

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий.

«Допущен к защите»  
Заведующий кафедрой \_\_\_\_\_

(Ф.И.О., ученая степень, звание)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка мобильного приложения для  
консультации клиентов компьютерного магазина

Специальность 5В070400 ВТ и ПО

Выполнил (а) Байрақ Н.С. ВТ-12-2  
(Фамилия и инициалы) группа

Научный руководитель Ахметова М.А., к.т.н., доцент  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бекмичева А.М., к.э.н., доцент

(Фамилия и инициалы, ученая степень, звание)

А-М « 03 » 05 2016 г.  
(подпись)

по безопасности жизнедеятельности:

Буркеев Н.Г., д.х.н., профессор

(Фамилия и инициалы, ученая степень, звание)

Н.Г. « 05 » 05 2016 г.  
(подпись)

по применению вычислительной техники:

Ахметова М.А., к.т.н., доцент

(Фамилия и инициалы, ученая степень, звание)

М.А. « 31 » 05 2016 г.  
(подпись)

Нормоконтролер: Ахметова М.А., к.т.н., доцент

(Фамилия и инициалы, ученая степень, звание)

М.А. « 31 » 05 2016 г.  
(подпись)

Рецензент:

(Фамилия и инициалы, ученая степень, звание)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Аэрокосмических и Информационных Технологий  
Специальность Вычислительная Техника и Программное Обеспечение  
Кафедра Компьютерных Технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Байсагір Жансерік Баядауылов  
(фамилия, имя, отчество)

Тема проекта Разработка мобильного приложения для  
консультации клиентов компьютерного магазина

утверждена приказом ректора № 21 от «10» марта 2016 г.

Срок сдачи законченной работы «\_\_» \_\_\_\_\_ 20\_\_ г.

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

Разработка мобильного приложения

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

- Анализ предметной области
- Выбор инструментов
- Технологическое обеспечение
- Практическая часть
- Технико-экономическое обоснование
- Безопасность жизнедеятельности

Перечень графического материала (с точным указанием обязательных чертежей)

- Блок-схема приложения
- Скриншоты приложения
- Структура каталогий
- Структурная диаграмма затрат на разработку

Рекомендуемая основная литература

- Бек Фрейм - HTML5 и CSS3.
- Веб-сайт "SuPrak" - <http://www.suprak.kz>
- Байзалова А.А - Охрана Труда
- Г.М. Лукасьян, Экономика от А до Я.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Эконом. часть	Бекмурзаева А.Ш	29.04 - 08.05.16	А.Ш
Б.И.И.	Трихобовко И.Г.	29.04 - 05.05.16	И.Г
Предметная область	Ахметова М.А.	21.04 - 19.05.16	М.А
Технологическая часть	Ахметова М.А.	10.03 - 17.03.16	М.А
Практическая часть	Ахметова М.А.	18.03 - 22.04.16	М.А
Методические материалы	Ахметова М.А.	05.04 - 07.04.16	М.А



## **Аннотация**

Тема дипломного проекта «Разработка мобильного приложения для консультации клиентов компьютерного магазина».

Данный дипломный проект включает в себя разработку мобильного приложения элементами которого является база знаний и база данных. Работа основана на связке данных компонентов, обеспечивая возможность дистанционной покупки компьютерных устройств.

Цель работы, разработка мобильного приложения, которое позволяет рационально выбирать компьютерную технику.

## **Андатпа**

Дипломдық жобаның тақырыбы «Компьютерлік дүкен тұтынушыларына кеңес беруге арналған мобильді қосымшаны құру».

Осы дипломдық жобада білім базасымен деректер қорына негізделу және есептеу құрылғыларын қашықтықтан сатып алу мүмкіндігі қарастырылған.

Жұмыстың басты мақсаты, компьютерлік техниканы ұтымды түрде сатып алуға мүмкіндік беретін беретін мобильді қосымшаны өңдеу.

## **Annotation**

The theme of the degree project «The Development of mobile applications for customer consultation of a computer store».

This project includes the development of mobile application element of which is the knowledge base and database. Is based on the conjunction of these components, providing the possibility of remote purchase computing devices.

The purpose of work, to develop a mobile application that allows you to rationally choose computer equipment.

## Содержание

Введение	8
1 Анализ предметной области	9
1.1 Обзор аналогов	9
1.2 Преимущества и недостатки приложений для подбора техники	11
1.3 Постановка задачи	12
1.3.1 Техническое задание	12
1.3.2 Требование к мобильному приложению	12
2 Подбор инструментов для разработки мобильного приложения	13
2.1 Инструменты для работы с графикой	13
2.2 Инструменты для разработки приложений	14
2.3 Инструменты для тестирования и отладки приложения	15
3 Технологическое обеспечение	16
3.1 Платформа для разработки гибридных приложений Framework7	16
3.1.1 Основные функции	16
3.1.2 Компоненты навигации	17
3.1.3 Компоненты пользовательского интерфейса	17
3.1.4 Утилиты	19
3.1.5 Сравнение Framework7 с похожими платформами разработки	20
3.2. Языки web-программирования	20
3.2.1 HTML5	20
3.2.2 CSS	20
3.2.3 JavaScript	22
3.3 Разработка приложений с помощью Apache Cordova	25
3.3.1 Обзор архитектуры	25
3.3.2 Поддерживаемые платформы устройств	25
4 Дизайн приложения в стиле Google Material Design	28
4.1 Обзор Material Design	28
4.2 Основные принципы Material Design	28
5 Практическая часть выполнения дипломного проекта	31
5.1 Порядок выполнения работы	31
5.2 Установка необходимых инструментов	31
5.3 Реализация приложения	31
5.4 Тестирование приложения	36
6 Техничко-экономическое обоснование	38
6.1 Описание работы и обоснование необходимости	38
6.2 Расчет затрат на разработку информационных технологий	38
6.3 Расчет цены программного продукта	46
6.4 Оценка социально - экономических результатов	47
7 Безопасность жизнедеятельности	48

7.1 Анализ труда	48
7.1.2 План помещения	48
7.1.3 Освещенность	49
7.1.4 Уровень шума	50
7.1.5 Оценка микроклимата	50
7.2 Расчет воздушной среды рабочей зоны	51
7.3 Расчет пожарной безопасности	54
Заключение	57
Список литературы	58
Приложение А	
Приложение Б	
Приложение В	
Приложение Г	

## Введение

Мобильные телефоны уже давно не являются чем-то необычным и неплохо справляются со своей функцией – являются средством коммуникации между людьми. Все смартфоны стали небольшой копией компьютера, который постоянно можно иметь при себе. В наше время стало популярным создание мобильных приложений для решения разных повседневных задач, от заказа еды, до оплаты счетов. В Казахстане в основном лидируют мобильные устройства под операционной системой Android от компании Google и iPhone под операционной системой IOS компании Apple. Разработка приложений для этих платформ является наиболее эффективным в плане дальнейшего развития и большой пользовательской базой.

С развитием интернета и информационных технологии, привычные для нас покупки и заказы теперь можно делать с помощью интернета прямо с мобильного телефона. Для реализации таких удобств для своих клиентов многие компании принимают решения создавать мобильные приложения. Не остается в стороне и компании, которые занимаются продажей компьютерной техники. В Казахстане уже существуют ряд крупных магазинов и дистрибьюторов электронной техники. К сожалению, не все из них имеют в своем вооружении мобильные приложения, которые не только позволили бы совершать покупку прямо в телефоне, но и максимально обезопасить пользователя от всякого рода обмана или неверных советов в выборе техники. Конечно, существуют сайты, которые поддерживают онлайн покупки и оплату. Но к сожалению, они не имеют мобильной версии, и с экрана мобильного телефона эти сайты неправильно отображаются и являются неудобными. Именно поэтому разрабатываются мобильные приложения, уже заранее ориентированные под мобильные устройства всех моделей и размеров. В сравнении с iPhone от компании Apple, устройства под управлением Android имеют очень много разновидностей, что усложняет разработку приложения, т.к. вся информация должна отображаться на всех телефонах одинаково. Но преимуществом разработки для платформы Android является его доступность.

Учитывая все факты, указанные выше, появилась идея создания мобильного приложения который позволит совершать выбор определенной техники основываясь на знаниях, которые будут заложены в нем для определения действительно нужной техники, нужной модели для конечного пользователя.



## 1 Анализ предметной области

Анализ предметной области – это очень значимый этап при создании программного продукта. Прежде чем приняться за разработку программы, стоит внимательно изучить и рассмотреть предметную область, вокруг которого и будет строиться программный продукт.

В наши дни, информационные технологии развиваются и охватывают все больше и больше сфер нашей повседневной деятельности. С развитием мобильных устройств и сферы онлайн торговли, на свет появились так мобильные приложения с возможностью покупать тот или иной товар, не выходя из дома, с экрана мобильного телефона.

Рассмотрим программный продукт для продажи компьютерной техники «Picker».

«Picker» - мобильное приложение, который частично заменяет консультанта в магазине компьютерной техники. Который предоставляет всю необходимую информацию о продукте в удобном для пользователя виде, при этом помогая определиться с выбором техники, не загружая его цифрами и характеристиками продукта.

На сегодняшний день в Казахстане нет мобильных интернет магазинов использующую алгоритм, позволяющий пользователям выбирать устройства, с помощью диалога.

### 1.1 Обзор аналогов

На рисунке 1.1 показано мобильное приложение одного из крупнейших магазинов электронной и бытовой техники «Клуб Sulpak».



Рисунок 1.1 – Мобильное приложение «Клуб Sulpak»

Приложение «Клуб Sulpak» – предназначено для получения информации о товарах, акциях, магазинах, наличии товара, о бонусах. Но к сожалению, в приложении нет возможности совершать покупки. Приложение является удобным для ведения бонусного счета, а также полностью может заменить физическую бонусную карту.

Также в качестве аналога можно рассмотреть сайт выше описанной компании «Sulpak», в которой полностью автоматизирован процесс покупки, доставки и оплаты. К недостаткам сайта можно отнести не ориентированность под мобильные устройства, что существенно усложняет пользование сайтом через мобильное устройство.

Pulser.kz – сайт компании, которая занимается продажей компьютеров на протяжении 24 лет (рисунок 1.2).

The image shows the logo for PULSER, which consists of the word "PULSER" in a bold, red, italicized sans-serif font. A registered trademark symbol (®) is located to the upper right of the letter "R".

Рисунок 1.2 – Сайт компьютерного магазина «Pulser»

Сайт ориентирован на покупателей, у которых должны быть хотя бы минимальные знания характеристик компьютера. Также компания является авторизованным дистрибьютором известных издателей устройств как «Kingston», «Intel», «Mustek», «SiliconPower».

На рисунке 1.3 показано мобильное приложение для android устройств под названием The Pc Constructor. Приложение позволяет собирать компьютер по комплектующим. В приложении пользователь должен выбрать каждую деталь своего компьютера, основываясь на цене, и в конце ему можно будет собрать компьютер нажав кнопку «Make Build».



Рисунок 1.3 – The Pc Constuctor

## **1.2 Преимущества и недостатки приложений для подбора техники**

В наши дни компьютерную технику может себе позволить каждый, но развитие техники идет так быстро, что с каждым днем, разновидности устройств набирает огромное количество видов. Таким образом, компьютеры можно классифицировать для различных сфер деятельности человека. Для упрощения выбора компьютера для пользователей из различных сфер деятельности, появилась идея разработки приложения, который позволит пользователям сделать выбор определенного устройства под свои нужды.

Приложение частично заменяет привычного консультанта в магазинах компьютерной техники. Нельзя исключать тот факт, что тот самый консультант может быть нацелен на увеличение продаж определенной модели и не заинтересован в целях и потребностях покупателя.

Основным преимуществом приложения является простота в использовании. Им может легко пользоваться люди как преклонного возраста, так и нынешнее поколение. К недостаткам можно отнести, что приложение не полностью заменяет эксперта и в некоторых аспектах, к сожалению, не способен помочь пользователю. Но это положение сводится к минимуму благодаря усовершенствованию и пополнением базы знаний и данных приложения. Также, приложение имеет модульную структуру, что позволит изменять разные части приложения не затрагивая другие, что упрощает дальнейшую поддержку приложения и выпуска обновлений.

Подробное сравнение приложения и живого консультанта указано в таблице 1.1.

Таблица 1.1 – Сравнение приложение и живого консультанта

Категория	Приложение для подбора техники	Живой консультант
Объективность	Алгоритм приложения устроен так, что он выдает результаты основываясь только на реальных данных.	Иногда, консультант может быть заинтересован в продаже товара, в целях увеличения общего числа продаж через его карту сотрудника.
Практичность	Пользоваться приложением для подбора техники вы можете, когда вам удобно, выбрать ту или иную технику, и заказать ее, не выходя из дома.	Для того чтобы проконсультироваться, вам придется сходить в магазин техники, и именно там уже разговаривать с консультантом.
Человеческий фактор	Знания в программу заносятся экспертами и специалистами, которые берут в основу многолетний опыт и статистические данные.	Существует вероятность того, что консультант вам попадется некомпетентный сотрудник, который не окажет вам должную помощь.

### 1.3 Постановка задачи

#### 1.3.1 Техническое задание

Цель работы – разработка мобильного приложения, который поможет сделать покупку компьютерной техники намного проще, за счет алгоритма подбора техники. Приложение рассчитано для пользователей всех возрастов, начиная с подростков, заканчивая пожилыми людьми. Также, приложение имеет простой, эргономичный дизайн – основанный на концепции Google Material Design, удобный функционал. Кроме того, программный продукт имеет возможность хранить в кэш-памяти ранее просмотренную информацию, что позволит сэкономить интернет трафик.

#### 1.3.2 Требования к мобильному приложению

Приложение должно удовлетворять следующим требованиям:

- простота и понятный интерфейс;
- поддержка старых и новых версии платформы Android;
- безопасное хранение данных пользователей;
- высокая производительность;
- низкий уровень потребления ресурсов.

## 2 Подбор инструментов для разработки мобильного приложения

Основными инструментами являются привычный набор веб-разработчика, которые делятся на следующие разделы:

- Графические инструменты
- Инструменты для разработки приложения
- Инструменты для тестирования приложения

### 2.1 Инструменты для работы с графикой

Графический редактор GNU Image Manipulation Program. GIMP – мощный графический редактор с лицензией GNU/GPL, что означает что программа является бесплатной и доступна для каждого. Функции программы вполне заменяют привычный для дизайнеров и фотографов Photoshop, который стоит немалых денег. Все изображения, которые использовались в разработке приложения обрабатывались и создавались в редакторе GIMP (рисунок 2.1).

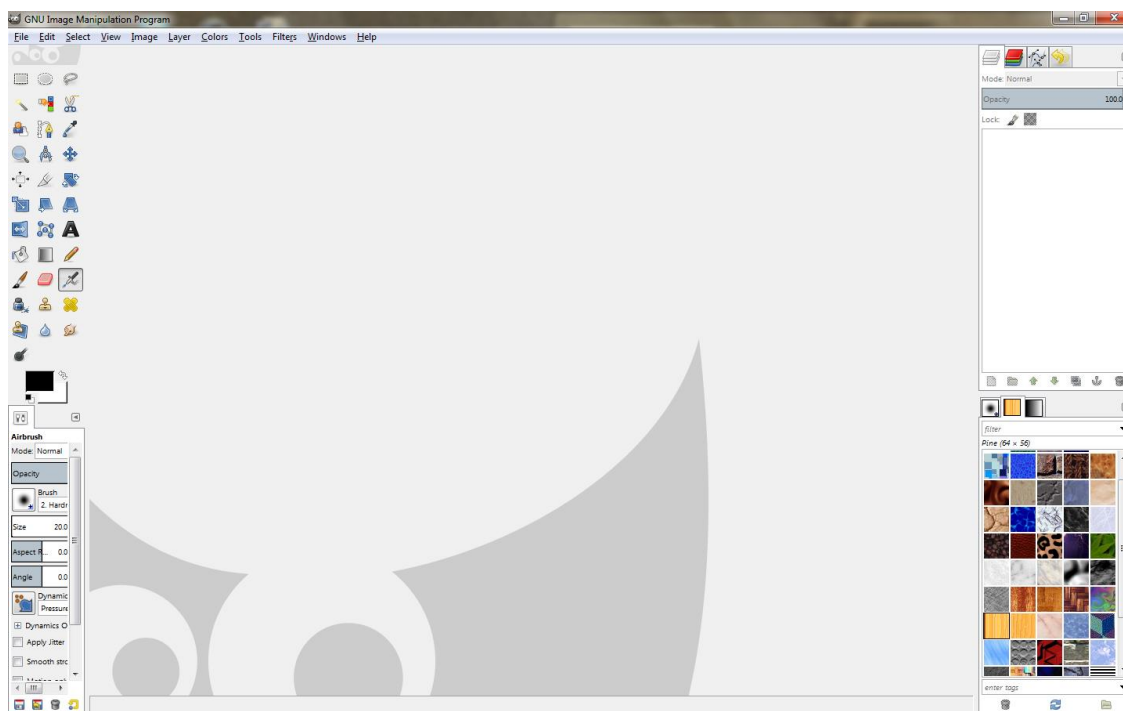


Рисунок 2.1 – Рабочее пространство программы Gimp

Редактор векторной графики. Разработка интерфейса программы иногда требует работы с векторной графикой. Например, создание иконок или изображений, которые независимо от разрешения экрана не должны терять качество. В этом мне помог также бесплатный редактор векторных изображений Inkscape (рисунок 2.2).

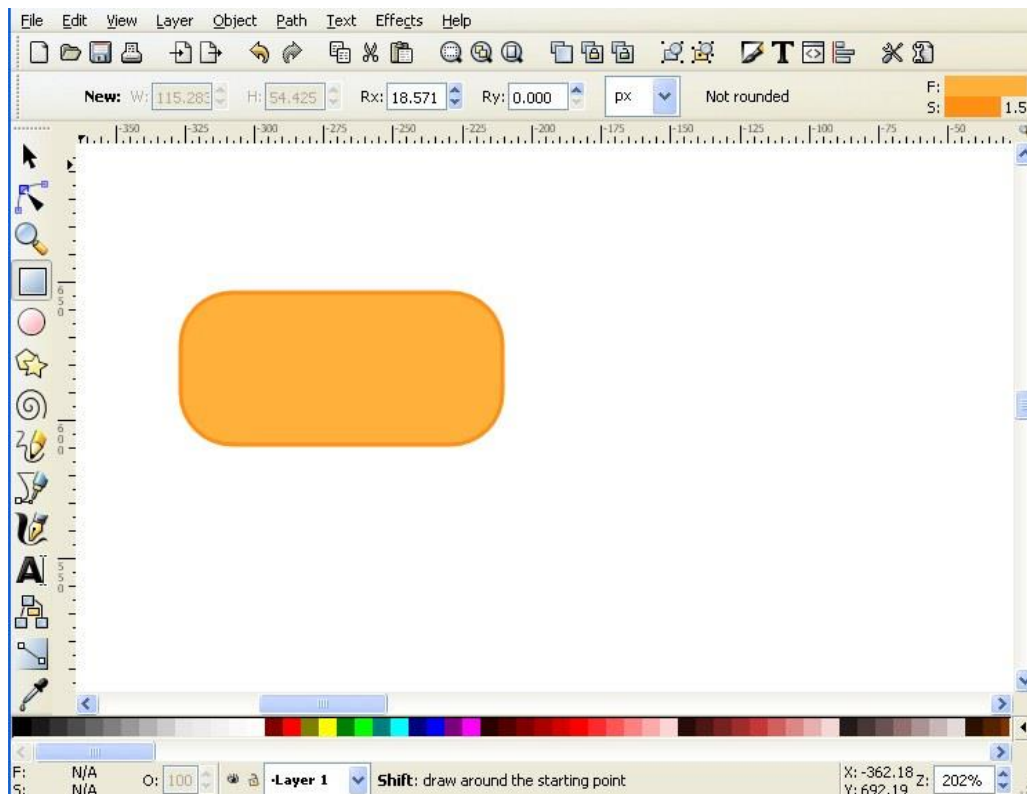


Рисунок 2.2 – Рабочее пространство программы Inkscape

## 2.2 Инструменты для разработки приложения

Текстовый редактор с подсветкой кода. При разработке мобильного приложения нужен был инструмент, позволяющий легко ориентироваться в коде, но также не был перегружен лишним функционалом в ущерб производительности. Учитывая все факторы был выбран очень популярный в последнее время редактор Sublime Text (рисунок 2.3).

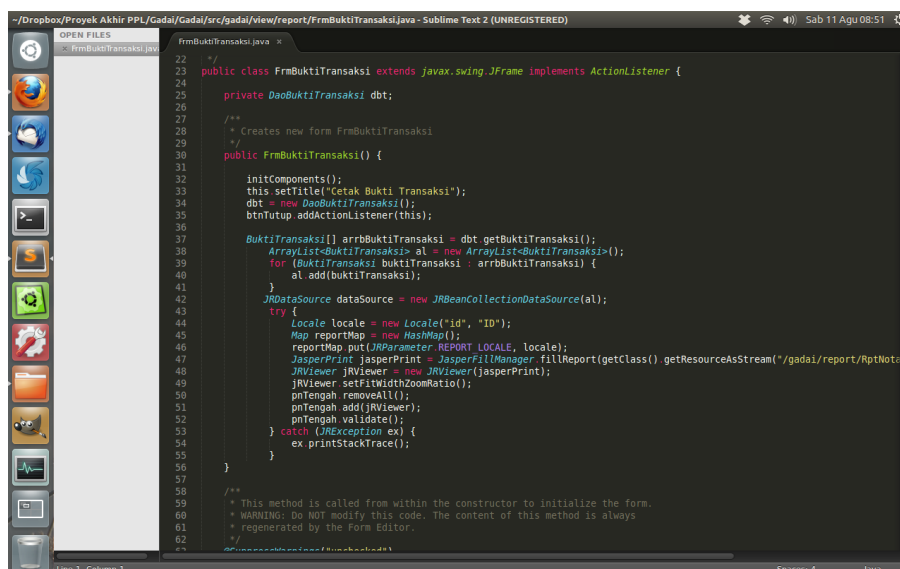


Рисунок 2.3 – Текстовый редактор Sublime Text 3

ConEmu – практичный и удобный терминал для Windows. В разработке, при создании проекта, сборе и компиляции приложения часто приходится использовать командные средства управления. Стандартный терминал Windows имеет ряд недостатков, перечислим основные из них:

- недружелюбный интерфейс
- без автодополнения ввода
- отсутствие многооконного режима

В качестве замены была выбран более удобный и функциональный терминал под названием ConEmu (рисунок 2.3).

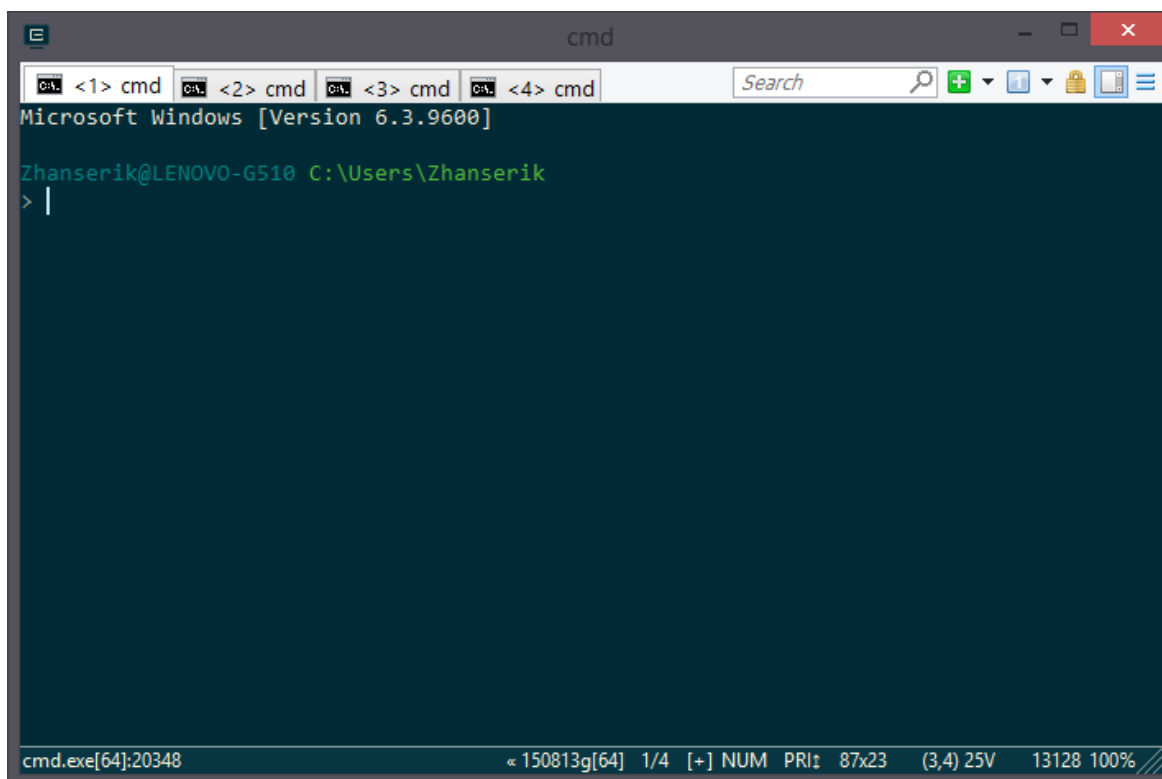


Рисунок 2.3 – Окно терминала ConEmu

### 2.3 Инструменты для тестирования и отладки приложения

Локальный веб-сервер. Локальный веб-сервер нужен для отладки и проверки HTML страниц в «полевых» условиях. Потому что именно эмуляция сервера максимально приближает процесс тестирования к реальному процессу работы приложения. В качестве локального веб-сервера был выбран XAMPP (X Apache MySQL PHP Perl).

Браузер в качестве среды разработки. При работе с веб-технологиями зачастую именно браузер является ключевой программой, т.к. именно браузер осуществляет показ содержимого веб-интерфейса. Для тестирования интерфейса приложения был выбран браузер Mozilla Firefox, из-за удобной функции отображения веб части в мобильном режиме.

### 3 Технологическое обеспечение

Существует 2 основных технологии разработки мобильных приложений:

- нативные – разрабатываются для каждой платформы отдельно
- гибридные – разрабатывается веб-сайт который перекомпилируется в мобильное приложение, также является кроссплатформенным, что позволяет разрабатывать приложения для разных платформ одновременно. Учитывая преимущества и ситуацию выбор пал на гибридную технологию разработки мобильных приложений.

#### 3.1 Платформа для разработки гибридных приложений Framework7

Framework7 – Фреймворк для разработки гибридных мобильных приложений на HTML5. Имеются базовые элементы интерфейсы из «коробки».

##### 3.1.1 Основные функции

Модифицированный DOM (от англ. Document Object Model – «объектная модель документа»). Framework7 не использует сторонние библиотеки, даже для манипулирования DOM. Он имеет свой собственный DOM7 – DOM библиотеку, которая использует самые передовые и высокоэффективные методы для манипулирования DOM. Он прост в использовании, потому что синтаксис очень похож на привычный всем синтаксис JQuery библиотеки. Все доступные методы освещены в таблице 3.1.

Пример использования:

```
$$('.something').on('click', function (e) {  
  $$(this).addClass('hello').attr('title','world').insertAfter('.somethin-else');  
});
```

Таблица 3.1 – Доступные методы DOM7

Классы	
.addClass(className)	Добавить класс элементов
.removeClass(className)	Удалить указанный класс
.hasClass(className)	Определить, являются ли какие-либо из соответствующих элементов присвоены данному классу
.toggleClass(className)	Удалить (если класс существует) или добавить (если нет)



Таблица 3.2 – Управление данными

Хранилище данных	
.data(key, value)	Хранить произвольные данные, связанные с совпадающими элементами
.data(key)	Возвращает значение в указанном хранилище данных для первого элемента в коллекции
.removeData(key)	Удалить указанные данные

### 3.1.2 Компоненты навигации

Маршрутизатор JavaScript API. Навигация в Framework7 очень гибкая. Это дает много возможностей для обработки следующих страниц:

- Ajax страницы – загружать страницы из разных файлов;
- Динамические страницы с помощью JavaScript API;
- Вы можете использовать встроенные страницы уже в DOM;

Представления (Views) должны быть подключены и инициализированы для добавления навигации в любую часть приложения.

Таблица 3.3 – Методы навигации

Параметры	Тип данных	Описание
url	string	Адрес страницы для загрузки
content	String или HTMLElement	Содержимое динамической загрузки страницы
pagename	string	Название страницы (данные страниц) страницы для загрузки. Только для встроенных страниц (DOM Cache)

### 3.1.3 Компоненты пользовательского интерфейса

Framework7 имеет в своем наборе основные элементы пользовательского интерфейса такие как: кнопки, меню, иконки, списки, модальные окна, боковые панели, карточки и т.д. Это облегчает разработчику жизнь, т.к. ему не приходится строить элементы интерфейса с нуля. Разработчик может на основе примеров создать свои элементы, или использовать уже существующие решения фреймворка. Стилистика компонентов пользовательского интерфейса выполнена в двух вариантах: в стиле IOS7 для смартфонов Apple и в стиле Material Design для смартфонов под управлением операционной системой Android. На рисунке 3.1 и 3.2 показан шаблон интерфейса приложения.

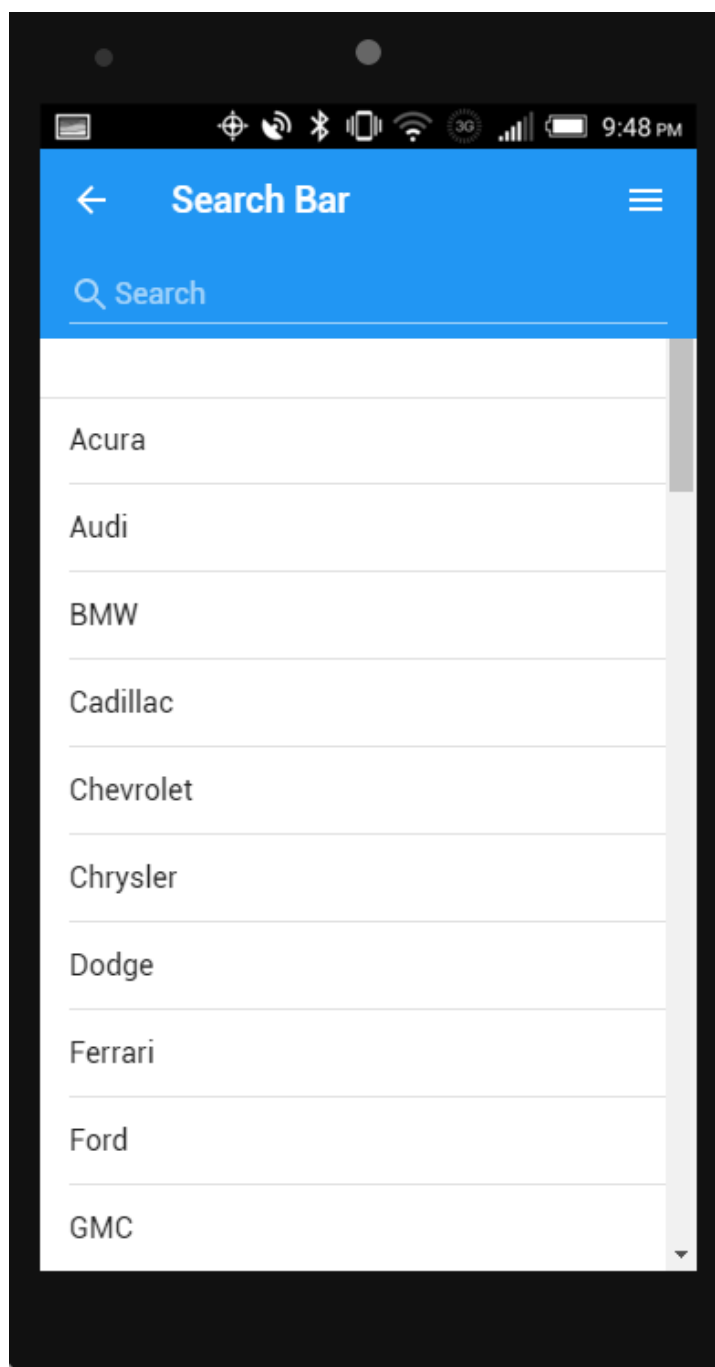


Рисунок 3.1 – Пример шаблона приложения с поиском

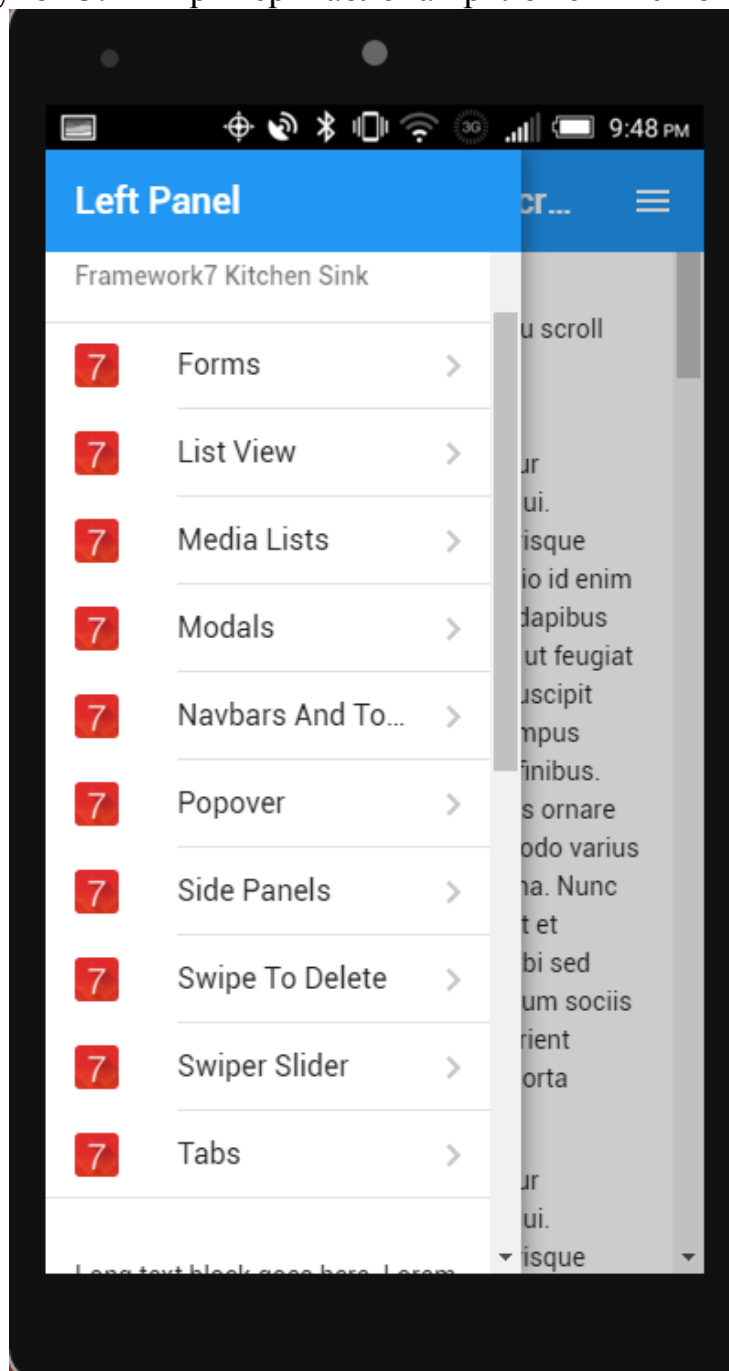


Рисунок 3.2 – Пример шаблона с боковым меню

### 3.1.4 Утилиты

#### API (интерфейс прикладного программирования) устройства

Framework7 поставляется с библиотекой обнаружения устройств. После инициализации приложения вы можете получить доступ к `myApp.device` объекту, который содержит полезную информацию об устройстве и платформе показанные в таблице 3.4.

Таблица 3.4 – Объект myApp.device

Объекты	Описание
myApp.device.os	строка. Содержит "Android" (для Android), "iOS" (для iOS)
myApp.device.osVersion	строка. Содержит версию операционной системы.
myApp.device.android	логическое значение. Содержит истинно для Android устройства, в противном случае ложно.
myApp.device.pixelRatio	цифра. Возвращает соотношение пикселей экрана устройства.
myApp.device.webView	логическое значение. Содержит истинно, если приложение работает в UIWebView

### 3.1.5 Сравнение Framework7 с похожими платформами разработки

Сравним выбранный фреймворк с похожими фреймворками других разработчиков, а именно с уже наиболее известными Ionic Framework и PhoneGap. Сравнение показано в таблице 3.5.[4]

Таблица 3.5 – Сравнение платформ гибридной разработки

Пункты	Framework7	Ionic	Phonegap
Реализованные элементы интерфейса для Android	+	-	-
Высокая производительность	+	+	+
Богатый функционал	+	+	-

## **3.2 Языки web-программирования**

### **3.2.1 HTML5**

Язык HTML5 делает акцент на упрощении разметки, необходимой для создания соответствующих W3C-стандартам страниц и объединения всего требуемого CSS и JavaScript кода, а также файлов изображений. Если вести речь о пользователях, которые, вероятно просматривают наши страницы, подключаясь к Интернету по каналу с невысокой скоростью передачи информации, а также об основной цели наших адаптивных веб-дизайнов, то необходимо, чтобы сайт не просто адаптировался к их более ограниченным областям просмотра, но и загружался настолько быстро, насколько это возможно [1].

HTML5 несет в себе дополнительные преимущества и функции по сравнению с предыдущей версией (HTML 4.01). Веб-разработчиков клиентских приложений, скорее всего, заинтересуют новые семантические HTML5-элементы, которые обеспечивают более осмысленный код для поисковых механизмов. HTML5 также позволяет организовать для пользователей обратную связь на основе базового взаимодействия с сайтами вроде отправки данных форм и т.п., зачастую устраняя необходимость в JavaScript-обработке форм, более требовательных к вычислительным ресурсам. [1].

### **3.2.2 CSS**

Каскадные таблицы стилей (CSS) это язык для определения, как документ должен представляться пользователям. Эти документы написаны на языке разметки, такие как HTML.

Документ представляет собой набор информации, которая построена с использованием языка разметки.

Представляя документ пользователю – означает преобразование его в удобной форме для вашей аудитории. Браузеры, как Firefox, Chrome или Internet Explorer, предназначены для представления документов визуально, например, на экране компьютера, проектор или принтер.

Использование CSS определяет стили для документов, в том числе дизайн, верстка и вариаций дисплея для различных устройств и размеров экрана. Разработчик может разместить свой CSS в <HEAD> документа с встроенной таблицей стилей, или приложите отдельный файл, который определяет ваши стили с внешней таблицы стилей. Чтобы связать внешнюю таблицу стилей к документу, нужно просто добавить ссылку на таблицу стилей в <HEAD> документа.

Внешняя таблица стилей имеет много преимуществ. Сохранение стилей отдельно от содержания HTML:

-Помогает избежать дублирования;

- Делает обслуживание проще;
- Позволяет сделать сайт масштабе изменения в одном месте.

Как работает CSS. Когда браузер отображает документ, он должен сочетать в себе содержание документа с информацией стиля. Он обрабатывает документ в два этапа:

Браузер преобразует язык разметки и CSS в DOM (Document Object Model). DOM представляет документ в памяти компьютера. Он сочетает в себе содержание документа с его стилем.

Браузер отображает содержимое DOM. Язык разметки использует элементы для определения структуры документа. Вы обозначаете элемент, используя теги, которые представляют собой строки, начинающиеся с '<' и заканчивая '>'. Большинство элементов имеют парные теги, начальный тег и конечный тег. Для начального тега, поместите имя элемента между '<' и '>'. Для конечного тега, поместите '/' после '<' и перед именем элемента.

В зависимости от языка разметки, некоторые элементы имеют только начальный тег, или один тег, где '/' приходит после того, как имя элемента. Элемент может также быть контейнером и включать в себя другие элементы, между его начальным тегом и конечным тегом. Только не забудьте всегда закрывать теги внутри контейнера.

DOM имеет древовидную структуру. Каждый элемент, атрибут и запуск текста на языке разметки становится узлом в древовидной структуре. Узлы определяются их связи с другими узлами DOM. Некоторые элементы являются родителями дочерних узлов, и дочерние узлы имеют братьев и сестер.

Понимание DOM позволяет создавать, отлаживать и поддерживать ваш CSS, так как DOM, где ваш CSS и содержание документа встречаются.

#### Каскадирование и наследование

Окончательный стиль для элемента может быть указан во многих различных местах, которые могут взаимодействуют сложным образом. Это сложное взаимодействие делает CSS мощным, но он также может сделать его запутанным и трудным для отладки.

Три основных источника информации о стиле образуют каскад. Они есть:

- Стили браузера по умолчанию для языка разметки.
- Стили, указанные пользователем, который читает документ.
- Стили связан с документом его автором. Они могут быть определены в трех местах:
  - Во внешнем файле: это руководство в первую очередь обсуждает этот метод определения стилей.
  - В определении в начале документа: использовать этот метод только для стилей, которые используются только на этой странице.

-На конкретный элемент в теле документа: это наименее ремонтпригодны метод, но может быть использован для тестирования.

Стиль пользователя изменяет стиль браузера по умолчанию. Стиль автора документа затем изменяет стиль еще немного. В этом учебнике вы являетесь автором вашего образца документа, и вы работаете только с автором таблицы стилей.

### **3.2.3 Javascript**

JavaScript® (часто сокращается до JS) представляет собой легкий, интерпретировать, объектно-ориентированный язык с функциями первого класса, и лучше всего известен как язык сценариев для веб-страниц, но он используется во многих не-браузера средах, а также. Это прототип на основе, несколько парадигм язык сценариев, который является динамическим, а также поддерживает объектно-ориентированный, императив, и функциональные стили программирования.

JavaScript работает на клиентской стороне полотна, которая может быть использована для разработки / программы, как веб-страницы ведут себя по наступлении события. JavaScript является простой в освоении, а также мощный язык сценариев, широко используемый для управления поведением веб-страницы.

Вопреки распространенному заблуждению, JavaScript не "Interpreted Java". В двух словах, JavaScript представляет собой динамичный язык сценариев поддержки прототипа на основе строительства объекта. Базовый синтаксис намеренно похож на как Java и C ++, чтобы уменьшить количество новых понятий, необходимых для изучения языка. Языковые конструкции, такие как, если заявления, и в то время для петель, а также переключатель и Try ... Catch блоки функционируют так же, как и на этих языках (или почти так).

JavaScript может функционировать как процедурного и объектно-ориентированного языка. Объекты создаются программно в JavaScript, путем присоединения методы и свойства, в противном случае пустые объекты во время выполнения, в отличие от синтаксических определений классов распространенных в скомпилированных языках, как C ++ и Java. После того, как объект был построен он может быть использован в качестве плана (или прототипа) для создания подобных объектов.

Динамические возможности JavaScript включают в себя во время выполнения объекта строительства, списки переменных параметров, функция переменных, динамического создания сценария (через Eval), объект интроспекции (через для ... в), а также восстановления исходного кода

(программы JavaScript могут декомпилировать функции органов обратно в исходный текст).

Для более подробного обсуждения программирования JavaScript следовать JavaScript ресурсов ссылки ниже.

Какие существуют реализации JavaScript доступны?

Проект Mozilla предоставляет две реализации JavaScript. Первый в истории JavaScript был создан Brendan Eich в Netscape, и с тех пор были обновлены, чтобы соответствовать ECMA-262 Edition, 5 и более поздних версий. Этот движатель, под кодовым названием SpiderMonkey, реализуется в C / C ++. Движитель Rhino, созданный в первую очередь Norris Boyd (также в Netscape) является реализация JavaScript написана на Java. Как SpiderMonkey, Rhino является ECMA-262 Выпуск 5 совместимый. Несколько крупных во время выполнения оптимизации, такие как TraceMonkey (Firefox 3.5), JägerMonkey (Firefox 4) и IonMonkey были добавлены к движателю SpiderMonkey JavaScript с течением времени. Работа всегда продолжается, чтобы улучшить производительность выполнения JavaScript. Кроме указанных выше реализаций, существуют и другие популярные движатели JavaScript, такие как: -V8 от Google, который используется в браузере Google Chrome и последних версиях браузера Опера. Это также движатель, используемый Node.js. JavaScriptCore (SquirrelFish / Nitro), используемый в некоторых WebKit браузерах, таких как Apple Safari.

Саракан в старых версиях Опера. Движитель чакра используется в Internet Explorer (хотя язык он реализует официально называется "JScript" для того, чтобы избежать проблем товарного знака). Каждый из движателей JavaScript Mozilla, разоблачить публичный API, который разработчики приложений могут использовать для интеграции JavaScript в свое программное обеспечение. До сих пор наиболее распространенной средой хоста для JavaScript является веб-браузеры. Веб-браузеры обычно используют публичный API для создания хост-объекты, отвечающие за отражая DOM в JavaScript.

Другой распространенный приложение для JavaScript является как (Web) на стороне сервера язык сценариев. Веб-сервер JavaScript подвергнет объекты, представляющие хост запрос HTTP и объекты ответа, которые затем можно было бы манипулировать программой JavaScript для динамического создания веб-страниц. Node.js является популярным примером этого. 3.3 Разработка приложений с помощью Apache Cordova

### **3.3 Разработка приложений с помощью Apache Cordova**

#### **3.3.1 Обзор архитектуры**

Приложение, созданное посредством Cordova, имеет большую структуру, которые показаны на рисунке 3.3.



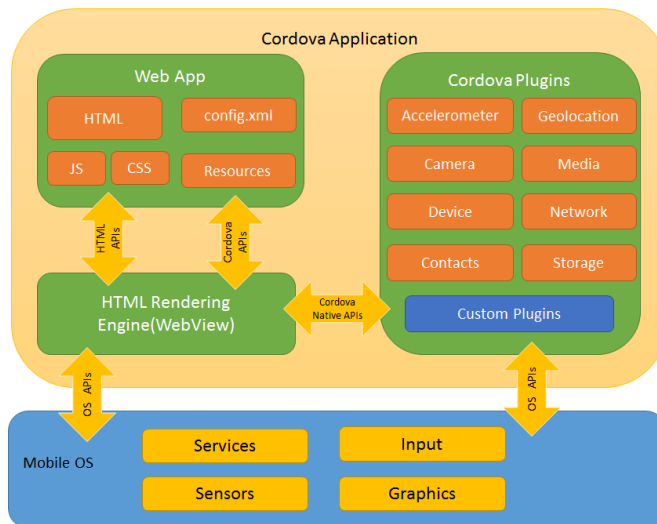


Рисунок 3.3 – Архитектура Apache Cordova

### 3.3.2 Поддерживаемые платформы устройств

Ниже на рисунке 3.4 приведен набор инструментов разработки и устройства интерфейсы API, доступные для каждой мобильной платформы. Устройство API-интерфейсы, перечисленные здесь предоставляются основные плагины, дополнительные интерфейсы API доступны через сторонние плагины. Заголовки столбцов отображаются имена стенографию CLI.

	amazon-fireos	android	blackberry10	Firefox OS	iOS	Ubuntu	WP8 (Windows Phone 8)	Windows (8.0, 8.1, 10, Телефон 8.1)	Tizen
Cordova CLI	✓ Mac, Windows, Linux	✓ Mac, Windows, Linux	✓ Mac, Windows	✓ Mac, Windows, Linux	✓ Mac	✓ Ubuntu	✓ Windows	✓	X
Встроенный WebView	✓ (см. подробности)	✓ (см. подробности)	X	X	✓ (см. подробности)	✓	X	X	X
Подключаемый модуль Интерфейс	✓ (см. подробности)	✓ (см. подробности)	✓ (см. подробности)	X	✓ (см. подробности)	✓	✓ (см. подробности)	✓	X

Рисунок 3.4 – Поддерживаемые платформы

WebView. Cordova поддержкой WebView может обеспечить приложение с помощью всего пользовательского интерфейса. На некоторых платформах, он также может быть компонентом которое смешивает WebView с носителями компонентов приложения.

Веб-приложение. Это та часть, где находится ваш код. Само приложение реализовано в виде веб-страницы, по умолчанию в локальном файле с именем index.html, который ссылается на CSS, JavaScript, изображения, мультимедийные файлы и другие ресурсы, которые необходимы

для его запуска. Приложение выполняется в WebView в пределах родной оболочки приложения, которое вы распространяете в магазинах приложений.

Этот контейнер имеет очень важный файл - файл config.xml, который предоставляет информацию о приложении и задает параметры, влияющие как она работает, например, отвечает ли она ориентации сдвигов.

Плагины. Плагины являются неотъемлемой частью экосистемы Cordova. Они обеспечивают интерфейс для Cordova и нативных компонентов для общения друг с другом и привязки к стандартным API устройств. Это позволяет ссылаться на машинный код с помощью JavaScript.

Проект Apache Cordova поддерживает набор плагинов называется ядро плагинов. Эти основные модули обеспечивают приложению доступ к возможностям устройств, таких как батареи, камера, контакты и т.д.

В дополнение к основным плагинов, есть несколько сторонних плагинов, которые обеспечивают дополнительные привязки к функциям, не обязательно доступных на всех платформах. Вы можете осуществлять поиск Cordova плагинов, используя поисковый плагин или NPM. Вы также можете разработать свои собственные плагины, как описано в руководстве по разработке плагинов. Плагины могут быть необходимы, например, для обмена данными между Cordova и пользовательских родных компонентов.

Примечание: При создании проекта Cordova он не имеет каких-либо плагинов, присутствующих. Это новое поведение по умолчанию. Любые плагины вы желаете, даже основные плагины, должны быть явно добавлены.

Cordova не предоставляет каких-либо элементов интерфейса пользователя или MVC основы. Кордова обеспечивает только среду выполнения, в которых те могут выполнять. Если вы хотите использовать виджеты UI и / или рамки MVC, вам нужно будет выбрать, и включить их в вашем приложении.

Пути развития. Cordova дает вам два основных рабочих процессов для создания мобильных приложений. В то время как вы часто можете использовать любой рабочий процесс для выполнения задачи, каждый из них имеет свои преимущества:

Кроссплатформенный (CLI) рабочий процесс: Используйте этот рабочий процесс, если вы хотите, чтобы ваше приложение для запуска на столько различных мобильных операционных систем, как это возможно, с небольшим количеством необходимости разработки конкретной платформы. Этот рабочий процесс сосредотачивается вокруг CLI Cordova. CLI является инструментом высокого уровня, который позволяет создавать проекты для многих платформ сразу, абстрагируясь от большую часть функциональности скриптов, более низкого уровня. Копии CLI общий набор веб-ресурсов в подкаталоги для каждой мобильной платформы, делает все необходимые изменения конфигурации для каждого, запускает сценарии сборки для создания бинарных файлов приложений. CLI также обеспечивает общий интерфейс для применения плагинов для вашего приложения. Для начала выполните действия, описанные в разделе создать свое первое приложение

руководства. Если у вас есть потребность в рабочий процесс платформы по центру, рекомендуется рабочий процесс кроссплатформенным.

Платформа в центре рабочего процесса: используйте этот рабочий процесс, если вы хотите, чтобы сосредоточиться на создании приложения для одной платформы и должны быть в состоянии изменить его на более низком уровне. Вы должны использовать этот подход, например, если вы хотите, чтобы ваше приложение, чтобы смешать пользовательские собственные компоненты с веб-компонентов на основе Cordova, как описано в Встраивание WebViews. Как правило, использовать этот рабочий процесс, если вам нужно изменить проект в SDK. Этот рабочий процесс опирается на набор скриптов, более низкого уровня, которые специально для каждой поддерживаемой платформы, а также отдельная Plugman утилита, которая позволяет применять плагины. В то время как вы можете использовать этот рабочий процесс для создания кроссплатформенных приложений, как правило, сложнее, потому что отсутствие инструмента более высокого уровня означает, что отдельные циклы сборки и плагинов модификаций для каждой платформы.

## 4 Дизайн приложения в стиле Material Design

### 4.1 Обзор Material Design

В 2014 году на конференции I/O, Google представила новый подход к проектированию системы, которая называется Material Design. Новая конструкция позволяет системе создать единый пользовательский интерфейс на всех экранах: настольный компьютер, смартфоны, планшеты, часы, телевизоры и автомобили. Для-всех приложений Android Material Design является эволюцией визуального языка и принципов дизайна Holo. является гибким во многих отношениях, и была создана для других разработчиков, а Google это только первый пользователь. Material Design устанавливает разумные пределы, но не слишком ограничительный характер.

### 4.2 Основные принципы Material Design

Material Design основан на четырех основных принципах:

**Тактильные поверхности.** Материал интерфейса состоит из материальных слоев так называемой "цифровой бумаги". Эти слои расположены на разной высоте и тени друг от друга, что помогает пользователям лучше понять анатомию интерфейса и взаимодействия с принципом.

**Дизайн полиграфии.** Если мы примем во внимание слои кусков "цифровой бумаги", в связи с "цифровыми чернилами" (все, что появляется на "цифровой бумаге"), используя традиционный подход к области графического дизайна, например, журнал и плакат.

**Значительные анимации.** В реальном мире вещи не приходят из ниоткуда и не исчезают в никуда. Таким образом, в области промышленного дизайна, мы все время думаем о том, как бы использовать анимацию в слоях и в "цифровых чернил", чтобы дать пользователям рекомендации по интерфейсу.

**Адаптивный дизайн.** Это, как мы применяем предыдущие три концепции на различных устройствах с различными разрешениями и размерами экранов.



Рисунок 4.1 – Основные 4 принципа Material Design

Тактильная поверхность. Давайте начнем с тактильных поверхностей. Это куски "цифровой бумаги", которые, в отличие от обычной бумаги имеют резервные мощности – а именно возможность растянуть и изменить их форму. Остальные действия выполняются в полном соответствии с законами физики (рисунок 4.2).

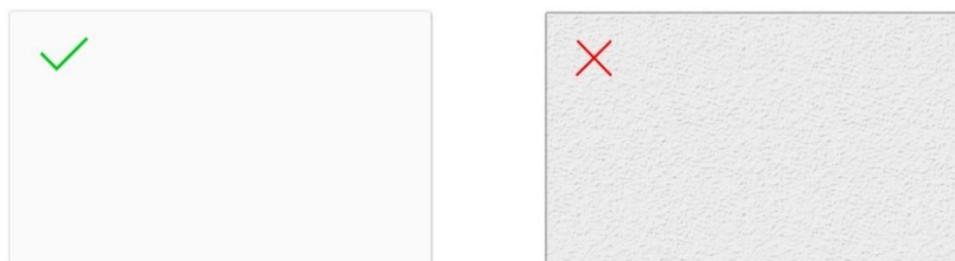


Рисунок 4.2 – Поверхность в Material Design

Глубина. В традиционном «плоском дизайне» избегают таких теней и все проявления масштаба, но они выполняют важную функцию, связанную со структурой и иерархией элементов на экране. Например, если элемент поднимется большее расстояние, и его тень становится больше. Это увеличение глубины помогает привлечь внимание пользователя, и сделать это изящно (Рисунок 4.3).

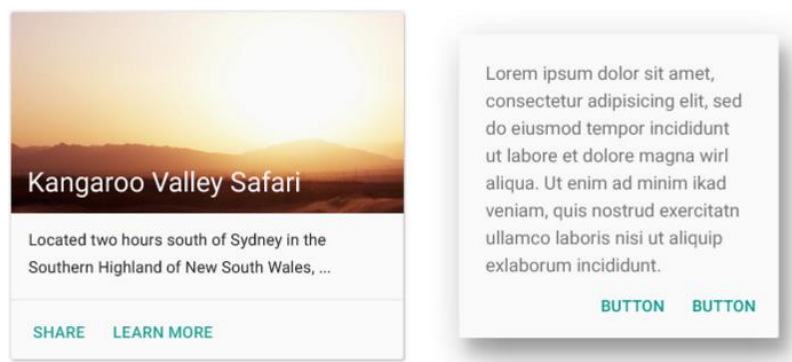


Рисунок 4.3 – Пример использования глубины

Дизайн полиграфии. Если поверхность в области промышленного дизайна называется «цифровой бумагой», то есть все, что на нем - текст, изображения, значки – «цифровые чернила». Material Design использует принципы классического дизайна в дизайне интерфейса печати (рисунок 4.4)

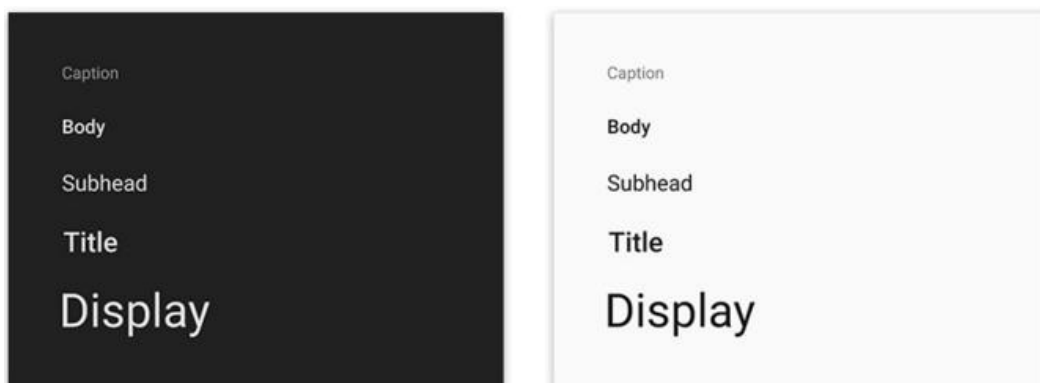


Рисунок 4.4 – Полиграфия

Значимые анимации. В реальном мире объекты не могут просто появляться из ниоткуда и исчезают в никуда. Это создало бы путаницу и ставило бы людей в тупик. Таким образом, значимые анимации Material Design используется для отображения того, что только что произошло.

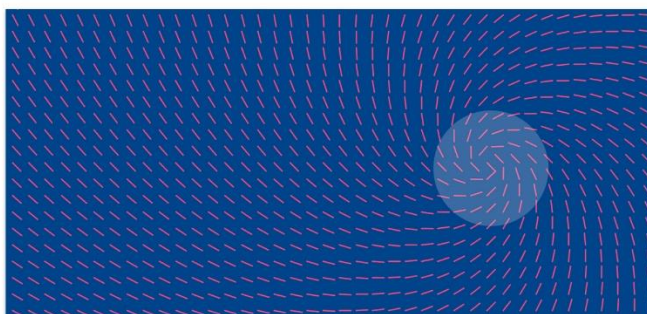


Рисунок 4.5 – Пример движения анимации

Адаптивный дизайн. Последняя, очень важная часть Material Design — концепция адаптивного дизайна. Вот как мы можем использовать все три принципа, описанные выше на различных устройствах и экранах.



Рисунок 4.6 – Пример адаптивного дизайна

Самый распространенный приём — отслеживать изменение размеров ширины экрана для уменьшения основного контента.

## **5 Практическая часть выполнения дипломной работы**

### **5.1 Порядок выполнения работы**

Работы выполняется самостоятельно и состоит из следующих шагов:

- изучение методических указаний по выполнению дипломной работы
- разработка мобильного приложения с помощью веб-технологии
- основные требования к мобильному приложению:
- информация о товарах должна размещаться максимально удобно для пользователя, не нагружая его лишней информацией.
- при разработке приложения важно соблюдать 3 основных правила:
- высокая скорость нахождения информации
- удобство при восприятии информации
- надежность информации

### **5.2 Установка необходимых инструментов**

Установка Apache Cordova. Для дальнейшей работы и компиляции приложения необходимо установить Cordova фреймворк. Для его установки требуется сначала установить NODE.JS. Затем выполнить в командной строке команды:

```
npm install -g cordova
```

Установка Framework7. Для установки Framework7 не требуется скачивать установщик. Стоит лишь скачать необходимые пакеты с репозитория разработчика на ресурсе GitHub. Потом распаковать где будет удобно пользоваться им в дальнейшем.

### **5.3 Реализация приложения**

Для начала разрабатывается алгоритм приложения, по которому будет строиться дальнейшие действия (рисунок 5.1). Именно следуя алгоритму выстраивается вся логика приложения, определяются основные моменты реализации, выбор решения для какого-либо случая. Для наглядного построения алгоритма используются блок-схемы. распространенный тип схем (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности.

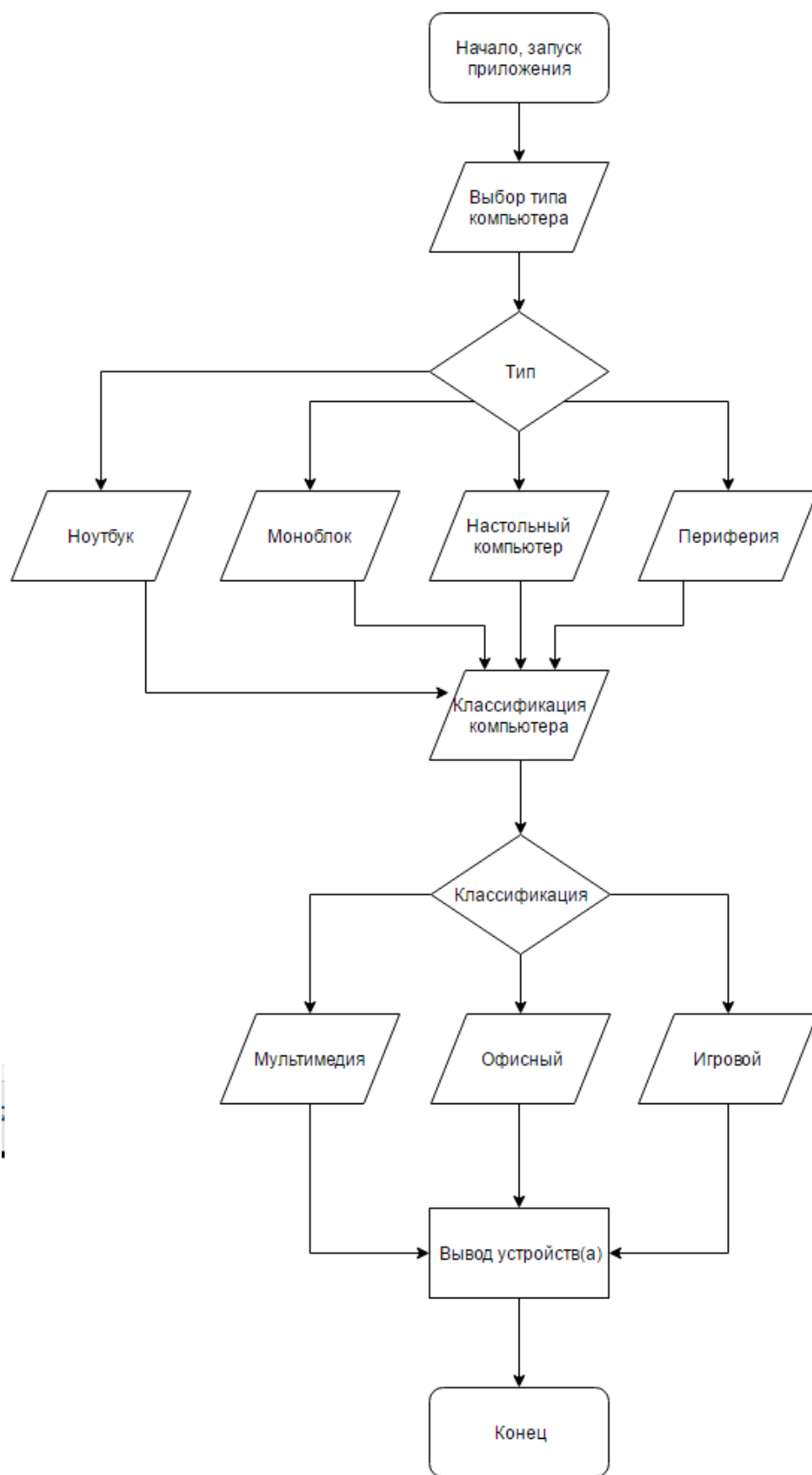
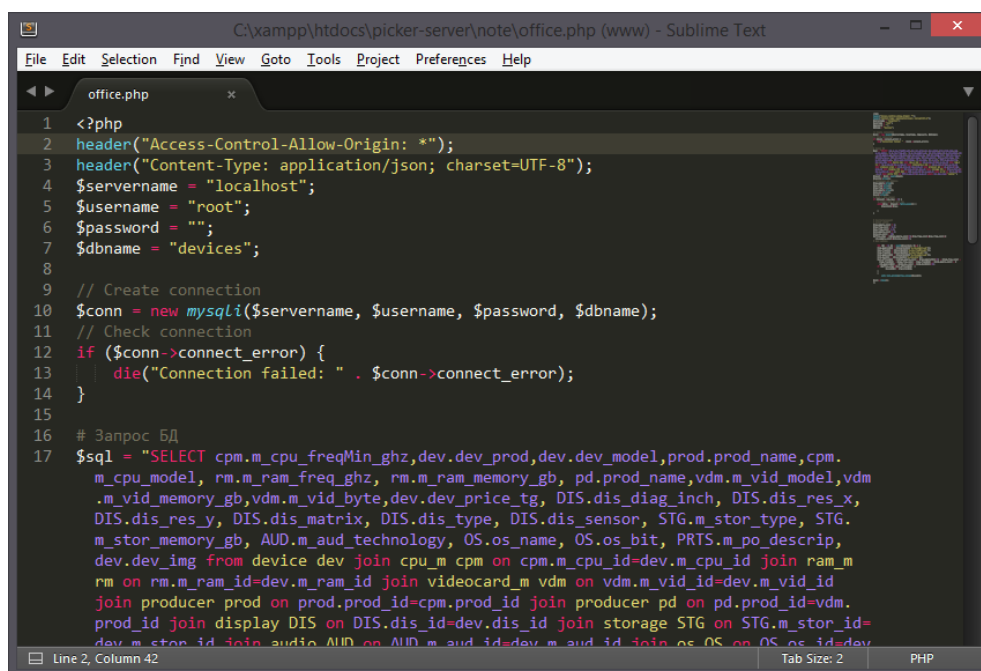


Рисунок 5.1 – Блок-схема работы приложения



Следующим этапом разработки было создание скриптов на серверном языке PHP, которые делают выборку компьютеров из базы данных основываясь на онтологии правил базы знаний.



```
1 <?php
2 header("Access-Control-Allow-Origin: *");
3 header("Content-Type: application/json; charset=UTF-8");
4 $servername = "localhost";
5 $username = "root";
6 $password = "";
7 $dbname = "devices";
8
9 // Create connection
10 $conn = new mysqli($servername, $username, $password, $dbname);
11 // Check connection
12 if ($conn->connect_error) {
13     die("Connection failed: " . $conn->connect_error);
14 }
15
16 # Запрос БД
17 $sql = "SELECT cpm.m_cpu_freqMin_ghz, dev.dev_prod, dev.dev_model, prod.prod_name, cpm.
    m_cpu_model, rm.m_ram_freq_ghz, rm.m_ram_memory_gb, pd.prod_name, vdm.m_vid_model, vdm.
    m_vid_memory_gb, vdm.m_vid_byte, dev.dev_price_tg, DIS.dis_diag_inch, DIS.dis_res_x,
    DIS.dis_res_y, DIS.dis_matrix, DIS.dis_type, DIS.dis_sensor, STG.m_stor_type, STG.
    m_stor_memory_gb, AUD.m_aud_technology, OS.os_name, OS.os_bit, PRTS.m_po_descrip,
    dev.dev_img from device dev join cpu_m cpm on cpm.m_cpu_id=dev.m_cpu_id join ram_m
    rm on rm.m_ram_id=dev.m_ram_id join videocard_m vdm on vdm.m_vid_id=dev.m_vid_id
    join producer prod on prod.prod_id=cpm.prod_id join producer pd on pd.prod_id=vdm.
    prod_id join display DIS on DIS.dis_id=dev.dis_id join storage STG on STG.m_stor_id=
    dev.m_stor_id join audio AUD on AUD.m_aud_id=dev.m_aud_id join os OS on OS.os_id=dev
```

Рисунок 5.2 – Отрывок кода PHP скрипта

Далее, была составлена структура для дерева категорий компьютеров. В качестве примера была развернута ветки выбора ноутбуков на рисунке 5.3.

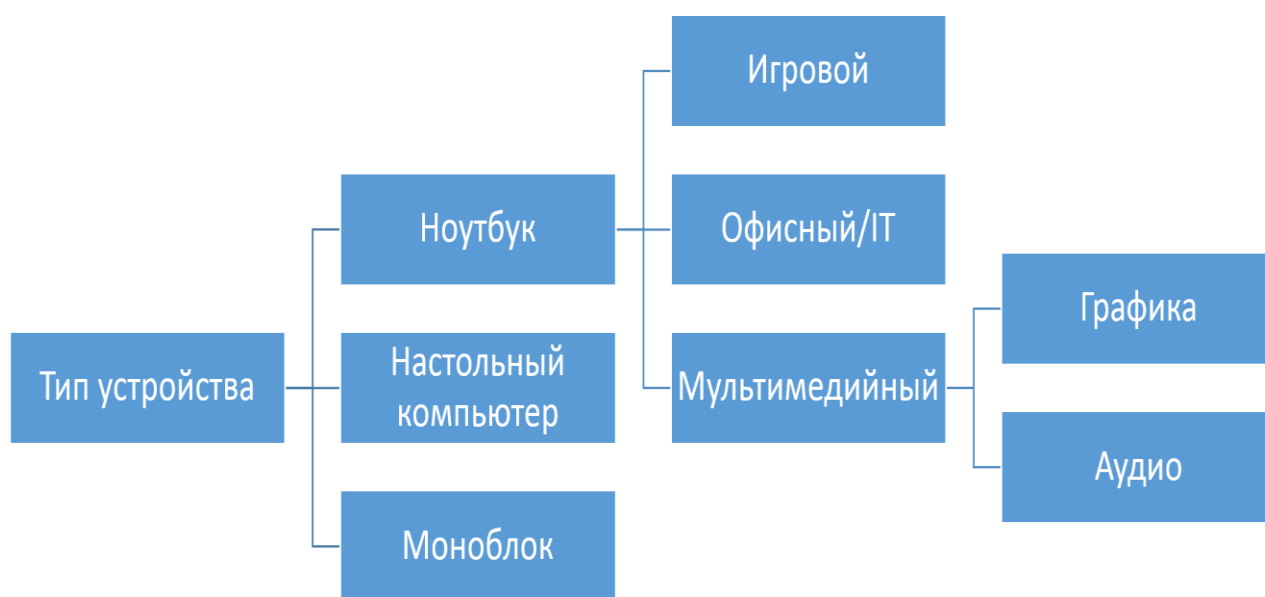


Рисунок 5.3 – Структура категорий

Разработка приложения производится в Sublime Text, где редактируется код графического интерфейса, который использует языки разметки HTML5, CSS и язык программирования JavaScript.

Главное окно приложение имеет 3 кнопки для перехода на категории устройств.

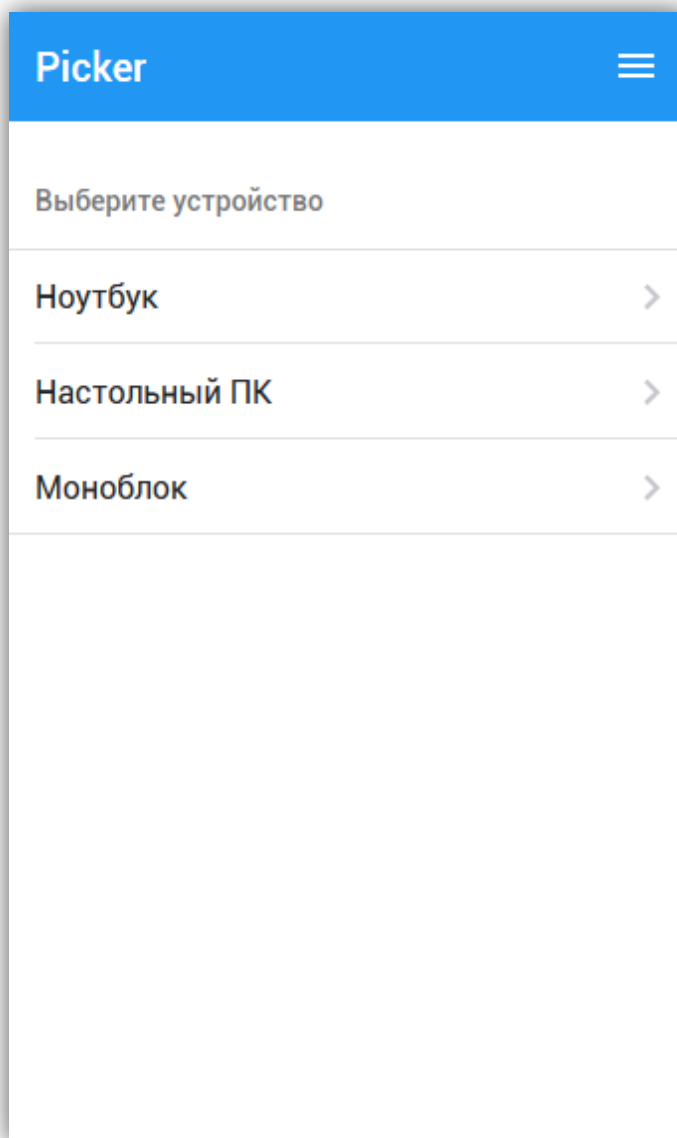


Рисунок 5.5 – Главное окно приложения

Дальше если перейти в ветку ноутбуков, то мы увидим следующий раздел в виде категорий по деятельности пользователя, а именно:

- для работы;
- для игр;
- мультимедиа.

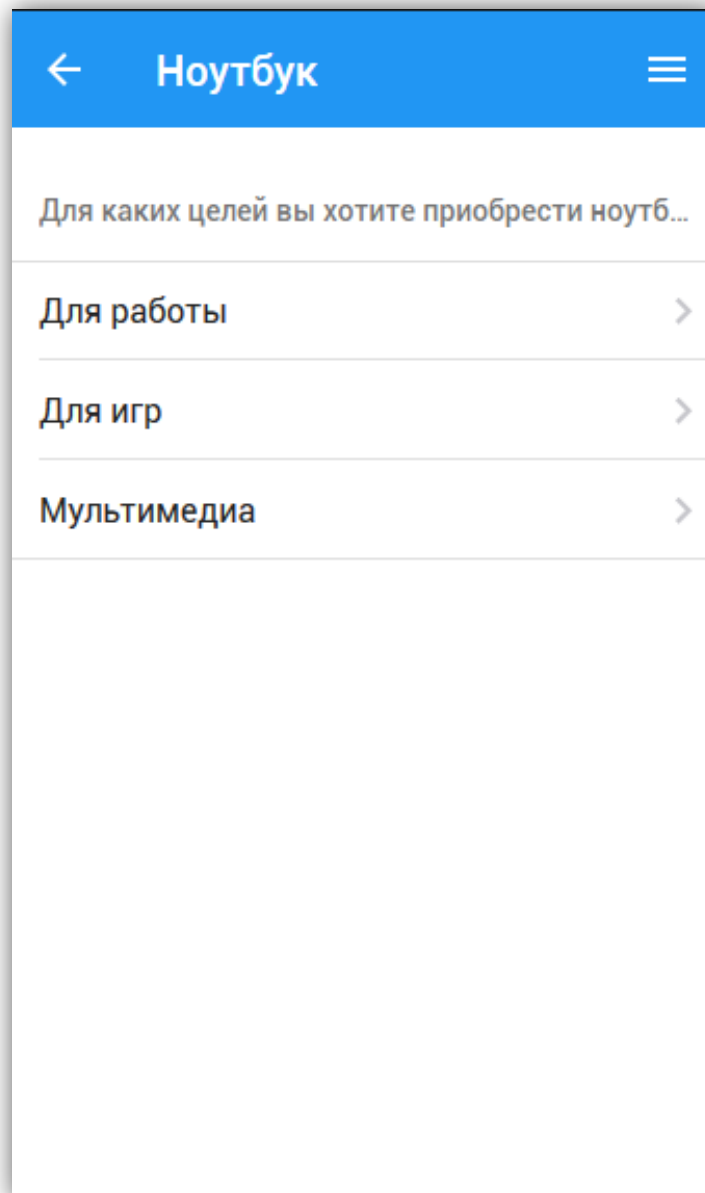


Рисунок 5.6 – Окно выбора деятельности

При переходе в категорию «Для работы» открывается окно с более развернутые категории: работы с документами, работа в it-сфере. В категории работы с документами будет показаны компьютеры, которые не отличаются особой мощностью. А в категории работы в it-сфере будет показаны компьютеры, подходящие для разработчиков и системных администраторов.

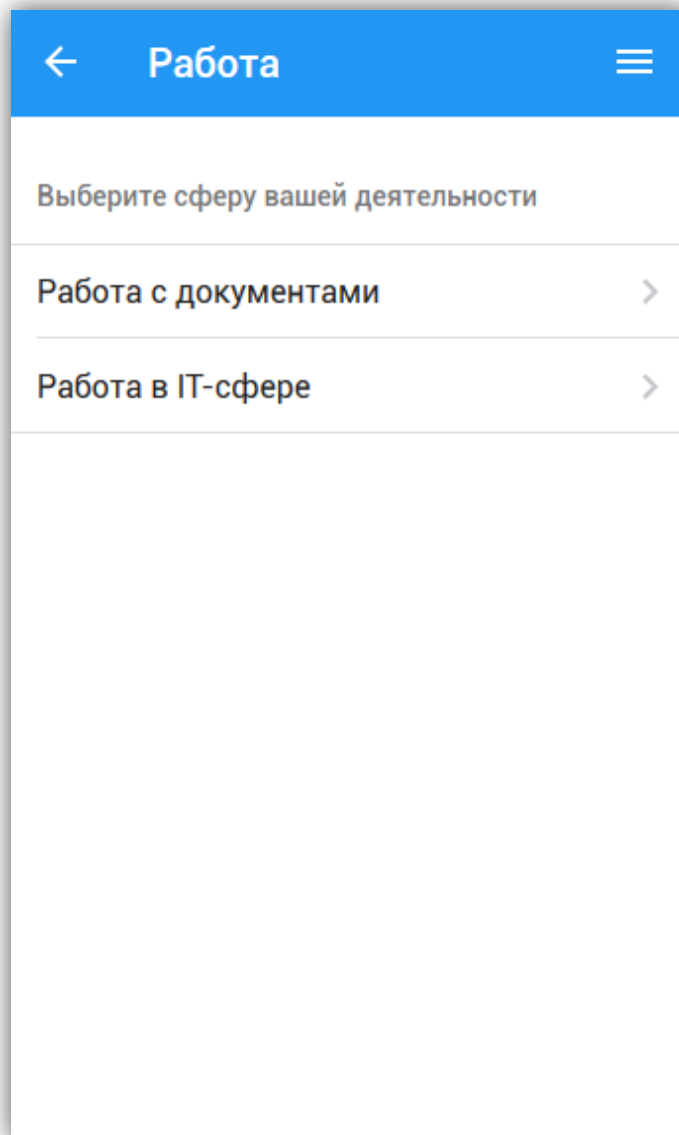


Рисунок 5.7 – Окно категорий «Для работы»

#### **5.4 Тестирование приложения**

Тестирование и отладка приложения производится прямо в браузере, в которых уже кстати есть режим мобильных устройств. Удобство заключается в том, что не нужно каждый раз запускать «тяжелые» и медленные эмуляторы устройств на компьютере. После смены какого-либо элемента интерфейса, просто обновляем страницу в браузере.

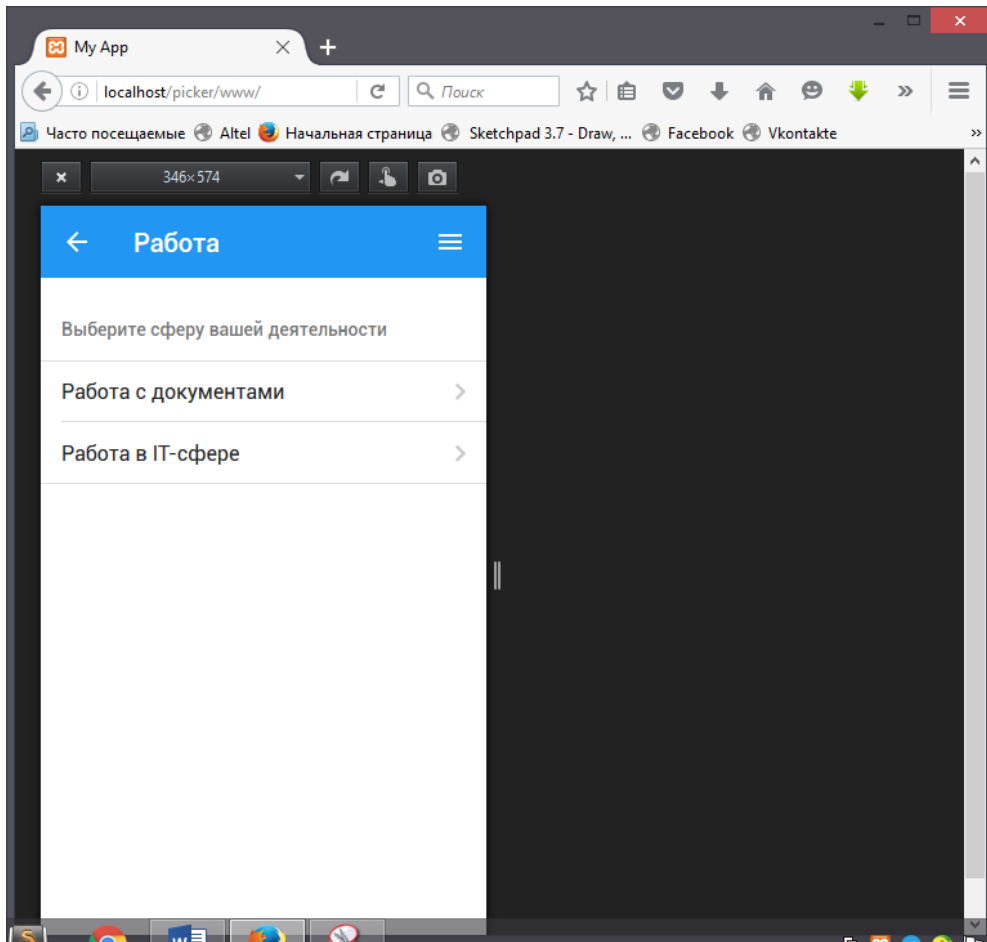


Рисунок 5.8 – Тестирование приложение в браузере Firefox

## 6 Технико-экономическое обоснование

### 6.1 Описание работы и обоснование необходимости

Целью данной дипломной работы является разработка мобильного приложения для продажи компьютеров. Мобильное приложение в первую очередь предназначено для помощи пользователям и покупателям в выборе компьютера или устройства, а также ремонта и технического обслуживания.

### 6.2 Расчет затрат на разработку информационных технологий

Расчет полных затрат на разработку проектного решения в виде информационных технологий ( $C_{ni}$ ) осуществляется по формуле

$$C_{ni} = Z_{\text{фот}} + Z_{\text{сзи}} + M_i + P_{ci} + P_{mi} + P_{\text{нки}} + P_{zi} + P_{ni} \quad (6.1)$$

где  $Z_{\text{фот}}$  – общий фонд оплаты труда разработчиков, тенге;

$Z_{\text{сзи}}$  – отчисления по социальному налогу, тенге;

$M_i$  – затраты на материалы, тенге;

$P_{ci}$  – затраты на специальные программные средства, необходимые для разработки проектного решения, тенге;

$P_{mi}$  – затраты, связанные с эксплуатацией техники, тенге;

$P_{\text{нки}}$  – затраты на научные командировки, тенге;

$P_{zi}$  – прочие затраты, тенге;

$P_{ni}$  – накладные расходы, тенге.

Размер фонда оплаты труда разработчиков ( $Z_{\text{фот}}$ ) рассчитывается по формуле:

$$Z_{\text{фот}} = Z_{oi} + Z_{di} \quad (6.2)$$

где  $Z_{oi}$  – основная заработная плата, тенге;

$Z_{di}$  – дополнительная заработная плата, тенге.

Общая трудоемкость проекта рассчитывается по формуле:

$$T_o = T_n * K_c * K_t * K_n \quad (6.3)$$

где  $K_c$  – коэффициент, учитывающий сложность ПО;

$K_t$  – поправочный коэффициент, учитывающий степень использования при разработке стандартных модулей;

$K_n$  – коэффициент, учитывающий степень новизны ПО;

$T_n$  – нормативная трудоемкость.

Расчет коэффициента сложности производится на основе данных, представленных в таблице 6.1 и составляет  $K_c = 0,26$  т.к. в ПО присутствует более 3-х характеристик - обеспечение хранения, ведения и поиска данных в

сложных структурах, интерактивный доступ, функционирование программного обеспечения в расширенной операционной среде.

Таблица 6.1 - Дополнительные коэффициенты сложности ПО

Характеристика ПО	Значения $K_c$
1. Функционирование ПО в расширенной операционной среде (связь с другими ПО)	0,08
2. Интерактивный доступ	0,06
3. Обеспечение хранения, ведения и поиска данных в сложных структурах	0,07
4. Наличие у ПО одновременно нескольких характеристик по табл.Г4.1, приложение Г	
– 4.1 2 характеристики	0,12
– 4.2 3 характеристики	0,18
– 4.3 Свыше 3-х характеристик	0,26

Поправочный коэффициент, учитывающий степень использования при разработке проекта стандартных модулей ( $K_T$ ), определяется на основе данных представленных в таблице 6.2 и составляет 0,8.

Таблица 6.2 – Значения поправочного коэффициента, учитывающего использование стандартных модулей типовых программ и ПО ( $K_T$ )

Степень охвата реализуемых функций разрабатываемого ПО стандартными модулями, типовыми программами и ПО	Значения $K_T$
1. От 60 % и выше	0,6
2. От 40 % до 60	0,7
3. От 20 % до 40 %	0,8
4. До 20 %	0,9
5. Типовые программы и ПО не используемые для реализации функций разрабатываемого ПО	1,0

Поправочный коэффициент, учитывающий новизну разрабатываемого проекта ( $K_H$ ) определяется на основе данных, представленных в таблице 6.3 и составляет 0,7.

Таблица 6.3 – Поправочные коэффициенты, учитывающие новизну ПО ( $K_n$ )

Категория новизны	Степень новизны	Использование		Значение $K_n$
		На основе нового типа ПК	В среде новой ОС	
А	Принципиально новые ПО, не имеющие доступных аналогов	+	+	1,75
		–	+	1,6
		+	–	1,2
		–	–	1,0
Б	ПО, являющиеся развитием определенного параметрического ряда ПО	+	+	1,0
		–	–	0,9
		+	–	0,8
В	ПО, являющиеся развитием определенного параметрического ряда ПО, разработанных для ранее освоенных типов конфигурации ПК и ОС	–	–	0,7

Базой для определения нормативной трудоемкости являются, укрупненные нормы времени на разработку проекта в зависимости от уточненного объема ПО и группы сложности. Нормативная трудоемкость проекта ( $T_n$ ) определяется на основе принятого в расчет объема ПП и категории сложности, которая уточняется с учетом сложности и новизны проекта и степени использования стандартных модулей при разработке.

$$T_n = 487 \quad (6.4)$$

Рассчитаем общий объем трудоемкости исходя из формулы

$$T_o = 487 \cdot 0,26 \cdot 0,8 \cdot 0,7 = 71 \text{ чел./дн.} \quad (6.5)$$

На основе трудоемкости определяются плановое число разработчиков ( $Ч_p$ ) и плановые сроки, необходимые для реализации проекта в целом ( $T_p$ ). При этом могут решаться следующие задачи:

- расчет числа исполнителей при заданных сроках разработки проекта;
- определение сроков разработки проекта при заданной численности исполнителей.

Численность исполнителей проекта ( $Ч_p$ ) рассчитывается по формуле:



$$Ч = T_0 / (T_p * \Phi_{эф}) \quad (6.6)$$

где  $\Phi_{эф}$  – эффективный фонд времени работы одного работника в течение года (дн.);

$T_0$  – общая трудоемкость разработки проекта (чел./дн.);

$T_p$  – срок разработки проекта (лет).

Срок разработки проекта ( $T_p$ ) определяется по формуле

$$T_p = T_0 / (Ч_p * \Phi_{эф}) \quad (6.7)$$

где  $Ч_p$  – плановое число разработчиков.

Эффективный фонд времени работы одного работника ( $\Phi_{эф}$ ) рассчитывается по формуле:

$$\Phi_{эф} = D_r - D_{п} - D_{в} - D_o \quad (6.8)$$

где  $D_r$  – количество дней в году;

$D_{п}$  – количество праздничных дней в году;

$D_{в}$  – количество выходных дней в году;

$D_o$  – количество дней отпуска.

Т.к., в соответствии с производственным календарем на 2016 год:

$D_r = 366$ ;

$D_{п} = 16$ ;

$D_{в} = 103$ ;

$D_o = 14$ , эффективный фонд времени одного работника составит

$$\Phi_{эф} = 366 - 16 - 103 - 14 = 233 \text{ дня}$$

Плановое число разработчиков  $Ч_p = 1$ , следовательно, по формуле

$$T_p = 71 / (1 * 233) = 0,30 = 110 \text{ день} \quad (6.9)$$

Таким образом, согласно произведенным расчетам и в соответствие с формулой

$$Ч = \frac{71}{0,30 * 233} = 1 \text{ человек} \quad (6.10)$$

Основная заработная плата исполнителей на конкретное ПО рассчитывается по формуле:

$$З_{oi} = \sum_{i=1}^n T_{чi} \cdot T_{ч} \cdot K \quad (6.11)$$

где  $n$  – количество исполнителей, занятых разработкой конкретного ПО;  
 $T_{чi}$  – часовая тарифная ставка  $i$ -го исполнителя (тыс.тенге);  
 $\Phi_{пi}$  – плановый фонд рабочего времени  $i$ -го исполнителя (дней), 110 раб день;  
 $T_{ч}$  – количество часов работы в день (час), 8 часов;  
 $K$  – коэффициент премирования, составляет 1,38.

По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов–исполнителей, участвующих в разработке ПО, с определением образования, специальности, квалификации и должности (таблица 6.4).

$\Phi_{п} = 26$  рабочих дней.

Таблица 6.4 – Сведения по работникам, задействованным в проекте

Специалист–Исполнитель	Количество, человек	Заработная плата в месяц, тенге
Программист	1	90000
Итого	1	90000

Средний дневной заработок каждого работника рассчитывается по формуле:

$$D = \frac{Z_m}{D_p} \quad (6.12)$$

где  $Z_m$ – ежемесячный размер заработной платы;  
 $D_p$ – количество рабочих дней в месяце (это 26 дней – шестидневная рабочая неделя).

$$D = \frac{90000}{26} = 3462 \text{ тенге} \quad (6.13)$$

Часовая тарифная ставка рассчитывается путем деления месячной тарифной ставки, установленную при 48–часовой недельной норме рабочего времени и общего фонда времени ( $\Phi_p$ )

$$T_{ч} = T_m / \Phi_p \quad (6.14)$$

где  $T_{ч}$  – часовая тарифная ставка (тыс.тенге);  
 $T_m$  – месячная тарифная ставка (тыс.тенге).

Общий фонд времени:

$$\Phi_p = T_q * \Phi_{\Pi} \quad (6.15)$$

Таким образом

$$\Phi_p = 8 * 26 = 208 \text{ час.}$$

Общее количество часов составит:

$$\Phi_p = 8 * 110 = 880 \text{ час.}$$

Рассчитаем тарифную ставку инженера–программиста

$$T_q = \frac{90\,000}{208} = 433 \text{ тенге/час}$$

В соответствие с формулой основная заработная плата инженера–программиста составит

$$З_{oi} = 433 * 8 * 110 * 1,38 = 525835,2 \text{ тенге}$$

Социальный налог составляет 11% (ст. 358 п. 1 НК РК) от дохода работника, и рассчитывается по формуле

$$З_{czi} = (\text{ФОТ} - \text{ПО}) * 11\% \quad (6.16)$$

Где ПО – пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются

$$\text{ПО} = \text{ФОТ} * 10\% \quad (6.17)$$

Таким образом

$$\text{ПО} = 525835,2 * 0,1 = 52583,5 \text{ тенге}$$

$$З_{czi} = (525835,2 - 52583,5) * 0,11 = 52057,7 \text{ тенге}$$

Затраты на материалы определяются по формуле

$$M_i = (З_{ocн} * Н_{мз})/100\% \quad (6.18)$$

где  $H_{мз}$  – норма расхода материалов основной заработной платы (3–5%).

$$M_i = 525835,2 \cdot 0,04 = 21033,4 \text{тенге}$$

Расходы по статье «Спецоборудование» ( $P_{ci}$ ) включают затраты средств на приобретение вспомогательных специального назначения технических и программных средств, необходимых для разработки конкретного ПО включая расходы на их проектирование, изготовление, отладку, установку и эксплуатацию:

$$P_{ci} = \sum_{i=1}^n C_{ci} \quad (6.19)$$

Таблица 6.5 – Перечень оборудования, необходимого для разработки

Наименование изделий	Характеристика	Количество единиц	Цена за единицу, тенге	Общая сумма, тенге
Ноутбук	Lenovo g510 Intel Core -i7	1	130000	130000
Итого				130000

Таблица 6.6 – Программное обеспечение, используемое в работе

Программное обеспечение	Стоимость, тенге
Sublime Text	бесплатно
Gimp	бесплатно
NetBeans	бесплатно
ХАМРР Control Panel	бесплатно
Итого	0

Амортизационные отчисления по используемому оборудованию рассчитываются по формуле:

$$A_i = \frac{N_A * C_{ПЕР} * N}{100 * n} \quad (6.20)$$

где  $N_A$  – норма амортизации;

$C_{ПЕР}$  – первоначальная стоимость оборудования;

$N$  – количество дней на выполнение работ;

$n$  – количество рабочих дней в году.

Таким образом

$$A_{об} = \frac{130000 * 25 * 110}{100 * 366} = 9768 \text{тенге}$$

Расходы по статье «Прочие затраты» ( $P_{zi}$ ) на конкретное ПО включают затраты на приобретение и подготовку специальной научно–технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате

$$P_{zi} = Z_{oi} * N_{пз}/100 \quad (6.21)$$

Где  $N_{пз}$  – норматив прочих затрат в целом по организации (20%). Таким образом

$$P_{zi} = 525835,2 \cdot 0,2 = 105167 \text{ тенге}$$

Затраты по статье «Накладные расходы» ( $P_{ни}$ ), рассчитывается по нормативу ( $N_{рн}$ ) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации

$$P_{ни} = Z_{oi} * N_{рн}/100\% \quad (6.22)$$

где  $P_{ни}$ – накладные расходы на конкретную ПО (тыс.тенге);

$N_{рн}$  – норматив накладных расходов в целом по организации (70%).

Таким образом накладные расходы составят

$$P_{ни} = 124288,3 \cdot 0,7 = 368084,6 \text{ тенге}$$

Полные затраты на разработку мобильного приложения для продажи компьютеров, составят

$$C_{ni} = 525835,2 + 52057,7 + 21033,4 + 9768 + 105167 + \\ + 368084,6 = 1081915,9 \text{ тенге.}$$

Сводные результаты расчета затрат на разработку ПО и их структура представлены в таблице 6.7 и на рисунке 1.

Таблица 6.7 – Затраты на разработку мобильного приложения для продажи компьютеров

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Фонд оплаты труда	$Z_{ФОТ}$	525835,2	48,6
Социальный налог	$Z_{сзи}$	52057,7	4,8

продолжение таблицы 6.7

Материалы	$M_i$	21033,4	1,9
Амортизационные отчисления	$A_{об}$	9768	0,9
Прочие затраты	$P_{zi}$	105167	9,7
Накладные расходы	$P_{ni}$	368084,	34,0
		6	
Итого:		1081915,9	100

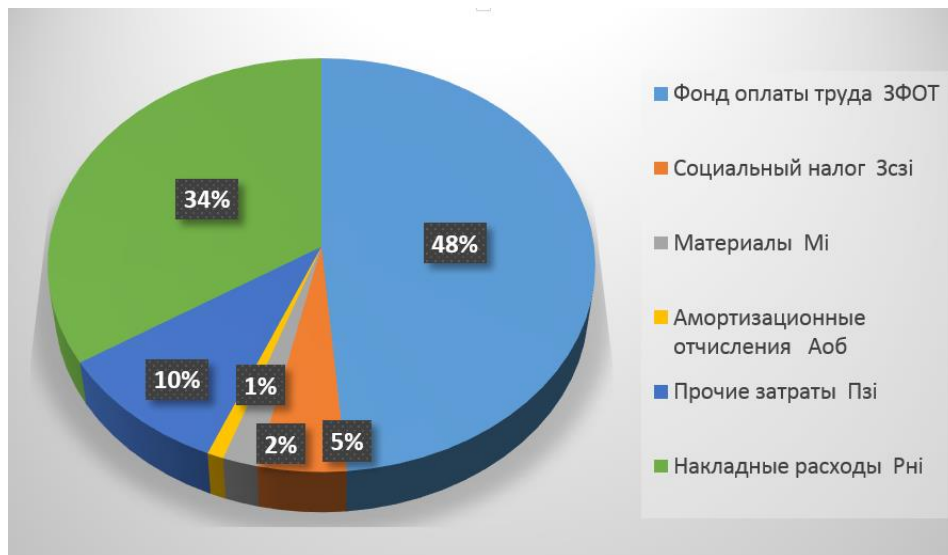


Рисунок 1 – Структура затрат на разработку ПП

### 6.3 Расчет цены программного продукта

Расчет цены ПП, который разработан одной организацией по заказу другой и не предназначен для тиражирования, осуществляется по формуле

$$C_{ПП} = Z_{РПР} + П_{П} + НДС \quad (6.23)$$

где  $C_{ПП}$  – цена программного продукта, тенге;

$Z_{РПР}$  – затраты на разработку проектного решения, в данном случае программного продукта, тенге;

$П_{П}$  – планируемая прибыль, тенге;

НДС – налог на добавленную стоимость, тенге.

Планируемая прибыль составляет (20%) от себестоимости разработки

$$П_{П} = Z_{РПР} \cdot 0,2 \quad (6.24)$$

$$П_{П} = 1081915,9 \cdot 0,2 = 216383,2 \text{ тенге}$$

НДС, начисленный на  $P_{\Pi}$ , определяется следующим образом

$$\text{НДС} = (P_{\Pi} + Z_{\text{рпр}}) * k_{\text{НДС}} \quad (6.25)$$

где  $k_{\text{НДС}}$  – ставка налога на добавленную стоимость.

Подставив данные в формулу получаем

$$\text{НДС} = (1081915,9 + 216383,2) \cdot 0,12 = 155795,9 \text{ тенге}$$

Цена реализации  $P_{\Pi}$  по формуле составляет

$$C_{\text{ПП}} = 1081915,9 + 216383,2 + 155795,9 = 1454094,9 \text{ тенге}$$

#### **6.4 Оценка социально - экономических результатов**

Программный продукт оправдывает себя с первых дней внедрения разработки, исключая бумажную волокиту, экономя время на поиск данных и регистрацию клиента. Таким образом, сокращается количество времени на обслуживание одного клиента и сервис может обслужить большее количество людей. Отпадает проблема хранения больших объемов информации на бумажных и отдельных электронных носителях, сохраняя все данные в единой базе. ПП не требует дальнейших разработок и поддержания специалистами своего рабочего состояния.

## **7 Безопасность жизнедеятельности**

### **7.1 Анализ условий труда**

Главной целью данной работы является разработка мобильного приложения для продажи компьютеров. Данный программный продукт служит для роста продаж компьютерной техники.

Технический персонал состоит из одного сотрудника – программист-разработчик.

Работа сотрудников непосредственно связана с компьютером, а соответственно с вредным дополнительным воздействием целой группы факторов, что существенно снижает производительность их труда.

При выполнении работ на персональном компьютере (ПК) согласно ГОСТу 12.0.003-74 “ССБТ. Опасные и вредные производственные факторы. Классификация” могут иметь место следующие факторы:

- повышенная температура поверхностей ПК;
- повышенная или пониженная температура воздуха рабочей зоны;
- выделение в воздух рабочей зоны ряда химических веществ;
- повышенная или пониженная влажность воздуха;
- повышенный или пониженный уровень отрицательных и положительных аэроионов;
- повышенное значение напряжения в электрической цепи, замыкание;
- повышенный уровень статического электричества;
- повышенный уровень электромагнитных излучений;
- повышенная напряженность электрического поля;
- отсутствие или недостаток естественного света;
- недостаточная искусственная освещенность рабочей зоны;
- повышенная яркость света;
- повышенная контрастность;
- зрительное напряжение;
- монотонность трудового процесса;
- нервно-эмоциональные перегрузки.

#### **7.1.1 План помещения**

Помещение имеет размеры: длина (L) = 6 метров, ширина (B) = 4 метра, высота (H) = 3,2 метра. Помещение находится в здании на 3-м этаже, рассчитано на 2 рабочих места.

План помещения выбранного для размещения оборудования и технического персонала изображен на (Рисунке 7.1).



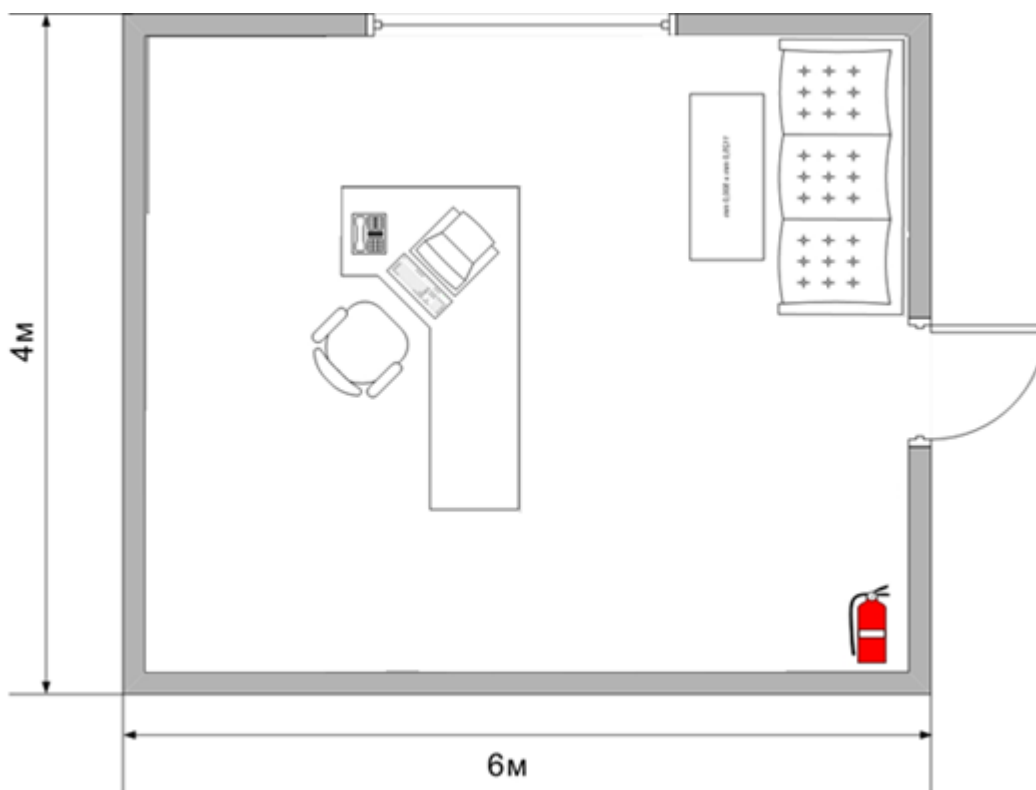


Рисунок 7.1 – План помещения

### 7.1.2 Освещенность

Искусственное освещение в помещениях эксплуатации компьютеров должно осуществляться системой общего равномерного освещения.

Освещенность на поверхности стола в зоне размещения документа должна быть 300-500 лк. Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк. Прямую блескость от источников освещения следует ограничить. Яркость светящихся поверхностей (окна, светильники), находящихся в поле зрения, должна быть не более 200 кд/м<sup>2</sup>.

Отраженная блескость на рабочих поверхностях ограничивается за счет правильного выбора светильника и расположения рабочих мест по отношению к естественному источнику света. Яркость бликов на экране монитора не должна превышать 40 кд/м<sup>2</sup>. Показатель ослепленности для источников общего искусственного освещения в помещениях должен быть не более 20, показатель дискомфорта в административно-общественных помещениях не более 40. Соотношение яркости между рабочими поверхностями не должно превышать 3:1 - 5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

Для искусственного освещения помещений с персональными компьютерами следует применять светильники типа ЛПО36 с

зеркализированными решетками, укомплектованные высокочастотными пускорегулирующими аппаратами. Допускается применять светильники прямого света, преимущественно отраженного света типа ЛПО13, ЛПО5, ЛСО4, ЛПО34, ЛПО31 с люминесцентными лампами типа ЛБ. Допускается применение светильников местного освещения с лампами накаливания. Светильники должны располагаться в виде сплошных или прерывистых линий сбоку от рабочих мест параллельно линии зрения пользователя при разном расположении компьютеров. При периметральном расположении - линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращенному к оператору. Защитный угол светильников должен быть не менее 40 градусов. Светильники местного освещения должны иметь не просвечивающийся отражатель с защитным углом не менее 40 градусов.

Для обеспечения нормативных значений освещенности в помещениях следует проводить чистку стекол оконных проемов и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

### **7.1.3 Уровень шума**

Уровни шума на рабочих местах пользователей персональных компьютеров не должны превышать значений более 50 дБА. На рабочих местах в помещениях для размещения шумных агрегатов уровень шума не должен превышать 75 дБА. Снизить уровень шума в помещениях можно использованием звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот 63-8000 Гц для отделки стен и потолка помещений. Дополнительный звукопоглощающий эффект создают однотонные занавески из плотной ткани, повешенные в складку на расстоянии 15-20 см от ограждения. Ширина занавески должна быть в 2 раза больше ширины окна.

### **7.1.4 Оценка микроклимата**

Работа технического персонала в помещении согласно ГОСТ 12.1.005-88 ССБТ "Воздух рабочей зоны, общие санитарно-гигиенические требования", относится к работе средней тяжести (таблица 5.1), т.к. управление оборудованием осуществляется дистанционно из комнаты дежурного персонала с помощью персональных компьютеров.

Таблица 5.1 – Категории работ по энергозатратам организма

Работа, Категория	Энергозатраты организма, Дж/с (ккал/час)	Характеристика работы
-------------------	--	-----------------------

продолжение таблицы 5.1

Физическая средней тяжести, Па	172 – 232	Связанная с ходьбой, выполняемая стоя или сидя, но не требующая перемещения тяжестей
--------------------------------------	-----------	--

В комнате дежурного персонала оптимальные условия микроклимата для физической работы средней тяжести согласно нормам ГОСТ 12.0.003-88. ССБТ (приведены в таблице 5.2), в полной мере обеспечиваются с помощью системы кондиционирования.

Таблица 5.2 - Оптимальные нормы параметров микроклимата

Период работы	Категор ия работы	Темпера тура, °С	Скорость движения воздуха не более, м/с
Холодный и переходный	II а	18-20	0,2
Теплый	II а	21-23	0,3

Оборудование, установленное в помещении, выделяет значительное количество тепла и для точного поддержания микроклимата в помещениях данного типа устанавливаются системы кондиционирования воздуха для технологических помещений, которые обеспечивают оптимальные климатические условия для поддержания продолжительной работы оборудования, показаны в таблице 5.3.

Таблица 5.3 – Оптимальные климатические условия для продолжительной работы оборудования

Темпера тура воздуха, °С	Относительная влажность воздуха не более, %	Скорость движения воздуха не более, м/с
от +5 до + 40	от 10 до 75	0,1 – 0,2

Помещение снабжается системой кондиционирования воздуха, которая обеспечивает наилучший микроклимат в помещении и условия работы точной и чувствительной аппаратуры. Расчет воздушной среды рабочей зоны приведен в пункте 7.2.

## 7.2 Расчет воздушной среды рабочей зоны

Расчет ведется согласно методическому указанию СНиП РК 4.02-42-2006.

При выборке требований к системам общеобменной вентиляции необходимо исходить из необходимости удаления из производственного помещения всех присутствующих вредностей, т.е. избытков тепла, влаги,

паров газов и пылей. Система вентиляции должна обеспечить требуемую кратность воздухообмена:

$$n = \frac{L_b}{V_{II}}, \quad (7.1)$$

где  $L_b$  – количество воздуха, поступающего (или удаляемого) в помещение, м<sup>3</sup>/ч;

$V_{II}$  – объем помещения, м<sup>3</sup>.

За счет тепловыделений производственного оборудования могут иметь место значительные избытки тепла. Обеспечение оптимальных для жизнедеятельности человека параметров микроклимата и воздушной среды осуществляется с помощью обширного комплекса методов и средств. Наиболее перспективным средством, обеспечивающим чистоту и нормальный микроклимат, является кондиционирование, т.е. создание искусственного микроклимата в производственном помещении с помощью кондиционирующих установок.

Количество воздуха  $L$  (м<sup>3</sup>/ч), которое необходимо вывести за один час из производственного помещения, чтобы вместе с ним удалить избыток тепла  $P_{изб}$ , определяется по формуле:

$$L = \frac{Q_{изб}}{C_e \cdot \Delta t \cdot \gamma_e}, \quad (7.2)$$

где  $Q_{изб}$  – избыточное тепло, ккал/ч;

$C_e$  – теплоемкость воздуха (0,24 ккал/кг °С);

$\gamma_e$  – удельная масса приточного воздуха ( $\gamma_e = 1,20$  кг/м<sup>3</sup>);

$\Delta t$  – разность температур выходящего и входящего воздуха в помещение, °С.

Избыточное тепло определяется следующим образом:

$$Q_{изб} = Q_{об} + Q_{осв} + Q_{л} - Q_{пр} - Q_{отд},$$

(7.3)

где  $Q_{об} + Q_{осв} + Q_{л}$  – тепло, выделяемое производственным оборудованием, системой искусственного освещения помещения и работающим персоналом (людьми) соответственно, ккал/ч;

$Q_{пр}$  – тепло, вносимое в помещение солнцем (солнечная радиация), ккал/ч;

$Q_{отд}$  – теплоотдача естественным путем, ккал/ч.

Тепло, выделяемое производственным оборудованием определяется как:

$$Q_{об} = 860 P_{об} \eta, \quad (7.4)$$

где  $P_{об}$  – мощность, потребляемая оборудованием,  $P_{об} = 4,8$  кВт (в помещении автозала);

$\eta$  – коэффициент перехода тепла в помещение,  $\eta = 0,75$  (для коммутаторного зала интеллектуальных услуг)

860 – тепловой эквивалент 1кВт/ч.

$$Q_{об} = 3400 \text{ ккал/ч.}$$

Тепло, выделяемое осветительными установками:

$$Q_{осв} = 860 P_{осв} \alpha \beta \cos \varphi, \quad (7.5)$$

где  $P_{осв}$  – мощность осветительных установок,  $P_{осв} = 0,16$  кВт;

$\alpha$  – КПД перевода электрической энергии в тепловую,  $\alpha = 0,1$ ;

$\beta$  – КПД одновременности работы аппаратуры в помещении,  $\beta = 1$ ;

$\cos \varphi$  – коэффициент,  $\cos \varphi = 0,7$ .

$$Q_{осв} = 100 \text{ ккал/ч}$$

Количество персонала, работающего в помещении ЦСБ, составляет пять человек, значение  $Q_{л}$  пренебрегается,  $Q_{л} = 0$ .

Солнечная радиация, т.е. тепло, вносимое в помещение солнцем, рассчитывается следующим образом :

$$Q_p = m \cdot F \cdot q_{осм}, \quad (7.6)$$

где  $m$  – количество окон в помещении,  $m = 1$ ;

$F$  – площадь одного окна,  $F = 3,5 \text{ м}^2$ ;

$q_{осм}$  – солнечная радиация через остекленную поверхность, т.е. количество тепла, вносимое за один час через остекленную поверхность площадью  $1 \text{ м}^2$ ,  $q_{осм} = 70$  ккал/ч.

$$Q_p = 245 \text{ ккал/ч.}$$

В помещениях с большими теплоизбытками считается, что теплоотдача естественным путем через конструкции помещения приблизительно равна количеству тепла, вносимого в помещения солнечной радиацией через окна,

т.е.  $Q_{от} = Q_p$ , ккал/ч.

Рассчитаем избыточное тепло по формуле 5.3:

$$Q_{изб} = 3400 - 245 = 3155 \text{ ккал/ч}$$

Необходимый воздухообмен рассчитаем по формуле 5.2:

$$L = \frac{3105,63}{0,24 \cdot 30 \cdot 1,206} = 357,89 \text{ м}^3/\text{ч}$$

Отношение количества воздуха, поступающего в помещение за 1 ч, к объему помещения называется кратностью воздухообмена:

$$n = \frac{357,89}{53} = 6,75$$

В результате расчета для обеспечения необходимых микроклиматических условий, отвечающих требованиям СНиП РК 4.02-05-2001. В качестве системы кондиционирования выбирается кондиционер LGS24LHP типа сплит – системы, который состоит из одного наружного и одного внутреннего блоков. Используется в жилых, служебных и других помещениях площадью до 70 м<sup>2</sup> с целью создания благоприятных условий для жизнедеятельности человека. Кондиционер обеспечивает: охлаждение воздуха; автоматическое поддержание заданной температуры; очистку воздуха от пыли; вентиляцию; уменьшение влажности воздуха (удаление влаги из воздуха); воздухообмен с окружающей средой.



Рисунок 7.1 – Кондиционер LGS24LHP

Техническая характеристика кондиционера LGS24LHP:

Габаритные размеры (ВхШхГ):

- внутренний блок 300x1090x178 мм;
- наружный блок 577x870x276 мм.

Потребляемая мощность – 2,7/2,7 кВт;

Номинальный воздушный поток (максимальное значение):

- внутренний 16 м<sup>3</sup>/мин;
- наружный 42 м<sup>3</sup>/мин.

Режим работы - Автоматический, регулируемый.

### 7.3 Расчет пожарной безопасности

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

Причинами возникновения пожара могут быть:

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

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

В соответствии с требованиями правил пожарной безопасности помещение оборудованы углекислотными огнетушителями ОУ-5 с учетом - один огнетушитель на 100 м<sup>2</sup>. Общая площадь помещения составляет 24 м<sup>2</sup>, таким образом устанавливается 1 огнетушитель. В качестве огнетушащего вещества применяется комбинированный углекислотно-хладоновый состав. Расчетная масса комбинированного углекислотно-хладонового состава  $m_d$ , для объемного пожаротушения определяется по формуле

$$(7.7) \quad m_d = k \cdot g_n \cdot V$$

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

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

$V$  - объем помещения определяется по формуле

$$(7.8) \quad V = A \cdot B \cdot H,$$

где  $A = 6\text{м}$  - длина помещения,

$B = 4\text{м}$  - ширина помещения,

$H = 3,2\text{м}$  - высота помещения.

Тогда

$$V = 6 \cdot 4 \cdot 3,2 = 76,8 \text{ м}^3$$

Следовательно

$$m_d = 1,2 \cdot 0,04 \cdot 76,8 = 3,68 \text{ кг}$$

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

Внутренний диаметр магистрального трубопровода  $d_i$ (мм), определяется по формуле

$$d_i = 12 \cdot \sqrt{2} = 17 \text{ мм} \quad (7.9)$$

Эквивалентная длина магистрального трубопровода  $l_2$  определяется по формуле

$$l_2 = k_1 \cdot l \quad (7.10)$$

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

$l=3,2$ м - длина трубопровода по проекту тогда.

$$l_2 = 1,2 \cdot 3,2 = 3,84 \text{ м}$$

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

Расчетное время подачи углекислотно-хладонового состава  $t$ , определяется по формуле (5.10)

$$t = m_d / 60Q. \quad (7.11)$$

Тогда

$$T = 3,68 / 76,8 \cdot 1,4 = 0,067 \text{ мин}$$

Масса основного запаса углекислотно-хладонового состава  $m$  определяется по формуле (5.11)

$$m = 1,1 \cdot m_d \cdot \left(1 + \frac{k_2}{k_1}\right) \quad (5.12)$$

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

$$m = 1,1 \cdot 3,68 \cdot \left(1 + \frac{0,2}{1,2}\right) = 4,72 \text{ кг}$$



Таким образом, из полученных результатов можно сделать вывод, что для обеспечения нормального функционирования системы автоматического пожаротушения потребуется 1 баллон углекислотно-хладонового состава вместимостью 20 литров, с массой смеси 7,2 кг.

## Заключение

В данном дипломном проекте было разработано мобильное приложение для подбора и продажи компьютерной техники. Актуальность работы заключена в создании нового направления в сфере продажи техники, которая исключает субъективное отношение к покупателю. В качестве инструментов для разработки были выбраны Sublime Text и ConEmu, так как они очень удобны и практичны для разработки. В качестве языка программирования были выбраны в основном Javascript, так как этот язык является подходящим для гибридной разработки мобильных приложений. Для разработки дизайна был использован концепт Material Design от компании Google. Для достижения поставленных целей задач была проделана следующая работа: детально изучен Фреймворк, который использовался для разработки, освоены знания в области программирования, а также изучены языки программирования Javascript, PHP.

Эффективность своего проекта вижу в том, что, был получен богатый опыт в разработке информационных систем, был получен большой опыт в разработке приложений и навыков программирования.

В ходе разработки программного продукта были изучены похожие программы и рынки в мировом сегменте. Кроме того, были проанализированы эффективные методы создания подобных систем, плюсы и минусы тех и иных программ. В разделе технико-экономического обоснования были произведены основные экономические расчеты, по которым видно, что проект вполне окупаемый. И цена за продукт является приемлемой. В разделе безопасность жизнедеятельности был произведён анализ условий труда, в результате которого был произведён расчет микроклимата и проверка на соответствие нормам системы кондиционирования. Разработанная система удовлетворяет всем требованиям, поставленным на этапе постановки задачи. При разработке системы, все функции и свойства системы были созданы с нуля. В дальнейшем система будет совершенствоваться и будут созданы новые возможности, в которых будет нуждаться пользователь.

## Список литературы

- 1 Бен Фрейн Ф86 HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. — СПб.: Питер, 2014. — 304 с.
- 2 Дженнифер Нидерст Роббинс. Web-дизайн. Справочник. — М.:Кудиц-Пресс, 2008. — 816 с.
- 3 В.В. Сенчилов, Т.А. Самойлова — Разработка гибридных приложений для мобильных устройств.— 828 с.
- 4 Вебсайт «Sulpak» - <http://www.sulpak.kz/>
- 5 Эстель Вейл — HTML5.Разработка приложений для мобильных устройств.
- 6 Вебсайт «Pulser» - <http://www.pulser.kz/>
- 7 Дэвид Макфарланд — Новая большая книга. CSS.
- 8 Никсон Р. — Создаем динамические веб-сайты с помощью PHP, MySQL, javascript, CSS и HTML5. 3-е изд.— 497 с.
- 9 Стив Суэринг, Тим Конверс, Джойс Парк — PHP и MySQL.— 912 с.
- 10 Дэвид Скляр, Адам Трахтенберг — PHP. Рецепты программирования
- 11 Байзакова А.А., Санатова Т.С. Охрана труда: Методические указания к выполнению расчетно-графических работ. — 2015.—29 с.
- 12 Белов С.В., Ильницкая А.В., Козьяков А.Ф. и др. Безопасность жизнедеятельности. Учебник для вузов. — М.: Высш.шк., 2005. — 448 с.
- 13 Хакимжанов Т.Е. Сборник задач по охране труда и безопасности жизнедеятельности: Учебное пособие для вузов. — Алматы: Эверо, 2007.— 274с.
- 14 Сивков В.П., Смирнов С.Г., Козьяков А.Ф. и др. Сборник типовых расчетов по курсу «Охрана труда». — М.: МВТУ, 1979. — 79 с.
- 15 Абикенова А.А., Методические указания к выполнению раздела «Пожарная профилактика» - Алматы: АИЭС, 2009. — 42 с.
- 16 Г.М. Гукасян. Экономика от А до Я: Тематический справочник. — М.: ИНФРА-М, 2009. — 480 с.
- 17 Экономика промышленного предприятия: учебник / И.Н. Иванов. — М.: ИНФРА-М, 2011. — 395 с.
- 18 Ю.М. Остапенко. Экономика труда: учебное пособие. —М.: ИЦ РИОР, 2010. — 160 с.
- 19 А.И. Рофе. Экономика труда: учебник. — М.: КиноРус, 2010. — 400 с.

## Приложение А

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Required meta tags-->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, minimum-scale=1, user-scalable=no, minimal-ui">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <!-- Color theme for statusbar -->
    <!-- Your app title -->
    <title>My App</title>
    <!-- Path to Framework7 Library CSS, Material Theme -->
    <link rel="stylesheet" href="css/framework7.material.min.css">
    <!-- Path to Framework7 color related styles, Material Theme -->
    <link rel="stylesheet" href="css/framework7.material.colors.min.css">
    <!-- Path to your custom app styles-->
    <link rel="stylesheet" href="css/my-app.css">
  </head>
  <body>
    <div class="statusbar-overlay"></div>
    <div class="panel-overlay"></div>
    <div class="panel panel-left panel-cover">
      <div class="view navbar-fixed">
        <div class="pages">
          <div data-page="panel-left" class="page">
            <div class="navbar">
              <div class="navbar-inner">
                <div class="center">Menu</div>
              </div>
            </div>
            <div class="page-content">
              <div class="list-block">
                <ul>
                  <li><a href="index.html" class="item-link close-panel">
                    <div class="item-content">
                      <div class="item-inner">
                        <div class="item-title">Главная</div>
                      </div>
                    </div></a>
                </li>
                </ul>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

## Продолжение приложения А

```
</li>
  <li><a href="#" class="item-link close-panel">
    <div class="item-content">
      <div class="item-inner">
        <div class="item-title">Настройки</div>
      </div>
    </div></a>
  </li>
  <li><a href="#" class="item-link close-panel">
    <div class="item-content">
      <div class="item-inner">
        <div class="item-title">Помощь</div>
      </div>
    </div></a>
  </li>
  <li><a href="about.html" class="item-link close-panel">
    <div class="item-content">
      <div class="item-inner">
        <div class="item-title">О приложении</div>
      </div>
    </div></a>
  </li>
</ul>
</div>
</div>
</div>
</div>
</div>
<!-- Views -->
<div class="views">
  <!-- Your main view, should have "view-main" class -->
  <div class="view view-main">
    <!-- Pages container, because we use fixed navbar and toolbar, it has
additional appropriate classes-->
    <div class="pages navbar-fixed">
      <!-- Page, "data-page" contains page name -->
      <div data-page="index" class="page">
        <!-- Scrollable page content --><div class="navbar">
```

*Продолжение приложения А*

```
<div class="navbar-inner">
  <div class="center">Picker</div>
  <div class="right"><a href="#" class="open-panel link icon-
only"><i class="icon icon-bars"></i></a></div>
</div>
</div>
<div class="page-content">
  <!--<div class="home-cards">
    <a href="#"><div class="card picker-card-header-pic">
      <div style="background-image:url(img/laptop.png)"
valign="bottom" class="card-header color-black no-border"></div>
      <div align="center" class="card-content">
        Ноутбук
      </div>
    </div>
  </div>-->
  <div class="content-block-title">Выберите устройство</div>
  <div class="list-block">
    <ul>
      <li><a href="pages/note/note-targeting.html" class="item-link
close-panel">
        <div class="item-content">
          <div class="item-inner">
            <div class="item-title">Ноутбук</div>
          </div>
        </div></a>
      </li>
      <li><a href="pages/pc/pc-targeting.html" class="item-link close-
panel">
        <div class="item-content">
          <div class="item-inner">
            <div class="item-title">Настольный ПК</div>
          </div>
        </div></a>
      </li>
      <li><a href="pages/mono/mono-targeting.html" class="item-link
close-panel">
        <div class="item-content">
```



## Приложение Б

```
// Initialize your app
var myApp = new Framework7({
  material: true,
});
// Export selectors engine
var $$ = Dom7;

$$ (document).on('pageInit', function (e) {
  // Page Data contains all required information about loaded and initialized page
  var page = e.detail.page;

  document.addEventListener("deviceready", appReady, false);
  function appReady(){
    document.addEventListener("backbutton", function(e){
      var page=myApp.getCurrentView().activePage;
      myApp.hidePreloader();
      if(page.name=="index"){
        e.preventDefault();
        if(confirm("Do you want to Exit!"))
        {
          navigator.app.clearHistory();
          navigator.app.exitApp();
        }
      }
      else {
        navigator.app.backHistory();
      }
    }, false);
  }

  if (page.name === 'note-work-docs') {
    var xmlhttp = new XMLHttpRequest();
    var url = "http://192.168.0.133/picker-server/note/office.php";
    xmlhttp.onreadystatechange=function() {
      if (xmlhttp.readyState === 4 && xmlhttp.status === 200) {
        myFunction(xmlhttp.responseText);
      }
    }
  }

  xmlhttp.open("GET", url, true);
```



```
xmlhttp.send();
```

```
function myFunction(response) {
    var arr = JSON.parse(response);
    var i;
    var ListHtml = "";
    for(i = 0; i < arr.length; i++) {

        ListHtml += '<li>'+ '<a href="#" class="item-link item-content">'+
            '<div class="item-media"></div>'+
            '<div class="item-inner">'+
            '<div class="item-title-row">'+
            '<div class="item-title dev-prod">'+
            arr[i].dev_prod + '</div>'+
            '<div class="item-after dev-price">'+
            arr[i].dev_price_tg + 'Т'+ '</div>'+ '</div>'+
            '<div id="dev-model" class="item-subtitle">'+
            arr[i].dev_model + '</div>'+
            '<div class="item-text dev-tiny-descr">'+
            arr[i].prod_name+'/' +arr[i].m_cpu_model+'/' +
            arr[i].m_cpu_freqMin_ghz+'/' +arr[i].m_ram_memory_gb+'GB'+
            '/' +arr[i].m_vid_model+'</div></div></a>'+
            '<div class="row">'+<div class="col-33">'+</div>'+
            '<div class="col-33">'+</div>'+
            '<div class="col-33">'+<a href="#buy" class="button">Купить</a>'+
            '</div>'+</div>'+</li>';

    }
    $$ (page.container).find('#cont').append(ListHtml);
}

};
```

## Приложение В

```
(function () {  
  
    'use strict';  
    /*=====*/  
    Framework 7  
    =====*/  
    window.Framework7 = function (params) {  
        // App  
        var app = this;  
  
        // Version  
        app.version = '1.4.2';  
  
        // Default Parameters  
        app.params = {  
            cache: true,  
            cacheIgnore: [],  
            cacheIgnoreGetParameters: false,  
            cacheDuration: 1000 * 60 * 10, // Ten minutes  
            preloadPreviousPage: true,  
            uniqueHistory: false,  
            uniqueHistoryIgnoreGetParameters: false,  
            dynamicPageUrl: 'content-{{index}}',  
            allowDuplicateUrls: false,  
            router: true,  
            // Push State  
            pushState: false,  
            pushStateRoot: undefined,  
            pushStateNoAnimation: false,  
            pushStateSeparator: '#!/',  
            pushStateOnLoad: true,  
            // Fast clicks  
            fastClicks: true,  
            fastClicksDistanceThreshold: 10,  
            fastClicksDelayBetweenClicks: 50,  
            // Tap Hold  
            tapHold: false,  
            tapHoldDelay: 750,  
            tapHoldPreventClicks: true,  
        }  
    }  
})
```

*Продолжение приложения В*

```
// Active State
  activeState: true,
  activeStateElements: 'a, button, label, span',
  // Animate Nav Back Icon
  animateNavBackIcon: false,
  // Swipe Back
  swipeBackPage: true,
  swipeBackPageThreshold: 0,
  swipeBackPageActiveArea: 30,
  swipeBackPageAnimateShadow: true,
  swipeBackPageAnimateOpacity: true,
  // Ajax
  ajaxLinks: undefined, // or CSS selector
  // External Links
  externalLinks: '.external', // CSS selector
  // Sortable
  sortable: true,
  // Scroll toolbars
  hideNavbarOnPageScroll: false,
  hideToolbarOnPageScroll: false,
  hideTabbarOnPageScroll: false,
  showBarsOnPageScrollEnd: true,
  showBarsOnPageScrollTop: true,
  // Swipeout
  swipeout: true,
  swipeoutActionsNoFold: false,
  swipeoutNoFollow: false,
  // Smart Select Back link template
  smartSelectOpenIn: 'page', // or 'popup' or 'picker'
  smartSelectBackText: 'Back',
  smartSelectPopupCloseText: 'Close',
  smartSelectPickerCloseText: 'Done',
  smartSelectSearchbar: false,
  smartSelectBackOnSelect: false,
  // Tap Navbar or Statusbar to scroll to top
  scrollTopOnNavbarClick: false,
  scrollTopOnStatusbarClick: false,
  // Panels
  swipePanel: false, // or 'left' or 'right'
  swipePanelActiveArea: 0,
```

## *Продолжение приложения В*

```
swipePanelCloseOpposite: true,  
swipePanelOnlyClose: false,  
swipePanelNoFollow: false,  
swipePanelThreshold: 0,  
panelsCloseByOutside: true,  
// Modals  
modalButtonOk: 'OK',  
modalButtonCancel: 'Cancel',  
modalUsernamePlaceholder: 'Username',  
modalPasswordPlaceholder: 'Password',  
modalTitle: 'Framework7',  
modalCloseByOutside: false,  
actionsCloseByOutside: true,  
popupCloseByOutside: true,  
modalPreloaderTitle: 'Loading... ',  
modalStack: true,  
// Lazy Load  
imagesLazyLoadThreshold: 0,  
imagesLazyLoadSequential: true,  
// Name space  
viewClass: 'view',  
viewMainClass: 'view-main',  
viewsClass: 'views',  
// Notifications defaults  
notificationCloseOnClick: false,  
notificationCloseIcon: true,  
notificationCloseButtonText: 'Close',  
// Animate Pages  
animatePages: true,  
// Template7  
templates: {},  
template7Data: {},  
template7Pages: false,  
precompileTemplates: false,  
// Material  
material: false,  
materialPageLoadDelay: 0,
```

## Продолжение приложения В

```
materialPreloaderSvg: '<svg xmlns="http://www.w3.org/2000/svg"
height="75" width="75" viewBox="0 0 75 75"><circle cx="37.5" cy="37.5"
r="33.5" stroke-width="8"/></svg>',
materialPreloaderHtml:
  '<span class="preloader-inner">' +
  '<span class="preloader-inner-gap"></span>' +
  '<span class="preloader-inner-left">' +
  '<span class="preloader-inner-half-circle"></span>' +
  '</span>' +
  '<span class="preloader-inner-right">' +
  '<span class="preloader-inner-half-circle"></span>' +
  '</span>' +
  '</span>',
materialRipple: true,
materialRippleElements: '.ripple, a.link, a.item-link, .button, .modal-
button, .tab-link, .label-radio, .label-checkbox, .actions-modal-button, a.searchbar-
clear, a.floating-button, .floating-button > a, .speed-dial-buttons a',
// Auto init
init: true,
};

// Extend defaults with parameters
for (var param in params) {
  app.params[param] = params[param];
}

// DOM lib
var $ = Dom7;

// Template7 lib
var t7 = Template7;
app._compiledTemplates = {};

// Touch events
app.touchEvents = {
  start: app.support.touch ? 'touchstart' : 'mousedown',
  move: app.support.touch ? 'touchmove' : 'mousemove',
  end: app.support.touch ? 'touchend' : 'mouseup'
};
```

*Продолжение приложения В*

```
// Link to local storage
app.ls = window.localStorage;

// RTL
app.rtl = $('body').css('direction') === 'rtl';
if (app.rtl) $('html').attr('dir', 'rtl');

// Overwrite statusbar overlay
if (typeof app.params.statusbarOverlay !== 'undefined') {
    if (app.params.statusbarOverlay) $('html').addClass('with-statusbar-
overlay');
    else $('html').removeClass('with-statusbar-overlay');
}

/*=====
***** Views *****
=====*/

app.views = [];
var View = function (selector, params) {
    var defaults = {
        dynamicNavbar: false,
        domCache: false,
        linksView: undefined,
        reloadPages: false,
        uniqueHistory: app.params.uniqueHistory,
        uniqueHistoryIgnoreGetParameters:
app.params.uniqueHistoryIgnoreGetParameters,
        allowDuplicateUrls: app.params.allowDuplicateUrls,
        swipeBackPage: app.params.swipeBackPage,
        swipeBackPageAnimateShadow:
app.params.swipeBackPageAnimateShadow,
        swipeBackPageAnimateOpacity:
app.params.swipeBackPageAnimateOpacity,
        swipeBackPageActiveArea:
app.params.swipeBackPageActiveArea,
        swipeBackPageThreshold: app.params.swipeBackPageThreshold,
        animatePages: app.params.animatePages,
```

## *Продолжение приложения В*

```
preloadPreviousPage: app.params.preloadPreviousPage
  };
  var i;

  // Params
  params = params || {};

  // Disable dynamic navbar for material theme
  if (params.dynamicNavbar && app.params.material)
params.dynamicNavbar = false;

  // Extend params with defaults
  for (var def in defaults) {
    if (typeof params[def] === 'undefined') {
      params[def] = defaults[def];
    }
  }
  // View
  var view = this;
  view.params = params;

  // Selector
  view.selector = selector;

  // Container
  var container = $(selector);
  view.container = container[0];

  // Fix Selector

  if (typeof selector !== 'string') {
    // Supposed to be HTMLElement or Dom7
    selector = (container.attr('id') ? '#' + container.attr('id') : '') +
(container.attr('class') ? '.' + container.attr('class').replace(/ /g, '.').replace('active', '')
: "");
    view.selector = selector;
  }

  // Is main
```

## *Продолжение приложения В*

```
view.main = container.hasClass(app.params.viewMainClass);

// Content cache
view.contentCache = {};

// Pages cache
view.pagesCache = {};

// Store View in element for easy access
container[0].f7View = view;

// Pages
view.pagesContainer = container.find('.pages')[0];
view.initialPages = [];
view.initialPagesUrl = [];
view.initialNavbars = [];
if (view.params.domCache) {
    var initialPages = container.find('.page');
    for (i = 0; i < initialPages.length; i++) {
        view.initialPages.push(initialPages[i]);
        view.initialPagesUrl.push('#' + initialPages.eq(i).attr('data-
page'));
    }
    if (view.params.dynamicNavbar) {
        var initialNavbars = container.find('.navbar-inner');
        for (i = 0; i < initialNavbars.length; i++) {
            view.initialNavbars.push(initialNavbars[i]);
        }
    }
}

view.allowPageChange = true;

// Location
var docLocation = document.location.href;

// History
view.history = [];
```



*Продолжение приложения В*

```
var viewURL = docLocation;
    var pushStateSeparator = app.params.pushStateSeparator;
    var pushStateRoot = app.params.pushStateRoot;
    if (app.params.pushState && view.main) {
        if (pushStateRoot) {
            viewURL = pushStateRoot;
        }
        else {
            if (viewURL.indexOf(pushStateSeparator) >= 0 &&
viewURL.indexOf(pushStateSeparator + '#') < 0) viewURL =
viewURL.split(pushStateSeparator)[0];
        }
    }

    // Active Page
    var currentPage, currentPageData;
    if (!view.activePage) {
        currentPage = $(view.pagesContainer).find('.page-on-center');
        if (currentPage.length === 0) {
            currentPage = $(view.pagesContainer).find('.page:not(.cached)');
            currentPage = currentPage.eq(currentPage.length - 1);
        }
        if (currentPage.length > 0) {
            currentPageData = currentPage[0].f7PageData;
        }
    }

    // View startup URL
    if (view.params.domCache && currentPage) {
        view.url = container.attr('data-url') || view.params.url || '#' +
currentPage.attr('data-page');
        view.pagesCache[view.url] = currentPage.attr('data-page');
    }
    else view.url = container.attr('data-url') || view.params.url || viewURL;

    // Update current page Data
    if (currentPageData) {
        currentPageData.view = view;
    }
}
```

*Продолжение приложения В*

```
currentPageData.url = view.url;
    if (view.params.domCache && view.params.dynamicNavbar &&
!currentPageData.navbarInnerContainer) {
        currentPageData.navbarInnerContainer =
view.initialNavbars[view.initialPages.indexOf(currentPageData.container)];
    }
    view.activePage = currentPageData;
    currentPage[0].f7PageData = currentPageData;
}

// Store to history main view's url
if (view.url) {
    view.history.push(view.url);
}

// Touch events
var isTouched = false,
    isMoved = false,
    touchesStart = {},
    isScrolling,
    activePage = [],
    previousPage = [],
    viewContainerWidth,
    touchesDiff,
    allowViewTouchMove = true,
    touchStartTime,
    activeNavbar = [],
    previousNavbar = [],
    activeNavElements,
    previousNavElements,
    activeNavBackIcon,
    previousNavBackIcon,
    dynamicNavbar,
    pageShadow,
    el;

view.handleTouchStart = function (e) {
    if (!allowViewTouchMove || !view.params.swipeBackPage ||
isTouched || app.swipeoutOpenedEl || !view.allowPageChange) return;
```

*Продолжение приложения В*

```
isMoved = false;
isTouched = true;
isScrolling = undefined;
touchesStart.x = e.type === 'touchstart' ? e.targetTouches[0].pageX
: e.pageX;
touchesStart.y = e.type === 'touchstart' ? e.targetTouches[0].pageY
: e.pageY;

touchStartTime = (new Date()).getTime();
dynamicNavbar = view.params.dynamicNavbar &&
container.find('.navbar-inner').length > 1;
};

view.handleTouchMove = function (e) {
  if (!isTouched) return;
  var pageX = e.type === 'touchmove' ? e.targetTouches[0].pageX :
e.pageX;
  var pageY = e.type === 'touchmove' ? e.targetTouches[0].pageY :
e.pageY;

  if (typeof isScrolling === 'undefined') {
    isScrolling = !(isScrolling || Math.abs(pageY - touchesStart.y) >
Math.abs(pageX - touchesStart.x));
  }
  if (isScrolling || e.f7PreventSwipeBack || app.preventSwipeBack) {
    isTouched = false;
    return;
  }
  if (!isMoved) {
    var cancel = false;
    // Calc values during first move fired
    viewContainerWidth = container.width();
    var target = $(e.target);
    var swipeout = target.hasClass('swipeout') ? target :
target.parents('.swipeout');
    if (swipeout.length > 0) {
      if (!app.rtl && swipeout.find('.swipeout-actions-left').length >
0) cancel = true;
      if (app.rtl && swipeout.find('.swipeout-actions-right').length >
0) cancel = true;
    }
  }
}
```

## Продолжение приложения В

```
activePage = target.is('.page') ? target : target.parents('.page');
    if (activePage.hasClass('no-swipeback')) cancel = true;
    previousPage = container.find('.page-on-left:not(.cached)');
    var notFromBorder = touchesStart.x - container.offset().left >
view.params.swipeBackPageActiveArea;
    if (app.rtl) {
        notFromBorder = touchesStart.x < container.offset().left -
container[0].scrollLeft + viewContainerWidth -
view.params.swipeBackPageActiveArea;
    }
    else {
        notFromBorder = touchesStart.x - container.offset().left >
view.params.swipeBackPageActiveArea;
    }
    if (notFromBorder) cancel = true;
    if (previousPage.length === 0 || activePage.length === 0) cancel
= true;

    if (cancel) {
        isTouched = false;
        return;
    }

    if (view.params.swipeBackPageAnimateShadow &&
!app.device.android) {
        pageShadow = activePage.find('.swipeback-page-shadow');
        if (pageShadow.length === 0) {
            pageShadow = $('<div class="swipeback-page-
shadow"></div>');
            activePage.append(pageShadow);
        }
    }

    if (dynamicNavbar) {
        activeNavbar = container.find('.navbar-on-
center:not(.cached)');
        previousNavbar = container.find('.navbar-on-
left:not(.cached)');
        activeNavElements = activeNavbar.find('.left, .center, .right,
.subnavbar, .fading');
```

*Продолжение приложения В*

```
previousNavElements = previousNavbar.find('.left, .center, .right, .subnavbar,
.fading');
    if (app.params.animateNavBackIcon) {
        activeNavBackIcon = activeNavbar.find('.left.sliding .back
.icon');
        previousNavBackIcon = previousNavbar.find('.left.sliding
.back .icon');
    }
}

// Close/Hide Any Picker
if ($('.picker-modal.modal-in').length > 0) {
    app.closeModal($('.picker-modal.modal-in'));
}
}
e.f7PreventPanelSwipe = true;
isMoved = true;
e.preventDefault();

// RTL inverter
var inverter = app.rtl ? -1 : 1;

// Touches diff
touchesDiff = (pageX - touchesStart.x -
view.params.swipeBackPageThreshold) * inverter;
if (touchesDiff < 0) touchesDiff = 0;
var percentage = touchesDiff / viewContainerWidth;

// Swipe Back Callback
var callbackData = {
    percentage: percentage,
    activePage: activePage[0],
    previousPage: previousPage[0],
    activeNavbar: activeNavbar[0],
    previousNavbar: previousNavbar[0]
};
if (view.params.onSwipeBackMove) {
    view.params.onSwipeBackMove(callbackData);
}
```

## Продолжение приложения В

```
container.trigger('swipeBackMove', callbackData);

// Transform pages
var activePageTranslate = touchesDiff * inverter;
var previousPageTranslate = (touchesDiff / 5 - viewContainerWidth
/ 5) * inverter;
if (app.device.pixelRatio === 1) {
    activePageTranslate = Math.round(activePageTranslate);
    previousPageTranslate = Math.round(previousPageTranslate);
}

activePage.transform('translate3d(' + activePageTranslate +
'px,0,0)');
if (view.params.swipeBackPageAnimateShadow &&
!app.device.android) pageShadow[0].style.opacity = 1 - 1 * percentage;

previousPage.transform('translate3d(' + previousPageTranslate +
'px,0,0)');
if (view.params.swipeBackPageAnimateOpacity)
previousPage[0].style.opacity = 0.9 + 0.1 * percentage;

// Dynamic Navbars Animation
if (dynamicNavbar) {
    var i;
    for (i = 0; i < activeNavElements.length; i++) {
        el = $(activeNavElements[i]);
        if (!el.is('.subnavbar.sliding')) el[0].style.opacity = (1 -
percentage * 1.3);
        if (el[0].className.indexOf('sliding') >= 0) {
            var activeNavTranslate = percentage *
el[0].f7NavbarRightOffset;
            if (app.device.pixelRatio === 1) activeNavTranslate =
Math.round(activeNavTranslate);
            el.transform('translate3d(' + activeNavTranslate + 'px,0,0)');
            if (app.params.animateNavBackIcon) {
                if (el[0].className.indexOf('left') >= 0 &&
activeNavBackIcon.length > 0) {
                    activeNavBackIcon.transform('translate3d(' + -
activeNavTranslate + 'px,0,0)');
```

*Продолжение приложения В*

```
for (i = 0; i < previousNavElements.length; i++) {
  el = $(previousNavElements[i]);
  if (!el.is('.subnavbar.sliding')) el[0].style.opacity = percentage
* 1.3 - 0.3;
  if (el[0].className.indexOf('sliding') >= 0) {
    var previousNavTranslate = el[0].f7NavbarLeftOffset * (1 -
percentage);
    if (app.device.pixelRatio === 1) previousNavTranslate =
Math.round(previousNavTranslate);
    el.transform('translate3d(' + previousNavTranslate +
'px,0,0)');
    if (app.params.animateNavBackIcon) {
      if (el[0].className.indexOf('left') >= 0 &&
previousNavBackIcon.length > 0) {
        previousNavBackIcon.transform('translate3d(' + -
previousNavTranslate + 'px,0,0)');
      }
    }
  }
};

view.handleTouchEnd = function (e) {
  if (!isTouched || !isMoved) {
    isTouched = false;
    isMoved = false;
    return;
  }
  isTouched = false;
  isMoved = false;
  if (touchesDiff === 0) {
    $([activePage[0], previousPage[0]]).transform("").css({ opacity: "",
boxShadow: "" });
    if (dynamicNavbar) {
      activeNavElements.transform("").css({ opacity: "" });
      previousNavElements.transform("").css({ opacity: "" });
      if (activeNavBackIcon && activeNavBackIcon.length > 0)
activeNavBackIcon.transform("");
    }
  }
};
```

*Продолжение приложения В*

```
    if (previousNavBackIcon && activeNavBackIcon.length > 0)
previousNavBackIcon.transform("");
        }
        return;
    }
    var timeDiff = (new Date()).getTime() - touchStartTime;
    var pageChanged = false;
    // Swipe back to previous page
    if (
        timeDiff < 300 && touchesDiff > 10 ||
        timeDiff >= 300 && touchesDiff > viewContainerWidth / 2
    ) {
        activePage.removeClass('page-on-center').addClass('page-on-
right');
        previousPage.removeClass('page-on-left').addClass('page-on-
center');
        if (dynamicNavbar) {
            activeNavbar.removeClass('navbar-on-
center').addClass('navbar-on-right');
            previousNavbar.removeClass('navbar-on-
left').addClass('navbar-on-center');
        }
        pageChanged = true;
    }
    // Reset custom styles
    // Add transitioning class for transition-duration
    $([activePage[0], previousPage[0]]).transform("").css({ opacity: ",
boxShadow: "}).addClass('page-transitioning');
    if (dynamicNavbar) {
        activeNavElements.css({ opacity: "})
        .each(function () {
            var translate = pageChanged ? this.f7NavbarRightOffset : 0;
            var sliding = $(this);
            sliding.transform('translate3d(' + translate + 'px,0,0)');
            if (app.params.animateNavBackIcon) {
                if (sliding.hasClass('left') && activeNavBackIcon.length >
0) {
                    activeNavBackIcon.addClass('page-
transitioning').transform('translate3d(' + -translate + 'px,0,0)');
```



*Продолжение приложения В*

```
    }  
    }  
  
    }).addClass('page-transitioning');  
  
    previousNavElements.transform("").css({ opacity:  
}).each(function () {  
    var translate = pageChanged ? 0 : this.f7NavbarLeftOffset;  
    var sliding = $(this);  
    sliding.transform('translate3d(' + translate + 'px,0,0)');  
    if (app.params.animateNavBackIcon) {  
        if (sliding.hasClass('left') && previousNavBackIcon.length  
> 0) {  
            previousNavBackIcon.addClass('page-  
transitioning').transform('translate3d(' + -translate + 'px,0,0)');  
        }  
    }  
    }).addClass('page-transitioning');  
    }  
    allowViewTouchMove = false;  
    view.allowPageChange = false;  
    // Swipe Back Callback  
    var callbackData = {  
        activePage: activePage[0],  
        previousPage: previousPage[0],  
        activeNavbar: activeNavbar[0],  
        previousNavbar: previousNavbar[0]  
    };  
    if (pageChanged) {  
        // Update View's URL  
        var url = view.history[view.history.length - 2];  
        view.url = url;  
  
        // Page before animation callback  
        app.pageBackCallback('before', view, {pageContainer:  
activePage[0], url: url, position: 'center', newPage: previousPage, oldPage:  
activePage, swipeBack: true});
```

*Продолжение приложения В*

```
app.pageAnimCallback('before', view, {pageContainer: previousPage[0], url:
url, position: 'left', newPage: previousPage, oldPage: activePage, swipeBack: true});

    if (view.params.onSwipeBackBeforeChange) {
        view.params.onSwipeBackBeforeChange(callbackData);
    }
    container.trigger('swipeBackBeforeChange', callbackData);
}
else {
    if (view.params.onSwipeBackBeforeReset) {
        view.params.onSwipeBackBeforeReset(callbackData);
    }
    container.trigger('swipeBackBeforeReset', callbackData);
}

activePage.transitionEnd(function () {
    $([activePage[0], previousPage[0]]).removeClass('page-
transitioning');
    if (dynamicNavbar) {
        activeNavElements.removeClass('page-
transitioning').css({opacity: ""});
        previousNavElements.removeClass('page-
transitioning').css({opacity: ""});
        if (activeNavBackIcon && activeNavBackIcon.length > 0)
activeNavBackIcon.removeClass('page-transitioning');
        if (previousNavBackIcon && previousNavBackIcon.length >
0) previousNavBackIcon.removeClass('page-transitioning');
    }
    allowViewTouchMove = true;
    view.allowPageChange = true;
    if (pageChanged) {
        if (app.params.pushState && view.main) history.back();
        // Page after animation callback
        app.pageBackCallback('after', view, {pageContainer:
activePage[0], url: url, position: 'center', newPage: previousPage, oldPage:
activePage, swipeBack: true});
        app.pageAnimCallback('after', view, {pageContainer:
previousPage[0], url: url, position: 'left', newPage: previousPage, oldPage:
activePage, swipeBack: true});
    }
}
```

## *Продолжение приложения В*

```
app.router.afterBack(view, activePage, previousPage);

    if (view.params.onSwipeBackAfterChange) {
        view.params.onSwipeBackAfterChange(callbackData);
    }
    container.trigger('swipeBackAfterChange', callbackData);
}
else {
    if (view.params.onSwipeBackAfterReset) {
        view.params.onSwipeBackAfterReset(callbackData);
    }
    container.trigger('swipeBackAfterReset', callbackData);
}
if (pageShadow && pageShadow.length > 0)
pageShadow.remove();
});
};
view.attachEvents = function (detach) {
    var action = detach ? 'off' : 'on';
    container[action](app.touchEvents.start, view.handleTouchStart);
    container[action](app.touchEvents.move, view.handleTouchMove);
    container[action](app.touchEvents.end, view.handleTouchEnd);
};
view.detachEvents = function () {
    view.attachEvents(true);
};

// Init
if (view.params.swipeBackPage && !app.params.material) {
    view.attachEvents();
}

// Add view to app
app.views.push(view);
if (view.main) app.mainView = view;

// Router
view.router = {
    load: function (options) {
```

```
return app.router.load(view, options);
    },
    back: function (options) {
        return app.router.back(view, options);
    },
    // Shortcuts
    loadPage: function (options) {
        options = options || {};
        if (typeof options === 'string') {
            var url = options;
            options = {};
            if (url && url.indexOf('#') === 0 && view.params.domCache)
            {
                options.pageName = url.split('#')[1];
            }
            else options.url = url;
        }
        return app.router.load(view, options);
    },
    loadContent: function (content) {
        return app.router.load(view, {content: content});
    },
    reloadPage: function (url) {
        return app.router.load(view, {url: url, reload: true});
    },
    reloadContent: function (content) {
        return app.router.load(view, {content: content, reload: true});
    },
    reloadPreviousPage: function (url) {
        return app.router.load(view, {url: url, reloadPrevious: true,
reload: true});
    },
    reloadPreviousContent: function (content) {
        return app.router.load(view, {content: content, reloadPrevious:
true, reload: true});
    },
    refreshPage: function () {
        var options = {
            url: view.url,
```

## Продолжение приложения В

```
        reload: true,
        ignoreCache: true
    };
    if (options.url && options.url.indexOf('#') === 0) {
        if (view.params.domCache && view.pagesCache[options.url])
        {
            options.pageName = view.pagesCache[options.url];
            options.url = undefined;
            delete options.url;
        }
        else if (view.contentCache[options.url]) {
            options.content = view.contentCache[options.url];
            options.url = undefined;
            delete options.url;
        }
    }
    return app.router.load(view, options);
},
refreshPreviousPage: function () {
    var options = {
        url: view.history[view.history.length - 2],
        reload: true,
        reloadPrevious: true,
        ignoreCache: true
    };
    if (options.url && options.url.indexOf('#') === 0 &&
view.params.domCache && view.pagesCache[options.url]) {
        options.pageName = view.pagesCache[options.url];
        options.url = undefined;
        delete options.url;
    }
    return app.router.load(view, options);
}
};

// Aliases for temporary backward compatibility
view.loadPage = view.router.loadPage;
view.loadContent = view.router.loadContent;
view.reloadPage = view.router.reloadPage;
```

## *Продолжение приложения В*

```
view.reloadContent = view.router.reloadContent;
view.reloadPreviousPage = view.router.reloadPreviousPage;
view.reloadPreviousContent = view.router.reloadPreviousContent;
view.refreshPage = view.router.refreshPage;
view.refreshPreviousPage = view.router.refreshPreviousPage;
view.back = view.router.back;

// Bars methods
view.hideNavbar = function () {
    return app.hideNavbar(container.find('.navbar'));
};
view.showNavbar = function () {
    return app.showNavbar(container.find('.navbar'));
};
view.hideToolbar = function () {
    return app.hideToolbar(container.find('.toolbar'));
};
view.showToolbar = function () {
    return app.showToolbar(container.find('.toolbar'));
};

// Push State on load
if (app.params.pushState && app.params.pushStateOnLoad &&
view.main) {
    var pushStateUrl;
    var pushStateUrlSplit = docLocation.split(pushStateSeparator)[1];
    if (pushStateRoot) {
        pushStateUrl = docLocation.split(app.params.pushStateRoot +
pushStateSeparator)[1];
    }
    else if (pushStateSeparator &&
docLocation.indexOf(pushStateSeparator) >= 0 &&
docLocation.indexOf(pushStateSeparator + '#') < 0) {
        pushStateUrl = pushStateUrlSplit;
    }
    var pushStateAnimatePages = app.params.pushStateNoAnimation ?
false : undefined;
    var historyState = history.state;
    if (pushStateUrl) {
```

## Продолжение приложения В

```
    if (pushStateUrl.indexOf('#') >= 0 && view.params.domCache &&
historyState && historyState.pageName && 'viewIndex' in historyState) {
        app.router.load(view, {pageName: historyState.pageName,
url: historyState.url, animatePages: pushStateAnimatePages, pushState: false});
    }
    else if (pushStateUrl.indexOf('#') >= 0 &&
view.params.domCache && view.initialPagesUrl.indexOf(pushStateUrl) >= 0) {
        app.router.load(view, {pageName: pushStateUrl.replace('#', ''),
animatePages: pushStateAnimatePages, pushState: false});
    }
    else app.router.load(view, {url: pushStateUrl, animatePages:
pushStateAnimatePages, pushState: false});
    }
    else if (view.params.domCache &&
docLocation.indexOf(pushStateSeparator + '#') >= 0) {
        if (historyState && historyState.pageName && 'viewIndex' in
historyState) {
            app.router.load(view, {pageName: historyState.pageName,
url: historyState.url, animatePages: pushStateAnimatePages, pushState: false});
        }
        else if (pushStateSeparator && pushStateUrlSplit.indexOf('#')
=== 0) {
            if (view.initialPagesUrl.indexOf(pushStateUrlSplit)) {
                app.router.load(view, {pageName:
pushStateUrlSplit.replace('#', ''), animatePages: pushStateAnimatePages, pushState:
false});
            }
        }
    }
}

// Destroy
view.destroy = function () {
    view.detachEvents();
    view = undefined;
};

// Plugin hook
app.pluginHook('addView',
```