

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Иркутский государственный университет»  
(ФГБОУ ВО «ИГУ»)  
Институт математики и информационных технологий  
Кафедра алгебраических и информационных систем

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**  
по направлению «09.03.03 Прикладная информатика»  
профиль подготовки «Информационная сфера»

**ЦИФРОВАЯ ТРАНСФОРМАЦИЯ СЕТИ  
ПАРИКМАХЕРСКИХ И БАРБЕРШОПОВ**

Студент 4 курса очного отделения  
Группа 02461–ДБ  
Пономарева  
Екатерина Вячеславовна

Руководитель:  
к.ф.-м.н., доцент  
\_\_\_\_\_ Рябец Л.В.

Допущена к защите  
Зав. кафедрой, д.ф.-м.н.  
\_\_\_\_\_ Пантелеев В.И.

Иркутск 2021

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>3</b>
<b>Глава 1. Задача управления потоком клиентов и их идентификации по лицу</b>	<b>5</b>
1.1. Анализ предметной области . . . . .	5
1.2. Обзор существующих программных продуктов, соответствующих предметной области . . . . .	6
1.3. Постановка задачи . . . . .	11
1.4. Среда реализации модулей . . . . .	13
<b>Глава 2. Разработка модулей информационной среды парикмахерских</b>	<b>16</b>
2.1. Реализация модуля для идентификации клиентов по лицу . . . . .	16
2.2. Реализация модуля для управления потоком клиентов . . . . .	24
<b>ЗАКЛЮЧЕНИЕ</b>	<b>32</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>33</b>
<b>ПРИЛОЖЕНИЕ 1. Функции для запроса идентификации клиента из мобильного приложения</b>	<b>36</b>
<b>ПРИЛОЖЕНИЕ 2. Поиск истории клиента по его фото</b>	<b>38</b>
<b>ПРИЛОЖЕНИЕ 3. Алгоритм выбора времени в очереди у мастера</b>	<b>42</b>
<b>ПРИЛОЖЕНИЕ 4. API идентификации клиента</b>	<b>45</b>
<b>ПРИЛОЖЕНИЕ 5. API электронной очереди</b>	<b>50</b>

# ВВЕДЕНИЕ

Современный бизнес становится неразрывно связан с технологиями. Всё большую роль начинает играть не только автоматизация процессов компании, но и их цифровизация. Цифровизация открывает большие возможности в сфере информационных технологий для традиционных организаций, которые не являются ИТ-компаниями. Однако движение в этом направлении требует от них перестройки внутренних и внешних бизнес-процессов, развитие новых практик управления и привлечения специалистов с новыми компетенциями [3].

Государство также осознает необходимость развития перспективных высокотехнологичных направлений. В рамках федерального проекта «Цифровые технологии» [7] были разработаны дорожные карты по развитию сквозных цифровых технологий — технологий не связанных с каким-то отдельным продуктом или сферой деятельности, а применимых во многих индустриях, отраслях и секторах экономики [11].

В связи с развитием современных направлений и технологий появляются новые программные продукты, которые предназначены для решения различных задач бизнеса.

Среди этих задач можно выделить одну из важнейших — повышение уровня лояльности клиентов. Компании заинтересованы в том, чтобы люди, с которыми они работают, чувствовали себя максимально комфортно и были довольны качеством предоставляемых услуг. Одной из таких компаний является международная сеть парикмахерских и барбершопов «Брадобрей» [2]. Данная компания уже использует различные программные продукты для автоматизации своих внутренних бизнес-процессов. Однако теперь они хотели бы расширить успешный опыт и воспользоваться технологиями для улучшения качества и скорости обслуживания клиентов путем создания онлайн-очереди и базы стрижек клиентов с поиском по лицам. В связи с этим **практически значимой** является

задача доработки программного обеспечения данной компании.

**Целью** выпускной квалификационной работы является доработка существующего мобильного приложения и конфигурации на платформе «1С: Предприятие» и создание web-сайта для электронной очереди.

Для достижения поставленной цели необходимо было решить следующие основные **задачи**:

- 1) провести обзор и сравнительный анализ существующих аналогов для идентификации клиентов по фото и электронной очереди;
- 2) рассмотреть способы организации электронной очереди;
- 3) реализовать модуль онлайн-очереди;
- 4) реализовать модуль для хранения и поиска фотографий стрижек клиентов;

**Структура выпускной квалификационной работы.** Данная работа представлена на 54 страницах, состоит из введения, 2 глав, заключения, списка использованных источников и 5 приложений. В работе присутствуют 2 листинга, 17 рисунков и 2 таблицы.

# Глава 1. Задача управления потоком клиентов и их идентификации по лицу

## 1.1. Анализ предметной области

В выпускной квалификационной работе рассматривается два модуля: для управления электронной очередью и для идентификации клиента по лицу. В условиях ограничений из-за пандемии COVID-19 компании были вынуждены уменьшить количество посетителей, одновременно находящихся в помещении. Вариантами решения этой проблемы является осуществление записи на конкретное время или организация электронной очереди. У обоих способов есть достоинства и недостатки. В табл. 1 представлено сравнение этих способов и обычной «живой» очереди, которую использовали многие компании до введения ограничений.

Электронная очередь — это комплекс аппаратных и программных средств, которые позволяют управлять очередью в автоматическом режиме [12].

Онлайн-очередь — это способ организации электронной очереди, при котором получение порядкового номера осуществляется через web-сайт или мобильное приложение.

Таблица 1

Сравнение способов управления потоком клиентов

Параметр	Запись на время на сайте	Обычная очередь	Электронная очередь с терминалом	Онлайн-очередь
Выбор конкретного мастера	Удобно	Не удобно	Удобно	Удобно
Отслеживание времени	Удобно	Не удобно	Не удобно	Удобно
Отсутствие «окон» между клиентами	—	+	+	+

Продолжение таб. 1

Возможность записать/встать в очередь удаленно	+	–	–	+
Отсутствие контактов с людьми	+	–	–	+

Основным преимуществом идеи очереди над записью для компаний, осуществляющих услуги, является отсутствие «окон» между клиентами в случае, если записанный человек не пришел. Непрерывный поток клиентов исключает возможность «простоя» сотрудников, во время которого компания не получает прибыли.

Рассмотрим предметную область, касающуюся второго модуля выпускной квалификационной работы: распознавания лиц.

Распознавание лиц может осуществляться с целью верификации или с целью идентификации. Верификация (или аутентификация) — это подтверждение идентичности переданного на распознавание лица одноименному лицу в базе данных. Если задачей распознавания является выбор из базы данных наиболее похожего на переданное изображение лица, то эту задачу можно определить как идентификацию [5].

Для CRM-систем задача идентификации является более актуальной, так как среди всех клиентов компании необходимо найти конкретного, а также информацию о прошлом взаимодействии с ним.

## **1.2. Обзор существующих программных продуктов, соответствующих предметной области**

В ходе исследования был проведен поиск и анализ уже существующих программных продуктов по автоматизации очереди и идентификации клиентов по лицу.

## Программные продукты для управления потоком клиентов

Были проанализированы следующие программные продукты для управления потоком клиентов:

- «YCLIENTS»;
- Электронная очередь от компании «BR System»;

«YCLIENTS» – это облачный сервис для автоматизации процессов компании сферы услуг [16]. Возможности данной системы также включают в себя онлайн-запись. Настройки позволяют управлять категориями услуг, добавлять сотрудников организации, указывать рабочие часы и возможность отмены или переноса записи. На рис. 1 продемонстрирована работа программы.

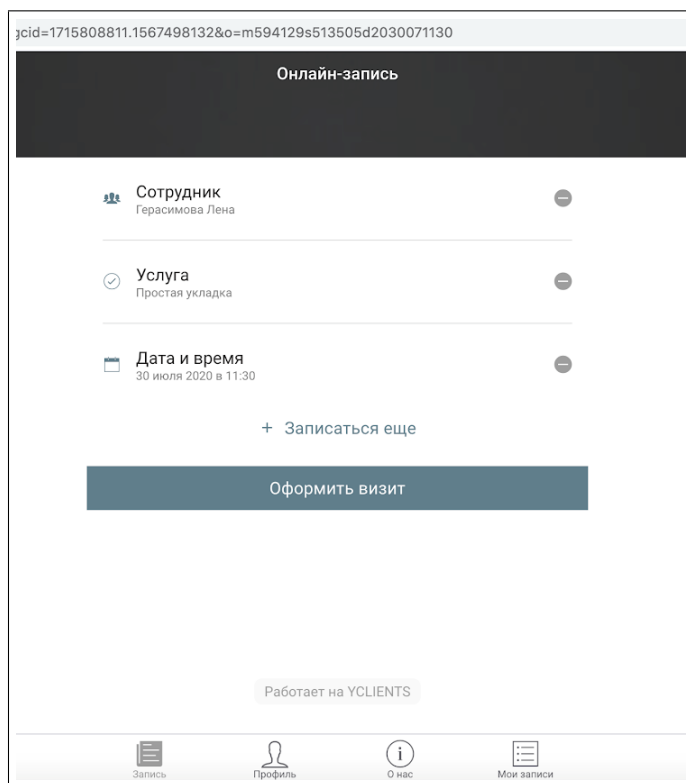


Рисунок 1. Форма записи для клиента

Электронная очередь от компании «BR System» [13] — это облачный сервис, который позволяет организовать очередь без применения специального оборудования. Создатели сервиса предлагают распечатать уникальный

QR-код для компании, отсканировав который клиент попадает на сайт и получает свой номер в очереди. Пример электронного талона представлен на рис. 2.

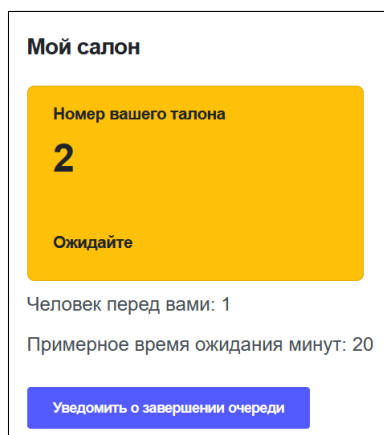


Рисунок 2. Электронный талон

Когда подходит очередь клиента, то ему на устройство приходит уведомление. Также есть возможность вывести на табло в зоне ожидания текущее состояние очереди.

В настройках сервиса можно указать среднее время на оказание услуги, и при получении электронного талона клиент узнает примерное время ожидания.

Основное отличие сервиса компании «BR System» от «YCLIENTS» состоит в отсутствии записи на конкретное время, что позволяет избежать «окон» в работе сотрудников. Более подробное сравнение представлено в табл. 2.

Таблица 2

Сравнение «BR System» и «YCLIENTS»

Параметр	«BR System»	«YCLIENTS»
Выбор конкретного мастера	Через создание нескольких личных кабинетов	+
Отсутствие «окон» между клиентами	+	–
Возможность записаться/встать в очередь удаленно	+	+
Интеграция с конфигурацией «1С: Салон красоты»	–	Через API
Бесплатный сервис	+	–



Рассмотренные программные продукты больше подходят для использования организациями, в которых отсутствует собственная учетная система, так как интеграция существующих решений с программными продуктами компании не всегда возможна.

В сети «Брадобрей» используется автоматизация на платформе «1С: Предприятие», поэтому модуль электронной очереди должен быть частью существующего программного комплекса организации. Это послужило причиной необходимости разработки собственной системы для онлайн-очереди.

### **Программные продукты для идентификации клиентов по лицу**

Рассмотрим следующие программные продукты для идентификации клиентов по лицу:

- «Face Recognition» для «Dynamics 365 CRM»;
- «Визит-трекер» для «Битрикс 24».

«**Face Recognition**» для «Dynamics 365 CRM» позволяет находить лица в непрерывном видеопотоке с веб-камеры и соотносить их с профилями клиентов в «Microsoft Dynamics 365 CRM» [14]. В карточку клиента предварительно необходимо добавить фотографию, а затем служба Azure Cognitive Services распознает лицо на фотографии и через Face API создает его «слепок» в хранилище Azure Person ID. Пример интерфейса системы представлен на рис. 3

«**Визит-трекер**» — это один из подключаемых к «Битрикс 24» модулей, который позволяет идентифицировать клиентов по фотографии лица на веб-камеру, вести запись разговора, искать профиль в социальной сети «ВКонтакте» по фотографии, связывать информацию из «Визит-трекера» с CRM [1]. Пример интерфейса системы представлен на рис. 4.

Рассмотренные программный продукты являются актуальными и хорошо применимыми в своей области. Однако не существует системы, которая позволяла бы производить идентификацию клиента не при помощи веб-камеры, а с

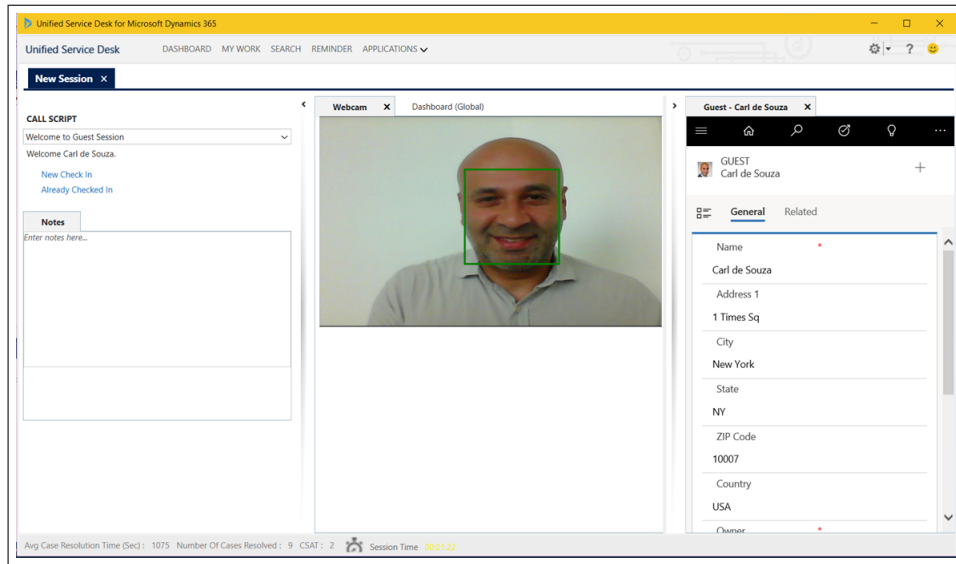


Рисунок 3. Идентификация клиента в «Dynamics 365 CRM»

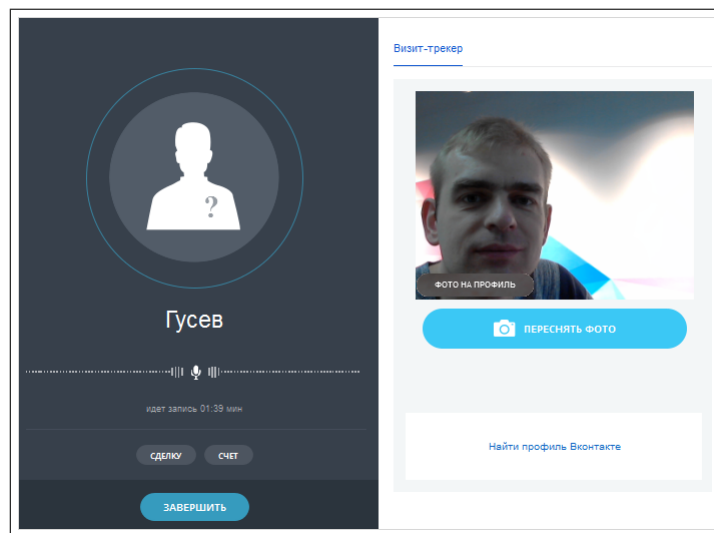


Рисунок 4. Фиксация посещения клиента

помощью встроенной камеры мобильного устройства и при этом имела интеграцию с конфигурацией «1С: Салон красоты», поэтому появляется необходимость в создании такого программного продукта.

### 1.3. Постановка задачи

#### Модуль для идентификации клиентов по лицу

Компания «Брадобрей» использует мобильное приложение, которое позволяет работникам вести учет своего времени, осуществлять подбор номенклатуры из прайс-листа и пробивать чеки. Было предложено расширение возможностей приложения для повышения уровня обслуживания клиентов. Стояла проблема, связанная с тем, что клиенты предпочитают каждый раз обращаться к одному и тому же мастеру. После проведенного опроса была выявлена основная причина — мастер помнит, какую стрижку он делал в прошлый раз, и клиенту не нужно объяснять, как его подстричь. Данная проблема приводит к тому, что к новым мастерам, имеющими такой же уровень подготовки, как и давно работающие парикмахеры, приходит меньше клиентов, а это тормозит развитие и расширение бизнеса.

Было предложено после стрижки фотографировать клиента с различных ракурсов, сохранять получившиеся изображения, и затем при следующем визите клиента показывать мастеру фотографии, для того чтобы отсутствовала необходимость словесно описывать желаемую стрижку. Разработка этого механизма является основной задачей в рамках реализации данного модуля.

После знакомства с конфигурациями и обсуждений с заказчиком была составлена последовательность действий, которые должны происходить для работы нового модуля приложения (см. рис. 5):

Далее возникла необходимость решить каким именно способом будет происходить идентификация клиента. После анализа данных о клиентах, которые содержатся в основной базе было выявлено, что клиенты могут быть записаны по ФИО, только по фамилии, только по имени, по номеру телефона или описательным комментарием (например, «ребенок 6 лет»), а также встречались комбинации перечисленных способов и дубли записей. Поиск вручную по та-

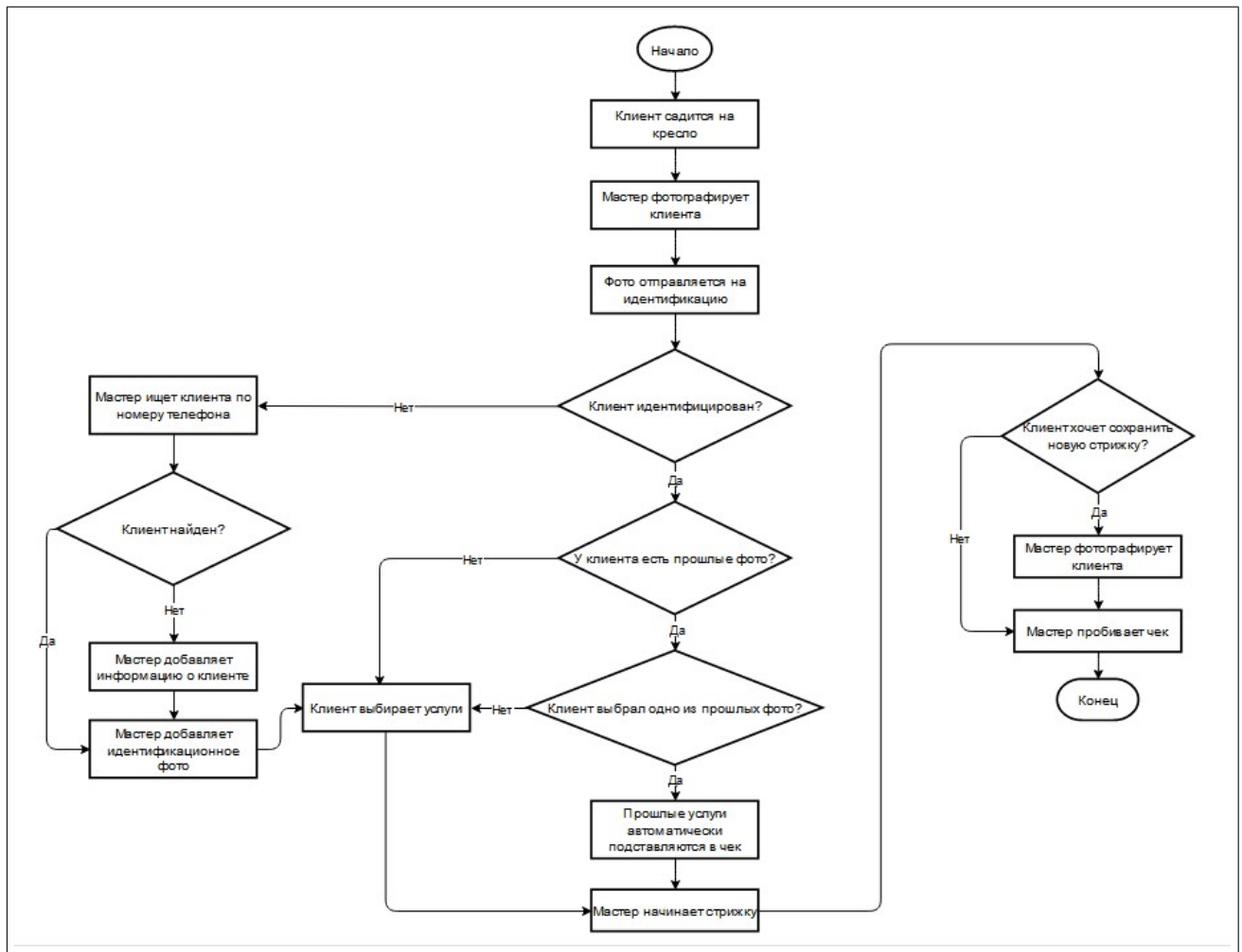


Рисунок 5. Блок-схема процесса идентификации и стрижки клиента

кой базе будет отнимать значительное время у мастеров, особенно принимая во внимание, что база клиентов расположена в основной конфигурации, а мастера работают с мобильных устройств. Поэтому было принято решение производить идентификацию клиента по фото при помощи распознавания лиц. Такой способ позволяет достичь высокой точности и скорости поиска, а также будет являться отличительной чертой данной сети парикмахерских.

### Модуль для управления потоком клиентов

По требованию заказчика модуль управления потоком клиентов должен представлять собой электронную очередь с возможностью встать в эту очередь удаленно. Также было высказано требование минимизировать возможные опоз-

дание клиентов и пропуск ими своей очереди, при этом максимально сократив простой мастеров.

Для этого было предложено отслеживать местоположение клиента, уточнять вид услуги и способ, которым он планирует добираться до салона. Эти данные позволяют рассчитать примерное время, которое клиент потратит на дорогу, и зарезервировать ему место в очереди.

Если в промежуток между последним клиентом мастера и следующим в очереди, который находится в пути, приходит клиент, оказание услуги которому укладывается в свободное время до следующего клиента, то мастер начинает его обслуживание.

## **1.4. Среда реализации модулей**

### **Модуль идентификации клиентов по лицу**

Реализация модуля для идентификации клиентов по лицу и поиску оказанных им ранее услуг выполнялась на платформе «1С: Предприятие», так как сеть парикмахерских и барбершопов «Брадобрей» использует комплекс программных средств фирмы «1С».

«1С: Предприятие» — это программный продукт, разработанный фирмой «1С», который предназначен для разработки прикладных решений, автоматизирующих различные бизнес-процессы организаций [6]. Платформа «1С: Предприятие» является технологическим ядром, на основе которого разрабатываются конфигурации, с которыми взаимодействуют пользователи.

«1С: Предприятие» позволяет запускать разработанные конфигурации на различных типах устройств: на персональных компьютерах, на мобильных устройствах, а также в веб-клиенте через браузер.

Фирмой «1С» были разработаны собственные конфигурации, которые предназначены для автоматизации процессов, характерных для определенных

отраслей бизнеса. Одним из таких отраслевых решений является конфигурация «1С: Салон красоты». Она предназначена для автоматизации управления и учета на предприятиях, оказывающих услуги в индустрии красоты.

Компания «Брадобрей» использует конфигурацию «1С: Салон красоты», а также мобильное приложение для мастеров, при помощи которого они выбирают услуги и пробивают чеки. Так как именно мастеру необходимо знать историю стрижек клиента, то идентификацию было решено производить при помощи камеры мобильного устройства, которым пользуется парикмахер. Модуль идентификации клиента было решено встроить в существующее мобильное приложение.

Мобильная платформа «1С: Предприятие» предназначена для разработки и сборки мобильных приложений. Платформа поддерживает операционные системы Android, iOS и Windows. Разработка под мобильные устройства ведется так же, как и под персональные компьютеры, но существует ряд особенностей. Некоторые объекты дерева конфигурации, например регламентные задания и HTTP-сервисы, не доступны для использования на мобильных устройствах. Однако в платформе существует функциональность, которая является специфичной для мобильных устройств [9]. Для выполнения задач модуля идентификации клиентов большую роль играет возможность использования камеры мобильного устройства.

Таким образом, модуль идентификации клиентов по лицу реализован с использованием мобильной платформы «1С: Предприятие», а также настроено взаимодействие с отраслевой конфигурацией «1С: Салон красоты».

### **Модуль для управления потоком клиентов**

Так как одним из требований заказчика было осуществление возможности вставать в очередь удаленно, то было решено реализовать модуль в виде клиент-серверного взаимодействия, где сервером является конфигурация «1С: Салон

красоты», а клиентом — сайт, на котором посетители салона могут встать в очередь. Использовать отдельные мощности для сервера и СУБД было бы не целесообразно.

Для создания сайта был выбран Vue.js — это фреймворк для разработки пользовательских интерфейсов. В отличие от React.js или Angular Vue.js является масштабируемым вниз, то есть подходит для разработки небольших проектов и при этом он обеспечивает высокую производительность [8].

## Глава 2. Разработка модулей информационной среды парикмахерских

### 2.1. Реализация модуля для идентификации клиентов по лицу

Первым шагом было необходимо научиться получать фотографии при помощи мобильного приложения на платформе «1С: Предприятие». После изучения документации [10], была разработана универсальная функция (см. листинг 1), которая затем была помещена в общий модуль конфигурации и вызывалась с необходимыми параметрами, когда требовалось сделать фотографию.

Листинг 1. Создание фотографии с мобильного устройства

```
&НаКлиенте
Функция ПолучитьДанныеФотоСнимка(ТипФото) Экспорт
    Данные = Неопределено;
    #Если МобильноеПриложениеКлиент Тогда
        Если СредстваМультимедиа.ПоддерживаетсяФотоснимок() Тогда
            РазрКамеры = СредстваМультимедиа
                .ПолучитьПоддерживаемыеРазрешенияКамеры(ТипФото);
            НомерРазрешенияКамеры = ПолучитьРазрешениеКамерыНаСервере(ТипФото);
            Индекс = 0;
            Данные = СредстваМультимедиа.СделатьФотоснимок(ТипФото, РазрКамеры
                [НомерРазрешенияКамеры], 100, , ТипПодсветкиКамеры.Выключена);
        Иначе
            Данные = Новый ФиксированнаяСтруктура("Ошибка", "Данное устройство не поддерживает фотоснимок");
        КонецЕсли;
    #Иначе
        Данные = Новый ФиксированнаяСтруктура("Ошибка", "Фотоснимки можно делать только на мобильном устройстве");
    #КонецЕсли
```



Возврат Данные;  
КонецФункции

В функции использованы константы «РазрешениеПереднейКамеры» и «РазрешениеЗаднейКамеры». Их необходимо заранее определить в настройках приложения. Это связано с тем, что каждое мобильное устройство, оборудованное камерой, способно делать фотографии определенного разрешения и набор этих разрешений различен для разных моделей, поэтому нельзя задать конкретное значение (например, 640x480) для всех устройств.

Для осуществления настройки камеры была создана соответствующая обработка (см. рис. 6), в которой пользователь сам должен выбрать желаемое разрешение из доступных.

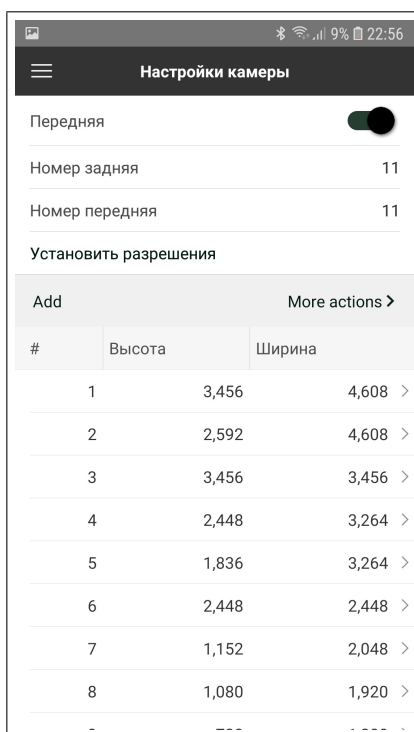


Рисунок 6. Настройка разрешения камеры

Оптимальными с точки зрения производительности для данной задачи являются фотографии в низком разрешении, весом до 250 Кбайт. Это связано с тем, что передача больших изображений занимает больше времени, что от-

рицательно сказывается на производительности системы. Также фотографии в высоком разрешении требуют большой объем памяти для хранения.

После того, как была опробована техническая возможность делать фотографии с мобильного устройства, нужно было создать форму для добавления идентификационного фото клиента. Первоначальная привязка Фото — Клиент должна быть выполнена мастером, следовательно, необходимо было реализовать поиск клиента по базе. Заказчик предложил осуществлять поиск по номеру телефона. Также должна быть возможность добавить нового клиента, указав для него ФИО, номер телефона, email и дату рождения. На форме (см. рис. 7) для добавления идентификационного фото были расположены следующие реквизиты:

- НомерТелефона — строка;
- Фамилия — строка;
- Имя — строка;
- Отчество — строка;
- ДатаРождения — дата;
- Email — строка;
- СогласиеНаПолучениеEmail — булево;
- СогласиеНаПолучениеSMS — булево;
- ИдентификационноеФото — строка.

Для поиска клиента была создана команда `РаспознатьКлиента()`, которая посылает HTTP-запрос с номером клиента в основную базу, а в ответ получает код клиента, если он существует. Исходный код функций, осуществляющих запрос к основной базе для поиска клиента приведен в приложении 1.

Для добавления нового клиента или изменения существующего была создана команда `ПоискИДобавлениеКлиента()` (см. листинг 2), которая посылает HTTP-

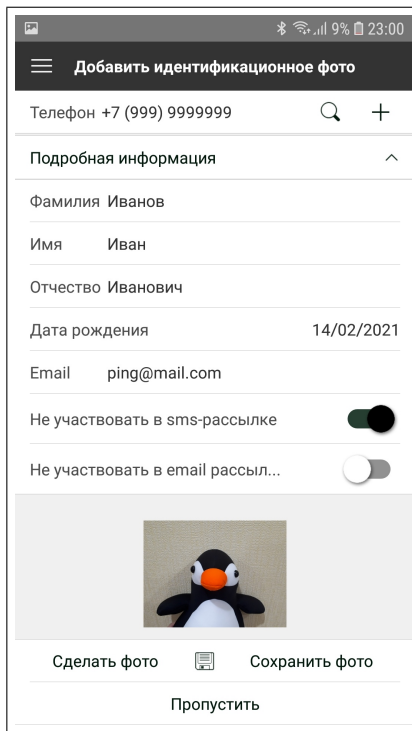


Рисунок 7. Добавление идентификационного фото

запрос с набором вышеперечисленных полей в основную базу, где в зависимости от того, существует ли клиент с таким номер телефона, функция либо обновляет данные, либо создает новый элемент справочника Контрагенты с соответствующими параметрами, а также добавляет для него данные в регистр сведений КонтактнаяИнформация. В ответе приходит код клиента.

Листинг 2. Формирование запроса на добавление или изменение клиента

&НаСервере

Функция ПоискДобавлениеКлиента(Добавлять, СтруктураНовыйКлиент = Неопределено)

Добавлять = ?(Добавлять, 1, 0);

СтруктураJSON = Новый Структура;

Если СтруктураНовыйКлиент <> Неопределено Тогда

СтруктураJSON.Вставить("moreinfo", СтруктураНовыйКлиент);

КонецЕсли;

СтруктураJSON.Вставить("phone", НомерТелефона);

СтруктураJSON.Вставить("add", Добавлять);

СтрокаJSON = ОбменПоHTTP.ПолучитьСтрокуJSON(СтруктураJSON);

```

Ресурс = "brad/hs/camera/findaddclient";
АдресРесурса = СокрЛП(Ресурс);
Результат = ОбменПоHTTP.ВыполнитьЗапрос("POST", "FindAddClient", АдресРесурса
, СтрокаJSON);
ТекстОшибки = "";
КодКлиента = "";
ФотоКлиента = Новый Массив;
Если ТипЗнч(Результат) = Тип("Строка") Тогда
    // Ошибка в приложении
    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = "Ошибка в приложении: " + Результат;
    Сообщение.Сообщить();
ИначеЕсли Результат.Свойство("error", ТекстОшибки) Тогда
    // Ошибка в основной базе
    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = "Ошибка на сервере: " + Результат;
    Сообщение.Сообщить();
ИначеЕсли Результат.Свойство("client_id", КодКлиента) Тогда
    Возврат КодКлиента;
КонецЕсли;
Возврат "";
КонецФункции

```

После того как клиент был найден или создан, мастер фотографирует клиента и сохраняет идентификационное фото. Фото отправляется на сервер в основную базу, записывается в файловое хранилище, а в карточке клиента в справочнике Контрагенты появляется строка с указанием пути до фотографии. Также идентификационное фото отправляется на сервер распознавания лиц, где происходит снятие и сохранение параметров лица.

Распознавание лиц было выполнено при помощи сервиса, созданного сотрудниками ГК «Форус», которые занимаются технической поддержкой информационных систем компании «Брадобрей». Данный сервис при отправке на него

нового фото клиента снимет маску, а затем при распознавании сравнивает маску, переданную на вход, с имеющимися в хранилище. Такой подход позволяет сервису не хранить исходные изображения.

Для взаимодействия мобильного приложения и основной базы был создан HTTP-сервис Камера и разработаны методы, которые обрабатывают входящие запросы. Сервер распознавания и основная база также общаются посредством HTTP-запросов. API представлено в приложении 3.

Следующим этапом реализации модуля было добавление фотографий клиента после оказания услуги. Для этого была доработана форма документа ЧекККМ (см. рис. 8).

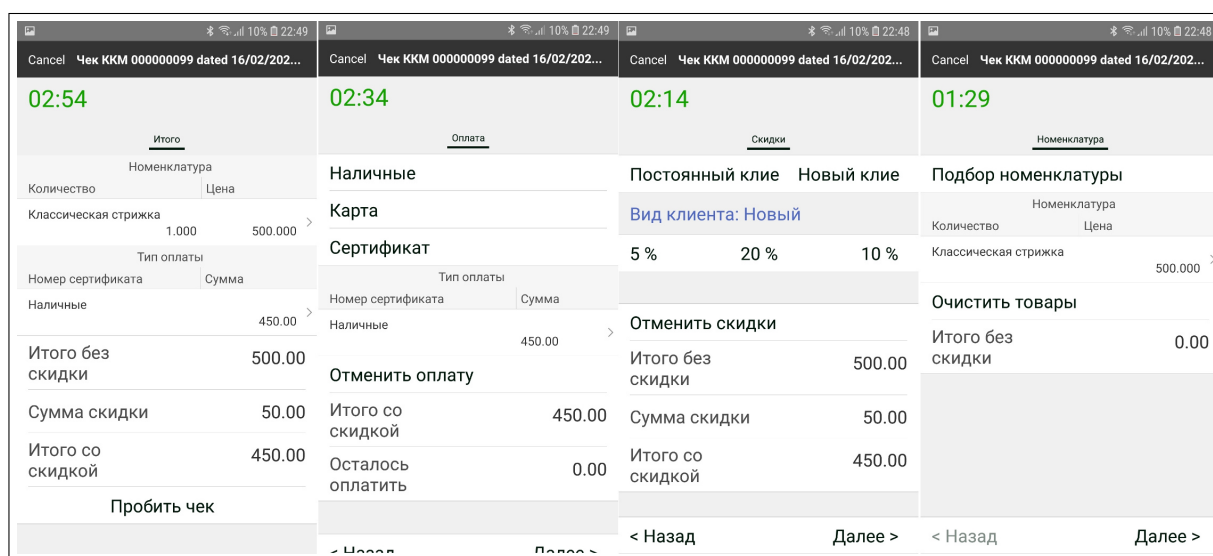


Рисунок 8. Форма документа ЧекККМ

Существующая последовательность действий Выбор номенклатуры — Скидка — Оплата — Пробытие чека была изменена: между выбором скидки и оплатой был добавлен этап фотографирования клиента (см. рис. 9). Мастер должен сделать пять фото по специальной методике и затем нажать на кнопку сохранить.

Также у клиента есть возможность отказаться фотографироваться, в этом случае можно сразу переходить к оплате. Если же клиент желает в будущем такую же стрижку, то после нажатия на кнопку сохранения фотографии и другие данные об услуге попадают в основную базу. В ней был создан регистр сведений

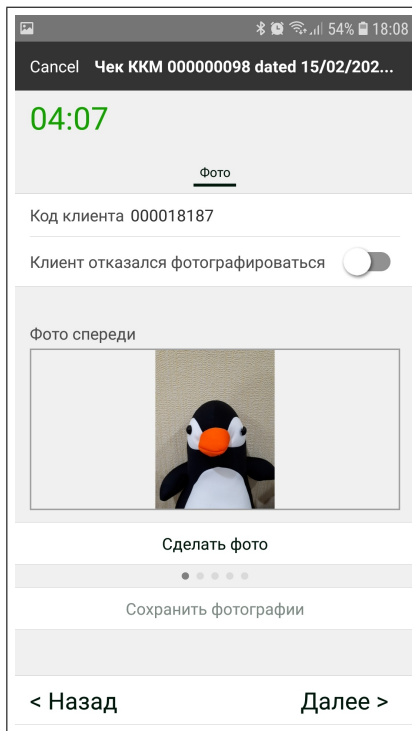


Рисунок 9. Фото после услуг

Хранилище ФотоКлиентов со следующими измерениями:

1. Клиент — СправочникСсылка.Контрагенты;
2. ЧекНомер — строка;
3. ЧекДата — дата.

И ресурсами:

1. ФотоСпереди — строка;
2. ФотоСлева — строка;
3. ФотоСправа — строка;
4. ФотоСзади — строка;
5. ФотоСверху — строка.

Чек записан не ссылкой на документ, а номером документа в виде строки и датой, так как чеки формируются на планшетах мастеров и добавляются в основную базу в конце рабочего дня, а сохранение фотографий нужно обеспечить ещё до момента попадания чека в базу.

Ресурсы представляют собой строки, в которых указан путь к файлу с фотографией. Хранение изображений в файловой системе, а не в базе позволяет осуществлять резервное копирование базы с меньшими ресурсными затратами.

После появления возможности добавлять идентификационное фото клиента и фото после услуг, было реализовано распознавание клиента и отображение связанных с ним фотографий.

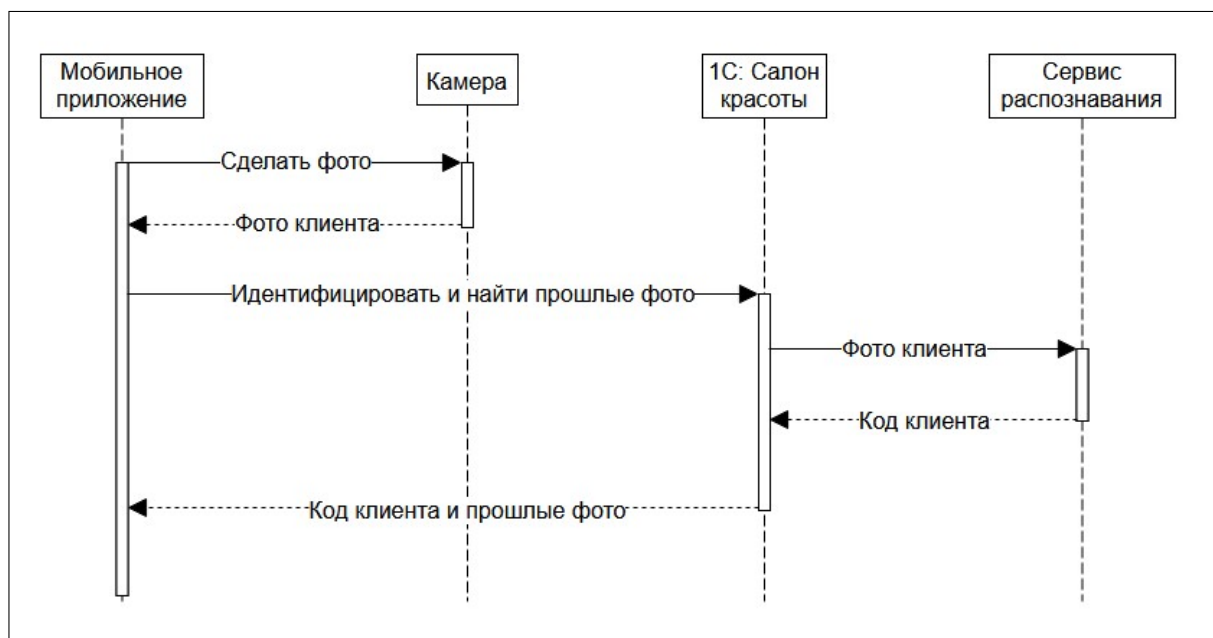


Рисунок 10. Модель взаимодействия

По требованию заказчика сразу после нажатия на кнопку «Услуга» должна включаться камера на мобильном устройстве и происходить распознавание, когда мастер сделает фото. Полученное изображение отправляется в основную базу и затем на сервер распознавания.

В ответ с сервера приходит код клиента, которому принадлежит переданное фото. Далее выполняется запрос к регистру ХранилищеФотоКлиентов и выбираются записи, которые относятся к текущему клиенту. В качестве ответа в мобильное приложение уходят фотографии и соответствующие им наборы услуг. Исходный код функции, выполняющей поиск, приведен в приложении 2. Модель взаимодействия при распознавании представлена на рисунке 10.

После получения изображений в мобильном приложении открывается форма обработки ПрошлыеФото (см. рис. 11). Так как заранее не известно количество оказанных ранее для этого клиента услуг, то все реквизиты добавляются на форму программно.

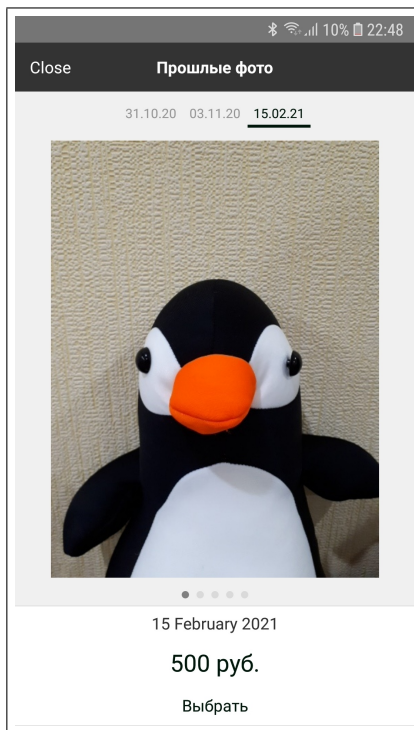


Рисунок 11. Прошлые услуги

После того как клиент выбирает стрижку, мастер нажимает кнопку «Выбрать» на соответствующей вкладке, открывается форма ЧекаККМ, и мастер начинает работу.

## **2.2. Реализация модуля для управления потоком клиентов**

При входе на сайт очереди у клиента запрашивается номер телефона, который подтверждается при помощи SMS.

После подтверждения номера телефона на сервер под управлением «1С: Предприятие» отправляется GET-запрос для получения списка услуг. API взаимо-



действия сайта с сервером приведено в приложении 4. Диаграмма последовательности представлена на рис. 12.

От идентификации клиента в момент записи было решено отказаться, так как в текущей базе не у всех клиентов есть привязка к номеру телефона. Также возможны случаи, когда один человек записывает сразу несколько клиентов, например, отец может записать на свой номер телефона сыновей.

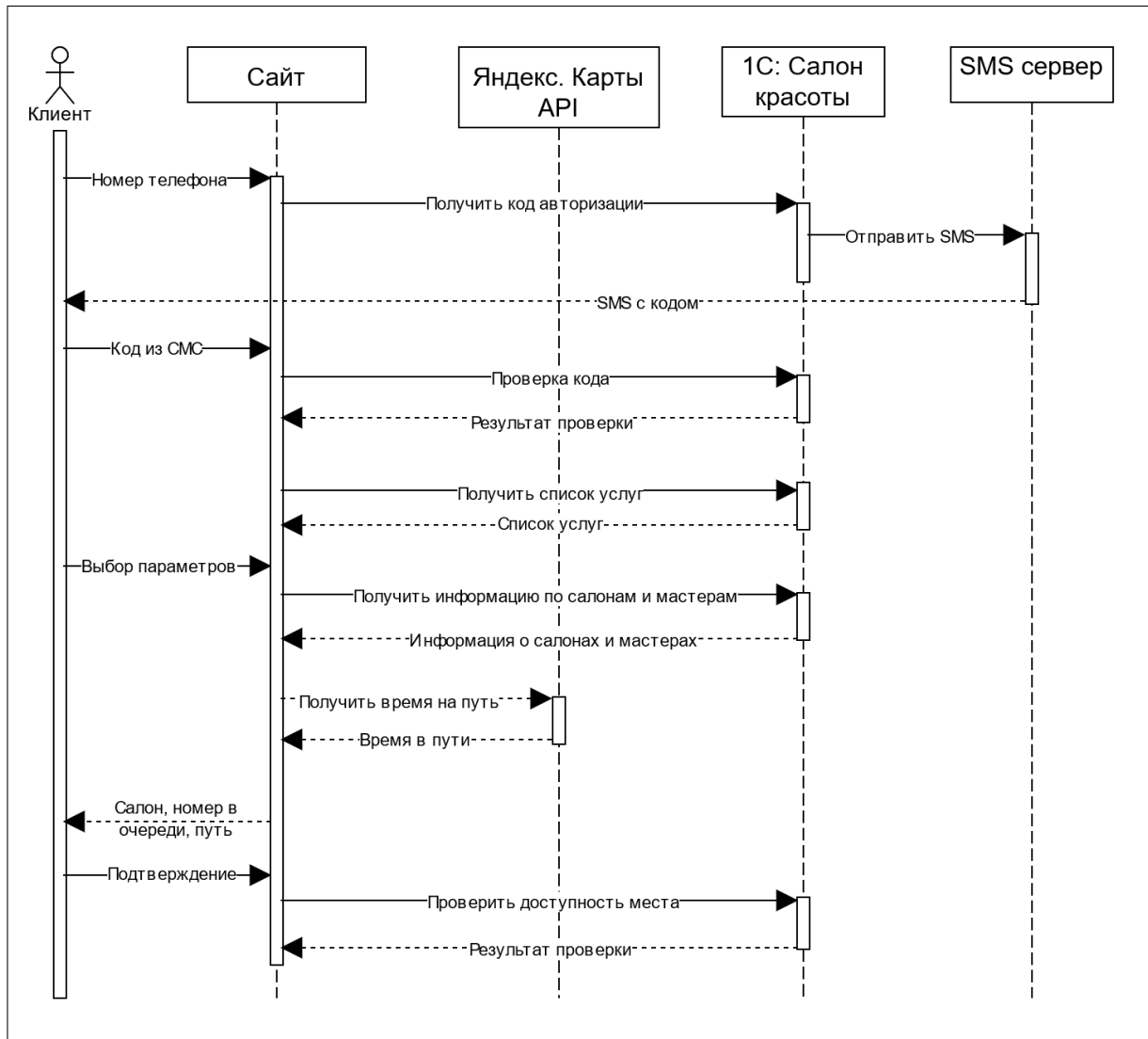


Рисунок 12. Диаграмма последовательности для электронной очереди

Для модуля очереди был создан новый справочник «Форус\_УслугиОчередь». В нем содержатся названия видов услуг, описание и длительность их оказания. Каждому элементу справочника «Номенклатура» сопоставляется элемент

из нового справочника. Это было необходимо, чтобы обеспечить удобство выбора услуг для пользователей сайта. Например, услуга с названием «Борода» на сайте позволяет получить несколько видов стрижки бороды, которые являются разными номенклатурами.

Далее клиент выбирает вид услуги, которую он хочет получить (см. рис. 13).

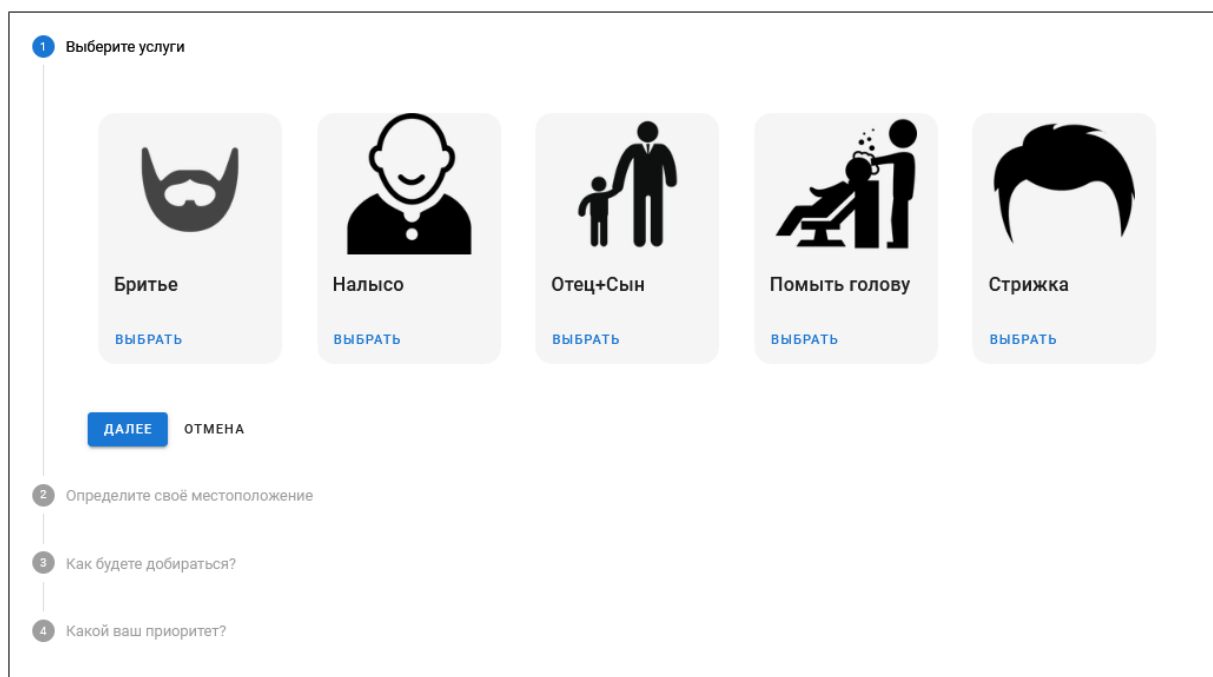


Рисунок 13. Выбор услуги

После выбора услуги клиент нажимает кнопку «Далее» и определяет свое местоположение (см. рис. 14). В зависимости от устройства, на котором используется сайт, точность может отличаться, поэтому клиент может самостоятельно двигать метку своего местоположения. Однако это действие необходимо в основном при использовании сайта на персональном компьютере.

Далее клиент выбирает как он планирует добираться до салона: на автомобиле, на общественном транспорте или пешком. Эта информация необходима для того, чтобы рассчитать, какое время будет потрачено на путь до салона.

После заполнения необходимых параметров на сервер отправляется GET-запрос для получения текущего состояния очереди. В параметрах GET-запроса

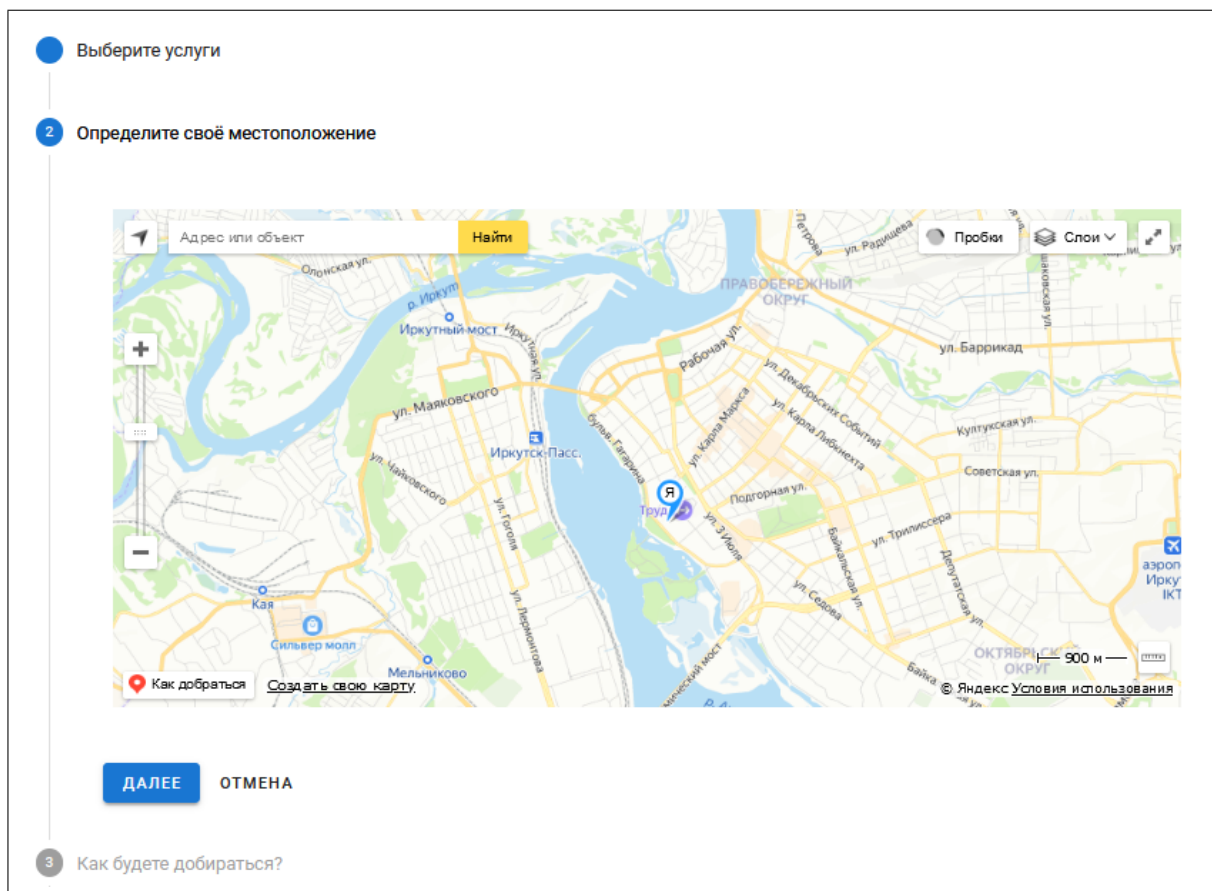


Рисунок 14. Определение местоположения

указывается текущая дата и время. Так как салоны сети «Брадобрей» расположены в различных часовых поясах, а на сервере используется иркутское время (GMT+8), то при помощи API сайта TimeZoneDB [15] время клиента преобразуется к времени по часовому поясу GMT+8.

В конфигурации «1С: Салон красоты» был создан HTTP-сервис «Очередь» для приема запросов с сайта. На GET-запрос о получении текущего состояния очереди сервер генерирует ответ в виде структуры, где содержится информация о салонах и времени их работы, мастерах, которые в этот день оказывают услуги, а также порядок клиентов в очереди.

Для хранения порядка клиентов в очереди был создан регистр сведений «Форус\_Очередь» со следующими измерениями:

1. НомерТелефона — строка;

2. Барбер — СправочникСсылка.Сотрудники;
3. ДатаВремяЗаписи — дата;
4. Салон — СправочникСсылка.СтруктурныеЕдиницы.

И ресурсами:

1. ВремяВПути — число;
2. Услуга — СправочникСсылка.Форус\_УслугиОчередь;
3. ДлительностьУслуги — число;
4. Отменена — булево.

Как упоминалось в разделе 1.3, порядок клиента в очереди зависит от совокупности факторов: местоположения клиента и салонов, способа, которым клиент планирует добираться и вида услуги. Местоположение и транспорт после обработки позволяют получить время, которое необходимо клиенту, чтобы добраться до салона. А вид услуги — время, которое необходимо мастеру на оказание выбранной услуги.

После того как сайт обработал полученную с сервера структуру данных, клиент должен решить, что для него в приоритете: подстричься как можно быстрее или по наименьшей цене. Так же клиент может выбрать конкретный салон или мастера.

Когда все данные для того, чтобы встать в очередь заполнены, алгоритм рассчитывает порядковый номер клиента для очереди. Для этого при помощи API Яндекс.Карт для построения маршрутов [4] вычисляется время, которое необходимо клиенту, чтобы добраться до салона. Затем, если выбран конкретный мастер на прошлом шаге, то алгоритм высчитывает время ожидания. Если к моменту своего приезда клиент будет первым, в очереди, то время ожидания будет равно нулю. Если нет, то алгоритм ищет первый не занятый промежуток времени, в который мастер успеет обслужить клиента, и тогда время от приезда клиента в салон до начала стрижки будет считаться временем ожидания.

Если клиент на предыдущем шаге выбрал запись в конкретный салон, то алгоритм в этом салоне автоматически выбирает мастера, у которого клиент затратит меньше всего времени. Исходный код алгоритма представлен в приложении 3. В упрощенной форме, можно сказать, что время вычисляется по следующей формуле:

$$t = t_{transport} + t_{wait} + t_{servise} * speed_{master} ,$$

где  $t_{transport}$  — время, которое необходимо клиенту, чтобы добраться до салона,  $t_{wait}$  — время на ожидание в очереди после прибытия в салон,  $t_{servise}$  — стандартное время, на оказание выбранной услуги,  $speed_{master}$  — коэффициент скорости мастера.

При выбранном приоритете по скорости алгоритм выбирает мастера также как описано выше, но при этом рассчитывает время для всех мастеров, работающих в этот день.

Для приоритета по стоимости алгоритм выбирает тех мастеров, у которых наименьшая цена на выбранную услугу. Если таких мастеров окажется несколько, то будет выбран тот, который находится в салоне, до которого  $t_{transport}$  является минимальным.

Если промежуток свободного времени между последним клиентом в очереди и новым больше, чем время на минимальную услугу (10 минут), то в этот промежуток будут добавлены «фантомные» клиенты, которые могут быть заменены на настоящих, если они появятся.

Рассмотрим пример с рисунка 15. Клиент К3 встает в очередь в 10:10, ему ехать до салона 45 минут и перед ним в очереди находится один клиент. Однако в промежутке между клиентом К2 и К3 можно успеть оказать услугу еще одному клиенту, если он появится. Поэтому, когда К3 встает в очередь, для него отображается, что перед ним три человека.

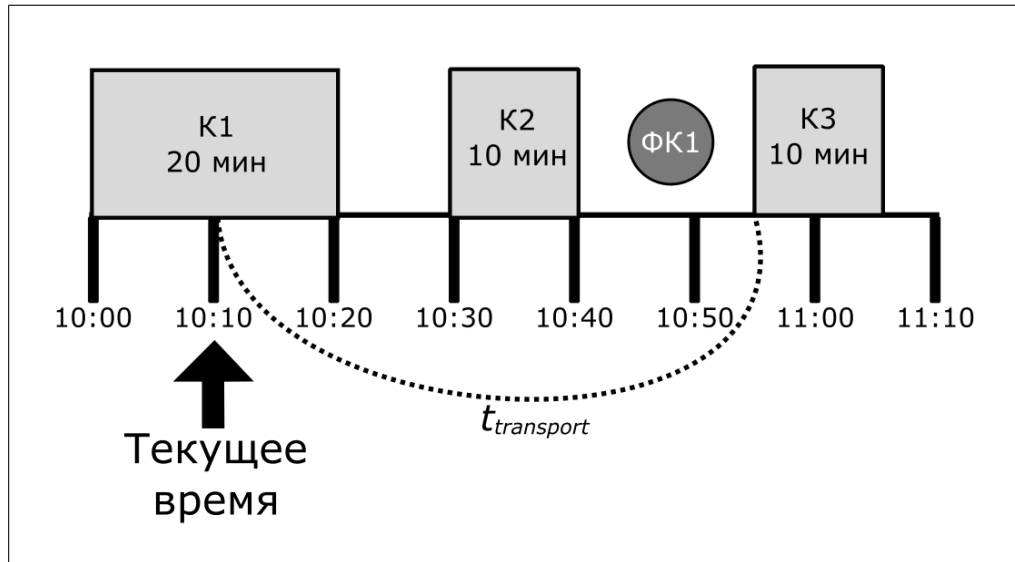


Рисунок 15. Запись клиента К3

Через десять минут на сайте к этому же мастеру на услугу длительностью 15 минут записывается клиент К4, и он будет в салоне уже в 10:45, что позволяет подстричь его раньше клиента К3, если сдвинуть его на пять минут (см. рис. 16). Это становится возможным, так как в отличие от записи электронная очередь не предполагает указания точного времени начала оказания услуги.

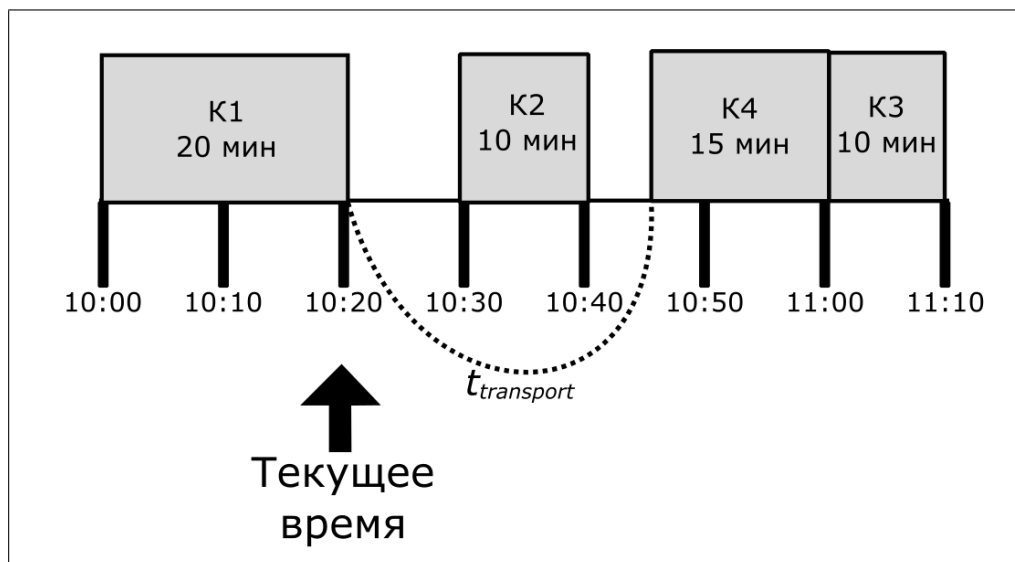


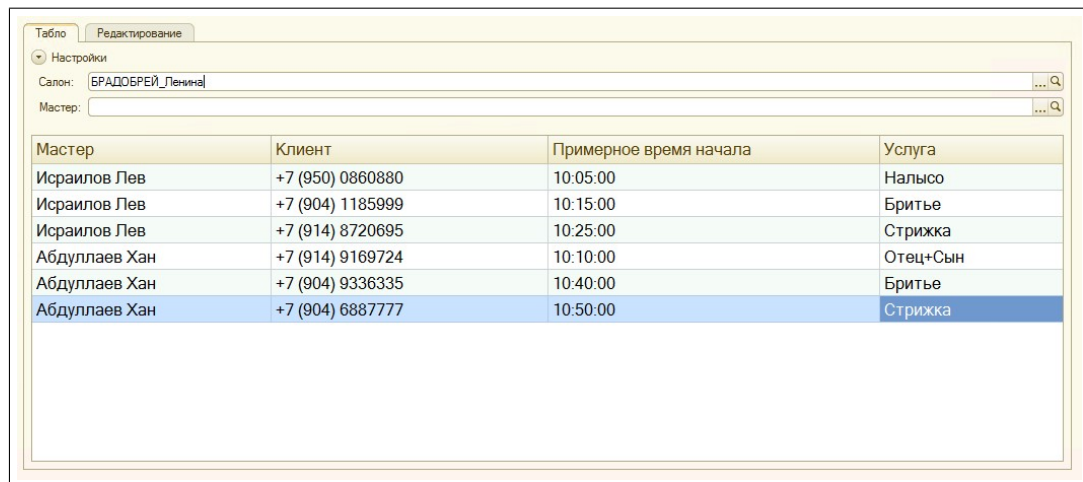
Рисунок 16. Запись клиента К4

Таким образом, данная система электронной очереди позволяет уменьшать количество «окон» между клиентами, что выгодно как бизнесу, так и кли-

ентам.

После того как клиент подтверждает, что он встал в очередь, информация об этом отправляется на сервер, где происходит проверка, не оказалось ли это место занятым другим клиентом, пока происходило заполнение формы. Если такое происходит, то уведомление об этом поступает на сайт и клиент может либо изменить параметры, либо согласиться с новым местом в очереди. Если место оказывается свободным, то в регистр вносится новая запись.

Для администратора салона было реализовано табло для просмотра текущего состояния очереди (см. рис 17). Администратор имеет возможность просматривать очередь вперед, а также видит примерное время начала услуги для клиента. Табло обновляется каждые 30 секунд. Если возникают какие-либо непредвиденные ситуации, то администратор может перейти на вкладку «Редактирование» и изменить очередь вручную.



Мастер	Клиент	Примерное время начала	Услуга
Исраилов Лев	+7 (950) 0860880	10:05:00	Нальсо
Исраилов Лев	+7 (904) 1185999	10:15:00	Бритье
Исраилов Лев	+7 (914) 8720695	10:25:00	Стрижка
Абдуллаев Хан	+7 (914) 9169724	10:10:00	Отец+Сын
Абдуллаев Хан	+7 (904) 9336335	10:40:00	Бритье
Абдуллаев Хан	+7 (904) 6887777	10:50:00	Стрижка

Рисунок 17. Табло администратора

Для клиентов также реализовано табло. Основным его отличием является невозможность просматривать очередь дальше, чем на одного клиента. В целях безопасности на клиентском табло отображается только часть номера телефона.

Когда мастер пробивает чек для клиента, его статус меняется на «Свободен». Если время до следующего клиента меньше, чем минимальное время оказания услуги, то приглашается следующий клиент.

## ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы были реализованы дополнительные модули для информационной системы компании «Брадобрей», позволяющие управлять потоком клиентов, а также производить их идентификацию по лицу. Таким образом, были выполнены все поставленные задачи:

- 1) проведен обзор и сравнительный анализ существующих аналогов для идентификации клиентов по фото и электронной очереди;
- 2) рассмотрены способы организации электронной очереди;
- 3) реализован модуль онлайн-очереди;
- 4) реализован модуль для хранения и поиска фотографий стрижек клиентов;

Все материалы, связанные с разработкой, были переданы в ГК «Форус» для последующего внедрения в сеть парикмахерских и барбершопов «Брадобрей».



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. «Визит-трекер» для «Битрикс 24» [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: <https://helpdesk.bitrix24.ru/open/5037837/> (дата обращения: 01.05.2021).
2. Брадобрей [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: <https://vk.com/bradobrey38> (дата обращения: 01.05.2021).
3. Кулагин, В. Digital@Scale : Настольная книга по цифровизации бизнеса [Текст] / В. Кулагин, А. Сухаревски, Ю. Мефферт. — М. : Интеллектуальная Литература, 2019. — 293 с.
4. Маршрутизация Яндекс [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: <https://yandex.ru/dev/maps/jsapi/doc/2.1/dg/concepts/router/routePanel.html> (дата обращения: 25.05.2021).
5. Методы обработки и распознавания изображений лиц в задачах биометрии [Текст] / Г. А. Кухарев, Е. И. Каменская, Ю. Н. Матвеев, Н. Л. Щеголева. — М. : Политехника, 2013. — 388 с.
6. Обзор системы «1С: Предприятие 8» [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: <https://v8.1c.ru/tekhnologii/overview/> (дата обращения: 01.05.2021).
7. Паспорт федерального проекта «Цифровые сквозные технологии» [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: <https://digital.gov.ru/uploaded/files/pasport-federalnogo-proekta-tsifrovyye-tehnologii.pdf> (дата обращения: 04.06.2021).

8. Сравнение Vue.js с другими фреймворками [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: <https://ru.vuejs.org/v2/guide/comparison.html> (дата обращения: 29.05.2021).
9. Функциональность, специфичная для мобильных устройств [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: <https://v8.1c.ru/platforma/funktsionalnost-spetsifichnaya-dlya-mobilnykh-ustroystv/> (дата обращения: 02.06.2021).
10. Хрусталева, Е. Ю. Знакомство с разработкой мобильных приложений на платформе «1С:Предприятие 8» [Текст] / Е. Ю. Хрусталева. — 2-е издание изд. — М. : ООО «1С-Паблишинг», 2015. — 276 с.
11. Цифровые сквозные технологии [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: <https://cdto.wiki/> (дата обращения: 04.06.2021).
12. Электронная очередь [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: [https://www.banki.ru/wikibank/elektronnaya\\_ochered/](https://www.banki.ru/wikibank/elektronnaya_ochered/) (дата обращения: 01.05.2021).
13. Электронная очередь от компании «BR Studio» [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: <https://app.ocheredi.net/> (дата обращения: 01.05.2021).
14. «Face Recognition» для «Dynamics 365 CRM» [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: [https://appsource.microsoft.com/en-us/product/web-apps/waveaccess.face\\_recognition\\_crm?tab=overview](https://appsource.microsoft.com/en-us/product/web-apps/waveaccess.face_recognition_crm?tab=overview) (дата обращения: 01.05.2021).
15. TimeZoneDB [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: <http://timezonedb.com/> (дата обращения: 02.04.2021).

16. Yclients [Электронный ресурс]. — [Б. м. : б. и.], 2021. — URL: <https://www.usclients.com/> (дата обращения: 01.05.2021).

# ПРИЛОЖЕНИЕ 1.

## Функции для запроса идентификации клиента из мобильного приложения

Листинг 3. Запрос идентификации клиента из мобильного приложения

&НаКлиенте

Процедура РаспознатьКлиента()

ДанныеФото = Форум\_Камера.ПолучитьДанныеФотоСнимка();

Если ТипЗнч(ФС) = Тип("ФиксированнаяСтруктура") Тогда

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = ФС.Ошибка;

Сообщение.Сообщить();

ИначеЕсли ДанныеФото <> Неопределено Тогда

ФотоДД = ДанныеФото.ПолучитьДвоичныеДанные();

Результат = НайтиПоФотоНаСервере(ФотоДД);

ТекстОшибки = "";

КодКлиента = "";

ФотоКлиента = Новый Массив;

Если ТипЗнч(Результат) = Тип("Строка") Тогда

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "Ошибка с фото в приложении: " + Результат;

Сообщение.Сообщить();

ИначеЕсли Результат.Свойство("error", ТекстОшибки) Тогда

Сообщение = Новый СообщениеПользователю;

Сообщение.Текст = "Ошибка с фото на сервере: " + Результат;

Сообщение.Сообщить();

ИначеЕсли Результат.Свойство("photos", ФотоКлиента) Тогда

//Клиент найден, отобразить форму прошлых фото

П = Новый Структура;

П.Вставить("Фотографии", Результат.photos);

П.Вставить("КодКлиента", Результат.client\_id);

Форма = ПолучитьФорму("Обработка.ПрошлыеФото.Форма.Форма", П);

```

Форма.Открыть();
ИначеЕсли Результат.Свойство("client_id", КодКлиента) Тогда
    //Клиент найден, прошлых фото нет, отобразить выбор услуги
    П = Новый Структура;
    П.Вставить("КодКлиента", Результат.client_id);
    Форма = ПолучитьФорму("Обработка.ПрошлыеФото.Форма.Форма1", П);
    Форма.Открыть();
ИначеЕсли Результат.Свойство("not_found") Тогда
    //Клиент не найден, отобразить форму добавления фото
    Форма = ПолучитьФорму
("Обработка.ДобавитьИдентификационноеФото.Форма.Форма");
    Форма.Открыть();
Иначе
    Сообщение = Новый СообщениеПользователю;
    Сообщение.Текст = "Ошибка: " + Результат;
    Сообщение.Сообщить();
КонецЕсли;
КонецЕсли;
КонецПроцедуры.

```

```

Функция НайтиПоФотоНаСервере(ДвоичныеДанныеСКамеры)
    СтрокаBase64 = Base64Строка(ДвоичныеДанныеСКамеры);
    СтруктураJSON = Новый Структура;
    СтруктураJSON.Вставить("photo", СтрокаBase64);
    СтруктураJSON.Вставить("p_type", 111);
    СтрокаJSON = ОбменПоHTTP.ПолучитьСтрокуJSON(СтруктураJSON);
    Ресурс = "brad/hs/camera/find";
    АдресРесурса = СокрЛП(Ресурс);
    Возврат Форум_ОбменПоHTTP.ВыполнитьЗапрос("POST", "FindByPhoto", АдресРесурс
а, СтрокаJSON);
КонецФункции

```

## ПРИЛОЖЕНИЕ 2.

### Поиск истории клиента по его фото

#### Листинг 4. Поиск истории клиента по его фото

```
Функция FindByPhotoFind(Запрос)
Сообщение = Запрос.ПолучитьТелоКакСтроку("UTF-8");
СтруктураJSON = ПолучитьСтруктуруИзСтроки(Сообщение);
ФотоКлиентаBase64 = "";
СтруктураОтвет = Новый Структура;
Если СтруктураJSON.Свойство("photo", ФотоКлиентаBase64) Тогда
ПолученныеКлиенты = ПолучитьКодКлиентаПоФото(СтруктураJSON);
Если ПолученныеКлиенты.Свойство("ids") Тогда
Если ПолученныеКлиенты.ids.Количество() <> 0 Тогда
Запрос = Новый Запрос;
Запрос.Текст =
"ВЫБРАТЬ
| ВложенныйЗапрос.ЧекНомер КАК ЧекНомер,
| ВложенныйЗапрос.ЧекДата КАК ЧекДата,
| ВложенныйЗапрос.ФотоСпереди,
| ВложенныйЗапрос.ФотоСлева,
| ВложенныйЗапрос.ФотоСправа,
| ВложенныйЗапрос.ФотоСзади,
| ВложенныйЗапрос.ФотоСверху,
| ВЫБОР
| КОГДА ЧекККМЗапасы.Количество ЕСТЬ NULL
| ТОГДА 0
| ИНАЧЕ ЧекККМЗапасы.Количество
| КОНЕЦ КАК Количество,
| ВЫБОР
| КОГДА ЧекККМЗапасы.Номенклатура.Код ЕСТЬ NULL
| ТОГДА ""01-00000297""
| ИНАЧЕ ЧекККМЗапасы.Номенклатура.Код
```

```

| КОНЕЦ КАК НоменклатураКод,
| ВЫБОР
|   КОГДА ВложенныйЗапрос.СуммаДокумента ЕСТЬ NULL
|     ТОГДА 500
|   ИНАЧЕ ВложенныйЗапрос.СуммаДокумента
| КОНЕЦ КАК СуммаДокумента
| ИЗ
| (ВЫБРАТЬ
|   Форум_ХранилищеФотоКлиентов.ЧекНомер КАК ЧекНомер,
|   Форум_ХранилищеФотоКлиентов.ЧекДата КАК ЧекДата,
|   Форум_ХранилищеФотоКлиентов.ФотоСпереди КАК ФотоСпереди,
|   Форум_ХранилищеФотоКлиентов.ФотоСлева КАК ФотоСлева,
|   Форум_ХранилищеФотоКлиентов.ФотоСправа КАК ФотоСправа,
|   Форум_ХранилищеФотоКлиентов.ФотоСзади КАК ФотоСзади,
|   Форум_ХранилищеФотоКлиентов.ФотоСверху КАК ФотоСверху,
|   ДокЧекККМ.Ссылка КАК Ссылка,
|   ДокЧекККМ.СуммаДокумента КАК СуммаДокумента
| ИЗ
|   РегистрСведений.Форум_ХранилищеФотоКлиентов КАК Форум
_ХранилищеФотоКлиентов
|     ЛЕВОЕ СОЕДИНЕНИЕ Документ.ЧекККМ КАК ДокЧекККМ
|     ПО Форум_ХранилищеФотоКлиентов.ЧекНомер = ДокЧекККМ.Номер
|       И Форум_ХранилищеФотоКлиентов.ЧекДата = ДокЧекККМ.Дата
| ГДЕ
|   Форум_ХранилищеФотоКлиентов.Клиент = &Клиент) КАК ВложенныйЗапрос
|     ЛЕВОЕ СОЕДИНЕНИЕ Документ.ЧекККМ.Запасы КАК ЧекККМЗапасы
|     ПО ВложенныйЗапрос.Ссылка = ЧекККМЗапасы.Ссылка
|
| УПОРЯДОЧИТЬ ПО
|   ЧекДата
| ИТОГИ ПО
|   ЧекНомер";
Клиент = Справочники.Контрагенты.НайтиПоКоду(ПолученныеКлиенты.ids[0]);

```

```

Запрос.УстановитьПараметр("Клиент", Клиент);
РезультатЗапроса = Запрос.Выполнить();
ВыборкаЧекККМНомер = РезультатЗапроса
.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам);
Если ВыборкаЧекККМНомер.Количество() = 0 Тогда
    СтруктураОтвет.Вставить("p_type", 211);
    СтруктураОтвет.Вставить("client_id", ПолученныеКлиенты.ids[0]);
    Возврат ВернутьУспехИОтвет(СтруктураОтвет);
КонецЕсли;
МассивФото = Новый Массив;
Пока ВыборкаЧекККМНомер.Следующий() Цикл
    ЭтоПерваяУслуга = Истина;
    СтруктураФото = Новый Структура;
    СтруктураФото.Вставить("code", ВыборкаЧекККМНомер.ЧекНомер);
    Выборка = ВыборкаЧекККМНомер.Выбрать();
    МассивНоменклатура = Новый Массив;
    Пока Выборка.Следующий() Цикл
        Если ЭтоПерваяУслуга Тогда
            ЭтоПерваяУслуга = Ложь;
            ФотоСпереди = ВернутьСуществующееФото(Выборка.ФотоСпереди);
            ФотоСлева = ВернутьСуществующееФото(Выборка.ФотоСлева);
            ФотоСправа = ВернутьСуществующееФото(Выборка.ФотоСправа);
            ФотоСзади = ВернутьСуществующееФото(Выборка.ФотоСзади);
            ФотоСверху = ВернутьСуществующееФото(Выборка.ФотоСверху);
            Если ПустаяСтрока(ФотоСпереди) ИЛИ ПустаяСтрока(ФотоСверху)
                ИЛИ ПустаяСтрока(ФотоСлева)
                ИЛИ ПустаяСтрока(ФотоСправа)
                ИЛИ ПустаяСтрока(ФотоСзади) Тогда
                    Возврат ВернутьОшибку(404, "Не найден искомый файл ");
            Иначе
                СтруктураФото.Вставить("front_photo", ФотоСпереди);
                СтруктураФото.Вставить("left_photo", ФотоСлева);
                СтруктураФото.Вставить("right_photo", ФотоСправа);

```



```

        СтруктураФото.Вставить("back_photo", ФотоСзади);
        СтруктураФото.Вставить("up_photo", ФотоСверху);
        КонецЕсли;
        СтруктураФото.Вставить("date", Строка(Формат(Выборка.ЧекДата,
"ДФ=ууууММdd")));
        СтруктураФото.Вставить("cost", Выборка.СуммаДокумента);
        КонецЕсли;
        СтруктураНоменклатура = Новый Структура;
        СтруктураНоменклатура.Вставить("name", Выборка.НоменклатураКод);
        СтруктураНоменклатура.Вставить("quantity", Выборка.Количество);
        МассивНоменклатура.Добавить(СтруктураНоменклатура);
        КонецЦикла;
        СтруктураФото.Вставить("services", МассивНоменклатура);
        МассивФото.Добавить(СтруктураФото);
        КонецЦикла;
        СтруктураОтвет.Вставить("p_type", 211);
        СтруктураОтвет.Вставить("client_id", ПолученныеКлиенты.ids[0]);
        СтруктураОтвет.Вставить("photos", МассивФото);
        Возврат ВернутьУспехИОтвет(СтруктураОтвет);
    Иначе
        СтруктураОтвет = Новый Структура;
        СтруктураОтвет.Вставить("p_type", 213);
        СтруктураОтвет.Вставить("not_found", "");
        Возврат ВернутьУспехИОтвет(СтруктураОтвет);
    КонецЕсли;
Иначе
    Возврат ВернутьОшибка(400, "Не пришел ответ с сервиса распознавания");
КонецЕсли;
Иначе
    Возврат ВернутьОшибка(400, "Не задан параметр img");
КонецЕсли
КонецФункции

```

## ПРИЛОЖЕНИЕ 3.

### Алгоритм выбора времени в очереди у мастера

Листинг 5. Алгоритм выбора времени в очереди у мастера

```
checkMaster(master, currentTime, pathTime){
  const MAX_TIME = 10000000
  var currentTimeSec = this.time2seconds(currentTime)
  var pathTimeSec = pathTime
  var eps = Math.max(5*60, Math.ceil(pathTimeSec * 1.3))
  var minTime = currentTimeSec + pathTimeSec + eps
  var serviceTime = this.selectServices.reduce((acc, { time }) => acc + time,
0) * 60
  var arrayWorkTime = master.salon_workTime.split('-')
  if (currentTimeSec + 3600 < this.time2seconds(arrayWorkTime[0]) || minTime
+ serviceTime > this.time2seconds(arrayWorkTime[1])) {
    return MAX_TIME
  }
  master.bookedTime.sort((a, b) => {
    let valueA = this.time2seconds(a.dateTime.slice(11)) + a.pathTime * 60
    let valueB = this.time2seconds(b.dateTime.slice(11)) + b.pathTime * 60
    return valueA - valueB
  })

  if (master.bookedTime.length === 0) {
    if (pathTimeSec > 3600) {
      return MAX_TIME
    } else {
      return Math.max(minTime, this.time2seconds(arrayWorkTime[0]))
    }
  }
  for (let index = 0; index < master.bookedTime.length; index++){
    let element = master.bookedTime[index]
```

```

    let startTime = this.time2seconds(element.dateTime.slice(11)) +
element.pathTime * 60
    let finishTime = startTime + element.serviceTime * 60
    if (index === 0) {
        if (Math.max(minTime, this.time2seconds(arrayWorkTime[0])) + serviceTime
<= startTime) {
            return Math.max(minTime, this.time2seconds(arrayWorkTime[0]))
        }
    }
    if (index < master.bookedTime.length - 1) {
        let startTimeNext = this.time2seconds(master.bookedTime[index +
1].dateTime.slice(11)) + master.bookedTime[index + 1].pathTime * 60
        if (Math.max(minTime, finishTime) + serviceTime <= startTimeNext) {
            return Math.max(minTime, finishTime)
        }
    } else {
        if (finishTime >= minTime) {
            if (finishTime + serviceTime > this.time2seconds(arrayWorkTime[1])) {
                return MAX_TIME
            } else {
                return finishTime
            }
        } else {
            if (pathTimeSec > 3600) {
                return MAX_TIME
            } else {
                return minTime
            }
        }
    }
}
},
checkPlace(place, currentTime, pathTime){

```

```

var answerMaster = ''
const MAX_TIME = 10000000
var answerTime = MAX_TIME
place.masters.forEach((element) => {
  let promAns = this.checkMaster(element, currentTime, pathTime)
  if (promAns < answerTime) {
    answerTime = promAns
    answerMaster = element.id
  }
})
return {master_id: answerMaster, time: answerTime}
},
checkSalons(){
  var answerMaster2 = ''
  var answerSalon2 = null
  const MAX_TIME = 10000000
  var answerTime2 = MAX_TIME
  this.salonsWithRoute.forEach((element) => {
    let promAns2 = this.checkPlace(element, element.currentTime,
element.timeRoute.value)
    if (promAns2.time < answerTime2) {
      answerTime2 = promAns2.time
      answerMaster2 = promAns2.master_id
      answerSalon2 = element
    }
  })
  return {salon: answerSalon2, master_id: answerMaster2, time: answerTime2}
}

```

## ПРИЛОЖЕНИЕ 4.

### API идентификации клиента

**Адрес:** /find

**Тип:** POST

**Источник:** мобильное приложение

**Описание:** запрос для идентификации клиента по фото.

Таблица 3

Параметры запроса

Поле	Тип	Пример
photo	Строка BASE64	\9iojoisjfije...

Таблица 4

Параметры ответа, если клиент найден и у него есть фото

Поле	Тип	Пример
client_id	Строка	000000016
photos	Массив объектов	[]
front_photo	Строка BASE64	\9ioj...
left_photo	Строка BASE64	\9ioj...
right_photo	Строка BASE64	\9ioj...
back_photo	Строка BASE64	\9ioj...
up_photo	Строка BASE64	\9ioj...
services_id	Массив объектов	[]
name	Число	000000016
quantity	Число	3

Таблица 5

Параметры ответа, если клиент найден и у него нет фото

Поле	Тип	Пример
client_id	Строка	000000016

Таблица 6

Параметры ответа, если клиент не найден

Поле	Тип	Пример
not_found	Строка	

Таблица 7

Параметры ответа, если произошла ошибка

Поле	Тип	Пример
error_id	Число	1
error	Строка	"Не доступен сервис распознавания"

**Адрес:** /addid

**Тип:** POST

**Источник:** мобильное приложение

**Описание:** запрос для добавления идентификационного фото клиента.

Таблица 8

Параметры запроса

Поле	Тип	Пример
client_id	Строка	000000016
photo	Строка BASE64	\9iojois. . .

Таблица 9

## Параметры ответа, если фото добавлено

Поле	Тип	Пример
good	Строка	

Таблица 10

## Параметры ответа, если произошла ошибка

Поле	Тип	Пример
error_id	Число	1
error	Строка	"На фото отсутствует лицо"

**Адрес:** /findaddclient

**Тип:** POST

**Источник:** мобильное приложение

**Описание:** запрос для поиска по номеру телефона и добавления клиента.

Таблица 11

## Параметры запроса

Поле	Тип	Пример
add	Число	1
phone	Строка	89041185999
moreinfo	Объект	
lastname	Строка	Иванов
firstname	Строка	Иван
secondname	Строка	Иванович
birthday	Строка	19990402
email	Строка	mail@mail.com
noemail	Число	0

Продолжение таблицы

posms	Число	0
-------	-------	---

Таблица 12

Параметры ответа

Поле	Тип	Пример
client_id	Строка	000000016

**Адрес:** /addservice

**Тип:** POST

**Источник:** мобильное приложение

**Описание:** запрос для добавления фото после услуги.

Таблица 13

Параметры запроса

Поле	Тип	Пример
client_id	Строка	000000016
check_id	Строка	000000356
date	Строка	20210624
front	Строка BASE64	\9ioj...
left	Строка BASE64	\9ioj...
right	Строка BASE64	\9ioj...
back	Строка BASE64	\9ioj...
up	Строка BASE64	\9ioj...

Таблица 14

Параметры ответа, если фото добавлено

Поле	Тип	Пример
good	Строка	



## Параметры ответа, если произошла ошибка

<b>Поле</b>	<b>Тип</b>	<b>Пример</b>
error_id	Число	1
error	Строка	"Произошла ошибка на сервере"

## ПРИЛОЖЕНИЕ 5.

### API электронной очереди

**Адрес:** /auth?phone=

**Тип:** GET

**Источник:** сайт

**Описание:** запрос кода для авторизации через СМС.

Таблица 16

Параметры запроса

Поле	Тип	Пример
phone	Строка	89041185999

**Адрес:** /authcode?phone=&code=

**Тип:** GET

**Источник:** сайт

**Описание:** запрос проверки правильности кода для авторизации.

Таблица 17

Параметры запроса

Поле	Тип	Пример
phone	Строка	89041185999
code	Число	1234

Таблица 18

Параметры ответа, если код верен

Поле	Тип	Пример
status	Число	200
token	Строка	bas7ph7dfan7ana8

Параметры ответа, если код не верен

Поле	Тип	Пример
status	Число	400

**Адрес:** /services

**Тип:** GET

**Источник:** сайт

**Описание:** запрос списка услуг.

Параметры ответа

Поле	Тип	Пример
services	Массив объектов	[]
id	Строка	000000005
name	Строка	Отец и сын
time	Число	15
ignoreCoeff	Булево	false
img	Строка BASE64	\9iojoisjfije...
description	Строка	Классическая стрижка для отца и сына

**Адрес:** /info?date=

**Тип:** GET

**Источник:** сайт

**Описание:** запрос информации о салонах и мастерах.

## Параметры запроса

Поле	Тип	Пример
date	Строка	20210624

## Параметры ответа

Поле	Тип	Пример
salons	Массив объектов	[]
id	Строка	000000005
name	Строка	Отец и сын
address	Строка	ул. Партизанская, 49
coords	Строка	104,522,52,556565
workTime	Строка	09:00-21:00
masters	Массив объектов	[]
id	Строка	000000005
name	Строка	Отец и сын
avatar	Строка BASE64	\9iojoisjfje...
speedCoeff	Число	0,9
language	Число	89
quality	Число	95
rating	Число	4,9
bookedTime	Массив объектов	[]
dateTime	Строка	24.06.2021 11:55
serviceTime	Число	15
pathTime	Число	20

Адрес: /waitinline

**Тип:** POST

**Источник:** сайт

**Описание:** запрос на добавление в очередь.

Таблица 23

Параметры запроса

Поле	Тип	Пример
phone	Строка	89041185999
services	Строка	000000005
masterId	Строка	000000001
salonId	Строка	000000003
dateTime	Строка	20210624115500
pathTime	Число	20
duration	Число	15

Таблица 24

Параметры ответа, если место свободно

Поле	Тип	Пример
status	Число	400

Таблица 25

Параметры ответа, если возникла ошибка

Поле	Тип	Пример
status	Число	404
error	Строка	400

## Параметры ответа, если место занято

Поле	Тип	Пример
status	Число	400
salons	Массив объектов	[]
id	Строка	000000005
name	Строка	Отец и сын
address	Строка	ул. Партизанская, 49
coords	Строка	104,522,52,556565
workTime	Строка	09:00-21:00
masters	Массив объектов	[]
id	Строка	000000005
name	Строка	Отец и сын
avatar	Строка BASE64	\9iojoisjfije...
speedCoeff	Число	0,9
language	Число	89
quality	Число	95
rating	Число	4,9
bookedTime	Массив объектов	[]
dateTime	Строка	24.06.2021 11:55
serviceTime	Число	15
pathTime	Число	20