

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский государственный профессионально-педагогический  
университет»  
Институт инженерно-педагогического образования  
Кафедра информационных систем и технологий

**МОБИЛЬНОЕ ПРИЛОЖЕНИЕ  
ДЛЯ СЕТИ ПАРИКМАХЕРСКИХ  
«БАРБЕРШОП»**

Выпускная квалификационная работа  
по направлению подготовки 09.03.03 Прикладная информатика  
профилю подготовки «Прикладная информатика в экономике»

Идентификационный номер ВКР: 320

Екатеринбург

2018

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Российский государственный профессионально-педагогический  
университет»  
Институт инженерно-педагогического образования  
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ:  
Заведующий кафедрой ИС  
\_\_\_\_\_ Н.С. Толстова  
« \_\_\_\_ » \_\_\_\_\_ 2018 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  
**МОБИЛЬНОЕ ПРИЛОЖЕНИЕ**  
**ДЛЯ СЕТИ ПАРИКМАХЕРСКИХ «БАРБЕРШОП»**

Идентификационный код ВКР: 320

Исполнитель:

студент группы № ИЭ-401п

А. О. Белоусов

Руководитель:

канд.пед.наук, доцент

И. А. Сулова

Нормоконтролер:

С. Ю. Ярина

Екатеринбург

2018

## АННОТАЦИЯ

Выпускная квалификационная работа состоит из разработанного мобильного приложения для платформы iOS и пояснительной записки на 54 страницах, содержащей 14 рисунков, 6 таблиц, 31 источников литературы, а также 1 приложение на одном листе.

Ключевые слова: МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, IOS, SWIFT, СЕТЬ ПАРИКМАХЕРСКИХ, БАРБЕРШОП.

**Белоусов А. О., Мобильное приложение для сети парикмахерских «Барбершоп»:** выпускная квалификационная работа / А. О. Белоусов; Рос. гос. проф.-пед. ун-т, Ин-т инж.-пед. образования, каф. информ. систем и технологий. — Екатеринбург, 2018. — 54 с.

ВКР состоит в изучении современных технологий и средств разработки мобильных приложений (Xcode, Swift, Autolayout, HTTP Client), в выборе оптимального средства реализации проекта, самой реализации и документировании проделанной работы в пояснительной записке.

Выбранный заказчик нуждался в средстве для удобной записи и информирования клиентов барбершопа. Запись клиентов происходила по телефону и, чтобы напомнить клиенту о ближайшей записи, администратору приходилось делать звонок. Данный способ обработки клиентов не устраивал заказчика, поэтому было принято решение разработать мобильное приложение в связке с сайтом.

Данная выпускная квалификационная работа посвящена разработке мобильного приложения для платформы iOS используя язык программирования Swift от составления технического задания заказчиком до реализации конечного продукта. Целью работы: разработать мобильное приложение для сети парикмахерских «Барбершоп».

Произведена оценка использованных и исследованных подходов. Определены основные преимущества и недостатки созданного решения.

# СОДЕРЖАНИЕ

Введение.....	5
1 Обзорно–аналитическая часть .....	7
1.1 Характеристика барбершопов.....	7
1.1.1 История появления барбершопов.....	7
1.1.2 Рынок барбершопов в России.....	9
1.2 Обзор мобильных операционных систем.....	10
1.3 Обоснование выбора архитектуры приложения.....	11
1.4 Обоснование выбора языка разработки для создания мобильного приложения.....	13
1.5 Описание преимуществ нативной разработки.....	15
1.6 Обзор аналогичных мобильных приложений.....	18
2 Проектная часть.....	20
2.1 Требования заказчика к продукту .....	20
2.2 Проектирование приложения.....	21
2.2.1 Описание макета приложения .....	21
2.2.2 Формирование слоя данных — Model.....	21
2.2.3 Описание API–сервера .....	24
2.2.4 Сервис для записей клиентов уClients .....	26
2.2.5 Верстка приложения — Слой View .....	26
2.2.6 Бизнес-логика приложения (Controller).....	27
2.3 SWOT-анализ.....	28
3 Описание мобильного приложения.....	30
3.1 Жизненный цикл разработки мобильного приложения.....	30
3.2 Особенности мобильной операционной системы .....	31
3.2 Пользовательский интерфейс .....	32
3.3 Описание этапа тестирования, апробирования, внедрения.....	38

3.4 Технические требования к проекту .....	39
4 Определение экономических затрат на разработку программного продукта .	40
4.1 Описание продукции .....	40
4.2 Оценка затрат на разработку программного продукта .....	41
4.3 Расчет годового фонда времени работы на персональном компьютере	44
4.4 Предполагаемая цена программного продукта с учетом нормы прибыли .....	47
Заключение .....	49
Список использованных источников .....	50
Приложение .....	54

## **ВВЕДЕНИЕ**

Сейчас каждый человек, независимо от социального положения и статуса пользуется сферой услуг. Поэтому такая обыденная услуга, как стрижка волос стала привычным делом. Обычно люди стригутся в салонах красоты и парикмахерских. Но в нынешнее время, мужчинам надоело стричься в стандартных салонах красоты, где рядом ведут разговоры женщины, когда им делают укладку, поэтому все больше мужчин меняют салоны унисекс на такого рода заведения как барбершоп. Чаще всего такие парикмахерские выглядят похожими на мужской гараж или, например, какой-то чисто мужской бар, где деревянный пол и шкура медведя висит на стене.

В наше время рынок барбершопов или мужских-парикмахерских находится на этапе бурного роста. За последние 5 лет в городе Екатеринбург появилось большое количество таких салонов. Одним из первых и самых популярных на данный момент является «KONTORA Barbershop».

В большинстве случаев запись на услуги производится непосредственно через телефонный звонок. В нынешнее время, когда каждый человек владеет смартфоном, набирают популярность мобильные приложения. По определению, мобильное приложение есть программное обеспечение, предназначенное для работы на смартфонах, планшетных компьютерах и на других мобильных устройствах, устанавливаемое под необходимую операционную систему. Одними из популярных операционных систем для мобильных устройств являются iOS, Android и Windows.

Актуальность данной работы связана с возникновением потребности компании в разработке нового программного обеспечения, которое позволяет записаться клиентам на услуги, узнать последние новости о компании, узнать интересующую информацию о барберах.

**Объектом исследования** является общество с ограниченной ответственностью (ООО) «Барбершоп», занимающееся предоставлением услуг парикмахерскими и салонами красоты.

**Предметом исследования** является процесс записи клиентов на услуги.

**Цель работы:** разработать мобильное приложение для сети парикмахерских «Барбершоп».

В соответствии с поставленной целью необходимо решить следующие задачи:

- провести исследование предметной области;
- проанализировать аналогичные мобильные приложения;
- разработать модель программного продукта для автоматизации процесса записи клиентов на услуги в компанию ООО «Барбершоп»;
- реализовать мобильное приложение;
- провести тестирование приложения на реальных пользователях.

# 1 ОБЗОРНО–АНАЛИТИЧЕСКАЯ ЧАСТЬ

## 1.1 Характеристика барбершопов

### 1.1.1 История появления барбершопов

Барбершоп — это исключительно мужская парикмахерская. Все услуги в таком месте только для мужчин. В таком заведении вам сделают мужскую стрижку или аккуратно подравняют бороду.

Первые чисто мужские парикмахерские появились в Америке и Европе примерно в 18 веке, уже тогда мастера стрижек поделились на два направления: одни работали с женщинами, другие же предпочли чисто мужские услуги: укладки и стрижки волос, стрижки бороды и усов, которые в то время были очень популярны. Название «барбершоп», происходит от латинского слова «Barba», что означает — борода. Отличительной чертой мужских парикмахерских являлось полное отсутствие женщин, как среди клиентов, так и среди мастеров. Более того, в те времена в барбершопах так же проводили незначительные медицинские манипуляции. Это было удаление зубов, банки, мелкие хирургические операции, а также очень популярная на то время процедура кровопускания. Именно кровопускание символизирует знаменитый барберпол который можно встретить сейчас в каждом таком заведении. Это вращающаяся трехцветная труба, синий цвет символизирует вены, красный — кровь, а белый — это цвет стерильности. Даже во времена запрета на проведение таких манипуляций цирюльники продолжали проводить их, особенно в небольших городах, где медицина была не развита. Практика подобных манипуляций была прекращена примерно в 1850 году, тогда мужские мастера оставили в своих услугах только стрижки волос на голове и их укладки, стрижки бороды и усов. К концу 19 века парикмахерские развивались бешеными темпами, уже в каждом, даже не большом городке можно было встретить цирюльню. В 1886 году был основан профсоюз барберов —



Barbers' Protective Union en Columbus, Ohio, и в 1887 году Journeymen Barbers International Union in Buffalo, New York. Вместе с появлением профсоюзов появились и школы парикмахерского искусства, в которых проводились обучения. Уже в начале 20 века к объединениям цирюльников подключились анатомы, биологи и химики, все вместе они трудились над уникальными средствами по уходу за волосами, ускорителями роста бороды, а барберы применяли их новшества на практике.

Примерно в 1970-х годах барбершопы получили небольшой упадок в связи со стремительно развивающейся модой на ношение мужчинами длинных волос, истинные цирюльники просто не умели работать с ними. Любители длинных волос обращались за стрижками в женские парикмахерские.

Мода на бороды внезапно возникла в начале десятих. 2013 год был даже признан «годом бороды». А, как известно, спрос рождает предложение. Если в 2014 году в Москве было всего 4 барбершопа, то в 2016 их количество уже превышает полсотни. Откуда же дует ветер? По большому счету, во всех культурных смыслах, в том числе и в моде, мы находимся в эпохе пост модернизма. Это значит, что на сегодняшний день принципиально чего-то нового не создается. Мы как бы оглядываемся на то, что было ранее и используем это сегодня, немного переработав. Т.е. все тенденции приходят к нам из прошлого. Причем в плане мужского стиля это долгоиграющие тренды — от 5 до 10, а порой и до 15 лет. Борода на протяжении всей истории развития общества то входила в моду, то теряла позиции. Видимо пришло ее время. И если верить экспертам, популярная в прошлом «метросексуальность» порядком всем надоела и захотелось чего-то «ретросексуального». Мужчины стали стремиться носить щетину или бороду, подчеркивая этим свою мужественность. Этот мировой тренд привел к тому, что западные эксперты отметили спад объема продаж лезвий и бритв на 7%. Психологи считают, что борода помогает мужчине легче преодолеть экономические и социальные трудности. А некоторые женщины считают, что мужчины с ухоженной бородой создают впечатление надежного тыла. В любом случае, пока мода на бороды спадать

не собирается, так что у брадобреев и барбершопов есть время побыть на волне популярности.

Барбершопы и барберинг называют культурой, а сами мастера считают это образом жизни. Цирюльники постоянно развиваются в своем деле, посещают и сами устраивают мастер классы, показательные выступления и соревнования повышая уровень своих навыков. Барбер — это не просто парикмахер, это хороший собеседник, который готов поддержать разговор на любую тему. Барбер всегда даст рекомендации по уходу за волосами, посоветует средства и укажет на недостатки. Посещение цирюльни становится ритуальным событием целью, которой уже есть не только качественная стрижка, но и общение. Прежде всего, это мужские клубы, где собираются хорошие компании, устраивают тусовки, слушают музыку, выпивают и общаются.

### **1.1.2 Рынок барбершопов в России**

За последние несколько лет по всей России открылось более 1000 барбершопов. Большее количество барбершопов, конечно, насчитывается в Москве и Санкт-Петербурге. В Москве сейчас примерно около 500 мужских-парикмахерских, а в Санкт-Петербурге примерно 200.

Именно сейчас культура качественной прически и ухоженной бороды в России набирает большие обороты. Такой тренд появился относительно недавно, примерно в 2011 году. В этом же году и открылась первая в России мужская парикмахерская, ею стала компания «Chop-Chop».

Сейчас в среднем по России в городе от 100 тысяч человек до 200 тысяч расположено 4 — 5 барбершопов. В городе от 200 тысяч человек до 500 тысяч примерно 7 — 8. А более 500 тысяч — от 10 и более.

#### **Барбершопы в городе Екатеринбург**

На текущее время в городе Екатеринбурге насчитывается около 50 салонов мужских парикмахерских. Родоначальником уральских барбершопов является «KONTORA Barbershop». Первая «KONTORA» открылась в городе Екатеринбурге в июне 2013 года. Это было ноу-хау для ураль-

ской индустрии красоты, начало новой эпохи мужских стрижек. Через месяц в городе запустится франшиза федеральной сети «Chop-Chop», и еще через год начнется настоящий барбербум.

## **1.2 Обзор мобильных операционных систем**

Мобильная операционная система — это операционная система для смартфонов, планшетов, карманных персональных компьютеров (КПК) и других мобильных устройств, сочетающая в себе функциональность операционной системы (ОС) для персонального компьютера с функциями для мобильных и карманных устройств, такими как сенсорный экран, сотовая связь, камера, диктофон, Wi-Fi, музыкальный плеер и другие.

Современными операционными системами являются iOS, Android, Windows Phone, Leneage OS, Fire OS, Tizen. Список лидеров на рынке мобильных платформ остается неизменным в течение последних нескольких лет: iOS, Android, Windows Phone.

Apple iOS — мобильная операционная система, разработанная компанией Apple на основе macOS для смартфонов iPhone, медиаплееров iPod Touch и планшетных компьютеров iPad. Была выпущена 29 июня 2007 года, регулярно обновляется.

Разработка под iOS предполагает наличие техники Apple и программы разработки Xcode, либо использование сторонних сервисов для удаленной компиляции приложений. Публикация происходит при наличии сертификатов от App Store — магазина приложений. Приложения, неподходящие по качеству или контенту, не допускаются к публикации в App Store.

Android — программная платформа для мобильных устройств, разработанная компанией Google. Включает в себя операционную систему, базирующуюся на ядре Linux 2.6, связующее программное обеспечение (ПО) и ключевые приложения. Распространяется под лицензией Apache 2.0. Первая

версия была выпущена 23 сентября 2008 года. Постоянно обновляется. Является самой распространенной в мире.

Android позволяет создавать Java-приложения с помощью программы Android Studio. Для публикации приложений используется Google Play.

Windows Phone — мобильная операционная система, разработанная компанией Microsoft. Появилась 11 октября 2010 года. Является преемником Windows Mobile, хотя и не совместима с ней.

Для создания приложений требуется среда разработки Visual Studio (Community, Professional, Ultimate) с пакетом обновлений не ниже SP2. Публикация приложений осуществляется в Windows Store.

### **Обоснование выбора мобильной операционной системы**

В планах у заказчика разработка мобильного приложения под 2 платформы, iOS и Android. Пожеланием заказчика было в первую очередь выпустить приложение под iOS. Приоритетом для заказчика являлась реализация мобильного приложения под iOS, затем под Android. В рамках дипломной работы была выбрана разработка только на iOS, так как одним из условий заказчика была разработка на нативном для платформы языке программирования. Кроссплатформенную разработку пришлось исключить.

### **1.3 Обоснование выбора архитектуры приложения**

Данные приложения часто обновляются, могут появиться новые услуги, барберы или товары, поэтому важно чтобы информация до пользователей доходила вовремя. Для этого в качестве основной архитектуры приложения выбрана клиент-серверная архитектура. Так, как если бы мы использовали хранение информации локально и обновление данных происходило за счет обновления приложения, пользователи получали бы обновления данных гораздо позже, чем они появились. Процесс выпуска обновления в магазине App Store занимает много времени и может быть вообще отклонен.

При использовании сервера в качестве системы администрирования, упрощается процесс заполнения данных, при этом администратор может добавлять и изменять информацию прямо с мобильного устройства. За счет этого появляется возможность добавлять новые места, находясь на улице рядом с ними, и далее тестировать их в приложении на мобильном устройстве в режиме разработчика. После проверки новые данные добавляются в базу данных на сервере, и клиенты могут сразу получить обновленную информацию.

Как было описано в предыдущем пункте, кроссплатформенная разработка не подходила для этого проекта. Для нативной разработки под iOS стандартом является архитектура, основанная на паттерне проектирования MVC (Model-View-Controller).

**Model-View-Controller** (MVC, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер») (рисунок 1) — схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер/

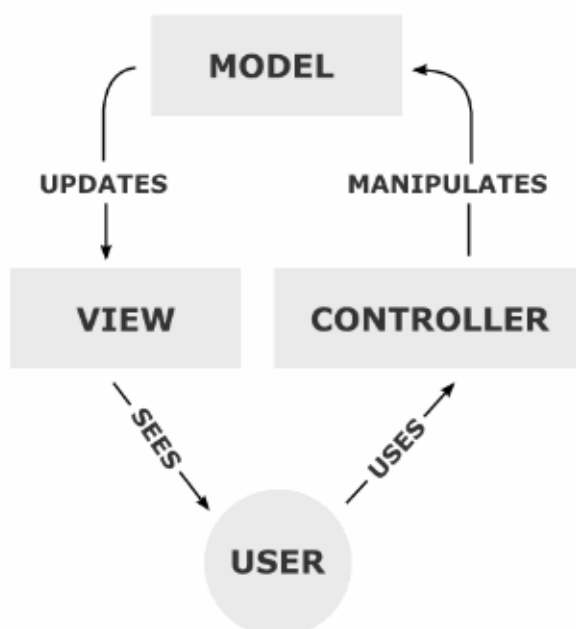


Рисунок 1 — Модель, представление и контроллер относительно пользователя

Таким образом, модификация каждого компонента может осуществляться независимо:

- модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние;
- представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели;
- контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

Данный паттерн подходит для большинства мобильных приложений, которые отображают информацию, полученную с сервера.

#### **1.4 Обоснование выбора языка разработки для создания мобильного приложения**

Рынок мобильных устройств с каждым годом достигает новых рекордных вершин. Мобильность — это один из основных драйверов ИТ-индустрии. По оценке экспертов Gartner Group и IEEE, облачные вычисления и мобильность являются лидерами списка из 10 наиболее значимых информационных технологий настоящего времени. Поэтому, несмотря на общую рецессию в сфере информационных технологий (ИТ), на отечественном рынке R&D (Research and Development), присутствует устойчивый спрос на услуги в сфере разработки мобильных приложений, что делает актуальным вопрос выбора среды разработки и языка программирования. В настоящей статье дается обоснование выбора языка разработки для создания мобильных приложений под iOS и macOS. Как следует из названия, за границами рассмотрения остаются open-source платформы разработки под Android и проприетарные системы, ориентированные на платформу Microsoft. Причиной ориентации на macOS-платформу, является учет одного из основных факторов риска, критичного для тиражируемых решений и систем заказной разработки проработанность модели монетизации проекта.

В рамках сравнения рассматривались 2 альтернативных варианта языка разработки: Swift или Objective-C. Компания Apple активно продвигает язык Swift и настоятельно рекомендует использовать его взамен Objective-C.

Swift имеет большое количество упрощений для разработки, что ускоряет сам процесс. Все популярные библиотеки и плагины переведены на Swift, к тому же использование библиотек, написанных на Objective-C стало возможным с появлением Cocoa Pods.

Ранее компания Apple сообщила, что рассматривают Swift, как удобный и мощный язык, который сможет заменить не только Objective-C в экосистеме Apple, но и сам класс C в целом. Теперь это утверждено на официальном сайте с описанием Swift. Стоит сразу отметить, что одним из недостатков Swift является его однозначная привязка к операционной системе macOS, в частности, в интегрированной среде разработки — Xcode. Помимо этого, нормальной документации с хорошо разобранными примерами на русском языке, на середину 2018 года, чрезвычайно мало.

И все же, все эти недостатки блекнут на фоне целого ряда преимуществ:

1. Swift более читаемый язык, чем Objective-C.
2. Swift легче поддерживать.
3. Swift обеспечивает более высокий уровень информационной безопасности, чем конкуренты.
4. Унифицировано согласование Swift с управлением памятью.
5. Swift обеспечивает более высокое быстродействие.
6. Меньше столкновений имен с проектами открытого исходного кода.
7. Swift поддерживает динамические библиотеки.

И еще один важный фактор в пользу Swift — это открытая система, на развитие которой можно повлиять. Разработчики Apple представили исходный код Swift'a, а также библиотеки Foundation, в своем репозитории на github. Это значит не только то, что в его доработке языка Apple стала силь-

нее опираться на сообщество разработчиков, но и то, что он становится мультиплатформенным. Другими словами, если вы разрабатываете приложение на Swift'е для iOS, то вы сможете перенести его на Android без всяких проблем и без знаний языка Java, используемый операционной системой от Google.

В заключение можно констатировать, что, несмотря на перечисленные недостатки, Swift обладает неоспоримыми достоинствами, которые позволяют позиционировать его, как ключевого игрока на рынке языков мобильной разработки. Swift, действительно заслуживает внимания, как основной язык приложения как под iOS, так и под macOS.

### **1.5 Описание преимуществ нативной разработки**

Под нативной (родной) разработкой подразумевается использование оригинальных языков и инструментов разработки мобильной операционной системы. Для iOS приложения создаются в среде разработки Xcode на языках Swift, Objective-C, C и C++. Для создания приложений под Android используется среда Android Studio и язык Java. Каждая среда разработки содержит целый комплекс утилит для написания кода, проектирования интерфейса, отладки, профилирования (мониторинга) и сборки приложений. И среда, и соответствующий набор утилит созданы специально под каждую мобильную операционную систему и являются максимально удобными и мощными средствами разработки мобильных приложений.

Разработка на родных технологиях и языках под iOS и Android имеет следующие положительные моменты:

1. Скорость работы приложения. Так как приложение создается с использованием оригинальных инструментов разработки (Xcode, Android Studio), получаемый в результате компиляции проекта код является оптимальным для данной платформы. Приложение получает полную аппаратную поддержку устройства (обработка тех же изображений осуществляется от-



дельным процессором, специально для этого предназначенным — GPU), используется многопоточность для реализации сложных задачи и загрузки контента в фоне, в процессе разработки программисты могут измерять скорость работы всех участков кода и при необходимости их оптимизировать, в их распоряжении также есть инструменты по мониторингу использования оперативной памяти, поиску возможных утечек и т.д.

2. Гибкость в реализации. В отличие от ограничений в построении интерфейса и сложности визуальных эффектов, накладываемых фреймворками для кроссплатформенной сборки проектов, в нативной разработке реализовать можно все, на что способны технологии той или иной мобильной операционной системы.

3. Использование последних технологий и зависимость от кроссплатформенных фреймворков. Новый программный и аппаратный функционал, предоставленный компаниями-производителями устройства и операционной системы, становится доступен для реализации сразу после выпуска соответствующих обновлений. К примеру, в iOS 9 была заложена возможность поиска внутри приложений: в каждом из них должен быть реализован специальный метод, который возвращает результаты по определенному поисковому запросу. В результате для тех приложений, в которых этот функционал реализован, доступна возможность поиска контента через системный раздел поиска в iOS, там же, где осуществляется поиск приложений, контактов, событий и прочей информации.

4. Легкость и качество тестирования. Помимо упомянутого инструментария для контроля использования приложением аппаратных ресурсов устройства в распоряжении разработчиков и тестировщиков есть целый комплекс технологий. Во-первых, все параметры системы в процессе работы приложения контролируются автоматически. Если приложение стало использовать больше памяти, чем это ожидается, или больше ресурсов центрального процессора, это не останется незамеченным. Во-вторых, возможности в широком применении юнит-тестов — автоматического тестирования практи-

чески каждого метода в приложении. Если какая-то часть приложения перестала работать корректно вследствие каких-либо изменений кода, новая версия просто не соберется, а программист сразу увидит причину. В-третьих, доступны широкие возможности в интеграции систем удаленного мониторинга ошибок: в каждый нативный проект встраивается соответствующий функционал, который позволяет увидеть ошибку и ее причину, возникшую на устройстве любого пользователя.

5. Полная поддержка со стороны магазинов приложений App Store и Google Play. Обе компании, Apple и Google, заинтересованы в том, чтобы пользователи получали максимально положительный опыт при использовании приложений на соответствующих платформах, который возможен на текущий момент. Это означает, что приложение должно выглядеть максимально качественно (если у экрана высокое разрешение, а изображения расплывчатые, в App Store приложение просто не пропустят), должно работать настолько быстро, насколько это возможно (если приложение отображает небольшой список элементов за 20–30 секунд, его так же не пропустят), и вообще все должно быть красиво и удобно. Если какие-то из этих параметров слишком низки или вообще не выполнены, приложение не пропустят в магазин. Если же они не на высоте, чего добиться с кроссплатформенными технологиями крайне сложно, а часто и невозможно в принципе, то ваше приложение никогда не будет рассмотрено соответствующими компаниями для размещения в специальных рекламных разделах (Featured). Среди приложений, находящихся во Featured-разделах и App Store, и Google Play, нет ни одного, сделанного с помощью кроссплатформенных технологий, за исключением игровых проектов, в которых интерфейс не является системным.

С технической точки зрения и с точки зрения качества создаваемого интерфейса нативная разработка имеет гораздо больше плюсов. Однако есть сферы, в которых кроссплатформенные технологии являются оправданными: это игровой сектор и тестовые проекты.

Если проект не является игровым, направлен на долгосрочное развитие и требует положительного впечатления от пользователей, нативная разработка остается более подходящим вариантом.

## 1.6 Обзор аналогичных мобильных приложений

В настоящее время для барбершопов разработано некоторое количество приложений. Ниже представлены скриншоты приложений, наиболее крупных игроков рынка (рисунок 2).

Функционал аналогичных мобильных приложений:

1. Возможность записи на услуги.
2. Возможность авторизации в профиле и просмотра записей на услуги.
3. Наличие информации, отзывов и контактов компании.
4. Возможность просмотра местоположения салонов.

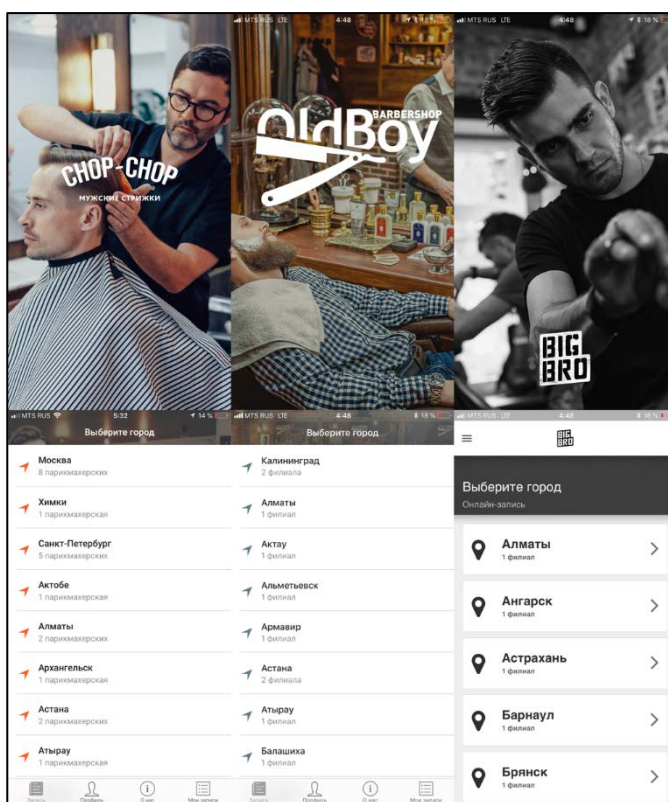


Рисунок 2 — Аналогичные мобильные приложения

Существующие аналоги имеют узкую функциональность, так как приложения однотипны, похожи и имеют одинаковые технологии разработки, а именно гибрид и PhoneGap. Большинство мобильных приложений в сфере барбершопов используют uClients.

## 2 ПРОЕКТНАЯ ЧАСТЬ

### 2.1 Требования заказчика к продукту

Разработка мобильных приложений разделяется на несколько этапов. Первым этапом является получение от заказчика требований и пожеланий для полного представления финального вида продукта. Следующий этап — создание примитивного макета, на котором расположены все экраны приложения с данными и переходами между страницами. Далее идет согласования макета с заказчиком и разработка архитектуры приложения, параллельно с этим идет подготовка макетов с финальным дизайном приложения. Конечным этапом разработки является верстка экранов и написание кода, реализующего весь функционал, указанный заказчиком.

При общении с заказчиком были выявлены следующие требования к приложению:

- разработка приложения для мобильной платформы iOS;
- разработка на нативном языке программирования для платформы iOS;
- на главном экране должна быть возможность переключения между пунктами: записи, блог, товары, услуги, барберы;
- использование сервиса уClients в приложении для записи в барбершоп и получения данных о текущих записях;
- отображение в приложении основных разделов, которые находятся на основном сайте;
- отображение статей в разделе блога, точно также как это выглядит в мобильной версии сайта, используя HTML Embed (внедрение HTML верстки в Web View).

## 2.2 Проектирование приложения

### 2.2.1 Описание макета приложения

На основе списка требований был спроектирован макет со всеми экранами и переходами (рисунок 3).

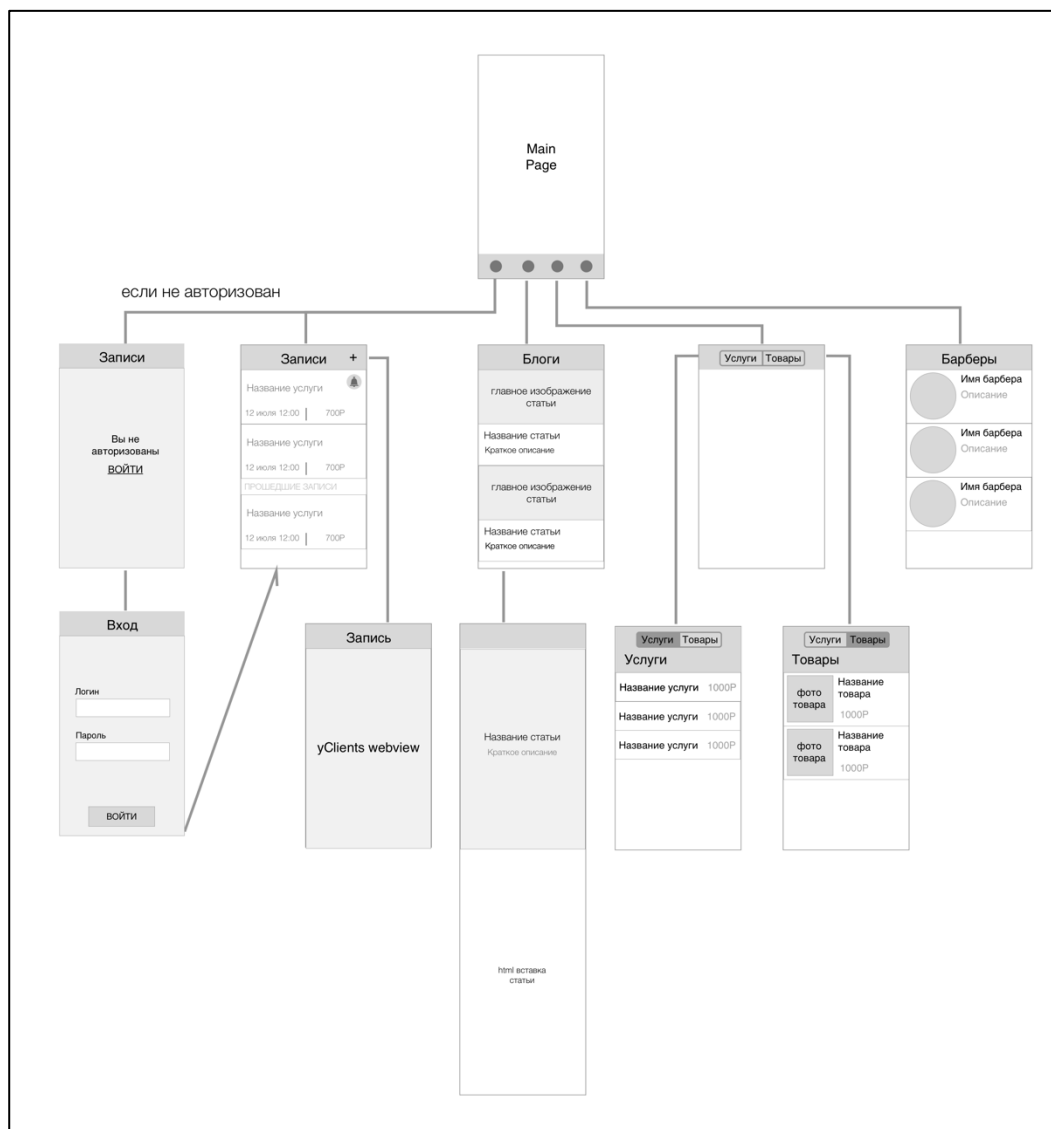


Рисунок 3 — Внешний вид структуры макета приложения

### 2.2.2 Формирование слоя данных — Model

После согласования макетов с заказчиком, необходимо спроектировать набор классов, которые будут хранить в себе необходимую информацию для экранов. В разработке мобильных приложений принят такой подход: сначала

создаются UX макеты, на которых показано, какая информация будет отображена на экране, дальше, исходя из макета, для каждого экрана описывается набор объектов, которые необходимы для полного отображения данных, чаще всего такие объекты называют DTO (Data Transfer Object). В паттерне MVC такой набор классов образует слой Model. API сервера для мобильного приложения создается специально под эти объекты, так как на каждом экране должен быть один запрос на получение данных. Если для отображение всей информации запросов требуется несколько запросов, за исключением загрузки картинок, то это является признаком некачественного проектирования API. Далее будут описаны классы DTO для каждого экрана в приложении.

### **MainPage**

Для основного экрана данные не требуются, так как он хранит в себе четыре самостоятельных страницы.

### **AppointmentPage**

Страница с записями. На ней имеется список посещений, для которых необходимо создать объект хранения AppointmentDTO, который состоит из следующих полей:

- String id — идентификационный номер записи;
- String serviceName — название услуги (стрижка мужская, оформление бороды/усов, уход за лицом);
- Decimal cost — стоимость услуги;
- DateTime date — дата записи.

### **BlogPage**

Страница со списком блогов. На этой странице есть список с кратким описанием статей, для которых необходимо создать объект хранения BlogDTO, который состоит из следующих полей:

1. String id — идентификационный номер статьи.
2. String title — название статьи.
3. String description — краткое описание.
4. NSUrl imageURL — ссылка на главную картинку статьи.

При открытии статьи в приложении происходит переход на страницу `AdvancedBlogPage`, на которой уже подгружается полная статья. На странице используются данные из `BlogDTO` и полные данные о статье, загруженные с сервера. Поэтому используется `AdvancedBlogDTO`, унаследованный от `BlogDTO`. Появляется еще одно поле: `NSString blogHTMLEmbedded` — HTML разметка статьи для отображения в `WebView`.

### **ServicePage**

Страница со списком услуг. Для хранения услуги необходимо создать объект `ServiceDTO`, который состоит из следующих полей:

1. `String id` — идентификационный номер услуги.
2. `String title` — название услуги.
3. `Int cost` — стоимость услуги.

### **MarketPage**

Страница со списком товаров, предоставляемых барбершопом. Для хранения товаров необходимо создать объект `MarketDTO`, который состоит из следующих полей:

1. `String id` — идентификационный номер товара.
2. `String title` — название товара.
3. `String Description` — описание товара.
4. `Int cost` — стоимость товара.
5. `NSURL` — ссылка на изображение товара.

### **BarbersPage**

Страница со списком барберов. Для хранения барберов необходимо создать объект `BarberDTO`, который состоит из следующих полей:

1. `String id` — идентификационный номер барбера.
2. `String name` — имя барбера.
3. `String position` — внутренняя должность барбера (шеф-барбер, старший барбер, барбер).
4. `String place` — адрес барбершопа, где он работает.
5. `NSURL` — ссылка на фотографию барбера.



## LoginPage

Страница авторизации. Для отправки данных для авторизации необходимо создать объект LoginDTO, который состоит из следующих полей:

1. String login — логин.
2. String password — пароль.

### 2.2.3 Описание API-сервера

Совместно с разработчиками сайта была спроектирована мобильная передача состояния представления (REST) интерфейса приложения (API) — Representational State Transfer Application Programming Interface на сервере. API представляет собой набор обычных HTTP-запросов (обычно «GET» или «POST»; такой запрос называют «*REST-запрос*»), а необходимые данные передаются в качестве параметров запроса. Ответ приходит в формате JSON (JavaScript Object Notation).

API разделено по категориям под каждый экран и состоит из запросов:

- LoginPage:
  - GET <http://barberkontora.ru/oAuth>: Username — параметр логин, Password — параметр пароль и ClientSecret — Id вшивается в приложение, необходим для реализации авторизации oAuth;
  - ответ: UserId — номер пользователя и AccessToken — токен для авторизованных запросов;
- AppointmentPage:
  - GET <http://barberkontora.ru/appointments>: AccessToken — необходим для получения записей именно авторизованного пользователя (получается при запросе авторизации);
  - ответ: список из объектов Appointment: ServiceName — название услуги, Cost — цена услуги, Date — дата записи, Id — идентификационный номер на сервере;

- BlogPage:
  - GET <http://barberkontora.ru/blogs>;
  - ответ: список из объектов Blog: Title — название статьи, Description — описание, ImageURL — ссылка на картинку, Id — идентификационный номер на сервере;
- AdvancedBlogPage:
  - GET [http://barberkontora.ru/blogs/id/{blog\\_id}](http://barberkontora.ru/blogs/id/{blog_id});
  - ответ: HTMLEmbeddedURL — ссылка на разметку статьи, Title — название статьи, Description — описание, ImageURL — ссылка на картинку, Id — идентификационный номер на сервере;
- MarketPage:
  - GET <http://barberkontora.ru/market>;
  - ответ: список из объектов Market: Title — название товара, Description — описание товара, Cost — цена товара, ImageURL — ссылка на картинку, Id — идентификационный номер на сервере;
- ServicePage:
  - GET <http://barberkontora.ru/services>;
  - ответ: список из объектов Service: Title — название услуги, Cost — цена товара, Id — идентификационный номер на сервере;
- BarberPage:
  - GET <http://barberkontora.ru/barbers>;
  - ответ: список из объектов Barber: Name — имя барбера, Position — должность барбера, Place — место работы, адрес, PhotoURL — ссылка на фото барбера, Id — идентификационный номер на сервере.

Данного набора запросов достаточно для отображения необходимой информации на всех экранах.

## **2.2.4 Сервис для записей клиентов уClients**

УClients автоматизирует онлайн-запись и администрирование компании. Виджет записи, с помощью которого клиенты будут выбирать удобные дату и время, можно установить на сайте, в мобильных приложениях и в социальных сетях, а его внешний вид корректируется.

Личный кабинет позволяет управлять расписанием, базой клиентов и программой лояльности, например, выявлять активных пользователей и выстраивать систему скидок. Также сервис предлагает отслеживать статистику — какая из услуг более востребована или какой сотрудник самый эффективной. Есть возможность автоматической рассылки напоминаний клиентам. Кроме того, сервис автоматизирует внутренние процессы: позволяет вести складской и финансовый учет, управлять филиалами сетевых компаний, рассчитывать зарплаты. УClients используют более 1300 компаний: салоны красоты, медицинские центры, квеструмы, автосервисы.

С разработкой нового сайта, заказчик внедрил сервис уClients. Экран «Запись» использует внутри себя отображение веб клиента уClients через Web View и создается, используя параметр уClientsURL, который передается в контроллер экрана после инициализации класса.

## **2.2.5 Верстка приложения — Слой View**

Верстка приложения создавалась в Xcode Designer с использованием AutoLayout, технология, которая позволяет создавать гибкую верстку под все размеры экранов сразу. Данный способ в отличии от верстки кодом, которая до недавнего времени являлась единственным решением для iOS, ощутимо ускоряет процесс и сразу виден результат работы. Есть возможность сразу настроить переходы между экранами (рисунок 4).

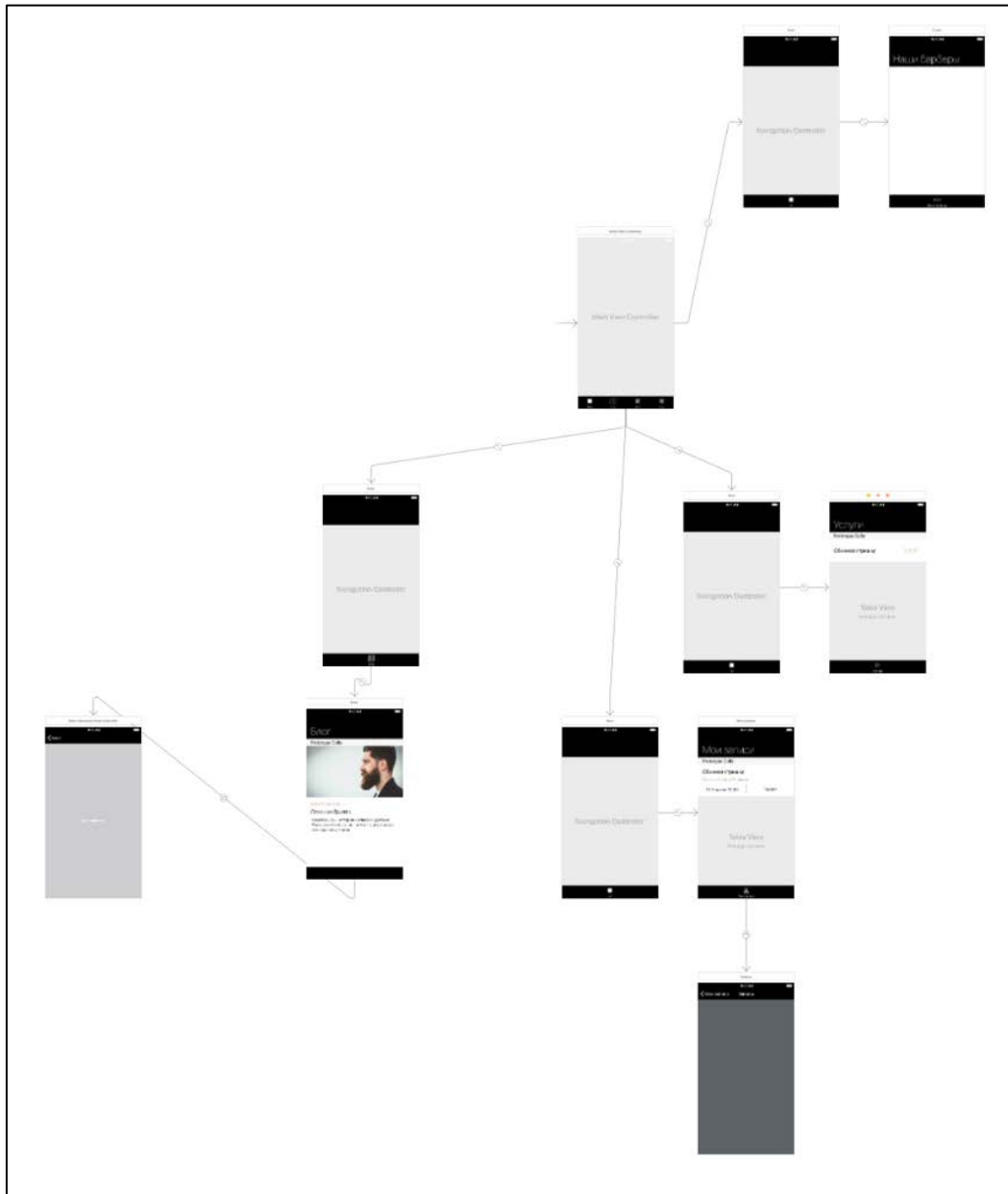


Рисунок 4 — Внешний вид переходов между экранами

## 2.2.6 Бизнес-логика приложения (Controller)

После верстки остается написание бизнес-логики, которая связывает Model и View. Контроллеры (слой Controller в паттерне MVC) создаются для каждого экрана, где есть определенная логика действий и отображение информации. Контроллер создается при открытии страницы. На тех страницах, где необходимо отобразить информацию сервера, делается запрос API, полученные данные конвертируются в объект DTO и дальше обновляются данные

на View части. В итоге в списках появляются элементы, картинки подгружаются и все это видно на экране.

### **2.3 SWOT-анализ**

Любое сегментирование начинается с всестороннего изучения рыночной ситуации, в которой работает компания, и оценки типов возможностей и угроз, с которыми она может столкнуться. Отправной точкой для подобного обзора служит SWOT-анализ, один из самых распространенных видов анализа в маркетинге. SWOT-анализ — анализ сильных и слабых сторон, возможностей и угроз является одним из важнейших этапов маркетингового плана. SWOT-анализ проводят как для деятельности фирмы в целом, так и для определенных продуктов, сегментов рынка, географических территорий. В первом случае анализ выходит достаточно общим и содержащим не так много полезной информации, а во втором дает менеджеру серьезную пищу для размышлений.

На данный момент основным недостатком деятельности сайта компании является отсутствие автоматизированного учета клиентов и мероприятий, а также ручное формирование документов, что заметно снижает скорость и эффективность работы.

Проще говоря, SWOT-анализ позволяет выявить и структурировать сильные и слабые стороны фирмы, а также потенциальные возможности и угрозы. Достигается это за счет того, что менеджеры должны сравнивать внутренние силы и слабости своей компании с возможностями, которые дает им рынок. Исходя из этого соответствия делается вывод о том, в каком направлении организация должна развивать свой бизнес и в конечном итоге определяется распределение ресурсов по сегментам.

Внедрение нового метода обработки информации позволит устранить слабые стороны и реализовать имеющиеся возможности.

SWOT-анализ текущего состояния информационной системы представлен в таблице 1.

Таблица 1 — SWOT-анализ предприятия до создания мобильного приложения.

Сильные внутренние стороны	Слабые внутренние стороны
1. Постоянные лояльные клиенты. 2. Узнаваемость в своем городе.	Отсутствие рекламы.
Возможности во внешней среде	Угрозы внешней среды
Возможности использования оффлайн рекламы	Снижение цен у конкурентов.

SWOT-анализ после создания мобильного приложения представлен в таблице 2.

Таблица 2 — SWOT-анализ предприятия после создание мобильного приложения.

Сильные внутренние стороны	Слабые внутренние стороны
1. Наличие нужной информации о предприятии. 2. Появление новых клиентов.	Низкий рейтинг мобильного приложения с момента создания.
Возможности во внешней среде	Угрозы внешней среды
1. Возможность использования новых видов рекламы (интернет-рекламы). 2. Использование средств интернет-продвижения мобильного приложения.	1. Конкуренты уже имеют мобильные приложения с большим рейтингом. 2. Сильная конкуренция среди существующих предприятий.

Таким образом, положительные и слабые стороны информационной обеспеченности сети парикмахерских, которые были обозначены в ходе проведенного анализа, дают возможность спланировать необходимые изменения, слабые стороны необходимо по возможности минимизировать, базирясь прежде всего на имеющихся сильных сторонах. Поддержка сильных сторон позволит не только улучшить работу сети парикмахерских в будущем, но и увеличить количество клиентов. Исходя из анализа можно сформировать стратегию деятельности, т.е. долгосрочный план по достижению определенных целей в будущем.

## 3 ОПИСАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

### 3.1 Жизненный цикл разработки мобильного приложения

Жизненный цикл разработки мобильных приложений по большей части совпадает с жизненным циклом для классических или веб-приложений. Его также можно разбить на 5 основных стадий:

1. **Зарождение** — любое приложение начинается с формирования идеи. Как правило, хорошо продуманная идея представляет собой прочный фундамент для дальнейшей разработки приложения.

2. **Проектирование** — на стадии проектирования вы определяете механизм взаимодействия приложения с пользователем, его общий макет, принципы работы и другие моменты, а также проектируете на их основе пользовательский интерфейс (как правило, для этого применяется графический конструктор).

3. **Разработка** — как правило, это наиболее ресурсоемкая стадия создания приложения, которая заключается в его фактическом построении.

4. **Стабилизация** — на определенной стадии разработки обычно начинается оптимизация качества приложения путем его тестирования и исправление выявленных ошибок. Часто приложения выпускаются в виде версий для ограниченного бета-тестирования, на стадии которого более широкая пользовательская аудитория имеет возможность познакомиться с вашим приложением и поделиться своими отзывами о нем.

5. **Развертывание** — многие из этих стадий часто перекрываются. Например, достаточно распространена практика, когда разработка идет уже на стадии окончательной доводки пользовательского интерфейса и может влиять на этот процесс. Кроме того, приложение может возвращаться на стадию стабилизации при добавлении новых функций в очередную версию.

## **3.2 Особенности мобильной операционной системы**

### **Многозадачность**

В iOS реализовано эффективное управление многозадачностью и определено множество правил и требований, которым должно соответствовать ваше приложение в тех случаях, когда на передний план выводится другое приложение. Если эти требования не соблюдаются, ваше приложение попросту может быть закрыто системой iOS.

### **Зависящие от устройств ресурсы**

Даже в устройствах, относящихся к одному форм-фактору, может использоваться совершенно разное оборудование. Например, на одних устройствах есть только основная камера, на других также фронтальная, а на некоторых камера отсутствует.

Из-за этих различий важно всегда проверять наличие функции на устройстве, прежде чем пытаться использовать ее.

### **Ограничения операционной системы**

Чтобы гарантировать работоспособность и безопасность приложений, в iOS применяется ряд обязательных правил и требований. Помимо правил в отношении многозадачности, определено множество методов событий, из которых приложение должно возвращаться не позднее установленного времени, иначе система iOS завершит его работу.

Также необходимо отметить, что приложения выполняются в так называемой «песочнице», изолированной среде, в которой приложение получает ограниченный доступ к ресурсам. Например, у приложения есть собственный каталог, в котором оно может считывать и записывать данные, однако при попытке записать данные в каталог другого приложения оно будет завершено.



## 3.2 Пользовательский интерфейс

Очень важную часть в разработке мобильных приложений занимает проектирование пользовательского интерфейса. Так как разработка интерфейса для приложений, в которых основная задача — предоставление информации, гораздо сложнее, чем разработка интерфейса для аналогичного приложения на ПК. Сложность заключается в том, что экраны мобильных устройств гораздо меньше, чем на ПК. И всю необходимую информацию нужно структурировать и разделять по экранам, продумывать удобство при работе с ней.

При равном количестве информации, пространства для ее размещения намного меньше. Необходимо продумать расположение всех элементов пользовательского интерфейса на экране так, чтобы не произошла информационная перегрузка. Поэтому для каждого экрана нужно четко определить его содержимое, исключив лишние детали.

Разработка интерфейса для приложения «KONTORA Barbershop» была основана на принципе: спрятать все лишнее и показать только нужное, удобный и интуитивно понятный доступ к данным. И каждый экран отображает одну логическую единицу информации. В приложении используется смешанная структура интерфейса. Основным экран приложения представляет собой табличную навигацию (Tab View). Снизу экрана находится панель (Tab bar) с набором пунктов, каждый из которых является кнопкой переключения на соответствующий раздел приложения.

Интерфейс приложения состоит следующих разделов (экранов):

- экран «Мои Записи»: список с информацией о записи (название услуги, дата, цена);
- экран записи на услуги через yClients;
- экран «Блог»: список статей (главная картинка статьи, название, краткое описание);
- экран просмотра статей;

- экран услуг: список с названием услуг и их стоимостью;
- экран товаров: отображение списка товаров (название, изображение, описание, цена);
- экран барберов: список с фотографиями барберов, их именами и должностями.

### Экран «Мои записи»

Экран «Мои записи» представляет собой список предстоящих и прошедших посещений барбершопа. В ячейке, отображающей одну запись, расположено название услуги, барбер, дата посещения и цена услуги. Список делится на текущие и прошедшие записи. Является стартовым пунктом в Tab Bar основного экрана. Открывается при запуске приложения, рисунок 5.

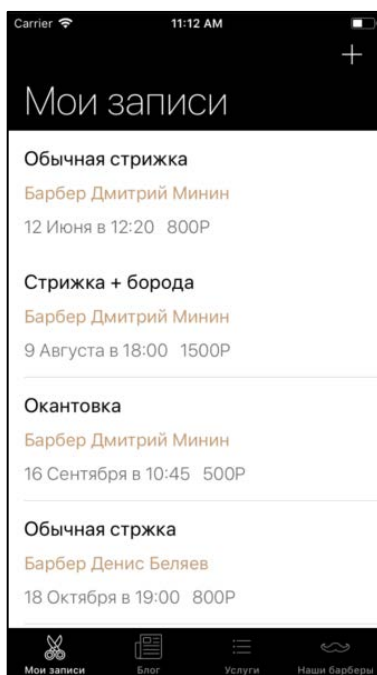


Рисунок 5 — Внешний вид экрана «Мои записи»

### Экран «Записи на услуги через uClients»

Экран сервиса uClients состоит из Web View, на котором открывается сервис по ссылке, передаваемой при создании экрана. Сам интерфейс uClients представляет набор экранов со списками барберов, услуг, выбором даты и времени записи (рисунок 6 и 7).

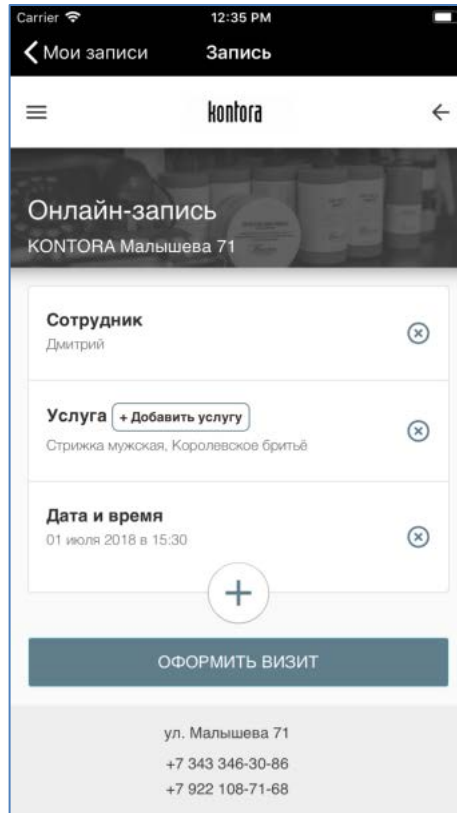


Рисунок 6 — Внешний вид экрана записи на услуги через uClients

Результат создания записи представлен на рисунке 7.

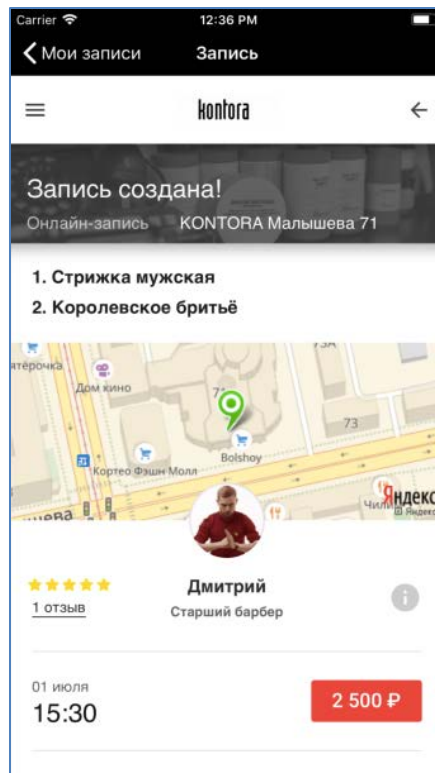


Рисунок 7 — Внешний вид экрана записи на услуги через uClients

## Экран «Блог»

Экран «Блоги» имеет список статей с краткой информацией. У статьи отображаются название, краткое описание и изображение (рисунок 8).



Рисунок 8 — Внешний вид экрана Блог

При нажатии на статью происходит переход на полный вид статьи в приложении (рисунок 9 и 10).

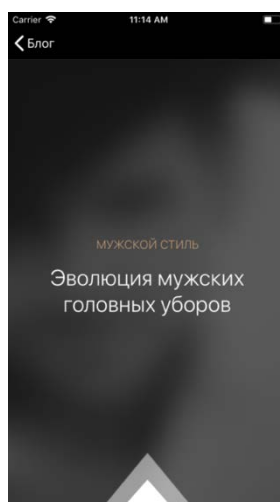


Рисунок 9 — Внешний вид экрана просмотра статей



Рисунок 10 — Внешний вид экрана просмотра статей

На экране самой статьи отображается HTML-верстка статьи, взятая с мобильной версии сайта.

### Экран услуги

Представляет собой комбинацию списка услуг и списка товаров. Переключение происходит с помощью segment control (переключатель), находящегося сверху экрана. Список услуг отображает название и цену услуги, ячейки не кликабельны. Список товаров состоит из ячеек в которых находятся: изображение товара, название и цена (рисунок 11 и 12).

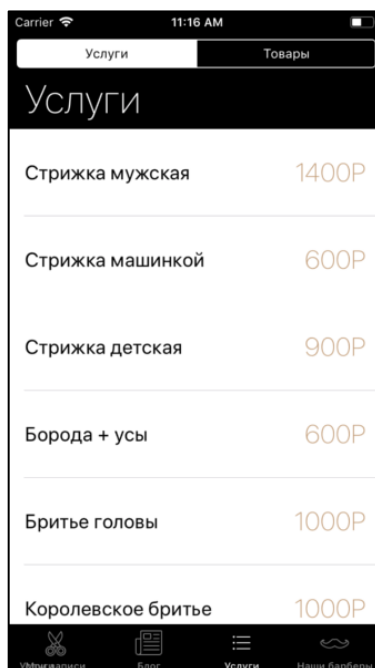


Рисунок 11 — Внешний вид экрана списка услуг

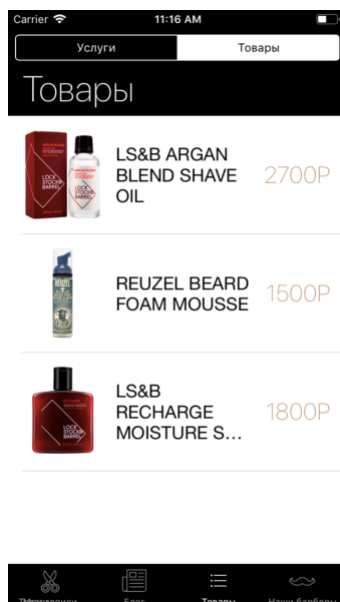


Рисунок 12 — Внешний вид экрана списка товаров

### Экран «Наши барберы»

Экран «Наши барберы» состоит из списка барберов, где в ячейках отображаются фото, имя барбера, его должность. Все барьеры в списке сгруппированы по месту работы (рисунок 13).

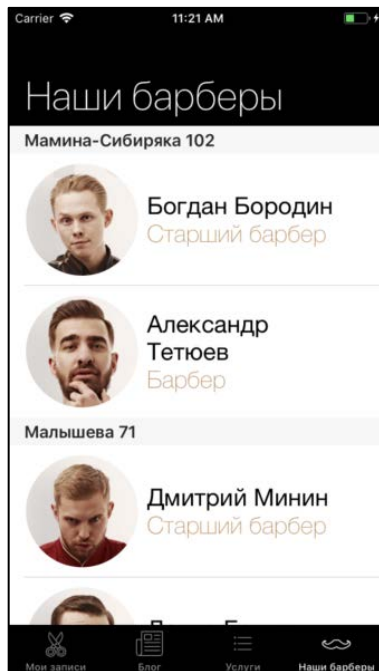


Рисунок 13 — Внешний вид экрана «Наши барберы»

При нажатии на ячейку происходит переход на страницу uClients, где откроется страница записи к этому барберу (рисунок 14).

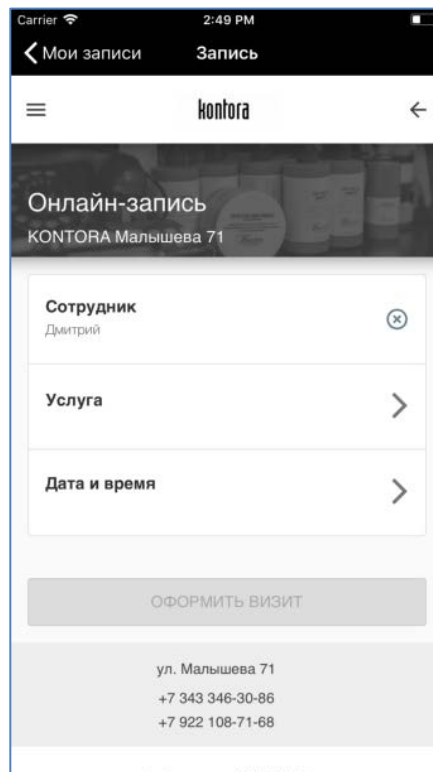


Рисунок 14 — Внешний вид экрана записи к барберу через uClients

### 3.3 Описание этапа тестирования, апробирования, внедрения

Для анализа работы приложения и выявления ошибок была внедрена система Microsoft AppCenter, которая позволяет собирать системные данные со всех, установленных приложений. Регистрируется аккаунт в системе AppCenter, регистрируется приложение, используя идентификационный номер Bundle Id, выдается серийный номер, который используется для инициализации AppCenter в самом приложении. В веб-интерфейсе системы можно просмотреть все вылеты и ошибки iOS приложений, а также всю статистику использования: количество скачиваний, активных клиентов, статистику по версиям iOS. Система AppCenter позволяет доработать продукт на этапе бета-тестирования и релиза приложения.

Для проведения тестирования приложения была использована система, созданная компанией Apple, TestFlight. В ней можно указать список аккаунтов AppleId и указанные пользователи получают приглашение на тестирование. В TestFlight также есть система сбора статистики, как и в AppCenter, но

набор функции уступает аналогу. Поэтому было решено использовать AppCenter и TestFlight в связке.

### 3.4 Технические требования к проекту

Исходя из требований заказчика были сформулированы технические требования к проекту:

- приложение поддерживает работу на версиях iOS от 9.0 и выше;
- приложение правильно отображает верстку на всех разрешениях экрана iPhone;
- приложение требует подключение к интернету;
- режим офлайн не поддерживается;
- приложение активно использует интернет трафик, так как производится загрузка изображений;
- приложение может отправлять данные статистики в AppCenter;
- приложение может работать с ограниченным набором функций, если пользователь не прошел авторизацию;
  - запись в барбершоп и получение списка записей происходит через сервис uClients;
  - при ненадлежащей работе сервиса uClient не будет доступен соответствующий функционал;
  - приложение может хранить токен авторизации для избегания повторной авторизации при выгрузке приложения из оперативной памяти.



## **4 ОПРЕДЕЛЕНИЕ ЭКОНОМИЧЕСКИХ ЗАТРАТ НА РАЗРАБОТКУ ПРОГРАММНОГО ПРОДУКТА**

### **4.1 Описание продукции**

Внедрение мобильного приложения связано с материальными затратами на разработку и функционирование программного продукта. Поэтому важнейшей задачей является анализ экономической эффективности его внедрения. Ее своевременное решение дает возможность сравнивать различные варианты автоматизации и установить оптимальный вариант, оценить его влияние на изменение показателей деятельности организации.

Эффективность внедрения программного обеспечения (ПО) обуславливается действием ряда факторов организационного, информационного и экономического характера.

Организационный эффект проявляется в упрощении записи на стрижку и возможности уведомления пользователя о ближайшей записи, а также в получении всей необходимой информации, структурированной по разделам.

Информационный фактор эффективности выражается в повышении лояльности пользователей.

Экономический фактор проявляется в том, что вся обрабатываемая информация, в конечном счете, направлена на улучшение использования трудовых и временных ресурсов.

Базой для оценки экономической эффективности внедрения ПО может служить количество постоянных клиентов.

Оценить эффективность применения ПО можно с помощью прямых и косвенных показателей. Прямые (или экономические) показатели дают оценку автоматизации в денежном выражении, включают в себя определение затрат на разработку и эксплуатацию информационной системы (т.е. определение полной стоимости владения информационной системой), определение

денежного потока, высвобожденного за счет информационной системы. К косвенным показателям можно отнести:

- повышение оперативности и актуальности информации;
- повышение качества информации, ее точности, детальности;
- снижение количества времени, затрачиваемого на обработку информации о клиенте, абонементе;
- повышение качества обслуживания клиентов;
- повышение качества труда за счет сокращения рутинных операций;
- совершенствование работы аппарата управления.

## 4.2 Оценка затрат на разработку программного продукта

Учет фактических временных затрат представляем в виде таблицы 3.  
Таблица 3 – Фактические временные затраты

Этапы разработки	Фактические затраты (час)
1. Подготовка технического задания с заказчиком	8
2. Проектирование прототипов экранов приложения	24
3. Согласование API сервера для мобильного приложения	8
4. Реализация дизайна мобильного приложения в макетах	65
5. Разработка мобильного приложения на одну платформу (iOS)	192
6. Подготовка документации по задаче	7
6.1. Подготовка материала в рукописи	5
6.2. Редактирование, печать и оформление документации	2
<b>ИТОГО</b>	<b>304</b>

Коэффициент, учитывающий сложность разработки и затраты на корректировку исходного модуля определяем по формуле:

$$K_{сл} = c \cdot (1 + p), \quad (1)$$

где  $c$  — коэффициент сложности программы;

$p$  — коэффициент коррекции программы в процессе разработки.

Таблица «Коэффициенты по категориям» (таблица 4) будет использована для расчета.

Таблица 4 — Коэффициенты по категориям

Наименование коэффициента	Категория			
	1	2	3	4
Коэффициент сложности программы	1,25	1,5	1,6	2,0
Коэффициент коррекции программы	0,05	0,1	0,5	1,0

$$K_{cl} = 1,5 \cdot (1 + 0,1) = 1,65$$

Затраты времени на разработку с учетом сложности программы выражаются произведением фактических временных затрат на разработку и коэффициента сложности.

$$t_{разр_{cl}} = t_{разр_{ф}} \cdot K_{cl} \quad (2)$$

$$t_{разр_{cl}} = 304 \cdot 1,65 = 502 \text{ час.}$$

### Расчет средней часовой оплаты программиста

Для определения средней часовой оплаты программиста определяем его годовой фонд заработной платы с учетом отчислений в социальные фонды (в Пенсионный фонд, Фонд обязательного медицинского страхования и Фонд социального страхования).

iOS-разработчик очень высокооплачиваемая профессия. Месячный оклад программиста мобильных приложений по нативной разработки (Swift) составляет примерно 75000 рублей. Учитывается процент премий (20%) и районный коэффициент (15%). На суммарный фонд заработной платы начисляется процент взноса в социальные фонды. В 2017 году он составляет 30%. Из него в ПФР — 22%, ФСС РФ — 2,9% и в ФФОМС — 5,1%.

Определяем месячный оклад программиста с учетом квалификации и всех надбавок.

$$ЗП_M = ЗП_{осн} \cdot (1 + K_{доп}) \cdot (1 + K_{ур}) \cdot (1 + K_{сн}), \quad (3)$$

где  $ЗП_{осн}$  — месячная оплата труда установленной квалификации;

$K_{\text{доп}}$  — коэффициент надбавок и премий;

$K_{\text{ур}}$  — уральский коэффициент;

$K_{\text{есч}}$  — коэффициент, учитывающий норму взноса в социальные фонды.

$$ЗП_M = 35000 \cdot (1+0,2) \cdot (1+0,15) \cdot (1+0,30) = 62790 \text{ руб.}$$

Годовой фонд заработной платы с учетом отчислений.

$$\Phi ЗР_G = ЗП_M \cdot 12, \quad (4)$$

где 12 — количество месяцев в году.

$$\Phi ЗР_G = 62790 \cdot 12 = 753480 \text{ руб.}$$

Число рабочих часов в году определяется согласно производственному календарю на 2017 год.

$$n_p = (N - N_{\text{пв}}) \cdot N_{\text{см}} - N_{\text{пп}} \cdot 1, \quad (5)$$

где  $N$  — общее число дней в году;

$N_{\text{пв}}$  — число праздничных и выходных дней в году;

$N_{\text{пп}}$  — число предпраздничных дней в году;

$N_{\text{см}}$  — продолжительность смены;

1 — величина сокращений предпраздничных рабочих дней.

Согласно производственного календаря на 2017 год продолжительность рабочего времени 1775 часов при 36-часовой неделе.

Средняя часовая оплата программиста определяется соотношением

$$C_{\text{разр}} = \frac{\Phi ЗР_G}{n_p}, \quad (6)$$

где  $\Phi ЗР_G$  — годовой фонд заработной платы с учетом отчислений;

$n_p$  — число рабочих часов в году.

$$C_{\text{разр}} = \frac{753480}{1775} = 425 \text{ руб./час.}$$

Расходы по оплате труда разработчика программы определяются по следующей формуле:

$$Z_{разр} = t_{разр.сл} \cdot C_{разр}, \quad (7)$$

где  $t_{РАЗР.СЛ}$  — трудоемкость создания программы, с учетом сложности программы, выраженная в часах;

$C_{РАЗР}$  — средняя часовая оплата труда инженера–программиста.

$$Z_{разр} = 502 \cdot 425 = 213350 \text{ руб.}$$

### 4.3 Расчет годового фонда времени работы на персональном компьютере

Определив действительный годовой фонд времени работы персонального компьютера (ПК) в часах, получим возможность оценить себестоимость часа машинного времени. Время профилактики: ежедневно — 0.5 часа, ежемесячно — 2 часа, ежегодно — 16 часов.

$$n_{ПК} = n_p - N_{РЕМ}, \quad (8)$$

где  $N$  — общее число дней в году;

$N_{ПВ}$  — число праздничных и выходных дней в году;

$N_{ПП}$  — число предпраздничных дней в году;

$N_{СМ}$  — продолжительность смены;

1 — величина сокращений предпраздничных рабочих дней;

$N_{РЕМ}$  — время на проведение профилактических мероприятий.

$$n_{ПК} = 1775 - 153,5 = 1621,5 \text{ часов.}$$

$$N_{РЕМ} = (N - N_{П} - N_{В}) \cdot K_{Д} + K_{М} \cdot 12 + K_{Г}, \quad (9)$$

где  $K_{Д}$  — коэффициент ежедневных профилактик (0,5);

$K_{М}$  — коэффициент ежемесячных профилактик (2);

12 — количество месяцев в году;

$K_{Г}$  — коэффициент ежегодных профилактик (6).

$$N_{РЕМ} = (365 - 118) \cdot 0,5 + 2 \cdot 12 + 6 = 153,5 \text{ часов.}$$

## Годовые отчисления на амортизацию персонального компьютера

Балансовая стоимость ПК

$$Ц_{ПК} = Ц_P \cdot (1 + K_{УН}), \quad (10)$$

где  $Ц_P$  — рыночная стоимость ПК (определяется по прайсу);

$K_{УН}$  — коэффициент, учитывающий затраты на установку и наладку.

$$Ц_{ПК} = 80000 (1+0,1) = 88000 \text{ руб.}$$

Сумма годовых амортизационных отчислений определяется по формуле:

$$З_{ГАМ} = Ц_{ПК} \cdot H_A, \quad (11)$$

где  $Ц_{ПК}$  — балансовая стоимость ПК;

$H_A$  — норма амортизационных отчислений за год.

$$З_{ГАМ} = 88000 \cdot 0,2 = 17600 \text{ руб.}$$

$$H_a = \frac{1}{T \frac{ПК}{ЭКС}} \cdot 100, \quad (12)$$

где  $T \frac{ПК}{ЭКС}$  — полезный срок действия ПК.

$$H_a = \frac{1}{5} \cdot 100 = 20\% .$$

## Затраты на электроэнергию

$$З_{ГЭЛ} = P_{ЧПК} \cdot T_{ГПК} \cdot Ц_{ЭЛ} \cdot K_{ИНТ}, \quad (13)$$

где  $P_{ЧПК}$  — установочная мощность ПК;

$T_{ГПК}$  — годовой фонд полезного времени работы машины ( $n_{ПК}$ );

$Ц_{ЭЛ}$  — стоимость 1 кВт/ч. электроэнергии ( $Ц_{ЭЛ}=3,7$  руб./кВт/ч);

$K_{ИНТ}$  — коэффициент интенсивного использования ПК (0,9).

$$З_{ГЭЛ} = 0,35 \cdot 1621,5 \cdot 3,7 \cdot 0,9 = 1890 \text{ руб.}$$

Текущие затраты на эксплуатацию ПК рассчитываются по формуле:

$$Z_{ГПК} = Z_{ГАМ} + Z_{ГЭЛ}, \quad (14)$$

где  $Z_{ГАМ}$  — годовые отчисления на амортизацию;

$Z_{ГЭЛ}$  — годовые затраты на электроэнергию для компьютера.

$$Z_{ГПК} = 17600 + 1890 = 19490 \text{ руб.}$$

Себестоимость часа работы на компьютере:

$$C_{ПК} = \frac{Z_{ГПК}}{n_{РПК}}, \quad (15)$$

где  $Z_{ГПК}$  — годовые затраты на ПК;

$n_{РПК}$  — годовой фонд полезного времени работы машины.

$$C_{ПК} = \frac{19490}{1621,5} = 12 \text{ руб./час}$$

### **Трудоемкость использования компьютера**

В ходе разработки программного продукта, машина используется на этапах программирования по спроектированной архитектуре приложения и прототипов экранов, подготовки документации по задаче.

Совокупные затраты машинного времени составляют:

$$t_{МАШ} = (t_{ПФ} + t_{ОПФ} + t_{ДФ}) \cdot K_{СЛ}, \quad (16)$$

$$t_{МАШ} = (65 + 192 + 7) \cdot 1,65 = 436 \text{ час.}$$

### **Затраты на оплату машинного времени**

Затраты на оплату машинного времени рассчитываются по формуле:

$$Z_{МАШ} = t_{МАШ} \cdot C_{ПК}, \quad (17)$$

$$Z_{МАШ} = 436 \cdot 12 = 5232 \text{ руб.}$$

### **Общие затраты на создание программы**

Общие затраты на создание программы определяем как сумма затрат на разработку программы и затрат на оплату машинного времени.

$$Z_{ОБЩ} = Z_{РАЗР} + Z_{МАШ} \quad (18)$$

$$Z_{\text{ОБЩ}} = 213350 + 5232 = 218582 \text{ руб.}$$

### Общие затраты на расходные материалы

При формировании общих затрат на создание программного продукта, кроме затрат на оплату труда разработчика и на оплату машинного времени, необходимо учитывать затраты, связанные с использованием расходных материалов и комплектующих в процессе проектирования, разработки и внедрения. Статьи затрат на расходные материалы с указанием их стоимости показаны в таблице 5.

Таблица 5 — Расходные материалы

Статьи затрат	Стоимость за единицу	Количество	Общая стоимость
1. Пользование ресурсами Интернет	0,87 руб./час	50 ч.	43,5 руб.
2. Бумага	0,3 руб./л.	30 л.	9 руб.
Итого: (З <sub>РМ</sub> )			52,5 руб.

Общие затраты на разработку программного продукта приведены в таблице 6.

Таблица 6 — Общие затраты на разработку программного продукта

Статьи затрат	Условное обозначение	Числовое значение
1. Общие затраты на заработную плату	З <sub>ОБЩ</sub>	218582 руб.
2. Расходные материалы	З <sub>РМ</sub>	52,50 руб.
Итого: (С <sub>ОБЩ</sub> )		218634,50 руб.

### 4.4 Предполагаемая цена программного продукта с учетом нормы прибыли

Вычислим предполагаемую цену продукта с учетом нормы прибыли:

$$Ц = C_{\text{ОБЩ}} \cdot (1+N), \quad (19)$$

где  $C_{\text{ОБЩ}}$  — общие затраты на разработку программного продукта;

$N$  — норма ожидаемой прибыли.



$$Ц = 218634,50 \cdot (1+0,2) = 262361,40 \text{ руб.}$$

Предполагаемая цена данного продукта с учетом нормы прибыли составила 262361,40 руб.

## ЗАКЛЮЧЕНИЕ

Поставленные задачи были выполнены, а именно:

- проведено исследование предметной области;
- проанализированы аналогичные мобильные приложения;
- разработана модель программного продукта для автоматизации процесса записи клиентов на услуги в компанию ООО «Барбершоп»;
- реализовано мобильное приложение;
- проведено тестирование приложения на реальных пользователях;
- получен опыт разработки нативных приложений на платформе iOS.

Результатом работы является рабочее приложение, которое может обеспечить нужной информацией и возможностями клиентов «KONTORA Barbershop», которые активно пользуются услугами компании. Приложение было протестировано реальными людьми и многим понравилось это приложение.

В будущем запланировано продолжение разработки и поддержание его работы. В ближайшее время планируется внедрение новых функций и возможностей:

- поддержка нескольких регионов;
- система общения с барберами;
- система рейтинга среди сотрудников ООО «Барбершоп»;
- система отзывов из приложения;
- внедрение социальной составляющей (взаимодействие с социальными сетями, конкурсы и т.п.);
- система push-уведомлений;
- реализация на мобильную платформу Android.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Барбершопы в России: мы двигаемся по западному пути, но с большим отставанием [Электронный ресурс]. — Режим доступа: <https://marketing.rbc.ru/articles/10216/> (дата обращения: 20.12.2017).

2. Введение в жизненный цикл разработки мобильного программного обеспечения [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/xamarin/cross-platform/get-started/introduction-to-mobile-sdlc> (дата обращения: 22.12.2017).

3. Как сделать приложение для iPhone самому [Электронный ресурс]. — Режим доступа: <https://www.iphones.ru/iNotes/543737> (дата обращения: 25.12.2017).

4. Как создать успешное приложение для iOS [Электронный ресурс]. — Режим доступа: <https://vc.ru/1188-ios-apps> (дата обращения: 26.12.2017).

5. Кроссплатформенная и нативная разработка [Электронный ресурс]. — Режим доступа: [https://appcraft.pro/blog/nativnaja\\_razrabotka/](https://appcraft.pro/blog/nativnaja_razrabotka/) (дата обращения: 05.01.2018).

6. Нативная или кроссплатформенная разработка? [Электронный ресурс]. — Режим доступа: <http://wnfx.ru/nativnaya-ili-krossplatformennaya-razrabotka-chto-luchshe/> (дата обращения: 10.01.2018).

7. Обзоры приложений для iOS [Электронный ресурс]. — Режим доступа: <https://appleinsider.ru/obzory-prilozhenij> (дата обращения: 14.01.2018).

8. Обучение созданию приложений на Swift [Электронный ресурс]. — Режим доступа: <http://swiftbook.ru> (дата обращения: 10.01.2018).

9. Основатель барбершопа KONTORA о своей компании [Электронный ресурс]. — Режим доступа: <http://barberkontora.ru/onas/history/> (дата обращения: 20.01.2018).

10. Руководство по языку программирования Swift [Электронный ресурс]. — Режим доступа: <https://metanit.com/swift/tutorial/> (дата обращения: 17.03.2018).

11. Рынок мобильных приложений [Электронный ресурс]. — Режим доступа: <https://magora-systems.ru/rynok-mobilnyh-prilozheniy-2017/> (дата обращения: 21.03.2018).

12. Самостоятельная разработка приложения для IOS в среде Xcode [Электронный ресурс]. — Режим доступа: <http://wnfx.ru/kak-sozdat-prilozhenie-dlya-iphone/> (дата обращения: 04.03.2018).

13. Семь веских причин, зачем мобильное приложение просто необходимо для бизнеса [Электронный ресурс]. — Режим доступа: <http://app-global.ru/blog/7-veskih-prichin-zachem-mobilnoe-prilozhenie-prosto-neobhodimo-dlya-biznesa/> (дата обращения: 05.04.2018).

14. Создание iOS приложения. От идеи до результата [Электронный ресурс]. — Режим доступа: <https://habr.com/post/250171/> (дата обращения: 06.04.2018).

15. Создание iOS-приложения в Sketch и Xcode [Электронный ресурс]. — Режим доступа: <http://sketchapp.me/sozдание-ios-prilozheniya-v-sketch-i-xcode-publikaciya-ego-v-app-store/> (дата обращения: 07.04.2018).

16. Что такое барбершоп? История появления барбершопов [Электронный ресурс]. — Режим доступа: <https://sborodoy.com/blog/vidy-borod/cto-takoe-barbersop-istoria-poavlenia-barbersopov> (дата обращения: 03.04.2019).

17. Язык программирования Swift. Русская версия [Электронный ресурс]. — Режим доступа: <https://habr.com/post/225841/> (дата обращения: 09.04.2018).

18. App Development with Swift [Электронный ресурс]. — Режим доступа: <https://is.gd/n5gsTz> (дата обращения: 13.05.2018).

19. Apple Developer [Электронный ресурс]. — Режим доступа: <https://developer.apple.com/> (дата обращения: 06.05.2018).

20. Apple iOS [Электронный ресурс]. — Режим доступа: <http://www.combonews.ru/encyclopedia/2322-apple-ios.html> (дата обращения: 15.04.2018).

21. Big Bro Barbershop [Электронный ресурс]. — Режим доступа: <https://itunes.apple.com/ru/app/big-bro-barbershop/id1189963629?l=en&mt=8> (дата обращения: 17.05.2018).

22. Chop-Chop [Электронный ресурс]. — Режим доступа: <https://itunes.apple.com/ru/app/chop-chop/id1002429142?mt=8> (дата обращения: 17.04.2018).

23. iOS программы [Электронный ресурс]. — Режим доступа: <https://4pda.ru/forum/index.php?showforum=140> (дата обращения: 06.05.2018).

24. iOS-разработка для начинающих: учимся создавать приложения для iOS с нуля [Электронный ресурс]. — Режим доступа: <https://haker.ru/2016/09/28/ios-from-scratch/> (дата обращения: 07.05.2018).

25. iOS-разработка: способы быстрого старта [Электронный ресурс]. — Режим доступа: <https://habr.com/companу/jugru/blog/337394/> (дата обращения: 21.04.2018).

26. OldBoy Барбершоп [Электронный ресурс]. — Режим доступа: <https://is.gd/26vppR> (дата обращения: 22.05.2018).

27. REST API Handbook [Электронный ресурс]. — Режим доступа: <https://developer.wordpress.org/rest-api/> (дата обращения: 20.04.2018).

28. Sketch the digital design toolkit [Электронный ресурс]. — Режим доступа: <https://www.sketchapp.com/> (дата обращения: 13.05.2018).

29. Swift. Язык программирования с открытым кодом [Электронный ресурс]. — Режим доступа: <https://www.apple.com/ru/swift/> (дата обращения: 17.04.2018).

30. Xcode 10 [Электронный ресурс]. — Режим доступа: <https://developer.apple.com/xcode/> (дата обращения: 17.05.2018).

31. YCLIENTS [Электронный ресурс]. — Режим  
доступа: <https://www.yclients.com/> (дата обращения: 25.05.2018).

# ПРИЛОЖЕНИЕ

**Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования**

**«Российский государственный профессионально-педагогический университет»**

Институт инженерно-педагогического образования  
Кафедра информационных систем и технологий  
направление 09.03.03 Прикладная информатика  
профиль «Прикладная информатика в экономике»

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Н. С. Толстова

«25» декабря 2017 г.

## **ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы бакалавра**

студентки 4 курса, группы ИЭ-401п Белоусов Артем Олегович

1. Тема «Мобильное приложение для для сети парикмахерских “Барбершоп”»  
утверждена распоряжением по институту от 25.12.2017 г. № 83.

2. Руководитель Сулова И.А, канд.пед.наук, доцент, доцент кафедры ИС

3. Место преддипломной практики ООО «Амба-Групп»

4. Исходные данные к ВКР

- Apple iOS [Электронный ресурс]. – Режим доступа: <http://www.combonews.ru/encyclopedia/2322-apple-ios.html>.

5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)

- провести исследование предметной области;
- проанализировать аналогичные мобильные приложения;
- разработать модель программного продукта;
- реализовать мобильное приложение.

6. Перечень демонстрационных материалов

- презентация, выполненная средствами PowerPoint,
- мобильное приложение для для сети парикмахерских «Барбершоп»

### 7. Календарный план выполнения выпускной квалификационной работы

№ п/п	Наименование этапа ВКР	Срок выполнения этапа	Процент выполнения ВКР	Отметка руководителя о выполнении
1	Сбор информации по выпускной работе и сдача зачета по преддипломной практике	23.04.2018	10	
2	Выполнение работ по разрабатываемым вопросам их изложение в выпускной работе:		60	
	Анализ предметной области	04.05.2018	10	
	Анализ аналогичных программных продуктов	06.06.2018	5	
	Разработка модели программного продукта	09.05.2018	5	
	Реализация мобильного приложения	11.05.2018	25	
	Определение экономической эффективности	30.05.2018	15	
3	Оформление текстовой части ВКР	15.05.2018	10	
4	Выполнение демонстрационных материалов к ВКР	01.06.2018	10	
5	Нормоконтроль	08.06.2018	5	
6	Подготовка доклада к защите в ГЭК	13.06.2018	5	

### 8. Консультанты по разделам выпускной квалификационной работы

Наименование раздела	Консультант	Задание выдал		Задание принял	
		подпись	дата	подпись	дата
Экономический	Зырянова Н.И.				

Руководитель \_\_\_\_\_ 25.12.17 Задание получил \_\_\_\_\_ 25.12.17  
подпись дата подпись студента дата

9. Выпускная квалификационная работа и все материалы проанализированы. Считаю возможным допустить Белоусова Артема Олеговича к защите выпускной квалификационной работы в государственной экзаменационной комиссии.

Руководитель \_\_\_\_\_  
подпись дата

10. Допустить Белоусова Артема Олеговича к защите выпускной квалификационной работы в государственной экзаменационной комиссии (протокол заседания кафедры от \_\_.\_\_.2018 №\_\_)

Заведующий кафедрой \_\_\_\_\_  
подпись дата