

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАБОТА ПРОВЕРЕНА

Рецензент
Начальник отдела администрирования
прикладных систем
ПАО «ЧЕЛЯБИНВЕСТБАНК»
_____ Ю.А. Крючко
“ ___ ” _____ 2017 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор
_____ Л.Б. Соколинский
“ ___ ” _____ 2017 г.

**РАЗРАБОТКА МОБИЛЬНОГО КЛИЕНТСКОГО
ПРИЛОЖЕНИЯ ДЛЯ СЕРВИСА «TABLETIME»
НА ПЛАТФОРМЕ WINDOWS 10**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2017.13-008-1401.ВКР

Научный руководитель
старший преподаватель кафедры СП
_____ К.В. Бородулин

Автор работы,
студент группы КЭ-401
_____ А.Д. Гасенко

Ученый секретарь
(нормоконтролер)
_____ О.Н. Иванова

“ ___ ” _____ 2017 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ	11
2.1. Функциональные требования	11
2.2. Нефункциональные требования	11
2.3. Варианты использования приложения	11
2.4. Архитектура системы	13
2.4.1. Архитектура разрабатываемого приложения.....	13
2.4.2. Схема базы данных приложения «TableTime»	15
2.5. Взаимодействие с сервером системы.....	16
2.6. Проектирование дизайна приложения.....	17
3. РЕАЛИЗАЦИЯ СИСТЕМЫ	21
3.1. Средства реализации.....	21
3.2. Реализация приложения	21
3.2.1. Реализация пользовательского интерфейса	21
3.2.2. Работа с сообщениями Firebase Cloud Messaging	25
4. ТЕСТИРОВАНИЕ	27
ЗАКЛЮЧЕНИЕ	31
ЛИТЕРАТУРА.....	32
ПРИЛОЖЕНИЕ	34

ВВЕДЕНИЕ

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Universal Windows Platform (UWP) – платформа, созданная компанией Microsoft, представленная на операционной системе Windows 10. Целью данной платформы является помощь в создании универсальных приложений Windows, запускаемых как на Windows 10, так и на Windows 10 Mobile без серьезных изменений в коде.

Заведение – место, в котором можно осуществить бронь столика, к примеру, рестораны или кафе.

Пользователь – человек, использующий приложение для бронирования столиков.

АКТУАЛЬНОСТЬ

За последние несколько лет мобильный Интернет сделал огромный шаг вперед в плане как увеличения зон покрытия и скорости, так и уменьшения стоимости услуг [13]. Люди стали чаще использовать различные мобильные приложения для заказа или покупки товаров и услуг, а также бронирования. Такие приложения позволяют не совершать телефонные звонки, поскольку пользователь либо не всегда может говорить по телефону, либо звонок может обернуться длительным ожиданием или тратой денежных средств.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ

Целью данной работы является разработка мобильного клиентского приложения для сервиса «TableTime» на платформе Windows 10.

Для достижения поставленной цели необходимо решить следующие задачи:

- выполнить анализ предметной области и произвести обзор существующих решений;
- изучить методы разработки приложений для платформы Universal Windows Platform;
- разработать архитектуру приложения;

- реализовать приложение;
- выполнить тестирование приложения.

ОБЗОР ЛИТЕРАТУРЫ

В работе [1] рассказываются основные принципы разработки приложений для платформы Windows 10. В работе [2] содержится информация о базах данных и работах с ними. Работа [3] рассказывает о функциональном тестировании приложений. В работе [4] содержится информация о языке UML и его применении. В работах [5, 6] описана разработка приложений на языке C#, в частности на платформе WPF. По ссылкам [10, 14, 15] представлены приложения, аналогичные разрабатываемому. По ссылке [8] содержится информация о сервисе Firebase Cloud Messaging. По ссылке [7] рассматривается пример создания локальной базы данных SQLite для платформы UWP. По ссылке [12] описаны основные принципы разработки и реализации дизайна для платформы UWP.

СТРУКТУРА И ОБЪЕМ РАБОТЫ

Работа состоит из введения, четырех глав, заключения, библиографии и приложения. Объем работы составляет 33 страницы, объем библиографии – 15 источников, объем приложения – 8 страниц.

СОДЕРЖАНИЕ РАБОТЫ

Во *введении* обосновывается актуальность, формируется цель и задачи работы.

Первая глава «Анализ предметной области» содержит обзор аналогичных проектов.

Вторая глава «Проектирование системы» содержит описание и анализ требований к мобильному приложению системы «TableTime», а также описание архитектуры и проектирование дизайна приложения.

Третья глава «Реализация системы» описывает реализацию приложения для платформы Windows 10.

Четвертая глава «Тестирование системы» описывает функциональное тестирование приложения.

В заключении описываются основные результаты, полученные при выполнении дипломной работы.

В приложении представлен графический интерфейс пользователя мобильного и десктопного приложений.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Обзор существующих решений

TopTable

TopTable – мобильное приложение для смартфонов на базе операционных систем Android и iOS для резервирования столиков в ресторанах города Баку. Приложение позволяет посмотреть расположение ресторанов на карте, текущие акции, действующие в заведениях, а также выполнить сортировку по типу кухни.

Экраны приложения TopTable изображены на рис. 1.

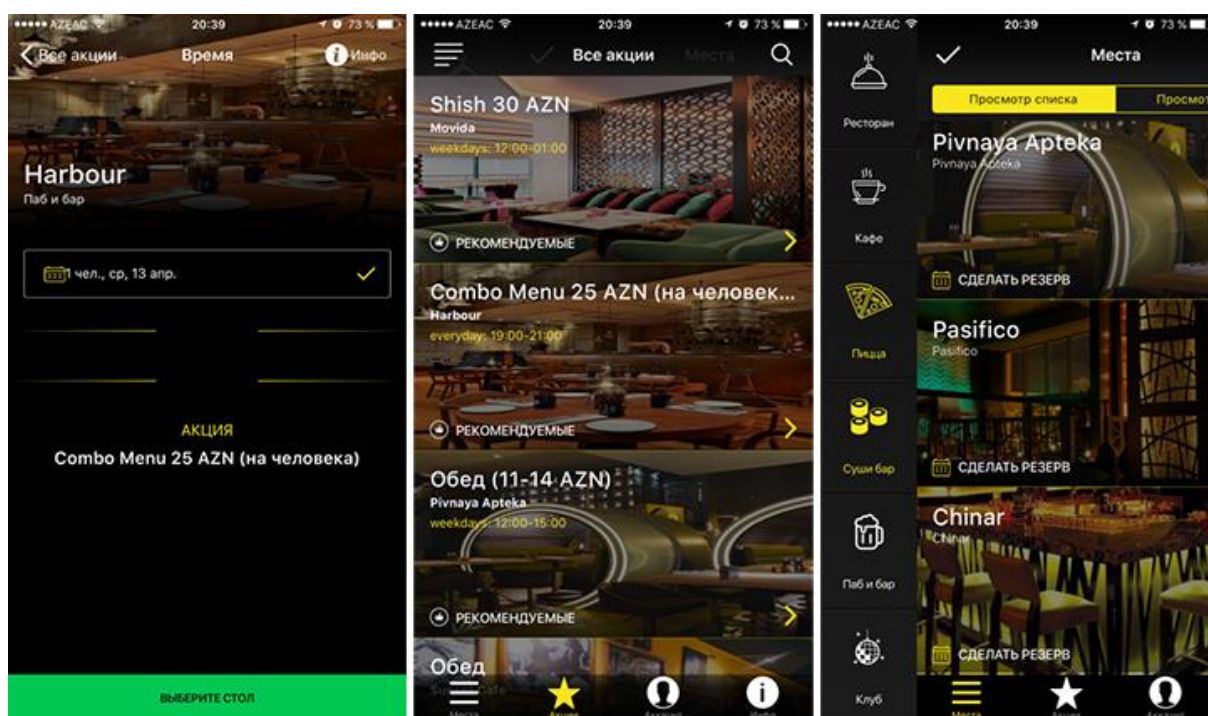


Рис. 1. Экраны приложения TopTable

После того, как пользователь выбрал заведение, в которое он хочет пойти, он может забронировать столик на любое время. Затем пользователю позвонит менеджер заведения для уточнения информации о бронировании.

Столики

Столики – мобильное приложение для смартфонов на операционной системе Android, которое позволяет забронировать столик в ресторане [14].

Пользователь должен выбрать заведение, заполнить заявку, в которой указываются дата, количество персон, желаемое время и контактный

телефон. После отправки заявки на смартфон придет смс со статусом заявки, который также отображается и внутри приложения. Заявку можно отменить в истории бронирования.

На рис. 2 представлены экраны приложения Столики.

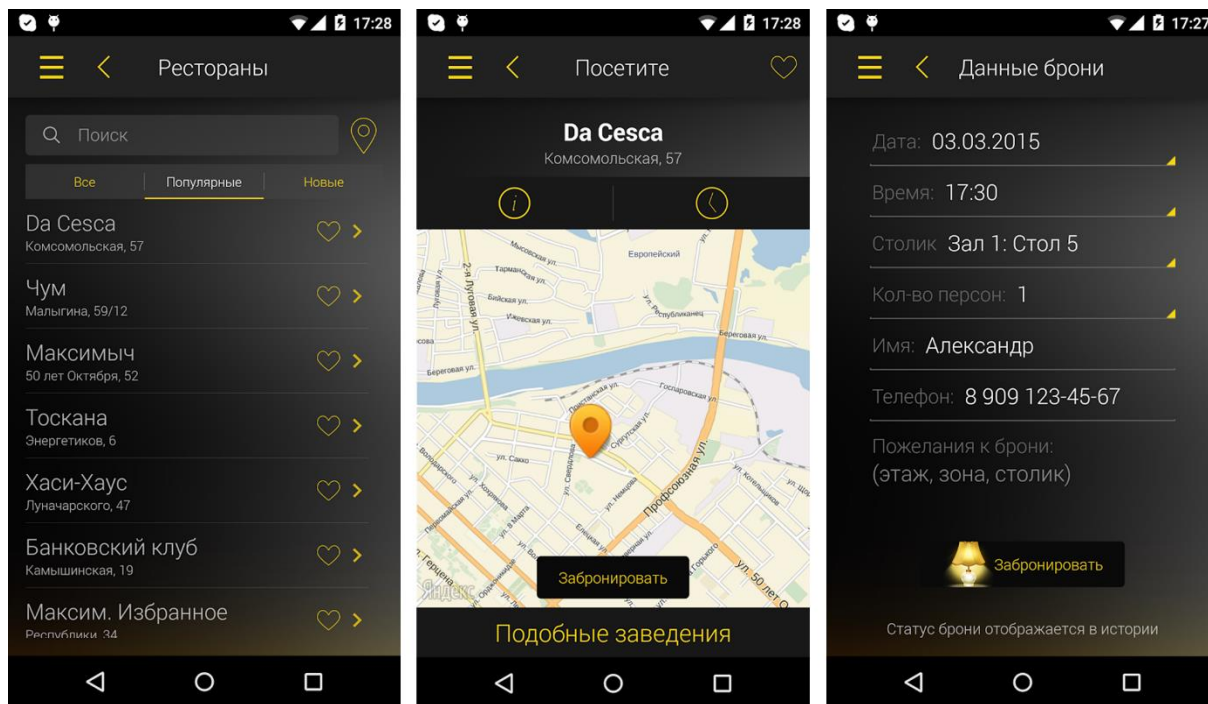


Рис. 2. Экраны приложения Столики

Приложение также позволяет посмотреть местоположение ресторанов на карте, однако сортировку по типу кухни осуществить уже не удастся.

Афиша–Рестораны

«Афиша-рестораны» – приложение для операционной системы Windows 10, позволяющее забронировать столик в ресторане [10].

Пользователю предоставляется возможность поиска заведения и бронирования столиков, просмотр адресов заведений, их телефонов, а также список обзоров и рекомендаций.

Предусмотрена возможность сортировки по типу заведения.

На рис. 3 представлен основной экран приложения Афиша-рестораны.

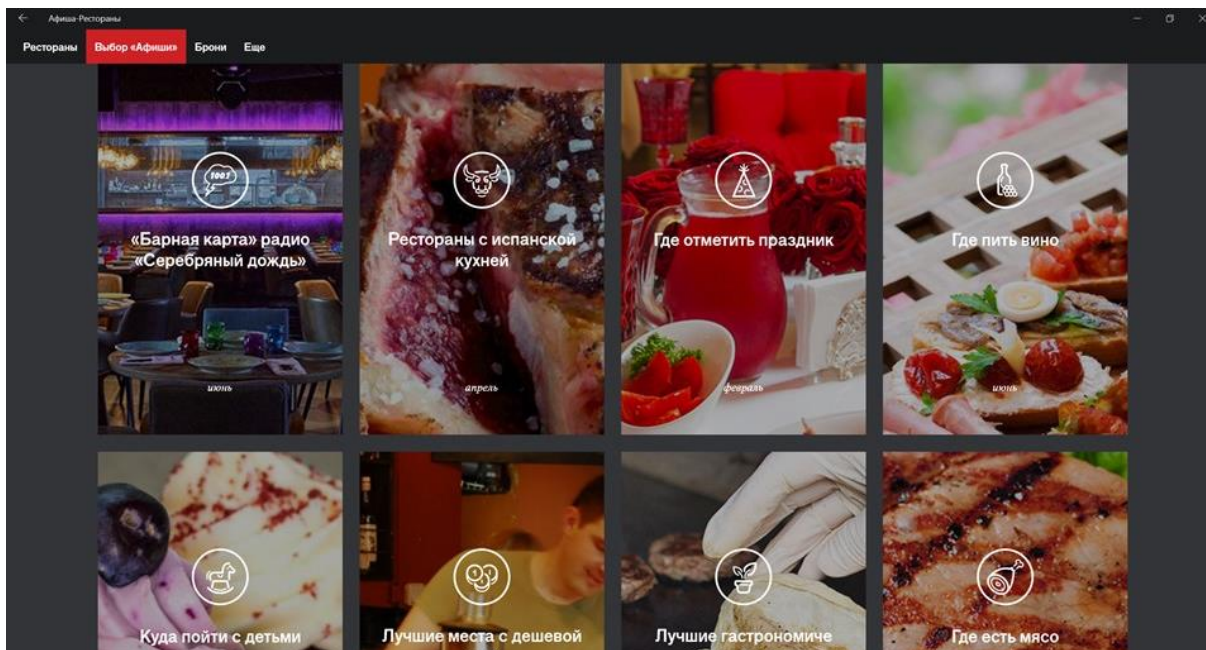


Рис. 3. Экран приложения Афиша-рестораны

ТоМесто

«ТоМесто» – приложение для операционной системы iOS [15]. На рис. 4 представлен один из экранов приложения ТоМесто.

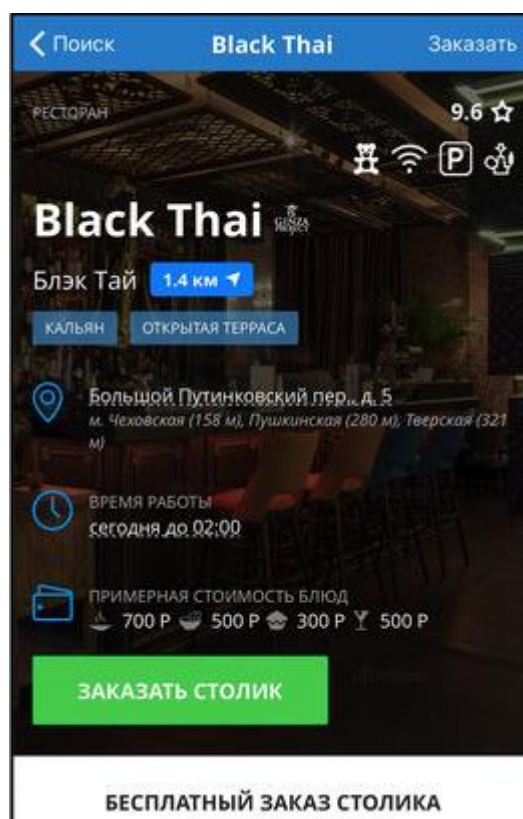


Рис. 4. Экран приложения ТоМесто

Приложение позволяет просматривать меню заведения, отзывы, фотографии, режим работы, описание места и др. Присутствует возможность бронирования столиков.

Также на экране заведения пользователь может посмотреть, сколько других пользователей помимо него просматривают данное заведение.

Уведомления о просмотре заведения некоторым числом человек могут перекрывать элементы управления, а сервис определения местоположения не всегда работает корректно.

Вывод

После изучения существующих аналогов было определено, что актуальными на текущий момент являются приложения «ТоМесто» и «Афиша-Рестораны».

Обзор аналогов показал, что в разрабатываемом приложении должна быть реализована работа без подключения к сети Интернет с заранее загруженной информацией.

Кроме того, приложение должно обладать интуитивно понятным интерфейсом и лаконичным дизайном, которые позволят пользователю комфортно пользоваться приложением.

2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1. Функциональные требования

Разрабатываемое приложение для бронирования столиков должно удовлетворять следующим функциональным требованиям:

- приложение должно позволять пользователю просматривать без подключения к Интернету предварительно загруженную информацию, а именно: списки заведений, подробную информацию о заведениях и список акций;
- приложение должно предоставлять пользователю возможность менять свои персональные данные: имя, адрес электронной почты и номер телефона;
- приложение должно предоставлять пользователю информацию о заведениях: фотографии заведения, схему зала, меню и акции;
- приложение должно позволять пользователю осуществлять поиск блюда по названию;
- приложение должно позволять пользователю создавать заявки на бронирование. Заявка создается с указанием времени, даты, количества персон. Номер столика и комментарий пользователя задаются по желанию;
- приложение должно позволять отменять заявки на бронирование.

2.2. Нефункциональные требования

В результате анализа функциональных требований и обзора аналогичных проектов были сформулированы следующие нефункциональные требования:

- приложение должно быть написано на языке C# под платформу Windows 10 с использованием технологии Universal Windows Platform [1, 5, 6].

2.3. Варианты использования приложения

Диаграмма вариантов использования мобильного приложения представлена на рис. 5 [3]. С приложением взаимодействует два актера – зарегистрированный и незарегистрированный пользователи.

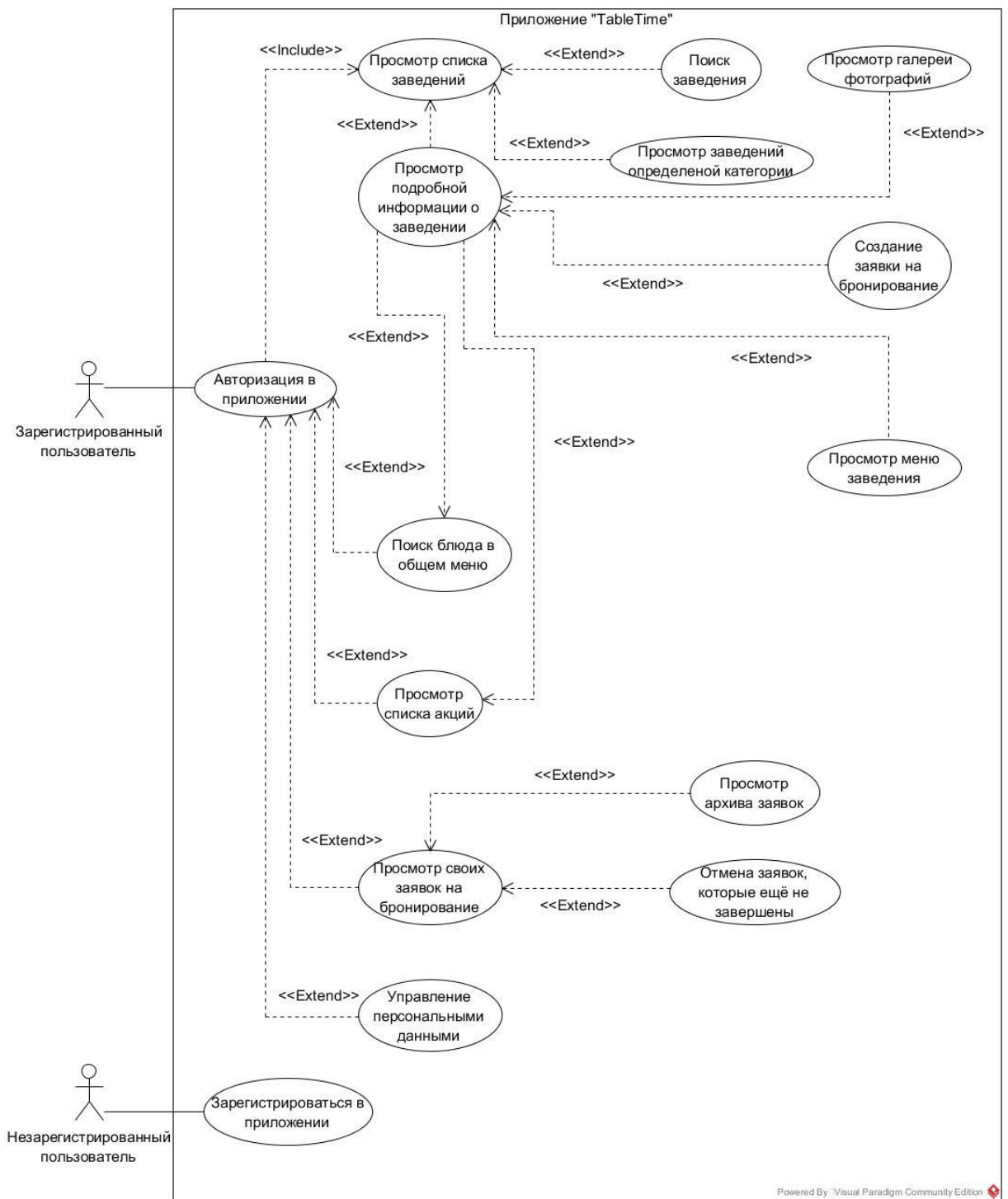


Рис. 5. Диаграмма вариантов использования мобильного приложения «TableTime»

Зарегистрированный пользователь может *авторизоваться в приложении* через номер телефона или адрес электронной почты. Без авторизации дальнейшее использование приложения невозможно.

Зарегистрированный пользователь может *посмотреть список заведений*.

Зарегистрированный пользователь может *выполнить поиск по заведениям*, набрав название нужного заведения.

Зарегистрированный пользователь может посмотреть *заведения определенной категории*, к примеру, кофейни или рестораны.

Зарегистрированный пользователь может посмотреть *подробную информацию о заведении*, где содержатся название, адрес, телефон заведения, а также схема зала.

Из подробной информации зарегистрированный пользователь может открыть *галерею фотографий* из заведения.

Зарегистрированный пользователь может *посмотреть меню заведения*, то есть список блюд, которые подаются в этом заведении.

Зарегистрированный пользователь может посмотреть *список акций*, которые проходят в заведениях.

Зарегистрированный пользователь может *выполнить поиск по блюду* в общем меню приложения. В результате поиска возвращается список заведений, где присутствует данное блюдо.

Зарегистрированный пользователь может *просматривать свои заявки на бронирование* и отслеживать их статус. Если заявка еще не обработана, или уже принята, но еще не завершена, пользователь может *отменить незавершенную заявку*. Также пользователь может *просматривать свой архив заявок*. В нем содержатся отмененные и завершенные заявки.

Зарегистрированный пользователь может *управлять персональными данными*, т.е. изменять имя, номер телефона и адрес электронной почты.

Незарегистрированный пользователь может *зарегистрироваться* в приложении.

2.4. Архитектура системы

2.4.1. Архитектура разрабатываемого приложения

На рис. 6 изображена диаграмма компонентов приложения.

Компонент меню – компонент, отвечающий за предоставление меню заведения по категориям и поиск заданного блюда по всем заведениям.

Компонент бронирований – компонент, который используется для получения истории бронирований, а также создания новых заявок и изменения статуса незавершенных заявок.

Компонент авторизации – компонент, предназначенный для регистрации или авторизации пользователя при необходимости.

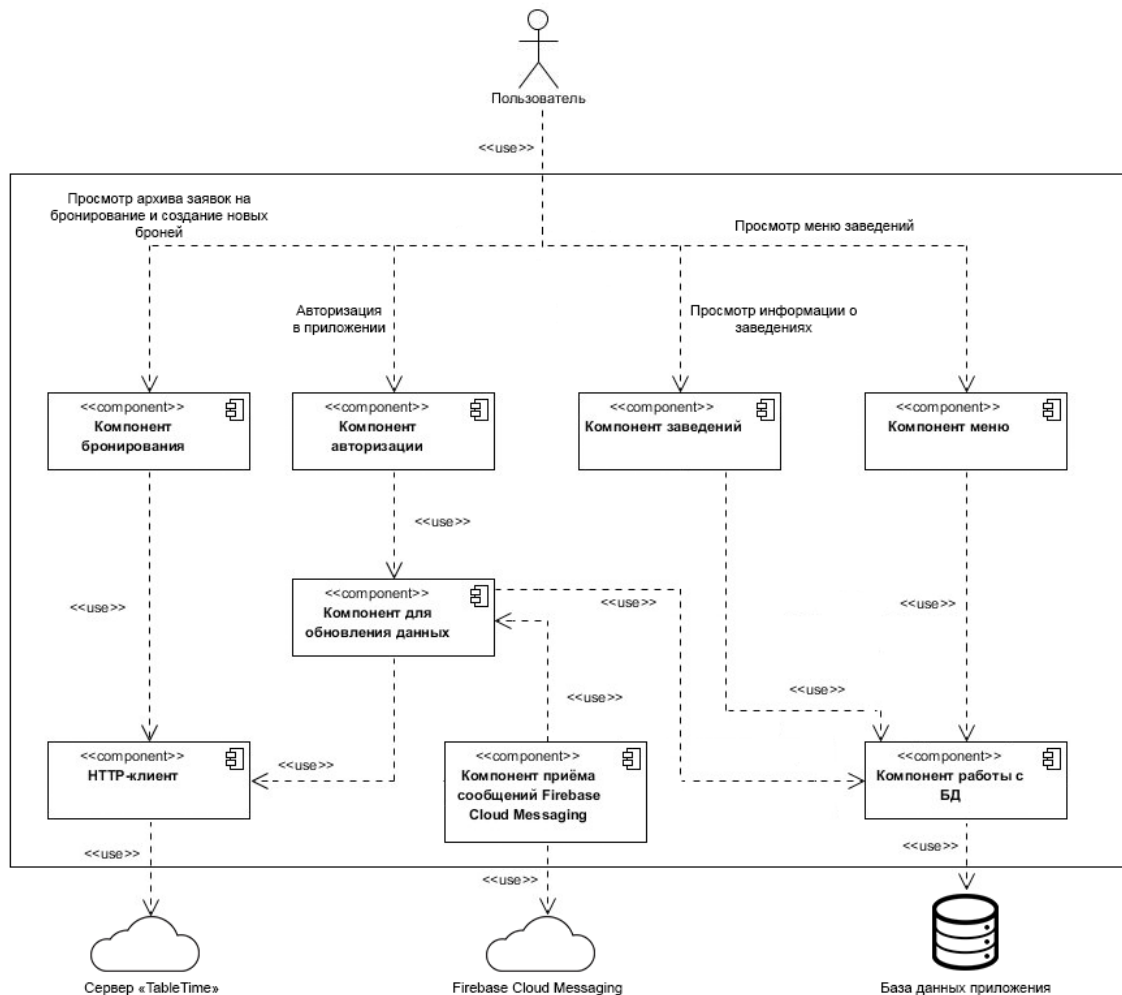


Рис. 6. Компоненты приложения «TableTime»

Компонент заведений – компонент, который используется для получения и предоставления информации о заведениях. Данный компонент отвечает также за поиск заведений по названию и сортировке по категориям.

Компонент обновления данных – компонент, обновляющий данные в базе данных приложения при нескольких условиях: пользователь авторизовался в приложении, сервер системы доступен и с момента последнего обновления данных прошло не менее суток.

HTTP клиент – компонент, реализующий взаимодействие с сервером системы «TableTime» посредством обращения к REST-интерфейсу сервера.

Менеджер работы с базой данных – компонент, содержащий конфигурационные данные для соединения с базой данных, скрипт инициализации базы данных и скрипт создания схемы базы данных.

Сервис приема сообщений Firebase Cloud Messaging – сервис, обрабатывающий входящие сообщения от сервиса Firebase Cloud Messaging и оповещающий пользователя о полученных сообщениях [8].

2.4.2. Схема базы данных приложения «TableTime»

На рис. 7 представлена схема базы данных приложения [4]. Семантика таблиц локальной базы данных приложения соответствует семантике одноименных таблиц сервера системы.

В базе данных присутствуют следующие таблицы:

– *Reservation* – таблица, в которой содержится информация о заявках пользователя на бронирование столиков;

– *Tag* – таблица, в которой хранятся теги для пунктов меню;

– *Menu* – таблица, в которой хранится информация о блюдах;

– *Menu_Type* – таблица-перечисление, в которой хранятся категории пунктов меню;

– *Place* – таблица, в которой хранятся данные о заведениях;

– *Place_Type* – таблица-перечисление, в которой хранятся категории заведений;

– *Image* – таблица для хранения URL, которые, в свою очередь, используются для доступа к фотографиям заведений;

– *Status* – таблица-перечисление для хранения типов заявок;

– *Schedule* – таблица для хранения часов работы в течение недели заведений;

– *Specials* – таблица для хранения акций, которые на текущий момент проходят в заведениях.

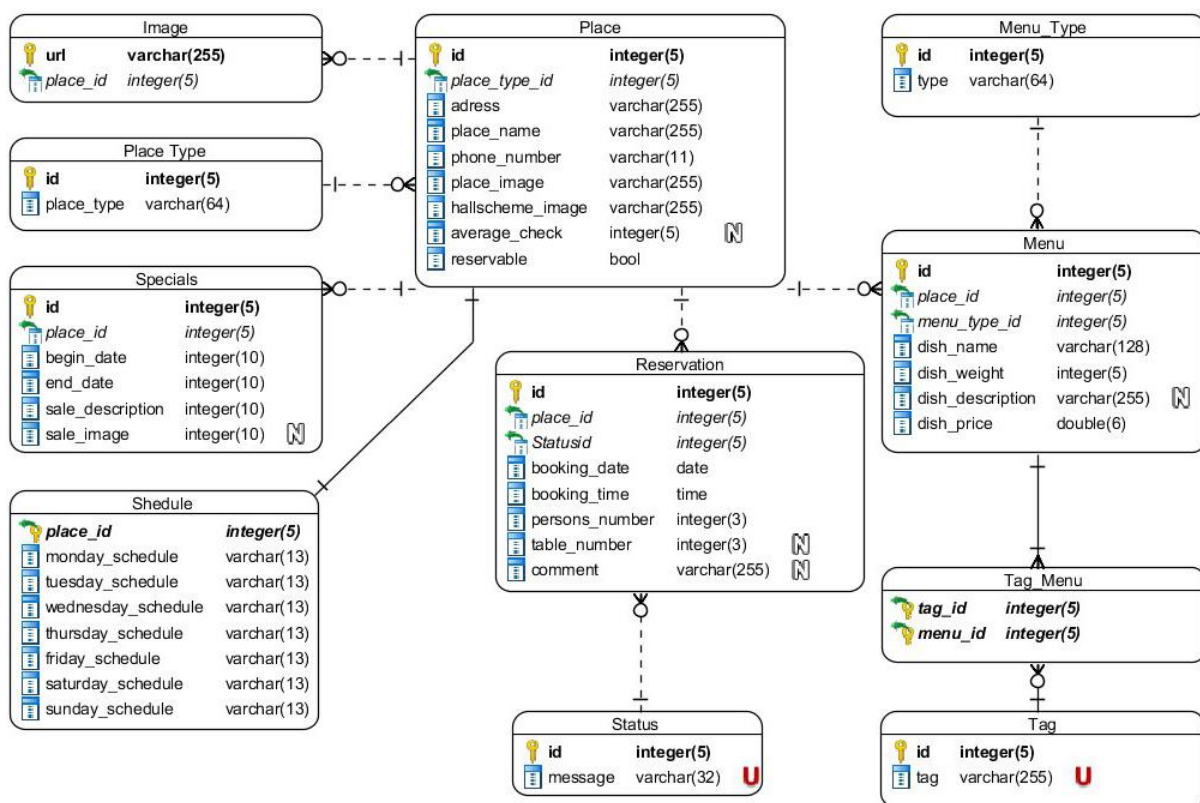


Рис. 7. Схема базы данных приложения

2.5. Взаимодействие с сервером системы

Сервер системы «TableTime» имеет REST-интерфейс, через который происходит обмен данными с приложением. Пример запроса к серверу системы показан на рис. 8.

```

GET /api/menu/
Headers:
JWT: <JSON Web Token >
Response:
[
  {
    "place": 11,
    "id": 73,
    "tags": [
      2
    ],
    "name": "Стейк из говядины с овощами гриль",
    "weight": 450,
    "price": 1100,
    "description": "",
    "category": "Горячие Блюда",
    "category_id": 1
  }
]

```

Рис. 8. Пример запроса на получение информации о заведениях

2.6. Проектирование дизайна приложения

Для приложения было решено использовать строгий и лаконичный дизайн. Цветовая гамма приложения базируется на контрасте темного фона и светлых активных элементов.

Кнопку бронирования было решено выделить другим цветом, поскольку на мобильной версии приложения она могла сливаться с другими элементами на экране. В архиве заявок их статусы также было решено дублировать посредством изменения цвета рамки. Цветовая гамма представлена на рис. 9.

При создании макетов приложения было решено использовать стандартную компоновку элементов, рекомендованную компанией Microsoft [12]. Навигация по приложению осуществляется при помощи навигационного меню, которое находится с левой стороны экрана. Каждый элемент списка акций, заведений и заявок оформлен в виде «карточек».

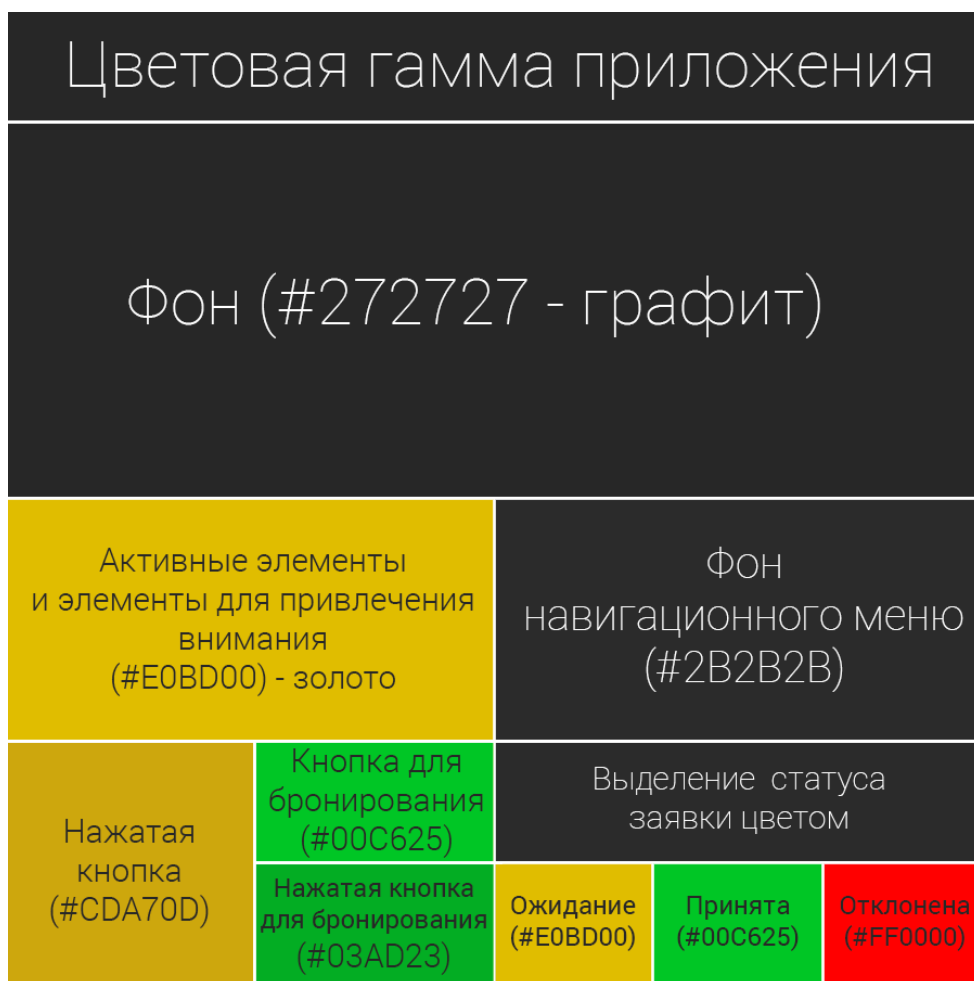


Рис. 9. Цветовая гамма приложения «TableTime»

При нажатии на карточку в списке заведений или списке акций приложение перенаправляет пользователя на экран с подробной информацией о заведении. Макет главного экрана приложения представлен на рис. 10.

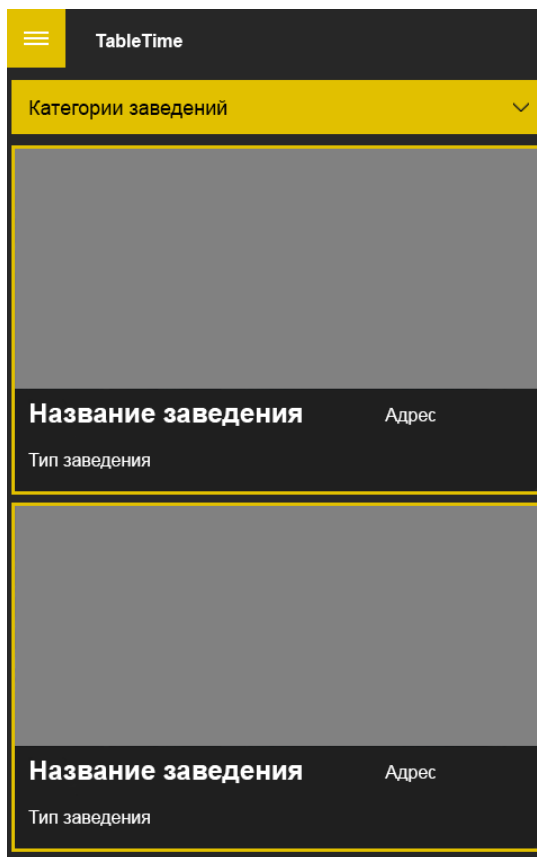


Рис. 10. Макет главного экрана приложения мобильного приложения для Windows 10 Mobile

Версия приложения для персональных компьютеров под управлением Windows 10 отличается от мобильной версии компоновкой элементов внутри приложения. Экран компьютера позволяет сделать приложение большим по ширине, что открывает ряд возможностей. К примеру, название, тип и адрес заведения теперь располагаются не снизу, а справа от превью-фотографии заведения. Это позволяет разместить больше заведений на экране. Аналогично было сделано и с экраном акций. Название, описание акции и срок ее прохождения написаны справа от превью заведения. Выбор категории заведения по той же причине было решено сделать отдельными

кнопками, а не при помощи ComboBox, как это сделано в мобильной версии. Такое решение позволяет ускорить действия пользователя и позволит избежать лишних нажатий мышкой. Изменения также коснулись поискового меню и навигационной панели, которые теперь находятся в постоянной видимости для пользователя. В мобильной версии они сворачивались, экономя множество пространства на экране. На рис. 11 представлен макет главного экрана десктоп-версии приложения.

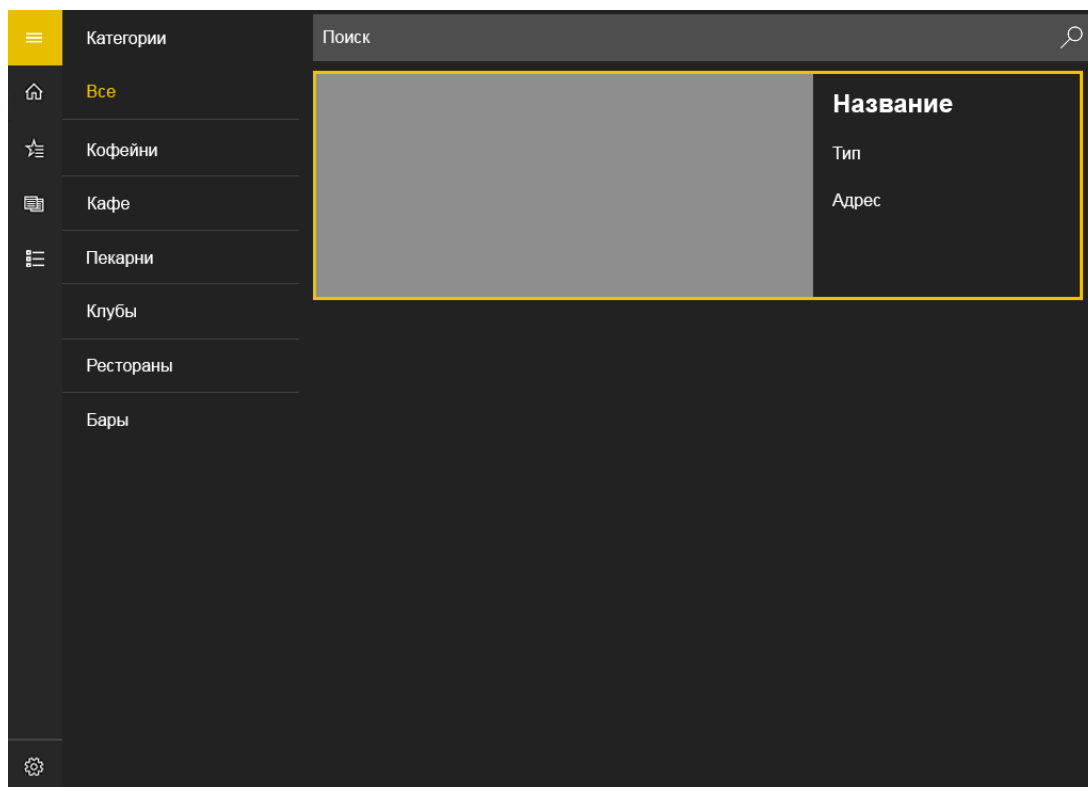


Рис. 11. Макет главного экрана приложения «TableTime» для персональных компьютеров под управлением Windows 10

Кроме того, был спроектирован дизайн и Android-версии приложения. Дизайн был спроектирован с учетом методологии Material Design, разработанной компанией Google.

Навигация по приложению также осуществляется через навигационное меню, расположенное с левой части экрана.

Макет главного экрана Android-версии приложения представлен на рис. 12.

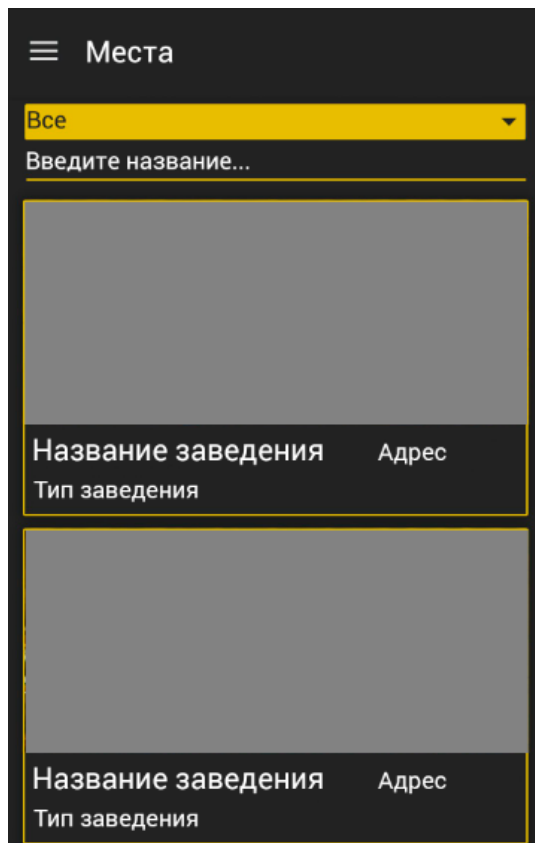


Рис. 12. Макет главного экрана мобильного приложения «TableTime» под управлением операционной системы Android

На логотипе приложения было решено изобразить столик, поскольку бронирование столиков является главной возможностью приложения. Логотип приложения «TableTime» представлен на рис. 13.



Рис. 13. Логотип приложения «TableTime»

Вывод

В соответствии с требованиями были спроектированы архитектура, база данных и дизайн приложения «TableTime» под платформу Windows 10.

3. РЕАЛИЗАЦИЯ СИСТЕМЫ

3.1. Средства реализации

Приложение реализовано для платформы Windows 10 на языке C#. Для хранения данных была реализована база данных с помощью СУБД SQLite [7]. Работа и взаимодействие приложения с сервером реализованы с помощью HTTP-запросов и сообщений Firebase Cloud Messaging.

3.2. Реализация приложения

3.2.1. Реализация пользовательского интерфейса

Дизайн приложения был разработан с учетом нарисованных макетов. На рис. 14 представлен основной экран мобильной версии приложения, на котором отображен список заведений.



Рис. 14. Скриншот основного экрана приложения

Переключение между экранами приложения осуществляется через выезжающее с левой стороны экрана меню, реализованное через паттерн Navigation View [9]. В левом верхнем углу экрана находится кнопка, позволяющая открывать и закрывать панель навигации. Данный вариант навигационной панели называется «Наложение». Мобильная версия навигационной панели представлена на рис. 15.

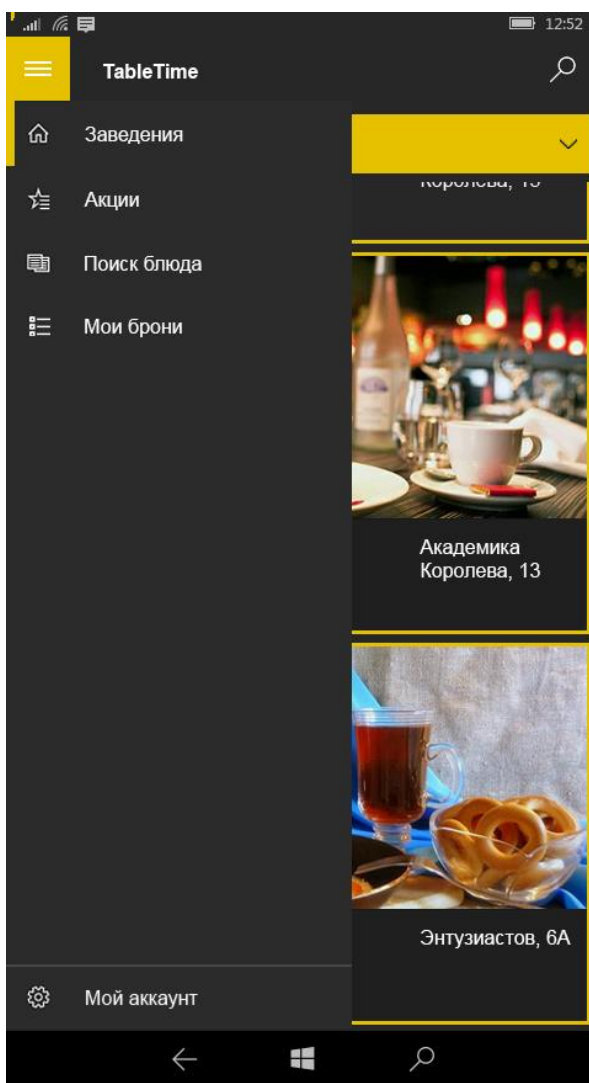


Рис. 15. Навигационное меню мобильного приложения

Для десктоп-версии приложения был выбран «Компактный» вариант навигационной панели. Разница с мобильной версией заключается в том, что навигационное меню постоянно находится с левой стороны, а по нажатию на кнопку-«гамбургер» оно разворачивается, накладываясь на содержимое. Десктоп-версия приложения показана на рис. 16 и 17.

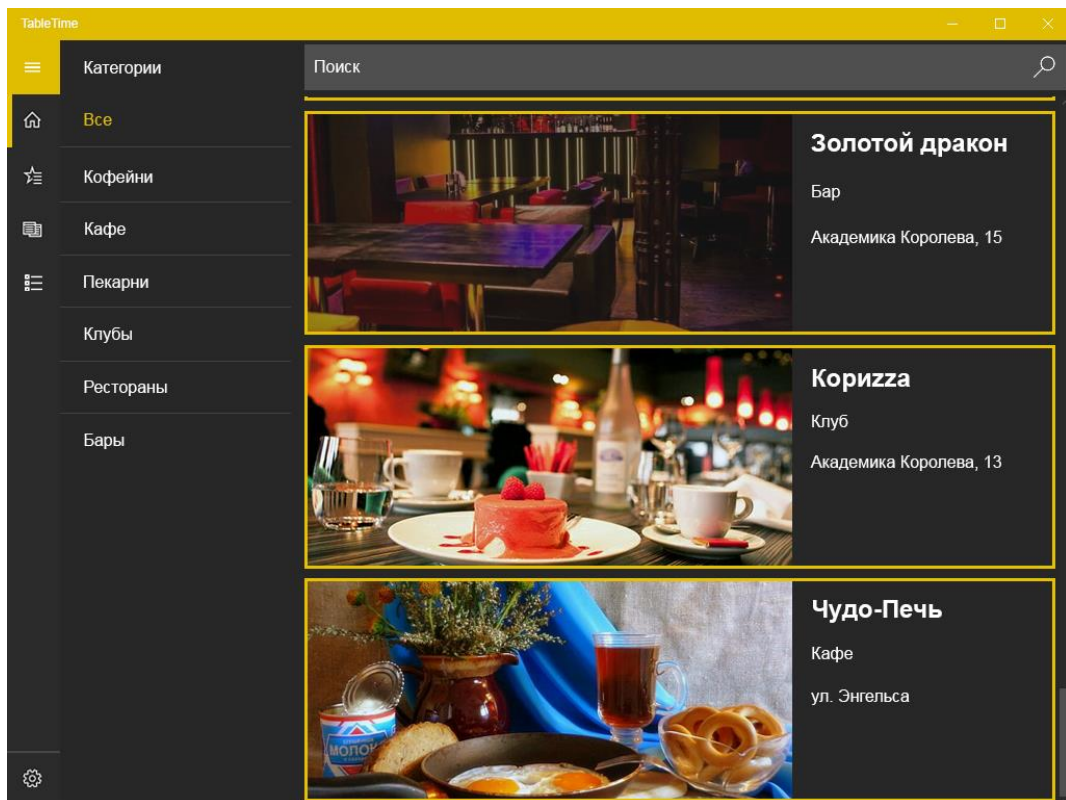


Рис. 16. Навигационное меню десктоп-версии приложения в свернутом положении

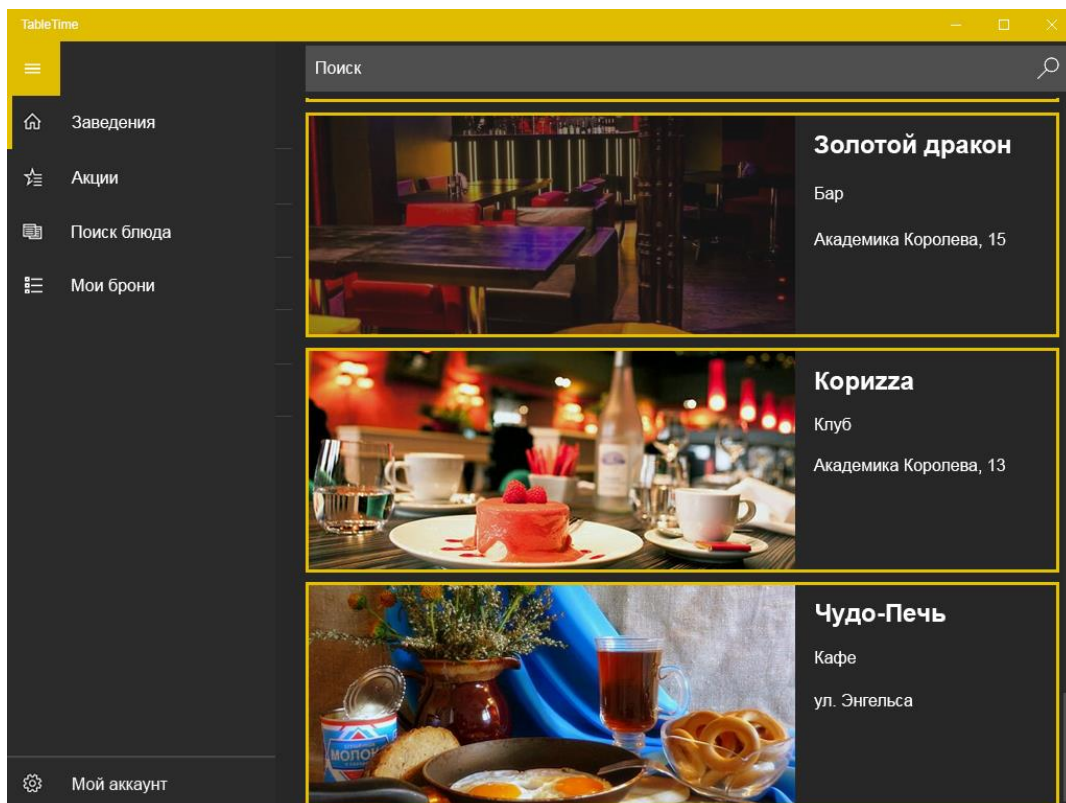


Рис. 17. Навигационное меню десктоп-версии приложения в развернутом положении

Существует два варианта компактного режима навигационной панели: `CompactOverlay` и `CompactInline`. В первом варианте панель навигации накладывается на содержимое, во втором варианте содержимое отодвигается в правую сторону. Было решено использовать `CompactOverlay`, поскольку компания Microsoft рекомендует использовать данный вариант.

Вариант навигационной панели выбирается в зависимости от ширины окна приложения. Для этого используется класс `VisualState`, который представляет внешний вид элемента управления, находящегося в определенном состоянии. Вариант «Наложение» используется при ширине окна не более 719 пикселей, в то время как вариант «Компактный» используется при ширине окна от 720 пикселя и выше. Для мобильного семейства устройств используется только вариант «Наложение». Выбор варианта навигационной панели представлен на рис. 18.

```
<VisualStateGroup>
  <VisualState>
    <VisualState.StateTriggers>
      <AdaptiveTrigger
        MinWindowWidth="720" />
    </VisualState.StateTriggers>
    <VisualState.Setters>
      <Setter
        Target="NavigationPane.DisplayMode"
        Value="CompactOverlay" />
    </VisualState.Setters>
  </VisualState>
  <VisualState>
    <VisualState.StateTriggers>
      <AdaptiveTrigger MinWindowWidth="0" />
    </VisualState.StateTriggers>
    <VisualState.Setters>
      <Setter
        Target="NavigationPane.DisplayMode" Value="Overlay" />
    </VisualState.Setters>
  </VisualState>
  <VisualState>
    <VisualState.StateTriggers>
      <triggers:DeviceFamilyStateTrigger DeviceFamily="Mobile" />
    </VisualState.StateTriggers>
    <VisualState.Setters>
      <Setter
        Target="NavigationPane.DisplayMode"
        Value="Overlay" />
    </VisualState.Setters>
  </VisualState>
</VisualStateGroup>
```

Рис. 18. Листинг выбора варианта отображения навигационной панели

Стоит также отметить кастомизацию элемента TitleBar. TitleBar – панель, находящаяся в верхней части диалогового окна приложения, на которой обычно располагаются кнопки закрытия, свертывания и развертывания приложения. По умолчанию она принимает цвет темы операционной системы Windows 10. Для изменения цвета требуется написать свой метод и вызвать его в конструкторе главной страницы приложения MainPage. Листинг метода представлен на рис. 19.

```
private void SetTitleBarBackground()
{
    var titleBar = ApplicationView.GetForCurrentView().TitleBar;

    titleBar.BackgroundColor = Color.FromArgb(1, 224, 189, 0);
    titleBar.ForegroundColor = Colors.White;

    titleBar.ButtonBackgroundColor = Color.FromArgb(1, 224, 189, 0);
    titleBar.ButtonForegroundColor = Colors.White;
}
```

Рис. 19. Листинг выбора варианта отображения навигационной панели

Для компоновки элементов на экранах приложения был использован элемент управления Grid. Grid – контейнер компоновки, использующийся для разбиения окна на меньшие области, которыми затем можно управлять с помощью других элементов. Создание компоновки на основе Grid является двухшаговым процессом: сперва задается необходимое количество строк и столбцов, а затем каждому располагаемому на Grid элементу указывается строка и столбец.

3.2.2. Работа с сообщениями Firebase Cloud Messaging

В системе «TableTime» сообщения Firebase Cloud Messaging используются для передачи информации о заявках. При создании новой заявки или отмене уже существующей заявки пользователем приложение отправляет на сервер сообщение, содержащее идентификатор заявки и новый код статуса заявки.

Если статус заявки изменился на сервере, то сервер отправляет FCM-сообщение на приложение пользователя.

В приложении предусмотрена возможность принудительного обновления статуса заявок через HTTP-запросы на случай, если в сервисе FCM или в системе «TableTime» произошел сбой.

Для реализации сервиса приема сообщений была использована библиотека FireSharp [11].

Вывод

На основе спроектированной архитектуры было реализовано приложение «TableTime» для платформы Windows 10.

4. ТЕСТИРОВАНИЕ

Для тестирования приложения использовались следующие методы тестирования:

- 1) функциональное тестирование;
- 2) проверка эргономичности;
- 3) тестирование интерфейса пользователя.

Функциональное тестирование – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определенных условиях решать задачи, нужные пользователям [2]. Функциональные требования определяют, что именно делает программное обеспечение, какие задачи оно решает. Набор тестов на функциональность представлен в табл. 1.

Табл. 1. Протокол тестирования приложения.

№	Аспект работы	Действия	Результат	Тест пройден?
1	Изменение персональных данных	<ol style="list-style-type: none">1. В приложении в боковом меню выбрать вкладку «Мой аккаунт»;2. Выбрать нужное поле с информацией и внести изменения;3. Нажать кнопку «Сохранить изменения».	Изменения персональных данных будут сохранены.	Да
2	Регистрация	<ol style="list-style-type: none">1. На экране регистрации ввести имя, номер телефона и адрес электронной почты;2. Нажать на кнопку «Зарегистрироваться»;3. Ввести код, полученный через СМС-сообщение.	Приложение перенаправляет пользователя на главный экран приложения.	Да

№	Аспект работы	Действия	Результат	Тест пройден?
3	Авторизация через номер мобильного телефона	<ol style="list-style-type: none"> 1. На экране авторизации ввести номер телефона; 2. Нажать кнопку «Войти»; 3. Ввести код, полученный через СМС-сообщение. 	Приложение перенаправляет пользователя на главный экран приложения.	Да
4	Просмотр активных заявок на бронирование столиков.	<ol style="list-style-type: none"> 1. В боковом меню выбрать вкладку «Мои брони». 	На экране отобразится список активных заявок на бронирование, которые ожидают либо подтверждения, либо завершения.	Да
5	Авторизация через адрес электронной почты	<ol style="list-style-type: none"> 1. а экране авторизации нажать кнопку «Войти через E-Mail»; 2. Ввести свой адрес электронной почты; 3. Нажать кнопку «Войти»; 4. Ввести код, полученный через электронную почту. 	Приложение перенаправляет пользователя на главный экран приложения.	Да
6	Просмотр архива заявок на бронирование	<ol style="list-style-type: none"> 1. В боковом меню выбрать вкладку «Мои брони»; 2. Выбрать вкладку «Архив». 	На экране появится список завершенных заявок.	Да
7	Поиск блюда	<ol style="list-style-type: none"> 1. В боковом меню выбрать вкладку «Поиск блюда»; 2. В поле поиска ввести название интересующего блюда. 	На экране отобразятся блюда и места, в которых они присутствуют. Если такого блюда нет в списках блюд среди всех заведений, появится надпись «Блюдо не найдено».	Да

№	Аспект работы	Действия	Результат	Тест пройден?
8	Выбор категории в списке заведений в мобильной версии приложения	1. В боковом меню открыть вкладку «Заведения»; 2. В разворачивающемся списке выбрать нужную категорию.	На экране появятся заведения нужной категории.	Да
9	Выбор категории заведения в десктоп-версии приложения	1. На вкладке «Заведения» нажать на кнопку с нужной категорией.	На экране появятся заведения нужной категории.	Да
10	Поиск заведения	1. В боковом меню открыть вкладку «Заведения»; 2. В поле поиска ввести название интересующего заведения.	На экране отобразится карточка с нужным заведением. Если в списке мест заведения с таким названием не оказалось, на экране появится надпись «Заведение не найдено».	Да
11	Просмотр списка акций	1. В боковом меню открыть вкладку «Акции»	На экране появится список акций, которые проходят на текущий момент во всех заведениях.	Да
12	Просмотр подробной информации о заведении	1. Выбрать заведение из экрана «Заведения», либо из экрана «Акции» или «Поиск блюда».	Приложение откроет экран подробной информации с описанием данного заведения.	Да

№	Аспект работы	Действия	Результат	Тест пройден?
13	Просмотр меню заведения	1. Из просмотра информации о заведении перейти во вкладку «Меню».	На экране появится меню заведения.	Да
14	Подать заявку на бронирование столика	1. Из просмотра информации о заведении нажать на кнопку «Забронировать»; 2. Выбрать дату, на которую будет осуществляться бронь; 3. Выбрать время; 4. Указать количество персон; 5. Номер столика и комментарий к брони указываются по желанию пользователя.	Во вкладке «Мои брони» появится активная заявка, ожидающая мацерации со стороны менеджера заведения.	Да
15	Отменить заявку на бронирование	1. В боковом меню выбрать вкладку «Мои брони»; 2. В списке текущих заявок на бронирование нажать кнопку «Отменить бронь».	Заявка исчезнет из списка активных заявок и появится в списке завершенных с подписью «Отменена».	Да
16	Отправка пустой заявки на бронирование	1. При создании заявки не заполнить одно или несколько обязательных полей.	Приложение выдаст сообщение об ошибке.	Да

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы бакалавра мною было реализовано мобильное клиентское приложение для сервиса «TableTime» на платформе Windows 10.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ

1. Был выполнен анализ предметной области и произведен обзор существующих решений.
2. Изучены методы разработки приложений Universal Windows Platform для платформы Windows 10.
3. Спроектирована архитектура приложения.
4. Реализовано приложение.
5. Проведено тестирование приложения.

НАПРАВЛЕНИЯ ДАЛЬНЕЙШИХ ИССЛЕДОВАНИЙ

Дальнейшие исследования и практические разработки могут быть направлены на улучшенный фильтр поиска блюд и выбор столика при бронировании через специальную карту столиков заведения.

ЛИТЕРАТУРА

1. Baidachniy S. Developing Windows 10 Application with C#. – North Charleston: CreateSpace Independent Platform, SC, 2016. – 520 p.
2. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. – СПб.: Питер, 2004. – 320 с.
3. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя. – СПб.: Издательство «Питер», 2003. – 432 с.
4. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. – М.: Вильямс, 2017. – 1440 с.
5. Мак-Дональд М. WPF: Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0 для профессионалов. 4-е изд. – М.: Вильямс, 2013. – 1024 с.
6. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. 6-е изд. – М.: Вильямс, 2015. – 1312 с.
7. Creating SQLite database in Windows 10. [Электронный ресурс] URL: <https://channel9.msdn.com/Series/A-Developers-Guide-to-Windows-10/10> (дата обращения: 23.02.2017).
8. Firebase Cloud Messaging. [Электронный ресурс] URL: <https://firebase.google.com/docs/cloud-messaging/> (дата обращения: 24.02.2017).
9. Navigation View Pattern. [Электронный ресурс] URL: <https://docs.microsoft.com/en-us/windows/uwp/controls-and-patterns/navigationview> (дата обращения: 25.03.2017).
10. Афиша–Рестораны. [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=ru.afisha.restaurants> (дата обращения: 14.02.2017).
11. Библиотека FireSharp. [Электронный ресурс] URL: <https://github.com/ziyasal/FireSharp/> (дата обращения: 25.03.2017).

12. Введение в проектирование приложений UWP [Электронный ресурс] URL: <https://docs.microsoft.com/ru-ru/windows/uwp/controls-and-patterns/controls-and-events-intro> (дата обращения: 24.02.2017).

13. РИА Новости – Крупнейшие сотовые операторы за год удвоили покрытие LTE – сетей. [Электронный ресурс] URL: <https://ria.ru/economy/20141209/1037401036.html> (дата обращения: 25.02.2017).

14. Столики. [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=com.saas.app> (дата обращения: 14.02.2017).

15. ТоМесто – лучшие рестораны. [Электронный ресурс] URL: <https://play.google.com/store/apps/details?id=ru.tomesto.tomesto> (дата обращения: 14.02.2017).

ПРИЛОЖЕНИЕ

Графический интерфейс приложения

Отображение интерфейса на эмуляторе Windows 10 Mobile, WXGA 4.5".



Рис. 1. Главный экран приложения с заведениями

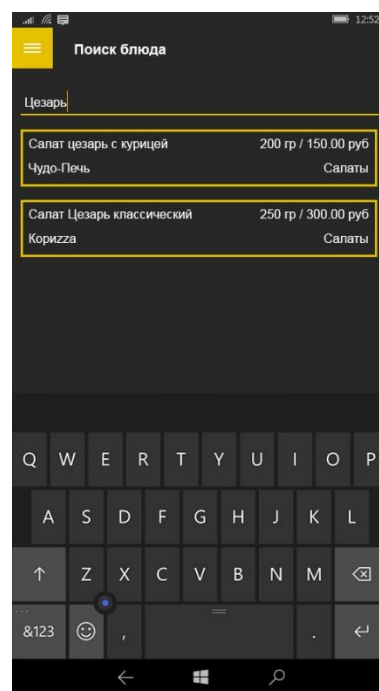


Рис. 2. Экран приложения с поиском блюд



Рис. 3. Экран приложения с акциями заведений

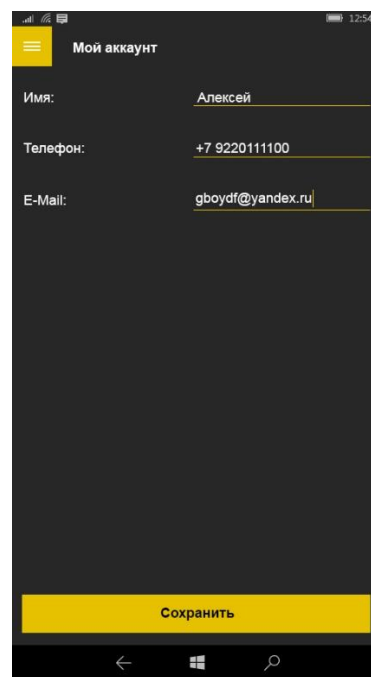


Рис. 4. Экран приложения с настройками аккаунта

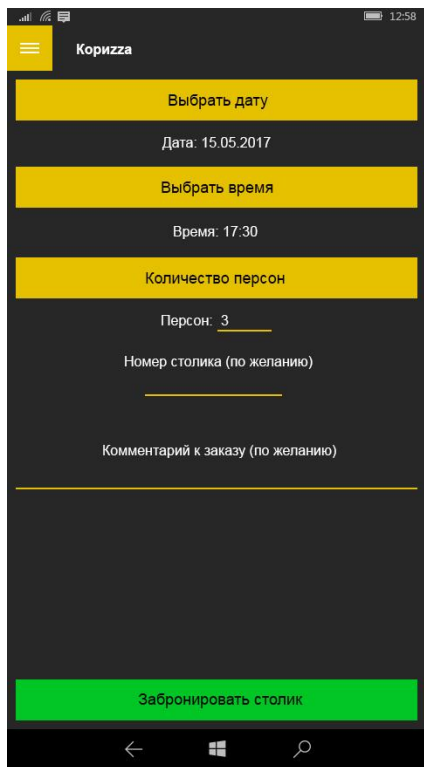


Рис. 5. Экран приложения с бронированием столика

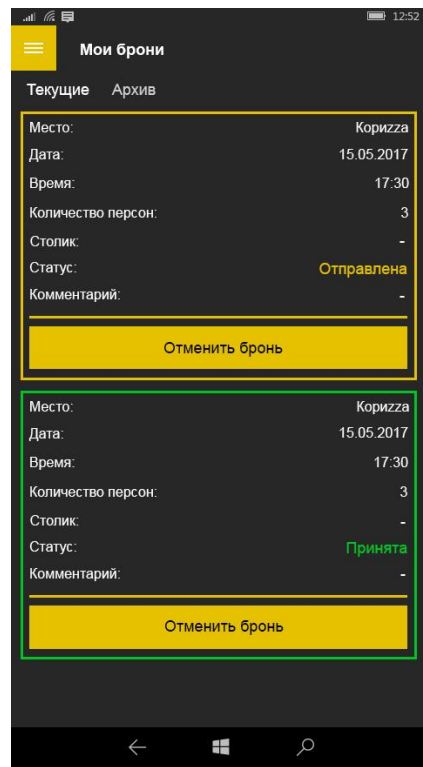


Рис. 6. Экран приложения с текущими заявками на бронирование

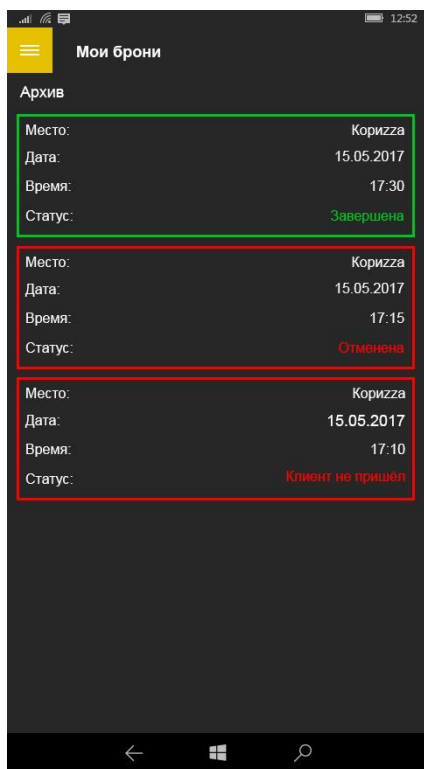


Рис. 7. Экран приложения с архивом заявок на бронирование

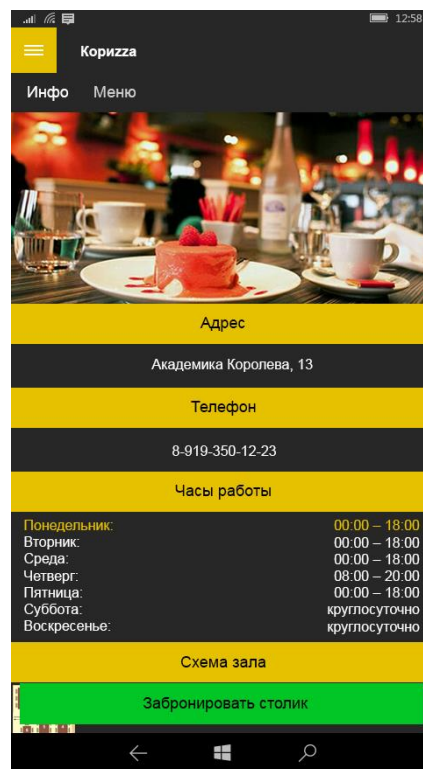


Рис. 8. Экран приложения с подробной информацией о заведении

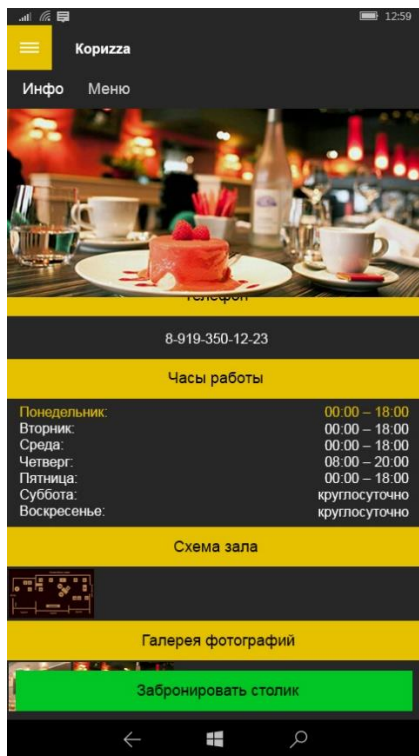


Рис. 9. Экран приложения с подробной информацией о заведении

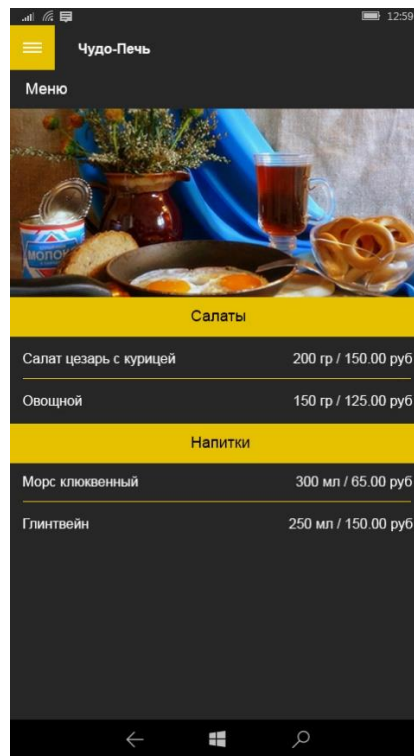


Рис. 10. Экран приложения с меню заведения

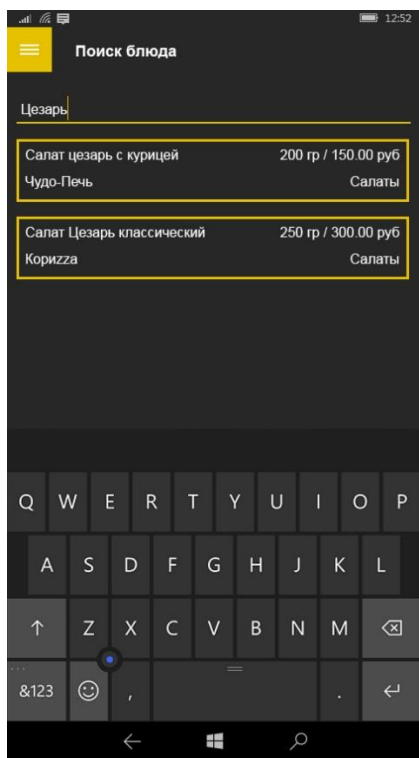


Рис. 11. Экран приложения с поиском блюда

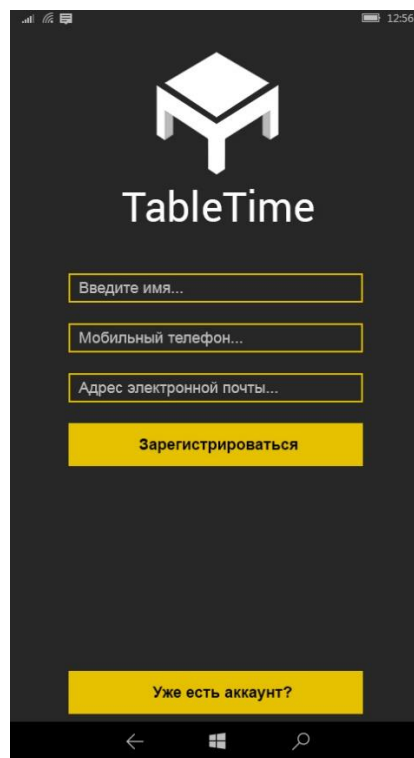


Рис. 12. Экран приложения с регистрацией

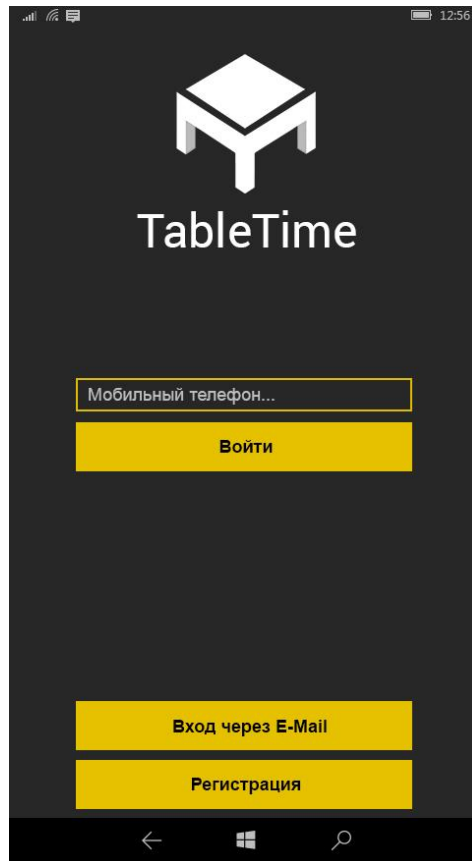


Рис. 11. Экран приложения с авторизацией

Отображение интерфейса на Windows 10 Creators Update.

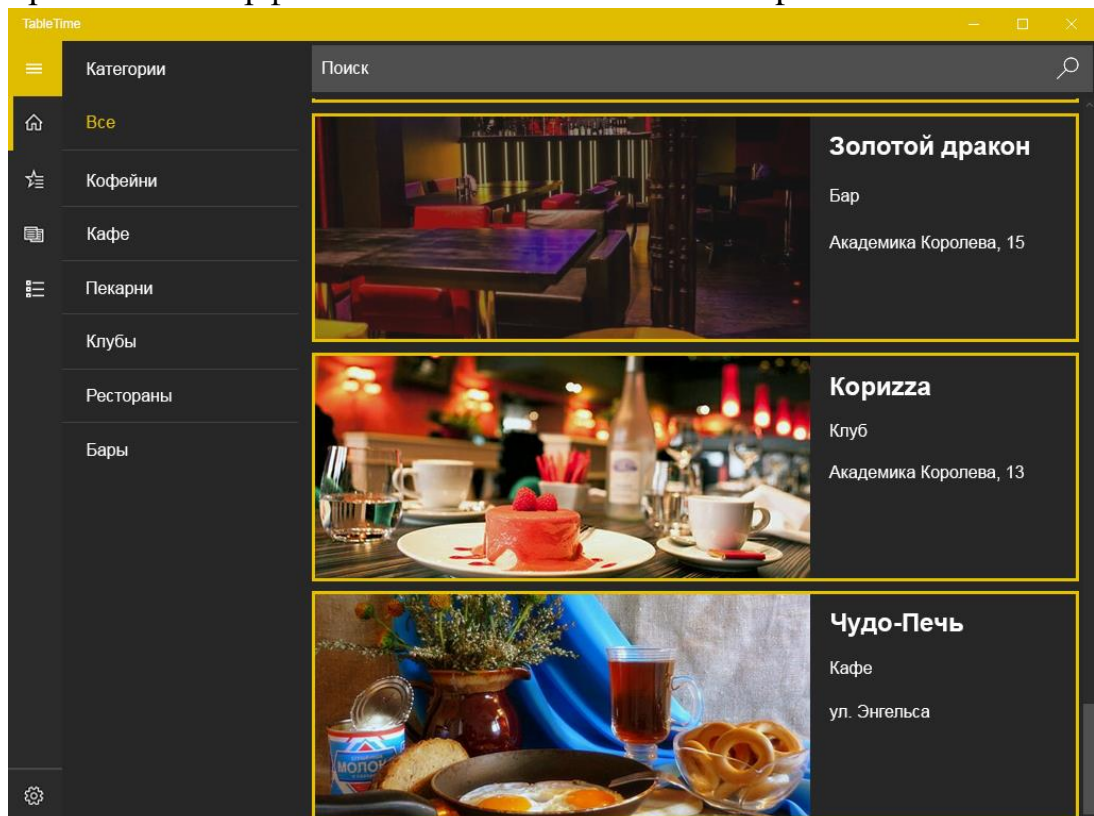


Рис. 12. Экран с заведениями десктоп-версии приложения

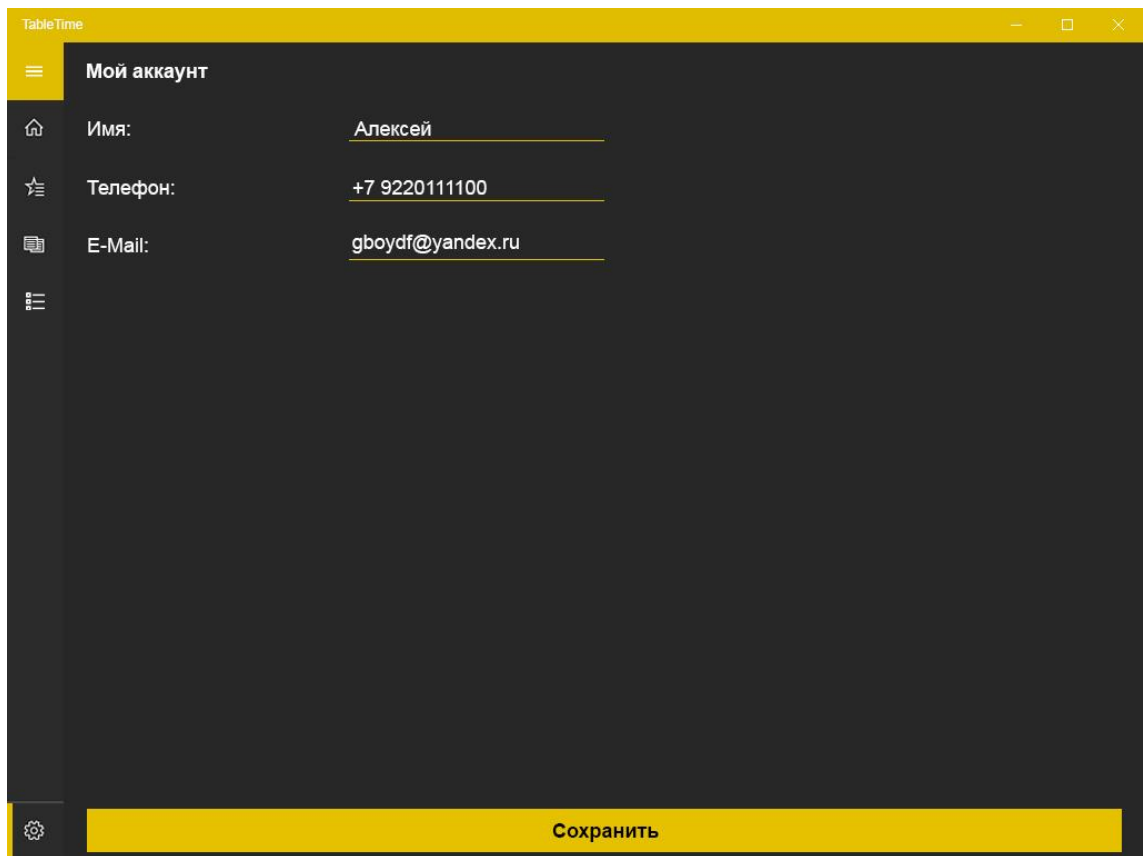


Рис. 13. Экран с персональными настройками десктоп-версии приложения

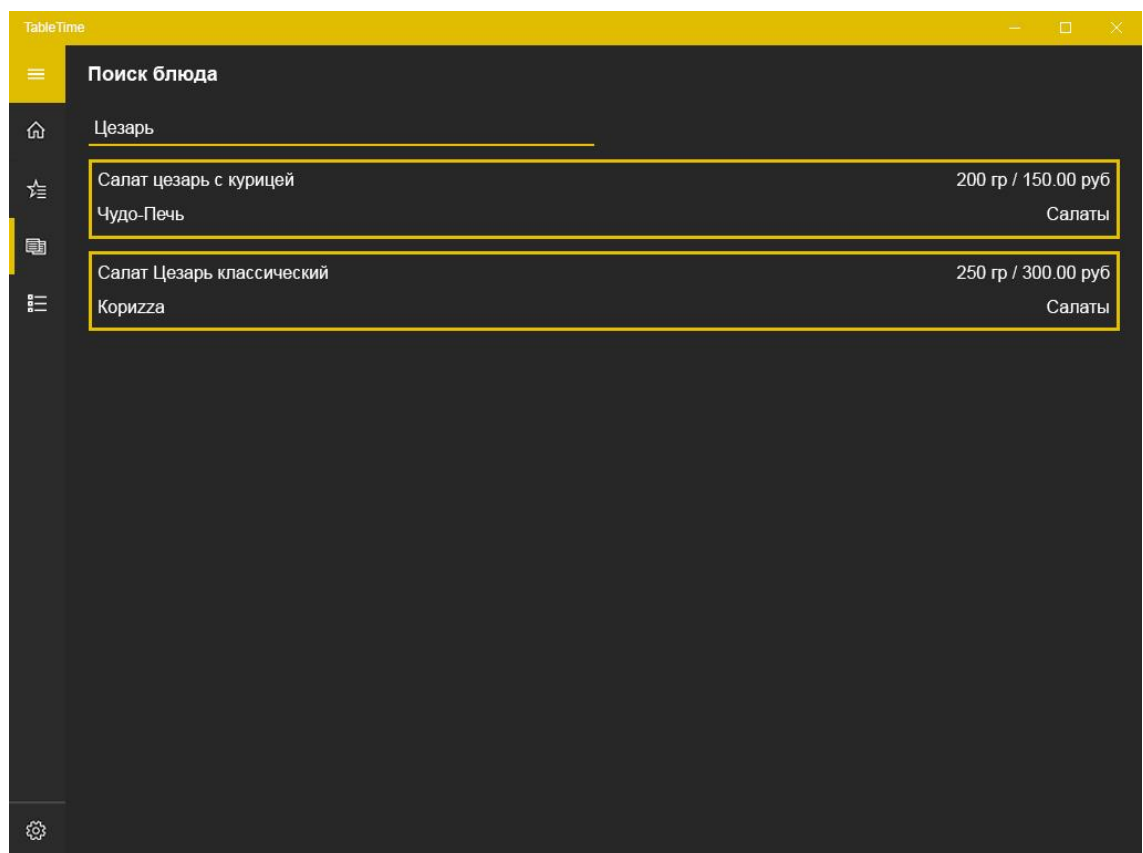


Рис. 14. Экран с поиском блюд десктоп-версии приложения

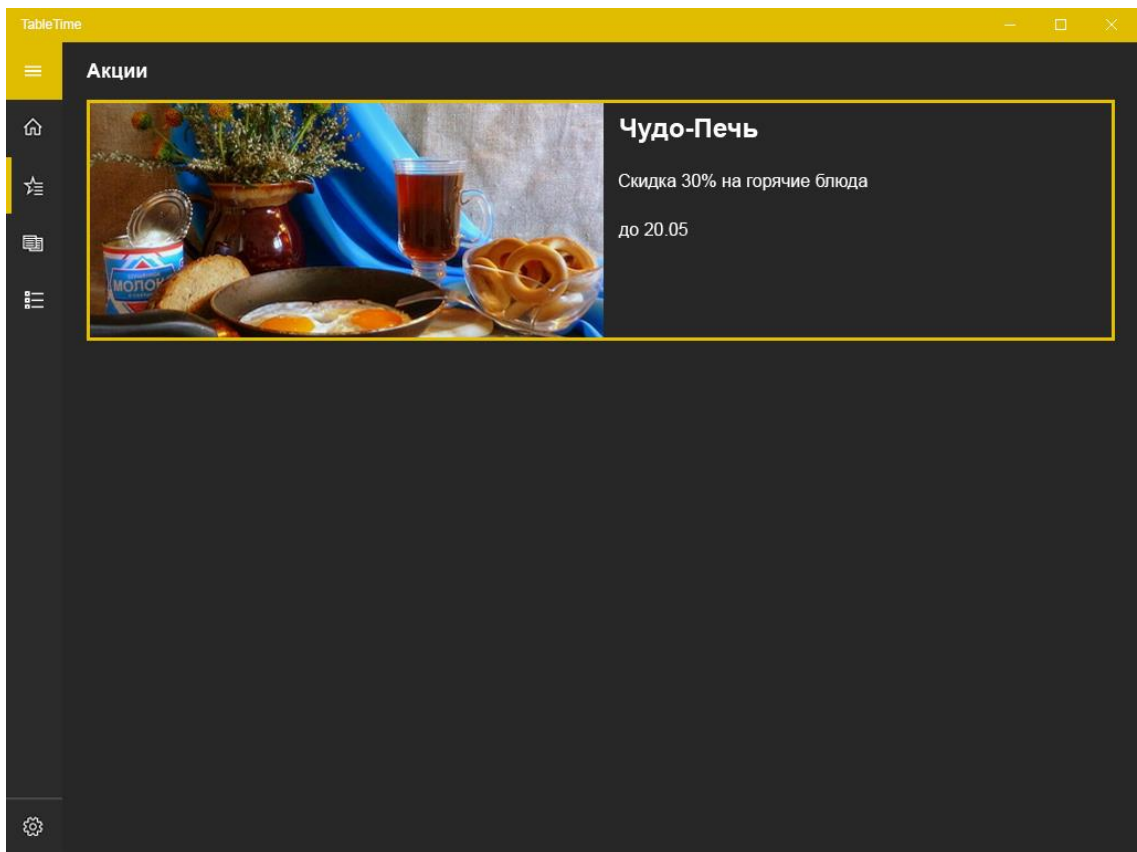


Рис. 15. Экран с акциями заведений десктоп-версии приложения

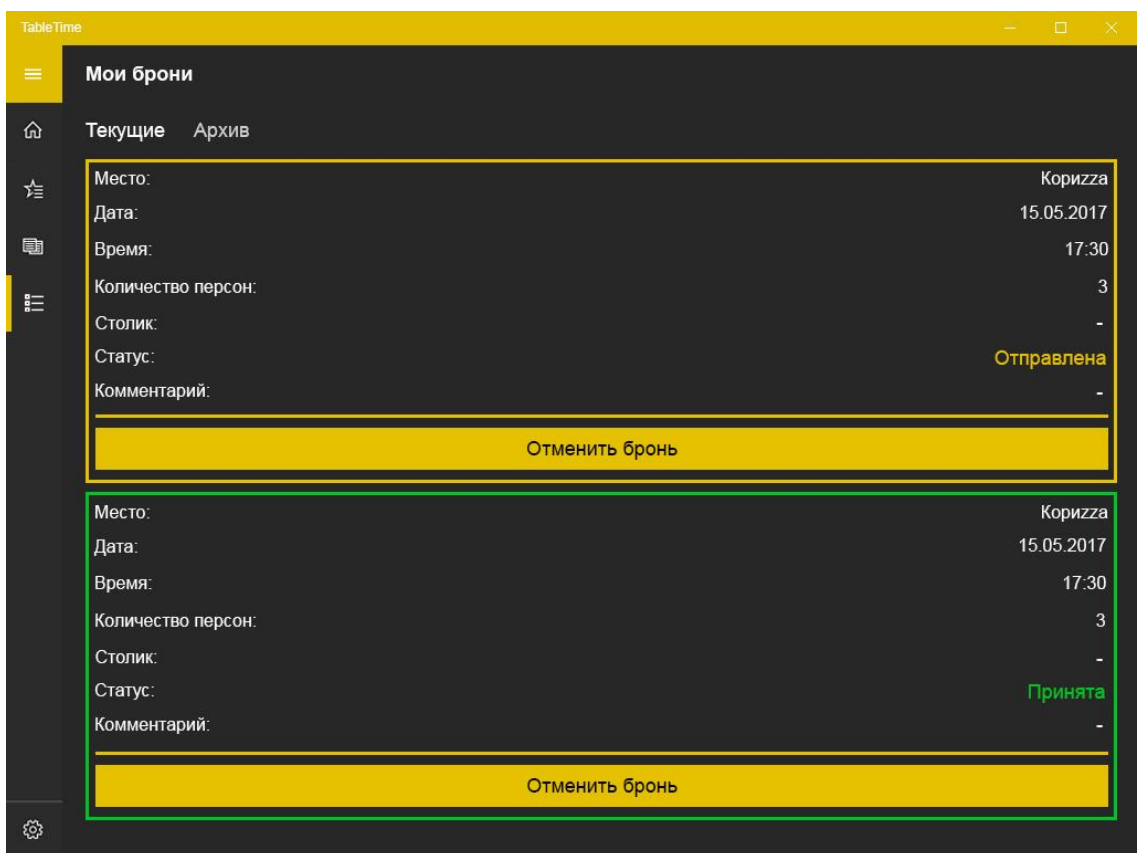


Рис. 16. Экран десктоп-версии приложения с текущими заявками на бронирование

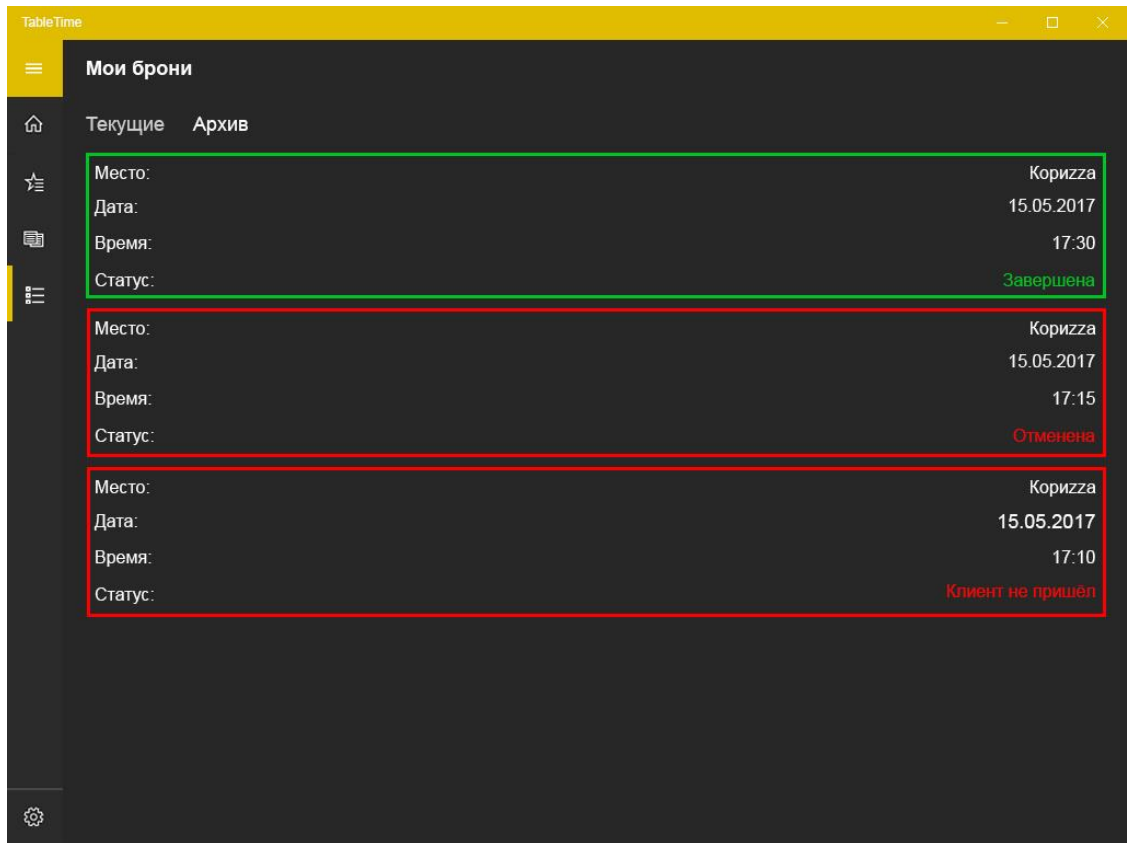


Рис. 17. Экран десктоп-версии приложения с архивом заявок на бронирование

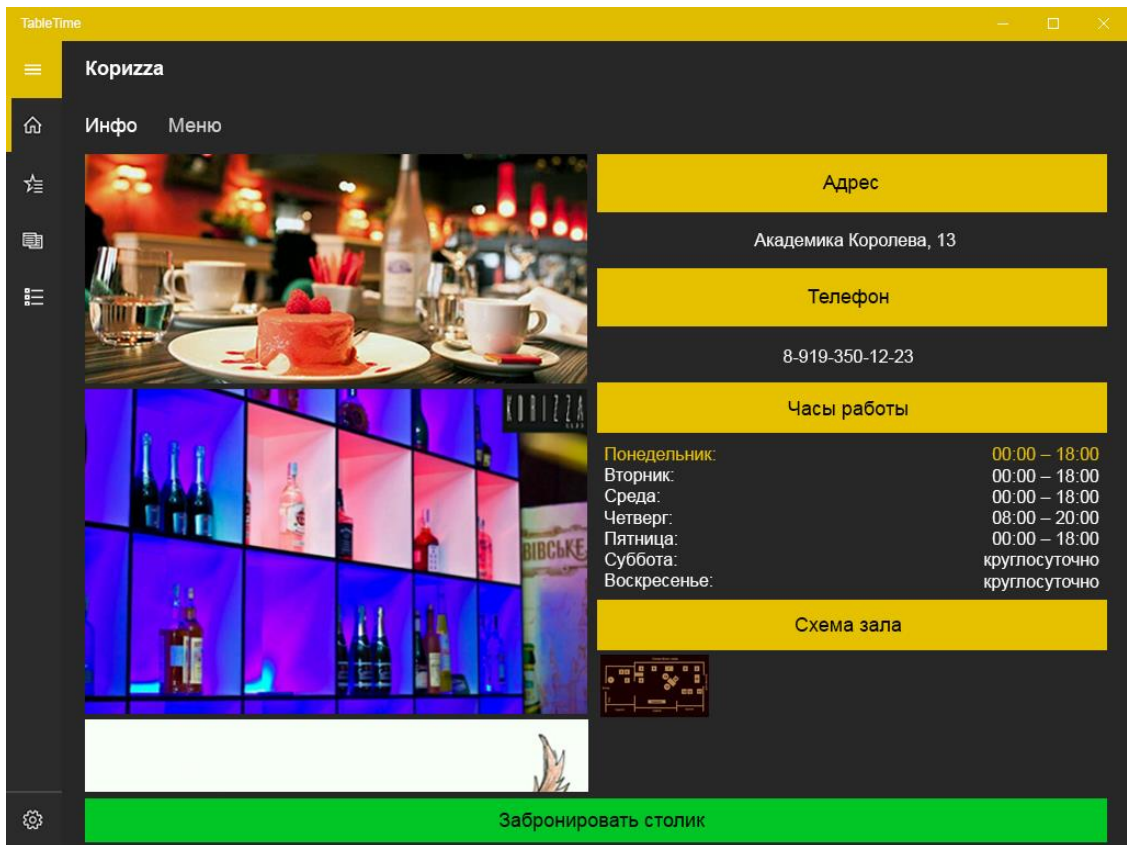


Рис. 18. Экран с подробной информацией о заведении

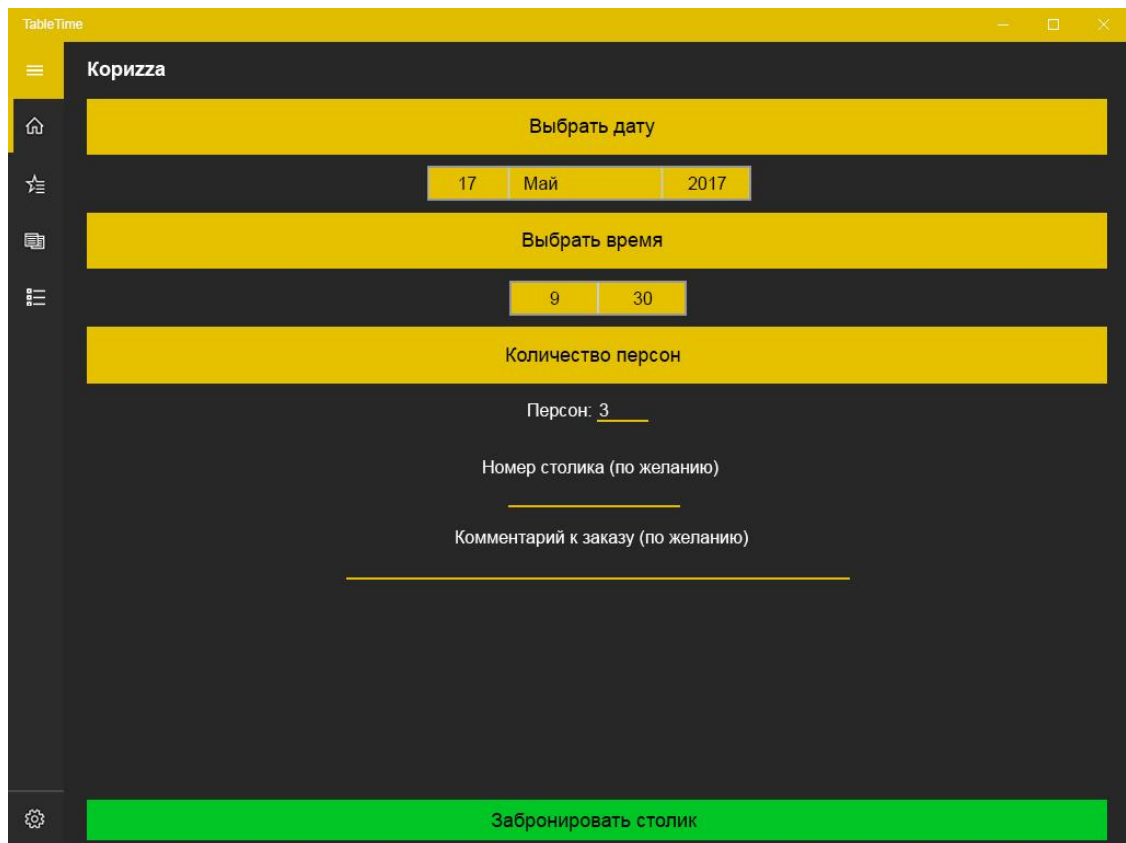


Рис. 19. Экран десктоп-версии приложения с бронированием столика