

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

**ДОПУЩЕН К ЗАЩИТЕ**

Заведующий кафедрой

PhD, доцент

\_\_\_\_\_ Т.С. Картбаев

« \_\_\_\_ » \_\_\_\_\_ 2019 г.

**ДИПЛОМНЫЙ ПРОЕКТ**

На тему: Разработка приложения «Медицинская помощь гражданам г.Алматы»  
на платформе Android.

Специальность: 5B070400 – «Вычислительная техника и программное  
обеспечение»

Выполнил: Сабырбек Т.О. Группа: ВТ-15-2  
Научный руководитель: к.т.н., доцент Табултаев С.С.

Консультанты:

по экономической части: к.э.н., профессор \_\_\_\_\_ Ж.Г. Аренбаева  
« 13 » \_\_\_\_\_ 2019 г.

по безопасности

жизнедеятельности: д.т.н., ст. преп. \_\_\_\_\_ Ш.Ш. Бекбасаров  
« 14 » \_\_\_\_\_ 2019 г.

по применению

вычислительной техники: ст. преп. \_\_\_\_\_ М.Н. Майкотов  
« 14 » \_\_\_\_\_ 2019 г.

Нормоконтролер: ст. преп. \_\_\_\_\_ А.А. Айтказина  
« 15 » \_\_\_\_\_ 2019 г.

Рецензент: д.т.н., профессор \_\_\_\_\_ С.Н.Абадуллаевна  
« \_\_\_\_ » \_\_\_\_\_ 2019 г.

Алматы 2019

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

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В070400 – «Вычислительная техника и программное обеспечение»

**ЗАДАНИЕ**

на выполнение дипломного проекта

Студенту Сабырбек Тамерлан Омыргазыулы

Тема проекта: Разработка приложения «Медицинская помощь гражданам г.Алматы» на платформе Android

Утверждена приказом по университету № 33 от «1» марта 2019 г.

Срок сдачи законченного проекта «24» мая 2019 г.

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

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

- аналитическая часть;
- проектная часть;
- экспериментальная часть;
- экономическая часть;
- безопасность жизнедеятельности;
- приложение А. Техническое задание;
- приложение Б. Листинг программы;
- приложение В. Акт внедрения.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 15 таблиц, 20 иллюстрации.

Основная рекомендуемая литература:

1 Коровкин С. Д., Левенец И. А., Ратманова И. Д., Старых В. А., Щавелёв Л. В. Решение проблемы комплексного оперативного анализа информации хранилищ данных // СУБД. - 1997. - № 5-6. - С. 47-51.

2 Распределенная обработка данных: курс лекций / Сост. Найханова Л.В. – Улан-Удэ, Издательство ВСГУТУ, 2001. – 122 с.

3 Гофман В.Э. Хомененко А.Д. Delphi 7 – СПб.: БХВ – Санкт-Петербург, 2005 – 1152с.

4 Гофман В.Э., Хомененко А.Д. Работа с базами данных в Delphi. – СПб.: БХВ – Петербург, 2000. – 656 с.

Консультации по проекту с указанием относящихся к ним разделов проекта

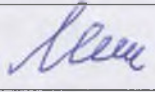
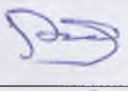
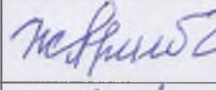

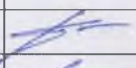


Раздел	Консультант	Сроки	Подпись
Экономическая часть	Аренбаева Ж.Г.	04.05.2019 - 13.05.2019	
Безопасность жизнедеятельности	Бекбасаров Ш.Ш.	12.03.2019 - 13.05.2019	
Программное обеспечение	Майкотов М.Н.	08.04-14.05 2019	
Нормоконтролер	Айтказина А.А.	02.04 - 15.05.19	

ГРАФИК  
подготовки дипломной проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Аналитическая часть	05.11.2018 - 21.12.2018	
Проектная часть	07.01.2019 - 31.01.2019	
Экспериментальная часть	04.02.2019 - 20.04.2019	

Дата выдачи задания « 29 » октября 2018 г.

Заведующий кафедрой \_\_\_\_\_ Т.С. Картбаев

Научный руководитель проекта \_\_\_\_\_ С.С. Табултаев

Задание принял к исполнению студент \_\_\_\_\_ Т.О. Сабырбек

## Аңдатпа

Дипломдық жобаның тақырыбы «Android платформасында "Алматы қаласының медициналық көмек" көсмшасын әзірлеу»

Дипломдық жобаның мақсаты– офис қызметкерлерінің жұмыс процесін талдау мен кәсіпқойлар жұмысының өнімділігін арттыруға мүмкіндік беретін программалық камтама құру.

Алға қойған мақсатқа жету үшін келесідей технологиялар қолданылды:

- клиенттік интерфейсті құру үшін: Android Studio, Emulator;
- мәліметтер базасын жобалау үшін: СУБД PostgreSQL;
- серверлік логиканы құру үшін: Java.

Дипломдық жоба кіріспеден, 5 бөлімнен және қорытындыдан тұрады.

Кіріспеде таңдалған тақырыптың өзектілігі ашылады, өңдеу мақсаты қойылады және орындауға қажетті міндеттер, өңдеуді қолдану аймағы анықталады.

Бірінші бөлімде есепті шешуге арналған технологияларға талдау жасау мен таңдау келтірілген.

Екінші бөлімде қосымшаны жобалау сипатталады.

Үшінші бөлім қосымшаны құруды сипаттаудан, қойылған мақсатты практикалық шешудің қадамдық сипатталуынан тұрады.

Төртінші бөлімде өңделген жобаның экономикалық мақсаттылығын негіздеу қарастырылған.

Бесінші бөлімде жүзеге асырылатын жоба аясында еңбек шарттарын жақсарту бойынша шаралар қарастырылған.

Қорытындыда клиент-серверлік Android-қосымшаларды құрудың қазіргі заманғы құралдары меңгерілген және қойылған мақсатқа сай өңдеу орындалған, ары қарай қосымшаны толыққанды өнімге дейін жақсарту стратегиялары құрылған.

## Аннотация

Тема дипломного проекта: «Разработка приложения «Медицинская помощь гражданам г.Алматы» на платформе Android.

Цель дипломного проекта – создание мобильного приложения для медицинского учреждения, оптимизация и повышение качества услуг.

Для достижения поставленной цели использовались технологии:

- для разработки клиентского интерфейса: Android Studio, Emulator.
- для проектирования базы данных: СУБД PostgreSQL;
- для разработки серверной логики: Java.

В дипломный проект входит введение, 5 глав и итоговое заключение.

Во введении раскрывается актуальность выбранной темы, ставится цель разработки и задачи, которые необходимо выполнить, определяется область применения разработки.

В первой главе производится общие сведения для разработки приложения.

Во второй главе описывается проектирование приложения.

Третья глава представляет собой описание разработки приложения, содержится пошаговое описание практического решения поставленной задачи.

В четвертой главе производится обоснование экономической целесообразности разрабатываемого проекта.

В пятой главе рассматриваются мероприятия по улучшению условий труда в рамках реализуемого проекта.

Заключение освещает результаты исследования и разработки:

- изучены современные средства разработки клиент-серверных Android-приложений и выполнена разработка согласно поставленной цели;
- разработана дальнейшая стратегия улучшения приложения до полноценного продукта.

## **Annotation**

Theme of the graduation project: «Development of applications «Medical assistance to citizens of Almaty» on the Android platform.

The purpose of the study: analyze the workflow of office staff and develop software that allows increasing the productivity of specialists.

Technologies used to achieve the goal:

- development of the client user interface: Android Studio, Emulator
- creation of the database: DBMS PostgreSQL;
- development of the server logic: Java.

The graduation project includes an introduction, 5 chapters and a conclusion.

The introduction reveals the relevance of the chosen topic, sets the development goal and the tasks that need to be performed, and determines the scope of the development.

The first chapter contains general information for developing applications.

The second chapter describes the design of the application.

The third chapter is a description of the development stage of the application and contains a step-by-step description of the practical solution of the task.

The fourth chapter examines measures to improve working conditions in the framework of the ongoing project.

The conclusion highlights research and development results such as:

- studied modern development tools for client-server Android applications and developed according to the set goal;
- further strategies to improve the application in to a full product.

## Содержание

Введение	Ошибка! Закладка не определена.	
1 Общие сведения	Ошибка! Закладка не определена.	
1.1 Актуальность мобильных устройств	Ошибка! Закладка не определена.	
1.2 Анализ предметной области	Ошибка! Закладка не определена.	
1.3 Выбор операционной системы	Ошибка! Закладка не определена.	
1.4 Выбор среды разработки	Ошибка! Закладка не определена.	
1.5 Особенности и специфика разработки	Ошибка! Закладка не определена.	
1.6 Описание языка программирования	Ошибка! Закладка не определена.	
1.7 Тренды в создании мобильных приложений	Ошибка! Закладка не определена.	
1.8 Обзор рынка мобильных приложений	Ошибка! Закладка не определена.	
2 Проектирование системы	Ошибка! Закладка не определена.	
2.1 Логическая структура	Ошибка! Закладка не определена.	
2.2 Хранение данных		47
2.3 Характерные черты дизайна мобильных Android приложений		47
2.4 Методы проектирования		50
3 Описание работы приложения		54
3.1 Функционал приложения		54
3.2 Установка приложения	Ошибка! Закладка не определена.	
3.3 Совместимость приложения	Ошибка! Закладка не определена.	
3.4 Класс NewGridAdapter	Ошибка! Закладка не определена.	
3.5 Класс MemoryActivity	Ошибка! Закладка не определена.	
3.6 Класс MemoryStart	Ошибка! Закладка не определена.	
4 Техничко-экономическое обоснование		61
4.1 Определение сложности разработки мобильного приложения		61
4.2 Расчет затрат на разработку мобильного приложения		62
4.3 Расчет затрат на электроэнергию		64
4.4 Расчет затрат на оплату труда		66
4.5 Расчет затрат по социальному налогу		67
4.6 Амортизация основных фондов и прочие затраты		68
4.7 Определение возможной (договорной) цены ПО		70
<u>4.7</u> Вывод		71
5 Охрана труда и безопасность жизнедеятельности		73
5.1 Расчет естественного освещения		75
5.2 Расчет искусственного освещения		84
<u>    </u> Заключение		92
<u>    </u> Список литературы		8

<u>Приложение А. Техническое задание</u>	83
<u>Приложение Б. Листинг программы</u>	85
Приложение В. Акты внедрения	102



## Введение

Во втором квартале 2017 года настройки мобильных приложений увеличились на 346%. Согласно Kenshoo, текущий рынок мобильных приложений вырос на 100% и более по сравнению с предыдущими годами. Этот факт способствует развитию телекоммуникационной отрасли и открывает широкие возможности для предпринимателей, инвесторов и разработчиков. В настоящее время таких быстрорастущих рынков немного.

Сегодня сложно представить современного человека без мобильного устройства. Теперь вы можете оставаться в сети, находясь в пути, на работе и в игре, Интернет стал доступен везде благодаря сетям Wi-Fi и тарифам на мобильные конкурсы. Важно поддерживать связь с целевой аудиторией в тех областях, где большую часть времени проводят, и теперь это мобильный интернет. Когда пациент нуждается в клинике, колл-центр не должен ждать вызова. Но людям всегда легко оставаться в личной информации для постановки диагноза вызова врача, который ведет переговоры о приеме или собирается пойти в зону озвучивания специалиста. Это мобильное приложение, которое избавит вас от неудобств будущих видений. В то же время, самый удивительный сайт, который появляется на большинстве смартфонов, даже неадекватен. Используя ваших клиентов, вы перейдете на основной сайт и даже узнаете о медицинском центре, его сотрудниках и услугах, не выходя из него.

Трудности, устраняемые мобильным приложением:

- контроль качества предоставляемых услуг;
- низкая экономическая эффективность;
- большое количество пропущенных посещений;
- быстрый возврат от клиентов;
- отсутствие средств для поощрения новых услуг организации;
- отсутствие средств для привлечения и поддержки клиентов.

Эта тема становится все более актуальной в нашей стране с каждым годом для пожилых людей и инвалидов, что считается обязательным в дополнение ко многим наличным платежам, которые повышают эффективность системы общественной безопасности.

Были определены следующие ключевые задачи:

- проанализировать предметную область;
- разработать современное программное обеспечение;
- анализировать существующие среды разработки;

- Разработайте программу.

## 1 Общая информация

### 1.1 Совместимость мобильных устройств

Стоит ли говорить, сколько времени человек провел на своем смартфоне? Жизнь около 8 лет. Почти каждый смартфон выходит в интернет сразу после обеда и ложится спать.

Большая часть контента поступает со смартфонов или планшетов. В Казахстане 51% интернет-трафика поступает с мобильных устройств.

И это неудивительно, ведь смартфон всегда маленький, и на экран легко нажимать несколько раз, чтобы получить нужную информацию. Смартфоны глубоко укоренились в нашей жизни, что нам не нужно присоединяться к ним в течение длительного времени.

Подводя итог вышесказанному, можно сказать, что смартфоны являются самыми популярными устройствами цифровой связи. Вот почему разработка мобильных приложений является правильным решением.

#### 1.1.1. Анализ рынка мобильных приложений в Казахстане

В наше время произошли значительные улучшения в развитии мобильных технологий: обращая внимание на популярность популярных планшетных компьютеров, они практически заменяют ноутбуки. Полный доступ в Интернет для большего числа пользователей компьютеров - мобильные версии. Многие крупные сайты разрабатывают не полные версии, но мобильные браузеры.

Мобильное приложение - это компонент, который подключен к мобильному серверу и загружен на мобильное устройство, которое управляет пользовательским интерфейсом и бизнес-логикой устройства.

Сегмент рынка мобильных приложений был полностью создан в 2008 году. Apple была новатором, которая представила новую модель распространения. Через некоторое время Google вышел на рынок и стал серьезным конкурентом Apple. По первой модели был распространен мобильный контент в соответствии с VAS (Value Added Services) сотовых операторов - кроме того, все мобильные услуги включают в себя ряд дополнительных услуг в мобильных сетях. Эти услуги включают в себя: MMS-сообщения (через GPRS или 3G), различные контентные услуги (мелодии,

временно замененные почтовыми голосами и т. Д.), Услуги передачи данных, услуги LBS (место нахождения телефона абонента). сети), удаленный доступ к ресурсам корпоративной сети и т. д.). С появлением новых платформ рынок мобильных приложений изменил свою операционную систему.

В свою очередь на рынке мобильных приложений, как и на любом рынке, есть два ключевых показателя: «спрос» и «предложение». По данным J'son & Partners, по данным консалтинговых услуг, включая объем рынка мобильных приложений и их активное продвижение в магазине приложений до конца 2016 года, лидеры IT-индустрии вырастут на 167%. С точки зрения поставок рынок мобильных приложений сегментирован небольшими компаниями, которые принимают 80% клиентов, что приводит к крупным загрузкам и небольшим заказам.

Казахстанский мобильный рынок относительно молодой, с меньшим количеством развитых сегментов. В то же время он демонстрирует динамичную динамику роста, поддерживаемую быстрым расширением мобильного интернета. Таким образом, есть много возможностей разработать выгодное место для разработчика.

Рынки мобильных приложений в России уже находятся в зачаточном состоянии. Например, многие туристические агентства могут приобрести западную бизнес-модель, которая формирует свои собственные программы со списками туров, адаптированных для работы на смартфонах iOS или Android. Конечно, невозможно говорить о наших внутренних реалиях. Скорее всего, мы ожидаем того же через 5 лет. Две трети мобильных устройств в Казахстане работают на Android, и их популярность постоянно растет. Перспектива iOS в Казахстане весьма сомнительна, в том числе решение Apple увеличить стоимость своего iPhone на 70% в декабре 2014 года. Windows Phone не смогла получить большую часть рынка. В январе 2016 года Tele2 и Евросеть сообщили, что продажи впервые в истории рынка смартфонов в Казахстане сократились. Из-за высокой цены устройства выручка растет. По предварительным оценкам Евросети, в 2015 году в Казахстан было продано 25,5 миллиона смартфонов, что на 8,4% меньше, чем в предыдущем году.

По данным J'son & Partners Consulting, общий объем продаж смартфонов в Казахстане в 2015 году составил 25,3 миллиона единиц. Впервые с 2009 года они показали отрицательную динамику: в результате продаж смартфонов в 2014 году реальное падение составило 3%.

Примерно 40% посещений проводятся на мобильных устройствах.

В начале 2016 года несколько производителей смартфонов объявили о новом повышении цен на продукцию в Казахстане на уровне 15-20%. По данным J'son & Partners Consulting, рост стоимости оборудования ожидается в 2016 году, с тенденцией покупки энергоносителей в 2015 году и сохранением

спроса на бюджетные смартфоны. В 2016 году объем рынка смартфонов будет зависеть от будущей макроэкономической ситуации в стране и курса рубля.

При планировании, разработке и внедрении приложений для рынка мобильных устройств следует учитывать такой фактор, как движущая сила рынка. Он перемещается из одного состояния в другое, влияя на положение рынка. Другими словами, это повод для изменения статуса водителя рынка.

Мы будем отмечать основные драйверы рынка в Казахстане.

Великолепной киллойков замыковников. На рынке мобильных разработок в 2015 году рынок предложения перешел на рынок спроса. Многие клиенты стали обращать внимание на мобильный сегмент как на сегмент массовых коммуникаций;

- Увеличение мобильного потребления. Смартфоны, продажи планшетов растут, мобильный трафик растет;

- Увеличение мобильной рекламы. Стоимость мобильных объявлений, естественно, растет с ростом количества мобильных приложений и ростом конкуренции за пользователей;

Этот рынок способствует развитию разработчиков на местном рынке путем составления, маркетинга и создания более благоприятных условий для сотрудничества между владельцами платформ Google, Apple и Microsoft. Цель владельцев платформы - предоставить максимальное количество приложений для своей платформы и более точный сервис для пользователя.

Несмотря на недавний кризис в российской экономике, прогнозы рынка мобильной связи являются оптимистичными. По прогнозам J'son & Partners Consulting, в 2015 году рынок мобильных приложений в Казахстане вырастет в среднем на 186%, а к 2016 году его объем составит 1,3 миллиарда долларов. Факторы роста рынка обеспечат рост, быстрый рост сетей 4G и мобильную мобильность (мобильное потребление, услуги мобильного банкинга и системы мобильных платежей, такие как полный сдвиг в правильных потребителях и навыках общения, рост популярности услуг геолокации, мобильных социальных сетей и мобильных сетей). Разработчики) Messenger - английский «курьер» или «соединяющий». Это программа обмена мгновенными сообщениями для пользователей. Это их главное преимущество перед обычной электронной почтой. Многие компании, которые пытаются подать заявку на заявку. Стартер, печаль или потеря безопасности. Это проблема получения доступа к беспроводной сети. Это проблема, чтобы держать вас подальше от окружающей среды. Операторы смогут мобилизовать мобильные телефоны в доке.

Это было до 65% для пользователей, которые использовали мобильный телефон более 10%. В течение длительного периода, до 2020 года, 90% зон

будут покрыты мобильными телефонами. Для подключения и расширения мобильных сетей.

Это будет расти в Интернете. Если вы думаете, что смартфон до 50%. Я хочу сделать много смартфонов для социализации.

Мобилизация мобильных приложений. Тем не менее, обслуживание клиентов и решение будет прекращено для компании.

Есть три приложения для мобильных приложений. Тактическое общество общества в разных аспектах деятельности компании. Это проблема.

В январе 2015 года была добавлена онлайн-аудитория в 82 миллиона интернет-пользователей. Тереть. 66% казахстанцев живут в городах и 12-64 года в городах. Общее количество студентов составляет 5%. Доступ в Интернет; Более того, динамик на 1 год 2015 будет таким же, как и один из самых стабильных ПК и ноутбуков.

В первый раз нет бесплатного доступа в интернет. Только 90% мобильных пользователей в Казахстане победили. В небольших городах Казахстана, а также в небольших городах (Казахстан, 0+) оценивается в 50 миллионов долларов. Shelvovek В целом, 10% от общего числа пользователей Интернета в стране (14% от общей численности населения 11,8 млн.) Являются по телефону. Для участков оценки были разработаны участки для приобретения земли.

## **1.2 Предопределенный анализ полюсов**

Многие люди любят быть активными, и они не могут идти в ногу с этим. Я всегда должен был идти. Если вы не хотите получать медицинскую помощь.

Приложение MediCare решило эту проблему. Это означает, что мы не мобильный медицинский центр программы и медсестра. Только 90% мобильных пользователей в Казахстане победили.

Также данный сайт не имеет русской локализации, вследствие чего мало удобен для пользования русскоязычных пользователей. На рис. 1.1 представлен скриншот сайта ([www.damumed.kz](http://www.damumed.kz))

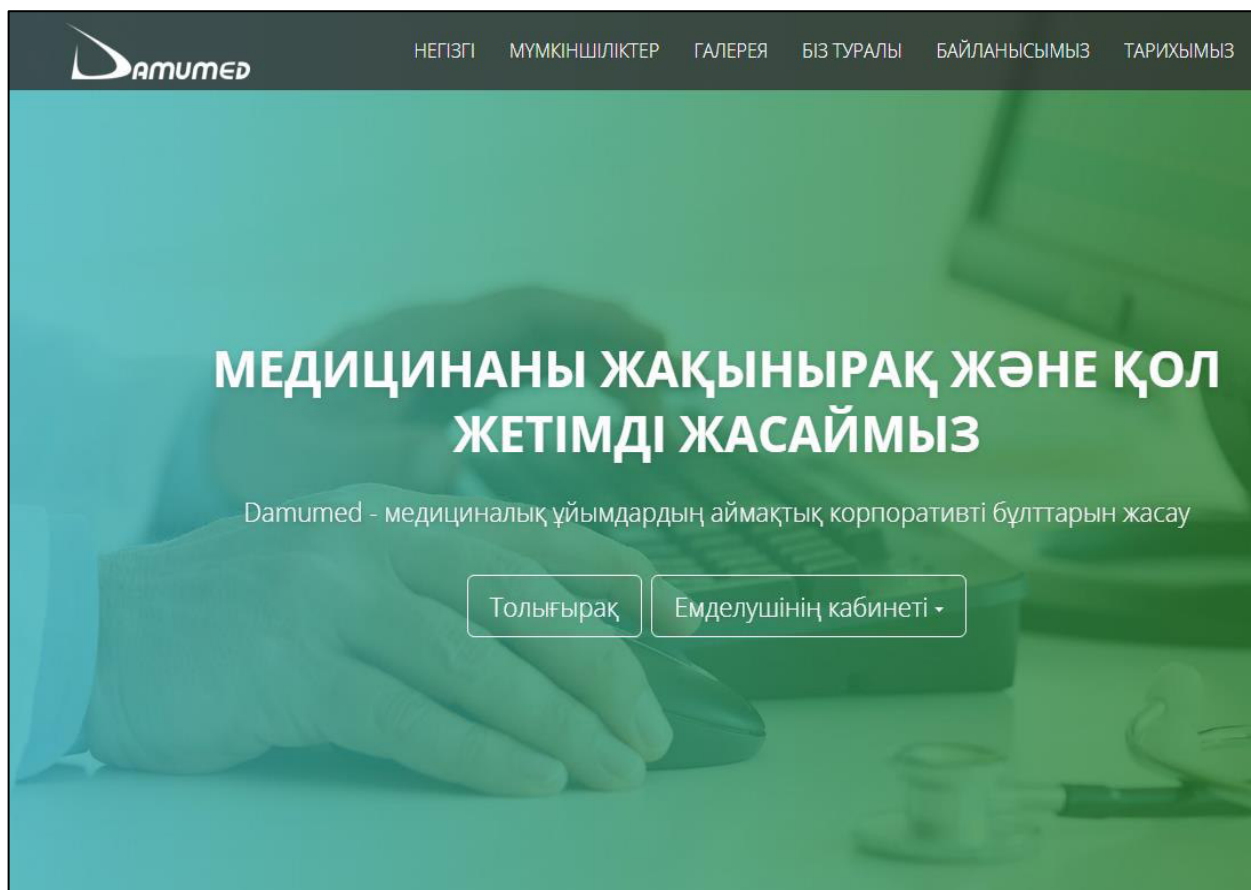


Рисунок 1.1 – Главная страница сайта [www.damumed.kz](http://www.damumed.kz)

### 1.3 Выбор операционной системы

На данный момент существует много различных мобильных операционных систем. Но все они имеют разную популярность. Если приложение не является кросс-платформенным, то разработка для целевого узкого круга пользователей может обречь его заранее до отказа. Ведь чем больше пользователей попытается приложение, тем больше они будут заинтересованы. Поэтому одной из основных задач является выбор операционной системы, для которой будет разработан приложение.

На рис. 1.2 - диаграмма, показывающая доли различных мобильных операционных систем на рынке.

Согласно статистике NetMarketShare, личный опыт и эта диаграмма, представленная на сайте (<http://it-news.club>), определили три наиболее популярные операционные системы, которые необходимо учитывать (в порядке возрастания популярности):

- оконный телефон;
- iOS;
- андроид.

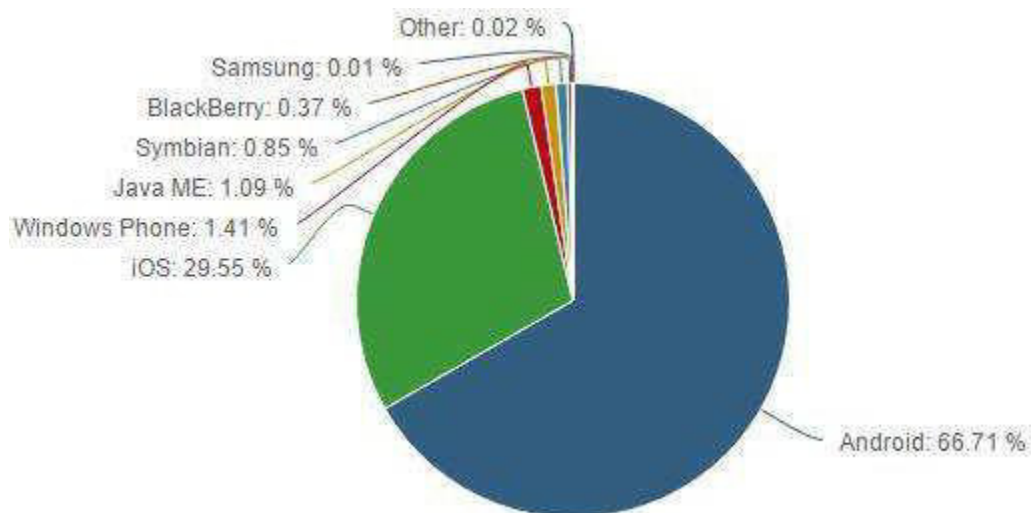


Рисунок 1.2 – Диаграмма показывающая популярность ОС

Согласно статистике NetMarketShare, личному опыту и этой диаграмме, предоставленной сайтом (<http://it-news.club>), были выделены три наиболее популярные операционные системы, которые необходимо учитывать (в порядке увеличения популярности):

- телефон с операционной системой Виндоус;
- iOS
- Android.

### 1.3.1 Windows Phone

Windows Phone - мобильная операционная система, разработанная американской компанией Microsoft. Он установлен на 1,41% смартфонов.

Из-за небольшой доли рынка и неопределенных перспектив количество приложений для Windows Phone значительно уступает количеству приложений для Android и iOS, но магазин приложений Windows Phone Store может удовлетворить практически любые потребности, поскольку количество приложений в нем превышает 300 тыс.

Однако, учитывая ежегодное падение популярности, разработка программы для этой ОС не является привлекательной перспективой, так как охват пользователей будет чрезвычайно рискованным.

### 1.3.2 IOS

iOS - операционная система для смартфонов, планшетных компьютеров и операторов, разработанная американской компанией Apple. Доля iOS на рынке мобильных операционных систем (29,55%) точно отражает рыночную долю телефонов iPhone на рынке смартфонов, поскольку iOS устанавливается только на iPhone и iPad (мобильные устройства Apple), а Android и Windows Phone используются по-разному. , производители смартфонов. Количество приложений iOS в магазине приложений App Store превышает миллион.

Эта операционная система очень привлекательна для разработки, так как имеет большую аудиторию пользователей. Но, несмотря на это, есть ряд недостатков:

- количество пользователей iOS все еще меньше, чем пользователей Android;
- недостойная учетная запись разработчика для загрузки приложения в App Store;
- отсутствие необходимых инструментов для развития.

iOS (24 июня 2010 г. - iPhone OS) - мобильная операционная система, разработанная и изготовленная американской компанией Apple. Был выпущен в 2007 году; изначально для iPhone и iPod touch, позже - для таких устройств, как iPad и Apple TV. В отличие от Windows Phone и Google Android, он доступен только для устройств Apple. При разработке приложений используется iOS SDK, язык разработки - Objective-C. iOS SDK (ранее iPhone SDK) - это набор для разработки для iOS, выпущенный Apple в марте 2009 года. IOS SDK доступен только для Mac OS X. Разработчики могут устанавливать любую цену, превышающую минимальную, установленную их приложением, которая будет распространяться через App Store, с которой они получают 70%. Кроме того, они могут распространять свои заявки бесплатно, в этом случае они должны платить только членские взносы в размере 99 долларов в год.

Apple iOS использует вариант того же ядра XNU, что и Mac OS X. SDK разбит на следующие наборы:

- Какао Touch;
- мультитач контроль;
- поддержка акселерометра;
- иерархия видов;
- локализация;
- поддержка камеры;
- мультимедиа
- открытый;



- микширование и запись звука;
- проигрывание видео;
- форматы изображений;
- кварц;
- анимационное ядро;
- OpenGL ES;
- сервисное ядро.

Популярность iOS явно уступает популярности Android.

Главным преимуществом Apple является живучесть аудитории. При таком низком объеме рынка (примерно 12%), половина дохода от приложений была на устройствах iOS - \$ 6400000000 в год. В App Store разработчик получает в среднем \$ 0,24 за загрузку. Учитывая высокую стоимость самой дешевой модели iPhone, покупатели iOS будут хорошей целью для разработчиков приложений. Кроме того, устройства Apple не так просто сломать. Это еще одна причина высокой покупательной способности платформы.

Разработчики предпочитают платформе Apple. Частично это связано с тем, что в разработке следует учитывать лишь несколько форматов дисплеев. Пожалуй, и потому, что компания смогла упаковать средства разработки в лучшие продукты по сравнению с конкурентами. Публикация созданной программы может вызвать серьезные проблемы. Аккаунт разработчика Apple iOS стоит \$ 99 в год. За эти деньги вы можете скачать приложение для дальнейшей модерации. С тщательной политикой, Apple может сталкиваться с каждым бизнесом.

Серьезность компании над внешними разработчиками устанавливается в основных рисках проекта. Лучше заранее подготовить дополнительные расходы.

### **1.3.3 Windows Phone**

Windows Phone - мобильная операционная система, разработанная компанией Microsoft, вышла 11 октября 2010. 21 октября начались поставки первых устройств на базе новой платформы. Все программы Windows Phone 7 создаются с помощью .NET-управляемых. В настоящее время C# - единственная поддерживаемая язык программирования. Microsoft Visual Studio 2010 Express для Windows Phone можно скачать бесплатно. Она включает в себя XNA Game Studio 4.0 и эмулятор на экране телефона и интегрируется с Visual Studio 2010. Визуальные элементы и анимации для приложений Silverlight можно создавать в Microsoft Expression Blend.

Платформы Silverlight и XNA для Windows Phone 7 имеют несколько общих библиотек. Это означает, что некоторые библиотеки XNA можно использовать в Silverlight и наоборот. Однако невозможно построить программу, сочетающую визуальные элементы обеих платформ. Silverlight обычно используется для программ, которые можно классифицировать как программы или утилиты. Описание размещения элементов управления и панелей в этих программах осуществляется с помощью языка XAML (Extensible Markup Language). В файлах выбранного кода могут быть реализованы операции инициализации и некоторая логика, но их основная цель - обрабатывать события управления.

С помощью Silverlight можно внедрять программы Rich Internet Applications (RIA) в Windows Phone, включая мультимедийные средства и Интернет. Для Windows Phone был введен Silverlight 3, который не содержал некоторых функций, которые не подходили для телефона, но были компенсированы рядом приложений.

Основная цель XNA - создание высокопроизводительных игр. В 2D-играх спрайт и субстраты описаны с помощью растровых изображений. Для 3D-игр создаются трехмерные модели. Действие игры, в которой графика перемещается по экрану и запрашивается пользователем, выполняется интегрированной игровой петлей XNA. Можно легко нарисовать границы и предположить, что Silverlight используется для программ и XNA для игр. Однако это не должно ограничиваться.

Несомненно, Silverlight можно использовать для реализации игр, и 19 традиционных приложений могут быть построены на XNA, хотя это будет очень трудно. Silverlight подходит для игр с низкими графическими требованиями или с векторной графикой, а не растровой графикой, скорость которой зависит от ответа пользователя, а не от таймера.

Windows Phone SDK содержит такие компоненты.

- Microsoft Visual Studio 2010 Express Edition для Windows Phone;

- Эмулятор Windows Phone;

- Создание Windows Phone SDK 7.1;

- SDK и DRT для Silverlight 4,

- расширение Windows Phone SDK 7.1 для XNA Game Studio 4.0;

- Microsoft Expression Blend SDK для Windows Phone 7, - Microsoft

Expression Blend SDK для Windows Phone 7.1;

- WCF Data Services Client для Windows Phone;

- Microsoft Advertising SDK для Windows Phone.

Лидером в количестве загрузок из всех платформ является, конечно, Google App Store с годовым ростом продаж в размере 1,2 миллиарда долларов.

Количество загрузок из магазина Windows Phone во много раз ниже, а его доход составляет 0,95 миллиарда долларов.

Выход, что в Среднем разработчик зарабатывает около \$ 0,23 за одно скачивание с Windows Phone магазина (доходы Android разработчик за Одно скачивание йими программы составляет \$ 0,04). В достоинства Windows Phone магазин можно записать и Слабый конкуренции Внутри магазин. В большинстве компаний разработки программ для Windows Phone стоит на третьем месте (отвращение после IOS и Android) поэтому часто к ней просто не доходят. Например, Instagram для WP8 Вышел несколько месяцев назад и только в бета-версии [4].

### 1.3.4 Android

Android - операционная система для смартфонов и множества других Устройство. Изначально разрабатывалась Калифорнийского компания Android Inc., как потей купив американский поисковый гигант Google. Доля Android на рынке ОС составляет 66,71. Количество приложений для Андроид в магазине приложений Google Play превосходит 1,430,000th

Й операционная система самая аудиторию пользователь, а отжим, как бул сказано вище, приверните бильшего отзыв среди установил Приложения. Аккаунт разработчика для Публикации приложения в Play Маркет тоже платный, против стоит намного меньше чем в App Store и платеж осуществляется единовременно (в App Store ежегодной платеж).

Проанализировав самые популярные мобильные операционные системы бул решено, что разработка будет выполнять под устройства с ОС Android, так как она Наиболее соответствует заявленным требованиям.

Язык Java активно використовується для создания мобильных приложений пида операционной системы Android. При цьих прогрев компилируются в нестандартных байтах-код, для использования ой виртуальной машины Dalvik. Для такой компиляции використовується дополнительный инструмент, а саме Software Kit развития, его разработали представитель компании Google.

Разработка приложений можно вести в середовищи Затмение, Використовуючи При этом плагин - Android Средства разработки (ADT) или в IntelliJ IDEA. Версия JDK при этом должна быть 5.0 или вище.

С точки зрения программиста, Android - платформа, абстрагирующая разработчика от ядра и позволяет ему создавать код на Java. Android имеет Несколькими Полезная возможен.

Во-первых, ца фреймворка, что предлагают большой набор API для созданной ризние типов приложения И, крое того, что обеспечивает

возможности повторного использования и замены компоненты, которые предлагают платформа и сторонних Приложения.

Как и вторая, наличие виртуальной машины Dalvik, что отвечает за запуск приложений на Android.

Для работы над приложениями есть множество библиотек: Бионический (библиотека стандартных функций, несовместимо с Glibc) Мультимедийные библиотеки на базе PacketVideo OpenCore (поддерживают Такие форматы, как MPEG-4, H.264, MP3, AAC, AMR, JPEG и PNG) SGL ( движок двухмерной графики) OpenGL ES 2.0 1.0ES (двигатель трехмерная графика) Поверхностное менеджер (Обеспечивает для приложений доступ к 2D / 3D) WebKit (готов движок для веб-браузера; обрабатывает HTML, JavaScript) FreeType (движок ОБРАБОТКИ шрифт) SQLite (легкая СУБД доступны для всех приложения) SSL (протокол, что обеспечивает безопасную передачу Сейчас по сети).

Сообщение По сравнению с обычным приложение Linux Приложения Android подчиняются Дополнительный правила: Контент-провайдеры - Обмен Сейчас между Приложения; Менеджер ресурсовой - доступ к таким ресурсам, как файлы XML, PNG, JPEG; Сообщение Manager - доступ к строке состояние; Руководитель направления - управление активно Приложения.

Google предлагает для свободного скачивания инструментарий для разработки (комплект разработки программного обеспечения), которые назначения для x86 машин пида операционных систем Linux, Mac OS X (10.4.8 или выше), Windows 14 XP, Windows Vista и седьмой Windows

Для разработки нужно JDK 5 или более новой. Разработка приложений для Android можно вести на языке Java (не ниже Java 1.5). Существует плагин для Затмение - Android Средства разработки (ADT), назначение для Eclipse, версий 3,3-3,7. Также существует плагин для IntelliJ IDEA, который облегчает разработка Android приложений, и для среды разработки NetBeans IDE, который, начиная с Версии NetBeans 7.0, перестав быть экспериментальный, хоть пока и не является официальной.

Кросс того, существует MOTODEV студии для Android - комплексно среду разработки на базе Eclipse, что позволяет работать непосредственно с Google SDK. В 2009 году на приложениях к ADT был опубликование Android Native Development Kit (NDK) - пакет инструментарий и библиотек, что позволяет реализовать часть программ на языке C / C ++. NDK рекомендуется использовать для разработки участков кода, критически к скорости.

В 2013 году Google представила новую среду разработки Android Studio, основано на IntelliJ IDEA от JetBrains.

В 2013 г. году видбегося релиза Embarcadero RAD Studio - xe5. Возможность разработка нативных приложения для платформы Android.

Процесс создания Android приложения не требует дополнительных устройств, а также собственных устройств Android (в принципе, можно обойтись и эмулятором).

Для данной исследовательской компании IDC, операционная система от Google продолжает лидеры как на русском (77%), так и на мировом (78%) рынка смартфонов. Причина - у крупных производителей - здесь и сверхдешевые телефоны, и мощные смартфоны за несколько тысяч долларов.

Впрочем, большая охота поможет и основные недостатки разработки для Android. Если ориентироваться на широкий спектр устройств, то примите производительность, непрерывную количество изображений и памяти. Как следует - увеличить затраты на проектирование нескольких интерфейсов и дополнительного тестирования. Стоимость разработки в ситуации увеличивается пропорционально поддерживаемых устройств.

Кроме того, обладатель Android не любят платить за приложения. Открытие - это легкая установка программы - чаще всего достаточно искать файл на форуме. С другой стороны, покупка дешевых моделей изначально НЕ нацелены на большие трассы в магазинах.

В Google Play может быть любой разработчик. Для этого нужно заплатить \$ 25 за помощью банковских карт, загружает скриншоты, добавьте его описание и код. В Google наружу столь жесткие требования к качественному приложения, как в Apple. Это снижает и риск затрат на доработку. Кроме того, доступны альтернативные магазины, такие как Amazon Apps, Samsung Apps, Opera Apps, SlideMe и Яндекс.Збериганья. В любом своем политическом модерации, однако присутствие в них дает обзор и присоединение пользователей.

### **1.3.5 Эмуляторы мобильных устройств**

Сперва приложение тестируется в образцах разработки по внедрением эмулятора. После этого приложения тестируется на устройствах. Эмуляторы является простым способом тестирования приложения для мобильного телефона, а не пользователем его физически. Представленные список доступных инструментов для тестирования приложений среди самых популярных мобильных операционных систем:

- Google Android Эмулятор Android Эмулятор выпускается на Windows, как отдельное приложение без необходимости пополнения и установления Android SDK;

- официальный эмулятор Android SDK включает в себя эмулятор мобильного устройства, реализующий все аппаратные и программные особенности типичного устройства;

- mobiOne MobiOne Разработчик веб-IDE для Windows, который помогает разработке программ, тестирование, скачивание и введение мобильные веб-приложения устройств, таких как iPhone, BlackBerry, устройства на Android и Palm Pre;

- testiPhone Основанный на браузере симулятор для быстрого тестирования веб-сайтов для iPhone. Работает с внедрением Internet Explorer 7, Firefox 2 и Safari 3,

- iPhone предоставляет точную среду веб-браузера, разработку Safari. Может быть использован для разработки веб-сайтов для iPhone. HE эмулятором iPhone. iPhoneу выпускается только на Mac OS X 10.4.7 и выше;

- симулятор BlackBerry существует множество официальных эмуляторов BlackBerry. С любыми из них возможна проверка того, как работать с устройствами, которые работают с программой.

### **1.3.6 Магазины приложений**

Растворимые иницирующие существуют от мобильных операторов и производителей. Разработчики программного обеспечения могут предлагать и публиковать свои программы в магазинах, которые доступны для расчета доходов по продажам.

Яркими известными являются App Store Apple, где только такие приложения могут принимать и запускаться на устройстве iOS (также известные, как огороженный сад), и Android Market Google, добавляя в любой рабочий устройство с ОС Android. Компания HP / Palm также каталог Palm App, где пользователи могут работать на HP / Palm webOS, могут загружать приложения непосредственно с устройства или просматривать ссылки на приложение с использованием уникального метода. Мобильные операторы Telefica Group и Telecom Italia запустили мегаплатформенный магазин-приложение для своих абонентов. Производитель мобильных устройств Nokia запустил Ovi App Store для смартфонов Nokia.

### **1.3.7 Платформы для разработки**

Каждая из платформ мобильных приложений имеет интегрированную среду разработки, которая предоставляет инструменты, позволяющие разработчику программировать, тестировать и развертывать приложения на целевой платформе. Чтобы сравнить мобильные приложения на Java с

приложениями, написанными на других языках, сравните платформу Android с iOS и Windows Phone.

## **1.4 Выбор среды разработки**

Наиболее популярные среды разработки для операционной системы Android:

- затмение;
- майкрософт ксамарин;
- андроид студия;

### **1.4.1 Eclipse**

Eclipse - это бесплатная программная платформа с открытым исходным кодом, управляемая Eclipse Foundation. написано

На языке программирования Java и основной целью его создания является повышение производительности процесса разработки программного обеспечения. Сам Eclipse не является средой разработки приложений для мобильных устройств, но его можно подключить к отдельному плагину ADT (Android Development Tools).

Однако эта среда разработки НЕ предназначена специально для разработки мобильных приложений, поэтому на этапе разработки могут возникать различные проблемы, которых можно избежать, выбрав другую среду разработки.

### **1.4.2 Microsoft Xamarin**

Microsoft Xamarin - это платформа для разработки мобильных приложений для создания собственных приложений для iOS, Android и Windows из общего кода C # или .NET, которая позволяет многократно использовать между платформами от 75% до почти 100% кода.

Программы, написанные на Xamarin и C #, имеют полный доступ к API базовой платформы и возможность создавать собственные пользовательские интерфейсы и компилировать код в машину, поэтому влияние на производительность во время выполнения незначительно.

Разработчики, знакомые с C #, .NET и Visual Studio, могут рассчитывать на одинаковые возможности и производительность при работе с Xamarin для мобильных приложений, включая удаленную отладку на устройствах Android, iOS и Windows, без необходимости изучать родные языки, такие как Objective-C или Java. Удивительно, но многие высокопроизводительные приложения с

красивыми пользовательскими интерфейсами, такие как NASCAR, Aviva и MixRadio, создаются с помощью Xamarin.

Однако из-за низкой популярности этой среды разработки и ожидаемых проблем с поиском информации из нее пришлось отказаться.

### 1.4.3 Android Studio

Android Studio - это интегрированная среда разработки (IDE) для платформы Android, анонсированная 16 мая 2013 года на конференции Google I / O [6]. Это молодая среда разработки, но она уже очень популярна среди разработчиков под управлением операционной системы Android. У него есть ряд положительных черт. Вот некоторые из них:

- расширенный редактор макетов: WYSIWYG, возможность работы с компонентами пользовательского интерфейса с помощью Drag-and-Drop, функция предварительного просмотра макетов на нескольких конфигурациях экрана;

- разработка приложений на базе Gradle;

- различные типы коллекций и генерация нескольких файлов .apk;

- рефакторинг кода;

- шаблоны для основных макетов и компонентов Android;

- Поддержка Android Wear и разработка приложений для Android TV;

- Android Studio 2.1 поддерживает Android N Preview SDK, что означает, что разработчики могут начать работу над созданием программы для новой программной платформы.

Кроме того, эта среда разработки, как следует из названия, ориентирована на приложения в операционной системе Android, что является значительным преимуществом по сравнению с другими средами разработки.

Вместе, после анализа существующих сред разработки, была выбрана среда разработки Android Studio, так как она идеально соответствует потребностям.

Приложение находится в стадии разработки, а также в Интернете имеется много информации о его настройке и работе с ним.

Разработчики, знакомые с C #, .NET и Visual Studio, могут рассчитывать на одинаковые возможности и производительность при работе с Xamarin для мобильных приложений, включая удаленную отладку на устройствах Android, iOS и Windows, без необходимости изучать родные языки.



### 1.4.3 Java Me

Не говоря уже о платформе JavaME. Несмотря на то, что диаграмма сейчас считается устаревшей, диаграмма показывает, что она все еще занимает значительную долю рынка мобильных устройств.

Основная проблема мобильных телефонов заключается в том, что все они запускают прошивку. Если смартфон может расширить функциональность устройства (на смартфонах и подставке), то это невозможно в обычных сотовых телефонах. А вот Java.

Идея заключается в том, что команды не передаются непосредственно к процессору, а к виртуальной машины Java (JVM - Java Virtual Machine). В Java ME, он также называется KVM, Kilobyte Virtual Machine. Вместо команд процессора программа Java является командой байт-кода, которую должна выполнять машина Java. Для того, чтобы программа работала, достаточно, чтобы в системе была установлена такая же Java-машина. На компьютерах она установлена в отдельную программу, а в большинстве телефонов она является частью прошивки.

Для программ, предназначенных для Java ME, существует специальное название - midlet. Их часто путают с апплеты, но они совершенно разные понятия. Апплеты - это программы Java, которые предназначены для работы в других программах, например, в интернет-браузере, а мидлет является полностью независимыми программами. Игра, "читатель", ICQ-клиент - ничего. Мобильные приложения не распространяются как разбросаны файлы, а как специальные архивы и файлы описания. Это файлы JAR и JAD. JAR означает Java Archive. На самом деле, это самый распространенный Zip-архив, только с другим расширением. В нем сохраняются все программные файлы: .class (они содержат байт-код), файлы ресурсов (например, изображения или звуки) и файл манифеста. Последняя описывает программу: имя, производитель, версия и другие данные.

JAD является файлом описания (что означает Java Application Descriptor). Он содержит ту же информацию, что и файл манифеста, плюс размер архива и путь к нему (URL). Для чего это, если вся информация уже содержится в файле манифеста? И так, что вы можете увидеть информацию о мидлет, не загружая архив, который может быть достаточно большим. Понятно, что для установления нужного файла JAR. JAD-файл на некоторых старых телефонах также требовался, но практически любой современный телефон может обойтись без проблем.

Одной из основных концепций, существующих в программировании, есть API (Application Programming Interface), интерфейс прикладного программирования. Это набор команд, которые программа может обеспечить

устройствами. Например, большинство современных телефонов имеют камеру. Но его простого присутствия для съемки с Java недостаточно. Нужно, чтобы был менеджер камеры API - иначе телефон просто не «поймет», чего он хочет от него. API и определяет функциональность устройства.

Наиболее базовый API, на котором построены все остальные - это или CDC (конфигурация подключенного устройства), или CLDC (конфигурация подключенных ограниченных устройств). Оба являются сильно урезанными наборами команд большого java.

CDC предназначен для смартфонов, коммуникаторов и КПК (как более мощный), в то время как CLDC остается для мобильных телефонов, конфигурация проще. Теперь есть две версии CLDC: 1.0, которые почти не используются, и 1.1, основное отличие, от которой предыдущая версия поддерживает расчеты с плавающей точкой. Оба они были созданы давно, но почему-то они еще не готовят замену.

Поскольку мобильные телефоны очень отличаются по уровню устройств от компьютеров, мне нужен API, который может обеспечить программистам удобные меню, сохранить настройки программы и другие функции, характерные для мобильных телефонов. Эта задача занимает API, который называется MIDP. Это слово, вероятно, видели все, кто принимал каталог телефонов. Это означает "мобильный информационный устройство". На данный момент существует несколько версий. MIDP 1.0 был создан очень, очень давно, в 2000 году.

Он наложил много ограничений на программу - его возможности были очень малы. Поэтому в 2002 году вышла новая версия MIDP 2.0. Эта версия используется и по сей день, и почти во всех новых телефонах. Теперь слова "Java ME" и "MIDP 2.0" почти синонимичны.

По сравнению со своим предшественником эта версия предоставляет намного больше возможностей: приемлемый звук, расширенные сетевые возможности, богатые инструменты для создания интерфейсов и игровой графики. Именно MIDP 2.0 дал толчок развитию мобильных игровых устройств. Стоит также отметить MIDP 2.1, который был разработан в 2006 году. Он не предоставляет никаких новых возможностей, но эта версия разъяснила некоторые особенности реализации Java на телефоне.

#### **1.4.4 Кодовое имя одного SDK**

Codename One позиционирует себя как уникальный SDK, позволяет разработчикам создавать свои собственные приложения для мобильных Java, используя единую кодовую базу.

Израильская компания заявляет, что с помощью Codename было создано более 1000 приложений, что делает платформу «одной из самых быстрорастущих».

Codename One основан на Java и может создавать пользовательские программы для iOS, Android, BlackBerry и Windows Phone 7. Идея состоит в том, что вся логика программируется один раз и сохраняется в каждой программе, а интерфейсы создаются индивидуально для разных систем каждый раз.

Затем приложение собирает из выходных данных собственное приложение в облачной службе. Это, по мнению создателей, поможет избежать фрагментации, связанной с разными платформами.

## **1.5 Особенности и особенности разработки**

### **1.5.1 Основные функции Android**

Операционная система Android достаточно уникальна. Для хорошего результата разработчик может знать много тонкостей и особенностей этой операционной системы. При проектировании есть несколько трудностей, которые важно учитывать. Как из:

После применения после установки требуется в два раза, а в некоторых случаях даже в четыре раза больше места, чем размер исходной программы;  
- на встроенной флешке скорость работы с файлами снижена в несколько десятков раз при небольшом объеме свободного места;

Es Процессы могут использовать до 16 МБ, а в некоторых случаях также 24 МБ ОЗУ. Между ядрами и приложением находится уровень API и нативный код - уровень библиотек.

Android может запускать неограниченное количество приложений, которые позволяют ОЗУ на вашем устройстве. Но только одно из приложений является базовым и отображается на экране. Из открытого приложения вы можете перейти в фоновый режим или запустить новое. Этот процесс визуально напоминает вкладки браузера.

Экран пользовательского интерфейса представлен классом активности в коде и содержится в процессах. Деятельность может занять больше времени. Отключение можно восстановить снова, сохранив всю необходимую нам информацию. В этом случае специальные механизмы используются для описания действий, основанных на намерении. Когда вам нужно позвонить, отправить письмо, показать окно или выполнить действие, это называется намерением.

Кроме того, Android включает в себя сервисы, подобные демонам в Linux, для выполнения необходимых действий в фоновом режиме, таких как воспроизведение музыки или видео. Для этих целей поставщики контента (контент-провайдеры) используются для обмена данными между приложениями.

### **1.5.2 Общая схема приложений Android**

Приложения Android используют окна в своей работе как окна Windows, хотя в этой системе окна имеют другое название - активность. Как и Windows, любое окно имеет свой жизненный цикл и свои особенности. Схема представляет жизненный цикл приложения Android, изображенный на рис. 1.3. Чтобы создать новое окно, вызывается метод `onCreate ()`; при разработке этот метод переназначается, и он иницирует программу и ее элементы.

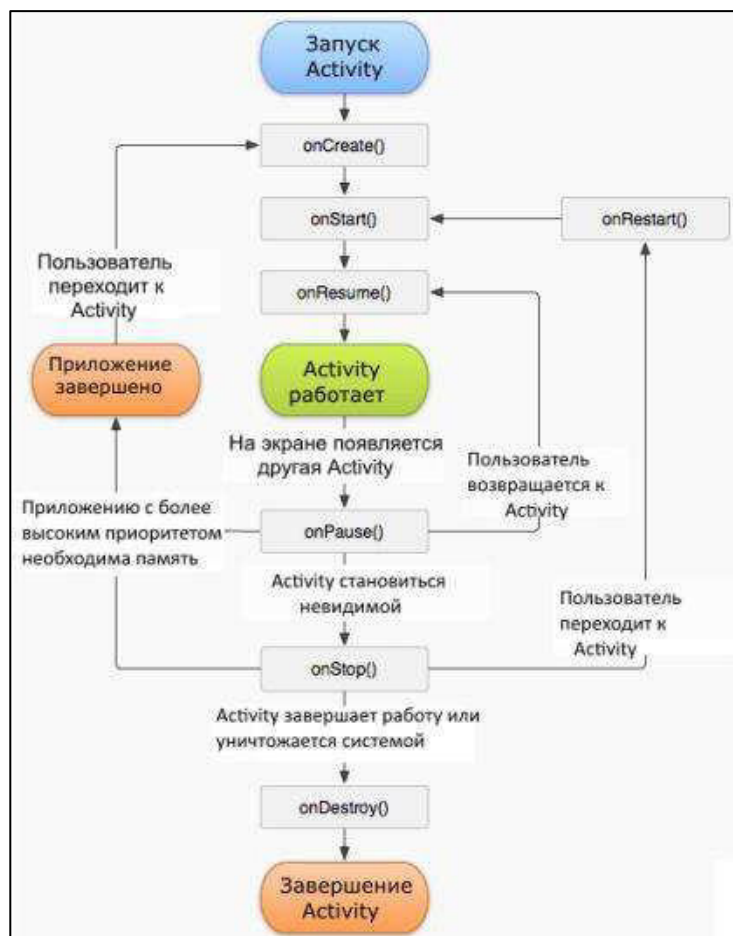


Рисунок 1.3 – Жизненный цикл приложения для системы

Ниже приведены метод `onStart()` и метод `onResume()`.

При составлении используется метод `onDestroy()`. Получить последовательность методов вызова и более полное их описание можно найти на официальном сайте.

## 1.6 Описание языка программирования

Java - это объектно-ориентированный язык программирования, разработанный Sun Microsystems с 1991 года и официально выпущенный 23 мая 1995 года.

Язык Java был создан в рамках проекта по разработке программного обеспечения для различных бытовых приборов. Проект был запущен на C++, но вскоре возник ряд проблем. Очевидно, необходим программно-независимый язык программирования, который позволяет создавать программы, которые должны быть получены отдельно для каждой архитектуры и могут использоваться на разных процессорах для различных операций.

Java была необходима для создания интерактивных продуктов для Интернета. На самом деле большинство архитектурных решений при создании Java это было продиктовано желанием предоставить синтаксис, аналогичный C и C ++. Java использует почти идентичные соглашения для объявлений, передачи параметров, операций и управления потоком исполняемого кода. Java добавляет все хорошие возможности C ++.

Джава:

Java широко использует свои апплеты (апплеты) - небольшие, надежные, динамические, недоступные платформы для активных сетевых приложений, встроенных в веб-страницу. Java-апплеты могут быть настроены и распространены.

- Java-приложения с высокой степенью сложности и объектно-ориентированной разработкой, сочетание простого и знакомого синтаксиса с надежной и удобной средой разработки. Это позволяет широкому кругу программистов быстро создавать новые программы и новые апплеты;

- Java предоставляет программисту богатый набор классов объектов для четкой абстракции многих системных функций, доступных при работе с окнами, сетевыми интерфейсами и интерфейсами ввода-вывода.

Основным преимуществом Java является обновление и поддержка языка, большое количество написанных библиотек, решающих различные задачи, а также тот факт, что приложения на основе Java, написанные на языке программирования, могут работать на всех устройствах, имеющих виртуальную машину Java.

Для написанных приложений Android используется версия Java 1.7. Написанные программы компилируются в виртуальной машине JVM. Android использует собственную виртуальную машину Dalvik.

## **1.7 Тенденции в создании мобильных приложений**

Очевидно, потенциал для развития велик. Фактически, прогноз состоит в том, что к 2020 году доход от мобильных приложений превысит 1 миллиард долларов во всем мире. Последний годовой отчет Эрикссон о мобильности показывает, что к 2020 году в мире будет доступно 6,1 миллиарда пользователей смартфонов, а с 2017 года Gartner прогнозирует, что к концу года будет загружено более 268 миллиардов мобильных приложений, что приведет к вырубке в 77 долларов. делает вложение одним из самых популярных вычислительных инструментов для пользователей во всем мире "(рисунок 1.4)

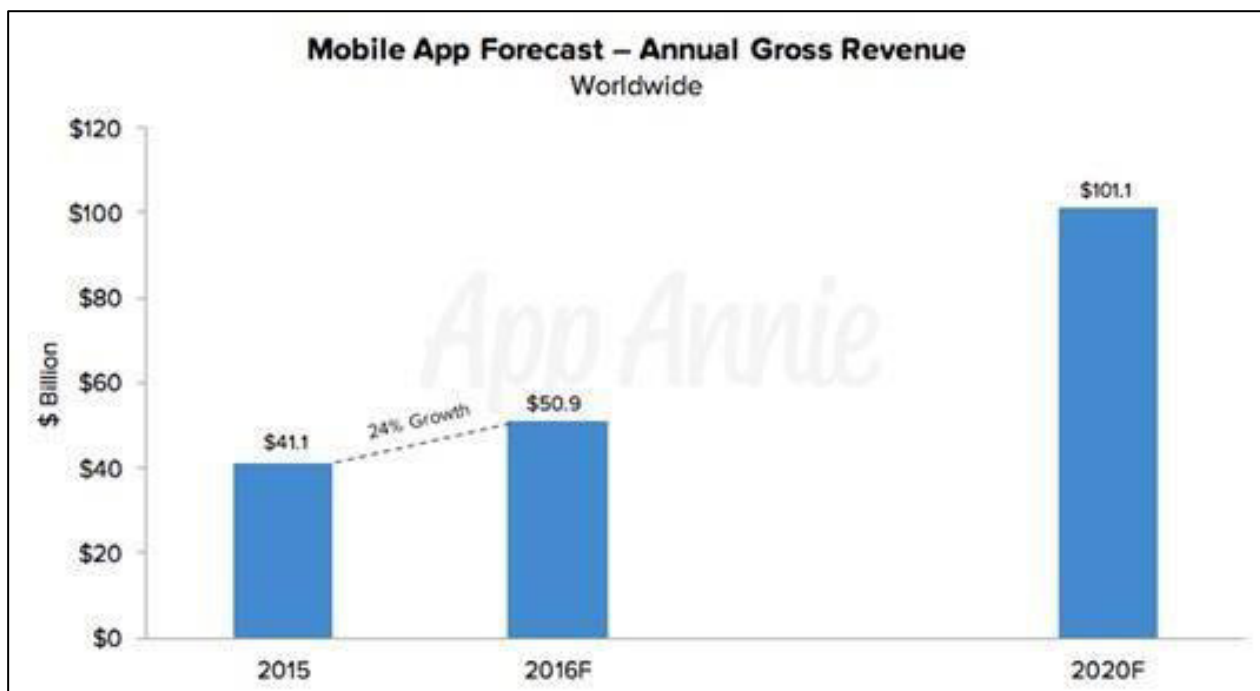


Рисунок 1.4 – Прогноз развития рынка мобильных приложений

Очевидно, что развитие данных мобильного телефона ясно. Продажа корпоративных данных для мобильных приложений. Star Gartner vd 2015 voку vaqzue на изючжу может потребовать от корпоративных лиц с этим фактом.

Mi bachimo, scho dodatki для мобильных устройств с большим складыванием. Моблинные стратегии повнни икорисовувинструменты, методы.

Тенденции Qia можно увидеть в 2017 году. Все более и более спокойно, номофобия и BYOD (чтобы прийти к власти). В словах, необходимых для матери всех вин.

Все, что вам нужно заработать, цепати працювати зи собственными специалистами на ходу. Деяки компании просуувт собственные программы (например, Red Robin, ресторан ресторанов, як свѣлили программу для списков Керування оцикування).

Поговорите с Gartner, расскажите своим собеседникам необязательно инвестировать в развитие микро-додатков, которые позволят спико выкоровывавать певческие корпоративные решения (например, программы для передачи информации).

Росробка AP и BP. Dvvatok dodatkiv AR (добавь реальную реальность) в VR (виртуальная реальность) є продовженням, просто слышим после феноменального успеха Pokemon Go. Я думаю, Pokemon podim doviv, posho all galuzi, yakю мьют зовсім нову площадку для ростерной доходів обятыт тим, хто досяг скорую. В 2017 году VR и AR в области дистрибуции (на

самом деле, прогноз Gartner, выяснить в регионе, узнать, как узнать, как найти на «сделай сам» с реальностью), не поиск, узнать, как узнать, как идти на рынок, но не искать, узнать, как узнать, как узнать, как перейти к реальности, посмотреть, что они ищут, узнать, как узнать, как идти к рынок, но не видеть, узнать, как посмотреть, что вы найдете на доске до реальности, а затем узнать, как смотреть - посмотреть, как найти купон на доске до реальности;

Прочитайте обзор Goldman Sachs Global Investments на программной платформе AR-i VR с помощью наступательных ракет (Рисунок 1.5):

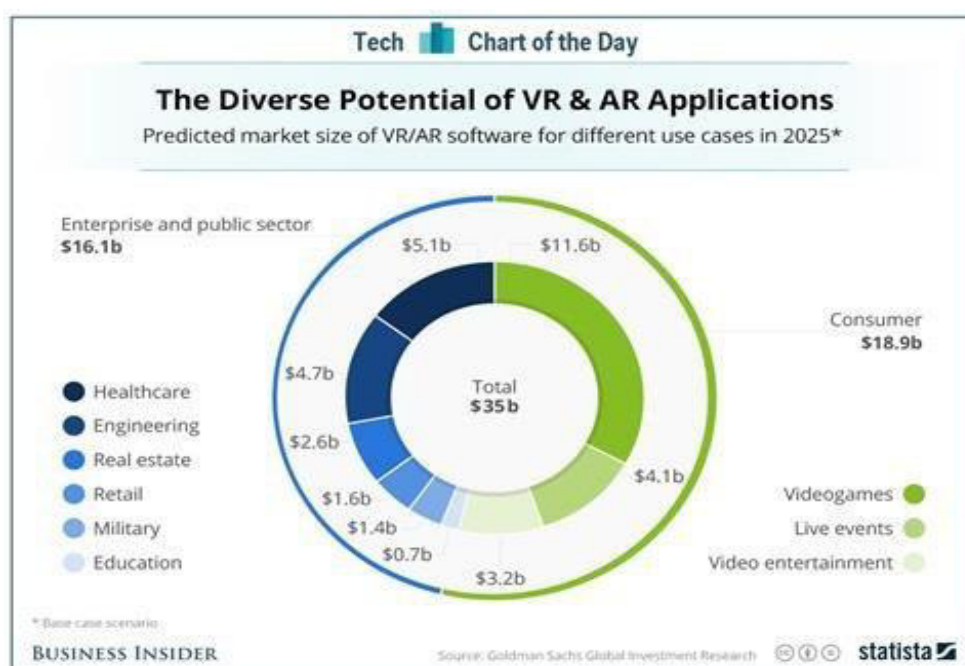


Рисунок 1.5 – Рассмотрение потенциала VR & AR приложений

Маяки и создание дополнений в базовой локации. Хотя это не волнует, я должен согласиться с этим. Все, что я могу сказать, это не только незначительные дополнения, но и заметные дополнения к данным интеграции. Другие тексты, имущество и номер.

Это маяк, утверждающий, что все необходимые услуги, эквивалентные доступным, к 2013 году достигнут 43,3 миллиарда долларов. «Анализ позиции маркетолога в 2016 году требует увеличения в направлении увеличения, вывода средств:» Каждый кто использует телефоны в США, использует услуги на основе локальных сетей », - говорится в заявлении, что существуют значительные возможности для увеличения (рис. 1.6).



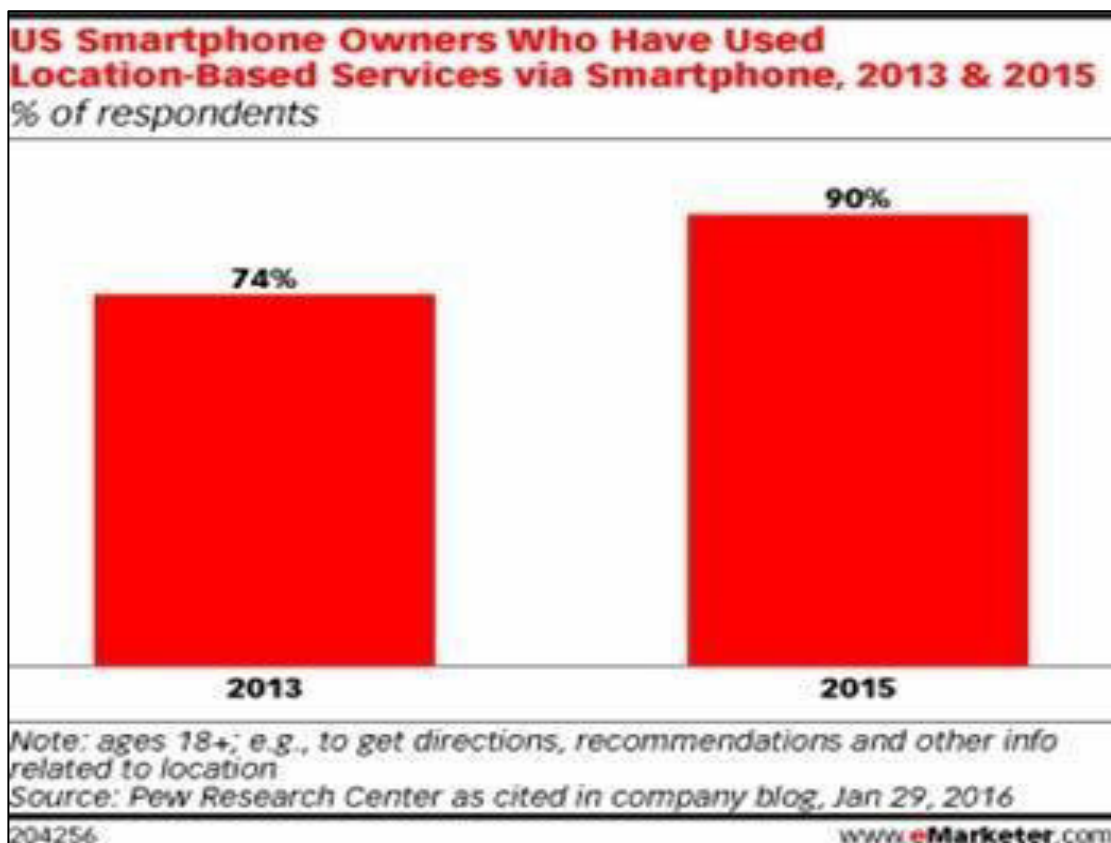


Рисунок 1.6 – Демонстрация увеличения использования приложений с функцией определения местоположения

Развитие аукциона мобильных дополнений движется впечатляющими темпами, в то время как, по окончательной информации, в 2012 году, по оценкам экспертов, его цена составляла \$ 53 млрд, а надежды в 2016 году были выражены примерно на увеличение на сто миллиардов долларов. На основе такого рода усредненной статистики можно сделать вывод о том, что переносной обмен считается чрезвычайно глубоким.

Основная составляющая заработка-прибыль от внутренних покупок в приложении, внутренняя реклама в сфере дополнений и созыва глубокой информации (big data). Такие группы считаются более популярными, а также связи с общественностью, эффективность, маркетинговые услуги и необходимые целевые дополнения (зерновые. 1.7).

К более развивающимся и растущим в достаточно быстром темпе базарам следует причислить Юго-Восточную Азию и Латинскую Америку.

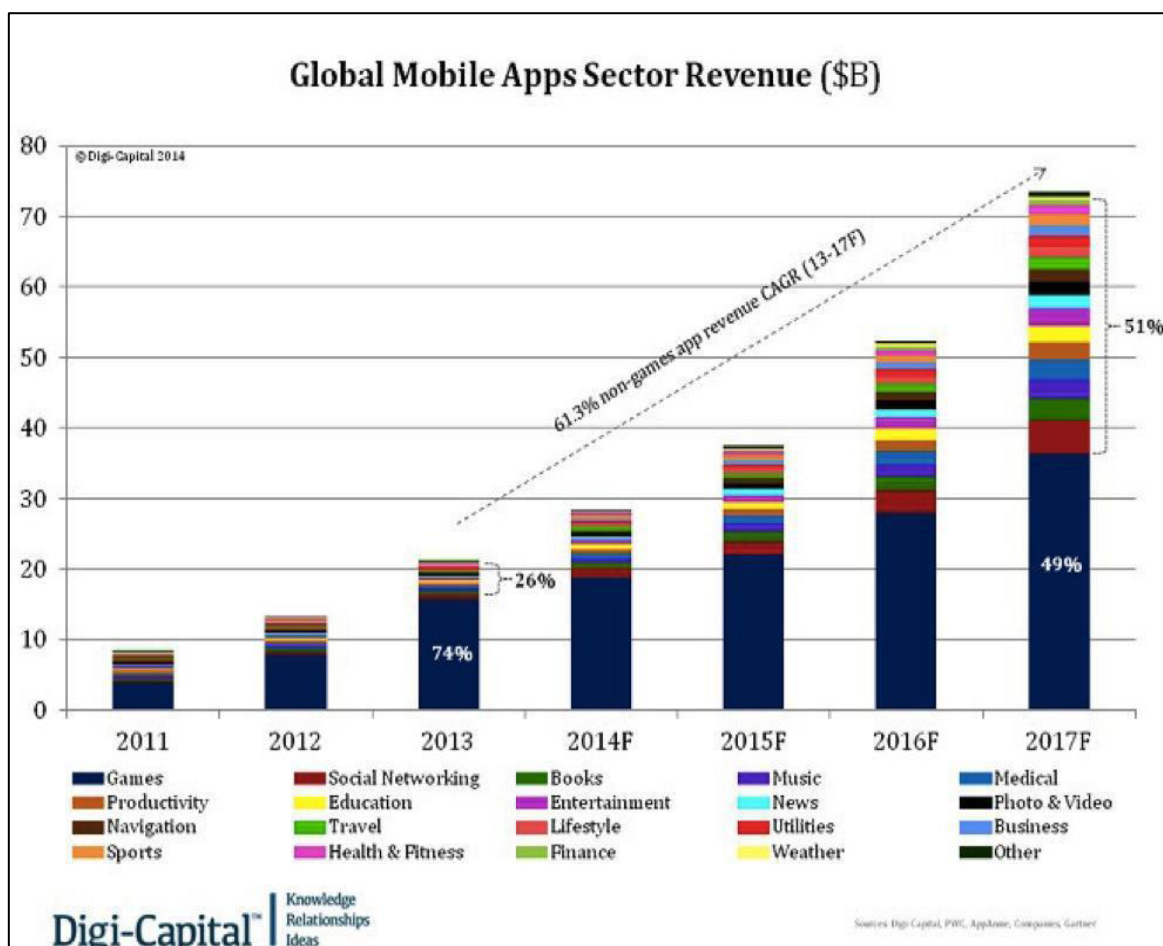


Рисунок 1.7 – Рост для рынка мобильных приложений

Статистика показывает, что любые 8 создателей, а непосредственно 2,3 миллиона человек в обществе формируют свои мобильные дополнения. По данным компании Apple, в 2013 году в ходе конференции WorldWide Developer Conference было объявлено о публикации 1, двадцати пяти миллионов дополнений к AppStore, которые существовали скачанные пользователями пятьдесят миллиардов раз, прибыль от реализации которых дала создателям 5 миллиардов долларов. Таблица 1.1. показана умеренная прибыль производителя различных концепций оператора.

По данным мониторинга, количество загрузок мобильных дополнений достигнет двухсот миллиардов в срок, а кроме того, прибыль составит те же 63,5 миллиарда долларов. Говоря о рыночной цене мобильных платежей по мнению всего общества, предполагается его увеличение с \$ 235 млрд в 2013 году до \$ 721,3 млрд к 2017 году.

Таблица 1.1 - Количество задействованных в разработке кадров

Показатель/компания	Google	Apple	Microsoft

Число пользователей(в млн. чел.)	900	600	12
Число приложений (в тыс. шт.)	800	1250	160
Число разработчиков(в тыс. чел.)	150	235	45
Число загрузок приложений (в млрд. раз)	48	50	65
Доход разраб.а (в млн. дол.)	900	5000	100
Средний доход на разработчика	\$6,000	\$21,276	\$2,222

Подобный значительный рост популярности мобильных приложений обуславливается небывалым увеличением спроса на планшеты, смартфоны и другие мобильные устройства. (рис.1.8а, 1.8b)



Рисунок 1.8 (а) – Загрузка приложений Android и iOS

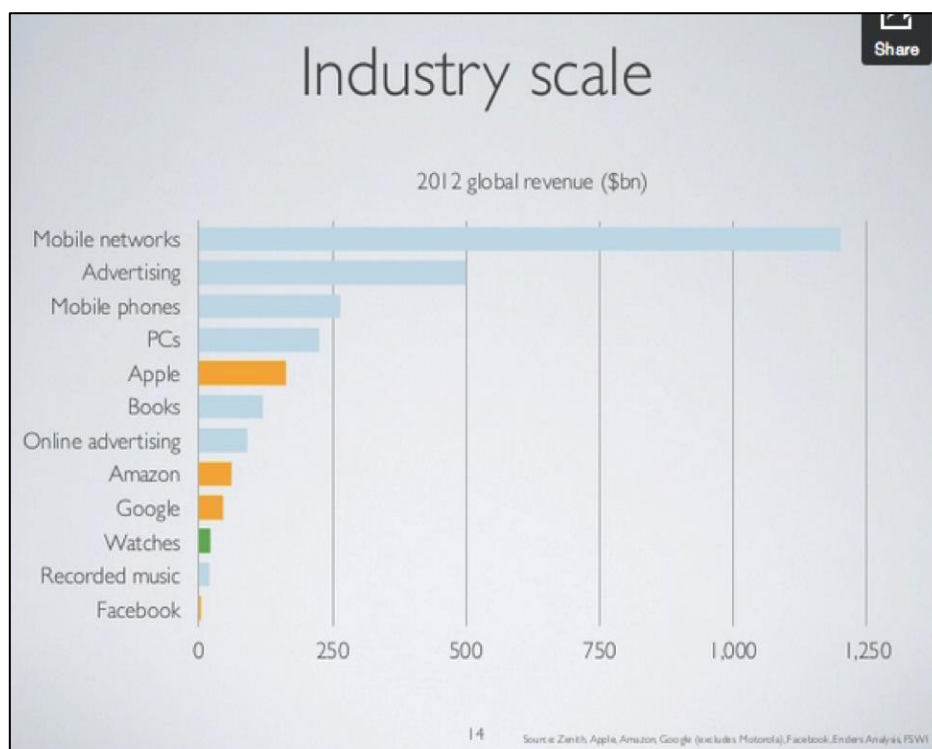


Рисунок 1.8 (b) –Использование приложений Android и iOS

Согласно диаграмме, наблюдается заметное увеличение по абсолютно всем категориям добавок. Для целей специалистов, изучающих обмен мобильными дополнениями, очевидно, что мобильные программные продукты владеют путем реорганизации устройств рекламной информации в способе распространения и передачи медиаданных. Такие особенности, а также размер охватываемых контактов, их колебания и коэффициент удержания в мобильных положениях составляют довольно значительное представление. Для того, чтобы доказать эту информацию, можно привести статистику по данным Соединенных Штатов Америки, в которых место использования мобильных устройств значительно опережает другие медиа-каналы, и непосредственно, составляет 82% от единичного периода.

В наш временной период значительная доля товаров мобильного торгова относится к положениям В2С, по этой причине, то, что продукты питания В2В находятся на начальной стадии назначения торгова. Однако возможности весьма впечатляющие. Значительная доля ученых и экспертов в речи говорит о том, что обмен мобильными добавками – единственное и более перспективное направление цифровой технологии.

Целевая аудитория мобильного приложения. Согласно статистическим данным, 67% пользователей телефонов в Соединенных Штатах Америки использовали его с целью установления отношений с внешним обществом через облигации интернета, и большинство не оставляли жилье из-за

отсутствия аксессуаров. Аналогичная ситуация зарегистрирована и в Казахстане, общество с достаточно низким заработком обычно выбирает ПК-телефон, чтобы иметь стабильные отношения с внешним обществом. По результатам исследований, почти пятьдесят процентов всего пользователя занимаются дополнениями, а 2 из его тридцати процентов используют, чтобы выделяться постоянно.

Если проанализировать индивидуальные особенности по абсолютно всем пользователям, то в этом случае значительная доля относится к возрастному диапазону от двадцати пяти до тридцати лет, находятся в Союзе, проживают в городе/пригородном регионе и имеют высший уровень созидания (зерновые культуры. 1.9)

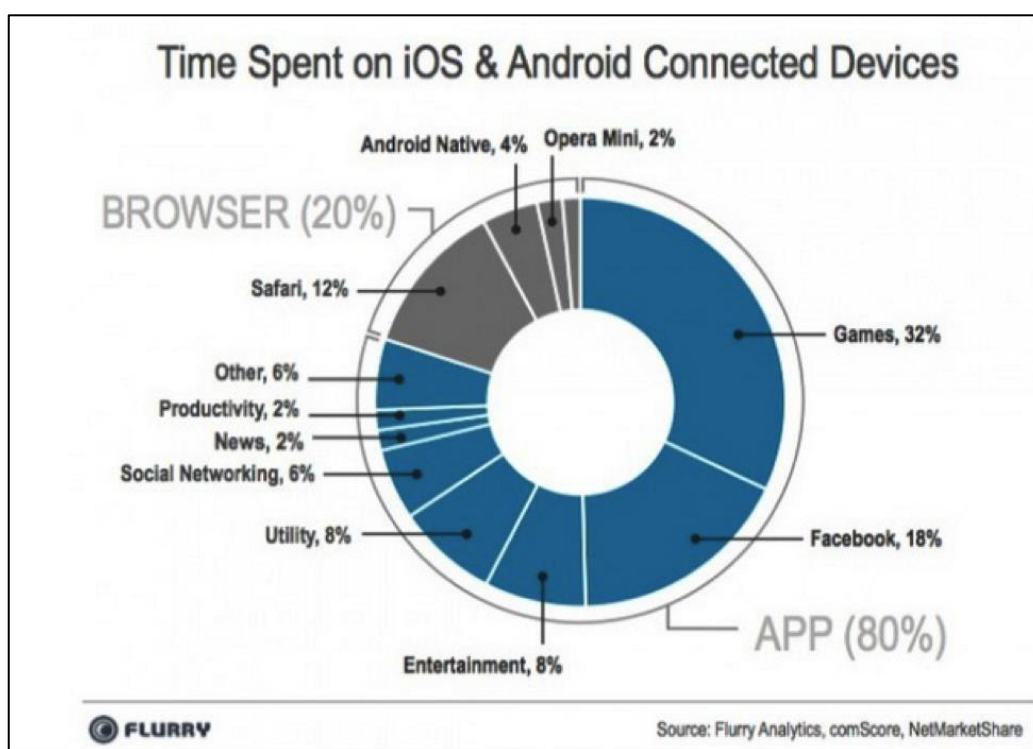


Рисунок 1.9 – Время, проведенное при использовании мобильных приложений

В принципе, получается, что общий состав пользователей состоит из стипендиатов, достаточно молодых, интеллигентных, получающих доходные средства раньше, чем те, на кого наружное дополнение не распространяется.

Компании, которые строят мобильную стратегию, не заставляют работать в базе этой статистики, отмечают совершенно новый тип пользователя/потребителя – пребывание в постоянном контакте с каналами связи. В телефонах такие аспекты больше приветствуются пользователями, а также многозадачность и мобильность. Google порекомендовал свой

собственный список пользователей мобильных дополнений, в соответствии с демографическими характеристиками.

Повышает лояльность пользователей за счет использования мобильных дополнений (очевидно, учитывая значительное присутствие отдельных сделок). Подтверждено, что лояльные пользователи демонстрируют результаты загрузок раньше, чем простые, и тратят в своем приложении до 2-х раз больше периодов, чем в мобильных версиях сайтов (зерновые. 1.10 а, 1.10 б)

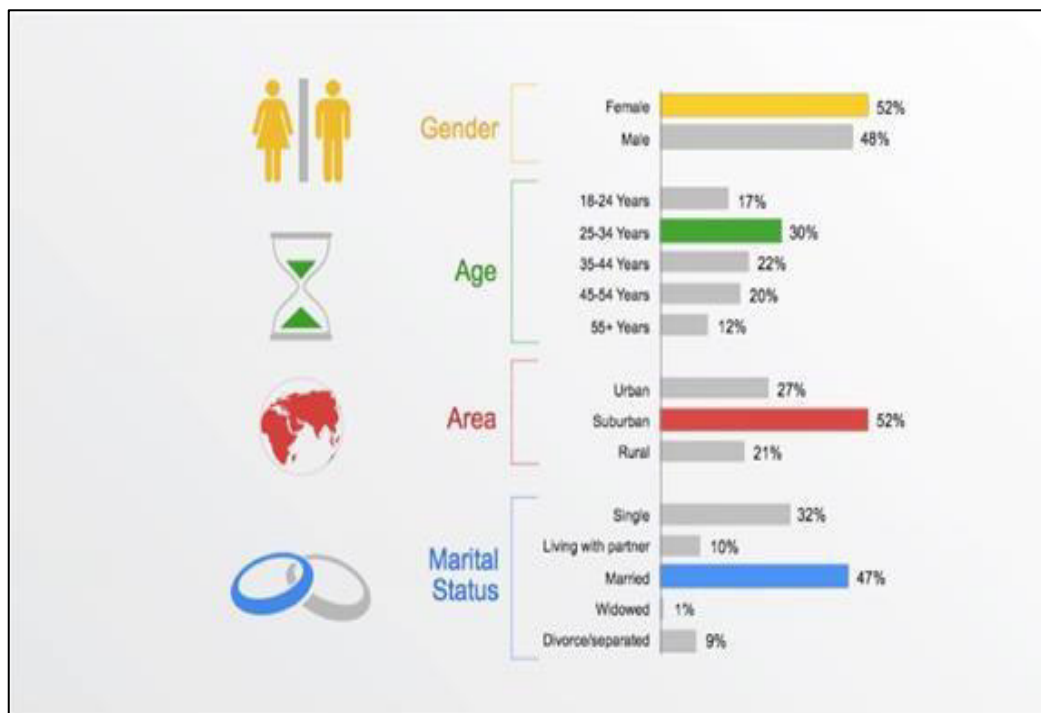


Рисунок 1.10 (а) – Демографические показатели, влияющие на использование мобильных устройств

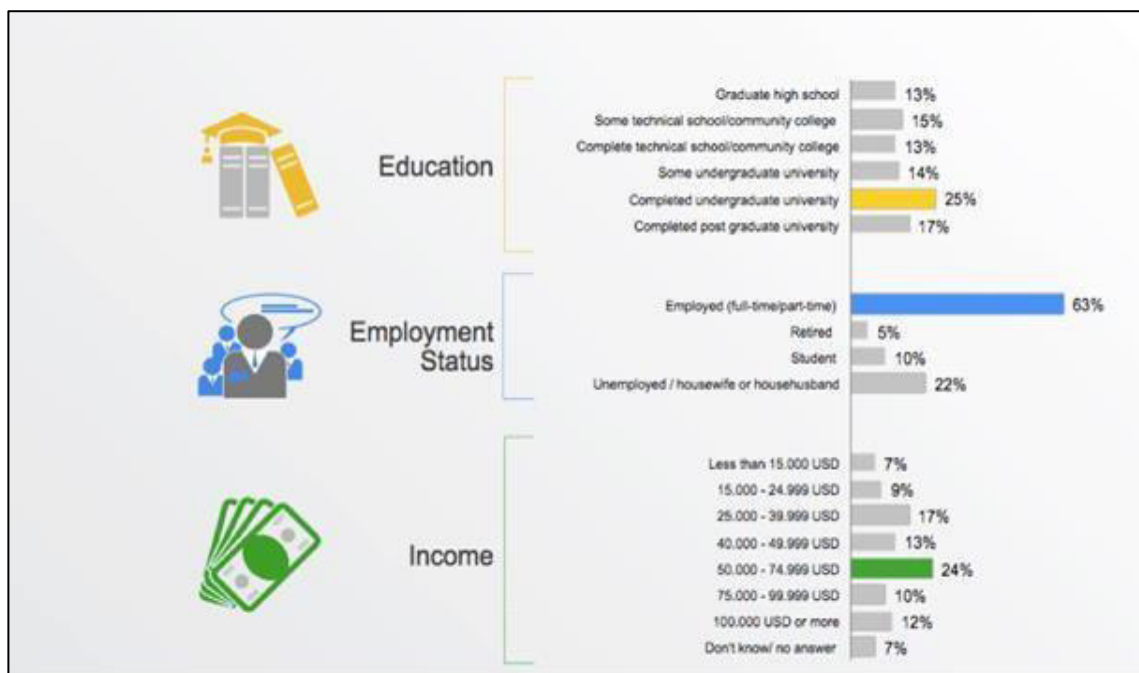


Рисунок 1.10 (b) – Социальные факторы, влияющие на использование мобильных устройств

Варианты мобильной разработки. Результатом станет новый рынок для поставщиков услуг мобильной связи, которые к 2018 году вырастут до 32,4 млрд. Долл. США (см. Рисунок 1 ниже).

Сегодня ни один вендор не может сделать все это, но преследуют гостя поставщики шести категорий - цифровые агентства, консультанты по менеджменту, мобильные специалисты, специалисты по разработке продуктов, системные интеграторы и телекоммуникационные компании. Выигрыш для вендоров, которые делают эти инвестиции, будет занять место за вашим столом в качестве долгосрочного партнера в вашем успехе участия (рис. 1.11).

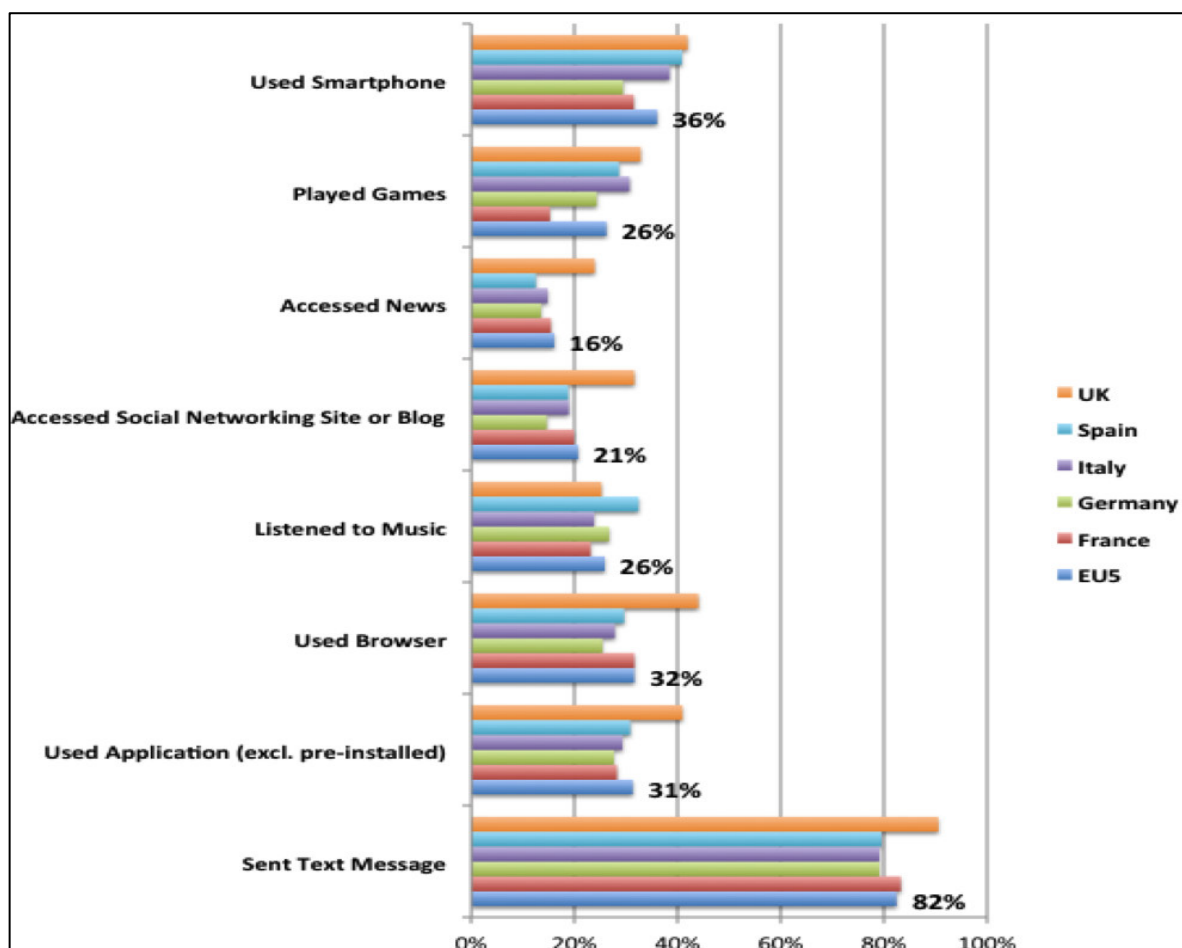


Рисунок 1.11 – Показатели использования мобильных приложений в разных странах мира

В 1-й модификации нынешнего подвижного состава дополнения не было просто вышестоящего лица. Однако в версиях 2, 3 и 4 Ваш мобильный комплект будет новейшей концепцией взаимодействия личности с целью помочь людям "осуществлять деятельность в их прямом контексте и в условиях необходимости".

Реализация этой миссии и использование положительных аспектов, близких к гарантийному объединению с вашими покупателями и работниками - указывает на разрешение гораздо более значительных проблем, чем снижение вашего сайта или очистка экрана с вашей концепцией SAP. Это относится к обслуживанию покупателей и сотрудников, чьи умы были переданы для того, чтобы ждать, что вам нужно, в каком месте вам нужно, в каждый период. Если Модификация 1-й с надстройками до 250 000 \$, то это не чем-в данном случае это необычно с целью 2-й версии цена 2 млн \$ (злак.1.12).



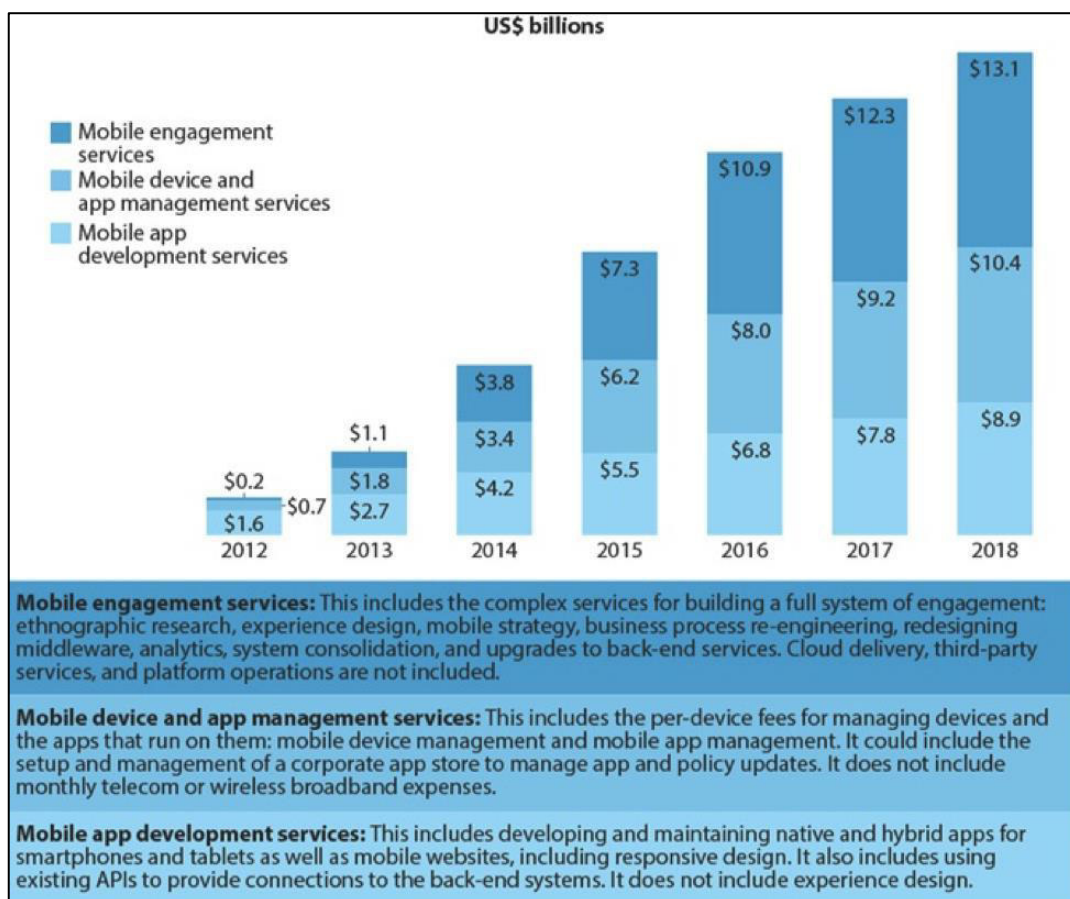


Рисунок 1.12 – Прибыль от реализации сервисов разработки мобильных приложений

Вы потратите эти средства в том случае, если ваши клиенты на самом деле хотят выполнить в мобильных устройствах, а затем сделать значительно простые мобильные эмоции о сложных концепциях взаимодействия. Вам дополнительно предстоит реорганизовать закрытый ключ движения, концепцию и питание, чтобы помочь народам в их "мобильных факторах".

Эта последняя 32-миллиардная биржа состоит из 3-х видов услуг:

- услуги по мобильному взаимодействию. Семо вступают в групповой уход при формировании абсолютных концепций взаимодействия: субэтническое исследование практики проектирования, мобильная политика, реинжиниринг предприятий, переориентация перехода программного обеспечения, специалист, унификация концепции и развития на отечественные отрасли. Предоставление облаков, иностранной услуги и процедуры с платформой никак не вошло;

- услуги в целях мобильных устройств и дополнений: содержит оплату за счет устройства за счет управления устройствами и дополнениями, которые в них работают: Управление мобильными устройствами и

управление мобильными дополнениями. Кто-то, способный создавать и управлять коллективными дополнениями торгового центра с целью управления обновлениями, дополнениями и политиками. Не содержит kadamasan расходы в сфере телекоммуникаций и беспроводного широкополосного соединения;

- услуги по изучению мобильных дополнений: это создание и помощь близких и смешанных дополнений с целью телефонов и планшетов, а также мобильных интернет-сайтов, включая тщательный дизайн. Один дополнительно содержит применение существующего API, с целью обеспечения комбинаций с внутренними концепциями. Кто-то не обладает навыками дизайна. (Писатель: Тед Шадлер считается вице-президентом и главным специалистом Forrester Research, обслуживающим ИТ-руководителей. Вы сможете следить за ним в Twitter (@TedSchadler).

Работая по плану, были изучены 3 основные задачи в соответствии с привлечением работодателей, которые используются для этой модификации учебной группы. Сформировать мощную бизнес-модель. Несмотря на то, что многие работодатели хотят "преуспеть, сделав выгоду", значительно меньшее количество людей способно показать какому-то бизнесу важность частных инвестиций в обществе. В целях содействия развитию предпринимательства-чемодан концепции привлечения персонала, категории ФСТ консультирование основательно исследовал влияние нанимать лучших практик. Они поделились своими знаниями не так давно в отчете об инциденте: "инвестирование в способности начальной степени: стратегия сбережений, которые действуют". Один содержит 10 ключевых инициатив по образцам, используемым в службе с точным бизнесом-модель, которая существовала, применялась фирмами в последнем Орлеане, поскольку они проанализировали, поскольку у них есть все шансы показать эффективность инвестиций в частные инвестиции. Я считаю, что эти инвестиции в исследования и информацию имеют большую роль, в связи с тем, что в том числе у более социально ориентированных работодателей есть акционеры, которые в них наблюдаются.

Содействие взаимному обучению. Практика в современном Орлеане была необходима для целей предпринимательства-фаворитов, так как кто-то позволял им создавать отношения со своими сверстниками в других фирмах. Подразумевается, что увеличение этой экосистемы Чемпионов сформирует решимость не только провести эксперимент, но и продвигать свои лучшие практики с деловыми и общественными секциями. Это значительно облегчает реализацию плана таким образом, если есть общество сверстников, которые подразумевают ваши трудности и стоят с вами, и я заметил, что это начинает реализовываться в современном Орлеане. В 1-й вечер встречи

единственный участник воскликнул, что в том числе, если кто-то не купил ничего другого с 3-х дней, линия мысли, что кто-то купил в первоначальном количестве дней, стоимость поездки, чтобы экипировать верных Чемпионов.

В нашем эксперименте расширение возможностей и предоставление некоторым внутренним чемпионам инструментов, ресурсов и межличностной помощи рассматривается как важный компонент укрепления перемен. Многочисленные компании, которые я стараюсь заинтересовать, достаточно значимы для их работы, а также сложные концепции. Изменение может быть трудным для тех, кто функционирует в этой концепции, в том числе с возможностями. Обеспечение того, чтобы чемпионы были либо "внутри процессора" в новаторских фирмах, является важным компонентом этой деятельности. Эти фавориты не обязаны находиться в с-люксе все время, но они должны иметь существенное влияние и выполнять правила. Часто те, кто испытал на себе практику преодоления личных барьеров, смогли управлять собственными фирмами, чтобы быть наиболее инклюзивными, но им нужно больше сторонников и партнеров в этой работе.

Меган Хансен, заместитель главы People in MOD Pizza, поделилась: "участвуя в этих акциях, я выяснила, что абсолютно все мы имеем схожие устремления - и сталкиваемся с похожими трудностями, если речь идет о формировании важных видов деятельности с целью компаний резидентов, с которыми обычно боролись. Возлюбленная объяснила, что это критический период для ее деятельности. "Это вдохновляющая группа, и, сосредоточившись вместе, я считаю, что могу помочь товарищу функционировать вместе, чтобы создать долгосрочные и позитивные изменения в наших фирмах и за рубежом."

Это время включает в себя back-end (хранение данных, управление пользованием, работа с сервером, интеграция данных, push-уведомления и выпуск новых версий) и front-end задачи (кэширование, синхронизация, создание макетов, UI-дизайн, UI-разработка и вытекающие коррективы) (рис. 1.13).



Рисунок 1.13 – Сколько времени необходимо на разработку мобильных приложений на iOS или Android

В исследовании Mobile нет поэтому неожиданного, если у создателей есть практическая деятельность с 3-4 различными надстройками. Однако, поскольку эта кодировка (для (машина и условия продолжают улучшаться, есть исключения. Более того, если помимо бэкэнд-логики задействована внешняя Ассоциация (значительная часть логики остается в бэкэнде, а абонентская область в основном, как и узкая потребительская область, по этой причине эти существенные исключения не считаются простыми в сторону покупателя). Ранг в каждом случае формируется, и большое количество инновационных мыслей и условий имеют все шансы сделать работу творцов наиболее сложной.

Вы должны выбрать эластичную форму, если есть самая долгосрочная политика, вам нужно начать с быстрых положений, быстро развивающихся и далее расширяющихся, чтобы дополнить функции Ierevenu, реализуемые путем покупки продуктов каждую неделю / 2 недели и возникающие с целью процветания.

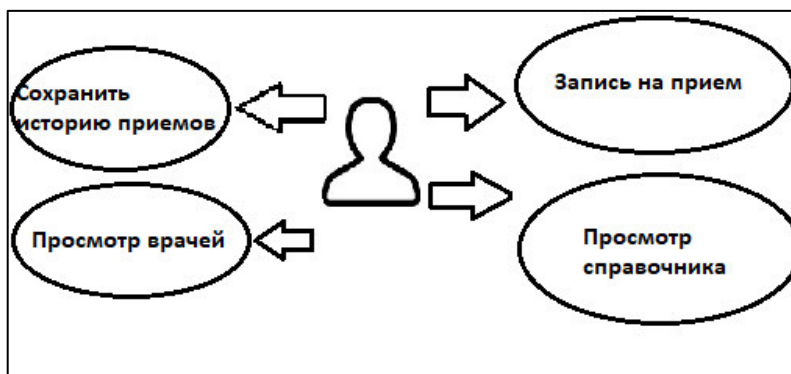


Рисунок 2.1 – Use-case диаграмма доступных действий пользователя

Метод-понятие последовательных действий (в соответствии с определенными инструкциями) для решения конкретной задачи. При этом метод заключается в порядке формирования страниц (деятельности) дополнения, установлении их значимости (основной, зависимой/последующей), формировании пластических и звуковых переходов от одной деятельности к другой.

Этот период рассматривается в основном, таким образом, а также кто-то является основой дома предстоящих положений, таким образом, в своем исследовании принимает помощь любого эксперта руководства, глядя на продукты под разными углами, разными глазами, что создает его более эластичным и ясным.

Качества превосходного метода. Необходимо четко определить вводимые и выводимые ресурсы. Любой этап метода должен быть точным и конкретным. Метод обязан быть более эффективным из числа большого количества методов разрешения трудности.

Акцент в воздействии пользователя создает ключевую функцию и дает четкие дорожные участки, с целью пользователя.

Движение гарантирует ценность. Движение чтит и увеличивает потребителя в свойстве первоначального мотора. Основное воздействие пользователя-это пятна, которые стимулируют движение, Преобразуя полный дизайн.

Все воздействия производятся в одной области. Предметы, представленные пользователю при отсутствии патологии непрерывности навыка, в том числе, если они изменены и реорганизованы.

Алгоритм не обязательно должен иметь компьютерный шифр. Вместо этого этот метод должен быть прописан аналогичным образом, чтобы он мог применяться в таких стилях программирования. В статье показана модель проекта построения присутствующего метода проектирования проекта формирования программных положений. 2.2.

## **2.2 Хранение данных**

База данных состоит из двух таблиц:

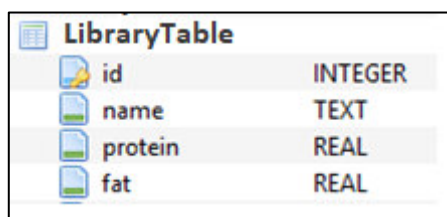
- library\_table;
- history\_table;

Таблицы не связаны между собой, они служат лишь для хранения данных.

Таблица library\_table содержит справочник по болезням, с которыми работает приложение. Ниже (на рис.2.2) представлены столбцы таблицы

library\_table:

- id – уникальный идентификатор;
- name – название болезни;
- protein – среднее количество заболеваний в году;
- fat – среднее количество смертей;



LibraryTable	
id	INTEGER
name	TEXT
protein	REAL
fat	REAL

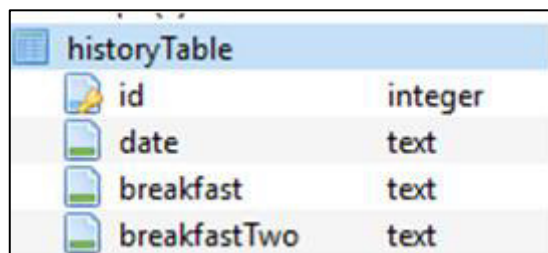
Рисунок 2.3– Таблица libraryTable

Таблица history\_table хранит прошедшие приемы у врача.

Ниже представлены столбцы таблицы history\_table:

- id – уникальный идентификатор;
- date – дата посещения;
- breakfast – названные симптомы болезни;
- breakfastTwo – названный курс лечения.

На рисунке 2.4 представлена данная таблица в графическом виде, с указанием типов данных которые хранятся в каждом столбце.



historyTable	
id	integer
date	text
breakfast	text
breakfastTwo	text

Рисунок 2.4 – Таблица historyTable

Такие данные как имя пользователя, история болезней, история посещений, данные болезней, фио врачей хранятся в специальном файле типа Preferences. Данный тип файлов служит для хранения системных настроек, аналогично INI файлам в Windows. Значения в файлах данного типа хранятся в виде пары:

- имя;
- значение.

Работать с таким типом данных, намного проще, чем с базой данных, и

он удобен для хранения небольшого количества данных.

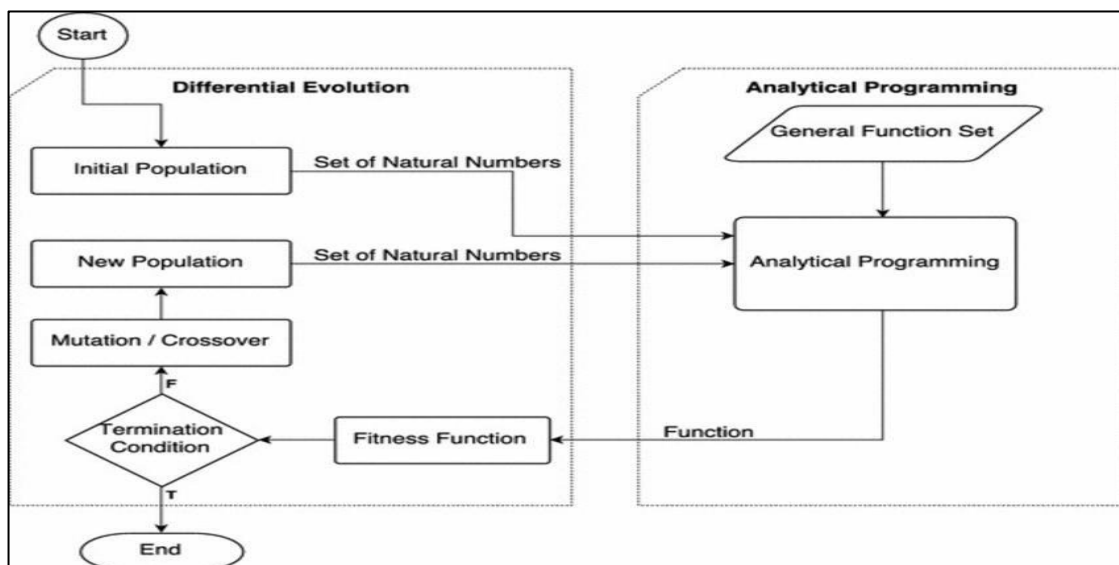


Рисунок 2.2 – Алгоритм создания программного продукта

### 2.3 Хранение данных

База данных состоит из двух таблиц:

- library\_table;
- history\_table;

Таблицы не связаны между собой, они служат лишь для хранения данных.

Таблица library\_table содержит справочник по болезням, с которыми работает приложение. Ниже (на рис.2.2) представлены столбцы таблицы library\_table:

- id – уникальный идентификатор;
- name – название болезни;
- protein – среднее количество заболеваний в году;
- fat – среднее количество смертей;

### 2.4 Характерные черты дизайна мобильных Android приложений

Цели. Сформируйте визуальную речь, которая синтезирует традиционные основы большого дизайна с инновациями и возможностями технологии и урока. Создать общую базовую концепцию, дающую возможность применять общую практику для различных платформ и объемов инструмента. Мобильные наборы основные, но детектор, речь, мышь и панели клавиатуры все без исключения методы ввода 1-го класса.

Материал этого сравнения. Реальное сравнение-это связующее понятие рационализированного места и понятие движения. Используемый материал основан на тактильной реальности, наполняет энтузиазмом изучение документа и клеветы, но технологически современен и не закрыт с целью воображения и магии.

Поверхности и стороны используемого материала гарантируют визуальные сигналы которые основаны в реальности. Использование тактильных атрибутов друзей может помочь пользователям быстро реализовать свое достоинство. Но эластичность используемого материала формирует новейшие способности, которые превосходят людей в физиологическом обществе, не нарушая законов физики.

Основы освещения, поверхностей и движения считаются ключевыми для передачи этого, и объекты движутся, взаимодействуют и существуют на месте и относительно друг друга. Реальный свет показывает швы, разделяет место и говорит о движущихся лепестках.

Смелый, графический, сознательный. Основными компонентами полиграфического дизайна, мини-типографии, сетки, места, размера, тона и использования отображений являются ориентированная визуальная обработка. Эти компоненты гораздо больше, чем приносят радость глазу. Они образуют иерархию, роль и тайну. Сознательный выбор цветов, с рукой до стороны, большой офис мини-печатан



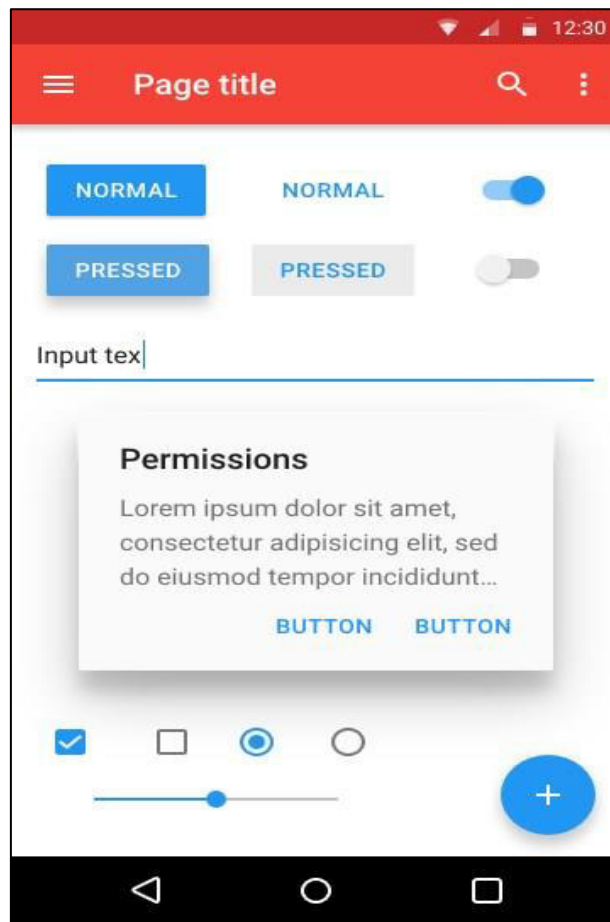


Рисунок 2.4 – Пример material дизайна для Android

Кроме того, самой яркой отличительной чертой Android созданных приложений является его дизайн, material design, который имеет свои характерные формы кнопок, порядок отображения информации и анимацию при выполнении каждой функции.

## 2.4 Методы проектирования

Проектирование программного продукта состоит из определенной последовательности этапов:

- предварительное проектирование;
- эскизное (рабочее или техно-рабочее) проектирование;
- изготовление, испытание и доводка опытного образца системы.

Каждый этап проводится с детализированным подходом.

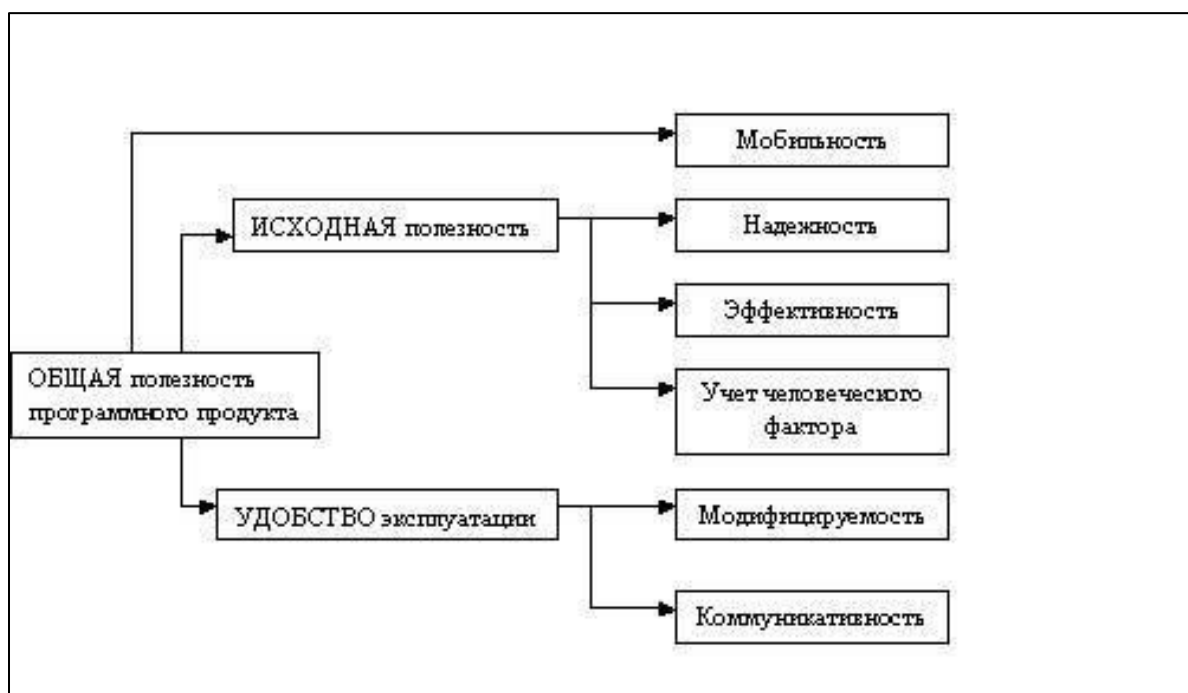
При предварительном проектировании программного продукта необходимо учесть такие факторы, как:

- разрешение экрана;
- универсальность дизайна для разных версий ПО;
- уникальность дизайна Android-приложений (дизайн кнопок, анимация);
- использование сервера для БД;
- оптимизация обновления исходной БД компании.

В практике существует два основных метода проектирования:

- автоматизированный (CASE - технология и ее элементы);
- неавтоматизированный (традиционный).

Рис. 2.5 наглядно показывает основные критерии при подходе к выбору методов проектирования, основываясь на которые можно оценить прделываемую работу.



## Рисунок 2.5 – Критерии создания

При наличии формирования собственных положений я следовал методам автоматизированного проектирования, так как это дает возможность снизить трудозатраты при реализации, непосредственно стимулировать процедуру, посредством систематизации исполняемых вопросов и построения актов по выполнению миссии, решению вопросов.

В дополнение к программе, Финансовая сторона плана должна быть исследована путем выплаты вероятной окупаемости положений программы. В период трудового проектирования следует выбирать сферу исследований.

Мой выбор определяется наличием широкого спектра многофункциональных способностей, в том числе автоматизированной разработки необходимых библиотек, плагинов и т. д. (Android Studio 2.3.2)

Android Studio-это сервисная область исследований Android. Кто-то разработан с целью Android, чтобы стимулировать исследования и помочь вам сформировать наиболее значимые свойства дополнений к цели любого устройства Android.

На основе IntelliJ IDEA Android Studio гарантирует самую быструю революцию в вашем кодировании и трудовом прогрессе.

Мгновенный запуск. Роль мгновенного запуска Android Studio позволяет изменять код и ресурсы в надстройке. Кто-то логически осведомлен об изменениях и часто приносит их в отсутствие перезапуска дополнения или возобновления APK, по этой причине вы сможете мгновенно заметить результаты.

Интеллектуальный программный код techred. Разорванный код может помочь вам исправить, записать код, быстро функционировать и быть наиболее эффективным в конце программного кода, рефакторинга и исследования кода. Наличие ввода Android Studio дает рецепты в раскрывающемся списке. Просто нажмите Tab, чтобы ввести код.

Подгонка:

- исправлена документа, фиксируя, как Genymotion вариантов, исключая пуск, как Genymotion в дистрибутивах Linux;
- Исправлена ошибка в показе неправильных функций устройства, присутствие отправлено в облако Genymotion;
- удалены ненужные библиотеки из пакета Linux (например, libdrm или libxcb);
- Исправлена ошибка, с - из-за чего кратковременный комп.данные были сохранены после отмены акции в "самом умном облаке»;
- удалено неуместное предупреждение о несоответствии

минорные версии в genyshell; - Исправлена ошибка, с-из-за чего разработка кэша одиночных устройств

Облако, как Genymotion не обновлялся;

- удалено неверное уведомление о дате истечения срока действия;

- добавлена профилактика ошибок, если gmtool обнаруживает спорные особенности;

- исправлено заключение сопоставления ImageReader.

Фрагментация устройств в нашем обществе часто называют на самом деле, из-за того, что это может быть трудно реализовать, а также ваши дополнения будут реализованы в огромном количестве автономных платформ.

GenyMotion-это эмулятор, который способен имитировать свойства различных Android-устройств. Наше руководство нашло это очень эффективным в обеспечении быстрого противоположного отношения к цели наших Android дополнений.



## 3 Описание работы приложения

### 3.1 Функционал приложения

Вспоминая основное назначение разрабатываемого приложения, а именно возможность найти медицинское учреждение, посмотреть какие врачи работают в клинике и какие услуги она оказывает. Помимо этого, алматинцы могут прямо через приложение задать свой вопрос в интересующую клинику, оставить отзыв или записаться на прием в медицинские центры, подключенные к работе с онлайн-записью.

Именно поэтому в приложении имеется следующий функционал:

- запись к врачу;
- справочник заболеваний;
- история посещений;
- отображение информации о пользователе, а именно прогресс в виде графика, история питания.

Приложение работает на основе выбора пользователя. Если он выбирает запись на прием, то ему предоставляется возможность выбрать дату посещения и ожидать звонка. Если же будет выбран справочник, то пользователь может в поиске найти интересующее заболевание и прочитать его симптомы и курс лечения.

Так же, человек, столкнувшийся с болезнью, попадает в новый мир. Именно так чувствует себя человек, который вчера был здоров и вел привычную жизнь, а теперь должен сдавать анализы, принимать лекарства и привыкать к новому образу жизни. Приложение может помочь больному войти в режим выздоровления с наименьшими затратами.

Существуют медицинские центры и клиники, специализирующиеся на конкретных заболеваниях. Это могут быть болезни сердца, онкология, кожные заболевания, гинекология, стоматология, урология и прочие. В приложении каждой категории можно продумать свой индивидуальный сценарий. Так же, человек, столкнувшийся с болезнью, попадает в новый мир.

Именно так чувствует себя человек, который вчера был здоров и вел привычную жизнь, а теперь должен сдавать анализы, принимать лекарства и привыкать к новому образу жизни.

После установки приложения и при первом запуске, сразу же открывается окно регистрации. Данное окно регистрации служит для ввода необходимых данных для работы приложения, данное окно представлено на рисунке 3.1.

Это могут быть болезни сердца, онкология, кожные заболевания, гинекология, стоматология, урология и прочие. В приложении каждой категории можно продумать свой индивидуальный сценарий. Так же, человек, столкнувшийся с болезнью, попадает в новый мир.

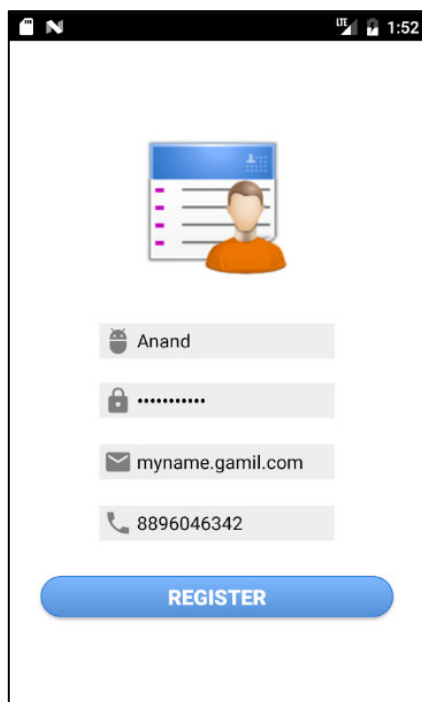
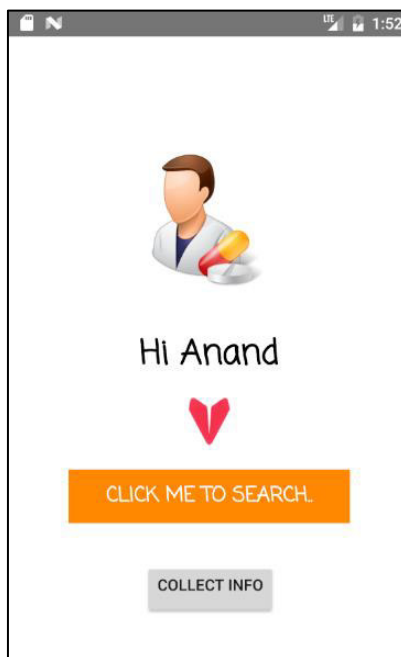


Рисунок 3.1 – Окно регистрации

После заполнения всех полей, данные записываются в специальный файл, описанный в предыдущей главе. Далее у нас открывается главное окно, где на основе введённых данных при регистрации.

После регистрации нас перебрасывает в экран приветствия, где мы переходим в следующее окно (рис. 3.2)



### Рисунок 3.3 – Окно приветствия

В окне приветствия нам доступны следующие действия:

- поиск;
- сбор информации.

При нажатии кнопки поиска, происходит переход в следующее окно где можно воспользовавшись поиском и вбив слово, можно не дожидаясь выбрать из предложенных вариантов нужное. Показано на рисунке 3.4.

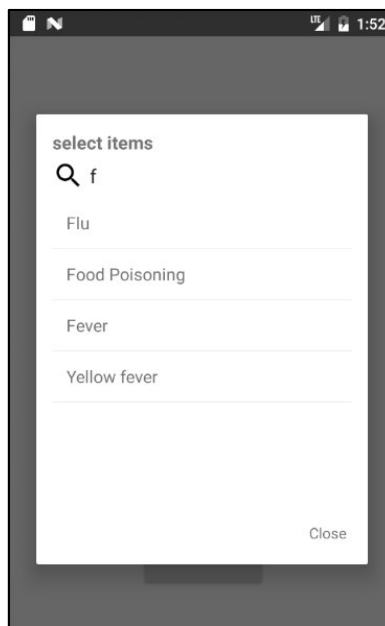


Рисунок 3.4 – Окно поиска

Выбрав нужное заболевание, можно узнать симптомы болезни, кликнув на “symptoms”. Показано на рис 3.5.



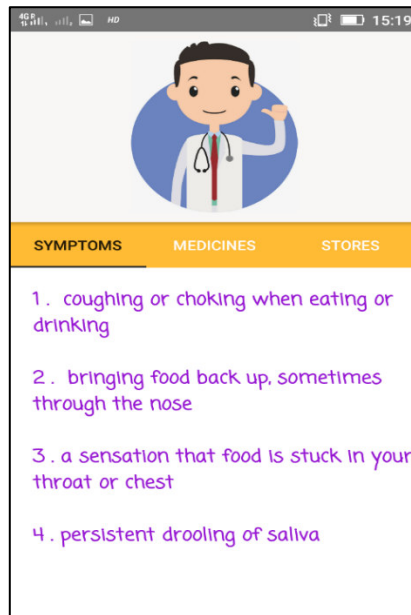


Рисунок 3.5 – Окно симптомов заболевания

Выбрав нужное заболевание, можно узнать методы лечения, кликнув на “medicines”. Показано на рис 3.6.

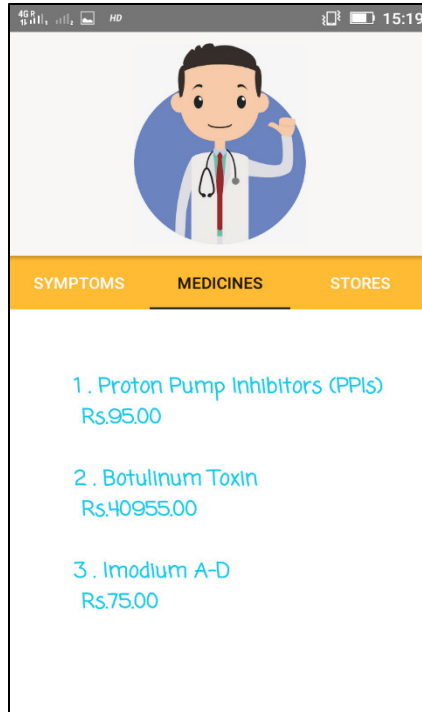


Рисунок 3.6 – Окно методов лечения

Выбрав нужное заболевание, можно узнать историю заболевания, кликнув на “Stories”. Показано на рис 3.7

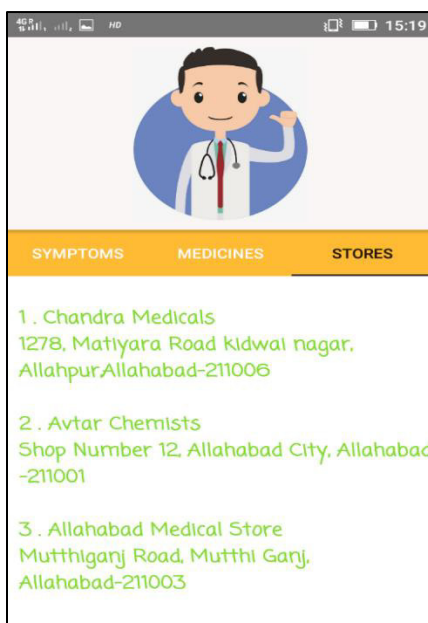


Рисунок 3.7 – Окно методов лечения

### 3.2 установка приложений

Приложение находится на стадии бета-тестирования. Установка производится путем загрузки файла APK, который должен использоваться в памяти устройства и с помощью файлового менеджера, чтобы раскрыть этот документ с целью строительства.

В конце этапа бета-тестирования будут обнаружены ошибки дополнения, предоставленные запросами пользователей. В рамках расследования будет выпущена 1-я стабильная модификация дополнения, которая будет загружена в Mall add-ons Play Store, из мест которой в будущем можно будет определить план.

### 3.3 сопоставимость приложения

Приложение содержит самую маленькую версию Android 4.0.4 и самую большую версию Android 9

Приложение хорошо оптимизировано для поддерживаемых версий, несмотря на использование компонентов Material design, которые возникли с выпуском Android 5.

### 3.4 Группа Newgridadapter

Механизм GridView будет использоваться для отражения иллюстраций на экране. Но этот компонент не вычисляется для непосредственного сохранения компонентов. Хранить и транслировать информацию я буду с поддержкой интегрированного адаптера (BaseAdapter) в поле зрения, образуя базу для целей GridView.

Функционал базового адаптера будет небольшим, поэтому нужно сделать другой. Составьте группу NewGridAdapter, которая будет унаследовала со встроенным BaseAdapter. Методы, реализованные в классе NewGridAdapter:

void MakePictArray () – поместить изображения в градиент и иметь их в случайном режиме.

View getViewByPos(int positionNumber, View convertView, ViewGroup parent) - дает картинку с номером ячейки position\_number, наличие которого с учетом его текущей позиции (CELL\_OPENED, CELL\_CLOSED, CELL\_DELETED)

void checkCellsOpened() – контроль в присутствии непокрытых яиц. Если 2 цифры раскрыта, в этом случае мы проводим их сравнение и в случае совпадения мы устраняем и, если они не похожи в любом случае.

void cellOpen (int positionNumber) – устанавливает отображение с позицией CELL\_OPENED int getElementNum () – дает количество отображений в GridView.

Object getObj (int positionNumber) – возвращает объект с номером positionNumber long.

getElementId (int positionNumber) - вернет личный номер товара с номером positionNumber boolean checkGameOver () - контролирует требование прекращения вида развлечений (без закрытых иллюстраций)

### 3.5 Группа MemoryActivity

В классе I MemoryActivity сформировал предмет GridView и связал его с адаптером, реализованным в классе GridAdapter.

Кроме того, я формирую расходомер и мгновенно не назначаю ему никакой роли, а также позволяю автоспуску.

Далее составляем обработчик действий setOnItemClickListener(), который будет провоцировать наличие клика в компоненте. Присутствие его апелляции должно быть соблюдено:

- контроль наличия открытых Яков. Если 2 цифры раскрыта, в этом случае мы проводим их сравнение и в случае совпадения мы устраняем и, если они не похожи в любом случае.

- показывает выбранное изображение

- в случае, если больше нет открытых карт, в этом случае веселье заканчивается с целью этого я использую методы checkCellsOpened (), cellOpen() и checkGameOver (), реализованные в GridAdapter. Также

необходимо подчеркнуть тот факт, что приращение счетчика перемещений сделало наличие какой-либо изобретенной ячейки. Метод `ShowGameOver ()` используется для уведомления о завершении развлечения. В `mute` есть интерактивное окно, в котором составляется уведомление об окончании вида развлечения, количестве ходов и количестве периода, проведенного в разработке. Кроме того, в интерактивном окне есть клавиша `end`, наличие нажатия на которую блокируется текущее действие.

### **3.6 Группа `MemoryStart`**

В этом классе я работаю с компонентами, которые видны в основном списке присутствия, начиная вид развлечения. Сначала я сформировал 2 куска формы Кнопкой, затем написал для любого из них собственный обработчик. Наличие нажатия клавиши "старт" возбуждает кстати `StartGame()`, в котором активность будет носиться с удовольствием. При наличии нажатия на кнопку "Выход" я активирую метод `finish ()`, который перекрывает текущую активность.

## 4 Технико-экономическое обоснование

Целью данной дипломной работы является создание мобильного приложения для медицинского центра, для более эффективного привлечения новых клиентов, улучшения работы медицинского центра, повышения информированности существующих клиентов о работе медицинского центра, а также закрепление практических навыков программирования для ОС Android в среде разработки Android Studio.

Разработкой приложения занимаются двое исполнителей: руководитель проекта и инженер-программист. В обязанности руководителя входит соблюдение и разработка рабочих графиков, их контроль и оптимизация. В обязанности инженер-программиста входит нахождение алгоритмов решения экономических и других задач, обеспечение возможности выполнения алгоритма и соответственно поставленной задачи средствами вычислительной техники, проводит тестирование и отладку.

Технико-экономическое обоснование содержит следующие пункты:

- определение сложности разработки приложения;
- расчет затрат на разработку приложения;
- определение ценности готового продукта;

### 4.1 Определение сложности разработки мобильного приложения

Количество этапов в разработке приложения зависит от объема проекта и существующих вводных данных на него. Определения того насколько проект будет сложным и трудоемким нужно разделить создание приложения на этапы. Для наглядности существует специальная форма разделения работ по этапам которая приведена ниже в таблице 4.1.

Таблица 4.1 – Этапы разработки приложения

Этапы разработки ПО	Вид работы	Трудоемкость, чел. час.
Этап 1	Планирование и оценка	5
Этап 2	Создание плана проекта	10
Этап 3	Проектирование интерфейса	20
Этап 4	Создание концепции дизайна	20
Этап 5	Разработка серверной	80
Этап 7	Написание кода	65
Этап 8	Тестирование	10
Этап 9	Устранение неполадок	25
Этап 10	Второе тестирование	15
Этап 11	Создание пререлизной версии	30
Итого: трудоемкость выполнения дип. проекта		280

Рассчитать трудоемкость выполнения дипломного проекта в днях:

$$280:8 = 35 \text{ (дней)}$$

Продолжительность рабочего дня равна 8 часам. В результате для реализации программного обеспечения необходимо 35 рабочих дней.

#### 4.2 Расчет затрат на разработку мобильного приложения

Определение затрат необходимых для разработки программного обеспечения производится на основе имеющейся сметы, которая включает следующие элементы:

- материальные затраты;
- затраты на оплату труда;
- социальный налог;
- амортизация основных фондов;
- прочие затраты.

Материальные затраты делятся на основные и вспомогательные затраты на материалы, энергию и другие затраты необходимые для разработки мобильного приложения. Расчет материальных затрат происходит по форме, предоставленной в таблице 4.2.

Таблица 4.2 – Затраты на материальные ресурсы

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Бумага А4	Снегурочка	Упаковка	1	2500	2500
Карандаши для ТЗ	Koh-i-Noor	Штук	5	185	925
Шариковые ручки	СТАММ	Штук	2	69	138
Фломастеры	Magic pens	Штук	1	1900	1900
Картридж	Ricoh Aficio	Штук	1	8000	8000
Итого:					13463

При разработке данного мобильного приложения я использовал ноутбук Apple MacBook, который полностью оснащен всеми необходимыми программными обеспечениями, который оснащен MacOS X и необходимым пакетом драйверов.

Общую сумму, необходимую на материальные средства ( $Z_M$ ) можно рассчитать по следующей формуле:

$$Z_M = \sum P_i * C_i, \quad (4.1)$$

Где:

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

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

$i$  - вид материального ресурса;

$n$  - количество видов материальных ресурсов.

Расчет затрат на необходимое оборудование и программное обеспечение производится по форме, приведенной в таблице 4.3.

Таблица 4.3 – Расчет затрат на оборудование и ПО, необходимое для проекта

Наименование	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
--------------	-------	---------------	------------	---------------------	---------------

материала					
Ноутбук	Apple MacBook	Штук	1	538500	538500
Роутер	ZTE MF910 4G LTE	Штук	1	12000	12000
Хостинг	PS.KZ	Штук	1	4900	4900
Домен	PS.KZ	Штук	1	2500	2500
Выделенный сервер	2xIntel Xeon e5-2620 2.0 ГГц	Штук	1	82000	82000
Принтер	Epson L805	Штук	1	92000	92000
Итого:					731 900

$$З_m = 13463 + 731\,900 = 745363 \text{ (тг)}$$

Подводя итог, создание мобильного приложения обойдется в 745363(тг).

### 4.3 Расчет затрат на электроэнергию

Разработка проекта требовало потребление электроэнергии для всех устройств. Разработка потребовало 280 часов, теперь необходимо рассчитать стоимость электроэнергии в указанное время. Помимо этого, нужно будет рассчитать 280 часов в днях.

Для просчета затраченной электроэнергии, воспользуемся формулой и заполненной таблицей 4.4.



Таблица 4.4 – Затраты на электроэнергию

Наименование приборов	Паспортная мощность, кВт	Коэффициент мощности	Время работы оборудования, ч	Цена ЭЭ тг/кВтч	Сумма, тг.
Ноутбук	0,7	0,6	280	18,32	2154,43
Принтер	0,5	0,9	80	18,32	659,52
Роутер	0,05	0,9	280	18,32	115,41
Мобильный телефон	0,2	0,5	80	18,32	146,56
Освещение	0,3	0,7	280	18,32	1077,21
Итого:					4153,13

$$Z_{\text{эл. эн. обор.}} = 4153,13 \text{ (тенге)}$$

На дополнительные потребности расходы подсчитываются на основе повышенного показателя в объеме 5% от расходов на электроэнергию (4.4).

$$Z_{\text{доп. нужды}} = 5\% * Z_{\text{эл. эн. обор.}} \quad (4.4)$$

Определим затраты на дополнительные потребности согласно формуле (4.5):

$$Z_{\text{доп. нужды}} = 0.05 * 4153,13 = 207,65 \text{ (тенге)} \quad (4.5)$$

Исходя из всех расчетов, полные расходы на электроэнергию составляют (4.6):

$$\Theta = 207,65 + 4153,13 = 4360,78 \text{ (тенге)} \quad (4.6)$$

## 4.4 Расчет затрат на оплату труда

Для создания мобильного приложения были приглашены следующие специалисты:

- руководитель проекта, занимающийся планировкой времени, корректировкой рабочих моментов, покупка необходимых;
- инженер-разработчик, занимающийся разработкой приложения, тестированием, и являющимся технической поддержкой.

Сумму расходов на оплату труда можно рассчитать по следующей формуле (4.7):

$$Z_{\text{тр}} = \sum ЧС_i * T_i \quad (4.7)$$

Где:

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

$T_i$  - трудоемкость разработки модели, чел.×ч;  $i$  - категория работника;

$n$  - количество работников, занятых разработкой ПП.

Во время реализации проекта рабочее время участников не равномерно, поэтому имеет смысл установить часовую ставку каждого работника и общий объем заработной платы.

Часовую ставку сотрудника можно рассчитать по следующей формуле(4.8):

$$ЧС_i = \frac{ЗП_i}{ФРВ_i} \quad (4.8)$$

Где:

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

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

Месячная заработная плата руководителя - 200 000 тенге

Месячная заработная плата инженер-программиста – 280 000.

Рассчитаем часовую ставку каждого работника согласно формуле (4.6):

$$ЧС_{\text{руководитель}} = \frac{200\,000}{35 * 8} = 714,28 \frac{\text{тг}}{\text{ч}}$$

$$\text{ЧС}_{\text{программист}} = \frac{280\,000}{35 * 8} = 1000 \frac{\text{тг}}{\text{ч}}$$

Рассчитав з/п сотрудников выходит:

- часовая ставка руководителя = 714,28 (тг/ч), трудоемкость его деятельности равняется 140 часам.

- часовая ставка инженер-программиста = 1000 (тг/ч), трудоемкость дизайнера равняется 280 часам.

$$З_{\text{тр}} = 714,28 * 140 + 1000 * 280 = 99999,20 + 280000 = 379999,20 (\text{тг})$$

Расчеты затрат по оплате труда показаны в таблице (4.5).

Таблица 4.5. – Расчет заработной платы

Категория работника	Квалификация	Трудоемкость разработки ПП, час.	Часовая ставка, тг/ч	Сумма, тг.
Руководитель проекта	Проектный руководитель	140	714,28	99999,20
Программист	Программист	280	10000	280000
Итого:				379999,20

#### 4.5 Расчет затрат по социальному налогу

Согласно Налоговому кодексу Республики Казахстан социальный налог составляет 9,5% от фонда оплаты труда. Социальный налог можно рассчитать по следующей формуле (4.9):

$$С_{\text{н}} = (\text{ФОТ} - \text{ПО}) * 0,095 \quad (4.9)$$

где ПО - отчисления в пенсионный фонд, Он составляет 10% от ФОТ.

$$\text{ПО} = 379999,2 * 0,1 = 37999,92 \text{ тенге}$$

$$C_H = (379999,2 - 37999,92) * 0,095 = 32489,93 \text{ тенге}$$

Результаты расчетов представлены в таблице (4.6):

Таблица 4.6 – Начисление социального налога

Категория работника	Количество человек	Заработная плата, тг	Пенсионные отчисления, тг	Социальный налог, тг
Руководитель	1	99999,2	9999,92	8549,93
Разработчик	1	280000	28000	23940
Итого:				32489,93

#### 4.6 Амортизация основных фондов и прочие затраты

Нормы амортизации ОФ необходимо определить в соответствии с налоговым кодексом РК. Амортизацию ОФ можно определить по следующей формуле (4.10):

$$A_r = \frac{C_{об} * H_a}{100} \quad (4.10)$$

Где:

$C_{об}$  – стоимость оборудования;

$H_a$  – норма амортизации (норма амортизация = 25);

Формула (4.8) позволяет рассчитать нужную сумму для амортизационных отчислений за год для ноутбука:

$$A_r = \frac{538500 * 25}{100} = 134625 \text{ тенге}$$

Теперь необходимо рассчитать норму амортизации за период разработки:

$$A_r = \frac{150\,000 * 35}{365} = 12909,20 \text{ тенге}$$

Подобным образом необходимо рассчитать норму амортизации для всего оборудования. Результаты расчетов приведены в таблице (4.7).

Таблица 4.7 – Амортизация ОФ

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Персональный компьютер	538500	25	134625	12909,20
Роутер	12000	25	3000	1150,70
Хостинг	4900	20	1225	117,50
Домен	2500	20	625	60
Выделенный сервер	82000	15	20500	1965,70
Принтер	92000	20	23000	2205,50
Итого:			182975	18408,60

Смета расходов на разработку ПО.

На основе всех представленных расчетов необходимо оформить смету расходов на разработку ПО согласно форме, которая приведена в таблице (4.8). На рисунке () продемонстрирована диаграмма рабочих расходов.

Таблица 4.8 – Смета затрат на разработку ПО

Статьи затрат	Сумма, тг	Процент, %
Материальные затраты	745363	61
Затраты на оплату труда	379999,20	31
Социальные налоги	32489,93	3
Затраты на электроэнергию	4360,78	0
Амортизация основных фондов	18408,60	1
Прочие расходы	50000	4
Итого по смете:	1230617,93	100

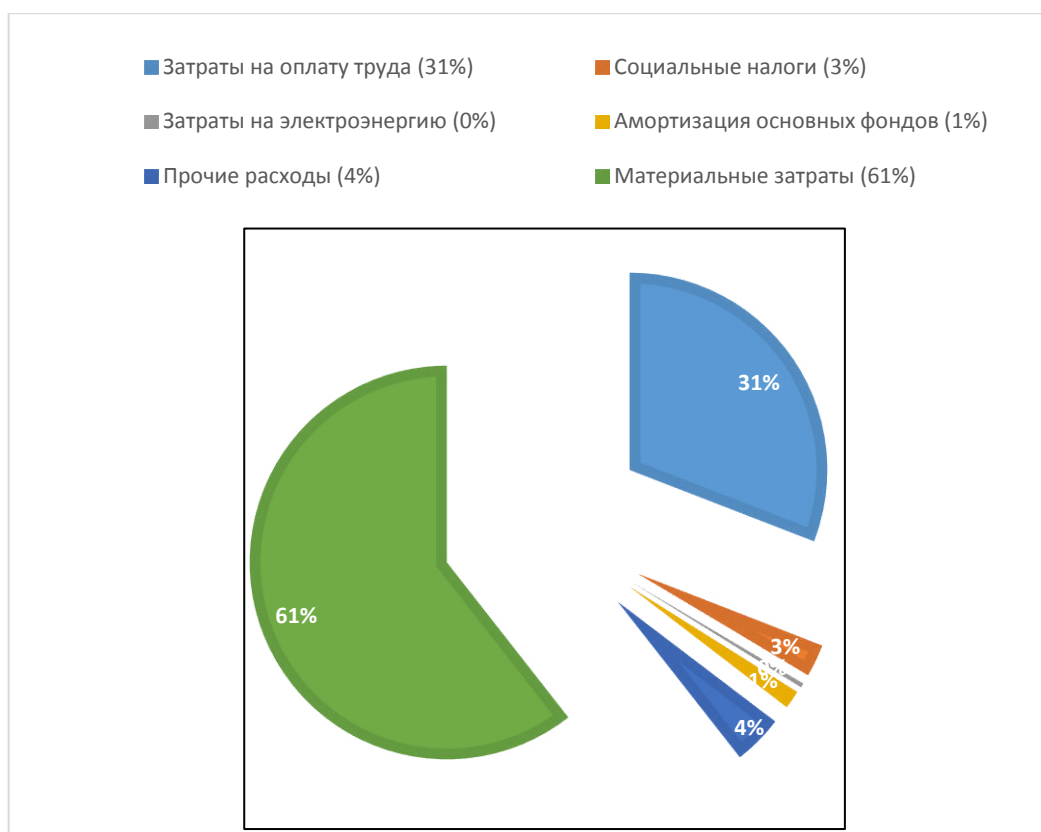


Рисунок 4.1 – Диаграмма затрат

#### 4.7 Определение возможной (договорной) цены ПО

Стоимость мобильного приложения определяется на основе качества разработанного продукта, сроков его разработки и производительности продукта. Стоимость  $C_d$  для приложения можно рассчитать по следующей формуле (4.11):

$$C_d = Z_{\text{нир}} * \left(1 + \frac{P}{100}\right), \quad (4.11)$$

Где:

$Z_{\text{нир}}$  – затраты на разработку программного обеспечения, тг;

$P$  – Средний уровень рентабельности ПО, (%). Данный параметр принят равным 25%.

Рассчитаем прибыль рентабельности (4.12)

$$P_{\pi}=1\,230\,617,93*\left(\frac{25}{100}\right)=1\,230\,617,93*0,25=307654,48 \text{ тенге} \quad (4.12)$$

Где:

$P_{\pi}$  – прибыль

$$Ц_{д}=1\,230\,617,93*\left(1+\frac{25}{100}\right)=1\,230\,617,93+307654,48=1\,538\,272,41 \text{ тенге} \quad (4.13)$$

Далее необходимо определить стоимость реализации с учетом НДС, ставка НДС устанавливается законодательством РК. На 2019 года ставка НДС составляет 12%. Стоимость реализации учитывая НДС можно рассчитать по следующей формуле (4.14):

$$Ц_{р}=Ц_{д}+Ц_{д}*НДС, \quad (4.14)$$

$$Ц_{р}=1\,538\,272,41+1\,538\,272,41*0,12=1\,722\,865,09 \text{ тенге}$$

Данную цену можно округлить до 1 723 000, 00 тенге.

## 4.8 Вывод

В этой части дипломного проекта был произведен расчет всех затрат для создания мобильного приложения с учетом налогов и прочих затрат.

Для покупателя основным показателем будет считаться оптимальная цена программного продукта и его производительность. Так же в последней главе был произведен расчет договорной цены программного продукта, который равняется 1723 000,00 тенге, данное значение является рациональным с точки зрения экономической эффективности. Себестоимость мобильного приложения равна 1 538 272,41. Прибыль равна 307654,48 тенге.



## 5. Охрана труда и безопасность жизнедеятельности

В данном разделе рассматриваются мероприятия по обеспечению безопасности жизнедеятельности и охране труда в рамках использования разрабатываемого программного продукта.

Разрабатываемый программный продукт предназначен для организации работы в различных сферах деятельности. При его помощи люди составляют себе рабочий план, согласно которому работают на протяжении всего дня, а также могут общаться со своими коллегами и обсуждать рабочие моменты.

Программный продукт реализуется в виде веб-приложения, поэтому необходимо рассмотреть характеристики условий труда в офисе, за персональным компьютером.

Для последующего за разработкой внедрения программного обеспечения предоставлены два помещения, служащие в которых занимаются разработкой и продажей программного обеспечения. Площадь помещений 24 м<sup>2</sup> и 42 м<sup>2</sup>. Большое помещение предназначено для разработчиков, малое – для отдела продаж. И те, и другие подавляющую часть своего рабочего дня проводят за персональным компьютером.

В помещениях отсутствует шум, который может потребовать дополнительной звукоизоляции и использования средств индивидуальной защиты от шума.

Микроклимат помещения также является комфортным для ежедневной работы, в том числе благодаря небольшой приточной системе и наличию кондиционеров, которые позволяют поддерживать температуру в помещении в пределах 23-25°С в теплое время года.

Эргономика рабочего места (удобные рабочие кресла, просторные рабочие поверхности, аксессуары, обеспечивающие доступность рабочих материалов на расстоянии вытянутой руки) создает приятную рабочую обстановку для продуктивной работы.

Единственным фактором, который препятствует идеальным условиям труда является освещение помещений. Поэтому предстоит проверить степень освещенности каждого из помещений естественным и искусственным светом, и предложить мероприятия по улучшению условий труда.

В малом помещении есть два окна общей площадью 2,8 м<sup>2</sup>, а в большом помещении три окна общей площадью 4,3 м<sup>2</sup>. Они обеспечивают естественное освещение помещений. Искусственное освещение обеспечивается 6 лампами в

малом помещении и 9 лампами в большом помещении. Используются лампы накаливания мощностью 40 Вт и световым потоком 420 Лм.

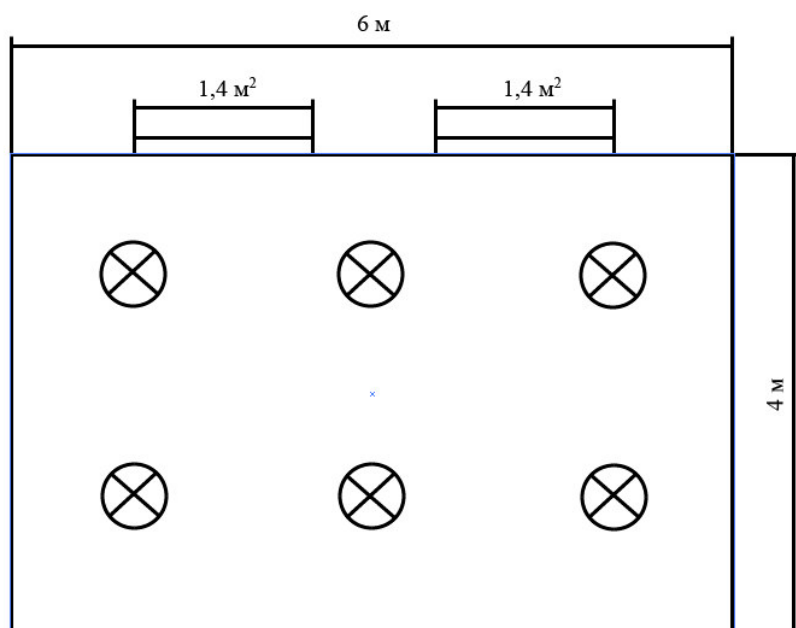


Рисунок 5.1 – Схема малого помещения

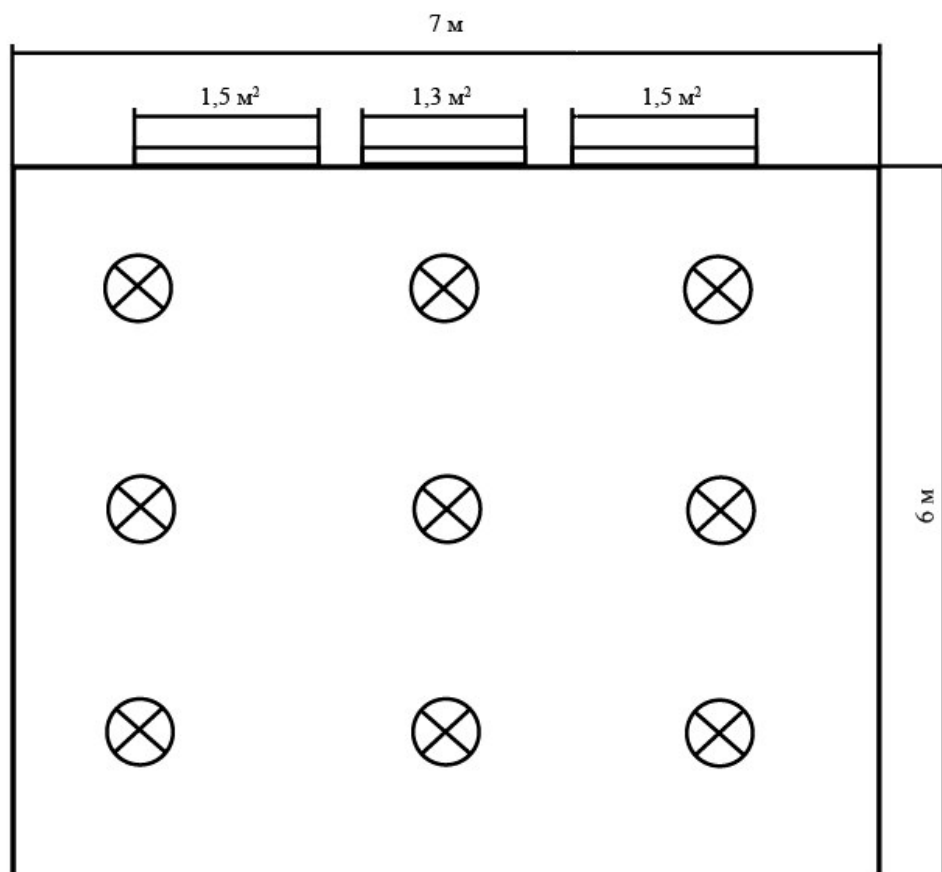


Рисунок 5.2 – Схема большого помещения

## 5.1 Расчет естественного освещения

Расчет естественного освещения включает в себя определение световых проемов в помещении. Световые проемы могут располагаться в стенах (боковое освещение) и в потолке (верхнее освещение). Так как предполагается офисная работа, и пользоваться разработанным программным продуктом будут в офисном помещении, то целесообразнее в качестве естественного освещения выбрать именно боковое освещение.

Необходимо рассчитать рекомендуемые размеры световых проемов для каждого помещения и сравним с фактическими размерами окон.

### 5.1.1 Расчет для малого помещения

Для расчета бокового освещения будет применена следующая формула:

$$100 \frac{S_0}{S_n} = \frac{e_n \times K_3 \times \eta_0}{\tau_0 \times r_1} \times K_{зд} \quad (5.1)$$

где:

$S_0$  - суммарная площадь световых проемов при боковом освещении, м<sup>2</sup>;

$S_n$  - площадь помещения, м<sup>2</sup>;

$e_n$  – нормируемое значение КЕО;

$K_3$  – коэффициент запаса;

$n_0$  – световая характеристика окон;

$t_0$  - общий коэффициент светопропускания;

$r_1$  - коэффициент, учитывающий повышение КЕО при боковом освещении, благодаря свету, отраженному от поверхности помещения и подстилающего слоя, примыкающего к зданию;

$K_{зд}$  - коэффициент, учитывающий затемнение окон противостоящими зданиями.

Общий коэффициент светопропускания рассчитывается следующим образом:

$$t_0 = t_1 t_2 t_3 t_4 t_5, \quad (5.2)$$

где:

$t_1$  – коэффициент светопропускания;

$t_2$  – коэффициент потери света в переплетах проема;

$t_3$  – коэффициент потери света в несущих конструкциях, равен 1 при боковом освещении;

$t_4$  – коэффициент потери света в солнцезащитных устройствах;

$t_5$  – коэффициент потери света в защитной сетке, под фонарями, равен 0,9.

Так как световым проемом будет окно с тройным стеклом, то

$$t_1=0,75.$$

Стальные двойные и открывающиеся оконные переплеты, по таблице означают, что:

$$t_2=0,6.$$

Коэффициент потери света в регулируемых жалюзи равен:

$$t_4=1.$$

На основании собранных данных общий коэффициент светопропускания равен(5.3):

$$t_0=0,75*0,6*1*1*0,9=0,405. \quad (5.3)$$

Коэффициент запаса для офисов с вертикальным светопропускаемым материалом, равен:

$$K_3=1,2.$$

Отношение длины помещения к глубине равно (5.4):

$$o_1 = \frac{6}{4} = 1,5 \quad (5.4)$$

Отношение глубины помещения, к разности высоты верхнего края окна и высоты рабочей поверхности (5.5):

$$o_2 = \frac{4}{2,5-1,2} = 3,1 \sim 3. \quad (5.5)$$

С помощью найденных значений находим значение световой характеристики окон при боковом освещении:

$$\eta_0 = 15.$$

Коэффициент  $r_1$ , с учетом определенных значений, равен:

$$r_1 = 4,3.$$

Ввиду отсутствия зданий напротив световых проемов, коэффициент затемнения близко стоящими зданиями принимается равным:

$$K_{зд} = 1.$$

Нормируемое значение КЕО рассчитывается по формуле (5.6) :

$$e_N = e_H * m_N \quad (5.6)$$

Где:

$N$  – номер группы обеспеченности естественным светом;

$e_H$  – значение КЕО;

$m_N$  – коэффициент светового климата.

Алматы относится к четвертой группе обеспеченности естественным светом,

$$N=4.$$

Световые проемы расположены на север, следовательно:

$$m_N = 0,75.$$

Значение КЕО, при высокой точности зрительной работы, которая соответствует работе разработчика:

$$e_H = 1,2$$

Теперь выполним расчет нормируемого значения КЕО (5.7):

$$e_N = 1,2 * 0,75 = 0,9 \quad (5.7)$$

Подставив полученные значения в пропорцию, предназначенную для поиска суммарной площади световых проёмов, получим (5.8):

$$100 \frac{S_0}{24} = \frac{0,9 * 1,2 * 15}{0,405 * 4,3} * 1 = 9,3 \quad (5.8)$$

Из полученной пропорции найдем суммарную площадь световых проёмов (5.9):

$$S_0 = \frac{9,3 \cdot 24}{100} = 2,2 \text{ (м}^2\text{)}.$$

(5.9)

Таким образом, получается, что в малом помещении площадь окон превышает рекомендуемую суммарную площадь световых проемов ( $8 \text{ м}^2 > 2,2 \text{ м}^2$ ) и обеспечивает достаточную освещенность в помещении.

### 5.1.2 Расчет для большого помещения

Для расчета бокового освещения будет применена следующая формула (5.10):

$$100 \frac{S_0}{S_n} = \frac{e_n \times K_3 \times \eta_0}{\tau_0 \times r_1} \times K_{зд} \quad (5.10)$$

где:

$S_0$  - суммарная площадь световых проемов при боковом освещении,  $\text{м}^2$ ;

$S_n$  - площадь помещения,  $\text{м}^2$ ;

$e_N$  - нормируемое значение КЕО;

$K_3$  - коэффициент запаса;

$n_0$  - световая характеристика окон;

$t_0$  - общий коэффициент светопропускания;

$r_1$  - коэффициент, учитывающий повышение КЕО при боковом освещении, благодаря свету, отраженному от поверхности помещения и подстилающего слоя, примыкающего к зданию;

$K_{зд}$  - коэффициент, учитывающий затемнение окон противостоящими зданиями.



Общий коэффициент светопропускания рассчитывается следующим образом (5.11):

$$t_0 = t_1 t_2 t_3 t_4 t_5 \quad (5.11)$$

Где:

$t_1$  – коэффициент светопропускания;

$t_2$  – коэффициент потери света в переплетах проема;

$t_3$  – коэффициент потери света в несущих конструкциях, равен 1 при боковом освещении;

$t_4$  – коэффициент потери света в солнцезащитных устройствах;

$t_5$  – коэффициент потери света в защитной сетке, под фонарями, равен 0,9.

Так как световым проемом будет окно с тройным стеклом, то

$$t_1 = 0,75.$$

Стальные двойные и открывающиеся оконные переплеты, по таблице означают, что:

$$t_2 = 0,6.$$

Коэффициент потери света в регулируемых жалюзи равен:

$$t_4 = 1.$$

На основании собранных данных общий коэффициент светопропускания равен (5.12):

$$t_0 = 0,75 * 0,6 * 1 * 1 * 0,9 = 0,405.$$

(5.12)

Коэффициент запаса для офисов с вертикальным светопропускаемым материалом, равен:

$$K_3=1,2.$$

Отношение длины помещения к глубине равно (5.13):

$$o_1 = \frac{7}{6} = 1,2 \sim 1. \quad (5.13)$$

Отношение глубины помещения, к разности высоты верхнего края окна и высоты рабочей поверхности (5.14):

$$o_2 = \frac{6}{2,5-1,2} = 4,61 \sim 5. \quad (5.14)$$

С помощью найденных значений находим значение световой характеристики окон при боковом освещении:

$$\eta_0=23.$$

Коэффициент  $r_1$ , с учетом определенных значений, равен:

$$r_1=10.$$

Ввиду отсутствия зданий напротив световых проемов, коэффициент затемнения близко стоящими зданиями принимается равным:

$$K_{зд} = 1.$$

Нормируемое значение КЕО рассчитывается по формуле (5.15):

$$e_N = e_H * m_N \quad (5.16)$$

Где:

$N$  – номер группы обеспеченности естественным светом;

$e_H$  – значение КЕО;

$m_N$  – коэффициент светового климата.

Он относится к четвертой группе обеспеченности естественным светом,

$$N = 4.$$

Световые проемы расположены на север, следовательно:

$$m_N = 0,75.$$

Значение КЕО, при высокой точности зрительной работы, которая соответствует работе разработчика:

$$e_H = 1,2.$$

Теперь выполним расчет нормируемого значения КЕО:

$$e_N = 1,2 * 0,75 = 0,9. \quad (5.17)$$

Подставив полученные значения в пропорцию, предназначенную для поиска суммарной площади световых проёмов, получим (5.18):

$$\frac{0,9*1,2*23}{0,405*10} * 1 = 6,13. \quad 100 \frac{S_0}{42} = \quad (5.18)$$

Из полученной пропорции найдем суммарную площадь световых проёмов (5.19):

$$S_0 = \frac{6,13*42}{100} = 2,6 \text{ (м}^2\text{)} \quad (5.19)$$

Таким образом, получается, что в большом помещении площадь окон превышает рекомендуемую суммарную площадь световых проёмов ( $4,3 \text{ м}^2 > 2,6 \text{ м}^2$ ) и обеспечивает достаточную освещенность в помещении.

## 5.2 Расчет искусственного освещения

Расчет искусственного освещения включает в себя такие важные пункты, как подбор системы освещения, типа ламп, которые будут использоваться для помещения, их количество, а также месторасположение в помещении.

В большинстве случаев, для расчета освещения в помещениях используются лампы двух видов: люминесцентные лампы и лампы накаливания. В последние годы проектировщики освещенности все чаще и чаще делают выбор в пользу люминесцентных ламп, так как они имеют ряд ключевых преимуществ перед лампами накаливания:

- КПД этих ламп в 1,5 – 2 раза превышает эту характеристику у ламп накаливания;

- при этом люминесцентные лампы имеют светоотдачу на порядок выше, чем лампы накаливания (в 3 – 4 раза);

- их срок службы более длительный, что позволяет предприятию экономить на замене ламп, которые вышли из строя, так как делать это приходится реже.

Для расчета будет использоваться метод коэффициента использования светового потока. Он используется для помещений с общим равномерно распределенным по горизонтальной поверхности светом, что подходит для данного случая.

Необходимо рассчитать освещенность в помещении и сравнить ее с минимальной нормированной освещенностью. В том случае, если освещенность окажется недостаточной для полноценного освещения, то необходимо рассчитать план по замене ламп в помещении на другие, с более мощным световым потоком.

Нормированная минимальная освещенность для офисного помещения равна 300 Лк.

### 5.2.1 Расчет для малого помещения

Расчет производится для помещения площадью 24 м<sup>2</sup> (ширина А = 4 м и длина В = 6 м).

Освещенность рассчитывается по формуле (5.20):

$$E = \frac{F \times n}{K \times S \times Z} \quad (5.20)$$

где:

- E - рассчитываемая освещенность, Лк.
- F - световой поток, обеспечиваемый нынешними лампами, Лм;
- S - площадь освещаемого помещения (в нашем случае S = 24м<sup>2</sup>);
- Z - отношение средней освещенности к минимальной (обычно принимается равным 1,1...1,2, пусть Z = 1,1);
- K - коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение зависит от типа помещения и характера проводимых в нем работ и в нашем случае K = 1,5);

–  $n$  - коэффициент использования (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризуемых коэффициентами отражения от стен ( $P_{\text{стены}}$ ), пола ( $P_{\text{пол}}$ ) и потолка ( $P_{\text{потолок}}$ )), значение коэффициентов:  $P_{\text{стены}} = 50\%$ ,  $P_{\text{пол}} = 30\%$ ,  $P_{\text{потолок}} = 70\%$ . Значение  $n$  определяется по таблице коэффициентов использования различных светильников. Для этого вычислим индекс помещения по формуле (5.21):

$$I = \frac{S}{h \times (A+B)}, \quad (5.21)$$

где:

$S$  - площадь помещения,  $S = 24 \text{ м}^2$ ;

$h$  - расчетная высота подвеса,  $h = 2.92 \text{ м}$ ;

$A$  - ширина помещения,  $A = 4 \text{ м}$ ;

$B$  - длина помещения,  $B = 6 \text{ м}$ .

Подставив значения получим результат (5.22):

$$I = \frac{24}{2,92 \times (4+6)} = \frac{24}{29,2} = 0,82 \quad (5.22)$$

Вычислив индекс помещения  $I = 0,82$ , определим по таблице на рисунке 5.3 значение коэффициента использования  $n = 0,46$ .

$\rho_{\text{т}}, \%$	70	70	50	30	0
$\rho_{\text{с}}, \%$	50	50	30	10	0
$\rho_{\text{р}}, \%$	30	10	10	10	0
$i$					
0,5	24	22	20	17	16
0,6	34	32	26	23	21
0,7	42	39	34	30	29
0,8	46	44	38	34	33
0,9	49	47	41	37	36
1	51	49	43	39	37
1,1	53	50	45	41	39
1,25	56	52	47	43	41
1,5	60	55	50	46	44
1,75	63	58	53	48	46
2	66	60	55	51	49
2,25	68	62	57	53	51
2,5	70	64	59	55	53
3	73	66	62	58	56
3,5	76	68	64	61	59
4	78	70	66	62	60
5	81	73	69	64	62

Рисунок 5.3 – Коэффициент использования светового потока

Рассчитаем световой поток, обеспечиваемый нынешними лампами (5.23):

$$F = F_{\text{л}} \times N \quad (5.23)$$

где:

$F_{\text{л}}$  – световой поток одной лампы, Лм;

$N$  – количество ламп в помещении.

$$F = 420 \times 6 = 2520 \text{ Лм} \quad (5.24)$$

Подставим значения в формулу найдем величину освещенности (5.25):

$$E = \frac{2520 \times 0,46}{1,5 \times 24 \times 1,1} = 29,27 \text{ Лк} \quad (5.25)$$

Полученное значение освещенности приблизительно в 10 раз меньше, чем минимальное нормированное значение освещенности.

В качестве мероприятий по улучшению условий освещенности было выбрано изменить лампы таким образом, чтобы световой поток новых ламп был значительно выше, чем у предыдущих.

Для освещения выбираем компактную люминесцентную лампу фирмы Phillips типа TL-D 58W/33, световой поток которой  $F = 4600$  Лк.

Рассчитаем световой поток, который будут обеспечивать новые лампы (5.26):

$$F = 4600 \times 6 = 27\,600 \text{ Лм} \quad (5.26)$$

Рассчитаем новую освещенность (5.27):

$$E = \frac{27\,600 \times 0,46}{1,5 \times 24 \times 1,1} = 320,6 \text{ Лк} \quad (5.27)$$

Значение освещенности с новыми лампами немногим превосходит значение минимальной нормированной освещенности.

Таким образом получается, что для увеличения освещенности до необходимой нормы, достаточно изменить тип ламп на более мощные.

### 5.2.2 Расчет для большого помещения

Расчет производится для помещения площадью  $42 \text{ м}^2$  (ширина  $A = 6$  м и длина  $B = 7$  м).

Освещенность рассчитывается по формуле (5.28):

$$E = \frac{F \times n}{K \times S \times Z} \quad (5.28)$$



где:

E - рассчитываемая освещенность, Лк.

F - световой поток, обеспечиваемый нынешними лампами, Лм;

S - площадь освещаемого помещения (в нашем случае  $S = 42\text{м}^2$ );

Z-отношение средней освещенности к минимальной (обычно принимается равным 1,1...1,2, пусть  $Z = 1,1$ );

K - коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение зависит от типа помещения и характера проводимых в нем работ и в нашем случае  $K = 1,5$ );

n - коэффициент использования (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризуемых коэффициентами отражения от стен ( $P_{\text{стены}}$ ), пола ( $P_{\text{пол}}$ ) и потолка ( $P_{\text{потолок}}$ )), значение коэффициентов:  $P_{\text{стены}} = 50\%$ ,  $P_{\text{пол}} = 30\%$ ,  $P_{\text{потолок}} = 70\%$ . Значение n определяется по таблице коэффициентов использования различных светильников.

Для этого вычислим индекс помещения по формуле (5.29):

$$I = \frac{S}{h \times (A+B)}, \quad (5.29)$$

где:

S - площадь помещения,  $S = 42 \text{ м}^2$ ;

h - расчетная высота подвеса,  $h = 2.92 \text{ м}$ ;

A - ширина помещения,  $A = 4 \text{ м}$ ;

B - длина помещения,  $B = 6 \text{ м}$ .

Подставив значения получим результат (5.30):

$$I = \frac{42}{2,92 \times (6+7)} = \frac{42}{37,96} = 1,1 \quad (5.30)$$

Вычислив индекс помещения  $I = 0,82$ , определим по таблице на рис.1 значение коэффициента использования  $n = 0,53$ .

Рассчитаем световой поток, обеспечиваемый нынешними лампами (5.31):

$$F = F_{\text{л}} \times N, \quad (5.31)$$

где:

$F_{\text{л}}$  – световой поток одной лампы, Лм;

$N$  – количество ламп в помещении.

$$F = 420 \times 9 = 3780 \text{ Лм} \quad (5.32)$$

Подставим все известные значения в формулу 1.1 и найдем величину освещенности (5.33):

$$E = \frac{3780 \times 0,53}{1,5 \times 42 \times 1,1} = 28,9 \text{ Лк} \quad (5.33)$$

Полученное значение освещенности приблизительно в 10 раз меньше, чем минимальное нормированное значение освещенности.

В качестве мероприятий по улучшению условий освещенности было выбрано изменить лампы таким образом, чтобы световой поток новых ламп был значительно выше, чем у предыдущих.

Для освещения выбираем компактную люминесцентную лампу фирмы Phillips типа TL-D 58W/33, световой поток которой  $F = 4600$  Лк.

Рассчитаем световой поток, который будут обеспечивать новые лампы (5.34) :

$$F = 4600 \times 9 = 41\ 400 \text{ Лм} \quad (5.34)$$

Рассчитаем новую освещенность (5.33):

$$E = \frac{41\,400 \times 0,53}{1,5 \times 42 \times 1,1} = 316,6 \text{ Лк} \quad (5.35)$$

Значение освещенности с новыми лампами немногим превосходит значение минимальной нормированной освещенности.

Таким образом получается, что для увеличения освещенности до необходимой нормы, достаточно изменить тип ламп на более мощные.

Проведя ряд расчетов для выяснения причин проблем с освещенностью в помещениях, были сделаны следующие выводы:

– в обоих помещениях соблюдаются нормы естественного освещения. Это означает, что в ясный день в этих помещениях не будет проблем с освещенностью;

– искусственное освещение в обоих помещениях слишком тусклое, и если его не изменить, то скоро это приведет к проблемам со здоровьем у служащих в помещениях.

Для устранения проблемы искусственного освещения в помещениях было принято следующее решение: изменить обычные лампы накаливания в обоих помещениях на новые, более мощные, дающие больший световой поток. Такое несложное мероприятие позволит улучшить условия работы в сумеречное время и пасмурные дни.

## Заключение

В результате выполнения дипломной работы, было разработано Android приложение для медицинского учреждения. Приложение работает на различных версиях Android и разных разрешениях экрана начиная с разрешения 800x480, заканчивая самым максимальным разрешением 4к.

Были сравнены различные мобильные платформы, основное внимание уделялось платформам, для разработки под которые используется язык Java. Было разработано приложение для улучшения и укрепления памяти, в качестве платформы была выбрана Android. При разработке использовалась среда Android Studio с установленным плагином Android Developer Tools и комплектом Android SDK. Приложение тестировалось на эмуляторе, поддерживающем самые последние версии мобильных устройств. Также предполагается расширение функционала приложения с целью улучшения его эффективности. Ввиду востребованности и роста популярности мобильных приложений, в частности приложений подобной направленности, можно сделать вывод, что проделанная работа является актуальной.

Таким образом, была проведена разработка приложения, которое полноценно выполняет все возложенные на него функции и предоставляет своим пользователям использование всего разработанного функционала.

## Список литературы

- 1 Стив Круг; [пер. с англ. М.А. Райтмана]. – 3-е издание. – Москва: Эксмо, 2018. – 256 с.: ил. – (Мировой компьютерный бестселлер). «Не заставляйте меня думать»
- 2 Кириченко А. В., Хрусталева А. А. – СПб: «Наука и Техника», 2018. – 352 с., / «ANDROID. Основы современного дизайна»
- 3 Джон Дакетт; [пер. с англ. М. А. Райтмана]. – Москва: Эксмо, 2019. – 480 с.: ил. - (Мировой компьютерный бестселлер). / «ANDROID. Разработка и дизайн приложений»
- 4 Д. Флэнаган. – СПб: Символ-Плюс, 2012. – 1080 с., ил. «Android. Подробное руководство, 6-е издание»
- 5 А. Н. Васильев. – Москва: Издательство «Э», 2017. – 720 с. – (Российский компьютерный бестселлер). «Android в примерах и задачах»
- 6 Джон Дакетт; [пер. с англ. М. А. Райтмана]. — Москва: Эксмо, 2017. — 640 с.: ил. — (Мировой компьютерный бестселлер). «Android studio. Интерактивная мобильных приложений»
- 7 А. П. Никольский. – СПб.: «Наука и техника», 2017. – 272 с. «Android на примерах»
- 8 Я. Файн, А. Моисеев. – СПб.: Питер, 2018. – 464 с. «Android и Android studio. Создание мобильных приложений для профессионалов»
- 9 <https://metanit.com/web/typescript/>; «Руководство по android studio»
- 10 Герберт Шилдт; [пер. с англ. И. Бернштейна]. – Москва: Вильямс, 2017. – 1376 с. «Java 8. Полное Руководство»
- 11 Р. В. Сеттер. – СПб: «Наука и Техника», 2016. – 240 с., ил. «Изучаем Java на примерах и задачах»
- 12 [Электронный ресурс] – Режим доступа: <https://ria.ru/sport/20150520/1065455422.html> / ВЦИОМ: число Казахстанн, занимающихся спортом
- 13 [Электронный ресурс] – Режим доступа: <http://topmira.com/tehnika/item/176-os-smartfony/> / Лучшие операционные системы для смартфонов
- 14 Режим доступа: <http://hightech.in.ua/content/art-eclipse-platform> Обзор платформы Eclipse [Электронный ресурс]
- 15 Режим доступа: <http://it-news.club/13588422> Сколько дней в году люди тратят на смартфоны [Электронный ресурс]
- 16 Режим доступа: <https://msdn.microsoft.com/ru-ru/library/mt299001.aspx> / Visual Studio и Xamarin [Электронный ресурс] –
- 17 Режим доступа: [https://ru.wikipedia.org/wiki/Android\\_Studio](https://ru.wikipedia.org/wiki/Android_Studio) Android Studio. Wikipedia [Электронный ресурс]
- 18 Режим доступа: <http://progopedia.ru/language/java> Java. Progopedia [Электронный ресурс]

19 <https://geektimes.ru/post/242645/> СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 30.12.2013. – Красноярск: СФУ, 2014. – 60с.

20 Гаврилов А. В., Клименков С. В., Харитонов А.Е., Цопа Е. А. – СПб: Университет ИТМО, 2015. – 126 с / «Программирование на языке Java. Конспект лекций» /

## Приложение А (обязательное)

### Техническое задание

#### **Техническое задание для разработки системы автоматической регистрации парковочных мест**

##### 1. Общие требования:

- наименование разрабатываемой системы:
- мобильное приложение для медицинского центра(МПдМЦ).
- цель разработки:
- повышение качества оказываемых услуг;
- оптимизация работы медицинского центра.
- запись на прием удаленно через мобильное приложение
- увеличение обработки заявок;
- предлагаемые технологии для разработки системы (на выбор разработчика):
  - google Cloud Platform;
  - amazon Web Services;
  - openCV;
  - simpleCV;
  - android Studio
  - eclipse;
  - luminoth;
  - выбор архитектуры построения:
  - клиент-Сервер.
  - предлагаемые языки и технологии программирования:
  - python;
  - ruby;
  - otlin;
  - java;
  - c#;
  - rust;
  - c++.
- общий объем программной части системы, Мб

- не более 500 Мб.

## 2. Технические требования:

- требования к программному обеспечению:
- умеренная скорость обработки входящей информации;

### *Продолжение приложения А*

- возможность работы с облачными решениями;
- код программы должен быть минималистичным и понятным.
- требования к аппаратному обеспечению:
- наличие минимального количества оборудования;
- поддержка системы;
- возможность подключаться к сети Интернет.
- тестирование и отладка системы:
- проверка работоспособности системы на тестовом объекте;
- тестирование системы на вероятность ложного позитива;
- тестирование системы на сбой.

## 3. Специфические требования:

- адаптивность системы:
- должна присутствовать возможность адаптировать систему под различные условия, в зависимости оборудования, присутствующее в медицинском учреждении;
- должна быть возможность добавления функционала к системе без каких-либо серьезных изменений программного и аппаратного обеспечения.

## 4. Экономические требования:

- расчет стоимости системы и стоимости разработки программного обеспечения (подлежит обсуждению):
- стоимость готового продукта 2 000 000тг;
- стоимость разработки 1 600 000тг.
- Потенциальные клиенты и области применения:
- медицинские учреждения;
- больницы;





## Приложение Б (обязательное)

### Листинг программы

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.mohit.splash">
    <uses-permission android:name="android.permission.INTERNET"></uses-permission>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.NoActionBar">
        <activity android:name=".splash">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".MainActivity"
            android:windowSoftInputMode="stateHidden"/>
        <activity
            android:name=".Hipage"
            android:theme="@style/Theme.AppCompat.Light.NoActionBar" />
        <activity android:name=".DetailsInfo"
            android:theme="@style/Theme.AppCompat.Light.NoActionBar"/>
    </application>
</manifest>
```

```
</application>
```

```
</manifest>
```

```
package com.example.mohit.splash;
```

```
import android.annotation.SuppressLint;
```

```
import android.os.Bundle;
```

```
import android.support.design.widget.AppBarLayout;
```

```
import android.support.design.widget.TabLayout;
```

```
import android.support.v4.view.ViewPager;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.widget.TextView;
```

```
import org.json.JSONArray;
```

```
import org.json.JSONException;
```

```
import org.json.JSONObject;
```

### *Продолжение приложения Б*

```
public class DetailsInfo extends AppCompatActivity {
```

```
    public TabLayout tabLayout;
```

```
    public AppBarLayout appBarLayout;
```

```
    public ViewPager viewPager;
```

```
    String json_string;
```

```
    JSONArray jsonArray;
```

```
    JSONArray jsonArray1,jsonArray2,jsonArray3,jsonArray4;
```

```
    JSONObject jsonObject;
```

```
    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_details);

    json_string=getIntent().getExtras().getString("json_data");

    Bundle bundle = new Bundle();
    Bundle bundle1 = new Bundle();
    Bundle bundle2 = new Bundle();

    try {
        jsonObject=new JSONObject(json_string);
        jsonArray = jsonObject.getJSONArray("server_response");

        int count=0;

        String Disease_name,Symptom,Medicine,stringArray;

        JSONObject JO= jsonArray.getJSONObject(0);
        Disease_name =JO.getString("Disease_name");

        jsonArray = JO.getJSONArray("Symptoms");
        jsonArray1 = JO.getJSONArray("Medicines");
        jsonArray2 = JO.getJSONArray("Price");
        jsonArray3 = JO.getJSONArray("Shop_name");
        jsonArray4 = JO.getJSONArray("Address");

        String json = "{\"Symptoms\":"+jsonArray+"}";
        JSONObject object = (JSONObject) new JSONTokener(json).nextValue();
        bundle.putString("params",object.toString());
    }
}

```

```
String json1 = "{\"Medicines\":"+jsonArray1+",\"+\"Price\":"+jsonArray2+"}";
JSONObject object1 = (JSONObject) new JSONTokener(json1).nextValue();
bundle1.putString("params",object1.toString());
```

```
String json2 = "{\"Shop_name\":"+jsonArray3+",\"+\"Address\":"+jsonArray4+"}";
JSONObject object2 = (JSONObject) new JSONTokener(json2).nextValue();
bundle2.putString("params",object2.toString());
```

### *Продолжение приложения Б*

```
    } catch (JSONException e) {
        e.printStackTrace();
    }
```

```
tabLayout = (TabLayout) findViewById(R.id.tabLayout_id);
appBarLayout = (AppBarLayout) findViewById(R.id.appBarLayout_id);
viewPager = (ViewPager) findViewById(R.id.viewPager_id);
ViewPagerAdapter adapter = new ViewPagerAdapter(getSupportFragmentManager());
viewPager.addOnPageChangeListener(new
TabLayout.TabLayoutOnPageChangeListener(tabLayout));
//Adding Fragment
FragmentInitial myObj = new FragmentInitial();
adapter.AddFragment(myObj, "Symptoms");
myObj.setArguments(bundle);
//adapter.AddFragment(new FragmentInitial(), "Symptoms");
FragmentIntermediate myObj1 = new FragmentIntermediate();
adapter.AddFragment(myObj1, "Medicines");
myObj1.setArguments(bundle1);
```

```

        FragmentCritical myObj2 = new FragmentCritical();
        adapter.AddFragment(myObj2,"Medical Stores");
        myObj2.setArguments(bundle2);

        //adapter.AddFragment(new FragmentIntermediate(),"Medicines");
        //adapter.AddFragment(new FragmentCritical(),"Medical Stores");
        //Adapter Setup

        viewPager.setAdapter(adapter);
        tabLayout.setupWithViewPager(viewPager);

    }
}
package com.example.mohit.splash;

import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.BundleCompat;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

/**
 * Created by hp on 29-Mar-18.

```

## *Продолжение приложения Б*

\*/

```
public class FragmentCritical extends Fragment {
    View view;
    String strtext;
    TextView textView;
    JSONArray jsonArray;
    JSONArray jsonArray1;
    JSONObject jsonObject;
    public FragmentCritical() {

    }

    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable
Bundle savedInstanceState) {
        strtext = getArguments().getString("params");

        view = inflater.inflate(R.layout.critical_fragment, container,false);

        //Bundle bundle=getArguments();
        //String jsonString=bundle.getString("key");

        JSONObject obj=null;
        try {
            obj = new JSONObject(strtext);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}
```

```

try {
    jsonArray = obj.getJSONArray("Shop_name");
    jsonArray1 = obj.getJSONArray("Address");
} catch (JSONException e) {
    e.printStackTrace();
}

```

```

StringBuilder s = new StringBuilder(100);
for(int i = 0, cnt = jsonArray.length(); i < cnt; i++)
{

```

```

    try {
        String jsonString = jsonArray.getString(i);
        s.append(i+1);
        s.append(" . ");
        s.append(jsonString.toString());
        s.append("\n");

```

```

        String str = jsonArray1.getString(i);
        s.append(str.toString());
        s.append("\n \n");

```

*Продолжение приложения Б*

```

    }
    catch (JSONException e) {
        e.printStackTrace();
    }
}

/*StringBuilder s1 = new StringBuilder(100);
for(int i = 0, cnt = jsonArray1.length(); i < cnt; i++)
{
    try {

```



```
        String jsonString = jsonArray1.getString(i);
        s1.append(jsonString.toString());
        s1.append("\n \n");
    }
    catch (JSONException e) {
        e.printStackTrace();
    }
}*/
```

```
textView = (TextView) view.findViewById(R.id.text3);
textView.setText(s);
```

```
/*textView = (TextView) view.findViewById(R.id.text4);
textView.setText(s);*/
return view;
}
```

```
@Override
```

```
public void onCreateView(View view, Bundle savedInstanceState) {
    super.onCreateView(view, savedInstanceState);
}
```

```
}
```

```
package com.example.mohit.splash;
```

```
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v4.app.BundleCompat;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
```

```
import android.view.ViewGroup;
import android.widget.TextView;
```

```
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
```

```
public class FragmentInitial extends Fragment {
    View view;
    String strtext;
```

*Продолжение приложения Б*

```
    TextView textView;
    JSONArray jsonArray;
    JSONObject jsonObject;
    public FragmentInitial() {

    }
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable
Bundle savedInstanceState) {

        strtext = getArguments().getString("params");

        view = inflater.inflate(R.layout.initial_fragment, container,false);

        //Bundle bundle=getArguments();
        //String jsonString=bundle.getString("key");

        JSONObject obj=null;
```

```

try {
    obj = new JSONObject(strtext);
} catch (JSONException e) {
    e.printStackTrace();
}

try {
    jsonArray = obj.getJSONArray("Symptoms");
} catch (JSONException e) {
    e.printStackTrace();
}

StringBuilder s = new StringBuilder(100);
for(int i = 0, cnt = jsonArray.length(); i < cnt; i++)
{
    try {
        String jsonString = jsonArray.getString(i);
        s.append(i+1);
        s.append(" . ");
        s.append(jsonString.toString());
        s.append("\n\n");
    }
    catch (JSONException e) {
        e.printStackTrace();
    }
}

textView = (TextView) view.findViewById(R.id.text);
textView.setText(s);
return view;

```

```
}
```

### *Продолжение приложения Б*

```
@Override
```

```
public void onViewCreated(View view, Bundle savedInstanceState) {  
    super.onViewCreated(view, savedInstanceState);  
}
```

```
}
```

```
package com.example.mohit.splash;
```

```
import android.content.Context;  
import android.os.AsyncTask;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.TextView;  
import android.content.Intent;  
import android.view.View;  
import android.widget.Toast;  
import android.app.ProgressDialog;  
import java.io.BufferedReader;  
import java.io.BufferedWriter;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.UnsupportedEncodingException;  
import java.io.InputStreamReader;  
import java.io.OutputStream;  
import java.io.OutputStreamWriter;  
import java.net.HttpURLConnection;
```

```

import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import in.galaxyofandroid.spinnerdialog.OnSpinnerItemClick;
import in.galaxyofandroid.spinnerdialog.SpinnerDialog;

public class Hipage extends AppCompatActivity {
    String json_string;
    String JSON_STRING;
    String temp;
    String prev="Wait";
    String jsonurl= "https://evacuative-builders.000webhostapp.com/getfromjson.php";

    ArrayList<String> items=new ArrayList<>();
    SpinnerDialog spinnerDialog;
    protected Button btnShow;

```

*Продолжение приложения Б*

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_hipage2);
    //final Intent i = new Intent(this,DetailsInfo.class);

    Bundle appleData = getIntent().getExtras();

```

```

if(appleData==null)
    return;
String firstmessage=appleData.getString("firstmessage");
String[] splited = firstmessage.split("\\s+");
final TextView Second_Text=(TextView) findViewById(R.id.Second_Text);
Second_Text.setText("Hi "+splited[0]);

initItems();

{
    spinnerDialog = new SpinnerDialog(Hipage.this, items, "select items");
    spinnerDialog.bindOnSpinerListener(new OnSpinerItemClick() {
        @Override
        public void onClick(String item, int position) {

            Toast.makeText(Hipage.this, item, Toast.LENGTH_SHORT).show();
            new JSONTask().execute(jsonurl, item);
        }
    });

    btnShow = (Button) findViewById(R.id.btnShow);
    btnShow.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            spinnerDialog.showSpinerDialog();
        }
    });
}

```

```
}  
}
```

```
private void initItems() {  
    for(int i=0;i<100;i++)  
    {  
        items.add("Asthma");  
        items.add("Allergies");  
        items.add("Underactive thyroid");  
        items.add("Yellow fever");  
        items.add("Cold");  
        items.add("Fever");
```

*Продолжение приложения Б*

```
        items.add("Gallstones");  
        items.add("Flu");  
        items.add("Food Poisoning");  
        items.add("Dysphagia");  
        items.add("Dehydration");  
        items.add("Dental abscess");  
        items.add("Diarrhoea");  
        items.add("Depression");  
        items.add("Gallstones");  
        items.add("Gum disease");  
        items.add("High Cholesterol");  
        items.add("Hepatitis B");  
        items.add("Indigestion");  
        items.add("Measles");  
        items.add("Malaria");  
        items.add("Insomnia");
```

```

items.add("Pneumonia");
items.add("Sunburn");
items.add("Tonsillitis");
items.add("Tuberculosis (TB)");
items.add("Ringworm");
items.add("Osteoarthritis");
items.add("Iron Deficiency Anaemia");
items.add("Labyrinthitis");
break;
}
}

```

```

public class JSONTask extends AsyncTask<String,Void,String >{
    ProgressDialog loading;
    @Override
    protected String doInBackground(String... params) {

        HttpURLConnection connection = null;
        BufferedReader reader = null;
        URL url ;

        try {
            url = new URL(params[0]);
            connection = (HttpURLConnection) url.openConnection();
            //connection.connect();
            connection.setRequestMethod("POST");
            connection.setDoInput(true);
            connection.setDoOutput(true);

```



```
HashMap<String,String > data = new HashMap<String, String>();
data.put("Disease_name",params[1]);
OutputStream os = connection.getOutputStream();
BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(os, "UTF-8"));
```

*Продолжение приложения Б*

```
writer.write(getPostDataString(data));
writer.flush();
writer.close();
os.close();

int responseCode=connection.getResponseCode();

String response ="";
if (responseCode == HttpURLConnection.HTTP_OK) {
    reader=new BufferedReader(new InputStreamReader(connection.getInputStream()));
    temp = reader.readLine();
    response = temp;
}
else {
    response="Error Registering";
}

return response;
}
catch (MalformedURLException e) {
    e.printStackTrace();
    return "MalformedURLException Exception";
} catch (IOException e) {
    e.printStackTrace();
    return "IO Exception";
```

```

    }

}

@Override
protected void onPreExecute() {
    super.onPreExecute();
    loading = ProgressDialog.show(Hipage.this, "Please Wait", null, true, true);
}

```

```

@Override
protected void onPostExecute(String s) {
    //super.onPostExecute(s);
    loading.dismiss();
    Intent intent_name = new Intent();
    intent_name.setClass(getApplicationContext(),DetailsInfo.class);
    intent_name.putExtra("json_data",s);
    startActivity(intent_name);
}

```

```

private String postDataString(HashMap<String, String> params) throws
UnsupportedEncodingException {

```

```

    StringBuilder result = new StringBuilder();
    boolean first = true;
    for (Map.Entry<String, String> entry : params.entrySet()) {

```

*Продолжение приложения Б*

```

        if (first)
            first = false;
        else
            result.append("&");

```

```

        result.append(URLEncoder.encode(entry.getKey(), "UTF-8"));
        result.append("=");
        result.append(URLEncoder.encode(entry.getValue(), "UTF-8"));
    }

    return result.toString();
}
}

}

package com.example.mohit.splash;

import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ListView;
import android.widget.TextView;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.StringReader;
import java.lang.Void;

```

```

import java.lang.String;
import java.net.URL;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;

public class Hipage5 extends AppCompatActivity {

    String json_string;
    JSONArray jsonArray;
    JSONObject jsonObject;

    ListView listView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Продолжение приложения Б

        setContentView(R.layout.activity_hipage5);
        json_string=getIntent().getExtras().getString("json_data");

        try {
            jsonObject=new JSONObject(json_string);
            jsonArray = jsonObject.getJSONArray("server_response");

            int count=0;

            String Disease_name,Symptom,Medicine,stringArray;
            StringBuilder s = new StringBuilder(100);

```

```

JSONObject JO= jsonArray.getJSONObject(0);
Disease_name =JO.getString("Disease_name");
TextView textView2= (TextView) findViewById(R.id.textView2);
textView2.setText(Disease_name);

jsonArray = JO.getJSONArray("Price");

String[] string = new String[jsonArray.length()];
for(int i = 0, cnt = jsonArray.length(); i< cnt; i++)
{
    try {
        String jsonString = jsonArray.getString(i);
        s.append(jsonString.toString());
        s.append(", ");
    }
    catch (JSONException e) {
        e.printStackTrace();
    }
}

} catch (JSONException e) {
    e.printStackTrace();
}

}

```

```
}
```

```
package com.example.mohit.splash;
```

```
import android.support.v7.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

*Продолжение приложения Б*

```
import android.view.MotionEvent;
```

```
import android.view.Window;
```

```
import android.view.View;
```

```
import android.view.WindowManager;
```

```
import android.view.Menu;
```

```
import android.view.inputmethod.InputMethodManager;
```

```
import android.view.MenuItem;
```

```
import android.widget.Button;
```

```
import android.widget.TextView;
```

```
import android.content.Intent;
```

```
import android.content.Context;
```

```
import android.widget.EditText;
```

```
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    EditText First_Input;
```

```
    EditText Second_Input;
```

```
    EditText Third_Input;
```

```
    EditText Fourth_Input;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

this.getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);
}

@Override // for keyboard goes disappear
public boolean dispatchTouchEvent(MotionEvent ev) {
    View view = getCurrentFocus();

    if (view != null && (ev.getAction() == MotionEvent.ACTION_UP || ev.getAction() == MotionEvent.ACTION_MOVE) && view instanceof EditText && !view.getClass().getName().startsWith("android.webkit.")) {
        int scrcoords[] = new int[2];
        view.getLocationOnScreen(scrcoords);
        float x = ev.getRawX() + view.getLeft() - scrcoords[0];
        float y = ev.getRawY() + view.getTop() - scrcoords[1];
        if (x < view.getLeft() || x > view.getRight() || y < view.getTop() || y > view.getBottom())

        ((InputMethodManager)this.getSystemService(Context.INPUT_METHOD_SERVICE)).hideSoftInputFromWindow((this.getWindow().getDecorView().getApplicationWindowToken()), 0);
    }

    if (super.dispatchTouchEvent(ev)) return true;
    else return false;
}

```

### *Продолжение приложения Б*

```

public void onClick(View view){
    Intent i = new Intent(this,HiPage.class);

    First_Input = (EditText) findViewById(R.id.editText);
}

```

```

String usermessage = First_Input.getText().toString();

if (usermessage.matches("")) {
    Toast.makeText(this, "You did not enter a username", Toast.LENGTH_SHORT).show();
    return;
}

Second_Input = (EditText) findViewById(R.id.editText2);
String usermessage2 = Second_Input.getText().toString();

if (usermessage2.matches("")) {
    Toast.makeText(this, "You did not enter a password", Toast.LENGTH_SHORT).show();
    return;
}

Third_Input = (EditText) findViewById(R.id.editText3);
String usermessage3 = Third_Input.getText().toString();

if (usermessage3.matches("")) {
    Toast.makeText(this, "You did not enter a email", Toast.LENGTH_SHORT).show();
    return;
}

{
    boolean x = isEmailValid(usermessage3);

    if(!x) {
        Toast.makeText(this, "You did not enter a valid email",
Toast.LENGTH_SHORT).show();
        return;
    }
}

```



```
}
```

```
Fourth_Input = (EditText) findViewById(R.id.editText4);
```

```
String usermessage4 = Fourth_Input.getText().toString();
```

```
if (usermessage4.matches("")) {
```

```
    Toast.makeText(this, "You did not enter a contact no",  
    Toast.LENGTH_SHORT).show();
```

```
    return;
```

```
}
```

```
{
```

```
    boolean y = isMobileValid(usermessage4);
```

### *Продолжение приложения Б*

```
if(!y) {
```

```
    Toast.makeText(this, "You did not enter a valid phone no",  
    Toast.LENGTH_SHORT).show();
```

```
    return;
```

```
}
```

```
}
```

```
i.putExtra("firstmessage",usermessage);
```

```
startActivity(i);
```

```
}
```

```
@Override
```

```
public void onPause(){
```

```
    super.onPause();
```

```

        // Clear all value here
        // editTextone.setText("");
        // editTexttwo.setText("");
        First_Input.setText("");
        Second_Input.setText("");
        Third_Input.setText("");
        Fourth_Input.setText("");
    }

    boolean isEmailValid(CharSequence email) {
        return android.util.Patterns.EMAIL_ADDRESS.matcher(email).matches();
    }

    private boolean isMobileValid(String phone) {
        return android.util.Patterns.PHONE.matcher(phone).matches();
    }

}

package com.example.mohit.splash;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.TextView;

public class splash extends AppCompatActivity {
    private TextView tv;

```

```

private ImageView iv;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_splash);

        Продолжение приложения Б

    tv = (TextView) findViewById(R.id.tv);
    iv = (ImageView) findViewById(R.id.iv);

    Animation myanim = AnimationUtils.loadAnimation(this,R.anim.mytransition);
    tv.startAnimation(myanim);
    iv.startAnimation(myanim);

    final Intent i = new Intent(this,MainActivity.class);

    Thread timer=new Thread(){
        public void run(){
            try{
                sleep(4000);
            } catch(InterruptedException e) {
                e.printStackTrace();
            }
            finally{
                startActivity(i);
                finish();
            }
        }
    };
    timer.start();
}

```

```

}
package com.example.mohit.splash;

import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;

import java.util.ArrayList;
import java.util.List;

public class ViewPagerAdapter extends FragmentPagerAdapter{

    private final List<Fragment> fragmentList = new ArrayList<>();
    private final List<String> FragmentListTitles = new ArrayList<>();

    public ViewPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        return fragmentList.get(position);
    }

    @Override
    public int getCount() {
Продолжение приложения Б
        return FragmentListTitles.size();
    }
}

```

```

@Override
public CharSequence getPageTitle(int position) {
    switch (position) {
        case 0:
            return "Symptoms";
        case 1:
            return "Medicines";
        case 2:
            return "Stores";
    }
    return null;
}

public void AddFragment(Fragment fragment,String Title) {
    fragmentList.add(fragment);
    FragmentListTitles.add(Title);
}

}

package com.example.mohit.splash;

import android.content.Context;
import android.support.test.InstrumentationRegistry;
import android.support.test.runner.AndroidJUnit4;

import org.junit.Test;
import org.junit.runner.RunWith;

import static org.junit.Assert.*;

```

```
/**
 * Instrumented test, which will execute on an Android device.
 *
 * @see <a href="http://d.android.com/tools/testing">Testing documentation</a>
 */
@RunWith(AndroidJUnit4.class)
public class ExampleInstrumentedTest {
    @Test
    public void useAppContext() throws Exception {
        // Context of the app under test.
        Context appContext = InstrumentationRegistry.getTargetContext();

        assertEquals("com.example.mohit.splash", appContext.getPackageName());
    }
}
```

## Приложение В (обязательное)

### Акты внедрения

#### **Акт внедрения мобильного приложения «MediCare»**

Настоящий Акт свидетельствует, что мобильное приложение «MediCare», разработанное Сабырбек Т.О., внедрено в ИП «Черткова Н.О.».

Процесс внедрения проходил с 2 по 4 мая 2019 г.

Заявленные характеристики системы предполагали наличие следующих основных возможностей:

- поддерживать связь с клиентом, где бы он ни находился;
- облегчить доступ к контактной информации
- оповещать клиентов о проведении специальных акций, появлении новых обновлений и информации;
- собирать статистическую информацию о реальной клиентской базе и потенциальном потребителе для проведения маркетинговых исследований.

В ходе опытной эксплуатации мобильного приложения подтверждено, что оно обладает всеми заявленными возможностями.

На момент подписания настоящего Акта система установлена и эксплуатируется сотрудниками данной компании. Программой пользуются 21 человек.

Начальник технического отдела

Зубайра Т.Г.