

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»

Заведующий кафедрой д.ф.-м.н.
профессор Курайбаев З.К.
(Ф.И.О., ученая степень, звание)

« » 20 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка приложения для отчета
услуг АО, казах телеком

Специальность 5В070400 - Вычислительная техника и программное
обеспечение

Выполнил (а) Арканова А.А. ВТ-10-5
(Фамилия и инициалы) группа

Научный руководитель Тургандаев Е.С., к.ф.-м.н. доцент
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Еркешева З.Д., старший преподаватель
(Фамилия и инициалы, ученая степень, звание)
Еркешева «13» 05 2014 г.
(подпись)

по безопасности жизнедеятельности:

Трапезнико И.Г. д.к.н, профессор
(Фамилия и инициалы, ученая степень, звание)
И.Г.Т. «05» мая 2014 г.
(подпись)

по применению вычислительной техники:

Тургандаев Е.С., к.ф.-м.н. доцент
(Фамилия и инициалы, ученая степень, звание)
Е.С.Т. «23» мая 2014 г.
(подпись)

Нормоконтролер: Тусунов Д.М., ассистент
(Фамилия и инициалы, ученая степень, звание)

Д.М.Т. «30» мая 2014 г.
(подпись)

Рецензент: Баймухамбетов А.А., д.ф.м.н. профессор
(Фамилия и инициалы, ученая степень, звание)

А.А.Б. «30» мая 2014 г.
(подпись)

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Информационных технологий
Специальность Вычислительная техника и программное обеспечение
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Азканова Айрама Аскаровна
(фамилия, имя, отчество)

Тема проекта Разработка приложений для смартфона
услуг АО "Казахтелеком"

утверждена приказом ректора № 115 от «24» сентября 2013 г.

Срок сдачи законченной работы «5» июня 2014 г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

разработать оптимально удобное мобильное приложение для смартфона услуг Акционерного общества "Казахтелеком" на платформе Android для пользователей разного уровня пользования.

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

Дипломная работа имеет целью разработать приложения для смартфона услуг АО "Казахтелеком".
Для достижения указанных цели необходимы:
1. Исследовать чертешетную оболочку АО "Казахтелеком"
2. Разработать оптимально удобное мобильное приложение для пользователей разного уровня.

Перечень графического материала (с точным указанием обязательных чертежей)

Архитектура ОС Android

Окно IOS

Сравнение мирового поискового фронта по запросам "mysql" и "Firebase"

Архитектура приложения

Архитектура серверной БД по "Качественной"

Архитектура LVE

Рекомендуемая основная литература

Сайт <http://ru.wikipedia.org/wiki>

Сайт <http://electronicity.ru>



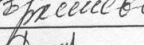
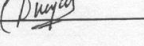
Сайт android.ru

Стив Сурин, Тим Конвер, Джейс Фарк. PHP и MySQL. Библия программиста. - М.: ДМК Пресс, 2010. - 912 с.

Маммаев Е. MySQL Server. Проектирование и реализация баз данных. Серийная компьютерная графика. - СПб.: БИТ, 2004. - 416 с.

Сайт <http://startandroid.ru/ru/>

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Основная часть	Ибрагимов Е. С.	3.03-23.05.14	
БД	Духовных И. Г.	11.04-06.05.14	
Анимация	Красильникова З. Д.	02.04-13.05.14	
Нормоконтроль	Гусев Д. М.	30.05.14 г.	

В данном дипломном проекте разработано приложение для оплаты услуг АО «Казактелеком». Приложение ориентировано на людей, использующие смартфоны, и которым дорого их время, и отвечать их требованиям. Разработка приложения позволит заменить кассы и терминалы, значительно расширить круг потенциальных клиентов и ознакомить с деятельностью организации.

Кроме того, сделан анализ условий труда при эксплуатации программного продукта.

В экономической части составлен бизнес план проекта, подтверждающий его экономическую целесообразность.

Аңдатпа

Бұл дипломдық жобада «Қазактелеком» АҚ қызметтерін төлеуге арналған қосымша әзірленді. Бұл қосымша смартфондарды қолданатын, уақыттары тығыз адамдарға және олардың талаптарына жауап беруге бағытталған. Қосымшаны әзірлеу кассалар мен терминалдарды алмастыруға, әсіресе потенциалды клиенттердің ортасын кеңейтуге және ұйым қызметімен танысуға мүмкіндік береді.

Сонымен қатар, жобада бағдарламалық өнімді пайдаланудағы еңбек шартының талдауы жасалған.

Экономика бөлімінде экономикалық мақсатқа сәйкестілігін растайтын жобаның бизнес жоспары құрылған.

Abstract

In this thesis project has developed an application for payment for services "Kazakhtelecom" JSC . Application is aimed at people who use smartphones, and that their time is expensive , and respond to their requirements. Development application will replace cash and terminals, greatly expand the range of potential customers and to introduce the activities of the organization .

In addition, the analysis made by the working conditions in the operation of software.

In the economic part of the project, a business plan, which confirmed its economic feasibility .

Содержание

Введение	13
1 Определение предметной области	16
1.1 Анализ предметной области	16
1.2 Описание предметной области	18
1.3 Актуальность создания интерактивных мобильных приложений	18
1.4 Анализ рынка разработки мобильных приложений	18
1.5 Обзор существующих программных продуктов	19
1.6 Проблемы разработки мобильных приложений	19
2 Определение системных и технических разработок	21
2.1 Фреймворк qt	21
2.2 Java	23
2.3 Android	24
2.4 IOS	38
2.5 Eclipse	39
2.6 Xml	41
2.7 Mysql	42
2.8 Oracle database 11g standard edition	43
2.9 Firebird	43
2.10 Вывод по определению системных и технических разработок	44
3 Постановка задачи	46
3.1 Назначение и основные возможности проектируемого приложения	46
3.2 АО «Казахттелеком»	46
3.3 Общая архитектура приложения	47
3.4 Структура приложения	48
3.5 Структура БД	49
4 Результаты реализации проекта	52
4.1 Архитектура приложения	52
4.2 Разработка пользовательского интерфейса	53
4.2.1 Разработка интерфейса	53
4.2.2 Оформление	54
4.2.3 Принципы навигации	60
4.3 Навигация	67
4.3.1 Окно оплаты услуг компании	68
4.3.2 Окна услуг компании	70
5 Безопасность жизнедеятельности	72
5.1 Анализ условий труда в отделе «информационных разработок	72
5.2 Расчет систем кондиционирования	74
5.3 Анализ пожарной безопасности	79
5.4 Вывод по разделу безопасности жизнедеятельности	82
6 Экономическое обоснование. Стоимость проекта	83
6.1 Техничко-экономическое обоснование эффективности разработки	83

6.2 Бизнес план	83
6.3 Расчет стоимости разработки по	85
6.3.1 Расходы на заработную плату	85
6.3.2 Социальный налог	86
6.3.3 Расчет затрат на амортизацию	87
6.3.5 Затраты на электроэнергию	88
6.3.6 Расчет затрат на накладные расходы	89
6.3.7 Себестоимость проекта	89
6.4 Цена программного продукта	90
6.5 Вывод по разделу экономическое обоснование	91
Заключение	92
Список использованной литературы	93
Приложение А	94

Введение

В настоящее время большое количество людей владеют мобильными устройствами, что всегда в курсе дел выводит информационные технологии на прямую линию по созданию все новых девайсов и гаджетов. Неудобство эксплуатации компьютеров и ноутбуков обуславливает появление различных нетбуков, планшетов, смартфонов и коммуникаторов, в основе которых лежит все та же операционная система. Лидирующие позиции на сегодняшний день занимают платформы Windows Phone, Android и iOS. Но эти платформы могут работать полноценно только при одном условии - если для них осуществляется разработка мобильных приложений.

В связи с этим появляется главная задача для многих программистов-разработчиков: разработать больше оптимальных и легких в использовании мобильных приложений. Предназначение этих приложений должно быть то, чтоб пользователи смартфонов могли в любое, подходящее для них время, выполнять нужные себе действия.

АО «Казахтелеком» - главный провайдер услуг связи и интернета нашей страны. Свыше 4 000 000 миллионов жителей Республики Казахстан являются пользователями их услуг такими как ID Net, ID Phone, Megaline, ID Net, а также услугами линии телефонной связи.

Android - операционная система для коммуникаторов, планшетных компьютеров, цифровых проигрывателей, цифровых фоторамок, наручных часов, нетбуков и смартбуков, основанная на ядре Linux. Основным языком для разработки служит Java, однако существуют библиотеки позволяющие вести разработку на языке C++. Пользователей данной операционной системой насчитывается больше двух трети пользователей мобильных устройств. На втором месте - iOS (Apple). На третьем, и очень мало кто пользуется - это Windows Phone.

Продажа приложений для мобильных устройств производится через специальные магазины такие как:

- Android - Play Market;
- iOS - App Store;
- Windows Phone - Market Place.

Исходя из всего выше изложенного, могу утверждать, что реализация программного продукта на платформе Android, имеет большую целостность проекта.

Задание для дипломного проекта - разработать оптимальное в использовании мобильное приложение на платформе Android для оплаты услуг компании АО «Казахтелеком».

Целью дипломной работы является разработка приложения для смартфонов под управлением операционной системы Android. Требовалось разработать масштабируемое приложение с поддержкой всех современных устройств на базе Android версии 1.2 и выше. Основной характеристикой

разрабатываемого приложения являлось использование сервисов и процессов, не зависящих от основного приложения и выполняющих обработку данных в фоновом режиме.

В наше время у многих людей нет времени на ожидание очереди в кассах оплаты, или бегать по городу в поисках работающего терминала оплаты. А если вовремя не оплатить свой счет, компания временно отключает услуги до полного погашения задолженности. Поэтому уважающий свое время человек будет искать простые и удобные для себя решения.

Главное предназначение приложения - быть удобным для пользователя приложения. Благодаря приложению пользователь сможет в удобное для себя время, будь он дома или на работе, вовремя оплатить свой счет.

Решаемые задачи: изучить и улучшить знания в разработке приложений для мобильных устройств, а также разработать вышеуказанную программу. Ознакомиться с многопоточными приложениями и особенностями платформы.

В первую очередь необходимо выяснить, для кого создается приложение с помощью каких программ и средств. Необходимо определиться с технологиями, которые нужно будет использовать при разработке приложения. В итоге, должно быть полностью разработано мобильное приложение, отвечающее всем требованиям компании. Для достижения цели необходимо изучение языков программирования приложений, которые являются самыми популярными, изучить структуры аналогичных, провести параллель и выявить то, что будет новшеством в данной области. Далее определение содержания приложения и его дальнейшая разработка.

1 Определение предметной области

1.1 Анализ предметной области

Исследование - это процесс познания определенной предметной области, объекта или явления с определенной целью. Процесс исследования заключается в наблюдении свойств объектов с целью выявления и оценки важных, с точки зрения субъекта-исследователя, закономерных отношений между показателями данных свойств.

Решение любой задачи в сфере разработки программного обеспечения должно начинаться с изучения предметной области.

Предметная область - это мысленно ограниченная область реальной действительности, подлежащая описанию или моделированию и исследованию. Она состоит из объектов, различаемых по свойствам и находящихся в определенных отношениях между собой или взаимодействующих каким-либо образом.

Предметная область - это часть реального мира, она бесконечна и содержит как существенные, так и второстепенные данные, с точки зрения проводимого исследования.

Исследователю необходимо уметь выделить существенную их часть. Например, при решении задачи "Выдавать ли кредит?" важными являются все данные про частную жизнь клиента, вплоть до того, имеет ли работу супруг, есть ли у клиента несовершеннолетние дети, каков уровень его образования и т.д. Для решения другой задачи банковской деятельности эти данные будут абсолютно неважны. Существенность данных, таким образом, зависит от выбора предметной области.

В процессе изучения предметной области должна быть создана ее модель. Знания из различных должны быть формализованы при помощи каких-либо средств.

Это могут быть текстовые описания предметной области или специализированные графические нотации. Существует большое количество методик описания предметной области: например, методика структурного анализа SADT и основанная на нем IDEF0, диаграммы потоков данных Гейна-Сарсона, методика объектно-ориентированного анализа UML и другие. Модель предметной области описывает процессы, происходящие в предметной области, и данные, которые в этих процессах используются.

Приложение - последовательность инструкций, предназначенных для исполнения устройством управления вычислительной машины. Программа - один из компонентов программного обеспечения. В зависимости от контекста рассматриваемый термин может относиться также и к исходным текстам программы. Приложение как объект авторского права и других прав

интеллектуальной собственности относится к категории нематериальных активов [1].

Современные технологии все более прочно входят в жизнь каждого человека, давая ему дополнительные возможности как для развлечений, так и для работы. Так, сегодня в офисах используют мобильный банк документов, Интернет стал незаменимым помощником в ведении переговоров и работе, а представить офис без компьютера и вовсе невозможно. Совершенствуются и различные гаджеты - так, теперь мобильный телефон является не только и не столько средством для совершения звонков, сколько многофункциональным устройством, способным выполнять множество задач.

Очень актуальна сегодня разработка android приложений, Java-приложений, программ для Windows, Symbian, благодаря чему возможности смартфонов на данных платформах расширяются с каждым днем. Само понятие мобильных приложений может охватывать очень широкий круг программ и задач - начиная беспроводными системами, где клиенты - это ноутбуки и серверные станции, и заканчивая WWW- и WAP-сервисами, которые рассчитаны на малогабаритные устройства.

Но в первую очередь речь идет про мобильные телефоны - как с операционными системами, так и без них. Популярностью пользуется разработка мобильных приложений как для простых устройств, так и для современных дорогих гаджетов. Функционально все программы являются очень разнообразными, но разработчики делят их на несколько основных групп.

В первую группу входят персональные приложения, которые обеспечивают доступ пользователя к самым разнообразным информационным сервисам - это системы мониторинга, удаленного управления теми или иными устройствами, системами. Также сюда входят и браузеры, предназначенные для работы в Интернете, просмотра страниц сайтов.

Ко второй группе относят мобильные игры - как обычные, так и сетевые многопользовательские. Такие игры с каждым днем приобретают все большую популярность среди потребителей самых разных категорий, так как дают возможность развлечься во время поездок, обеденных перерывов, ожидания встреч и т.д. Игры становятся все более совершенными и разнообразными, с качественной графикой и сюжетами. Так, если еще несколько лет назад выбор таких приложений ограничивался тетрисом и пасьянсами, то сегодня это огромная сфера с большим выбором контента.

Третью группу составляют бизнес-приложения, программы мобильной коммерции. С их помощью деловые люди могут успешно вести дела, использовать дополнительные сервисы, необходимые для работы с документами, организации коммуникации, работы с определенными сайтами и сервисами [2].

1.2 Описание предметной области

Приложение изначально предназначено для оказания услуг пользователям АО «Казахтелеком». Приложение служит для оплаты услуг компании. С развитием информационных технологий смартфоны стали неотъемлемой частью жизни. Это безукоризненно сказывается на большей разработке приложений на смартфоны.

Приложение является средством экономии времени пользователей. Клиент, не успевающий оплатить или находящийся все время в разъездах, может быстро и оперативно оплатить свой счет. Поэтому это актуально в настоящее время. Компания АО «Казахтелеком» идет в ногу со временем, поэтому им важно это приложение.

В настоящее время под приложением чаще понимается именно его программная версия для персонального компьютера или терминала. Его безусловными преимуществами является объем вводимых данных, автоматическое обновление. Приложение кроме оплаты может показывать ваш текущий счет, подключать или отключать предоставляемые услуги, а также узнавать новости компании.

Однако при использовании приложения возникают проблемы, не позволяющие оплатить/подключить/отключить услуги, такие как:

- не производится подключение к сети;
- на вашей кредитной карточке нет средств;
- на сервере произошел сбой;
- неправильно введены личные данные.

Используемыми платформами для мобильных устройств являются iOS, Android и Windows Phone.

1.3 Актуальность создания интерактивных мобильных приложений

Актуальность исследований в области вопросов построения мобильных приложений связан с тем, что данный вид программного обеспечения:

- перспективен, как объект электронной коммерции;
- предоставляет широкие возможности технологического взаимодействия;
- в обозримом будущем может составить реальную конкуренцию нативным приложениям мобильных операционных систем (Apple iOS, Google Android, Windows Phone).

1.4 Анализ рынка разработки мобильных приложений

В настоящее время казахстанский рынок разработок мобильных приложений меняется очень быстро и, по ряду оценок, считается динамичным. Уже на период 2014-го года в Казахстане существовало около 50 компаний,

оказывающих подобные услуги. Главным «рычагом» разработок остается рост аудитории Internet-пользователей, который стимулирует разработчиков к совершенствованию своих web-проектов и увеличению бюджетов на оптимизацию.

Сейчас в сети Internet совершаются торговые сделки, существует множество электронных валют (Яндекс.Деньги, Webmoney и другие), открыты представительства банков, где пользователи могут работать со своими счетами. Ежедневно миллионы пользователей посещают поисковые системы, электронные энциклопедии, блоги (дневники пользователей), новостные сайты.

1.5 Обзор существующих программных продуктов

На данный момент существуют несколько способов оплаты услуг, такие как:

- кассы оплаты;
- терминалы;
- онлайн-оплата на официальном сайте через личный кабинет.

Все эти способы очень актуальны, но из-за популярности среди населения мобильными устройствами, приложение будет пользоваться большим количеством спросом в Play Markete.

1.6 Проблемы разработки мобильных приложений

По своей сути запуск приложения ничем не отличается от загрузки обычной обычного приложения: нажимаем на приложение - и оно перед нами, точнее, верхняя часть айсберга, которой является интерфейс пользователя. В этом есть несколько плюсов. Первый из них - это то, что само по себе приложение абсолютно не зависит от того, какая операционная система установлена на компьютере пользователя, то есть оно, по сути, является кросс-платформенным. Однако есть здесь и один момент, который может создать некоторые трудности при работе с приложениями - это различные реализации некоторых спецификаций в мобильных устройствах, а также возможность настраивать разные параметры вроде отображения шрифтов, что может привести к некорректной работе некоторых приложений. Но стоит отметить, что данный минус хотя и имеет место, но сколь либо существенного влияния на работу большинства мобильных приложений он не оказывает.

Вторым плюсом мобильных приложений можно назвать то, что сам факт их существования полностью меняет способ распространения продукта. Здесь создатели отходят от традиционных способов распространения программных продуктов путем продажи копий и установки их на каждый мобильных телефонов пользователей. Теперь все гораздо проще: единственная версия приложения расположена на сервере, а все пользователи имеют доступ к ней, вернее, к ее пользовательскому интерфейсу из любого места в мире. Из любого,

где есть Интернет. При этом пользователю даже не нужно устанавливать новую версию приложения - сразу после своего появления она доступна всем, причем многие могут и не заметить каких-либо изменений, тем более если эти изменения не касаются внешнего вида интерфейса. Во всем этом явно виден и положительный момент для разработчиков - им не нужно заботиться о совместимости версий своих приложений, поскольку все пользователи одновременно получают доступ и работают с самой последней версией приложения.

Третьим плюсом для пользователя мы назовем то, что ему нет необходимости устанавливать и настраивать программное обеспечение - все уже установлено на серверах и настроено разработчиками. Все, что требуется от пользователя, это изменить на свой вкус несколько ключевых параметров. Это весьма приятно для пользователей, поскольку большинство не любит возиться с настройками и предпочитает программные продукты, полностью готовые к использованию сразу после их инсталляции, хотя есть и такие, которые предпочитают полностью настроить программу под свой вкус и нужды. Однако в случае мобильных приложений мы избавлены даже от процесса инсталляции.

Четвертым плюсом можно назвать то, что для работы с приложением от пользователя, по большому счету, ничего и не нужно. Разве что мобильный телефон и подключенный интернет. Но тут проблем нет - Интернет-браузер уже имеется в любой операционной системе, и для доступа к необходимому мобильному приложению достаточно просто загрузить его URL в браузер. Использование мобильных приложений во многом снимает ограничения, накладываемые на аппаратную часть мобильного телефона. То есть определенные системные требования к ПК все же есть, но их уровень автоматически достигнут компьютером, раз на нем уже запущены ОС и браузер.

Следующий положительный момент мобильных приложений касается их разработчиков. Ввиду того, что основная часть мобильного приложения сконцентрирована на сервере в одном месте, куда проще заниматься его настройкой, не нужно содержать огромные команды специалистов технической поддержки, занимающихся консультациями пользователей и настройкой приложения на мобильных телефонах во всем мире. Это гораздо менее затратно в финансовом плане и куда более эффективно. При этом пользователю невидима архитектура приложения, в любой момент можно добавить любое количество серверов, на которых установлена основная составляющая приложения, добавить вычислительные мощности, и пользователь этого даже не заметит.

Таким образом, мы видим, что мобильных приложения имеют большое количество плюсов при отсутствии видимых минусов, самым большим и очевидным из которых является невозможность использования приложений при отсутствии доступа к Всемирной паутине.

2 Определение системных и технических разработок

Для разработки приложения в последующих статьях будут рассматриваться и досконально изучаться различные ПО такие как:

- QT creator;
- Java (язык программирования и технологии);
- Android;
- IOS;
- XML (язык гипертекстовых разметок);
- MySql;
- Oracle.

После изучения будут рассматриваться оптимальные для разработки программного продукта.

2.1 Фреймворк Qt

Фреймворк (framework) - программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Qt является кроссплатформенным (переносимым) фреймворком. В большей степени используется для разработки приложений на C++, на так же может работать с такими языками, как Python, Ruby, Java, PHP и другими.

Отличительная особенность Qt от других библиотек - использование Meta Object Compiler (MOC) - предварительной системы обработки исходного кода (в общем-то, Qt - это библиотека не для чистого C++, а для его особого наречия, с которого и «переводит» MOC для последующей компиляции любым стандартным C++ компилятором). MOC позволяет во много раз увеличить мощь библиотек, вводя такие понятия, как слоты и сигналы. Кроме того, это позволяет сделать код более лаконичным. Утилита MOC ищет в заголовочных файлах на C++ описания классов, содержащие макрос Q_OBJECT, и создаёт дополнительный исходный файл на C++, содержащий метаобъектный код.

Механизм сигналов и слотов полностью замещает модель функций обратного вызова, он очень гибок и полностью объектно-ориентирован.

Сигналы и слоты могут быть соединены друг с другом, причем сигнал может быть соединен с большим количеством слотов. Слот, в свою очередь, тоже может быть соединен со многими сигналами. В случае когда слот не делает ничего, кроме отправки полученного сигнала дальше, то можно вообще обойтись без него, а просто соединить сигналы друг с другом. Методы сигналов должны быть обозначены в определении класса специальным словом signals, а слоты - slots. При этом слоты являются обыкновенными методами языка C++ и в их определении могут присутствовать модификаторы public, protected, private. Реализацию кода для сигналов берет на себя MOC. Отправка сигнала производится при помощи ключевого слова emit. Класс, содержащий сигналы и

слоты, должен быть унаследован от класса QObject или от класса, унаследованного от этого класса. Сигнально-слотовые соединения всегда можно удалить (отсоединить), воспользовавшись методом disconnect(), но это нужно крайне редко, т.к. при удалении объекта автоматически уничтожаются все его соединения.

QObject - класс, по сути являющийся основным классом при программировании с использованием Qt. Конструктор класса QObject имеет два параметра: первый используется для создания объектных иерархий, а второй - для присвоения объекту имени.

Свойства объектов важны, т.к. позволяют получать информацию о классе и об объекте в процессе исполнения программы. Все объекты класса QObject или унаследованные от него классы должны создаваться динамически оператором new, а об освобождении памяти созданной объектной иерархии программист может не беспокоиться.

Qt Creator - интегрированная среда разработки

Кроссплатформенная свободная IDE для разработки на C, C++ и QML. Разработана Trolltech (Digia) для работы с фреймворком Qt. Включает в себя графический интерфейс отладчика и визуальные средства разработки интерфейса как с использованием QtWidgets, так и QML.

Qt Creator поддерживает проекты систем qmake и cmake, начиная с версии 2.5 поддерживается autotools. Для проектов, созданных под другими системами, может использоваться в качестве редактора исходных кодов. Есть возможность редактирования этапов сборки проекта. Также IDE нормативно поддерживает системы контроля версии, такие как Subversion, Mercurial, Git, CVS, Vazaar, Perforce. Начиная с версии 2.5, в поле комментария к правке поддерживается автодополнение.

Реализован ряд возможностей при работе с сигнатурами методов, а именно:

- автогенерация пустого тела метода после его обновления;
- возможность автоматически изменить сигнатуру метода в определении, если она была изменена в объявлении и наоборот;
- возможность автоматически поменять порядок следования аргументов.
- среда разработки имеет графический интерфейс для следующих отладчиков: GDB, CDB и QML/JavaScript. В качестве отдельной опции реализовано отображение содержимого контейнеров, таких как QString, std::map и прочих. Поддерживаются следующие режимы отладки:
 - простой для отладки локально запущенных приложений, таких как GUI приложения на Qt;
 - терминал для отладки локально запущенных процессов, которым требуется консоль, обычно это приложения без GUI;

- подключённый для отладки локальных процессов, запущенных вне Qt Creator;
- удалённый для отладки запущенных на другой машине процессов (используя gdbserver);
- ядро для отладки завершившихся аварийно процессов на Unix;
- Post-mortem для отладки завершившихся аварийно процессов на Windows;
- TRK для отладки процессов, запущенных на устройстве Symbian.

При навигации по коду доступно переключение между определением и объявлением метода, переход к объявлению метода, переименование метода как в отдельном проекте, так и во всех открытых. Также есть возможность вызвать справку согласно текущему контексту.

2.2 Java

Java является основой практически для всех типов сетевых приложений и всеобщим стандартом для разработки и распространения встроенных и мобильных приложений, игр, веб-контента и корпоративного программного обеспечения. В мире насчитывается более 9 миллионов специалистов, разрабатывающих приложения на Java, которая позволяет эффективно разрабатывать, внедрять и использовать превосходные приложения и услуги.

Преимуществами Java являются:

- Java используется на 97% корпоративных настольных ПК;
- Java используется на 89% настольных ПК в США;
- 9 млн разработчиков на Java в мире;
- инструмент номер 1 среди разработчиков;
- программа номер 1 среди разработчиков;
- Java используется в 3 млрд мобильных телефонов;
- Java входит в комплект поставки 100% всех проигрывателей дисков Blu-ray;
- используется 5 млн Java Card;
- Java используется в 125 млн ТВ-устройств;
- 5 из 5 основных производителей оригинального оборудования включают в комплект поставки Java ME.

Технология Java протестирована, усовершенствована, расширена и проверена участниками сообщества разработчиков Java, архитекторов и энтузиастов. Java позволяет разрабатывать высокопроизводительные портативные приложения практически на всех компьютерных платформах. Доступность приложений в разнородных средах позволяет компаниям предоставлять более широкий спектр услуг, способствует повышению производительности, уровня взаимодействия и совместной работы конечных пользователей и существенному снижению стоимости совместного владения

корпоративными и потребительскими приложениями. Java стала незаменимым инструментом для разработчиков и открыла для них следующие возможности:

- написание программного обеспечения на одной платформе и его запуск практически на любой другой платформе;
- создание программ, работающих в веб-браузере и имеющих доступ к веб-службам;
- разработка приложений на стороне сервера для форумов в Интернете, магазинов, опросов, обработки форм HTML и много другого;
- объединение приложений или служб с использованием языка Java для создания высокоспециализированных приложений или служб;
- создание многофункциональных и эффективных приложений для мобильных телефонов, удаленных процессоров, микроконтроллеров, беспроводных модулей, датчиков, шлюзов, потребительских продуктов и практически любых других категорий электронных устройств.

Для Java-платформы существуют различные множество фреймворков, основными из которых являются:

- Spring Framework;
- JBoss Seam;
- Apache Wicket;
- Apache Click;
- Apache Tapestry;
- Apache Struts.

2.3 Android

Android - это портативная (сетевая) операционная система для коммуникаторов, планшетных компьютеров, электронных книжек, цифровых проигрывателей, наручных часов, нетбуков и смартбуков, основанная на ядре Linux. Изначально разрабатывалась компанией Android Inc., которую затем купила Google. Впоследствии Google инициировала создание альянса Open Handset Alliance, который сейчас и занимается поддержкой и дальнейшим развитием платформы. Android позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки. Android Native Development Kit создаёт приложения, написанные на Си и других языках [3].

Android - ОС для смартфонов, планшетов и нетбуков. Компания Google приобрела разработчика программного обеспечения Android inc. в 2005 году. Операционная система Android основана на модифицированном ядре Linux и собственной реализации Java от Google.

Приложения для ОС Android включают в себя java-приложения и библиотеки, которые запускаются виртуальной машиной Dalvik с JIT компилятором. Сами приложения устанавливаются в виде файла формата .APK, который по своей сути является архивом с не скомпилированными

компонентами приложения, задача же виртуальной машины Dalvik - компиляция кода приложения, которая происходит непосредственно во время запуска ПП. Библиотеки включают в себя систему управления, графику OpenGL ES 2.0, движок WebKit, графический движок SGL, SSL и библиотеки Bionic.

С выпуском последней версии Android Kit Kat в тестовом режиме, доступном только в меню для разработки появилась возможность выбора альтернативной Dalvik среды запуска приложений ART (Android Runtime), которая имеет ряд преимуществ перед Dalvik. Основное достоинство ART заключается в компиляции приложения непосредственно во время его установки, что значительно снижает нагрузку на центральный процессор во время его выполнения. Однако у данного метода есть и свои отрицательные стороны, одной из которых является увеличение размера занимаемой памяти устройства приложением, а также неправильное компилирование некоторых ПП, в результате чего требуется их адаптация разработчиками. Основным же недостатком новой среды является значительное увеличение времени установки приложения в систему, а также необходимость повторной их компиляции после каждого обновления системы. Так, активно используемое устройство, при после обновления ОС, может загружаться свыше часа. Помимо этого, в новой версии ОС была добавлена поддержка OpenGL ES 3.0, что позволяет создавать невероятной красоты объемные объекты, сравнимые по качеству изображения с современными игровыми консолями.

Разработчикам Google предлагает для свободного скачивания инструментарий для разработки (Android SDK), который предназначен для x86-машин под операционными системами Windows (XP или выше), Mac OS X (10.4.8 или выше) и Linux. Для разработки требуется также наличие установленного Java Development Kit (JDK) версии 5 - 7, версия 8 на данный момент не поддерживается. Помимо этого, существуют плагины для Eclipse - «Android Development Tools» (ADT), предназначенный для Eclipse версий 3.3-3.5, плагин для IntelliJ IDEA. А также предлагается официальная среда разработки Android Studio, которая является альтернативой для вышеперечисленных, основанная на IntelliJ IDEA.

На рисунке 2.1 представлена архитектура ОС Android.

На основе рисунка 2.1 можно выделить следующие уровни ОС Android:

- уровень приложений (Applications) - в состав Android входит комплект базовых приложений: клиенты электронной почты и SMS, календарь, различные карты, браузер, программа для управления контактами и много другое. Все приложения, запускаемые на платформе Android написаны на языке Java;



Рисунок 2.1 - Архитектура ОС Android

- уровень каркаса приложений (Application Framework) - Android позволяет использовать всю мощь API, используемого в приложениях ядра. Архитектура построена таким образом, что любое приложение может использовать уже реализованные возможности другого приложения при условии, что последнее откроет доступ на использование своей функциональности. Таким образом, архитектура реализует принцип многократного использования компонентов ОС и приложений. Основой всех приложений является набор систем и служб:

- система представлений (View System) - это богатый набор представлений с расширяемой функциональностью, который служит для построения внешнего вида приложений, включающий такие компоненты, как списки, таблицы, поля ввода, кнопки и т.п.;

- контент-провайдеры (Content Providers) - это службы, которые позволяют приложениям получать доступ к данным других приложений, а также предоставлять доступ к своим данным;

- менеджер ресурсов (Resource Manager) предназначен для доступа к строковым, графическим и другим типам ресурсов;

- менеджер извещений (Notification Manager) позволяет любому приложению отображать пользовательские уведомления в строке статуса;

- менеджер действий (Activity Manager) управляет жизненным циклом приложений и предоставляет систему навигации по истории работы с действиями;

- уровень библиотек (Libraries) - платформа Android включает набор C/C++ библиотек, используемых различными компонентами ОС. Для разработчиков доступ к функциям этих библиотек реализован через использование Application Framework. Ниже представлены некоторые из них:

- System C library - BSD-реализация стандартной системной библиотеки C (libc) для встраиваемых устройств, основанных на Linux;

- Media Libraries - библиотеки, основанные на PacketVideo's OpenCORE, предназначенные для поддержки проигрывания и записи популярных аудио- и видео- форматов (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG и т.п.);

- Surface Manager - менеджер поверхностей управляет доступом к подсистеме отображения 2D- и 3D- графических слоев;

- LibWebCore - современный движок web-браузера, который предоставляет всю мощь встроенного Android-браузера;

- SGL - движок для работы с 2D-графикой;

- 3D libraries - движок для работы с 3D-графикой, основанный на OpenGL ES 1.0 API;

- FreeType - библиотека, предназначенная для работы со шрифтами;

- SQLite - мощный легковесный движок для работы с реляционными БД;

- уровень среды исполнения (Android Runtime) - в состав Android входит набор библиотек ядра, которые предоставляют большую часть функциональности библиотек ядра языка Java. Платформа использует оптимизированную, регистр-ориентированную виртуальную машину Dalvik, в отличие от нее стандартная виртуальная машина Java - стек-ориентированная. Каждое приложение запускается в своем собственном процессе, со своим собственным экземпляром виртуальной машины. Dalvik использует формат Dalvik Executable (*.dex), оптимизированный для минимального использования памяти приложением. Это обеспечивается такими базовыми функциями ядра Linux, как организация поточной обработки и низкоуровневое управление памятью. Байт-код Java, на котором написаны ваши приложения, компилируются в dex-формат при помощи утилиты dx, входящей в состав SDK;

- уровень ядра Linux (Linux Kernel) - Android основан на ОС Linux версии 2.6, тем самым платформе доступны системные службы ядра, такие как управление памятью и процессами, обеспечение безопасности, работа с сетью и драйверами. Также ядро служит слоем абстракции между аппаратным и программным обеспечением.

Для разработчиков доступны следующие библиотеки:

- Bionic - библиотека стандартных функций, несовместимая с libc;

- SSL - шифрование;

- Media Framework (PacketVideo OpenCORE, MPEG4, H.264, MP3, AAC, AMR, JPG, PNG);

- Surface Manager;

- LibWebCore (на базе WebKit);
- SGL - 2D-графика;
- OpenGL ES - 3D-библиотека;
- FreeType - шрифты;
- SQLite - легковесная СУБД.

По сравнению с обычными приложениями Linux, приложения Android подчиняются дополнительным правилам:

- Content Providers - обмен данными между приложениями;
- Resource Manager - доступ к таким ресурсам, как файлы XML, PNG, JPEG;
- Notification Manager - доступ к строке состояния;
- Activity Manager - управление активными приложениями.

На данный момент существуют различные версии Android [4]:

- **Android 1.0 «Apple Pie (раннее название «Astroboy»)»** - первая стабильная версия системы, появившаяся в 2008 г;

- **Android 1.1 «Banana Bread (раннее название «Bender»)»:**

- исправление нескольких проблем;
- изменения в API;
- добавлены подробности и отзывы к картам;
- увеличен период отключения экрана при использовании в режиме телефона;
- добавлены кнопки «Show» и «Hide» в меню вызова;
- добавлена поддержка сохранения вложений из MMS;
- добавлена поддержка меток в раскладках;

- **Android 1.5 «Cupcake»:**

- возможность установки сторонних клавиатур;
- новая программная клавиатура с функцией автозаполнения и возможностью работы при различных положениях экрана;
- поддержка виджетов и папок на рабочем столе;
- запись и воспроизведение видео в MPEG-4 и 3GP;
- поддержка Bluetooth-профиля A2DP и AVRCP;
- возможность автоматического подключения к Bluetooth гарнитуре, находящейся на определённом расстоянии;
- обновление WebKit и Squirrelfish Javascript Engine;
- возможность публикации фотографий (Picasa) и видео (YouTube) в интернете;
- добавлен поиск по веб-странице и возможность работы с текстом;
- визуальные изменения в браузере;
- изменения списка контактов и истории звонков;
- добавлены инструменты для обслуживания и автоматического определения файловой системы карты памяти;

- анимация при переключении между окнами;
- **Android 1.6 «Donut»:**
 - изменён дизайн и улучшена работа с магазином приложений Android Market;
 - интегрирован интерфейс для работы с фото, видеокамерой и галереей снимков, позволяющий легко переключаться между режимом фото и видео, а в галерее появилась возможность выделять сразу несколько объектов для удаления;
 - добавлена функция мультязычного голосового поиска;
 - обновлённая функция поиска, позволяющая вести поиск среди закладок, истории, контактов, а также в интернете;
 - повышена скорость работы приложений поиска и камеры;
 - добавлена поддержка CDMA, 802.1x, VPN, а также функция синтеза речи;
 - поддержка разрешений WVGA;
 - добавлен фреймворк жестов и инструмент GestureBuilder;
 - добавлена бесплатная возможность пошаговой навигации от Google;
- **Android 2.0 «Eclair»:**
 - добавлена возможность использования нескольких аккаунтов Google;
 - обновлён интерфейс клиента электронной почты для работы с несколькими аккаунтами;
 - добавлена поддержка Microsoft Exchange Server через Exchange ActiveSync 2.5;
 - добавлена возможность быстрого доступа к контактам;
 - добавлена возможность поиска по SMS и MMS сообщениям и автоматического удаления старых сообщений при достижении лимита;
 - обновление камеры: поддержка вспышки, цифрового увеличения и эффектов;
 - улучшена раскладка экранной клавиатуры и механизм обучения словаря;
 - новый пользовательский интерфейс браузера и поддержка HTML5;
 - обновлён календарь;
 - обновление графической архитектуры, позволившее улучшить аппаратное ускорение;
 - поддержка Bluetooth 2.1 и профиля OPP и PVAR;
 - добавлена поддержка новых размеров и разрешений экранов;
 - изменён пользовательский интерфейс;
 - улучшен класс MotionEvent для поддержки нескольких касаний;

- **Android 2.0.1** произошли мелкие изменения в API, багфиксы, framework behavioral changes;
- **Android 2.1 «Eclair»:**
 - добавлены «живые» обои;
 - улучшена контрастность фона;
 - добавлены улучшения Google Maps 3.1.2;
- **Android 2.2 «Froyo»:**
 - общая оптимизация ОС Android, памяти и производительности;
 - дополнительные улучшения скорости работы приложений, использующих JIT-компиляцию;
 - интеграция в браузер JavaScript-движка V8, ранее реализованного в Chrome;
 - улучшенная поддержка Microsoft Exchange (политики безопасности, автоматическое распознавание, просмотр GAL, синхронизация календарей, удаленная работа);
 - улучшен запуск приложений через ярлыки телефона и браузера;
 - добавлена функциональность тетеринга посредством USB и Wi-Fi;
 - добавлена возможность блокировать доступ к данным по сотовой связи;
 - обновлено приложение для работы с Android Market, в которое добавлена возможность групповых и автоматических обновлений;
 - быстрое переключение между несколькими клавиатурными раскладками и соответствующими им словарями;
 - голосовой набор и обмен контактами через Bluetooth;
 - поддержка числовых и символьно-числовых паролей;
 - поддержка полей в браузере для загрузки файлов;
 - поддержка установки приложений в расширенную память;
 - поддержка Adobe Flash 10.2;
 - поддержка экранов со сверхвысоким разрешением (320 dpi), как например 4-дюймовый экран с разрешением 720p.
- **Android 2.2.1** исправлены некоторые ошибки, включены обновления безопасности и производительности;
- **Android 2.2.2** исправлены мелкие ошибки, в том числе проблема с SMS на Nexus One;
- **Android 2.2.3** произошло обновление протоколов безопасности;
- **Android 2.3.0 «Gingerbread»:**
 - обновлённый дизайн пользовательского интерфейса;
 - поддержка сверхвысоких размеров экранов и разрешений (WXGA и выше);
 - встроенная поддержка протокола SIP VoIP-телефонии;

- поддержка проигрывания видео форматов WebM/VP8, а также поддержка аудио стандарта AAC;
 - новые звуковые эффекты: реверберация, эквалайзер, виртуализация наушников, усиление басов;
 - поддержка стандарта Near Field Communication;
 - системная поддержка копирования и вставки;
 - переработанная программная клавиатура с поддержкой нескольких касаний;
 - улучшенная поддержка встроенной разработки кода;
 - улучшения для разработчиков игр в области аудио, графической части и ввода информации;
 - параллельная сборка мусора для улучшения производительности;
 - встроенная поддержка большего числа сенсоров (например, гироскопы и барометры);
 - менеджер скачивания для длительных загрузок;
 - улучшено управление питанием и контроль за приложениями;
 - встроенная поддержка нескольких камер;
 - переход с использования файловой системы YAFFS на ext4;
- **Android 2.3.2** исправлены баги SMS/MMS;
- **Android 2.3.3** немного улучшений и API для платформы Android 2.3;
- **Android 2.3.4** появился видео и голосовой чат для Google Talk;
- **Android 2.3.5 «Gingerbread»:**
 - улучшения производительности сетевого стека для аппарата Nexus S 4G, другие изменения и улучшения;
 - исправлен баг Bluetooth на Samsung Galaxy S;
 - улучшено приложение Gmail;
 - исправлена проблема с часовыми поясами (GMT+3 и GMT+4);
- **Android 2.3.6:** исправлен баг голосового поиска;
- **Android 2.3.7:** добавлена поддержка платежной системы Google Wallet для аппарата Nexus S 4G;
- **Android 3.0 «Honeycomb»:**
 - поддержка многоядерных процессоров;
 - улучшенная поддержка планшетов благодаря новому пользовательскому интерфейсу;
 - трёхмерный рабочий стол с переписанными виджетами;
 - улучшенная многозадачность;
 - улучшения обозревателя, среди которых закладки для открытых веб-страниц, автозаполнение форм, синхронизация ссылок с Google Chrome, приватный режим просмотра;
 - поддержка видеочатов при помощи Google Talk;
 - поддержка аппаратного ускорения;
- **Android 3. «Honeycomb»:**

- расширен список недавно запущенных приложений;
- изменяемый размер виджетов рабочего стола;
- поддержка внешних клавиатур, мышей, джойстиков/геймпадов;
- нативное воспроизведение аудио-файлов в формате FLAC;
- поддержка HTTP-прокси для каждой подключенной WiFi-точки;
- **Android 3.2 «Honeycomb»:**
 - внесены оптимизации для поддержки более широкого спектра планшетов;
 - добавлен специальный режим масштабирования приложений с фиксированным размером;
 - расширен API поддержки экрана для разработчиков;
 - лёгкий доступ приложений к файлам на SD-карте, например для синхронизации;
- **Android 3.2.1«Honeycomb»:**
 - незначительные исправления безопасности и стабильности, улучшения в Wi-Fi;
 - обновление для Android Market с автоматическими обновлениями и упрощённо-читаемым текстом Правил и Положений;
 - обновление Google Books;
 - улучшена поддержка Adobe Flash в браузере;
 - улучшенный предиктивный ввод на китайском;
- **Android 3.2.2** произошли мелкие улучшения и исправления для 4G Motorola Xoom;
- **Android 4.0.1 «Ice Cream Sandwich»:**
 - использование единой оболочки для планшетов, смартфонов и других устройств на базе OS Android;
 - Open Accessory API - API для интеграции с разнообразным оборудованием;
 - поддержка Real-time Transport Protocol API для аудио;
 - улучшенная многозадачность;
 - изменение панели уведомлений;
 - создание папок на рабочем столе для приложений;
 - масштабируемые виджеты;
 - изменённая панель блокировки с вынесением ярлыков для быстрого запуска камеры, доступа к функциям телефона (вызовов), доступа к текстовым сообщениям и меню для работы с электронной почтой;
 - улучшенный ввод текста и проверки орфографии;
 - улучшенный режим голосового надиктовывания текста;
 - система контроля использования интернет-трафика;

- программное улучшение камеры: внедрение режима панорамной съёмки, программная стабилизация изображения, эффекты в реальном времени при видеосъёмке;
 - поддержка снятия скриншотов (screen capture) средствами операционной системы;
 - обновлённый браузер с поддержкой вкладок, синхронизацией закладок Google Chrome. Также обновлено ядро WebKit и движок V8 с поддержкой Crankshaft;
 - расширенные возможности безопасности: разблокировка устройства через распознавание лица владельца, полное шифрование аппарата, ASLR, новый VPN API;
 - поддержка Wi-Fi Direct;
- **Android 4.0.2** исправлены ошибки для Galaxy Nexus;
- **Android 4.0.3** оптимизирована и исправлена найденных недочетов для смартфонов и планшетов;
- **Android 4.0.4** улучшена производительности камеры, увеличение стабильности, более плавный поворот экрана и некоторые исправления;
- **Android 4.1.1 «Jelly Bean»:**
 - обновленный интерфейс, использование технологии Project Butter для получения более плавного UI. Технология включает тройную буферизацию графического конвейера, чтобы добиться отсутствия скачков в частоте кадров при анимации интерфейса, а также технологию вертикальной синхронизации;
 - автоматическое масштабирование виджета и изменение расположения остального содержимого, если он не вмещается на рабочий стол;
 - обновление виртуальной клавиатуры, получившей систему предиктивного ввода и расширенный словарь. Поддержка новых языков;
 - возможность голосового ввода в автономном режиме, без использования подключения к сети Интернет;
 - слепые или слабовидящие получают возможность подключить к смартфону или планшету на Android 4.1 устройство ввода со шрифтом Брайля;
 - более информативная панель уведомлений;
 - обновленный поиск, благодаря которому на манер голосового помощника можно получить ответы на заданные голосовые вопросы;
 - добавлен сервис Google Now. Он будет снабжать пользователя полезной информацией, опираясь на календарь, историю поисковых запросов и маршруты перемещений телефона;
 - фотографии контактов теперь хранятся и отображаются в высоком качестве;
 - улучшенное приложение «Камера» добавляет некоторые возможности;

- новый интерфейс для планшетов, классический интерфейс Android 3.0-4.0.4 доступен через редактирование Build.Prop изменение DPI на значение ниже 14;
- **Android 4.1.2 «Jelly Bean»:**
 - исправлены ошибки и увеличена производительность;
 - увеличение и уменьшение размеров уведомлений в панели теперь выполняются одним пальцем;
 - добавлен ландшафтный режим рабочего стола в Nexus 7;
- **Android 4.2 «Jelly Bean»:**
 - одним из основных нововведений в Android 4.2 стала поддержка профилей, теперь на одном устройстве может использоваться несколько учетных записей, каждая со своими настройками, программами и личной информацией;
 - клавиатура теперь поддерживает жесты для ввода по типу популярного Swype. Для ввода слов теперь можно водить пальцем по клавиатуре выбирая необходимые буквы, а встроенный словарь будет стараться предоставлять нужные варианты. В целом, по заявлению Google, в Android 4.2 словари стали более точными и полными;
 - в приложении «Камера» появился новый режим съемки под название Photo Sphere, он позволяет создавать панорамы 360 градусов и публиковать их в Google+ или же в Google Maps, создавая таким образом свою версию Street View;
 - появилась поддержка беспроводной передачи видео и игр на совместимые телевизоры по технологии Miracast;
 - в панели оповещений теперь есть доступ к меню быстрых настроек;
 - появился новый режим ожидания Daydream, когда устройство подключено к док-станции или находится в режиме сна, оно может отображать на дисплее разную полезную информацию, новости, погоду или же просто фотографии из альбома;
 - Google Now также получило небольшое улучшение, теперь оно может сканировать почту Gmail на предмет нахождения различного релевантного контента, такого как билеты на самолет или приглашения на встречу, и создавать карточки с напоминанием;
 - появилась возможность размещать виджеты на экране блокировки;
 - обновлён интерфейс камеры;
 - обновление ядра Linux до ветки 3.4;
 - удален классический планшетный интерфейс, используемый в 3.0-4.0.4;
- **Android 4.2.1 «Jelly Bean»:**
 - устранена ошибка недостающего месяца (декабрь) в приложении «Контакты»;

- улучшена работа вибрации при уведомлениях;
- появилась возможность подключить к устройству HID геймпады и джойстики по Bluetooth;
- **Android 4.2.2 «Jelly Bean»:**
 - некоторые усовершенствования системы безопасности, исправление ряда программных ошибок;
 - в шторке центра уведомлений при загрузке теперь отображается не только заполняющийся прогресс-бар, но и проценты и время скачивания;
 - ярлыки быстрой активации Wi-Fi и Bluetooth в панели переключателей получили дополнительную опцию: теперь долгое нажатие иконки сразу включает беспроводной модуль, а одиночное нажатие отправляет пользователя в настройки;
 - появились новые звуки: при подключении устройства к станции беспроводной зарядки, новый сигнал оповещения о критическом уровне заряда батареи;
 - исправлены ошибки в передаче звука по Bluetooth;
 - повышена безопасность ADB;
- **Android 4.3 «Jelly Bean»:**
 - Keep и Hangouts теперь будет включен в прошивку и Google Apps по умолчанию;
 - недавно найденная ошибка под названием MasterKey исправлена;
 - Smart или Bluetooth 4.0 LowEnergy включен в новую прошивку, что повышает энергоэффективность устройства при работе по данному протоколу;
 - в области уведомлений теперь показаны все работающие приложения, даже в фоновом режиме. Отключить отображение значка в панели можно в настройках;
 - в настройках можно установить постоянную работу Wi-Fi для улучшенного гео-позиционирования;
 - AVRCP 1.3, доступный совместно с Bluetooth Smart, позволяет при помощи Bluetooth объединять два устройства и использовать их как управляющие друг другом;
 - доступна новая камера и галерея;
 - система теперь поддерживает OpenGL ES 3.0;
 - появилась скрытая настройка индивидуальных разрешений для приложений;
- **Android 4.3.1** устранены небольшие ошибки и улучшена стабильность работы системы и её быстродействия. Только для Nexus 7 (2013) LTE;
- **Android 4.4 «KitKat»:**

- постоянно активный голосовой помощник. Прикосаться к экрану не нужно, достаточно лишь произнести «Ok Google», затем можно дать команду или продиктовать, что нужно найти;
- отображение обложек и кнопок управления на экране блокировки при воспроизведении музыки или показе фильмов через Chromecast;
- кнопки навигации и панель уведомлений автоматически скрываются;
- более быстрое переключение между задачами и оптимизированное распределение памяти;
- приоритетность в телефонной книге. Контакты, с которыми пользователь общается чаще всего, показаны вверху списка. Прямо в приложении «Контакты» можно искать людей, места на карте и организации;
- умный определитель номера. Если номер не найден в «Контактах», телефон попытается определить по данным Google Maps, из какой организации звонят;
- центр общения. Приложение Hangouts позволяет вести переписку в чате, совершать и принимать видеозвонки, отправлять и получать SMS и MMS-сообщения и делиться GIF-анимацией;
- японские смайлики. В стандартную клавиатуру добавлены красочные миниатюрные картинки Emoji;
- поддержка облачных принтеров. Фотографии, документы и веб-страницы можно распечатывать на принтерах, подключенных к сервису Google Cloud Print, и на других принтерах, которые поддерживают печать через мобильные приложения;
- быстрое сохранение файлов в облако. Некоторые приложения (например, обновленный QuickOffice) поддерживают моментальную отправку файлов в Диск Google;
- поддержка Message Access Profile в автомобилях, оснащенных модулями Bluetooth;
- поддержка Chromecast;
- запуск веб-приложений через Chrome;
- отображение субтитров к фильмам в стандартном видеоплеере;
- встроенный сервис «Удаленное управление Android»;
- обновленный дизайн загрузчика файлов с сортировкой и настройкой способа отображения скачанных файлов: списком или миниатюрами;
- переключение между установленными лаунчерами через системные настройки;
- обновленное приложение электронной почты с папками, фотографиями аккаунтов и улучшенной навигацией;
- поддержка приложений, использующих инфракрасный порт;

- доступ к настройкам определения местоположения через значок в «Быстрых настройках»;
- настройка способа определения местоположения: точное или с меньшим расходом батареи. В системных настройках можно посмотреть, какие приложения пытаются определить местоположение;
- запуск приложений в песочнице Security-Enhanced Linux;
- поддержка шагомеров;
- осуществление NFC-платежей через Google Wallet и другие платежные системы. Статистика хранится в облаке или памяти устройства;
- экспериментальный предкомпилятор Android Runtime (ART);
- иммерсивный режим, который автоматически скрывает лишние данные на экране устройства во время игры, чтения или просмотра видео;
- **Android 4.4.1** устранены небольшие ошибки;
- **Android 4.4.2** улучшена камера Nexus 5, уменьшение лага затвора, более точная фокусировка и лучший баланс белого. Также включает исправление некоторых ошибок и улучшение производительности. NEXUS 7 2012 добавили «беспроводной монитор» Настройки - Экран;
- **Android 4.4.3 «KitKat»:**
 - исправление ошибки отключения передачи данных;
 - исправление ошибки, связанной с процессом камеры, который влиял на заряд аккумулятора;
 - исправление фокусировки в обычном и HDR-режимах;
 - другие исправления в работе камеры;
 - исправление проблемы, связанной с переходом устройства в режим "глубокого сна";
 - некоторые исправления в работе Bluetooth;
 - исправления ошибки случайной перезагрузки устройства;
 - исправление ошибки, когда ярлыки пропадали с рабочего стола после обновления приложения;
 - исправление в безопасности при USB-подключении;
 - устранение проблемы с автоматическим подключением к Wi-Fi;
 - исправления зависания в MMS, Email/Exchange, Calendar, People/Dialer/Contacts, DSP, IPv6, VPN при активации;
 - исправление в работе LED-индикатора при пропущенных вызовах;
 - исправление в работе субтитров;
 - исправление в отображении статистики передачи данных;
 - исправления в работе Интернет-телефонии;
 - исправления для соблюдения требования Федерального Агентства Связи (FCC).

2.4 iOS

iOS - операционная система на основе которой работает iPhone и другие устройства разработанные компанией Apple (iPod Touch, iPad). Ввод информации в iOS происходит по средствам касаний и жестов на экране устройства, а технология мультитач (Multi-touch) распознает манипуляции с использованием нескольких пальцев.

Основные составляющие операционной системы iOS представлена на рисунке 2.2.



Рисунок 2.2 - Окно iOS

1 Рабочий экран (Home Screen) - вмещает 16 элементов (иконок или папок).

2 Строка состояния (Status Bar) - отображает уровень сети или авиарежим, EDGE, Wi-Fi, Bluetooth, часы или блок экрана, индикатор заряда батареи телефона и гарнитуры, будильник, воспроизведение музыки и ГТТ.

3 Панель навигации с рабочими экранами и поиском (Spotlight).

4 Док (Dock) с 4-мя элементами, который остается в нижней области независимо от выбранного рабочего экрана.

В быту iOS называют прошивкой для iPhone или iPhone OS. С момента появления iPhone первого поколения операционная система постоянно обновляется, исправляются ошибки и появляются новые возможности.

Следует отметить, что свежие версии iOS перестают поддерживать «устаревшие» iPhone, поэтому обладатели первых моделей не имеют возможности установить свежую версию операционной системы, соответственно не получится запустить новые версии игр и приложений.

2.5 Eclipse

Eclipse является бесплатной программной платформой с открытым исходным кодом, контролируется организацией Eclipse Foundation. Написана на языке программирования Java и основной целью её создания является повышение продуктивности процесса разработки программного обеспечения. Претендует на статус наиболее популярной Java IDE и является единственным конкурентом такой мощной платформы как NetBeans.

Но в отличие от NetBeans который для создания элементов пользовательского интерфейса использует платформу-независимую библиотеку Swing, в Eclipse используется платформу-зависимая библиотека SWT - Standard Widget Toolkit.

IDE разработанные на базе платформы Eclipse применяются для создания программного обеспечения на различных языках программирования, так как Eclipse является платформой для разработки любых интегрированных сред программирования и расширений для себя же, по принципу "Плагины для Eclipse разрабатываются в самой Eclipse".

Особенности платформы Eclipse:

- кроссплатформенность - работает под операционными системами Windows, Linux, Solaris и Mac OS X;
- используя Eclipse можно программировать на множестве языков, таких как Java, C и C++, PHP, Perl, Python, Cobol и других;
- является фреймворком для разработки других инструментов и предлагает обширный набор API для создания модулей;
- используя подход RCP (Rich Client Platform) Eclipse является инструментом для создания практически любого клиентского программного обеспечения.

Работа над проектом Eclipse ведётся в нескольких направлениях, основные три - работа над платформой Eclipse, разработка Java IDE, разработка плагинов для расширения функциональности Eclipse.

Гибкость и расширяемость достигается благодаря модульности платформы.

Архитектура платформы Eclipse:

- основным элементом является исполняющая среда - Eclipse Runtime, в которой выполняются коды расширений и модулей. Она обеспечивает всю базовую функциональность платформы - управление расширениями и обновлениями, взаимодействие с операционной системой, обеспечение работы системы помощи;
- следующим элементом является собственно IDE - она отвечает за управление основными элементами программы, их расположением и настройками, управление проектами, отладку и сборку проектов, поиск по файлам и командную разработку.

В стандартную поставку Eclipse SDK включены два плагина - Java Development Tools или JDT, и Plug-in Developer Environment или PDE, таким образом мы получаем полностью готовую IDE для Java программирования и для разработки расширений для Eclipse.

Eclipse SDK - это минимальная версия, идеальная для знакомства с платформой и обучения. В дальнейшем, определившись с целями, вы можете скачать и использовать любую подходящую для ваших задач сборку, уже укомплектованную необходимыми расширениями.

Примеры других специализированных сборок Eclipse:

- Eclipse IDE for Java Developers - среда разработки на языке Java;
- Eclipse IDE for Java EE Developers - среда разработки веб приложений и корпоративных приложений с использованием технологии Java EE;
- Eclipse IDE for C/C++ Developers - функциональная IDE для программирования на C и C++;
- Eclipse IDE for JavaScript Web Developers - IDE для разработки веб приложений с использованием HTML, XML, JavaScript и CSS.

Eclipse IDE

Свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.

Eclipse служит в первую очередь платформой для разработки расширений, чем он и завоевал популярность: любой разработчик может расширить Eclipse своими модулями. Уже существуют Java Development Tools (JDT), C/C++ Development Tools (CDT), разрабатываемые инженерами QNX совместно с IBM, и средства для языков Ada (GNATbench, Hibachi), COBOL, FORTRAN, PHP и пр. от различных разработчиков. Множество расширений дополняет среду Eclipse менеджерами для работы с базами данных, серверами приложений и др.

Eclipse JDT (Java Development Tools) - наиболее известный модуль, нацеленный на групповую разработку: среда интегрирована с системами управления версиями - CVS, GIT в основной поставке, для других систем (например, Subversion, MS SourceSafe) существуют плагины. Также предлагает поддержку связи между IDE и системой управления задачами (ошибками). В основной поставке включена поддержка трекера ошибок Bugzilla, также имеется множество расширений для поддержки других трекеров (Trac, Jira и др.). В силу бесплатности и высокого качества, Eclipse во многих организациях является корпоративным стандартом для разработки приложений.

Eclipse написана на Java, потому является платформо-независимым продуктом, за исключением библиотеки SWT, которая разрабатывается для всех распространённых платформ (см. ниже). Библиотека SWT используется вместо стандартной для Java библиотеки Swing. Она полностью опирается на нижележащую платформу (операционную систему), что обеспечивает быстроту

и естественный внешний вид пользовательского интерфейса, но иногда вызывает на разных платформах проблемы совместимости и устойчивости приложений.

2.6 XML

XML - это расширяемый язык разметки (Extensible Markup Language), разработанный специально для размещения информации в World Wide Web, наряду с HTML, который давно стал стандартным языком создания Web-страниц. В отличие от HTML, вместо использования ограниченного набора определённых элементов вы имеете возможность создавать ваши собственные элементы и присваивать им любые имена по вашему выбору. Примечание: подразумевается, что читатель данной статьи хотя бы очень поверхностно знаком с языком HTML.

XML решает ряд проблем, которые не решает HTML, например:

- представление документов любого (не только текстового) типа, например, музыки, математических уравнений и т.д.;
- сортировка, фильтрация и поиск информации;
- представление информации в структурированном (иерархическом) виде.

В зависимости от уровня соответствия стандартам документ может быть "верно сформированным" ("well-formed"), либо "валидным" ("valid"). Вот несколько основных правил создания верно сформированного документа:

- каждый элемент XML должен содержать начальный и конечный тэг (либо пустой тэг типа <TAG />, который может нести информацию посредством своих атрибутов);
- любой вложенный элемент должен быть полностью определён внутри элемента, в состав которого он входит;
- документ должен иметь только один элемент верхнего уровня;
- имена элементов чувствительны к регистру.

Есть три основных способа сообщить браузеру, как отображать каждый из созданных вами XML-элементов:

- каскадная таблица стилей (Cascading Style Sheet - CSS) или расширяемая таблица в формате языка стилевых таблиц (Extensible Stylesheet Language - XSL);
- связывание данных. Этот метод требует создания HTML-страницы, связывания с ней XML-документа и установления взаимодействий HTML-элементов с элементами XML. В дальнейшем HTML-элементы автоматически отображают информацию из связанных с ними XML-элементов;
- написание сценария. Этот метод требует создания HTML-страницы, связывания с ней XML-документа и получение доступа к XML-элементам с помощью кода сценария JavaScript или VBScript.

Индивидуальный пользователь, компания или комитет по стандартам может определить необходимый набор элементов XML и структуру документа, которые будут применяться для особого класса документов. Подобный набор элементов и описание структуры документа называют XML-приложением или XML-словарём.

XML-приложение обычно определяется созданием описателя типа документа (DTD), который является допустимым компонентом XML-документа. DTD устанавливает и определяет имена элементов, которые могут быть использованы в документе, порядок, в котором элементы могут появляться, и доступные к применению атрибуты элементов. DTD обычно включается в XML-документ и ограничивает круг элементов и структур, которые будут использоваться. Примечание: приложение XML Schema позволяет разрабатывать подробные схемы для ваших XML-документов с использованием стандартного синтаксиса XML и является альтернативой DTD.

2.7 MySQL

MySQL - это одна из самых популярных и самых распространенных СУБД (система управления базами данных) в интернете. Она не предназначена для работы с большими объемами информации, но ее применение идеально для интернет сайтов, как небольших, так и достаточно крупных [4].

MySQL отличается хорошей скоростью работы, надежностью, гибкостью. Работа с ней, как правило, не вызывает больших трудностей. Поддержка сервера MySQL автоматически включается в поставку PHP.

Немаловажным фактором является ее бесплатность. MySQL распространяется на условиях общей лицензии GNU (GPL, GNU Public License).

Ранее для долговременного хранения информации мы работали с файлами: помещали в них некоторое количество строчек, а затем извлекали их для последующей работы. Задача длительного хранения информации очень часто встречается в программировании Web-приложений: подсчёт посетителей в счётчике, хранение сообщений в форуме, удалённое управление содержанием информации на сайте и т.д.

Между тем, профессиональные приёмы работы с файлами очень трудоёмки: необходимо заботиться о помещении в них информации, о её сортировке, извлечении, при этом не нужно забывать, что все эти действия будут происходить на сервере хост-провайдера, где с очень большой вероятностью стоит один из вариантов Unix - следовательно, нужно так же заботиться о правах доступа к файлам и их размещении. При этом объём кода значительно возрастает, и совершить ошибку в программе очень просто.

Все эти проблемы решает использование базы данных. Базы данных сами заботятся о безопасности информации и её сортировке и позволяют извлекать и размещать информацию при помощи одной строчки. Код с использованием

базы данных получается более компактным, и отлаживать его гораздо легче. Кроме того, не нужно забывать и о скорости - выборка информации из базы данных происходит значительно быстрее, чем из файлов.

2.8 Oracle Database 11g Standard Edition

Oracle Database 11g является единственной СУБД, предназначенной для распределенных вычислительных сред (Grid). Выпуская Oracle Database 11g, Oracle предоставляет клиентам возможность управлять корпоративной информацией проще, чем когда-либо прежде, лучше понимать свой бизнес и быстрее внедрять инновации. Oracle Database 11g обеспечивает высочайшие уровни производительности, масштабируемости, доступности, безопасности и удобства управления в сетях распределенных вычислений на базе стандартных серверов и систем хранения. СУБД Oracle Database 11g предназначена для эффективного развертывания на базе различных типов оборудования, от небольших blade-серверов до мощных симметричных многопроцессорных серверных систем и кластеров любых масштабов. Этот продукт предлагает функции автоматизации управления, обеспечивая удобство и эффективность эксплуатации. Уникальные возможности управления всеми данными предприятия - от традиционной бизнес-информации до XML и трехмерных пространственных данных - делает Oracle Database 11g идеальным решением для эффективной реализации OLTP приложений, хранилищ данных и систем управления контентом.

Стоимость решения - от 500 000 тенге.

2.9 Firebird

Firebird (FirebirdSQL) - компактная, кросс-платформенная, свободная система управления базами данных (СУБД), работающая на GNU/Linux, Microsoft Windows и разнообразных Unix платформах.

В качестве преимуществ Firebird можно отметить многоверсионную архитектуру, обеспечивающую параллельную обработку оперативных и аналитических запросов (это возможно потому, что читающие пользователи не блокируют пишущих), компактность (дистрибутив 5Mb), высокую эффективность и мощную языковую поддержку для хранимых процедур и триггеров.

Среди недостатков: отсутствие кеша результатов запросов, полнотекстовых индексов.

Стоимость - бесплатно.

2.10 Вывод по определению системных и технических разработок

Безусловным лидерством на рынке мобильных операционных систем на сегодняшний день пользуются две: Android и iPhone. Сторонников как одной, так и другой системы достаточно много, однако, несмотря на свою «молодость», некоторые преимущества Android перед iPhone очевидны.

Первым существенным преимуществом ОС Android является отсутствие ее привязки к аппаратному обеспечению. Это позволяет каждому пользователю мобильного устройства выбрать именно тот его тип и вид, который ему более привычен, а не адаптироваться к тому, который навязывает, например Apple. Операционная система iPhone такой возможности не предоставляет, поскольку может быть установлена исключительно на аппараты iPod Touch, iPad и iPhone, а их форм-фактор и модификации определяются корпорацией Apple. Это очень ограничивает возможности удовлетворения предпочтений пользователя, который привык, например, пользоваться аппаратами определенной фирмы, имеющими специфические аппаратные возможности, форму корпуса или его цвет.

Вторым преимуществом ОС Android стал открытый исходный код. Это дает возможность не только очень просто устанавливать нужные пользователю приложения, темы оформления или игры от сторонних разработчиков, но и загружать разнообразные сборки ОС Android, которые призваны обеспечить аппарату именно ту функциональность, которая необходима каждому конкретному владельцу смартфона. А также существует возможность заказать мобильное приложение для Android. В этом плане iPhone находится в проигрышной позиции, поскольку загрузка приложений может быть осуществлена лишь с использованием специальной программы iTunes, а устанавливаемые продукты должны быть лицензионными.

Открытый исходный код позволяет пользователю удалить или заменить абсолютно любое системное приложение на то, функции которого ему больше необходимы. Эта возможность позволяет не только экономить доступную память аппарата, но и наполнить его лишь теми программами, которые действительно актуальны. Касаясь iPhone можно сказать, что такая возможность здесь практически отсутствует, так как предустановленные фирмой-производителем программы ни удалить, ни отключить нельзя, а расширить функциональность аппарата можно только путем установки поверх них различных расширений, причем «родное» приложение останется на своем месте, и будет занимать никогда не лишнее место в памяти устройства.

Третье преимущество ОС Android состоит в огромной гибкости настроек рабочего стола, доступной возможности использования разнообразных виджетов и даже создания нескольких рабочих столов, каждый из которых возможно настроить индивидуально. Это дает владельцу смартфона возможность устроить работу с аппаратом максимально рационально и комфортно, а также придать исключительный, индивидуальный и

неповторимый внешний облик ОС Android, не ограничиваясь предустановками от компании-производителя.

Решение Oracle является неприемлемым с точки зрения стоимости лицензии, а также необходимой аппаратной части для работы этой СУБД.

СУБД FireBird по сравнению с MySQL похожи по функциональности, однако FireBird не поддерживает кэширование запросов, что означает низкую скорость работы на чтение данных. Также FireBird обладает намного меньшим сообществом в Интернет, что означает увеличение времени поиска решений для возможных проблем (Рисунок 2.3).

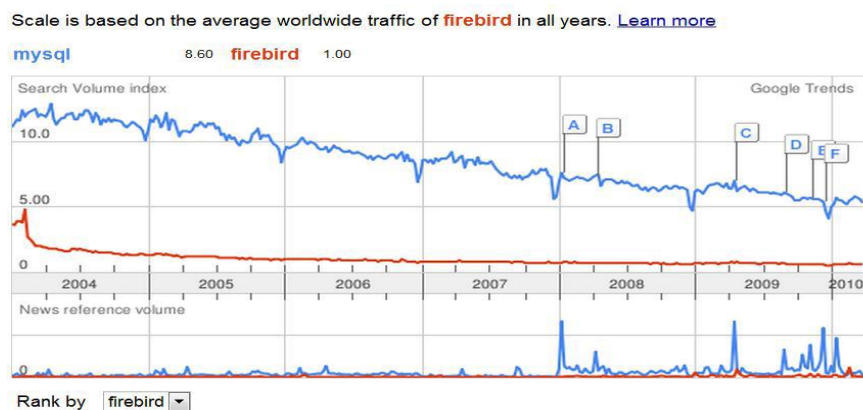


Рисунок 2.3 - Сравнение мирового поискового трафика по запросам «mysql» и «firebird»

MySQL полностью совместима с сетевой ОС Ubuntu Server 8.04. Поддерживается ветвь MySQL 5.0.x, которая является наиболее стабильной на данный момент.

MySQL имеет двойное лицензирование. MySQL может распространяться в соответствии с условиями лицензии GPL. Однако по условиям GPL, если какая-либо программа включает исходные коды MySQL, то она тоже должна распространяться по лицензии GPL. Это может расходиться с планами разработчиков, не желающих открывать исходные тексты своих программ. Для таких случаев предусмотрена коммерческая лицензия, которая также обеспечивает качественную сервисную поддержку.

Это означает, что в рамках разрабатываемого приложения возможно использовать бесплатную версию MySQL Community Edition 5.0.10.

MySQL имеет API для языков Delphi, C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk и Tcl, библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC.

3 Постановка задачи

3.1 Назначение и основные возможности проектируемого приложения

Мобильное приложение создается для клиентов компании простого в использовании и эксплуатации.

Основными составляющими проектируемого мобильного приложения являются:

- удобная система просмотра и оплаты своего счета;
- возможность отправки заявки на подключение/отключение услуг компании;
- простой и понятный интерфейс.

Проект имеет потенциал полноценной «точки входа» в мир приложений Play Market, для этого проект обладает всеми необходимыми технологическими и техническими данными.

3.2 АО «Казахтелеком»

Акционерное Общество «Казахтелеком» - ведущий оператор связи Республики Казахстан, предоставляющий широкий спектр инфокоммуникационных услуг. Центральный офис компании расположен в столице Казахстана - городе Астана. В компании работают около 30 тысяч человек. АО «Казахтелеком» имеет региональные подразделения в каждой области страны и обеспечивает предоставление услуг связи по всей территории страны. 51% акций АО «Казахтелеком» принадлежит АО «ФНБ «Самрук-Казына».

Основные направления деятельности - Телекоммуникационные услуги.

АО «Казахтелеком» создано и действует в соответствии с Гражданским кодексом Республики Казахстан от 27 декабря 1994 года, Указом Президента Республики Казахстан, имеющим силу Закона «О хозяйствующих товариществах» от 2 мая 1995 года, Постановлением Кабинета Министров Республики Казахстан № 666 от 17 июня 1994 года «О создании национальной акционерной компании «Казахтелеком».

Органами управления Общества являются:

- высший орган управления - Общее собрание акционеров;
- наблюдательный орган - Наблюдательный совет;
- исполнительный орган - Правление;
- контрольный орган - ревизионная комиссия.

Общество распространяет свою деятельность по всей территории Республики Казахстан и состоит из филиалов, имущество которых передано в Уставный фонд Общества. Филиалы не являются юридическими лицами.

Филиалы осуществляют свою деятельность на основе Положения о филиале, утвержденного Общим собранием акционеров. Каждый филиал состоит из структурных подразделений. Структура, штатное расписание всех подразделений определяется в соответствии с Положением о филиале, а также решениями Общего собрания акционеров, Наблюдательного совета, приказами, распоряжениями и указаниями Правления в пределах компетенции.

Руководство текущей деятельностью филиалов зависимых и дочерних предприятий и других подразделений осуществляет Генеральный директор, назначаемый и смещаемый с должности Наблюдательным советом. Полномочия Генерального директора определяются доверенностью, полученной от Общества, а также на основе контракта о найме.

В число организационной структуры входят следующие службы:

- производственные службы:
 - служба материального обеспечения;
- коммерческая служба:
 - отдел маркетинга;
 - отдел сбыта;
- финансовая служба:
 - финансово - экономическая служба;
 - отдел бухгалтерского учета;
- служба главного инженера;
- отдел информационных технологий;
- отдел кадров;
- управляющий делами.

3.3 Общая архитектура приложения

Дальнейшее изложение этапов создания конечного приложения будет основываться на итерационном подходе, в основе которого лежит с постепенное усложнение и детализация первоначальной структуры приложения [5].

В основе разрабатываемой системы лежит архитектура «клиент-сервер», в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемых серверами, и заказчиками услуг, называемых клиентами. В качестве среды взаимодействия клиента с сервером используется Интернет (Рисунок 3.1).



Рисунок 3.1 - Общая архитектура приложения

Основными достоинствами архитектуры «клиент-сервер» являются:

- возможность, в большинстве случаев, распределить функции вычислительной системы между несколькими независимыми компьютерами в сети. Это позволяет упростить обслуживание вычислительной системы. В частности, замена, ремонт, модернизация или перемещение сервера, не затрагивают клиентов;
- все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа;
- отказоустойчивость. Так как данные содержатся в основном на серверах, в них можно использовать систему отказоустойчивого хранения данных (т. е. RAID, Redundant Array of Independent Disks - матрица независимых дисковых накопителей с избыточностью) для того, чтобы избежать потери данных из-за поломок дисковых накопителей или системных отказов. Это позволяет сделать сервер более надёжным и снизить вероятность его простоя;
- позволяет объединить различные клиенты. Использовать ресурсы одного сервера часто могут клиенты с разными аппаратными платформами, операционными системами и т.п.

3.4 Структура приложения

Приложение - программный продукт, ориентированный на разного уровня пользования компании. Приложение позволяет клиентам в любое время

просматривать и оплачивать счет. Для этого им нужно подключение к сети интернет и сумма денег на платежной карте. Кроме того отправлять на сервер заявки о подключении и отключении услуг компании.

Третье новшество приложения - это просмотр новостей компании.

Перед собой поставила следующую цель: разработать оптимальное приложение, способное подключаться к тестирующей базе данных и хранить данные.

Реализация работы программного продукта представлена на рисунке 3.2.

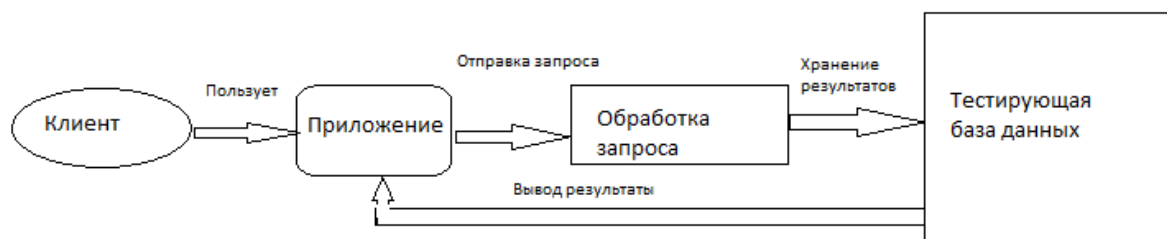


Рисунок 3.2 - Схема реализации приложения

Для разрабатываемого приложения требуется две категории группы лиц:

- программист;
- пользователи системы.

Программист системы имеет следующие права:

- просмотр заявок, оставленных в базе данных;
- вносить изменения в приложение.

Пользователи системы могут выполнять следующие действия:

- просматривать информацию о своем счете;
- оплачивать счет через платежную карту;
- отправлять заявки на подключение и отключение услуг;
- просматривать новости компании.

Пользователем системы может стать любой человек, зашедший в приложение.

3.5 Структура БД

В ходе реализации проекта подключение к главной базе данных компании не возможно. но для выполнения подключения приложение существует тестирующая база данных, которая осуществляет сохранение и проверку новых программных продуктов компании.

Для это они используют СУБД MySQL, которая находится в открытом доступе.

В период прохождения производственной практики в отделе информационных технологий ДИС АО "Казахтелеком", был получен доступ к их тестирующей базе. Реализация подключения изображено на рисунке 3.3.



Рисунок 3.3 - Архитектура тестирующей базы данных

Этап физического проектирования БД направлен на разработку методов взаимодействия с БД, которые позволяют обеспечить эффективность (высокую скорость) запросов к базе данных, т.е. каким образом, имея в виду особенности конкретной СУБД, расположить данные во внешней памяти, создания каких дополнительных структур может быть необходимо и т.д. Индекс - объект базы данных, создаваемый с целью повышения производительности выполнения запросов. Таблицы в базе данных могут иметь большое количество строк, которые хранятся в произвольном порядке, и их поиск по заданному значению путем последовательного просмотра таблицы строка за строкой может занимать много времени. Индекс формируется из значений одного или нескольких столбцов таблицы и указателей на соответствующие строки таблицы и, таким образом, позволяет находить нужную строку по заданному значению. Ускорение работы с использованием индексов достигается в первую очередь за счёт того, что индекс имеет структуру, оптимизированную под поиск - например, сбалансированного дерева.

В таблицах баз данных используется распространенный движок (система хранения) MyISAM. Таблицы MyISAM хорошо подходят для использования в WWW и других средах, где преобладают запросы на чтение. Таблицы типа MyISAM показывают высокую скорость работы при выборках SELECT. Во многом это связано с отсутствием поддержки транзакций и внешних ключей. Эта система хранения поддерживается многими СУБД и является оптимальной для разрабатываемой системы.

На рисунке 3.4 изображено уточнение (детализация) процесса прохождения запроса к серверу баз данных. В этот процесс также включена операционная система, которая работает как «мост» между СУБД и физическим хранилищем данных.



Рисунок 3.4 - Детализация процесса прохождения запроса к серверу БД

4 Результаты реализации проекта

В данной части приводится описание разработанного приложения: описание архитектуры приложения с кратким пояснением использованных шаблонов проектирования, описание разработанных модулей приложения, описание разработанных интерфейсов с кратким обзором процесса их создания.

4.1 Архитектура приложения

Под архитектурой приложения понимается представление, которое даёт информацию о компонентах составляющих систему, о взаимосвязях между этими компонентами и правилах, регламентирующих эти взаимосвязи.

Для разработки насыщенных мобильных приложений можно использовать различные технологии. Правильный выбор комплекса технологий позволяет сократить время разработки приложений и предоставить пользователям функционально богатые мобильные приложения.

К современным мобильного приложениям предъявляются все более высокие требования. Они должны уметь работать с учетными данными пользователей, обеспечивать загрузку контента и потокового видео. Эти задачи требуют использования технологий, ускоряющих процесс разработки и в то же время обеспечивающих поддержку всех наиболее востребованных функций. Разработчики приложений должны уметь правильно выбирать комплексы технологий, позволяющие наилучшим образом решать поставленные задачи.

Eclipse - это бесплатная программная платформа с открытым исходным кодом, контролируется организацией Eclipse Foundation. Написана на языке программирования Java и основной целью её создания является повышение продуктивности процесса разработки программного обеспечения. Претендует на статус наиболее популярной Java IDE и является единственным конкурентом такой мощной платформы как NetBeans. Технология Java, применяемая на стороне сервера, обеспечивает такие возможности, как подключение к реляционным СУБД, многопоточная обработка запросов на обслуживание и оптимальное масштабирование в случаях увеличения нагрузки. Совместное использование этих двух технологий даёт мощную техническую основу для создания RIA-приложений.

В основе серверной части приложения лежит фреймворк Qt Framework. Фреймворк - программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

Qt Creator - одна из самых популярных и надежных основ для построения мобильных приложений любой сложности. QT используется во многих крупных корпорациях, таких как IBM, Adobe, Panasonic, Samsung и многие другие.

В основе простейшего мобильного приложения лежит архитектура MVC, сокращенное от «model-view-controller», которое переводится как «модель-представление-контроллер».

Можно сказать, что MVC является «сердцем» любого приложения на основе Flex. Суть этой архитектуры сводится к разделению модели данных (model), пользовательского интерфейса (view) и управляющей логики (controller).

На рисунке 4.1, изначально запрос пользователя поступает на тестирующий сервер, который определяет, к какому мобильному приложению относится запрос. Далее, запрос транслируется на контроллер, который, как правило, запрашивает некоторую информацию из базы данных. Для этого контроллер направляет запрос в модель, в ответ получает массив данных, транслирует полученные данные от модели в графический интерфейс и после всех этих операций возвращает готовый документ тестирующему серверу. Тестирующий сервер на основе полученных данных формирует HTTP-ответ, который содержит запрашиваемый документ и ряд параметров в заголовке документа [6].

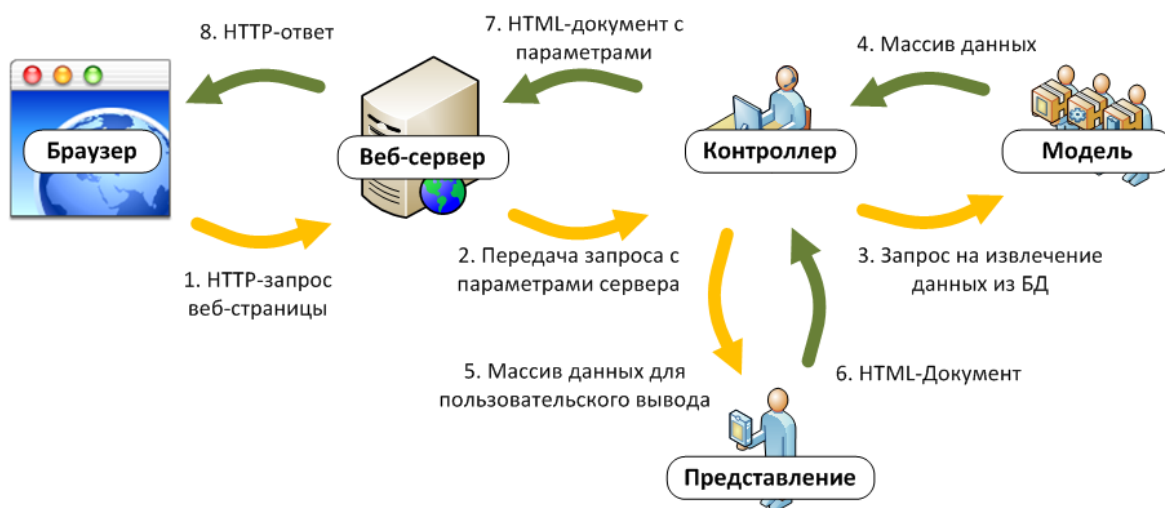


Рисунок 4.1 - Архитектура MVC

4.2 Разработка пользовательского интерфейса

4.2.1 Разработка интерфейса

Внешний интерфейс приложения будет состоять из 4 разделов:

- 1 начать (Get Started);
- 2 стиль (Style);
- 3 шаблоны (Patterns);
- 4 строительные блоки (Building Blocks).

В свою очередь раздел «Начать» поделен на 3 части. Первая часть носит название «Vision», и не несет в себе информацию, где разработчика подталкивают к тому, чтобы интерфейс его ПП был максимально красив, удобен и довольно прост в понятие. Чтобы приложение было оригинально, помогало решать поставленную задачу быстро и легко.

Вторая часть называется «Design» - методы дизайна или методы проектирования. Здесь также содержатся довольно общие рекомендации для разработчика, с образцами на примерами удобной версией Android 2.2 Froyo.

Третья часть именуется «UI» и описывает основные системные элементы интерфейса ОС, такие как панель навигации, шторка уведомлений и т.д.

Раздел «Стиль» уже содержит непосредственную информацию по оформлению интерфейса, какие цвета, шрифты, иконки рекомендуется использовать.

Раздел «Шаблоны» содержит информацию по оформлению различных элементов интерфейса, а также рекомендации о навигации в приложении и взаимодействии с ним.

Раздел «Строительные Блоки» несет в себе информацию о формах используемых в приложении.

4.2.2 Оформление

На Android работают сотни миллионов телефонов, планшетов и других устройств с самыми разнообразными размерами экрана и форм-факторами. Используя преимущества гибкой графической системы Android, есть возможность создания приложения, которые изящно смотрятся на различных устройствах, от маленьких телефонов до больших планшетов. На рисунке 4.2 представлены основные форм-факторы устройств.

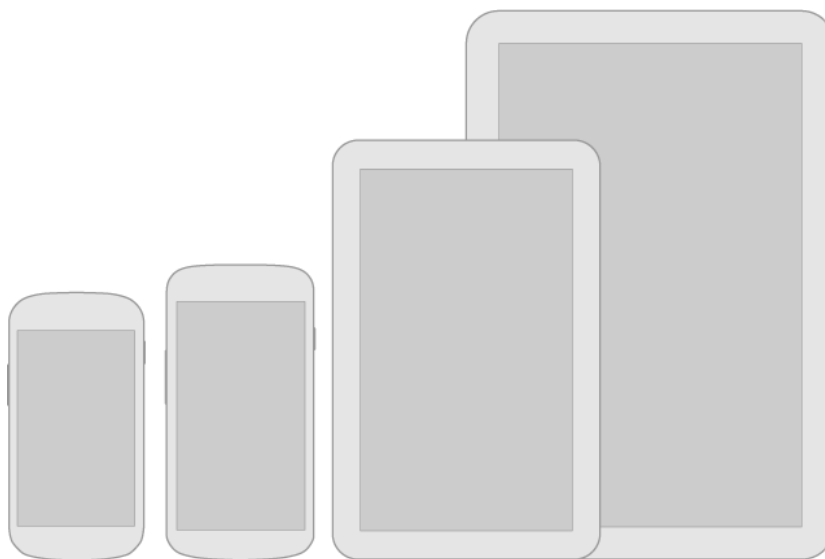


Рисунок 4.2 - Форм-факторы устройств

Темы оформления - механизм Android для приведения приложений к основным стилям. Стил ь определяет визуальные свойства элементов, составляющие пользовательский интерфейс, такие как цвет, высота и размера шрифта. В целях закрепления единого стиля между всеми приложениями на платформе Android, Google предоставляет две системных темы, которые вы можете выбрать при создании приложений:

- Holo Light.
- Holo Dark.

Применение этих помогут вам создавать приложения, которые соответствуют общему визуальному стилю Android.

Выбирается тема, наиболее соответствующая потребностям и эстетики дизайна для приложения. Для создания желаемого создания более оригинального облика для приложения, то используют одну из системных тем в качестве отправной точки для идеи.

Для отображения активности, отключённой кнопки или нажатия на нее используйте различные цвета, а также эффекты осветления и затемнения.

На рисунке 4.3 представлены стандартные шаблоны кнопок.

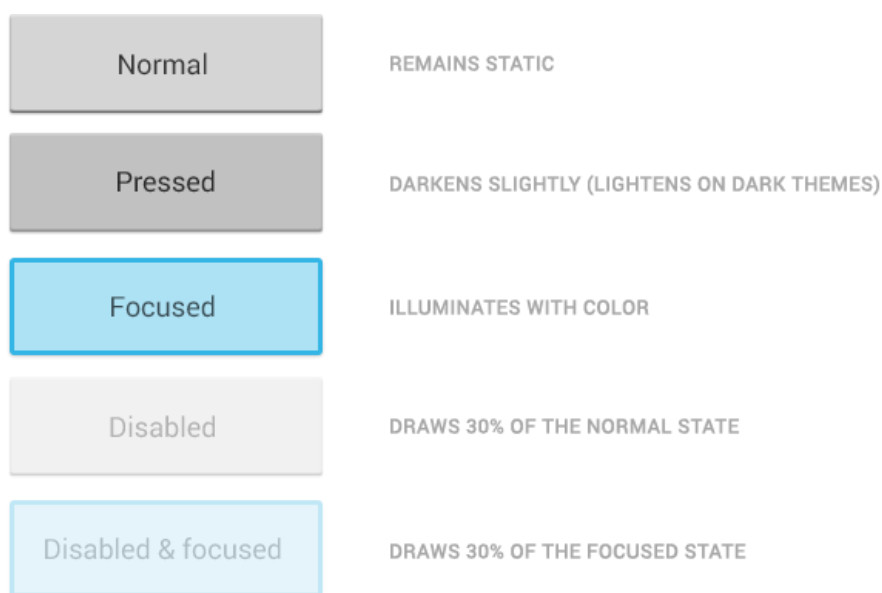


Рисунок 4.3 - Шаблоны кнопок

Использовать жесты любой сложности для навигации в приложении или же совершения каких-либо действий.

При использовании прокручиваемых элементов интерфейса, использовать эффект границы экрана.

Устройства различаются не только по физическими размерами экранов, но и плотностью пикселей на нем (DPI). Для упрощения работы дизайнеров существуют несколько заготовленных образцов для телефонов и планшетов,

благодаря чему вам не нужно беспокоиться как будет выглядеть ваше приложения на каждом отдельном устройстве.

Существуют DPI различных размеров, для простоты они поименованы LDPI, MDPI, HDPI, XHDPI, XXHDPI и XXXHDPI, и изначально в прошивке каждого устройства прописан соответствующий ему уровень плотности.

Необходимо всегда учитывать данные макеты экрана, так как в противном случае ваше приложение может исказиться и полностью испортиться интерфейс ПП, а также впечатление о нем. Однако это не составит никакого труда, потому что все шаблоны уже заложены в инструментарий для разработчиков, который автоматически подгонит интерфейс под требуемую плотность, если же вас что-то будет не устраивать, то вы непосредственно в процессе разработки, в графическом виде сможете все исправить и организовать так, как вам требуется.

На рисунке 4.4 представлены возможные варианты плотности экрана.

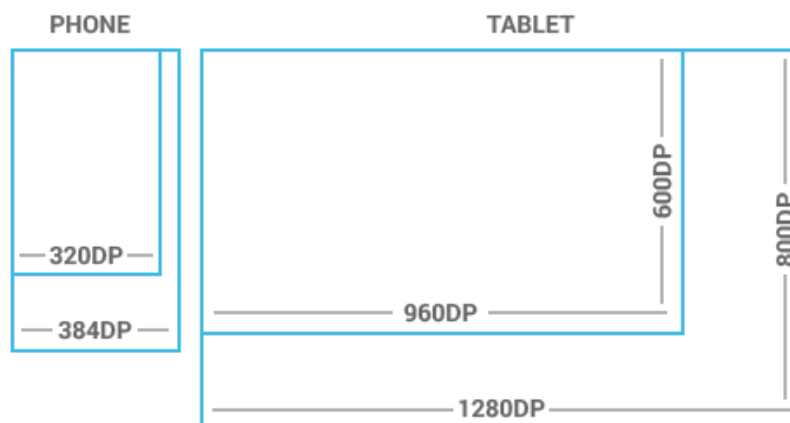


Рисунок 4.4 - Макеты плотности экрана

Все элементы интерфейса выполняются в размере кратном 48dp, то есть от 48x48 dp и выше. Данный размер соответствует примерно 9мм. на экране и является оптимальным по удобству. На рисунках 4.5 и 4.6 отображены рекомендуемые размеры для элементов интерфейса.

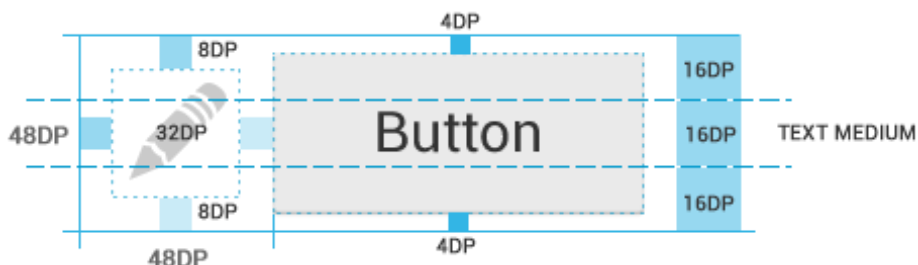


Рисунок 4.5 - Пример значка и кнопки

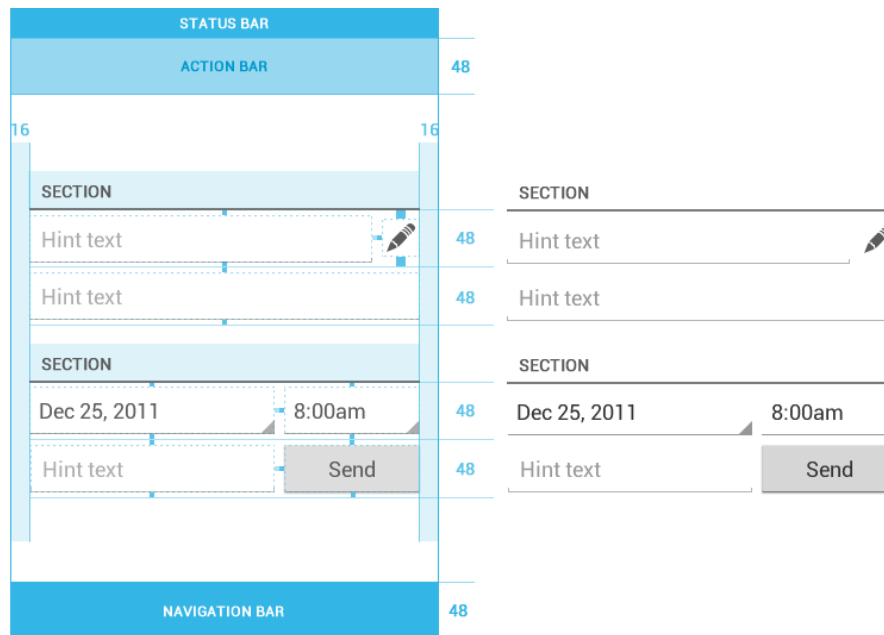


Рисунок 4.6 - Пример элементов размером 48dp в интерфейсе

В качестве системного шрифта в Android версии 2+ используется шрифт Roboto различных видов (Roboto Thin, Roboto Light, Roboto Regular, Roboto Medium, Roboto Bold, Roboto Black, Roboto Condensed Light, Roboto Condensed и Roboto Condensed Bold).

Android UI использует следующие цветовые стили по умолчанию: `textColorPrimary` и `textColorSecondary`. Для легких тем используется `textColorPrimaryInverse` и `textColorSecondaryInverse`. Рамки, используемые вокруг текста также поддерживают светлую и темную темы, а также имеют графическую обратную связь при нажатии. Размер же варьируется от 12sp до 22sp. На рисунке 4.7 изображены примеры светлой и темной тем текста.

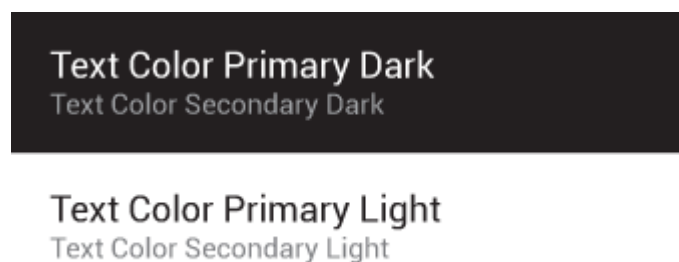


Рисунок 4.7 - Примеры светлого и темного текстов

При работе с цветом предлагается использование так называемых «Нюансов», которая изображена на рисунке 4.8.

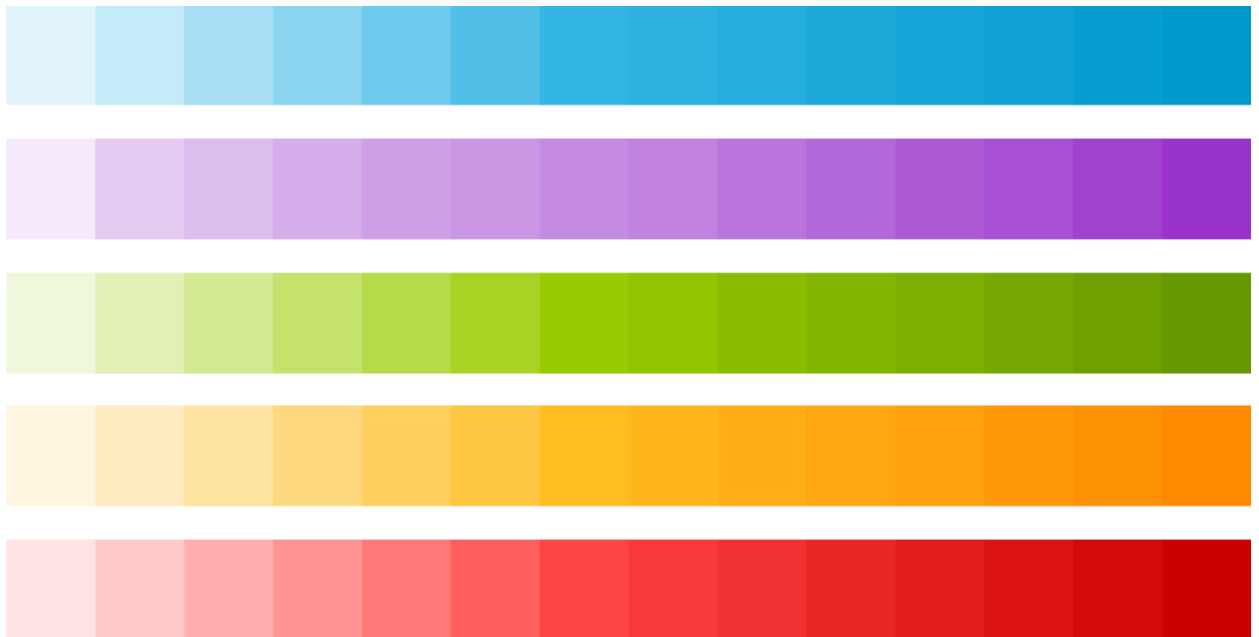


Рисунок 4.8 - Палитра цветов

Значок (иконка) приложения - небольшой графический ярлык расположенный в меню приложений или же вынесенный на один рабочих столов домашнего экрана, служащий для запуска приложения, а также его графического отображения в системе. При проектировании значка необходимо учитывать, что ваше приложение может быть установлено на различных устройствах, с различной плотностью экрана, а также необходимо его отображение в магазине Google Play. Поэтому используя макеты плотности экрана создайте несколько иконок разных размеров, для разных экранов. За отправную точку следует брать экраны с MDPI плотностью, с дальнейшим увеличением в пропорция равных 1,5x для HDPI, 2x для XHDPI, 3x для XXHDPI и 4x для XXXHDPI. Исходный размер должен быть равен 48dp, в результате на экранах с плотность MDPI размер значка будет равен 48x48 точек, для экранов HDPI - 72x72 точек, для XHDPI - 96x96 и т.д. Для публикации в Google Play также требуется изображение размером 512x512 точек. На рисунке 4.9 приведены размеры значка приложения.



Рисунок 4.9 - Пример размеров значка

Для значков различных действий непосредственно в приложении рекомендуется использование стандартных Нolo-иконок, предоставляемых

Google в векторном формате для графического редактора Adobe Illustrator. Физический размер таких элементов должен составлять 32x32 точек, а видимая часть 24x24.

Цвета также рекомендуется использовать стандартные. Более светлый (#333333) для темного фона, с прозрачностью 40% и более темный (#FFFFFF) для светлого фона, с прозрачностью 20%.

На рисунке 4.10 показаны примеры иконок действий и их размер.

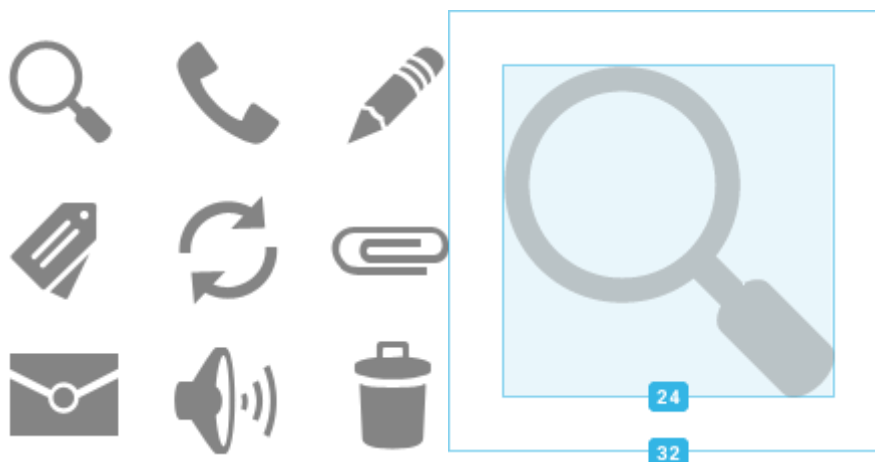


Рисунок 4.10 - Примеры иконок для действий

При использовании контекстных иконок, следует использовать физический размер равный 16x16 с видимой частью 12x12, как показано на рисунке 4.11.

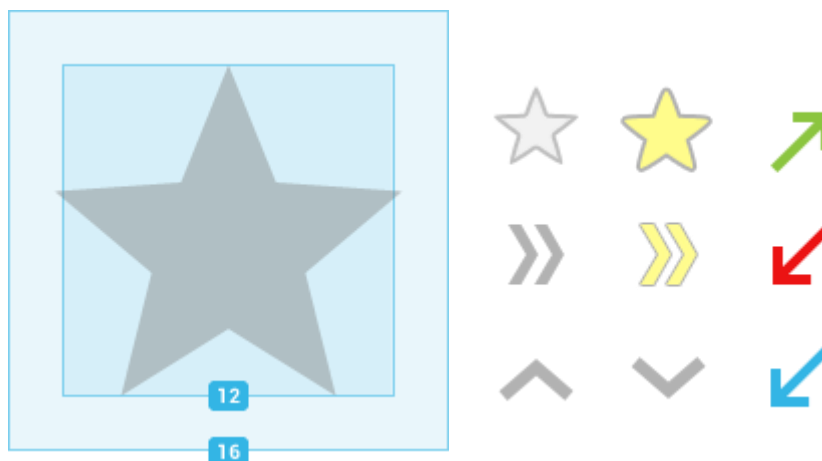


Рисунок 4.11 - Пример контекстной иконки

Если использовать систему уведомлений, то для отображения в шторке уведомления требуемой информации, также нужен значок размером 24x24 белого цвета.

При создании элементов интерфейса рекомендуется использование векторных элементов там, где это возможно, с максимальным избеганием растровых изображений. Это поможет избежать проблем с оптимизацией для различных размеров экранов и различной плотности.

При названии файлов лучше использовать префиксы, тогда иконки будут группироваться внутри директории и их поиск существенно облегчиться.

В коде программы также лучше разбить все элементы интерфейса на отдельные группы, что снизит затраты времени в случае необходимости что-либо исправить.

Например:

```
res /...
  drawable - ldpi /...
    finished_asset . png
  drawable - mdpi /...
    finished_asset . png
  drawable - hdpi /...
    finished_asset . png
  drawable - xhdpi /...
    finished_asset . png
```

При использовании текста в интерфейсе, описании его элементов и прочего, старайтесь максимально сократить его размер и упростить его восприятие. Избегайте использования ненужных слов, несущих не требующуюся информацию, сокращайте размер текста используя цифры вместо букв. Однако ненужно переусердствовать, если данный текст не отражает всю суть информации, которую требуется донести, то лучше сделать его более объемным, но пользователь полностью поймет, что от него хотят.

4.2.3 Принципы навигации

Главным элементом управления с современных смартфонах и планшетах является сенсорный экран. У ОС Android есть набор жестов, на которые особым образом реагирует система. Некоторые из них зарезервированы системой, такие как нажатие кнопки «домой» на навигационной панели, другие же зависят от приложения, запущенного в данный момент. Однако для во избежание путаницы пользователя рекомендуется использование стандартных действий и жестов, работающих в системе.

Так, например, одиночное, кратковременное нажатие активирует элемент, на который направленно данное действие. Длительное удержание активирует выбор объекта или нескольких. Длительное нажатие и движение пальца по экрану - перемещение объекта. «Свайп» (сдвиг) в сторону вызывает переключение на соседнюю вкладку. Двойное нажатие на активирует увеличение масштаба в 2 раза (например, в галереи, при просмотре изображений), а «Свайп» вверх или вниз вызывает пролистывание списка.

Разведение или сведение двух пальцев вызывает увеличение, или уменьшение изображения соответственно.

Возможные жесты показаны на рисунке 4.12.

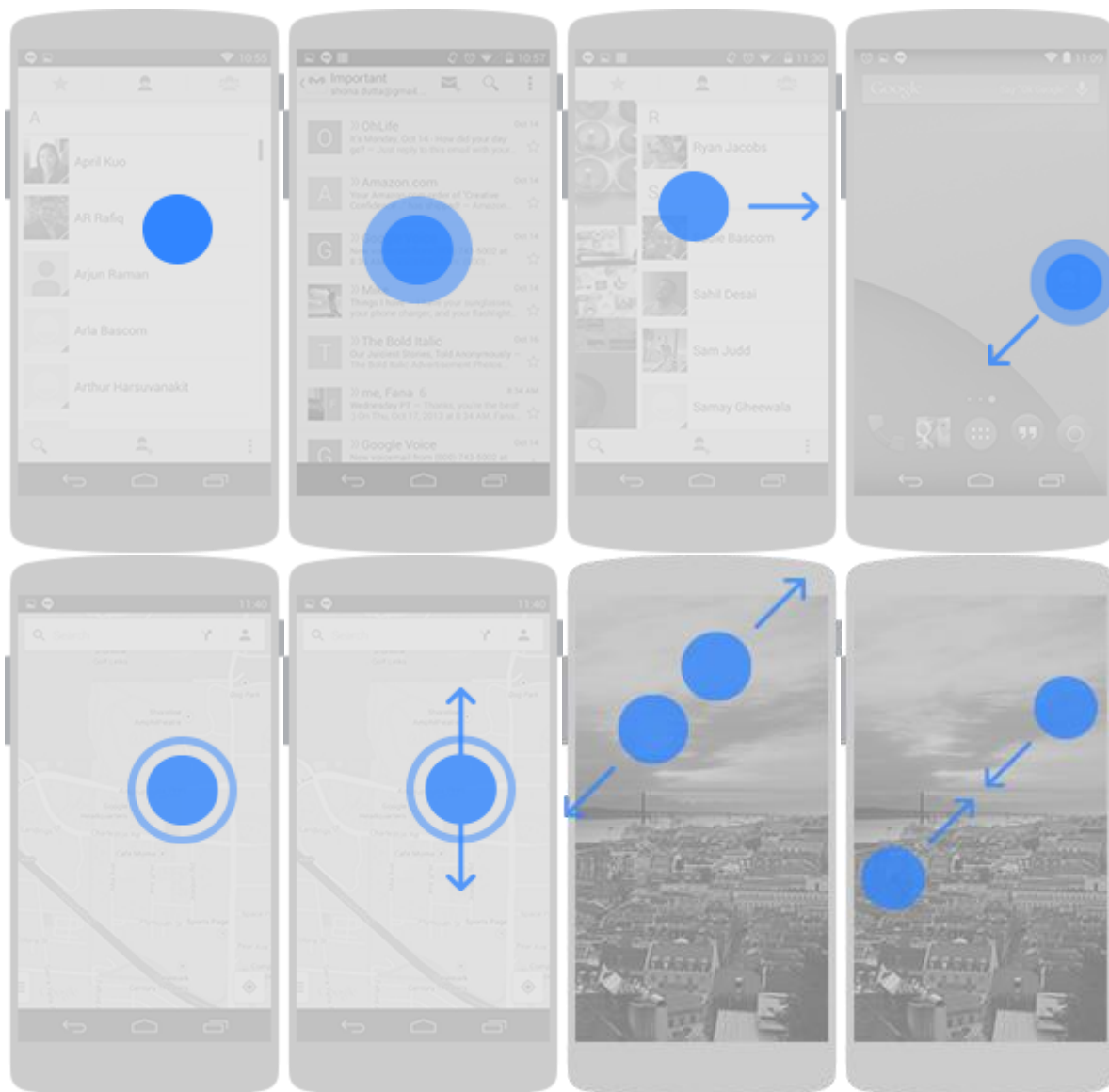


Рисунок 4.12 - Пример жестов

Построение правильно навигации - одна из главных задач и проектировании интерфейса ПП, так как она должна полностью совпадать с навигацией в ОС и не вводить пользователя в положение, когда она не знает, что вызовет нажатие той или иной кнопки.

Должное внимание при этом следует уделить клавише «назад». Она должна выполнять следующие функции:

- если пользователь находится в на главном экране приложения, то нажатие клавиши «назад» вызовет закрытие приложения;

- если пользователь находится в какой-либо вкладке ПП, на 1 уровень, то нажатие клавиши возвращает его на главный экран приложения;
- если пользователь последовательно перешел на несколько уровней, то нажатие клавиши возвращает его на предыдущий уровень (Рисунок 4.13).

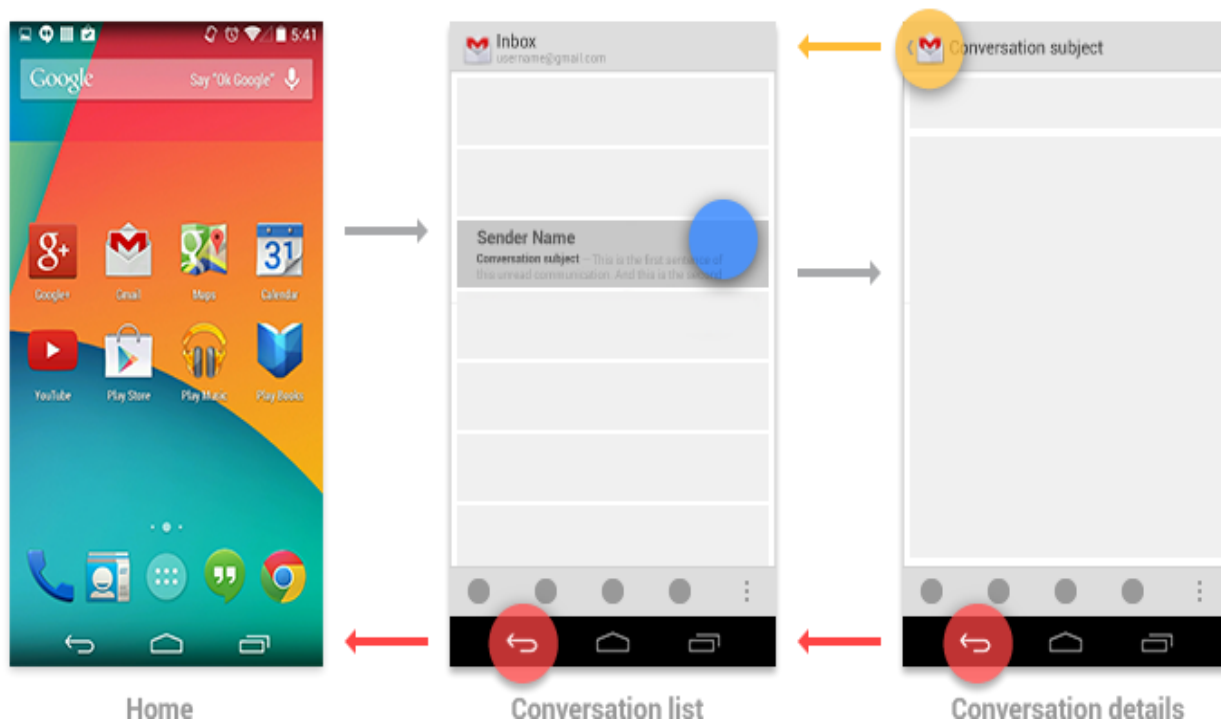


Рисунок 4.13 - Нажатие клавиши «назад» при нахождении на 1 уровне

В случае моментальной необходимости возврата на главный экран приложения, требуется нажать на его иконку, сверху, слева (Рисунок 4.14).

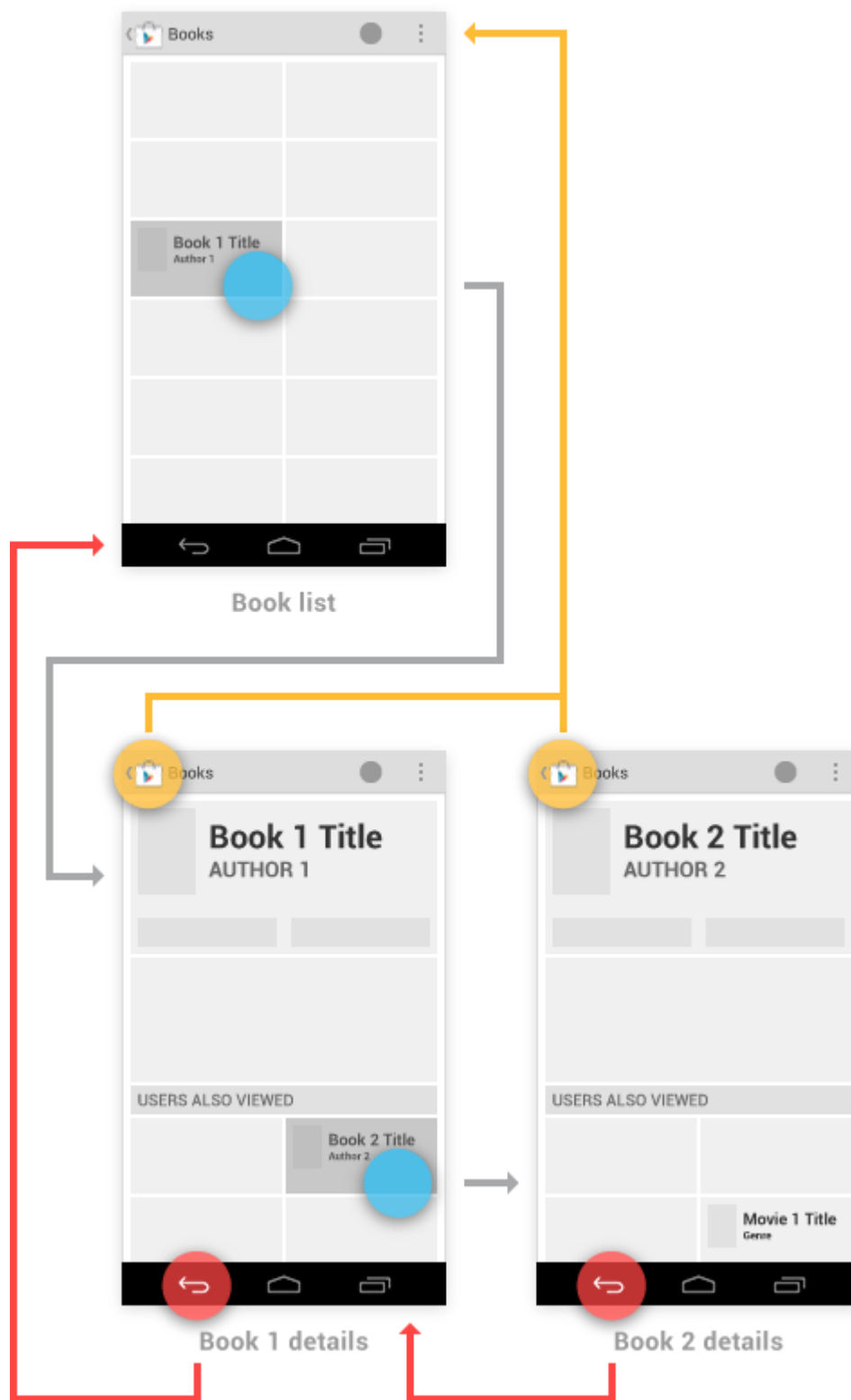


Рисунок 4.14 - Возврат к главному экрану

В случае, если у приложения несколько главных экранов (вкладок), то необходимо организовать динамическую клавишу, находящуюся сверху, слева (Рисунок 4.15).

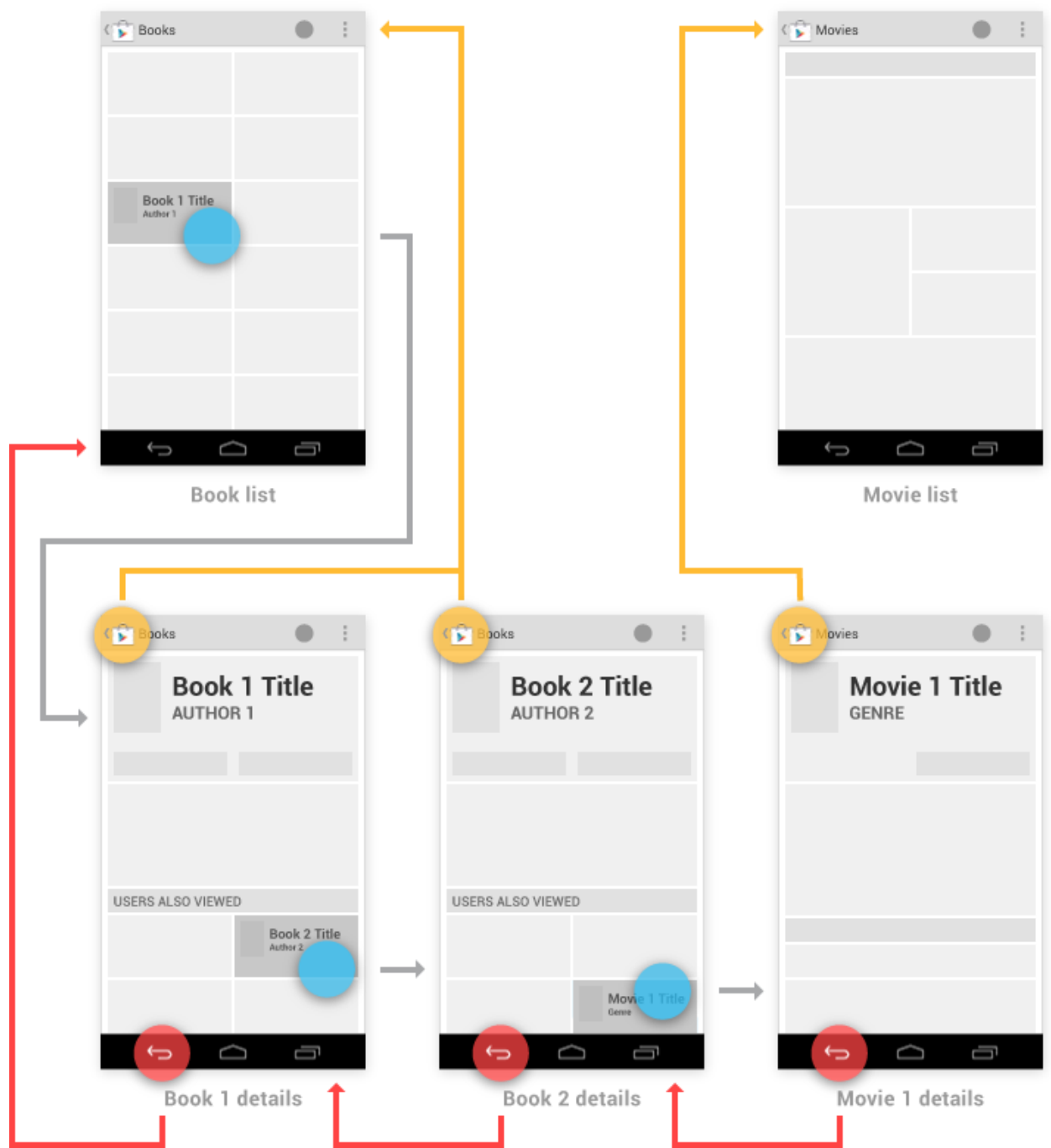


Рисунок 4.15 - Возврат к разным главным экранам

Ниже, на рисунке 4.16 приведена схема действия клавиши «назад» при переходе в приложения с виджета на экране.

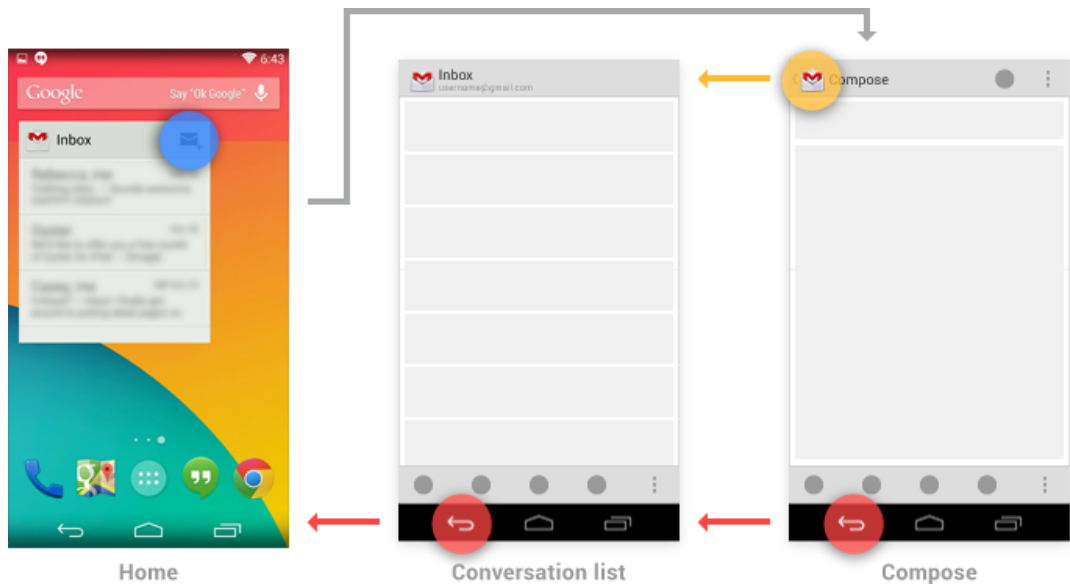


Рисунок 4.16 - Возврат после активизации виджета

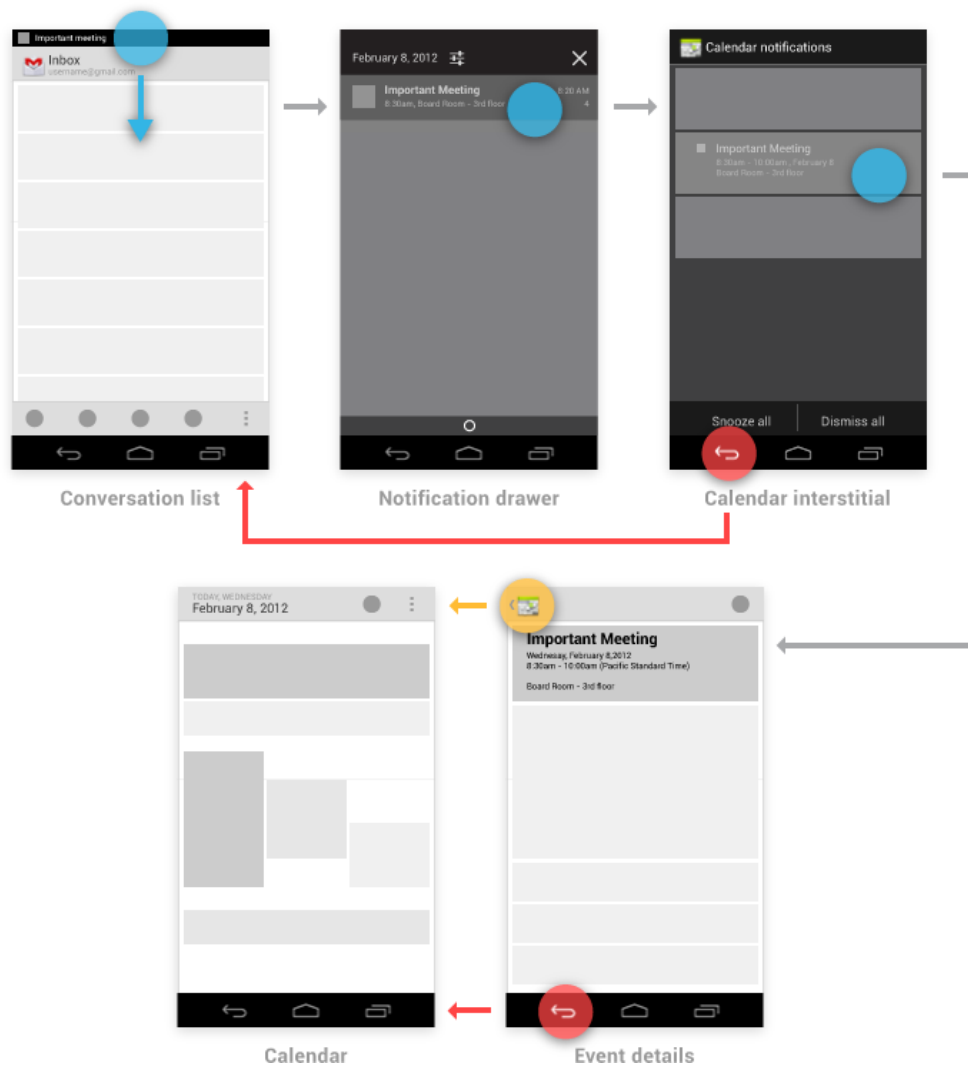


Рисунок 4.17 - Работа клавиши «назад» со шторкой уведомлений



Рисунок 4.18 - Работа клавиши «назад» с всплывающими уведомлениями

При переходе из приложения «А» по ссылке в приложение «Б», после завершения действий в приложении «Б», оно автоматически свернется, и пользователь вернется к приложению «А». В случае отмены действия, нажатие клавиши «назад» работает также, как и обычно, за исключением того момента,

когда вы достигните главной вкладки приложение «Б», дальнейшее нажатие клавиши «назад» вызовет закрытие приложения «Б» и возврат к «А».

На рисунках 4.13-4.18 продемонстрированы основные правила навигации для приложений Android.

4.3 Навигация

Шаблоны первичной навигации. Хорошая навигация, как и грамотный подход к проектированию, является невидимой вещью. Приложения с хорошей навигацией интуитивно понятны и позволяют с легкостью справиться с любой задачей, начиная с просмотра списка друзей и заканчивая обращением за кредитом на приобретение автомобиля. Вариантов навигации по содержимому в рамках приложения может быть много (Рисунок 4.19).

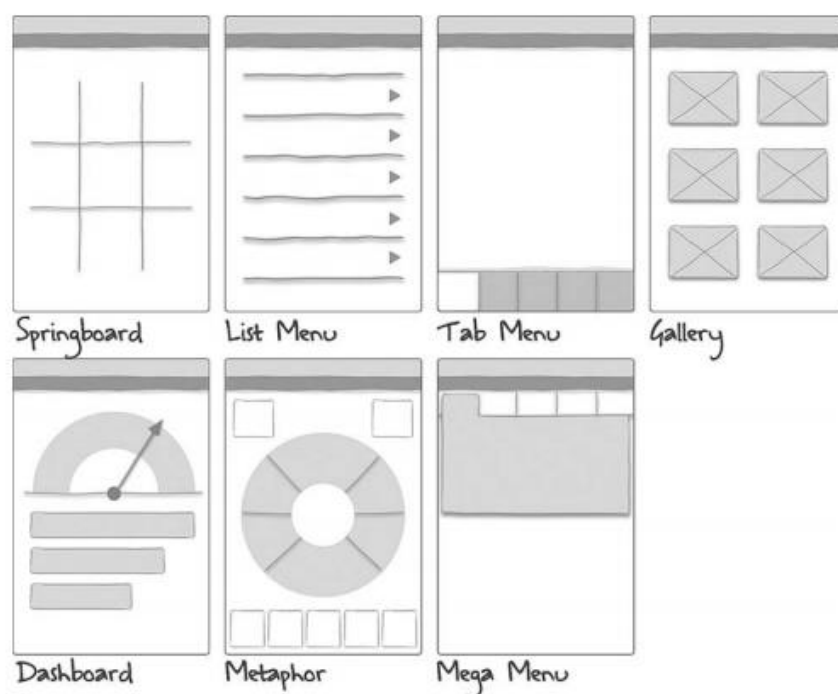


Рисунок 4.19 - Шаблон первичной навигации

Поскольку в каждой операционной системе имеются собственные принципы расположения вкладок, я выбрала шаблон Blank Activity (Пустое окно). В результате пустого окна можно разрабатывать что и как угодно, позволяющее самому предполагать дизайн приложения. На рисунке 4.20 изображено меню приложения.

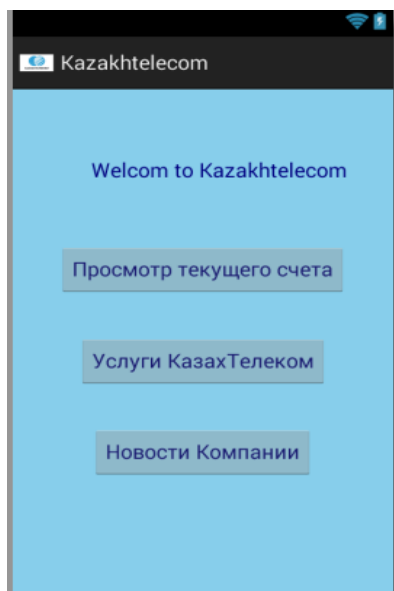


Рисунок 4.20 - Окно главного меню

4.3.1 Окно оплаты услуг компании

После нажатия кнопки «Просмотр текущего счета», осуществляется переход в окно «Номер клиента», где осуществляется введение номера клиента. Представлено на рисунке 4.21.

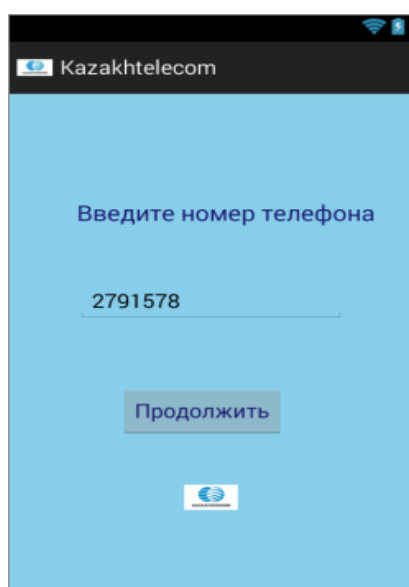


Рисунок 4.21 - Окно ввода номера

После происходит подключение к базе и вывод информации о счете номера, показанного на рисунке 4.22.



Рисунок 4.22 - Окно о счете клиента

В результате просмотра счета, клиент решает оплачивать счет или нет. Если клиент хочет продолжить дальнейшие действия, он нажимает кнопку «Оплатить». Появляется следующее окно, представленное на рисунке 4.23.

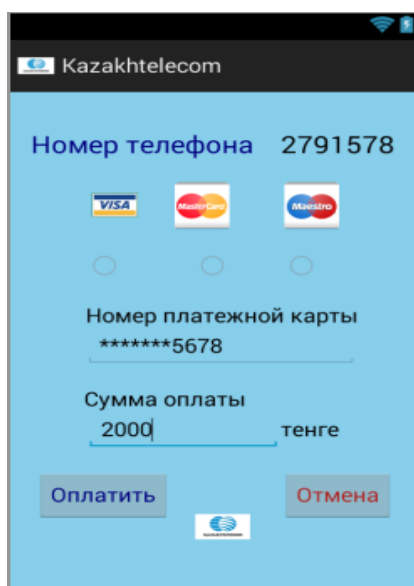


Рисунок 4.23 - Окно платежа

В данном окне выбираете вид платежной карты, вводите номер карты и сумму оплаты. После введения данных, нажимается кнопка «Оплатить».

Для осуществления транзакции нужны такие факторы:

- наличие подключение к сети Интернет;

- достаточно сумму денег на счет платежной карты;
- бесперебойной работы сервера.

После удачной транзакции, появляется окно, представленной на рисунке 4.24.

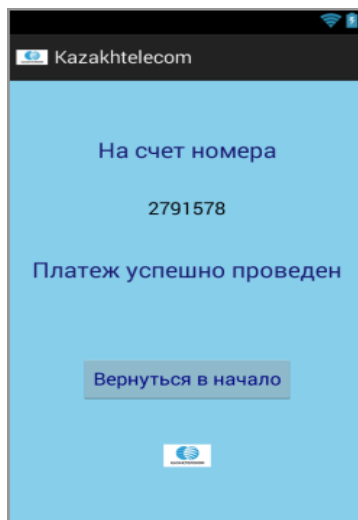


Рисунок 4.24 - Окно завершенной транзакции

4.3.2 Окна услуг компании

Открыв главное окно компании и нажав кнопку «Услуги КазахТелеком», осуществляется переход в окно «Услуги компании». В данном окне перечисляются услуги компании, и рядом с ними находятся кнопки «Подключить» и «Отключить». Если рядом находится кнопка «Подключить», это означает, что вы не подключены к ней. Тоже самое про кнопку «Отключить», означающую, что вы к данной услуге подключены. Представлено на рисунке 4.25.



Рисунок 4.25 - Окно услуг компании

Нажав одну из этих кнопку, происходит отправка заявки на сервер. После удачной отправки заявки появляется окно, показанное на рисунке 4.26.

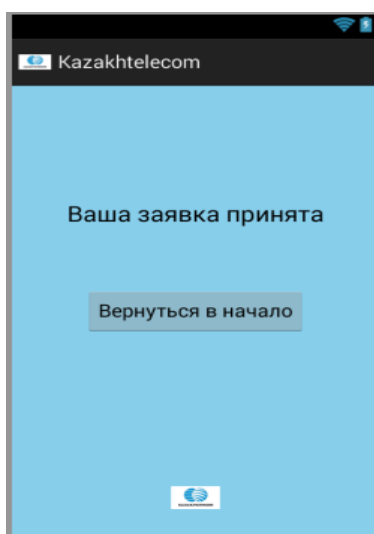


Рисунок - 4.26 - Окно об отправленной заявке

5 Безопасность жизнедеятельности

5.1 Анализ условий труда в отделе «Информационных разработок»

В данном дипломном проекте разработано приложение на узле Отдела информационных технологий в Дирекции Информационных Систем филиала «АО «Казакхтелеком»», включающий набор журналов регистрации записей по оплате услуг по номеру телефона и отчетов контроля исполнения.

При разработке программного обеспечения большую роль играет инструкция по безопасности и охране труда при работе на персональных компьютерах. У каждой национальной компании имеется данная инструкция.

Персональные компьютеры должны быть снабжены элементами защитного заземления (зануления), присоединяемыми к общей сети заземления (зануления) через штепсельные разъемы с заземляющими контактами.

Рабочие места должны соответствовать требованиям СанПиН №767 от 28 сентября 2010 года «Санитарно-эпидемиологические требования к эксплуатации компьютеров, видеотерминалов и условия работы с ними».

Напряженность электромагнитного поля по электрической и магнитной составляющей (на расстоянии 50 см от монитора) должна составлять не выше ПДУ: при 5-2000Гц - Э/п 25В/м, М/п250н/Тл. ПДУ при 2-400кГц -Э/п 2,5В/м, М/п25н/Тл.

Проверить надежность заземления, электросоединений и исправность ПК, принтера. О всех замеченных неполадках сообщить начальнику.

Не следует располагать экран к работнику ближе чем на 50 см.

Сидеть за ПК следует прямо, свободно и не напрягаясь.

В процессе работы на ПК необходимо соблюдать режим труда и отдыха в зависимости от вида работ (набор, правка, верстка, программирование), сложности набора текста (на русском, европейских, языках, формулы, таблицы и т.д) в соответствии с СанПиН №3073 от 18 сентября 2006 года «Санитарно-эпидемиологические требования к эксплуатации компьютеров, видеотерминалов и условия работы с ними» [7].

Регламентированные перерывы следует заполнять комплексами упражнений для глаз, упражнениями и массажем для снятия утомления с плечевого пояса и кистей рук.

Продолжительность работы на ПК без перерывов не должна превышать 2 часов.

При неисправности ПК отключить от электросети и вызвать наладчика (инженера - электронщика).

При задымлении и пожаре действовать согласно Инструкции о мерах пожарной безопасности (ИДО-04.02-3) [8].

При травмах сообщить руководству и оказать первую медицинскую помощь пострадавшим (Конспект вводного инструктажа, ИДО-04.02-1).

В процессе работы необходимо делать регламентирующие перерывы и проводить ряд упражнений для глаз, рук, плечевого сустава, туловища и ног.

Характеристики рабочего помещения

Рассматривается рабочее помещения, расположенное в здании, которое не находится в непосредственной близости от железнодорожной магистрали или нагруженной автомагистрали, аэропорта и так далее, поэтому внешних источников шума, влияющих на процесс работы - нет.

План помещения представлен на рисунке 5.1.

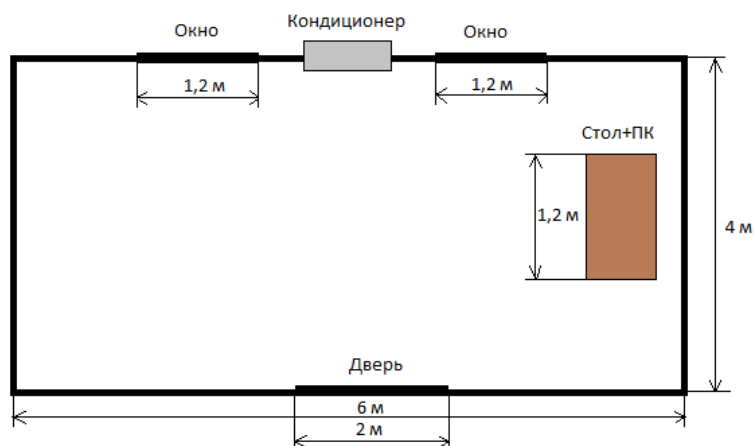


Рисунок 5.1 - План помещения

Помещение имеет следующие параметры:

- находится на первом этаже одноэтажного здания;
- размеры помещения (комнаты): длина 6 м, ширина 4 м, высота 3 м;
- вид светопропускающего материала - стекло листовое, двойное;
- вид переплета - стальные двойные открывающиеся;
- солнцезащитные устройства - убирающиеся регулируемые жалюзи и шторы;
- два окна размером 1,5*1,2;
- внутренняя отделка стен - светлая;
- помещение по зрительным условиям работы относится к категории легких работ (легкая физическая, категория Ia, работа производится сидя и не требует физического напряжения) (ГОСТ 12.2.032-78) [9];
- искусственное освещение - 2 светильника с двумя люминесцентными лампами.

Характеристики используемого в работе оборудования:

- Intel(R) core 2Duo CPU E8400 @ 3.0 GHz 3.0 GHz, 4 Гб RAM / HDD 1000 Gb;
- монитор Samsung SyncMaster 932 BF, диагональ 19”;

- электропитание: переменное напряжение 220-250 В, частотой 50 Гц. Мощность 400 Вт;

- 2 светильника, 4 люминесцентные лампы;

- электропитание: переменное напряжение 220-250 В, частотой 50 Гц, мощность светильника 2x28 Вт.

Электротехническое оборудование является потенциальным источником возникновения пожарной опасности. Оборудование малошумящее - вредность в качестве повышенного шума отсутствует.

5.2 Расчет систем кондиционирования

В таблице 5.1 приведены оптимальные нормы параметров микроклимата с учетом периода года согласно СНиП 2.04.05-91 [10] для легкой физической работы. Оборудование, установленное в рабочем помещении, не является источником выделения тепла (очень незначительное выделение тепла аппаратурой никаким образом не оказывает влияние на микроклимат рабочего помещения). Климатические условия эксплуатации оборудования полностью совпадают с климатическими условиями, нормируемыми для рабочего персонала [10].

Т а б л и ц а 5.1 - Оптимальные нормы температуры, относительной влажности и скорости движения воздуха в обслуживаемой зоне жилых, общественных и административно-бытовых помещений (СНиП 2.04.05-91)

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Легкая - 1а	22-24	40-60	0,1
	Легкая - 1б	21-23	40-60	0,1
Теплый	Легкая - 1а	23-25	40-60	0,1
	Легкая - 1б	22-24	40-60	0,2

Для вентиляции офисного помещения используются каналы естественной вентиляции, прокладываемые при строительстве здания и открытое окно летом. В теплый период года при достижении температуры в офисе выше норм, приведенных в таблице 5.2, для поддержания оптимального микроклимата используется кондиционер. Нормальный микроклимат в офисе обеспечивает хорошее самочувствие сотрудников в любое время года, и соответственно продуктивность работы увеличивается. Таким образом, для поддержания условий микроклимата в помещении, целесообразно оборудовать его системой кондиционирования.

Ниже представлен расчет системы кондиционирования в рабочем помещении. Кондиционирование обеспечит соответствие климата в рабочем помещении нормативам.

Количество приточного воздуха $L_{пр}$, $\frac{м^3}{ч}$ определяем по формуле [10]

$$L_{пр} = \frac{Q_{изб}}{c \cdot \rho_{пр} \cdot (t_{выг} - t_{пр})} \quad (5.1)$$

где $Q_{изб}$ - избыточное выделение явной теплоты, $\frac{кДж}{ч}$;

c - удельная теплоемкость воздуха при постоянном давлении, равная $c=1 \frac{кДж}{кг \cdot ^\circ C}$;

$\rho_{пр}$ - плотность поступающего в помещение воздуха, равная $1,2 \frac{кг}{м^3}$;

$t_{выг}$ - температура удаляемого из помещения воздуха за пределы рабочей или обслуживаемой зоны, $^\circ C$;

$t_{пр}$ - температура приточного воздуха, $^\circ C$.

Температура удаляемого из помещения воздуха $t_{выг}$, $^\circ C$, определяется по формуле [11]

$$t_{выг} = t_{рз} + \Delta t \cdot (h_{вп} - z) \quad (5.2)$$

где $t_{рз}$ - температура в рабочей зоне, которая не должна превышать допустимую по нормам ($t_{рз} \leq t_{доп}$), $^\circ C$;

$h_{вп}$ - расстояние от пола до центра вытяжных проемов (кондиционера), м;

z - высота рабочей зоны, м.

Поскольку расчет производится для теплого периода года, то примем $t_{рз} = 22^\circ C$.

Внутренняя часть кондиционера расположена на высоте $h_{вп} = 3 м$

$$t_{выг} = 22 + 1,2 \cdot (3 - 3)^\circ C = 22$$

Температура приточного воздуха $t_{пр}$ при наличии избытка явной теплоты должна быть на $5 - 7^\circ C$ ниже температуры воздуха в рабочей зоне

$$t_{пр} = 22 - 7 = 15^\circ C$$

Величину избыточного выделения явной теплоты $Q_{\text{изб}}$ находят на основании баланса теплоты в помещении по формуле [11]

$$Q_{\text{изб}} = \sum Q - \sum Q_{\text{yx}} \quad (5.3)$$

где $\sum Q$ - суммарное количество поступающей в помещение явной теплоты;
 $\sum Q_{\text{yx}}$ - суммарное количество уходящей из помещения теплоты (за счет теплопотерь ограждениями, нагрева поступающего в помещение воздуха и т. п.).

Основными источниками избыточного тепла являются светильники, люди и др. Кроме того, необходимо учитывать теплопоступления от солнечной радиации. В данном помещении тепловыделением электронного оборудования можно пренебречь. Поэтому учитываем тепловыделения от искусственного освещения, от людей, количество тепла, поступающего в помещение через окна от солнечной радиации.

Тепловыделения от искусственного освещения Q_2 , рассчитывают, предполагая, что практически вся затрачиваемая энергия, в конечном счете, преобразуется в тепло, по формуле [11]

$$Q_2 = 1000 \cdot N \quad (5.4)$$

где N - расходуемая мощность светильника, кВт.

$$Q_2 = 1000 \cdot 0,14 \cdot 3 = 420 \text{ кВт}$$

Тепловыделения от людей Q_3 определяют по формуле [11]

$$Q_3 = n \cdot q_{\text{ч}} \quad (5.5)$$

где n - число работающих;

$q_{\text{ч}}$ - количество тепла, выделяемое одним человеком, представлено в таблице 5.2.

Т а б л и ц а 5.2 - Количество тепла, выделяемое одним человеком в зависимости от категории работ и температуры окружающей среды [10]

Категория работ	Тепло, Вт			
	Полное		Явное	
	при 100° С	При 350° С	при 100° С	При 350° С
Легкая	180° С	145° С	150° С	5° С

$$Q_3 = 9 \cdot 145 = 1305 \text{ Вт}$$

Количество тепла, поступающего в помещение от солнечной радиации $Q_{\text{ост.рад}}$, определяют по формуле [11]

$$Q_{\text{ост.рад}} = F_{\text{ост}} \cdot q_{\text{ост}} \cdot A_{\text{ост}} \quad (5.6)$$

для покрытий

$$Q_{\text{п.рад}} = F_{\text{п}} \cdot q_{\text{п}} \cdot k_{\text{п}} \quad (5.7)$$

где $F_{\text{ост}}$ и $F_{\text{п}}$ - площадь поверхности и покрытия, м^2 ;

$q_{\text{ост}}$ и $q_{\text{п}}$ - теплопоступления через 1 м^2 поверхности остекления и поверхности покрытия, при коэффициенте теплопередачи, равном $1 \frac{\text{Вт}}{\text{м}^2 \cdot ^\circ\text{С}}$;

$A_{\text{ост}}$ - коэффициент остекления;

$k_{\text{п}}$ - коэффициент теплопередачи покрытия, $1 \frac{\text{Вт}}{\text{м}^2 \cdot ^\circ\text{С}}$.

Значение $q_{\text{ост}}$ в зависимости от географической ориентации поверхности и характеристики окон или фонарей принимается в пределах 70 - 210, а коэффициента $A_{\text{ост}}$ в зависимости от вида остекления и его солнцезащитных свойств - в пределах 0.25 - 1.25, средние значения теплопоступления от солнечной радиации через покрытие в зависимости от географической широты и вида покрытия принимают в пределах 6 - 24.

$$F_{\text{ост}} = 2,5 \cdot 5 \cdot 1 = 10 \text{ м}^2$$

Окно рабочего помещения направлено на север, поэтому примем значение $q_{\text{ост}}$ равным $140 \frac{\text{Вт}}{\text{м}^2 \cdot ^\circ\text{С}}$. Примем $A_{\text{ост}} = 0,35$

$$Q_{\text{ост.рад}} = 10 \cdot 140 \cdot 0,35 = 490 \text{ Вт}$$

Среднее значение теплопоступления для покрытия с учетом географической широты примем равным $Q_{\text{п.рад}} = 18 \text{ Вт}$.

Потери тепла из помещения $Q_{\text{ух}}$, кВт через стены двери, окна оценивают ориентировочно по формуле [11]

$$Q_{yx} = \frac{\lambda \cdot S \cdot (t_{\text{выт}} - t_{\text{пр}})}{\delta} \quad (5.8)$$

где λ - теплопроводность стен, $\frac{\text{Вт}}{\text{м} \cdot ^\circ\text{C}}$;

S - площадь, м^2 ;

δ - толщина стен, м.

Стены рабочего помещения изготовлены из кирпичей М350, теплопроводность которого равна $0,7 \frac{\text{Вт}}{\text{м} \cdot ^\circ\text{C}}$. Толщина стен $\delta = 0,3$ м.

$$Q_{yx} = \frac{0,7 \cdot 50 \cdot (22 - 15)}{0,3} = 816 \text{ Вт}$$

Вычислим суммарное количество поступающей в помещение явной теплоты. [11]

$$\sum Q = Q_2 + Q_3 + Q_{\text{ост.рад}} + Q_{\text{н.р.р}} \quad (5.9)$$

$$\sum Q = 420 + 1305 + 490 + 18 = 2233 \text{ кВт}$$

Так как расчет производится для летнего периода величина избыточного выделения явной теплоты равна:

$$Q_{\text{изб}} = 2233 \text{ кВт}$$

Вычислим количество приточного воздуха:

$$L_{\text{пр}} = \frac{2233}{1 \cdot 1,2 \cdot (22 - 15)} = 670 \frac{\text{м}^3}{\text{ч}}$$

По данным расчетам оптимальным вариантом является кондиционер КЭВ-11П101, параметры показаны в таблице 5.3 и на рисунке 5.2.

Т а б л и ц а 5.3 - Параметры кондиционера КЭВ-11П101 [11]

Модель завесы	КЭВ-11П101
Параметры питающей сети, В/Гц	380/50
Максимальный расход воздуха, м ³ /час	1000
Подогрев воздуха при максимальной мощности, °С	
- минимальный расход	33
- максимальный расход	26
Габаритные размеры, мм	190*200*1350
Вес, кг	17
Максимальный ток, А	17.3
Ток двигателя, А	0.44
Звуковое давление на расстояние 5 м, дБ (А)	54



Рисунок - 5.2 - Кондиционер КЭВ-11П101

Установленного кондиционера достаточно для обеспечения расхода воздуха $L=670$ м³/ч, что является достаточным для обеспечения комфортного микроклимата.

5.3 Анализ пожарной безопасности

Согласно СНиП 2.04.09-84 [10] и РНТП-01-94 [12] здание по степени опасности развития пожара, от функционального назначения и пожарной нагрузки горючих материалов, относится к 1-ой группе категории Д.

Причинами возникновения пожара могут быть:

- возгорание элементов аппаратуры;
- возгорание отделочных материалов от неисправных выключателей, розеток;
- несоблюдение режимов эксплуатации оборудования, неправильное действие персонала.

При возникновении пожара может пострадать не только помещение, но и дорогостоящая аппаратура, привести к человеческим жертвам. Поэтому необходимо чтобы были приняты меры по раннему выявлению и ликвидации пожаров. Источниками зажигания могут оказаться электронные схемы ЭВМ, приборы, применяемые для технического обслуживания, устройства электропитания, кондиционеры воздуха, где в результате различных нарушений образуются перегретые элементы, и др.

В соответствии с требованиями правил пожарной безопасности помещение оборудованы углекислотными огнетушителями ОУ-5 с учетом - один огнетушитель на 100 м². Общая площадь помещения составляет 20 м², таким образом устанавливается 1 огнетушитель. В качестве огнетушащего вещества применяется комбинированный углекислотно-хладоновый состав. Расчетная масса комбинированного углекислотно-хладонового состава m_d , для объемного пожаротушения определяется по формуле [13]

$$m_d = k \cdot g_n \cdot V \quad (5.10)$$

где $k = 1,2$ - коэффициент компенсации не учитываемых потерь углекислотно-хладонового состава,

$g_n = 0,04$ - нормативная массовая концентрация углекислотно-хладонового состава,

V - объем помещения определяется по формуле [13]

$$V = A \cdot B \cdot H \quad (5.11)$$

где $A = 10$ м - длина помещения,

$B = 5$ м - ширина помещения,

$H = 3$ м - высота помещения.

Тогда:

$$V = 10 \cdot 5 \cdot 3 = 150 \text{ м}^3$$

Следовательно

$$m_d = 1.2 \cdot 0.04 \cdot 150 = 7.2 \text{ кг}$$

Расчетное число баллонов x определяется из расчета вместимости в 20-литровый баллон 12 кг углекислотно-хладонового состава.

Внутренний диаметр магистрального трубопровода d_i (мм), определяется по формуле [13]

$$d_i = 12 \cdot \sqrt{2} = 17 \text{ мм} \quad (5.12)$$

Эквивалентная длина магистрального трубопровода l_2 определяется по формуле [13]

$$l_2 = k_1 \cdot l \quad (5.13)$$

где $k_1=1,2$ - коэффициент увеличения длины трубопровода для компенсации не учитываемых местных потерь;

$l=3$ м - длина трубопровода по проекту тогда.

$$l_2 = 1.2 \cdot 3 = 3.6 \text{ м}$$

Расход углекислотно-хладонового состава Q , в зависимости от эквивалентной длины и диаметра трубопровода равна 1,4 кг/с.

Расчетное время подачи углекислотно-хладонового состава t , определяется по формуле [13]

$$t = m_d / 60Q \quad (5.14)$$

Тогда

$$T = 7.2/150 \cdot 1.4 = 0.034 \text{ мин}$$

Масса основного запаса углекислотно-хладонового состава m определяется по формуле [13]

$$m = 1,1 \cdot m_d \cdot \left(1 + \frac{k_2}{k_1}\right) \quad (5.15)$$

где $k_2=0,2$ - коэффициент учитывающий остаток углекислотно-хладонового состава в баллонах и трубопроводах. Тогда

$$m = 1,1 \cdot 7,2 \cdot (1+0,2/1,2) = 9,24 \text{ кг}$$

Таким образом, из полученных результатов можно сделать вывод, что для обеспечения нормального функционирования системы автоматического пожаротушения потребуется 1 баллон углекислотно-хладонового состава вместимостью 20 литров, с массой смеси 7,2 кг (Рисунок 5.3). Автоматические установки газового пожаротушения имеют устройства для автоматического пуска в соответствии с ГОСТ 12,4.009-83 [7].

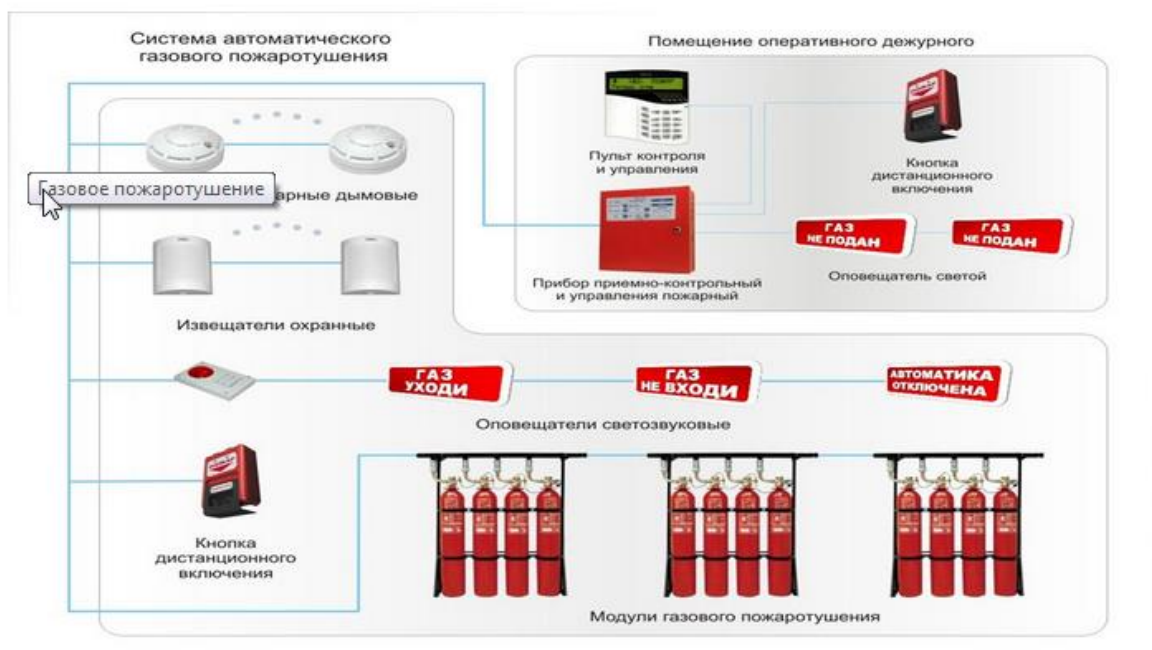


Рисунок 5.3 - Система автоматического газового пожаротушения

5.4 Вывод по разделу безопасности жизнедеятельности

В данном разделе был произведён анализ условий труда в рабочем помещении для девяти сотрудников. Уровень условий труда признан допустимым, и данные, полученные из расчетов полностью удовлетворяют требованиям стандартов безопасности жизнедеятельности.

Так как в помещении есть одно окно, которое в дневное время суток обеспечивает достаточным проветриванием помещения для работы, необходимо было рассчитать кондиционирования в основном для теплого времени года. Поэтому для создания нормированного кондиционирования понадобится 1 кондиционер для обеспечения расхода воздуха $L=670$ м³/ч, находящегося на высоте 3 м, достаточно обеспечивающего температуру в помещении 16-22 С, и не осушающий воздух.

Электротехническое оборудование в помещения является потенциальным источником возникновения и пожароопасности. Из расчетов получили, что для обеспечения нормального функционирования системы автоматического пожаротушения потребуется 1 баллон углекислотно-хладонового состава вместимостью 20 литров, с массой смеси 7,2 кг.

6 Экономическое обоснование. Стоимость проекта

6.1 Технико-экономическое обоснование эффективности разработки

Приложение оплаты услуг для телефонов - это новшество в современном мире, которое постоянно в движении. Приложение позволяет современному человеку не тратить свое время в ожидании очереди, чтоб оплатить услуг, предоставленные АО «Казахтелеком».

Как в свое время терминалы упростили жизнь людям, так и приложение делает ее еще проще.

6.2 Бизнес план

Приложение в качестве бизнес-проекта (некоммерческие не берем) может быть одного из двух видов:

- самостоятельный бизнес. Владельцы приложения извлекают доход из самого приложения с помощью размещения рекламы;
- вспомогательный проект для основного бизнеса. Приложение служит дополнительной площадкой для общения с потенциальными клиентами и помогает "делать деньги".

В моем случае это самостоятельный проект прибыль приходит с самого сайта за счет размещения рекламы и дополнительных платных услуг и др.

Целью данного проекта является создание аналога обычного терминала для смартфонов, потому что в настоящее время каждый второй человек имеет смартфон. Приложение является дополнительной услугой для осуществления оплаты. Для оплаты нужно всего лишь подключение к интернету через Wi-Fi или 3G, и можете оплачивать из любой точки нашей страны.

Интернет-исследования, проведенные мной показывают, что приложений с подобной тематикой почти нет и серьезной конкуренции создаваемому приложению составить не могут. Поэтому единственной трудностью является раскрутка собственного приложения и наработка постоянных посетителей.

А так как круг потенциальных клиентов очень велик, приложение нужно любому уважающему себя человеку, которому дорого его время.

План разработки программного продукта:

- для того, чтобы определить затраты на разработку и внедрение, необходимо составить план проведения работы и смету затрат;
- жизненным циклом программы считается весь цикл от принятия решения о проведении разработок до его реализации.

Разработка и внедрение программного продукта состоит из 4 основных этапов.

Моделирование.

Составление модели программного обеспечения. Данный этап включает в себя разработку принципов работы программного обеспечения: определение алгоритмов, основных компонентов и их назначения, проработку интерфейсов взаимодействия между компонентами системы, проработку внешнего, удобного для пользователя интерфейса, определение используемых технологий, оценку системных требований для программного обеспечения. Участники этапа: программист. Длительность этапа: 5 рабочих дней.

Программирование.

На данном этапе будет осуществляться описание разработанной на первом этапе модели с помощью алгоритмических языков программирования. Участники этапа: программист. Длительность этапа: 15 рабочих дней.

Тестирование ПО.

На данном этапе будет производиться тестирование разработанного ПО, выявление ошибок реализации, будут рассмотрены принципиальные схемы внедрения ПО. Также, по ходу данного этапа будет производиться исправление найденных ошибок. Участники этапа: программист. Длительность этапа: 3 рабочих дня.

Внедрение ПО.

В рамках данной работы будет производиться внедрение разработанного программного обеспечения в сеть Интернет. Участники этапа: программист. Длительность этапа: 1 рабочий день.

Сводные данные по плану проведения работ по разработке программного продукта представлены в таблице 6.1.

Т а б л и ц а 6.1 - План проведения работ по разработке ПО

Наименование этапов и содержание работ	Исполнитель	Количество исполнителей	Длительность цикла
Моделирование	программист	1	20.03.14 - 24.03.14 5 рабочих дней 40 нормо-часов
Программирование	программист	1	27.03.14 - 15.04.14 15 рабочих дней 120 нормо-часов
Тестирование	программист	1	18.04.14 - 20.04.14 3 рабочих дня 24 нормо-часа
Внедрение	программист	1	21.04.14 1 рабочий день 8 нормо-часов
Итого		1	24 рабочих дней 192 нормо-часа

6.3 Расчет стоимости разработки ПО

Себестоимость - это все затраты на производство и реализацию продукции. [14].

Себестоимость ПО складывается из следующих статей затрат:

- оплата труда;
- социальный налог;
- амортизационные отчисления;
- расходы на электроэнергию;
- накладные расходы.

Таким образом, себестоимость разработки проекта определяется по следующей формуле

$$C = \text{ФОТ} + C_n + A + C_{эл} + N \quad (6.1)$$

где ФОТ - фонд оплаты труда;

C_n - социальный налог;

A - амортизационные отчисления;

$C_{эл}$ - расходы на электроэнергию;

N - накладные расходы.

6.3.1 Расходы на заработную плату

Для расчета затрат на заработную плату необходимы следующие данные:

- численность задействованного персонала;
- среднемесячная заработная плата каждого работника;
- длительность разработки проекта и каждого вида выполняемых работ;
- трудоемкость.

В процессе разработки данного программного обеспечения участвует 1 человек - программист.

Месячная заработная плата программиста составляет 140000 тенге.

Зарботную плату за один час рассчитаем по формуле [14]

$$D = \frac{ЗПм}{D_p \cdot Ч_p} \quad (6.2)$$

где $ЗПм$ - ежемесячный размер заработной платы;

D_p - количество рабочих дней в месяце (22 рабочих дня);

$Ч_p$ - количество часов рабочего дня (при 8 часовом рабочем дне).

Зарботная плата программиста за один час составляет

$$D = \frac{140000}{22 \cdot 8} = 795,45 \text{ тенге/час}$$

На основе данных из бизнес-плана по длительности каждого этапа разработки программного продукта, произведем расчет количества нормо-часов для программиста.

Количество нормо-часов рассчитывается по формуле

$$T = t_n \cdot z \quad (6.3)$$

где t_n - количество часов в рабочем дне;

z - рабочие дни.

Для программиста

$$T = 24 \cdot 8 = 192 \text{ нормо-часа}$$

Сводные данные расчета заработной платы программиста представлены в таблице 6.2.

Т а б л и ц а 6.2 - Расчет заработной платы программиста.

Наименование содержания работ	Исполнитель	Трудоёмкость, норма-час	Зар. плата за час работы (тг/час)	Сумма заработной платы (тг.)
Моделирование	Программист	40	795	31 800
Программирование	Программист	120	795	95 400
Тестирование	Программист	24	795	19 080
Внедрение	Программист	8	795	6 360
ИТОГ (ФОТ)		192		152 640

6.3.2 Социальный налог

Согласно разделу 12 «Социальный налог» статья 358 «Ставки налога» НК РК юридические лица - резиденты Республики Казахстан, а также нерезиденты, осуществляющие деятельность в Республике Казахстан через постоянное учреждение, филиалы и представительства, уплачивают социальный налог 11% [15].

В расчетах используется максимальная процентная ставка 11%.

Социальный налог (Сн) рассчитывается по формуле

$$C_n = (\Phi OT - Пф) \cdot 11\% \quad (6.4)$$

где ФОТ - фонд оплаты труда;
Пф - пенсионные отчисления.

Согласно Закону РК «О пенсионном обеспечении» пенсионные отчисления будут рассчитываться согласно формуле

$$Пф = \Phi OT \cdot 10\% \quad (6.5)$$

Таким образом, пенсионные и социальные отчисления составляют

$$Пф = 152640 \cdot 10\% = 15264 \text{ тенге,}$$

$$C_n = (152640 - 15264) \cdot 11\% = 15111 \text{ тенге}$$

6.3.3 Расчет затрат на амортизацию

Амортизационные отчисления на основные средства за 24 дня разработки программного продукта вычисляются по формуле

$$A = \frac{H_A \cdot K_{обор}}{365 \cdot 100\%} \cdot 24 \quad (6.6)$$

где H_A - норма амортизации 12%;
 $K_{обор}$ - стоимость оборудования.

Стоимость оборудования, необходимого на разработку программного продукта представлены в таблице 6.3.

Т а б л и ц а 6.3 - Оборудование, необходимого на разработку ПП.

Наименование материала	Единицы измерения	Количество	Цена единицу за в тенге	Сумма в тенге
Компьютер Core2Duo 3000/S775/4Gb/1000 Gb/512MbPCI/SB	штук	1	56 000	56 000
Монитор 19"	штук	1	20 000	20 000
Клавиатура PS/2	штук	1	1 600	1 600

Окончание таблицы 6.3

Наименование материала	Единицы измерения	Количество	Цена за единицу тенге	Сумма в тенге
Мышка PS/2	штук	1	1 200	1 200
			Итого:	78 800

Общая стоимость расходов на оборудование составляет 78 800 тенге.

Амортизационные отчисления за весь период разработки ПО составляют

$$A = \frac{12 \cdot 78800 \cdot 24}{100\% \cdot 365} = 621 \text{ тенге}$$

6.3.5 Затраты на электроэнергию

Важной статьёй затрат являются затраты на потребляемую электроэнергию. Затраты на электроэнергию рассчитывается по следующей формуле [16]

$$C_{эл} = W \cdot T \cdot S \quad (6.7)$$

где W- потребляемая мощность, кВт;

T - количество часов работы;

S - стоимость киловатт-часа электроэнергии.

Виды используемого оборудования, а так же потребляемая ими мощность представлены в таблице 6.4. Исходя из этих данных, рассчитывается стоимость расхода электроэнергии.

Т а б л и ц а 6.4 - Потребляемая мощность оборудования

Наименование	Потребляемая мощность, Вт/час
Персональный компьютер - 1 шт.	500
Итого:	500

Согласно установленному тарифу по энергопотреблению стоимость 1 кВт составляет 24,32 тенге.

С учетом длительности восьмичасового рабочего дня и длительности разработки, внедрения проекта, количество часов работы составит

$$T = 8 \cdot 24 = 192 \text{ часов}$$

В соответствии с формулой 6.7 расходы на электроэнергию составят

$$C_{эл} = 0,5 \cdot 192 \cdot 24,32 = 2334 \text{ тенге}$$

6.3.6 Расчет затрат на накладные расходы

Накладные расходы на разработку проекта составляют от 25% от общей суммы затрат и рассчитываются по формуле [17]

$$H = (\text{ФОТ} + C_n + A + C_{эл}) \cdot 0,25 \quad (6.8)$$

Тогда, согласно формуле (6.8), накладные расходы будут равны

$$H = (152640 + 15111 + 621 + 2334) \cdot 0,25 = 42676 \text{ тенге}$$

6.3.7 Себестоимость проекта

В соответствии с произведенными расчетами по статьям затрат себестоимость проекта, согласно формуле 6.1, будет равна

$$C = 152640 + 15111 + 621 + 2334 + 42676 = 213382 \text{ тенге}$$

Сводные результаты расчета себестоимости разработки программного продукта и структура представлены в таблице 6.5 и на рисунке 6.1.

Т а б л и ц а 6.5 - Результирующая таблица себестоимости приложения.

№	Статья расходов	Сумма, тенге	Процентное отношение к общей сумме, %
1	Фонд оплаты труда (ФОТ)	152640	71,5
2	Социальный налог (Сн)	15111	7,1
3	Расходы на амортизацию (А)	621	0,3
4	Расходы на электроэнергию (Сэл)	2334	1,1
5	Накладные расходы (Н)	42676	20
	Итого	213382	100

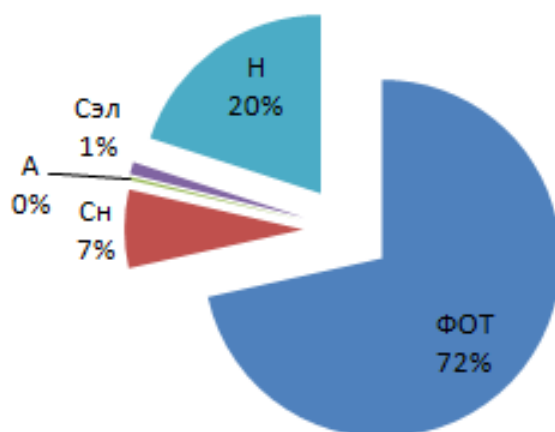


Рисунок 6.1 - Структура себестоимость приложения

6.4 Цена программного продукта

Цена реализации приложения складывается из себестоимости и чистого дохода, и вычисляется по формуле [17]

$$Ц = C + П \quad (6.9)$$

где С - себестоимость продукта;
П - чистый доход.

Первоначальная цена рассчитывается через рентабельность проекта. Учитывая, что желаемый уровень рентабельности для отрасли телекоммуникации составляет 40%, применим следующую формулу

$$Ц_{п} = C \cdot \left(1 + \frac{P}{100}\right) \quad (6.10)$$

где P - рентабельность (40%).

Согласно формуле 6.9 первоначальная цена равна

$$Ц_{п} = 213382 \cdot (1 + 0,4) = 298734,8 \text{ тенге}$$

Цена реализации готовой продукции рассчитывается по формуле

$$Ц_{р} = Ц_{п} + НДС \quad (6.11)$$

Поскольку на сегодняшний день размер НДС в РК составляет 12%, следовательно

$$НДС = Ц_{п} \cdot 12\% \quad (6.12)$$

$$НДС = 298734,8 \cdot 12\% = 35848,18 \text{ тенге}$$

Тогда согласно формуле 6.12 цена реализации составит

$$Ц_p = 298734,8 + 35848,18 = 334583 \text{ тенге}$$

6.5 Вывод по разделу экономическое обоснование

Конечная цена разрабатываемого продукта является более чем адекватной за продукт, который обладает такими богатыми функциональными возможностями.

С использованием обозначенных формул и уравнений, а также информации из трудового кодекса республики Казахстан, были вычислены все неизбежные и предполагаемые затраты, которых стоит ожидать при приведении в жизнь задуманного проекта. В результате расчета затраты на создание данного программного продукта составили 213382 тенге. Данная цифра представляется вполне умеренной, но так как имеющиеся на рынке подобные продукты довольно специфичны и количество их очень мало, оценить реальные сроки окупаемости и, как следствие, успешность проекта представляется довольно трудной и неоднозначной задачей. Многое в этом деле будет зависеть от метода распространения данного программного средства (здесь можно рассматривать такие методы как продажа подписки и аналогичные) и его конкурентоспособности в данный период времени.

Заключение

В ходе выполнения данного дипломного проекта разработывалось мобильное приложение для оплаты услуг АО «Казахтелеком», главного провайдера телекоммуникационных услуг РК. Приложение позволяет пользователям смартфонов с операционной системой Android в любое время (главное чтоб было подключение к сети Internet):

- просматривать свой текущий счет по номеру телефона;
- оплачивать свой счет через платежную карту;
- просматривать услуги, которые предоставляет компания, после чего можно подключиться/отключиться к/от услуг;
- просмотр новостей компании.

Для достижения поставленной цели и решения предложенной задачи была проделана следующая работа:

- досконально изучена методика создания приложений, которая в дальнейшем использовалась при разработке данного приложения;
- детально изучены наиболее популярные и оптимальные инструментальные средства разработки приложений: Java, Eclipse, Android, MySql, XML;
- подобраны инструменты для создания приложения.

Практическую ценность своего проекта вижу в том, что:

- получила большой опыт в создании приложений для мобильных устройств, при этом были глубоко освоены инструментальные средства для разработки мобильных приложений;
- проект в дальнейшем будет использоваться пользователями Android, что принесет компании большую популярность.

Список использованной литературы

- 1 Сайт <http://ru.wikipedia.org/wiki>
 - 2 Сайт <http://electroniccity.ru/news.php/>
 - 3 Сайт <http://android.ru/>
 - 4 Стив Суэринг, Тим Конверс, Джойс Парк. PHP и MySQL. Библия программиста. - М.: Диалектика, 2010. - 912 с.
 - 5 Мамаев Е. MySQL Server. Проектирование и реализация баз данных. Сертификационный экзамен. - СПб.: BHV, 2004. - 416с.
 - 6 Сайт <http://startandroid.ru/ru/>
 - 7 Сайт <http://www.damu.kz/428>
 - 8 Правила пожарной безопасности при производственно-монтажных работ: ППБ-05-86: Утв. 26 февраля 1986 г./М-во внутренних дел СССР. Гл. упр. пожарной охраны. - М.: Стройиздат, 1988.
 - 9 Сайт http://www.ohranatruda.ru/ot_biblio/normativ/data_normativ/
 - 10 Сайт <https://docs.google.com/document/>
 - 11 Хакимжанов Т. Е.. Безопасность жизнедеятельности. Расчет аспирационных систем. Дипломное проектирование: Методические указания для студентов всех форм обучения специальности. - Алматы: АИЭС, 2002. - 30 с.
 - 12 Сайт http://online.zakon.kz/Document/?doc_id=1042810
 - 13 Белов С.В., Ильницкая А.В., Козьяков А.Ф. и др. Безопасность жизнедеятельности: Учебник для вузов. - М.: Высшая школа, 2005. - 448 с.
 - 14 Бериккулы А.Б. Дипломное проектирование: Методические указания к выполнению дипломных проектов для студентов специальности 3501 - Электронные системы и технологии и 3703 - Организация и технология защиты информации всех форм обучения. -Алматы: АИЭС, 2007. -50 с.
 - 15 Сайт <http://online.zakon.kz/ContentPages/>
 - 16 Базылов К.Б., Алибаева С.А., Бабич А.А. Методические указания по выполнению экономического раздела выпускной работы бакалавров для студентов всех форм обучения специальности 05071 - Радиотехника, электроника и телекоммуникации -Алматы: АИЭС, - 2008. -19 с.
- З.Д. Еркешева, Г.Ш. Боканова. Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 - Вычислительная техника и программное обеспечение. -Алматы: АУЭС, 2013 -40 с.

Приложение А

(обязательное)

Листинг программы

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:s="library://ns.adobe.com/flex/spark"
  xmlns:mx="library://ns.adobe.com/flex/mx" minWidth="955" minHeight="600"
  creationComplete="send_data()">
<fx:Declarations>
  <s:HTTPService id="podrazdRequest" url=" http://kulinar.kz/search /baz.php"
    useProxy="false" method="POST">
    <mx:request xmlns="">
      <ingrid>{ ingrid.text}</blud>
      <categ>{ categ.text}</categ>
    </mx:request>
  </s:HTTPService>
</fx:Declarations>
<fx:Script>
  <![CDATA[
private function send_data():void {
podrazdRequest.send();
}
]]>
</fx:Script>
<mx:Form x="20" y="10" width="300">
  <mx:FormItem>
    <s:Label text=" blud " />
    <s:TextInput id=" blud "/>
  </mx:FormItem>
  <mx:FormItem>
    <s:Label text="" />
    <s:TextInput id="categ"/>
  </mx:FormItem>
  <s:Button label="Submit" click="send_data()"/>
</mx:Form>
<mx:DataGrid id="dgPodrazdRequest" x="20" y="160"
  dataProvider="{podrazdRequest.lastResult.vid.podrazd}">
```

Продолжение приложения А

```

    <mx:columns>
        <mx:DataGridColumn headerText=" " dataField="categ"/>
        <mx:DataGridColumn headerText=" " dataField=" blud "/>
    </mx:columns>
</mx:DataGrid>
<s:TextInput x="20" y="340" id="selectedcateg"
text="{ dgPodrazdRequest.selectedItem.categ }"/>
<s:Button x="60" y="10" width="142" label="Просмотр текущего счета"/>
    <s:Button x="60" y="40" width="142" label="Услуги КазахТелеком"/>
    <s:Button x="60" y="70" width="142" label="Новости Компании"/>
typographicCase="default"><s:VGroup x="244" y="303">
    <s:RadioButton label="Услуги компании" groupName="Разное"/>
<s:RadioButton label="Оплатить счет" groupName="Назад"/>
    <s:RadioButton label="Оплатить счет" groupName="Назад"/>
</s:VGroup></s:DropDownList>
    <s:DropDownList x="414" y="71" color="#000000"
pageScrollingEnabled="false" requireSelection="true" selectedItem="Выбор
карточки"><s:labelDisplay>элементарно</s:labelDisplay></s:DropDownList>
    <mx:Tree x="305" y="134"></mx:Tree>
    <s:List x="353" y="318"></s:List>
    <s:VGroup x="244" y="303">
        <s:RadioButton label="Введение номер карточки"
groupName="Разное"/>
        <s:RadioButton label="Введение суммы" groupName="Разное"/>
        <s:RadioButton label="новогодние рецепты" groupName="Разное"/>
    </s:VGroup>
    <s:ComboBox x="406" y="108"> </s:ComboBox>
    <s:List x="477" y="166" labelField="1"></s:List>
    <s:Label x="325" y="82" text="Подключить"/>
    <s:Label x="495" y="199" text="Отключить"/>
    <s:Button x="484" y="40" width="142" label="Назад"/>
</s:Application>

```

Листинг взаимодействия с базой данных

```

<?php
define( "DATABASE_CATEG", "caneg" );
define( "DATABASE_BLUD", "blud" );
define( "DATABASE_VREM", "vrem" );
define( "DATABASE_NAME", "sample" );

```

Продолжение приложения А

```

$mysql = mysql_connect(DATABASE_CATEG, DATABASE_BLUD,
DATABASE_VREM);
mysql_select_db( DATABASE_BOOK );
function quote_smart($value)
{
    if (get_magic_quotes_gpc()) {
        $value = stripslashes($value);
    }
    if (!is_numeric($value)) {
        $value = "" . mysql_real_escape_string($value) . "";
    }
}
return $value;
}
if( $_POST["categ"] AND $_POST["blud"])
{
$query = sprintf("INSERT INTO vid VALUES (' , %s, %s)",
    quote_smart($_POST['blud']), quote_smart($_POST['categ']));
$result = mysql_query( $Query );
}
$query = "SELECT * from vid";
$result = mysql_query( $Query );
$return = "<vid>";
while ( $Schet = mysql_fetch_object( $Result ) )
{
$return .= "<Schet><Summa>".$Podrazd->categ."</Summa>< Schet >".
    $Podrazd->blud."</ blud ><categ>".
    $Podrazd->categ."</categ></podrazd>";
}
$return .= "</vid>";
mysql_free_result( $Result );
print ($Return)
?>

```

Листинг сайта

```

<!DOCTYPE html>
<!-- saved from url=(0029)http://www.telecom.kz/-->
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">

```