общ./чнн} = 0,28 \cdot 1400 = 392 \frac{\text{Мбит}}{\text{с}}$$

Таким образом,  $R_N > R_{общ./чнн}$ . Это условие показывает, что проектируемая сеть не будет подвергаться перегрузкам в ЧНН.

## **5. Безопасность и охрана труда**

### **5.1 Анализ опасных и вредных производственных факторов при работе с вычислительной техникой**

На сегодняшний день положение дел, связанных с обеспечением благоприятных условий труда сотрудников ВЦ не удовлетворяют современные требования. Практически каждый день они сталкиваются негативным воздействием таких физиологически опасных и вредных производственных факторов, как повышенный уровень шума, повышенная температура внешней среды, недостаточная освещенность рабочей зоны, электрический ток, статическое электричество и другое.

Кроме физических факторов на работников ВЦ влияют психофизические параметры, которые проявляются в умственном, зрительном и слуховом перенапряжении, эмоциональные перегрузки, а также монотонность и статичность труда.

Все это перечисленное неблагоприятно воздействует на человека, приводя к понижению работоспособности, прогрессирующему утомлению, понижению производительности труда, к ухудшению слуха и всего состояния

организма. Длительная работа, перегрузка, а также воздействие перечисленных факторов окружающей среды в большинстве своем случае приводит к профессиональным заболеваниям.

Анализ травматизма среди работников ВЦ указывает, что в основном несчастные случаи происходят от воздействия физически опасных производственных причин при исполнении работниками нехарактерных им работ, а в частности при воздействии электрического тока.

### **5.1.1 Размещение и безопасная эксплуатация технологического оборудования**

При организации рабочего места и размещения технологического оборудования наиболее важным направлением многими специалистами рассматривают эргономические факторы. Среди таковых большое внимание уделяют на соответствие всех элементов рабочей зоны антропометрическим, физическим и психологическим параметрам работника. При этом, рабочее место программиста должны быть наилучшим образом проработано с учетом всех необходимых направлений – от размещения необходимого оборудования до расчетных данных рабочей зоны, которое позволяет работнику совершать периодические необходимые движения и перемещения.

При проектировании рабочих мест основными параметрами являются характеристики помещения, досягаемость предметов и параметры самого оборудования, мебели и пространства. Наиболее важными среди представленных эргономически обустроенного рабочего места являются расположение предметов, документов и средств труда, характеристики рабочего стола и кресла, регулируемость элементов рабочего места и др.

Главными элементами рабочего места программиста являются стол и кресло, благодаря чему он в положении сидя совершает основную свою деятельность.

Размещение предметов, средств труда необходимо располагать в зоне легкой досягаемости рабочего пространства, а непосредственно в зоне досягаемости рук. Максимальной зоной досягаемости рук называют определенную часть моторного поля рабочей зоны, которая описывается дугами в следствии движения плечевых суставов на уровне вытянутых рук. В свою очередь моторным полем определяют то пространство рабочего места, в котором осуществляются основные действия человека.

В этой связи выделяют оптимальную зону, которая определяется частью моторного поля, которая описывается дугами вследствие движения локтевых суставов с опорой в точке локтя без напряжения и описываются предплечьями, когда плечо относительно неподвижно.

Основные максимальные и оптимальные зоны досягаемости рук рабочей зоны программиста представлены на рисунке 5.1.

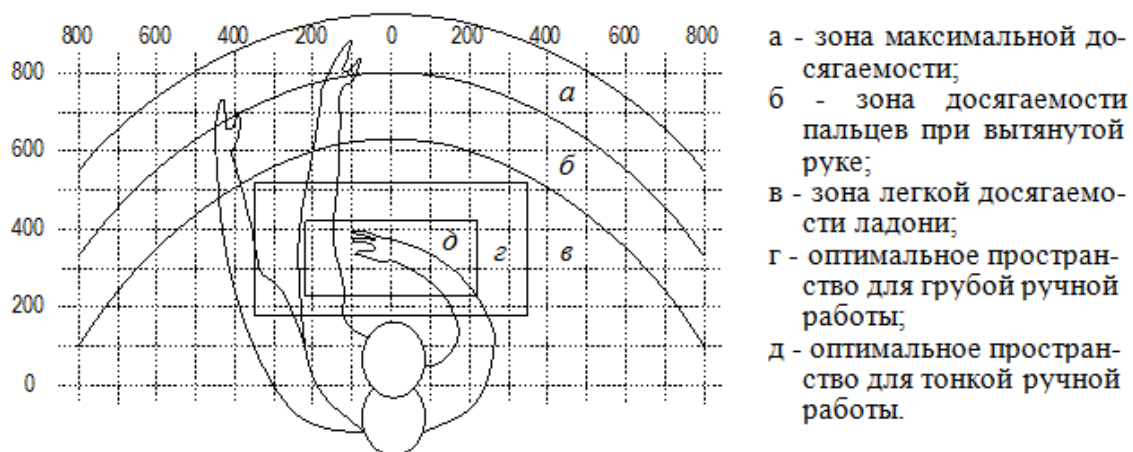


Рисунок 5.1 – Зоны досягаемости рук работника при работе в положении сидя (горизонтальная проекция).

Как видно из рисунка, предметы, средства и документации при работе программиста необходимо размещать максимально удобно для оптимальной производительности труда. Так, в центре рабочей зоны надо расположить основной предмет – дисплей компьютера, для системного блока в современных столах выделяют пространство в нише стола. Однако, необходимо отметить, что современные ЭВМ разработаны таким образом, что в самом экранном элементе располагают системные составляющие самого электронного оборудования. Клавиатура в большинстве случаев расположена в оптимальной зоне работы рук, а именно кисти. Здесь же располагают мышку, чтобы быстро и оперативно совершать удобные маневры. Чуть дальше в зоне досягаемости рук можно разместить сканер, принтер, копимашину или универсальное устройство, которое сочетает в себе все эти функции – печать, копию и сканирование.

Следом наиболее важным элементом рабочей зоны является расположение необходимой документации, к которым относятся литература, личные документы, чистые бумаги формата А4 и др. их следует располагать в выдвижных ящиках стола, то есть также рядом в зоне легкой досягаемости ладони.

Примером графического расположения основных и периферийных составляющих персонального компьютера сотрудника может служить рисунок 2, представленный ниже.

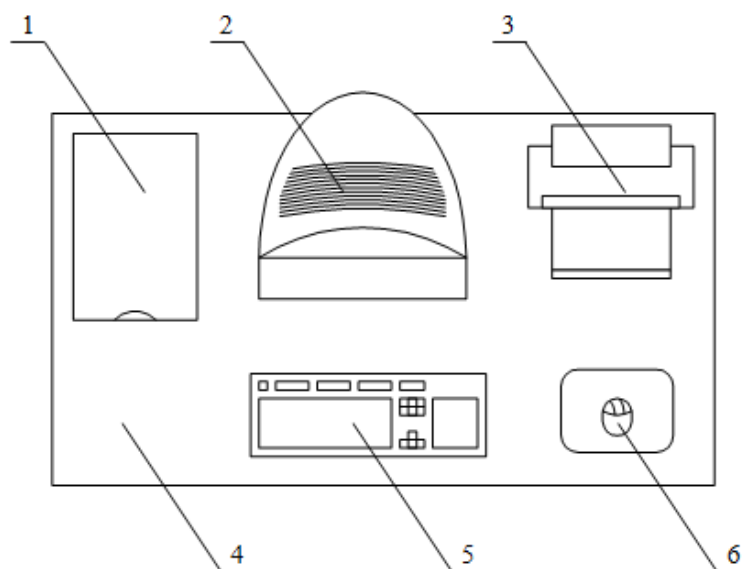


Рисунок 5.2– Размещение предметов при работе работника с компьютером и его составляющими элементами:

1 – сканер, 2 – монитор, 3 – принтер, 4 – поверхность рабочего стола,  
5 – клавиатура, 6 – манипулятор типа «мышь»

Для более результативной и комфортной работы необходимо, чтобы мебель в рабочей зоне удовлетворяла основным требованиям. В первую очередь, при выборе стола необходимо большое внимание уделить на его форму, наличие приятных углов, с приятной матовой поверхностью, чтобы отражение не мешало работать за компьютером. Сидя за ним, должно быть комфортно и удобно работать. Высота нижней части стола должна быть не сильно низкая, чтобы сидящему не нужно было поджимать ноги, ящики в столе не травмировали ноги и выдвигались удобно. Клавиатуру можно располагать ниже основной поверхности стола. Высота рабочей поверхности должна быть в пределах 680-760мм.

Что касается кресла, то это также важная составляющая рабочей зоны. Его высота сиденья над уровнем пола должна находиться в пределах 420-550 мм.

Положение экрана при работе также следует предусматривать по эргономическим параметрам. Так, расстояние считывания должно находиться в пределах 0,6...0,7м; должна предусматриваться возможность регулирования экрана. Регулировка яркости экрана и масштабов значков также влияет не только на зрение, но и производительность самого персонала. Если расстояние от глаз оператора до экрана дисплея составляет 60...80 см, то высота знака должна быть не менее 3мм, оптимальное соотношение ширины и высоты знака составляет 3:4, а расстояние между знаками – 15...20% их высоты. Соотношение яркости фона экрана и символов – от 1:2 до 1:15. Многими медицинскими работниками установлено, что располагая монитор на расстоянии 50-60см от глаз, это оптимальное решение для работы за

компьютером длительное время. Причем, верхняя часть дисплея должна быть на уровне глаз или чуть ниже.

При работе на столе за компьютером следует помнить о маленьких правилах – голову наклонять менее, чем на  $20^0$ , плечи держать в расслабленном состоянии, предплечья и кисти располагать в горизонтальном положении. Без соблюдения таких мелочей у работника возникают неприятные ощущения, боли в мышцах, суставах и сухожилиях.

## **5.2 Обеспечение электробезопасности**

К основным причинам производственного травматизма наиболее часто относят электрический ток, который представляет собой скрытый тип опасности. Существует определенное нормирование воздействия электрического тока на организм человека. Принято считать, что смертельно опасным для жизни человека является ток, величина которого превышает 0,05А, тогда как ток менее 0,05А считается безопасным. Наиболее важно предусмотреть допуск лиц, которые прошли инструктаж и хорошо ознакомлены с правилами по технике электробезопасности.

Практически все оборудования, которое включает в себя ЭВМ характеризуется применением электрического тока и воздействием его на организм человека. Электрические установки представляют для человека большую потенциальную опасность, так как в процессе эксплуатации или проведении профилактических работ человек может коснуться частей, находящихся под напряжением. Среди них специфическую опасность электроустановок представляют собой токоведущие проводники, корпуса стоек ЭВМ и прочее оборудование, которое находится под напряжением в результате повреждения изоляции. Необходимо отметить, что реакция человека на электрический ток несколько запоздалая, т.к. тело человека отвечает реакцией после прохождения тока через него спустя какое-то время. В аспекте предотвращения электро травматизма важное значение имеет не только правильная организация обслуживания действующих электроустановок ВЦ, но и своевременное проведение ремонтных, монтажных и профилактических работ.

В ВЦ разрядные токи статического электричества чаще всего возникают при прикосновении к любому из элементов ЭВМ. Такие разряды опасности для человека не представляют, но кроме неприятных ощущений они могут привести к выходу из строя ЭВМ. Для снижения величины возникающих зарядов статического электричества в ВЦ покрытие технологических полов следует выполнять из однослойного поливинилхлоридного антистатического линолеума. Другим методом защиты является нейтрализация заряда статического электричества ионизированным газом. В промышленности широко применяются радиоактивные нитрализаторы. К общим мерам защиты от статического электричества в ВЦ можно отнести общие и местное увлажнение воздуха.



### 5.2.1 Расчет вентиляции вычислительного центра

Одним из необходимых метеорологических условий среды при длительной работе за компьютером является температурный режим, который устанавливается благодаря отоплению в зимнее время и кондиционированию или вентиляции в теплое время года. Системы регулирования температуры в рабочем помещении следует располагать высоко под потолками, поток воздуха регулировать таким образом, чтобы прямое направление воздуха (теплого или прохладного) избегать от непосредственной работы сотрудников.

В этом отношении в системы удаления загрязненного воздуха и поступления свежего выделяют естественную, искусственную и смешанную вентиляции. В этом отношении основным параметром, определяющим характеристики вентиляционной системы, является кратность обмена, т.е. сколько раз в час сменится воздух в помещении.

Расчет кратности произведения вентиляции для вычислительного центра представлен ниже:

$V_{\text{вент}}$  – объем воздуха, необходимый для обмена;

$V_{\text{пом}}$  – объем рабочего помещения.

Для расчета примем следующие размеры рабочего помещения:

–длина  $B = 6$  м;

–ширина  $A = 5$  м;

–высота  $H = 3.5$  м.

Соответственно объем помещения равен:

$$V_{\text{помещения}} = A * B * H = 105 \text{ м}^3 \quad (5.3.1)$$

Необходимый для обмена объем воздуха  $V_{\text{вент}}$  определим исходя из уравнения теплового баланса:

$$V_{\text{вент}} * C (t_{\text{уход}} - t_{\text{приход}}) * Y = 3600 * Q_{\text{избыт}}, \quad (5.3.2)$$

где  $Q_{\text{избыт}}$  – избыточная теплота (Вт);

$C = 1000$  – удельная теплопроводность воздуха (Дж/кгК);

$Y = 1.2$  – плотность воздуха (мг/см).

Температура уходящего воздуха определяется по формуле:

$$t_{\text{уход}} = t_{\text{р.м.}} + (H - 2) * t, \quad (5.3.3)$$

где  $t = 1-5$  градусов – превышение  $t$  на 1м высоты помещения;

$t_{\text{р.м.}} = 25$  градусов – температура на рабочем месте;

$H = 3.5$  м – высота помещения;

$t_{\text{приход}} = 18$  градусов.

$$t_{\text{уход}} = 25 + (3.5 - 2) * 2 = 28$$

$$Q_{\text{избыт}} = Q_{\text{избыт1}} + Q_{\text{избыт2}} + Q_{\text{избыт3}}, \quad (5.3.4)$$

где  $Q_{\text{избыт}}$  – избыток тепла от электрооборудования и освещения.

$$W Q_{\text{избыт}} = E * p, \quad (5.3.5)$$

где  $E$  – коэффициент потерь электроэнергии на теплоотвод ( $E = 0.55$  для освещения);

$$p – \text{мощность, } p = 40 \text{ Вт} * 15 = 600 \text{ Вт}$$

$$Q_{\text{избыт1}} = 0.55 * 600 = 330 \text{ Вт}$$

где  $Q_{\text{избыт2}}$  – теплопоступление от солнечной радиации,

$$Q_{\text{избыт2}} = m * S * k * Q_c, \quad (5.3.6)$$

где  $m$  – число окон, примем  $m = 4$ ;

$S$  – площадь окна,  $S = 2.3 * 2 = 4.6 \text{ м}^2$ ;

$k$  – коэффициент, учитывающий остекление. Для двойного остекления

$$k = 0.6;$$

$Q_c = 127 \text{ Вт/м}$  – теплопоступление от окон.

$$Q_{\text{избыт.2}} = 4.6 * 4 * 0.6 * 127 = 701 \text{ Вт}$$

где  $Q_{\text{избыт3}}$  – тепловыделения людей.

$$Q_{\text{избыт3}} = n * q, \quad (5.3.7)$$

где  $q = 80 \text{ Вт/чел.}$ ,  $n$  – число людей, например,  $n = 2$

$$Q_{\text{избыт3}} = 2 * 80 = 160 \text{ Вт}$$

$$Q_{\text{избыт}} = 330 + 701 + 160 = 1191 \text{ Вт}$$

Из уравнения теплового баланса следует:

$$V_{\text{вент}} = \frac{3600 * 1191}{1000 * (28 - 18)} = 428,76 \text{ м}^3$$

Для создания благоприятного микроклимата в помещении, кроме вентиляции и отопления, предусматривают систему кондиционирования, где

также рассчитываются физические параметра атмосферного воздуха – влажность, температура, скорость движения воздуха, его направление и др.

### **5.2.2 Мероприятия по борьбе с шумом и вибрации**

Шум ухудшает условия труда, оказывая вредное действие на организм человека. Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах и т. д. Под воздействием шума снижается концентрация внимания, нарушаются физиологические функции, появляется усталость в связи с повышенными энергетическими затратами и нервно-психическим напряжением, ухудшается речевая коммутация. Все это снижает работоспособность человека и его производительность, качество и безопасность труда. Длительное воздействие интенсивного шума (выше 80 дБ(А)) на слух человека приводит к его частичной или полной потере.

Уровень шума на рабочем месте математиков-программистов и операторов видеоматериалов не должен превышать 50 дБА, а в залах обработки информации на вычислительных машинах - 65 дБА. Для снижения уровня шума стены и потолок помещений, где установлены компьютеры, могут быть облицованы звукопоглощающими материалами. Уровень вибрации в помещениях вычислительных центров может быть снижен путем установки оборудования на специальные виброизоляторы.

### **5.2.4 Обеспечение нормативной освещенности**

Помещение компьютерного зала должно иметь естественное и искусственное освещение. Неблагоприятные условия освещения затрудняют работу за видеотерминалами и способствуют формированию отклонений в функции зрения.

Рациональное освещение рабочего места является одним из важнейших факторов, влияющих на эффективность трудовой деятельности человека, предупреждающих травматизм и профессиональные заболевания. Правильно организованное освещение создает благоприятные условия труда, повышает работоспособность и производительность труда. Утомляемость органов зрения зависит от ряда причин:

- недостаточность освещенности;
- чрезмерная освещенность;
- неправильное направление света.

Недостаточность освещения приводит к напряжению зрения, ослабляет внимание, приводит к наступлению преждевременной утомленности. Чрезмерно яркое освещение вызывает ослепление, раздражение и резь в глазах. Неправильное направление света на рабочем месте может создавать резкие тени, блики, дезориентировать работающего. Все эти причины могут привести к несчастному случаю или профзаболеваниям, поэтому столь важен правильный расчет освещенности.

Расчет освещенности ВЦ сводится к выбору системы освещения, определению необходимого числа светильников, их типа и размещения.

Искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп. Будем использовать люминесцентные лампы, которые по сравнению с лампами накаливания имеют существенные преимущества:

- по спектральному составу света они близки к дневному, естественному освещению;

- обладают более высоким КПД (в 1.5-2 раза выше, чем КПД ламп накаливания);

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

- более длительный срок службы.

Расчет освещения производится для комнаты площадью 30 м<sup>2</sup>, ширина которой 5 м, высота – 3,5 м. Воспользуемся методом светового потока.

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

$$F = \frac{E * K * S * Z}{n}, \quad (5.4.1)$$

где F – рассчитываемый световой поток, Лм;

E – нормированная минимальная освещенность, Лк (определяется по таблице). Работу программиста, в соответствии с этой таблицей, можно отнести к разряду точных работ, следовательно, минимальная освещенность будет E = 400 Лк;

S – площадь освещаемого помещения (в нашем случае S<sub>ВЦ</sub> = 30 м<sup>2</sup>);

Z – отношение средней освещенности к минимальной (обычно принимается равным 1.1-1.2, пусть Z = 1.1);

K – коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение определяется по таблице коэффициентов запаса для различных помещений и в нашем случае K = 1.5);

n – коэффициент использования, (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризующихся коэффициентами отражения от стен (Рс) и потолка (Рп)), значение коэффициентов Рс и Рп определим по таблице зависимостей коэффициентов отражения от характера поверхности: Рс=30%, Рп=50%.

Значение n определим по таблице коэффициентов использования различных светильников. Для этого вычислим индекс помещения по формуле:

$$I = \frac{S}{h * (A + B)}, \quad (5.4.2)$$

где  $S$  – площадь помещения,  $S = 30 \text{ м}^2$ ;  
 $h$  – расчетная высота подвеса,  $h = 3.2 \text{ м}$ ;  
 $A$  – ширина помещения,  $A = 5 \text{ м}$ ;  
 $B$  – длина помещения,  $B = 6 \text{ м}$ .

Подставив значения получим:

$$I = \frac{30}{3.2 * (5 + 6)} = 0.85$$

Мощность осветительной установки  $P$  определяется из выражения:

$$P = n * N * P_i, \quad (5.4.3)$$

где  $P_i$  - потребляемая мощность одной лампы, кВт.

Обычно для светлых помещений компьютерных залов  $r_{\text{п}} = 70\%$ ,  $r_{\text{с}} = 50\%$ ,  
 $r_{\text{р}} = 30\%$ .

Зная индекс помещения  $I$ ,  $P_{\text{с}}$  и  $P_{\text{п}}$ , находим  $n = 0.32$

Подставим все значения в формулу для определения светового потока  $F$ :

$$F = \frac{400 * 1.5 * 30 * 1.1}{0.32} = 61875 \text{ Лм}$$

Для освещения выбираем люминесцентные лампы типа ЛБ40-1, световой поток которых  $F = 4220 \text{ Лк}$ .

Рассчитаем необходимое количество ламп по формуле:

$$N = F / F_{\text{л}}, \quad (5.4.4)$$

где  $N$  – определяемое число ламп;

$F$  – световой поток,  $F = 63642,857 \text{ Лм}$ ;

$F_{\text{л}}$  - световой поток лампы,  $F_{\text{л}} = 4220 \text{ Лм}$ .

$$N = \frac{61875}{4220} = 14.66 \approx 15 \text{ шт.}$$

Мощность осветительной установки:

$$P = 2 * 12 * 0,04 = 1,2 \text{ кВт.}$$

При выборе осветительных приборов используем светильники типа ОД. Каждый светильник комплектуется двумя лампами. Размещаются светильники двумя рядами, по четыре в каждом ряду.

### **5.3. Общие правила пожарной безопасности**

Пожарная безопасность в компьютерном зале организуется в соответствии с «СНиП РК 2.02-05-2002 пожарная безопасность зданий и сооружений» вычислительный центр должен быть оборудован средствами пожаротушения.

Для эффективного пожаротушения компьютерного зала следует иметь минимум средств такие как:

- порошковые и углекислотные огнетушители (не менее двух), размещаемые непосредственно в помещении зала и лаборантской или в непосредственной близости от них;

- специальная коробка с крышкой с сухим песком объемом  $0,05 \text{ м}^3$ , с наличием совка и не менее 2 кг песка. Разрешается так же хранить песок в сосудах вместимостью 4-6 кг;

- накладки из огнеупорной ткани размером  $1,2 \times 1,8 \text{ м}$  и  $0,5 \times 0,5 \text{ м}$ .

- возгорание в зале с компьютерами следует сразу ликвидировать, в начале отключить подачу тока, при этом

- электропроводку следует гасить песком, огнезащитной тканью, порошковыми огнетушителями;

- обесточенную аппаратуру можно накрывать огнезащитной тканью, гасить порошковыми огнетушителями;

- Обесточенную проводку можно тушить водой.

#### **5.3.1 Пожарная безопасность в ВЦ**

Все виды кабелей от трансформаторных подстанций и двигатель-генераторных агрегатов прокладывают в металлических газовых трубах вплоть до распределительных щитов и стоек питания. В машинных залах кабельные линии прокладывают под технологическими съемными полами, которые выполняют из негорючих или трудно горючих материалов с пределом огнестойкости не менее 0,5 ч.

Подпольные пространства под съемными полами разделяют негоряемыми перегородками с пределом огнестойкости не менее 0,75 ч на отсеки площадью не более  $250 \text{ м}^2$ , коммуникации через которые прокладывают в специальных обоях с применением негорючих уплотняющих материалов.

Входы в вычислительный центр делают через тамбуры-шлюзы, оборудованные самозакрывающимися двупольными дверями со специальным уплотнением. Двери тамбура-шлюза должны открываться в сторону машинного зала, который всегда находится под избыточным давлением воздуха. Ширина дверей должна быть не менее 1,5 м, высота – не менее 2 м, ширина коридоров – не менее 1,8 м. Из машинных залов площадью более  $250 \text{ м}^2$  предусматривается не менее двух выходов. На эвакуационных путях устраивают как естественное, так и искусственное аварийное освещение.

Для сигнализации пожара устанавливают тепловую пожарную установку СТПУ-1, срабатывающую автоматически при повышении температуры или конденсации дыма с подачей сообщений с помощью световых и акустических сигналов. Время срабатывания СТПУ-1 не более 15 секунд.

Наиболее эффективным средством пожарной защиты вычислительной техники являются стационарные автоматические огнетушащие установки с использованием углекислоты и фреона. Наиболее подходящим для пожаротушения веществом является фреон 114В2, или тетрафтордибромэтан ( $C_2F_4Br_2$ ) – бесцветная жидкость с температурой кипения  $46,4^{\circ}C$  и температурой самовоспламенения  $580^{\circ}C$ . Из 1 литра жидкости образует 254 литра пара, который в 8,96 раза тяжелее воздуха. По огнетушащей способности фреон в 10 раз превосходит углекислый газ. Расчётное время тушения фреоном примерно равно 60 секунд, водой и пеной 10-15 минут. Пары фреона после пожара удаляются приточно-вытяжной вентиляцией.

### 5.3.2 Расчёт количества огнетушащего состава

При тушении пожара в вычислительном центре лучше всего использовать автоматическую установку газового пожаротушения с электрическим пуском типа БАЭ и расчётным запасом огнетушащего состава фреона 114В2.

Количество гасящего вещества (фреона) определяется по формуле (5.5.1):

$$G_t = G_d W_n K_y, \quad (5.5.1)$$

где  $G_t$  – количество гасящего вещества;

$G_d$  – концентрация газового состава;

$W_n$  – расчётный объём защищаемого помещения;

$K_y$  – коэффициент, учитывающий особенности процесса газообмена в защищаемом помещении.

Для ВЦ  $K_y = 1,2$ ;

$G_d = 0,25$  кг/м для тетрафтордибромэтана (фреона).

Объём защищаемого помещения вычисляется из произведения площади и высоты, где площадь помещения  $30 \text{ м}^2$ , а высота 3,5 м. получаем объём  $105 \text{ м}^3$ .

Тогда

$$G_t = 0,25 * 105 * 1,2 = 31,5 \text{ кг.}$$

Необходимое количество баллонов определяется по формуле (5.5.3):

$$N_b = G_t / (V_b * \rho * a), \quad (5.5.2)$$

где  $V_b$  – объём баллона, равный 40 литров;

$\rho$  – плотность фреона  $2,17 \text{ кг/л}$ ;

$a$  – коэффициент наполнения баллона, равный 0,9.



Следовательно

$$N_b = 31,5 / (40 \cdot 2,17 \cdot 0,9) = 0,40,$$

то есть необходим 1 баллон с фреоном.

Объем воздушных баллонов можно вычислить по формуле (5.5.3):

$$W_b = (P_{cmin} + 1)(W_c + W_R) / (P_{max} + P_{dmin}), \quad (5.5.3)$$

где  $P_{cmin}$  – конечное давление в сосуде с гасящим составом;

$P_{dmin}$  – конечное давление в воздушных баллонах;

$P_{max}$  – начальное давление воздуха в баллоне;

$W_c + W_R$  – объем сосудов с гасящим составом и трубопроводов.

$$W_b = (5 + 1)(40 + 5) / (150 + 5) = 1,74 \text{ литра.}$$

Для сжатого воздуха можно использовать 25 литровый баллон.

#### **5.4 Мероприятия по защите объекта при ЧС техногенного и природного характера**

Обычно любая угроза землетрясения заранее оповещается с помощью радио и главное не нужно паниковать, действовать быстро, уверенно, спокойно, без паники.

Во время землетрясения или при непосредственной угрозе его возникновения по распоряжению администрации предприятия или учреждения все работы прекращаются, производится безаварийная остановка производства, там, где это возможно, отключается электрический ток, снижается давление газа, кислорода, пара, воды, рабочие и служащие организовано, под руководством начальников цехов и отделов занимают безопасные места, личный состав формирований гражданской обороны направляется в районы сбора.

Как только почувствуете толчок землетрясения отключите свет, газ и воду. При не сильном землетрясении можно оставаться там где ты находишься. При сильном землетрясении нужно начать покидать помещение если находишься на первом этаже, если выше второго этажа не выходите из помещения. Встаньте в безопасном месте у внутренней стены, в углу, в дверном проеме, у опорной колонны, лягте в ванну.

Залезьте под кровать или стол – они защитят вас от падающих предметов и обломков. Держитесь подальше от окон и тяжелой мебели. Не пользуйтесь лифтом если вы находитесь на улице, отойдите на открытое место подальше от зданий и линий электропередач, не подходите к оборванным электрическим проводам. Не бегайте вдоль зданий и не входите в них. Если вы находитесь в автомобиле, оставайтесь на открытом месте, не

покидая автомобиль, пока толчки не прекратятся. Помните: во время землетрясения очень редко причиной человеческих жертв бывает движение почвы. Главными причинами несчастных случаев при землетрясении являются:

- обрушение отдельных частей зданий;
- оборванные электропровода;
- падение тяжелых предметов в квартире;
- неконтролируемое поведение людей при панике.

После землетрясения:

Окажите первую медицинскую помощь нуждающимся. Освободите пострадавших из завалов, если для этого не требуется дополнительного снаряжения. Держите включенным радио, слушайте сигналы служб. Не подходите к явно поврежденным зданиям и не входите в них. Будьте готовы к повторным толчкам, наиболее опасными считаются в этом отношении первые два-три часа после землетрясения, хотя они могут случиться и через несколько суток, недель и даже месяцев. Чем больше времени прошло со времени первого землетрясения, тем меньше вероятность повторных толчков. Синдром длительного сдавливания, так называемый крэш-синдром – одна из самых опасных травм. При длительном воздействии тяжелых предметов: обломков строительных конструкций, стволов упавших деревьев, массивной мебели и тому подобного – на руку или ногу человека, кровоток в конечности прекращается. Это ведет к нарушению обмена веществ в организме человека. Последствия крэш-синдрома нередко связаны с ампутацией. Определив, что у пострадавшего есть признаки проявления синдрома длительного сдавливания, осмотрите пострадавшего; дайте обезболивающее средство; выше места сдавливания наложите жгут; освободив пострадавшего, согрейте его. Помните, нельзя терять ни минуты. Срочно доставьте человека в больницу.

Обрушение здания. Полное или частичное внезапное обрушение здания –это чрезвычайная ситуация, возникающая по причине ошибок, допущенных при проектировании здания, отступлении от проекта при ведении строительных работ, нарушении правил монтажа, при вводе в эксплуатацию здания или отдельных его частей с крупными недоделками, при нарушении правил эксплуатации здания, а также вследствие природной или техногенной чрезвычайной ситуации.

Как действовать при внезапном обрушении здания. Услышав взрыв или обнаружив, что здание теряет свою устойчивость, постарайтесь как можно быстрее покинуть его, взяв документы, деньги и предметы первой необходимости. Покидая помещение, спускайтесь по лестнице, не пользуйтесь лифтом, так как он в любой момент может выйти из строя. Пресекайте панику, давку в дверях при эвакуации, останавливайте тех, кто собирается прыгать с балконов и окон из этажей выше первого, а также через застекленные окна. Оказавшись на улице, не стойте вблизи зданий, перейдите на открытое пространство.

Как действовать в завале. Дышите глубоко, не поддавайтесь панике, сосредоточьтесь на самом важном, пытайтесь выжить любой ценой, верьте, что помощь придет обязательно. По возможности окажите себе первую медицинскую помощь. Попытайтесь приспособиться к обстановке и осмотреться, поискать возможный выход. Постарайтесь определить, где вы находитесь, нет ли рядом других людей: прислушайтесь, подайте голос.

## **6. Расчет показателей экономической эффективности проекта**

### **6.1 Расчёт затрат на разработку и внедрение Android-приложения**

В этом разделе рассматривается расчёт затрат на разработку Android-приложения.

Затраты на разработку и Android-приложения определяют по следующей формуле:

$$C = C_{\text{кТС}} + C_{\text{ТЗ}} + C_{\text{разр}} + C_{\text{отл}}, \quad (6.1)$$

где  $C_{\text{кТС}}$  – затраты на приобретение комплекса технических средств;  
 $C_{\text{разр}}$  – затраты на разработку и создание приложения;  
 $C_{\text{отл}}$  – затраты на написание и отладку программы;  
 $C_{\text{ТЗ}}$  – затраты на разработку технического задания.

### **6.2 Расчет затрат на комплекс технических средств**

Для внедрения Android-приложения нам потребуется один ноутбук и один смартфон.

Затраты на комплекс технических средств составляют:

1. При использовании ноутбука и смартфона на платформе Android

$$C_{\text{кТС}} = C_{\text{ноут}} + C_{\text{см}} \quad (6.2)$$

где  $C_{\text{ноут}}$  – стоимость ноутбука;  
 $C_{\text{см}}$  – стоимость смартфона на платформе Android.

Стоимость ноутбука LenovoZ510 Intel(R) Core(TM) i5-4200M CPU 2.50 GHz 6 ГБ ОЗУ

$$C_{\text{ноут}} = 129000 \text{ (тенге)}$$

Стоимость смартфона LenovoS930

$$C_{\text{см}} = 78000 \text{ (тенге)}$$

$$C_{\text{кТС}} = 129000 + 78000 = 207000 \text{ (тенге)}$$

### **6.3 Расчёт стоимости часа машинного времени.**

Стоимость часа машинного времени рассчитывается по следующей формуле

$$c_{\text{чмв}} = (A_o + C_{\text{эл.э}} + (ЗП_{\text{опер}} + ОТ_{\text{опер}}) + C_{\text{накл}}) / \Phi_{\text{р.вр}}, \quad (6.3)$$

где  $A_o$  – амортизационные отчисления на комплекс технических средств в год, 25% от  $C_{\text{кТС}}$ ;

$C_{\text{эл.э}}$  – затраты на электроэнергию за год;

$ЗП_{\text{опер}}$  – годовая заработная плата оператора;

$ОТ_{\text{опер}}$  – отчисления на социальные нужды 11% за год;

$C_{\text{накл}}$  – накладные расходы в год, 10% от  $C_{\text{кТС}}$ ;

$\Phi_{\text{р.вр}}$  – годовой фонд рабочего времени в часах.

Вычисляем:

1) амортизационные отчисления на комплекс технических средств за год.

$$A_o = C_{\text{кТС}} \cdot 0,25 \quad (6.4)$$

$$A_o = 207000 \cdot 0,25 = 51750 \text{ (тенге)}$$

2) затраты на электроэнергию за год рассчитываются по формуле:

$$C_{\text{эл.э}} = W \cdot T \cdot S, \quad (6.5)$$

где  $W$  – потребляемая мощность;

$T$  – время работы в часах за год;

$S$  – тариф.

Для ноутбука и смартфона потребляемая мощность 236Вт:

$$W = 0,236 \text{ кВт},$$

$$T = Ч \cdot Д,$$

где  $Ч$  – время работы ПК в день (8 часов);

$Д$  – количество рабочих дней в году.

$$T = 8 \cdot 252 = 2016 \text{ часа}, \quad S(1\text{кВтч}) = 14,36 \text{ (тенге)}$$

$$C_{\text{эл.э}} = 0,236 \cdot 14,36 \cdot 8 \cdot 252 = 6833 \text{ (тенге)}$$

3) годовая заработная плата оператора, оклад которого установлен в размере 75000 тенге в месяц;

$$ЗП_{\text{опер}} = ЗП \cdot 12$$

$$ЗП_{\text{опер}} = 75000 \cdot 12 = 900000 \text{ (тенге)}$$

4) отчисления на социальные нужды 11% за год:

$$От_{\text{опер}} = ЗП_{\text{опер}} \cdot 0,11$$

$$От_{\text{опер}} = 900000 \cdot 0,11 = 99000 \text{ (тенге)}$$

5) накладные расходы в год, 10% от  $C_{\text{кТС}}$ :

$$C_{\text{накл}} = C_{\text{кТС}} \cdot 0,10$$

$$C_{\text{накл}} = 207000 \cdot 0,10 = 20700 \text{ (тенге)}$$

6) годовой фонд рабочего времени в часах:

$$\Phi_{\text{р.вр}} = Ч \cdot Д$$

$$\Phi_{\text{р.вр}} = 8 \cdot 252 = 2016 \text{ (часов)}$$

Таким образом, стоимость часа машинного времени составляет:

$$c_{\text{чМВ}} = (51750 + 6833 + (900000 + 99000) + 20700) / 2016 = 534,86 \text{ (тенге)}$$

#### **6.4 Расчёт затрат на разработку технического задания**

Затраты на разработку технического задания:

$$C_{\text{ТЗ}} = ЗП_{\text{р}} \cdot В + От + C_{\text{МВ}}, \quad (6.6)$$

где  $ЗП_{\text{р}}$  – оклад разработчика;  
 $В$  – время, затраченное на разработку технического задания;  
 $От$  – отчисления на социальные нужды 11%;  
 $C_{\text{МВ}}$  – стоимость машинного времени.

Стоимость машинного времени, потраченного на разработку технического задания

$$C_{\text{МВ}} = c_{\text{чМВ}} \cdot t_{\text{МВ}},$$

где  $c_{\text{чМВ}}$  – стоимость часа машинного времени;  
 $t_{\text{МВ}}$  – машинное время, затраченное на разработку технического задания в часах, (0,25 месяца – 5 рабочих дней).  
 $t_{\text{МВ}} = Ч \cdot Д$

$$t_{\text{MB}} = 8 \cdot 5 = 40 \text{ (часов)}$$

$$C_{\text{MB}} = 534,86 \cdot 40 = 21394 \text{ (тенге)}$$

Техническое задание разрабатывалось программистом (оклад которого установлен в размере 75000 тенге в месяц) в течение полумесяца. Отчисления на социальные нужды составляют 11%. Далее вычисляем:

$$O_{\text{T}} = 3\Pi_{\text{p}} \cdot 11\%$$

$$O_{\text{T}} = 75000 \cdot 0,11 = 4400 \text{ (тенге)}$$

$$C_{\text{TЗ}} = 75000 \cdot 0,25 + (75000 \cdot 0,25 \cdot 0,11) + 21394 = 42207 \text{ (тенге)}$$

### 6.5 Расчёт затрат на разработку и создание приложения

Затраты на разработку и создание приложения:

$$C_{\text{разр}} = 3\Pi_{\text{p}} \cdot B + O_{\text{T}} + C_{\text{MB}}, \quad (4.7)$$

где  $3\Pi_{\text{p}}$  - оклад разработчика;  
 $B$  – время, затраченное на разработку и создание приложения;  
 $O_{\text{T}}$  - отчисления на социальные нужды 11%.  
 $C_{\text{MB}}$  – стоимость машинного времени.

Стоимость машинного времени, потраченного на разработку и создание приложения

$$C_{\text{MB}} = c_{\text{чMB}} \cdot t_{\text{MB}},$$

где  $c_{\text{чMB}}$  – стоимость часа машинного времени;  
 $t_{\text{MB}}$  – машинное время, затраченное на разработку и создание в часах, (1 месяца – 21 рабочих дней).

$$t_{\text{MB}} = \text{Ч} \cdot \text{Д}$$

$$t_{\text{MB}} = 8 \cdot 21 = 168 \text{ (часов)}$$

$$C_{\text{MB}} = 534,86 \cdot 168 = 89857 \text{ (тенге)}$$

Приложение разрабатывался и создавался программистом в течение одного месяца, оклад которого установлен в размере 75000 тенге в месяц. Отчисления на социальные нужды составляют 11%. Далее вычисляем:

$$От = ЗП_p \cdot 11\%$$

$$От = 75000 \cdot 0,11 = 8250 \text{ (тенге)}$$

$$C_{разр} = 75000 \cdot 1 + (75000 \cdot 1 \cdot 0,11) + 89857 = 173107 \text{ (тенге)}$$

## 6.6 Расчёт затрат на написание и отладку программы

Затраты на написание и отладку программы:

$$C_{отл} = ЗП_p \cdot В + От + C_{мв}, \quad (4.9)$$

где  $ЗП_p$  - оклад программиста;  
 $В$  – время, затраченное на написание и отладку программы;  
 $От$  - отчисления на социальные нужды 11%.  
 $C_{мв}$  – стоимость машинного времени.

Стоимость машинного времени, потраченного на написание и отладку программы.

$$C_{мв} = c_{чмв} \cdot t_{мв},$$

где  $c_{чмв}$  – стоимость часа машинного времени;  
 $t_{мв}$  – машинное время, затраченное на написание и отладку программы в часах, ( две недели – 10 рабочих дней).

$$t_{мв} = Ч \cdot Д$$

$$t_{мв} = 8 \cdot 10 = 80 \text{ (часов)}$$

$$C_{мв} = 534,86 \cdot 80 = 42789 \text{ (тенге)}$$

Программа была написана и отлажена программистом в течение одного месяца, оклад которого установлен в размере 75000 тенге в месяц. Отчисления на социальные нужды составляют 11%. Далее вычисляем:

$$От = ЗП_p \cdot 11\%$$

$$От = 75000 \cdot 0,11 = 8250 \text{ (тенге)}$$

$$C_{отл} = 75000 \cdot 0,5 + (75000 \cdot 0,5 \cdot 0,11) + 42789 = 84414 \text{ (тенге)}$$

Таким образом, затраты на разработку и внедрение приложения по формуле (4.1) при работе на стационарном компьютере составляют:



$$C = 207000 + 42207 + 173107 + 84414 = 506728,00 \text{ (тенге);}$$

Стоимость проекта. Приложение будет продаваться по 1 доллара, в приложении не будет рекламы.

Чтобы оправдать затраты необходимо, чтобы приложение купили 1650 пользователей, это будет 610500 тенге.

$$C_{\text{пр}} = 610500 - 506728 = 103772 \text{ (тенге).}$$

где  $C_{\text{пр}}$  – прибыль.

$$C_{\text{со}} = 103772 / 506728 = 0,2 \text{ года (6 месяцев).}$$

где  $C_{\text{со}}$  – срок окупаемости.

## **Заключение**

Программные обеспечения имеют все шансы сделать многие операционные системы готовой для работы, развлечения, выхода в Интернет, просмотра веб-страниц и т.д., что создает возможности для обычного телефона быть компактным карманным компьютером с совершенным набором многофункциональных способностей. Любое приложение может быть разработанным для многих других платформ, и персонально для каждой системы. Практика нам показывает, что возможности индивидуального ПО гораздо шире, что делает целесообразнее устанавливать соответствующее обеспечение, а следовательно и разрабатывать необходимые приложения.

В данной работе я использовал свои знания для создания программы как AUPET Player. В основном я отдаю предпочтения другим программам, с более расширенными возможностями, чем AUPET Player, для того, чтобы прослушивать музыку. Однако на данной программе довольно качественная обработка звука. И хотя для меня главным критерием в программах не наличие всевозможных оболочек, а ее работоспособность, AUPET Player все же является достаточно неплохой программой с множеством удобств для работы пользователя. По моему мнению, ни разу персональный смартфон не мог обойтись без мультимедийных приложений.

На основании выше изложенного мы можем сказать о том, что внедрение Android-приложения позволит при незначительных экономических затратах содействовать:

- обеспечению пользователям быстрого доступа к материалам для изучения языка программирования Java;
- значительной экономии времени и денежных средств на изучение языка Java .

Необходимо отметить, что разработка данного Android-приложения экономически оправдывается и окупается за счет многих факторов, таких как: реклама, стоимость приложения, актуальность, оперативность в получении информации и услуг.

## Перечень сокращений

3GPP – объединение по разработке стандартов мобильной связи 3-го поколения

Cdma 2000 – стандарт мобильной связи 3-го поколения в эволюционном развитии сетей IS – 95

eNB – базовая станция стандарта LTE

E-UTRAN – сеть радиодоступа стандарта LTE

ETSI – европейский институт телекоммуникационных технологий

FDD – дуплекс с частотным разделением направлений

GERAN – сеть радиодоступа стандарта GSM/EDGE  
GSM – глобальная система мобильной связи

HSPA – технология беспроводной широкополосной радиосвязи, использующая пакетную передачу данных в сетях WCDMA/UMTS

IMS – мультимедийная система передачи данных на основе протокола IP

MIMO – технология передачи данных с помощью N антенн и их приема M антеннами

OFDM – технология мультиплексирования с ортогональным частотным разделением каналов

SAE – архитектура ядра сети, разработанная для стандарта LTE

SC-FDMA – множественный доступ с мультиплексированием с частотным разнесением передачи на одной несущей

TDD – дуплекс с временным разделением направлений

UMTS – универсальная мобильная телекоммуникационная система

WCDMA – широкополосный множественный доступ с кодовым разделением каналов

СМР – строительно-монтажные работы ЧНН – час наибольшей нагрузки

ЦП – циклический префикс

## Список литературы

- 1 Блог на хабре о разработке под Android на сайте [http://www.habrahabr.ru/blogs/android\\_development/](http://www.habrahabr.ru/blogs/android_development/).
- 2 Официальная справка для Android разработчиков на сайте <http://www.developer.android.com/index.html>.
- 3 Википедия на сайте <http://www.wikipedia.org>.
- 4 Программирование для Android. Самоучитель / Колисниченко Д.
- 5 Android 2. Программирование приложений для планшетных компьютеров и смартфонов / Рето Майер
- 6 Статьи о программировании для Android на сайте <http://flashbot.ru/android-dev>.
- 7 Официальная справка по среде программирования [Электронный ресурс] на сайте <http://www.eclipse.org>.
- 8 Форум о программировании для Android на сайте <http://www.cyberforum.ru/android-dev>.
- 9 Программирование под Android / Блэйк Мик .
- 10 Смартфоны Android без напряжения. Руководство пользователя / Андрей Жвалевский
- 11 Программирование для Android. Самоучитель / Денис Колисниченко
- 12 “The Busy Coder's Guide to Android Development” author Mark L.Murphy
- 13 Курсы по обучению Android на сайте <https://www.coursera.org/>.
- 14 Рублев В.С., Языки логического программирования [Текст]/ В.С. Рублев - [Электронный ресурс]/ В.С. Рублев - - Интернет-университет информационных технологий. (<http://www.intuit.ru/departments/expert/logicpr/>).
- 15 Экономика связи / Под ред. док. эк.наук профессора Срапионовой О.С. и к.э.н. Болдиной В.Н., – М, Радио и связь, 1984
- 16 Производственное освещение - Мет. указания Алма-Ата-2015.
- 17 Баклашов Н.И., Китаев Н.Ж. Охрана труда на предприятиях связи и охрана окружающей среды. – Киев: Техника,2012
- 18 Системы вентиляции и кондиционирования. Теория и практика. Учебное пособие. М.: Евроклимат,2000 – 416 с
- 19 ГОСТ 12.2.032-78. «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования»;
- 20 СН 245-71. «Санитарные нормы проектирования промышленных предприятий»
- 21 СН 2.2.4/2.1.8.562-96 «Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки»
- 22 Кошулько Л.П. Производственное освещение. Методические указания - Алма-Ата, 1989-40с.
- 23 ГОСТ 12.1.004-91. Пожарная безопасность. Общие требования.- М,1992-264с.

24 СНиП РК 2.04-05-2002. Естественное и искусственное освещение. Общие требования.- М: Стройиздат, 2002.

25 Пожарная безопасность. // Электронная версия на сайте: [www.emer.gov.kz](http://www.emer.gov.kz).

26 Бабаков В. Ю., Вознюк М. А., Михайлов П. А. Сети мобильной связи. Частотно-территориальное планирование. Учебное пособие для ВУЗов. М.: Горячая линия, 2007.– 223 с.

27 Гельгор А. Л. Технология LTE мобильной передачи данных: учебное пособие. – СПб.: Издательство Политехн. ун-та, 2011. – 188 с.

28 Гольдштейн Б. С., Соколов Н. А., Яновский Г. Г. Сети связи: Учебник для ВУЗов. – СПб.: БХВ – Петербург, 2010. – 225 с.

29 Кааринен Х. Сети UMTS. Архитектура, мобильность, сервисы. – М.: Техносфера, 2007. – 110 с.

30 Печаткин А. В. Системы мобильной связи. Часть 1. – РГАТА, Рыбинск, 2008. – 229 с.

31 Тихвинский В. О., Терентьев С. В., Юрчук А. Б. Сети мобильной связи LTE: технология и архитектура. – М.: Эко-Трендз, 2010. – 176 с.

## Приложение А

### Справочный материал

Таблица А.1 - Диапазон частот для сети радиодоступа E-UTRA

| Порядок<br>рабочих<br>диапазонов | Диапазон частот, МГц  |                      | Вид дуплекса |
|----------------------------------|-----------------------|----------------------|--------------|
|                                  | Линия «вверх»<br>(UL) | Линия «вниз»<br>(DL) |              |
| 1                                | 1920 - 1980           | 2110 - 2170          | FDD          |
| 2                                | 1850 – 1910           | 1930 – 1990          | FDD          |
| 3                                | 1710 – 1785           | 1805 – 1880          | FDD          |
| 4                                | 1710 – 1755           | 2110 – 2155          | FDD          |
| 5                                | 824 – 849             | 869 – 894            | FDD          |
| 6                                | 830 – 840             | 875 – 885            | FDD          |
| 7                                | 2500 – 2570           | 2620 – 2690          | FDD          |
| 8                                | 880 – 915             | 925 – 960            | FDD          |
| 9                                | 1749,9 – 1784,9       | 1844,9 – 1879,9      | FDD          |
| 10                               | 1710 – 1770           | 2110 – 2170          | FDD          |
| 11                               | 1427,9 – 1452,9       | 1475 – 1500,9        | FDD          |
| 12                               | 698 – 716             | 728 – 746            | FDD          |
| 13                               | 777 – 787             | 746 – 756            | FDD          |
| 14                               | 788 – 798             | 758 – 768            | FDD          |
| 17                               | 704 – 716             | 734 – 746            | FDD          |
| 18                               | 815 – 830             | 860 – 875            | FDD          |
| 19                               | 830 – 845             | 875 – 890            | FDD          |
| 33                               | 1900 – 1920           |                      | TDD          |
| 34                               | 2010 – 2025           |                      | TDD          |
| 35                               | 1850 – 1910           |                      | TDD          |
| 36                               | 1930 – 1990           |                      | TDD          |
| 37                               | 1910 – 1930           |                      | TDD          |
| 38                               | 2570 – 2620           |                      | TDD          |
| 39                               | 1880 – 1920           |                      | TDD          |
| 40                               | 2300 – 2400           |                      | TDD          |

Таблица А.2 - Основные параметры оптических кабелей

| Параметры                                      | Тип прокладки кабеля |               |               |
|--|----------------------|---------------|---------------|
|  | в<br>канализации     | в<br>грунте   | подвесной     |
| Марка кабеля                                   | ОКС-М                | ОКБ-Т         | ОКА-Т         |
| Параметры                                      | Тип прокладки кабеля |               |               |
|  | в<br>канализации     | в<br>грунте   | подве<br>сной |
| Число оптических волокон                       | 4 – 72               | 4 – 24        | 4 – 24        |
| Компания–производитель<br>волокна              | Fujikura             | Fujik<br>ura  | Fujik<br>ura  |
| Тип волокна                                    | ОМ                   | ОМ            | ОМ            |
| Число пластмассовых<br>(металлических) модулей | 6 – 12               | 1<br>(метал.) | 1<br>(метал.) |
| Диаметр трубки модуля, мм                      | 2                    | 3,0 – 6,0     | 3,0 – 6,0     |
| Число/диаметр корделей                         | -/2,0                | -/3,0 – 6,0   | -/3,0 – 6,0   |
| Внешний диаметр кабеля, мм                     | 15,0                 | 18,5          | 18,5          |
| Масса кабеля, кг/км                            | 190                  | 436           | 540           |
| Рабочая температура<br>окружающей среды, °С    | -40 - +50            | -40 - +50     | -60 - +60     |
| Минимальный радиус изгиба<br>кабеля, мм        | 250                  | 250           | 300           |
| Допустимое растягивающее<br>усилие, кН         | 1,5                  | 7,0           | 3,5 – 7,0     |
| Длина поставки, км                             | 2,0                  | 2,0           | 2,0           |

## Приложение Б

### Листинг программы

Б.1 Исходный код BaseSwipeListViewListener.java

```
package com.fortysevendeg.swipelistingview;
public class BaseSwipeListViewListener implements SwipeListViewListener {
    @Override
    public void onOpened(int position, boolean toRight) {
    }
    @Override
    public void onClosed(int position, boolean fromRight) {
    }
    @Override
    public void onListChanged() {
    }
    @Override
    public void onMove(int position, float x) {
    }
    @Override
    public void onStartOpen(int position, int action, boolean right) {
    }
    @Override
    public void onStartClose(int position, boolean right) {
    }
    @Override
    public void onClickFrontView(int position) {
    }
    @Override
    public void onClickBackView(int position) {
    }
    @Override
    public void onDismiss(int[] reverseSortedPositions) {
    }
    @Override
    public int onChangeSwipeMode(int position) {
        return SwipeListView.SWIPE_MODE_DEFAULT;
    }
    @Override
    public void onChoiceChanged(int position, boolean selected) {
    }
```



```
}  
@Override  
public void onChoiceStarted() {  
}  
@Override  
public void onChoiceEnded() {  
}  
@Override  
public void onFirstListItem() {  
}  
@Override  
public void onLastListItem() {  
}}
```

## **Б2. Исходный код SwipeListView.java**

```
package com.fortysevendeg.swipelistview;  
import android.content.Context;  
import android.content.res.TypedArray;  
import android.database.DataSetObserver;  
import android.support.v4.view.MotionEventCompat;  
import android.support.v4.view.ViewConfigurationCompat;  
import android.util.AttributeSet;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.ViewConfiguration;  
import android.widget.ListAdapter;  
import android.widget.ListView;  
import java.util.List;  
public class SwipeListView extends ListView {  
    public final static int SWIPE_MODE_DEFAULT = -1;  
    public final static int SWIPE_MODE_NONE = 0;  
    public final static int SWIPE_MODE_BOTH = 1;  
    public final static int SWIPE_MODE_RIGHT = 2;  
    public final static int SWIPE_ACTION_CHOICE = 2;  
    public final static int SWIPE_ACTION_NONE = 3;  
    public final static      String      SWIPE_DEFAULT_FRONT_VIEW      =  
    "swipelist_frontview";  
    public final static      String      SWIPE_DEFAULT_BACK_VIEW      =  
    "swipelist_backview";  
    private final static int TOUCH_STATE_REST = 0;
```

*Продолжение приложения Б*

```
private final static int TOUCH_STATE_SCROLLING_Y = 2;
private int touchState = TOUCH_STATE_REST;
private float lastMotionX;
private float lastMotionY;
private int touchSlop;
int swipeFrontView = 0;
int swipeBackView = 0;
private SwipeListViewListeners swipeListViewListener;
private SwipeListViewTouchListener touchListener;
public SwipeListView(Context context, int swipeBackView,
int swipeFrontView) {
    super(context);
    this.swipeFrontView = swipeFrontView;
    this.swipeBackView = swipeBackView;
    init(null);
}
public SwipeListView(Context context, AttributeSet attrs) {
    super(context, attrs);
    init(attrs);
}
public SwipeListView(Context context, AttributeSet attrs, int defStyleAttr) {
    super(context, attrs, defStyleAttr);
    init(attrs);
}
private void init(AttributeSet attrs) {
    int swipeMode = SWIPE_MODE_BOTH;
    boolean swipeOpenOnLongPress = true;
    boolean swipeCloseAllItemsWhenMoveList = true;
    long swipeAnimationTime = 0;
    int swipeDrawableChecked = 0;
    int swipeDrawableUnchecked = 0;
    int swipeActionLeft = SWIPE_ACTION_REVEAL;
    int swipeActionRight = SWIPE_ACTION_REVEAL;
    if (attrs != null) {
        TypedArray styled = getContext().obtainStyledAttributes(attrs,
R.styleable.SwipeListView);
        swipeMode = styled.getInt(R.styleable.SwipeListView_swipeMode,
SWIPE_MODE_BOTH);
        swipeActionLeft = styled.getInt(R.styleable.SwipeListView_swipeActionLeft,
SWIPE_ACTION_REVEAL);
```

## Продолжение приложения Б

```
swipeActionRight = styled.getInt(R.styleable.SwipeListView_swipeActionRight,
SWIPE_ACTION_REVEAL);
swipeOffsetLeft =
styled.getDimension(R.styleable.SwipeListView_swipeOffsetLeft, 0);
swipeOffsetRight =
styled.getDimension(R.styleable.SwipeListView_swipeOffsetRight, 0);
swipeOpenOnLongPress =
styled.getBoolean(R.styleable.SwipeListView_swipeOpenOnLongPress, true);
swipeAnimationTime =
styled.getInteger(R.styleable.SwipeListView_swipeAnimationTime, 0);
swipeCloseAllItemsWhenMoveList =
styled.getBoolean(R.styleable.SwipeListView_swipeCloseAllItemsWhenMoveList,
true);
swipeDrawableChecked =
styled.getResourceId(R.styleable.SwipeListView_swipeDrawableChecked, 0);
swipeDrawableUnchecked =
styled.getResourceId(R.styleable.SwipeListView_swipeDrawableUnchecked, 0);
swipeFrontView =
styled.getResourceId(R.styleable.SwipeListView_swipeFrontView, 0);
swipeBackView =
styled.getResourceId(R.styleable.SwipeListView_swipeBackView, 0  }
if (swipeFrontView == 0 || swipeBackView == 0) {
swipeFrontView =
getContext().getResources().getIdentifier(SWIPE_DEFAULT_FRONT_VIEW, "id",
getContext().getPackageName());
swipeBackView =
getContext().getResources().getIdentifier(SWIPE_DEFAULT_BACK_VIEW, "id",
getContext().getPackageName());
if (swipeFrontView == 0 || swipeBackView == 0) {
throw new RuntimeException(String.format("You forgot the attributes
swipeFrontView or swipeBackView. You can add this attributes or use '%s' and '%s'
identifiers", SWIPE_DEFAULT_FRONT_VIEW,
SWIPE_DEFAULT_BACK_VIEW));} }
final ViewConfiguration configuration = ViewConfiguration.get(getContext());
touchSlop =
ViewConfigurationCompat.getScaledPagingTouchSlop(configuration);
touchListener = new SwipeListViewTouchListener(this, swipeFrontView,
swipeBackView);
if (swipeAnimationTime > 0) {
```

## *Продолжение приложения Б*

```
touchListener.setAnimationTime(swipeAnimationTime);
    }
    touchListener.setRightOffset(swipeOffsetRight);
    touchListener.setLeftOffset(swipeOffsetLeft);
    touchListener.setSwipeActionLeft(swipeActionLeft);
    touchListener.setSwipeActionRight(swipeActionRight);
    touchListener.setSwipeMode(swipeMode);
    touchListener.setSwipeClosesAllItemsWhenListMoves(swipeCloseAllItemsWhenMoveList);
    touchListener.setSwipeOpenOnLongPress(swipeOpenOnLongPress);
    touchListener.setSwipeDrawableChecked(swipeDrawableChecked);
    touchListener.setSwipeDrawableUnchecked(swipeDrawableUnchecked);
    setOnTouchListener(touchListener);
    setOnScrollListener(touchListener.makeScrollListener());
    }
    public void recycle(View convertView, int position) {
        touchListener.reloadChoiceStateInView(convertView.findViewById(swipeFrontView), position);
    }
    public boolean isChecked(int position) {
        return touchListener.isChecked(position);
    }
    public List<Integer> getPositionsSelected() {
        return touchListener.getPositionsSelected();
    }
    public int getCountSelected() {
        return touchListener.getCountSelected();
    }
    public void unselectedChoiceStates() {
        touchListener.unselectedChoiceStates();
    }
    @Override
    public void setAdapter(ListAdapter adapter) {
        super.setAdapter(adapter);
        touchListener.resetItems();
        adapter.registerDataSetObserver(new DataSetObserver() {
            @Override
            public void onChanged() {
                super.onChanged();
                onListChanged();
                touchListener.resetItems();
            }
        });
    }
}
```

*Продолжение приложения Б*

```
public void dismiss(int position) {
    int height = touchListener.dismiss(position);
    if (height > 0) {
        touchListener.handlerPendingDismisses(height);
    } else {
        int[] dismissPositions = new int[1];
        dismissPositions[0] = position;
        onDismiss(dismissPositions);
        touchListener.resetPendingDismisses();
    }
}

public void dismissSelected() {
    List<Integer> list = touchListener.getPositionsSelected();
    int[] dismissPositions = new int[list.size()];
    int height = 0;
    for (int i = 0; i < list.size(); i++) {
        int position = list.get(i);
        dismissPositions[i] = position;
        int auxHeight = touchListener.dismiss(position);
        if (auxHeight > 0) {
            height = auxHeight;
        }
    }
    if (height > 0) {
        touchListener.handlerPendingDismisses(height);
    } else {
        onDismiss(dismissPositions);
        touchListener.resetPendingDismisses();
    }
    touchListener.returnOldActions();
}

public void openAnimate(int position) {
    touchListener.openAnimate(position);
}

public void closeAnimate(int position) {
    touchListener.closeAnimate(position);
}

protected void onDismiss(int[] reverseSortedPositions) {
    if (swipeListViewListener != null) {
        swipeListViewListener.onDismiss(reverseSortedPositions);
    }
}

protected void onStartOpen(int position, int action, boolean right) {
    if (swipeListViewListener != null && position !=
    ListView.INVALID_POSITION) {
```

*Продолжение приложения Б*

```
    protected void onStartClose(int position, boolean right) {
        if (swipeListViewListener != null && position !=
ListView.INVALID_POSITION) {
            swipeListViewListener.onStartClose(position, right);}
    protected void onClickFrontView(int position) {
        if (swipeListViewListener != null && position !=
ListView.INVALID_POSITION) {
            swipeListViewListener.onClickFrontView(position);}
    protected void onClickBackView(int position) {
        if (swipeListViewListener != null && position !=
ListView.INVALID_POSITION) {
            swipeListViewListener.onClickBackView(position);}
    protected void onOpened(int position, boolean toRight) {
        if (swipeListViewListener != null && position !=
ListView.INVALID_POSITION) {
            swipeListViewListener.onOpened(position, toRight);}
    protected void onClosed(int position, boolean fromRight) {
        if (swipeListViewListener != null && position !=
ListView.INVALID_POSITION) {
            swipeListViewListener.onClosed(position, fromRight);}
    protected void onChoiceChanged(int position, boolean selected) {
        if (swipeListViewListener != null && position !=
ListView.INVALID_POSITION) {
            swipeListViewListener.onChoiceChanged(position, selected);}
    protected void onChoiceStarted() {
        if (swipeListViewListener != null) {
            swipeListViewListener.onChoiceStarted();}}
    protected void onChoiceEnded() {
        if (swipeListViewListener != null) {
            swipeListViewListener.onChoiceEnded();}}
    protected void onFirstListItem() {
        if (swipeListViewListener != null) {
            swipeListViewListener.onFirstListItem();}}
    protected void onLastListItem() {
        if (swipeListViewListener != null) {
            swipeListViewListener.onLastListItem();}}
    protected void onListChanged() {
        if (swipeListViewListener != null) {
            swipeListViewListener.onListChanged();}}
    protected void onMove(int position, float x) {
        if (swipeListViewListener != null && position !=
ListView.INVALID_POSITION) {
```

*Продолжение приложения Б*

```
protected int changeSwipeMode(int position) {  
    if (swipeListViewListener != null && position !=  
    ListView.INVALID_POSITION) {  
        return swipeListViewListener.onChangeSwipeMode(position);  
    }  
    return SWIPE_MODE_DEFAULT;  
}  
public void setSwipeListViewListener(SwipeListViewListenerswipeListViewLi  
stener) {  
    this.swipeListViewListener = swipeListViewListener;  
}  
public void resetScrolling() {  
    touchState = TOUCH_STATE_REST;  
}  
public void setOffsetRight(float offsetRight) {  
    touchListener.setRightOffset(offsetRight);  
}  
public void setOffsetLeft(float offsetLeft) {  
    touchListener.setLeftOffset(offsetLeft);  
}  
public void setSwipeCloseAllItemsWhenMoveList(boolean swipeCloseAllItems  
WhenMoveList) {  
    touchListener.setSwipeClosesAllItemsWhenListMoves(swipeCloseAllItemsWh  
enMoveList);  
}  
public void setSwipeOpenOnLongPress(boolean swipeOpenOnLongPress) {  
    touchListener.setSwipeOpenOnLongPress(swipeOpenOnLongPress);  
}  
public void setSwipeMode(int swipeMode) {  
    touchListener.setSwipeMode(swipeMode);  
}  
public int getSwipeActionLeft() {  
    return touchListener.getSwipeActionLeft();  
}  
public void setSwipeActionLeft(int swipeActionLeft) {  
    touchListener.setSwipeActionLeft(swipeActionLeft);  
}  
public int getSwipeActionRight() {  
    return touchListener.getSwipeActionRight();  
}  
public void setSwipeActionRight(int swipeActionRight) {  
    touchListener.setSwipeActionRight(swipeActionRight);  
}  
public void setAnimationTime(long animationTime) {
```



*Продолжение приложения Б*

```
touchListener.setAnimationTime(animationTime);
    }
    @Override
    public boolean onInterceptTouchEvent(MotionEvent ev) {
        int action = MotionEventCompat.getActionMasked(ev);
        final float x = ev.getX();
        final float y = ev.getY();
        if (isEnabled() && touchListener.isSwipeEnabled()) {
            if (touchState == TOUCH_STATE_SCROLLING_X) {
                return touchListener.onTouch(this, ev);
            }
            switch (action) {
                case MotionEvent.ACTION_MOVE:
                    checkInMoving(x, y);
                    return touchState == TOUCH_STATE_SCROLLING_Y;
                case MotionEvent.ACTION_DOWN:
                    touchListener.onTouch(this, ev);
                    touchState = TOUCH_STATE_REST;
                    lastMotionX = x;
                    lastMotionY = y;
                    return false;
                case MotionEvent.ACTION_CANCEL:
                    touchState = TOUCH_STATE_REST;
                    break;
                case MotionEvent.ACTION_UP:
                    touchListener.onTouch(this, ev);
                    return touchState == TOUCH_STATE_SCROLLING_Y;
                default:
                    break;
            }
        }
        return super.onInterceptTouchEvent(ev);
    }

    private void checkInMoving(float x, float y) {
        final int xDiff = (int) Math.abs(x - lastMotionX);
        final int yDiff = (int) Math.abs(y - lastMotionY);
        final int touchSlop = this.touchSlop;
        boolean xMoved = xDiff > touchSlop;
        boolean yMoved = yDiff > touchSlop;
        if (xMoved) {
            touchState = TOUCH_STATE_SCROLLING_X;
            lastMotionX = x;
            lastMotionY = y;
        }
    }
}
```



## Продолжение приложения Б

```
touchState = TOUCH_STATE_SCROLLING_Y;  
lastMotionX = x;  
lastMotionY = y;}}  
public void closeOpenedItems() {  
touchListener.closeOpenedItems();}}
```

## Приложение В Пользовательский интерфейс

Пользовательский интерфейс представлен на рисунках В.1 – В.7.

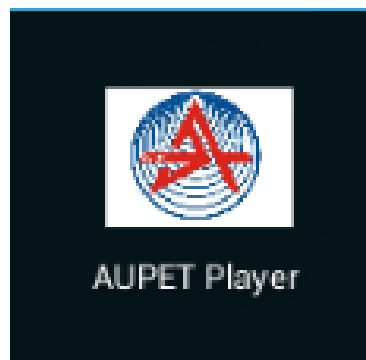


Рисунок В.1 – Значок приложения



Рисунок В.2 – Начальное меню приложения

## Продолжение приложения В

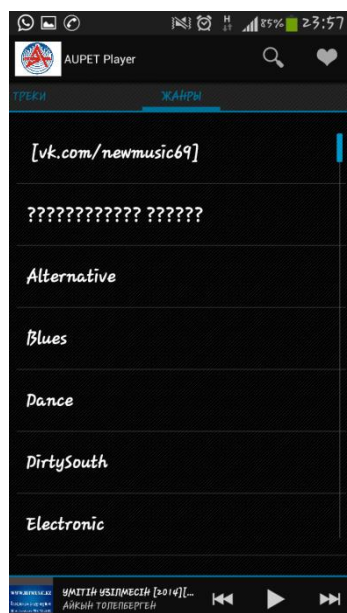


Рисунок Б.3 – Вкладка жанры

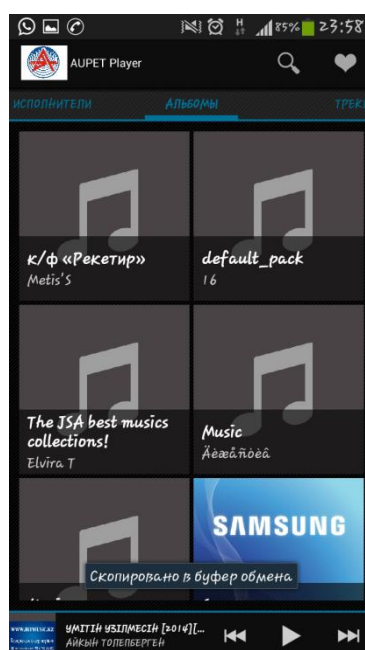


Рисунок Б.4 – Вкладка альбомы

## Продолжение приложения В

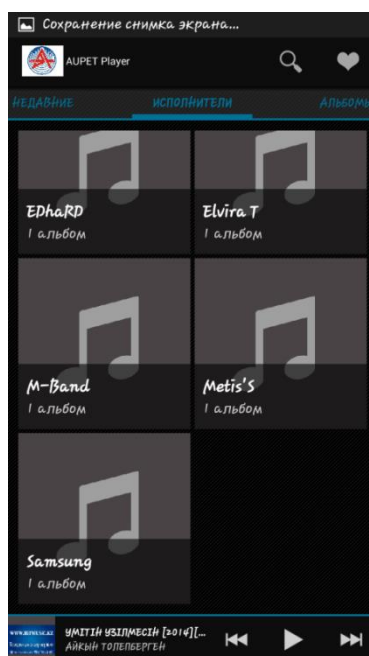


Рисунок Б.5 – Вкладка исполнители



Рисунок Б.6 – Вкладка недавние

## Продолжение приложения В

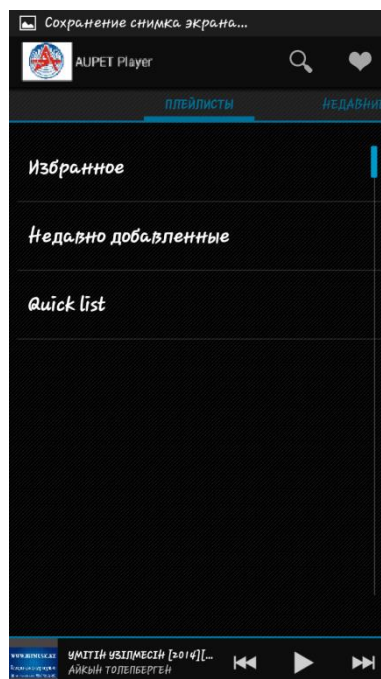


Рисунок Б.7 – Вкладка плейлисты