Научно-исследовательская часть

1. Постановка задачи проектирования.

Задачей проектирования данной курсовой работы является разработка веб-приложения для сети автомоек, предназначенного для записи. Так же необходимо разработать удобный пользовательский интерфейс для работы с приложением.

1. Описание предметной области.

Автомойка – организация, осуществляющая мойку автомобилей и оказывающая сопутствующие услуги.

В городе Х есть сеть автомоек (5-7 шт) и в этом городе постоянные очереди на эти автомойки. Для решения данной проблемы руководство принимает решение о создании сайта, в котором будет реализован функционал:

* Регистрация, авторизация;
* Выбор филиала;
* Выбор времени;
* Выбор услуг (мойка ковриков, химчистка сидений);
* Просмотр истории моек;
* 2 типа учётных записей: Администратор (просмотр статистики, добавление автомоек, добавление услуг и тд), Клиент (выбор времени, места, услуг);
* Система скидок, которые будут автоматически рассчитываться исходя из общей стоимости, оказанных услуг для каждого пользователя.

Необходим именно сайт т.к. его разработка дешевле и быстрее, чем написание кроссплатформенных приложений. Также любое приложение необходимо установить и скачать, а сайт – нет. Также сайт будет заодно и рекламировать сеть автомоек.

Отслеживать очередь в режиме реального времени нет необходимости, т.к. при нажатии кнопки записаться будет видно ближайшее время.

У каждой записи в автомойку есть: только один пользователь, совершивший эту запись; только один филиал автомойки, выбранный пользователем; дата и время; одна или множество услуг, выбранных пользователем; стоимость услуг; длительность услуг. У каждого филиала автомойки есть свой адрес. У каждой услуги есть название, её цена и время оказания. У филиала может временно не быть услуг, также и у услуги может временно не быть филиалов, где её оказывают. Каждый пользователь имеет скидку, на основании суммы его заказов.

1. Анализ аналогов и прототипов.

У некоторых автомоек уже существуют сайты, на которых пользователю предлагают записаться на обслуживание, разберём их плюсы и минусы:

1. «formula76»

Ссылка на сайт: <https://formula76.ru/tireservice/washrecord>

Сеть автосервисов «Формула» — многофункциональный центр по обслуживанию автомобилей в Ярославле. У них есть 3 филиала, где можно вымыть автомобиль, выполнить ремонтные работы, купить шины, диски и АКБ.

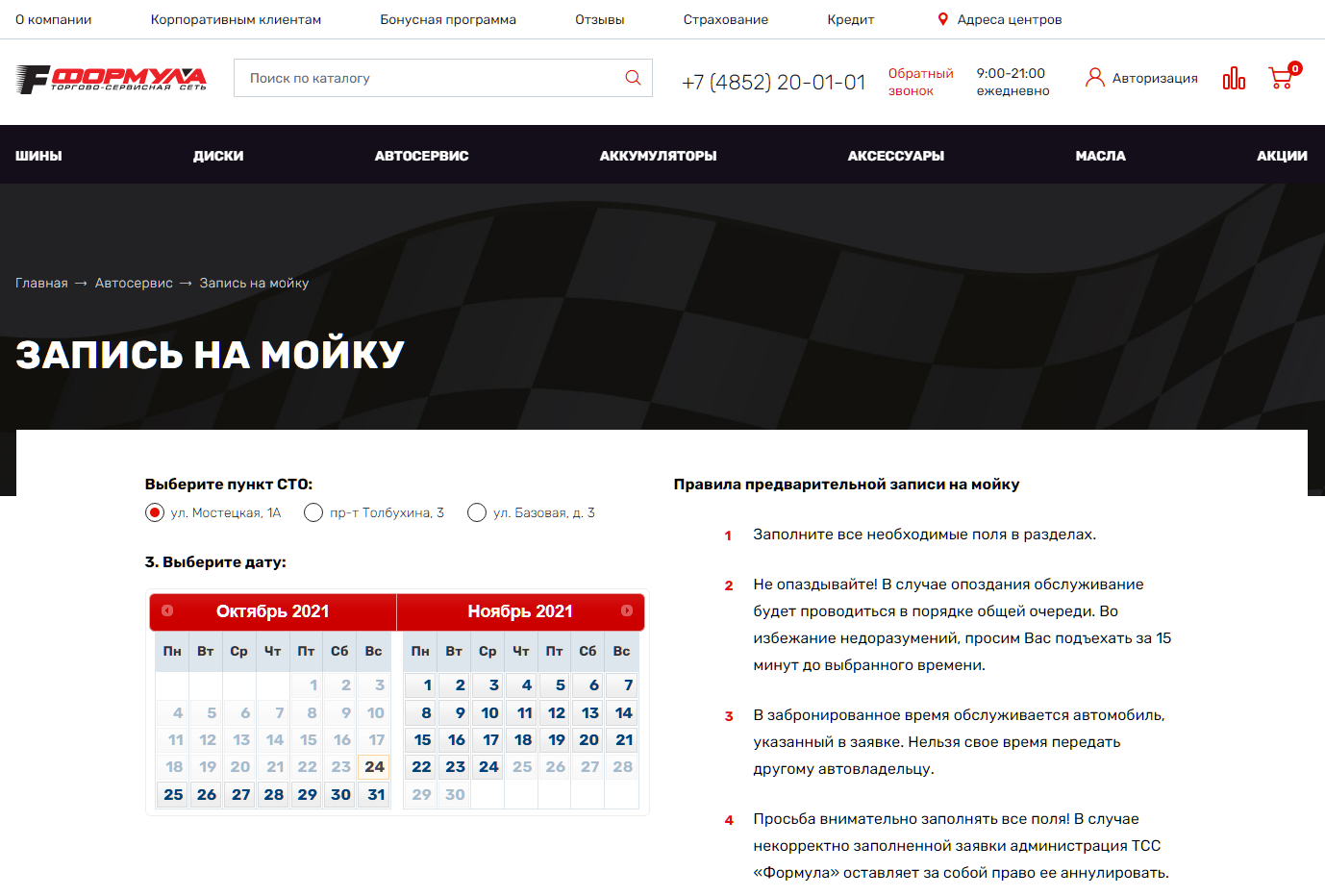


Рисунок 1. Веб-сайт «formula76».

Плюсы:

* Возможность записаться на автомойку без регистрации и авторизации на сайте;
* Удобный выбор даты.

Минусы:

* Часовые промежутки во времени, что крайне неудобно в текущих реалиях мира, когда у некоторых людей день расписан буквально по минутам.
* Нет списка оказываемых услуг, и тем более их стоимости и времени выполнения. Соответственно человек, который зашёл на этот сайт не имеет возможности узнать, выполняется ли интересующая его услуга на данной автомойке или нет. Также любой человек, прежде чем записаться, хочет узнать цену, которую ему придётся заплатить.

1. Автотехцентр «Гараж»

Ссылка на сайт: <http://atc-garage.ru>

«Гараж» – это сеть многофункциональных технических центров премиум класса, оказывающих полный спектр услуг по первоклассному обслуживанию автомобилей. Этот автотехцентр является лидером на рынке авторемонтных, автомоечных и автокосметических услуг Калужской области.

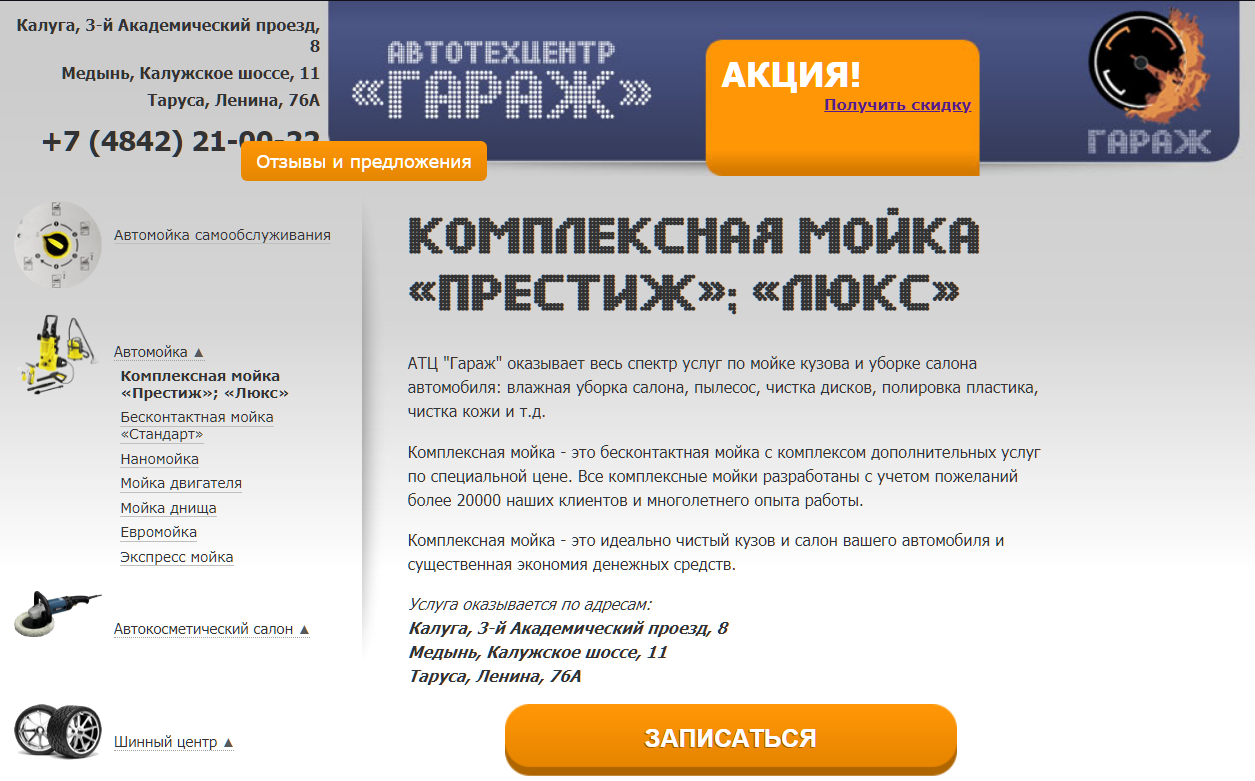


Рисунок 2. Веб-сайт «atc-garage».

Плюсы:

* Возможность записаться на автомойку без регистрации и авторизации на сайте

Минусы:

* Получасовые интервалы, это уже лучше, чем в прошлом примере, но всё же недостаточно удобно
* При записи можно выбрать только один тип мойки
* Как и в прошлом примере неизвестны цена и длительность оказания услуги

1. «RocketWash»

Ссылка на сайт: <https://www.rocketwash.me/avtolyubitelyam/mobile-app>

Фактически, это не автомойка, а CRM-система для автомоек, которая призвана автоматизировать все процессы. Но т.к. этот сайт выдаётся поисковиком по запросу «Записаться на автомойку», мы всё равно его рассмотрим. Конкретно на сайте, нет возможности записаться на автомойку, есть только ссылки на приложения в AppStore и GooglePlay. Это не совсем подходит для темы курсовой работы, но всё же для исследования предметной области, скачаем приложение.

При запуске приложения мы видим автомойки подключенные, к данной CRM-системе (Рисунок 3). При выборе автомойки мы видим ближайшее свободное время и примерные цены на услуги (Рисунок 4). Нажмём кнопку «Записаться», приложение требует указать марку и модель авто, именно от этого будут зависеть точные цены на услуги (полезная фишка, но в нашем случае – не обязательная). Далее мы видим список услуг с указанием их цены, и при нажатии на значок вопроса появляется дополнительная информация по услуге

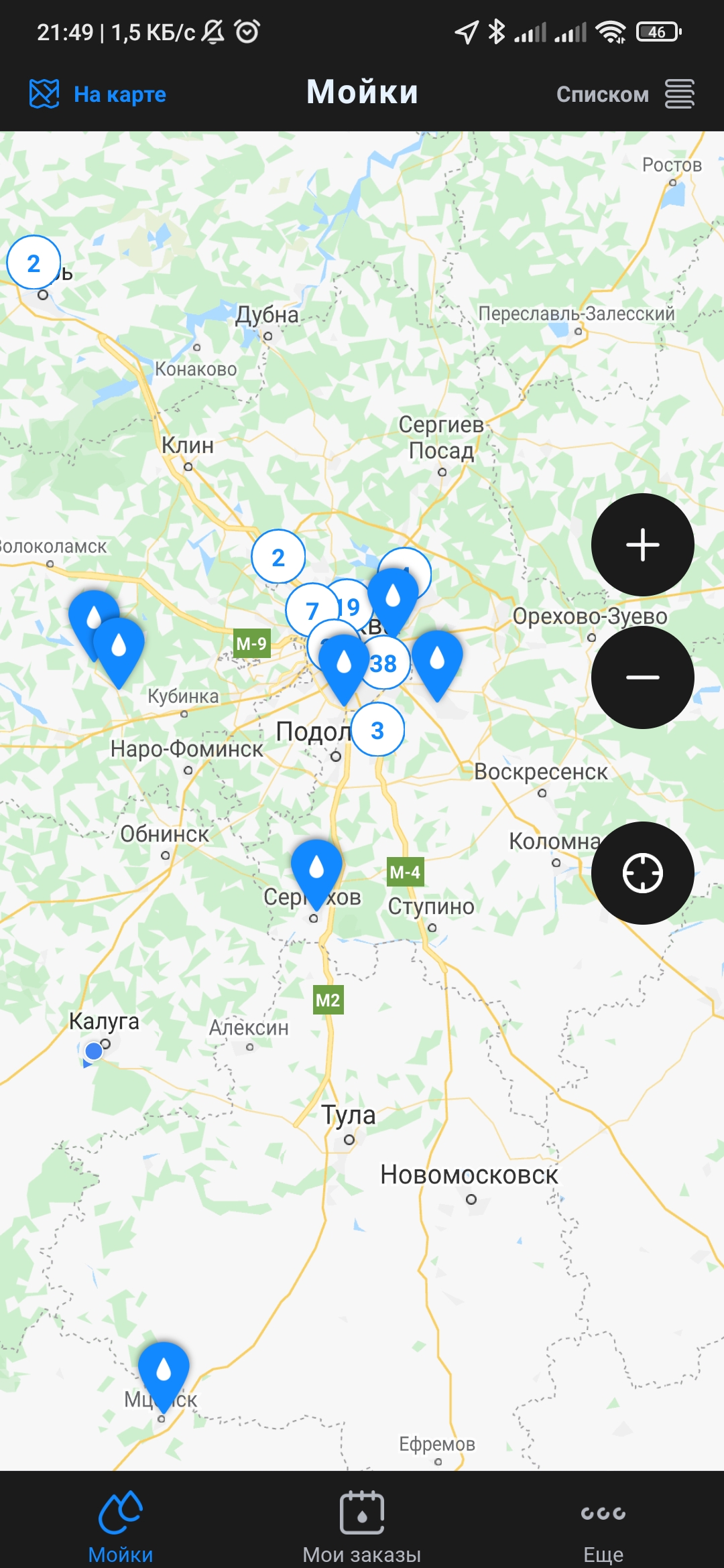


Рисунок 3. Карта в мобильном приложении «RocketWash».

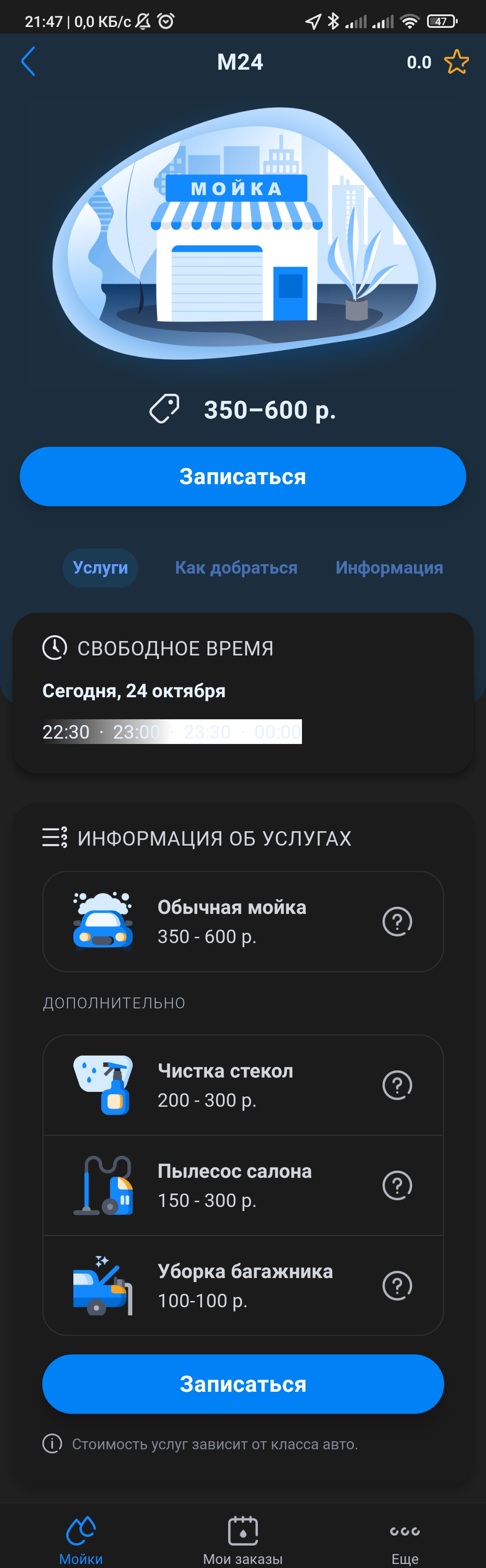


Рисунок 4. Выбрана автомойка.

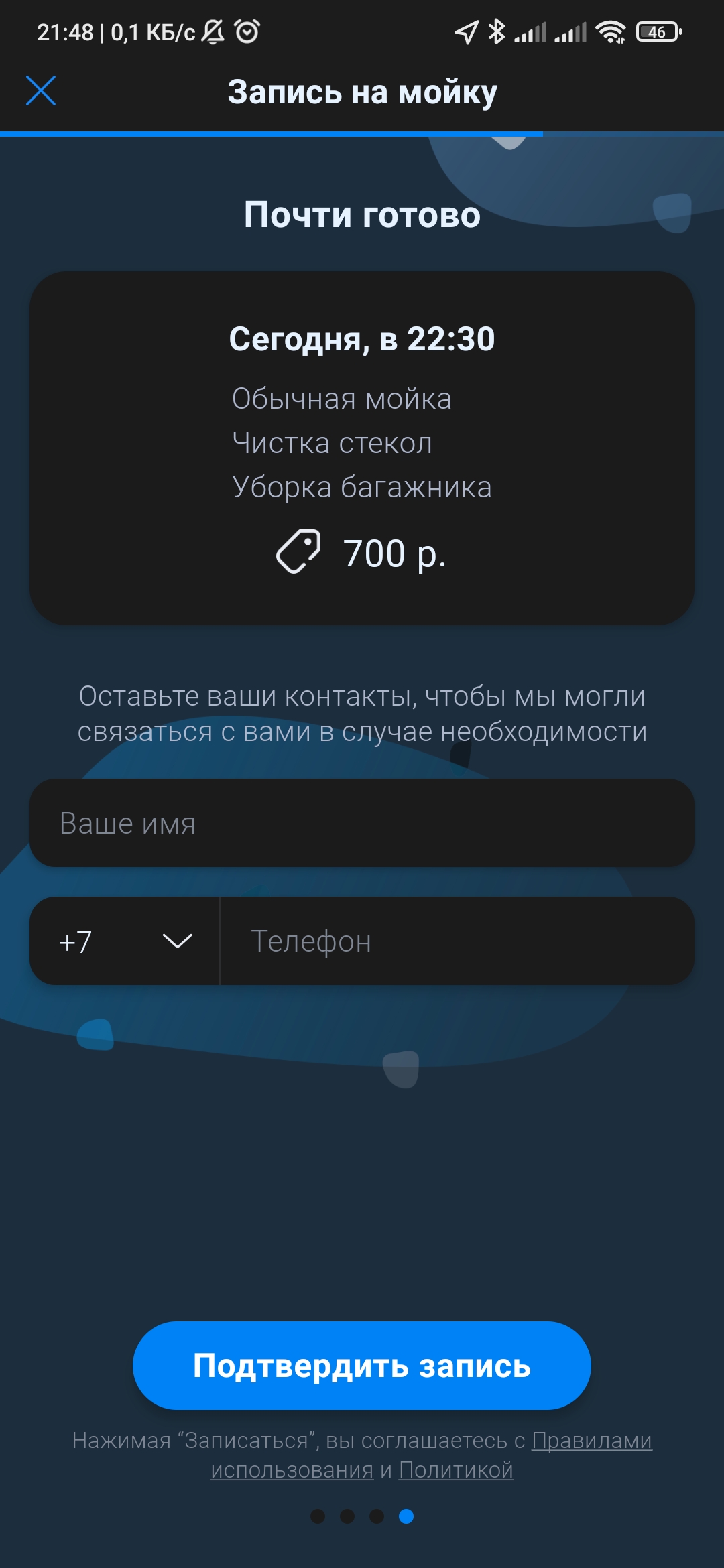
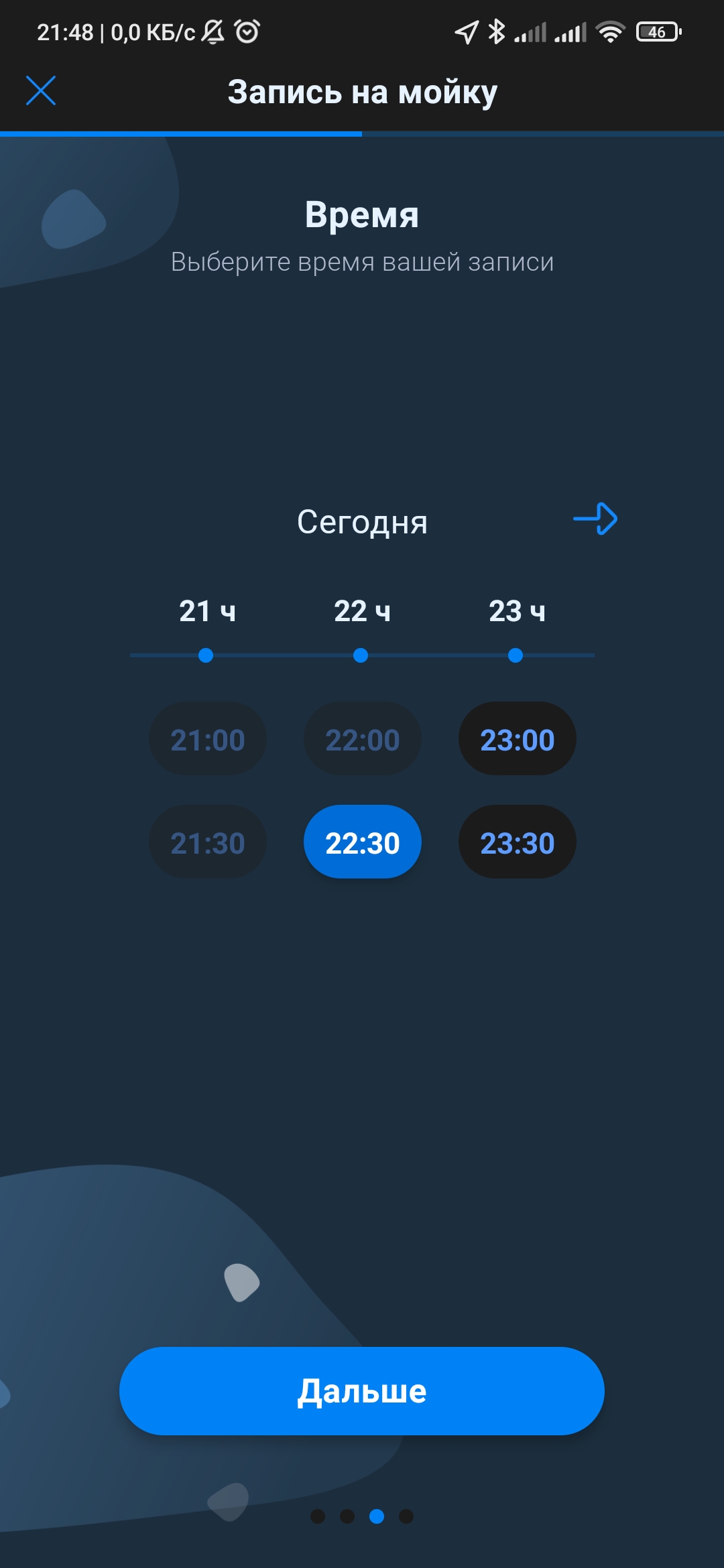
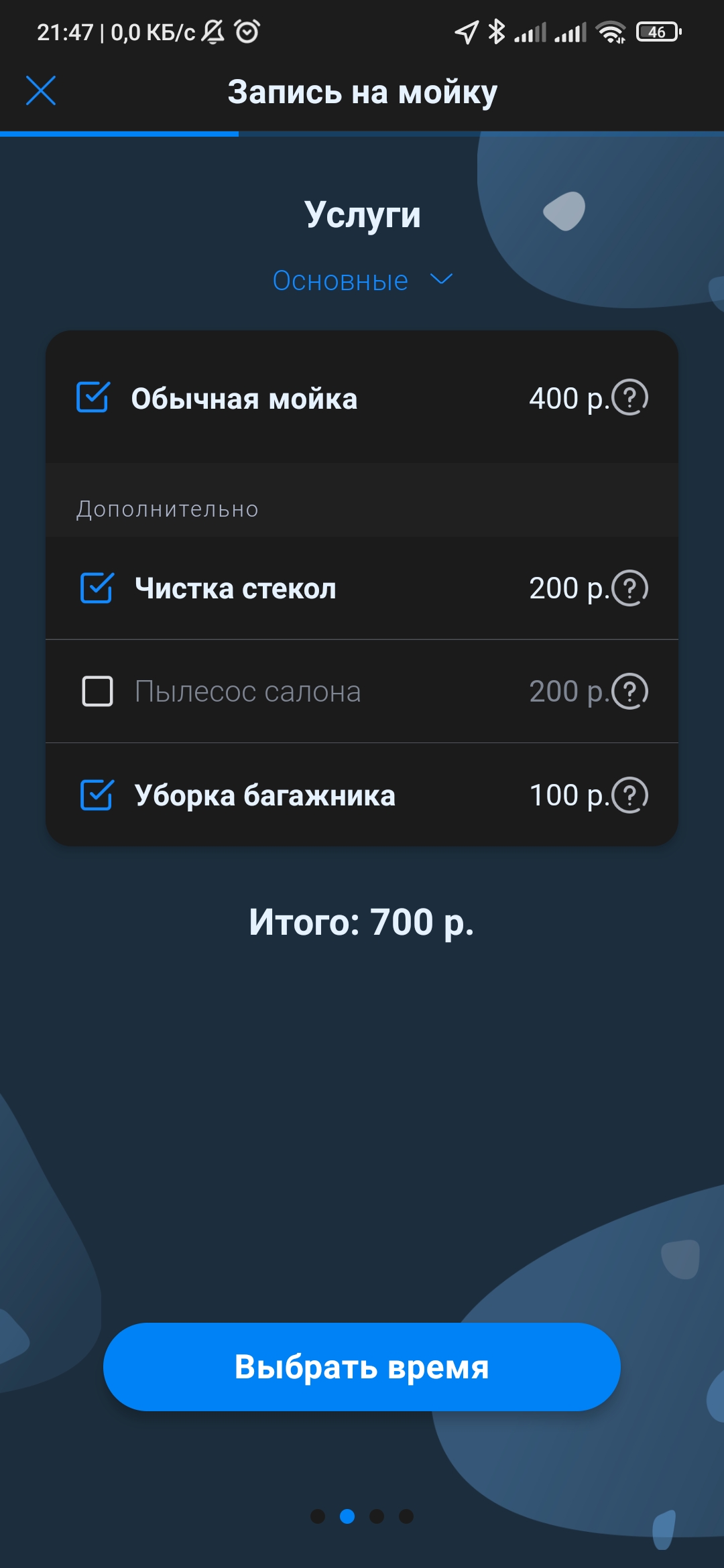


Рисунок 5. Остальные этапы записи на автомойку в приложении «RocketWash».

Плюсы:

* Информация о стоимости каждой услуги
* Выбор списка необходимых услуг с подсчётом итоговой цены
* Возможность записаться на автомойку без регистрации и авторизации в приложении

Минусы:

* Необходимость скачивания приложения
* Получасовые интервалы
* Нет информации по длительности оказания услуг

Естественно, здесь были рассмотрены далеко не все аналоги, но минусы везде примерно одинаковые.

1. Перечень задач, подлежащих решению в процессе разработки.

Исходя из результатов анализа существующих аналогов было принято решение разработать собственное веб-приложение для сети автомоек.

Интерфейс пользователя должен предоставлять пользователю функционал:

* выбор филиала автомойки;
* выбор необходимых услуг из списка;
* просмотр цены и стоимости каждой услуги;
* просмотр итоговой стоимости и времени оказания выбранных услуг;
* удобный выбор даты для записи;
* запись с временными интервалами в 5 минут (можно сделать и поминутную запись, но в этом список свободного времени будет слишком большой, что будет неудобно для пользователей).

Интерфейс администратора должен предоставлять функционал:

* добавление филиалов автомойки;
* добавление услуг с указанием цены и длительности оказания;
* добавление услуг в филиал;

В процессе разработки должен быть выделен следующий перечень задач:

* Разработать удобный и интуитивно-понятный пользовательский интерфейс с вышеперечисленным функционалом;
* Разработать личный кабинет для администратора с учётом вышеперечисленного функционала;
* Реализовать всплывающие окна с предупреждениями при неверных действиях пользователя;
* При реализации необходимо предусмотреть проверку на корректность введенных данных;

Также нужно протестировать и отладить все этапы процесса разработки ПО.

1. Обоснование выбора инструментов и платформы для разработки.

1) В качестве среды разработки используется Visual Studio Code. Интегрированная среда разработки Visual Studio Code — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio Code включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для упрощения процесса разработки.

Согласно опросу, проведенного среди 3240 разработчиков в феврале 2018 организацией Standard, разработчики выделили такие средства разработки, как наиболее часто используемые:

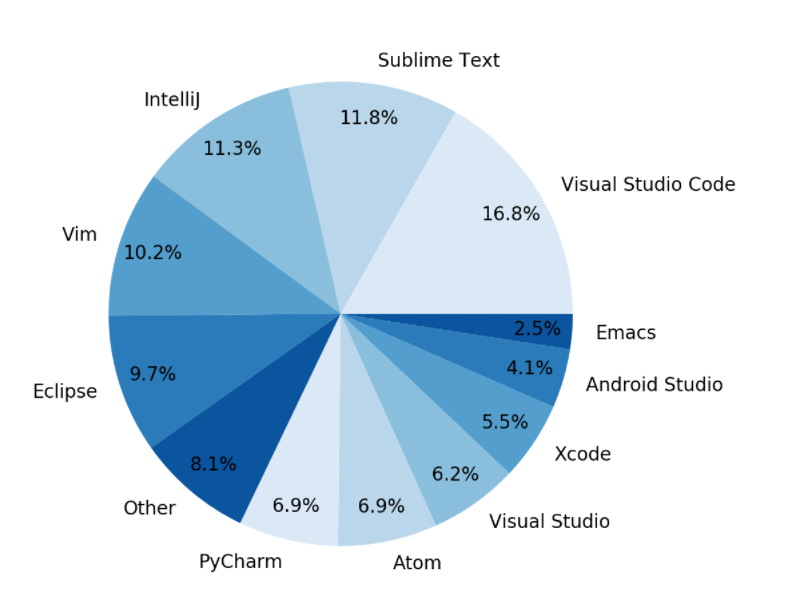


Рисунок 6. Статистика использования сред для разработки систем в мире.

Первое место среди IDE у Visual Studio Code с 16.8% использующих его разработчиков.

Также, Visual Studio Code обладает хорошим отладчиком, с помощью которого можно легко отследить ошибки в коде и без проблем их исправить, позволяет удобно и быстро создавать проекты и работать с ними.

По сравнению со средой разработки Eclipse, Visual Studio Code имеет более простой интерфейс в использовании.

1. В качестве платформы разработки будет использоваться Node.js.

Node.js – программная платформа, основанная на движке V8, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения (при помощи NW.js, AppJS или Electron для Linux, Windows и Mac OS) и даже программировать микроконтроллеры (например, tessel и espruino). В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом. Популярность в области построения web-сайтов определяется наличием большого набора встроенных средств и крупнейшего менеджера подключаемых пакетов npmjs.com для разработки web-приложений.

Основные из встроенных средств и подключаемых пакетов:

* извлечение POST и GET-параметров, а также переменных окружения web-сервера в предопределённые массивы;
* отправка HTTP-заголовков;
* работа с cookies и сессиями;
* обработка файлов, загружаемых на сервер;
* работа с HTML заголовками и HTTP авторизацией.

3) Как можно догадаться из используемой платформы, в качестве языка программирования будет использоваться JavaScript.

JavaScript – объектно-ориентированный скриптовый язык программирования. Является диалектом языка ECMAScript.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

4) Для создания интерактивного пользовательского интерфейса будут использоваться библиотеки React и React-Bootstrap.

React – JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. Его главная задача – обеспечение вывода на экран того, что можно видеть на веб-страницах. React значительно облегчает создание интерфейсов благодаря разбиению каждой страницы на небольшие фрагменты.

React-Bootstrap – это библиотека React-компонентов, подходящих для повторного использования, которая реализует возможности популярного шаблона Bootstrap. А Bootstrap в свою очередь, это – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

5) В качестве базы данных (далее по тексту – БД) была выбрана PostgreSQL.

PostgreSQL не просто реляционная, а объектно-реляционная СУБД. Это даёт ему некоторые преимущества над другими SQL базами данных с открытым исходным кодом, такими как MySQL, MariaDB и Firebird.

Фундаментальная характеристика объектно-реляционной базы данных – это поддержка пользовательских объектов и их поведения, включая типы данных, функции, операции, домены и индексы. Это делает Постгрес невероятно гибким и надежным. Среди прочего, он умеет создавать, хранить и извлекать сложные структуры данных.

6) Для работы с базой данных будет использоваться Sequalize.

Sequelize - это ORM-библиотека для приложений на Node.js, которая осуществляет сопоставление таблиц в бд и отношений между ними с классами. При использовании Sequelize мы можем не писать SQL-запросы, а работать с данными как с обычными объектами. Причем Sequelize может работать с рядом СУБД - MySQL, Postgres, MariaDB, SQLite, MS SQL Server.

В итоге, разработка приложения будет производиться в среде Visual Studio Code, на платформе Node.js, на языке JavaScript с использованием библиотек React, React-Bootstrap и Sequelize. В качестве БД будет использоваться PostgreSQL. Также естественно будут использоваться HTML и CSS.