

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»
Профиль подготовки: «Вычислительные машины, комплексы, системы и сети»

БАКАЛАВРСКАЯ РАБОТА

на тему

ВЕБ-СЕРВИС ПО ПОИСКУ ПОПУТЧИКОВ

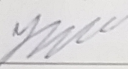
Студент


(подпись, дата)

15.06.2023

Нырков М.А.

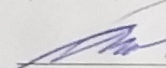
Руководитель


(подпись, дата)

15.06.2023

Слепцов Н.В.

Нормоконтролёр

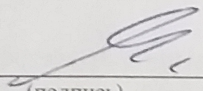

(подпись, дата)

19.06.2023

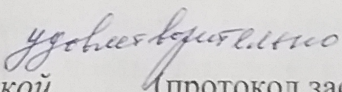
А.С. Бычков

Работа допущена к защите (протокол заседания кафедры от 02.06.2023 г. №14)

Заведующий кафедрой


(подпись)

М.А. Митрохин

Работа защищена с отметкой  (протокол заседания ГЭК от 20.06.23 № 10)

Секретарь ГЭК


(подпись)

Е.И. Гурин

Пенза 2023

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«26» 04 2023г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Нырков Михаил Александрович гр.19BBB1 факультета вычислительной техники.

2. Тема работы: «Веб-сервис по поиску попутчиков»

Тема утверждена приказом ПГУ № 458/0 от "26" апреля 2023 г.

3. Руководитель бакалаврской работы к.т.н., доцент, каф. ВТ ПГУ Слепцов Н.В.

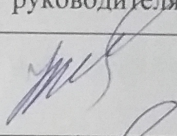
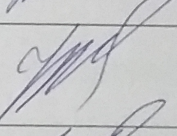
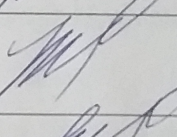
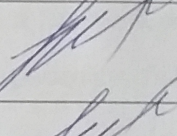
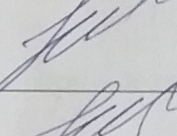
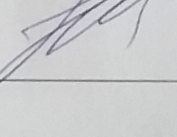
4. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать приложение «Веб-сервис по поиску попутчиков», оно должно обрабатывать функции двух видов пользователей: пассажиры и водители. Пользователь-пассажир должен иметь функционал по поиску поездок на основании определенных критериев. Приложение должно обладать графическим интерфейсом, для взаимодействия пользователей разных категорий. Разрабатываемое приложение должно предоставлять возможность фильтрации данных доступных пользователю.

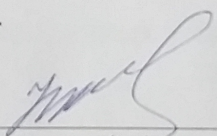
5. Перечень подлежащих разработке вопросов

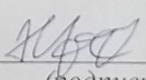
Требуется произвести анализ предметных областей: изучить существующие аналоги, выделить их достоинства и недостатки. Выбрать языки программирования и средства разработки, которые подойдут для создания графического интерфейса. Определить архитектуру веб-приложения и вспомогательные средства разработки.

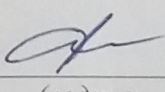
6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Изучение предметной области	32	26.04.2023 – 28.05.2023	
2	Анализ требований	10	29.04.2023 – 30.05.2023	
3	Выбор технологий, применяемых при проектировании и разработке	20	4.05.2023 – 8.05.2023	
4	Разработка клиентской части	50	9.05.2023 – 15.05.2023	
5	Разработка серверной части приложения	35	16.05.2023 – 23.05.2023	
6	Оформление пояснительной записки	54	24.05.2023 – 31.05.2023	

Дата выдачи задания "26" апреля 2023 г.

Руководитель бакалаврской работы  Слепцов Н.В.
(подпись, дата)

Задание к исполнению принял студент  Нырков М.А.
(подпись, дата)

Работу к защите допустить
Декан факультета  Фионова Л.Р.
(подпись, дата)
20.06.2023

Реферат

Бакалаврская работа содержит 107 листов, 18 рисунков, 6 таблиц, 10 источников, 11 приложений

ПОИСК ПОПУТЧИКОВ, ПУТЕШЕСТВИЕ, СОВМЕСТНАЯ ПОЕЗДКА, РАЗДЕЛЕНИЕ РАСХОДОВ ПОЕЗДКИ, ГРУППОВАЯ ПОЕЗДКА, ПОПУТЧИКИ НА АВТОМОБИЛЕ

Цель работы – разработать веб-приложение, которое по заданным критериям позволяет пассажирам определять маршрут предполагаемой поездки, а водителям размещать предложения для поиска попутчика.

Разработка проводилась в среде программирования *Visual Studio Code*. В качестве языков программирования и технологий веб-разработки были выбраны *HTML, CSS, JS, PHP, MySQL*.

В результате проведенной работы разработано приложение, удовлетворяющее поставленной задаче, а именно:

- наличие пользовательского интерактивного интерфейса;
- возможность размещения данных поездки пользователями в общей базе данных;
- наличие функции фильтрации, поиска маршрута и попутчика по заданным критериям.

ПГУ 1.090301.15.001 ПЗ

Веб-сервис по поиску попутчиков
Пояснительная записка

Лит.	Лист	Листов
	2	107

ФВТ, каф ВТ
гр. 19ВВВ1

Изм.	Лист	№ докум.	Подпись	Дата
Разраб.		Нырко М.А.		
Провер.		Слепцов Н.В.		
Н. Контр		Бычков А.С.		
Утверд.		Митрахин М.А.		

Содержание

Перечень сокращений и обозначений.....	5
Введение.....	6
1 Веб-сервис по поиску попутчиков	7
1.1 Исследование предметной области	7
1.1.1 Актуальность темы	7
1.1.2 Анализ потребностей пользователей	8
1.1.3 Анализ существующих аналогов.....	9
1.1.3.1 BlaBlaCar.	9
1.1.3.2 Едем.рф.	10
1.1.4 Постановка задачи.....	12
1.2 Выбор языков программирования и технологий веб-разработки.....	13
1.2.1 Язык гипертекстовой разметки <i>HTML</i>	13
1.2.2 Язык стилей <i>CSS</i>	14
1.2.3 Язык программирования <i>JS</i>	15
1.2.4 Язык программирования <i>PHP</i>	16
1.2.5 Система управления базами данных <i>MySQL</i>	16
1.3 Выбор инструментов разработки	18
1.3.1 Выбор среды разработки	18
1.3.2 Выбор вспомогательных библиотек.....	19
1.3.3 Выбор ПО для тестирования.....	19
1.4 Архитектура приложения.....	21
1.4.1 Паттерн <i>MVC</i>	21
1.4.2 Обзор веб-сервера <i>Apache</i>	24
1.5 Разработка приложения	26
1.5.1 Разработка фронтенда.....	26
1.5.1.1 Разметка макета страниц	27
1.5.1.2 Разработка стилей страниц.....	30
1.5.1.3 Разработка модального окна	33

1.5.2 Разработка бэкенда	35
1.5.2.1 Разработка Базы Данных	35
1.5.2.2 Отправка данных из модального окна на БД	37
1.5.2.3 Вывод содержимого из базы данных на страницу	39
1.5.2.4 Удаление устаревших данных из базы данных	39
1.5.2.5 Разработка поиска постов.....	40
1.5.3 Размещение сайта на хостинг	42
1.5.3.1 Что такое хостинг.....	42
1.5.3.2 Размещение веб-сервиса на хостинг	43
1.6 Описание веб-сервиса.....	44
1.6.1 Руководство пользователя.....	44
2 Безопасность жизнедеятельности.....	48
Список использованных источников	51
Приложение А – <i>UML</i> - Диаграммы.....	52
Приложение Б - Листинг	55
Приложение В - Презентация	100

Перечень сокращений и обозначений

В настоящей работе применяют следующие сокращения и обозначения

БД – база данных

ПК – персональный компьютер

ПО – программное обеспечение

СУБД – система управления базами данных

API (Application Programming Interface) - набор готовых правил и определений

CSS (Cascading Style Sheets) - каскадные таблицы стилей

GUI (Graphical User Interface) - графический интерфейс пользователя

HTML (HyperText Markup Language) - язык разметки

IDE (Integrated Development Environment) - интегрированная среда разработки

MVC (Model View Controller) – шаблон проектирования архитектуры приложения

UML (Unified Modeling Language) - унифицированный язык моделирования

XML (eXtensible Markup Language) - расширяемый язык разметки

Введение

Веб-сервис по поиску попутчиков — это платформа, объединяющая единомышленников, которые ищут попутчиков. Эти веб-сервисы предоставляют множество вариантов выбора, позволяя пользователям фильтровать потенциальных попутчиков по месту назначения, датам поездки, интересам и другим важным критериям. С помощью этих веб-сервисов путешественники могут найти кого-то, кто разделяет их интересы, бюджет и стиль путешествия, что упрощает планирование и проведение приятной поездки.

Целью выпускной работы является разработка веб-сервиса по поиску попутчиков.

В ходе данной работы необходимо создать веб-сервис по поиску попутчиков. Веб-сервис должен предоставлять удобный веб-интерфейс для визуализации всей необходимой информации по поиску попутчиков. Входным файлом является база данных с информацией о путешественниках, из которой необходимо импортировать информацию о доступных попутчиках. Результатом работы веб-сервиса будет предоставление пользователю таблицы с информацией о попутчиках, подходящих для совместной поездки.

1 Веб-сервис по поиску попутчиков

1.1 Исследование предметной области

1.1.1 Актуальность темы

Актуальность темы, прежде всего, обусловлена тем, что веб-сервис по поиску попутчиков может предоставить людям удобный и надежный способ расширить свои сети поездок, поделиться советами и рекомендациями и потенциально сэкономить деньги за счет разделения расходов. Кроме того, это может помочь одиноким путешественникам найти компанию и сделать их поездку более комфортной и безопасной.

Веб-службы являются важным инструментом для многих людей для доступа к информации или услугам в Интернете. Одним из таких сервисов, набравших популярность в последние годы, являются веб-сервисы для поиска попутчиков. Эти сервисы позволяют пользователям находить других путешественников, которые направляются в тот же пункт назначения или имеют схожие планы поездок, тем самым позволяя им сотрудничать и делиться опытом путешествий.

Одним из самых значительных преимуществ этого веб-сервиса является то, что он помогает пользователям экономить деньги. Сотрудничая с другими путешественниками, пользователи могут разделить расходы, такие как транспорт, проживание и питание. Это не только снижает стоимость поездки, но и позволяет путешественникам исследовать больше мест в рамках своего бюджета.

Еще одним существенным преимуществом этого веб-сервиса является то, что он повышает безопасность и защищенность путешественников. Найдя попутчиков, пользователи могут путешествовать вместе, что делает их путешествие более безопасным. Кроме того, пользователи также могут получить рекомендации о безопасных местах для посещения, избежать мошенничества и перемещаться по незнакомым территориям.

Сегодня веб-сервис для поиска попутчиков предлагает пользователям множество преимуществ. Он создает сообщество путешественников, повышает безопасность и позволяет пользователям экономить деньги. Таким образом, эта услуга является отличным инструментом для всех, кто любит путешествовать и хочет сделать свое путешествие более приятным и запоминающимся. На рисунке 1.1 показан график популярности запроса «попутчик» с 2015 по 2023 года.



Рисунок 1.1 – График популярности запросов

1.1.2 Анализ потребностей пользователей

Исходя из темы «веб-сервис по поиску попутчиков», можно выделить следующие возможные потребности пользователей:

- интуитивно понятный интерфейс. Пользователям нужен простой и удобный интерфейс, который безошибочно направил бы его на совершение нужного действия;
- параметры поиска и фильтрации. Пользователям нужна эффективная и удобная система поиска и фильтрации, которая помогла бы им найти

попутчиков на основе пункта назначения, даты поездки и множества других соответствующих критериев;

- наличие инструмента для общения. пользователи должны иметь возможность общаться со своими потенциальными попутчиками через систему обмена сообщениями или другие средства связи, предоставляемые службой;

- безопасность. Пользователям нужна безопасная платформа, которая защитит их и обеспечит безопасность во время поездки;

- отзывы. Пользователям нужна система, которая позволяла бы им оставлять отзывы о своем путешествии со своими попутчиками, помогая другим пользователям принимать обоснованные решения;

- удобный для мобильных устройств интерфейс. Пользователям нужен удобный для мобильных устройств интерфейс, который позволил бы им получить доступ к услуге на своих мобильных устройствах, что упростило бы им поиск попутчиков на ходу;

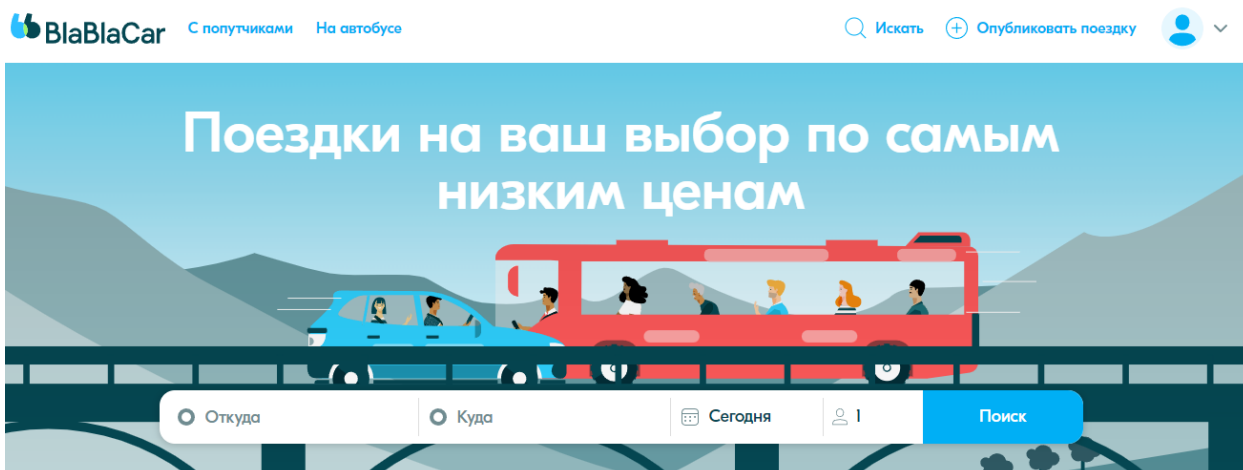
- в целом веб-сервис по поиску попутчиков должен быть удобным, надежным и безопасным, предоставляя пользователям платформу для поиска подходящих попутчиков и повышения удовольствия от путешествия.

1.1.3 Анализ существующих аналогов

Среди аналогов веб-сервисов по поиску попутчиков, можно выделить:

1.1.3.1 *BlaBlaCar*.

Сайт для быстрого поиска попутной машины или автобуса для междугородних поездок. Работает с 2006 г. в 22 странах мира. Свыше 50 млн пользователей, 96% которых из РФ. Более 5,5 млн визитов в месяц. Плюсы: низкие цены, отзывы о водителях, мобильное приложение. На рисунке 1.2 показана главная страница данного сервиса.



- Ваша поездка по низкой цене**
Куда бы вы ни собирались, на автобусе или с попутчиками, вы сможете найти свою идеальную поездку среди множества маршрутов и доехать по самой низкой цене.
- Доверяйте своим попутчикам**
Мы стараемся узнать ваших будущих попутчиков и автобусных перевозчиков как можно лучше. Мы проверяем отзывы, профили и паспортные данные попутчиков, чтобы вы знали, с кем поедете.
- В дорогу за пару кликов!**
Забронировать поездку проще простого. В нашем приложении легко разобраться: мощный алгоритм за пару минут найдет водителя поблизости, и вам останется нажать пару кнопок для брони.

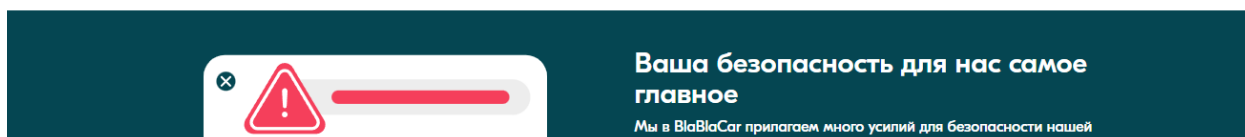


Рисунок 1.2 – Сервис поиска автомобильных попутчиков *BlaBlaCar*

Присутствуют следующие разделы:

- главная, на ней располагаются разделы для поездок на автобусе или автомобиле, а так же специальное поле для пассажиров, через которое они могут найти интересующую их поездку. Так же имеются раздел с популярными маршрутами и рекомендациями по безопасности;
- помимо этого на сайте имеется информация о сайте, контактные данные и вся необходимая дополнительная информация необходимая покупателю;
- ссылка на авторизацию на сайте. Данный веб-сервис основан в 2006 году Фредериком Маззелла во Франции в г.Париж.

1.1.3.2 Едем.рф.

Сайт поиска попутчиков на машине для выгодных поездок по России. Более 1 млн. зарегистрированных пользователей. Свыше 700 тыс. посещений

					ПГУ 1.090301.15.001 ПЗ	Лист 10
Изм.	Лист	№ докум.	Подпись	Дата		

в месяц. Плюсы: удобный интерфейс, мобильное приложение. Минусы: относительно небольшая клиентская база, за объявление о поездке с водителем берется плата.

На рисунке 1.3 показана главная страница данного сервиса.

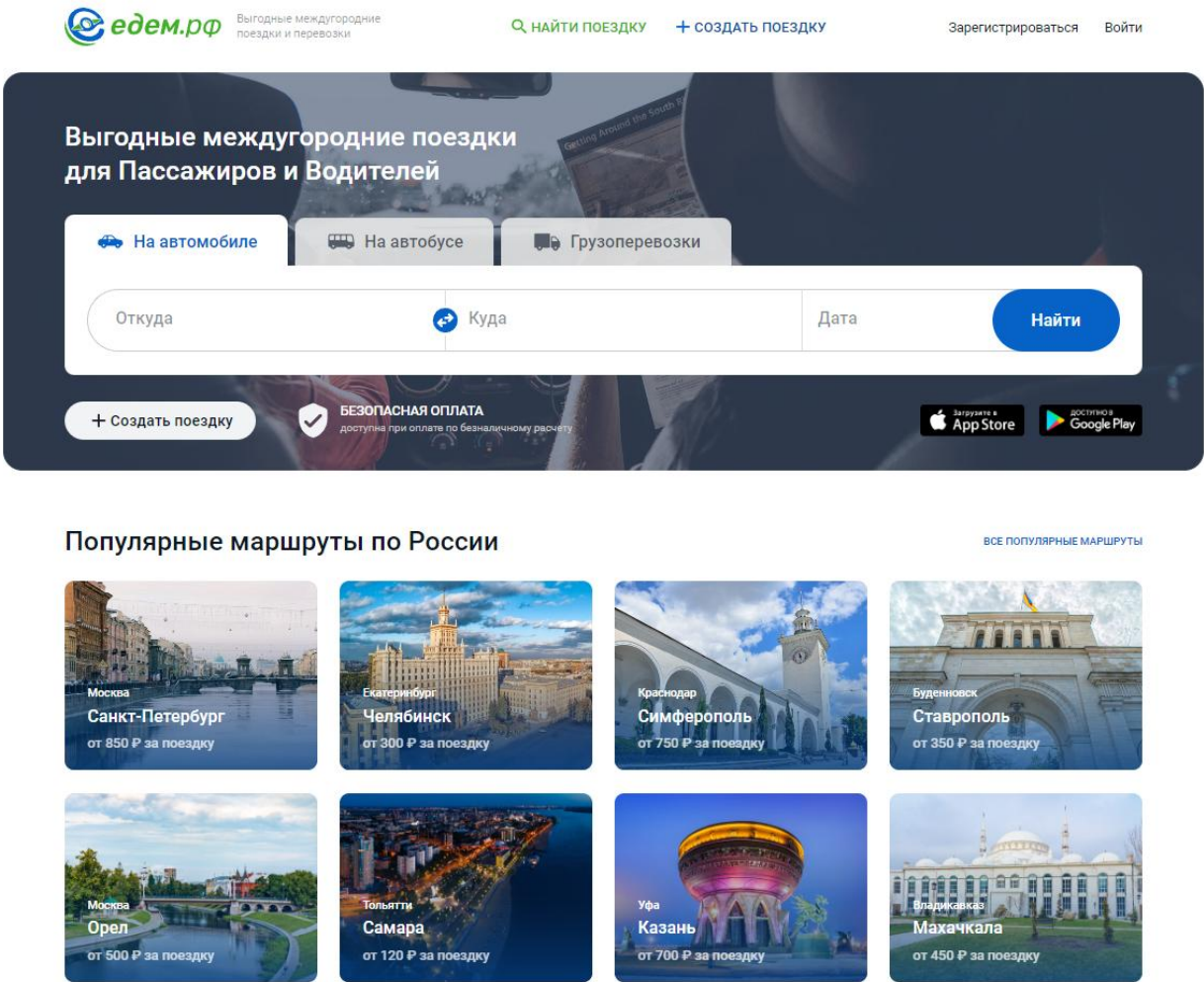


Рисунок 1.3 – Сервис поиска автомобильных попутчиков *едем.рф*

Присутствуют следующие разделы:

- главная, на ней располагаются разделы для поездок на автобусе, автомобиле и возможность воспользоваться грузоперевозкой, а также специальное поле для пассажиров, через которое они могут найти интересующую их поездку;

- так же имеется раздел с популярными маршрутами и рекомендациями по безопасности и ссылка на авторизацию;помимо этого на сайте имеется информация о сайте, контактные данные и вся необходимая дополнительная информация необходимая покупателю.

1.1.4 Постановка задачи

Для того чтобы правильно и объективно сформулировать задачи, необходимо четко представлять себе предметную область разработки. На рисунке 1.4 представлен план разработки.

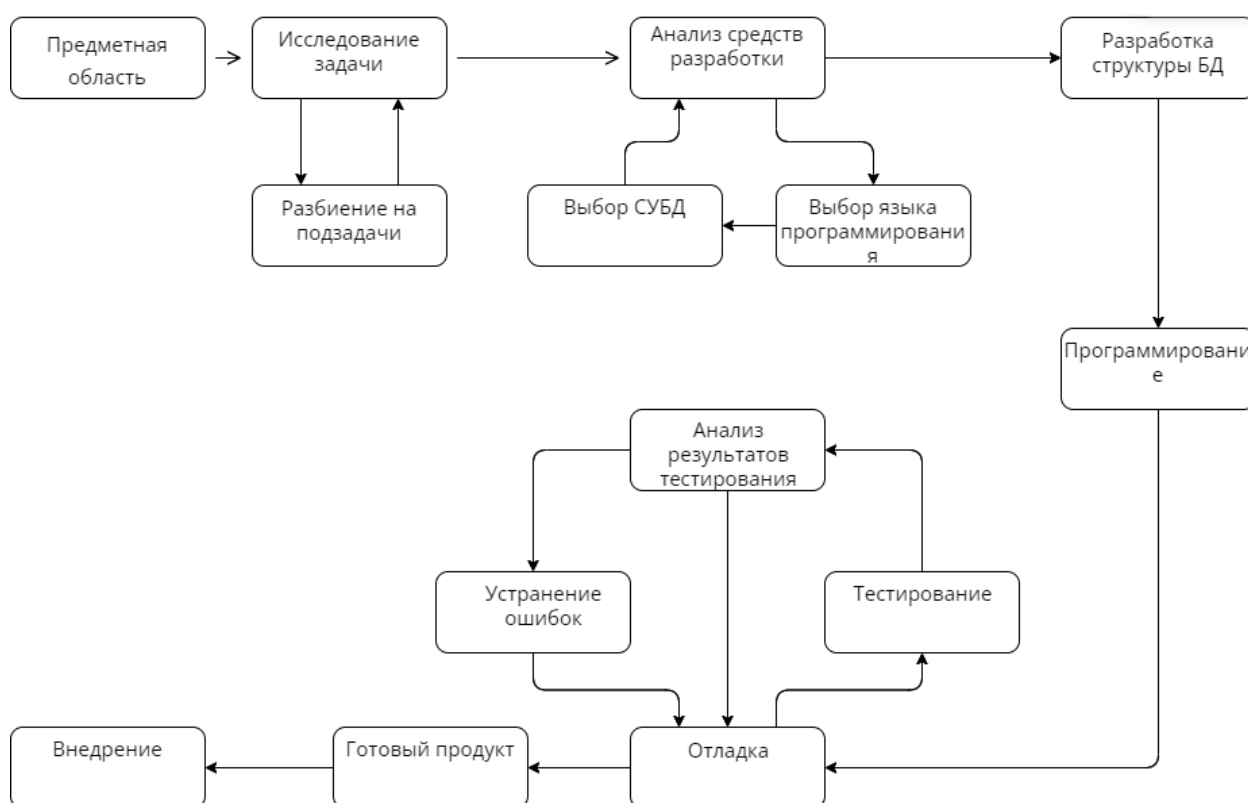


Рисунок 1.4 – Разработка программного продукта

Необходимо разработать программу «Веб-сервис по поиску попутчиков».

В данном веб-сервисе будет реализован функционал, отвечающий за создание и отправку данных о поездке от водителей на страницу с просмотром существующих предложений. Возможности сервиса должны предоставлять взаимодействие между водителем и потенциальным

пассажиром, форма создания поездки водителем, просмотр активных поездок и фильтрация контента. Для решения этих задач необходимо разработать и реализовать простой, удобный и понятный пользовательский интерфейс.

Веб-приложение должно быть разработано с несколькими логически организованными модулями. Такой подход упрощает процесс добавления новых функций в программное обеспечение в будущем, а также упрощает выявление и исправление любых ошибок во время отладки и тестирования.

1.2 Выбор языков программирования и технологий веб-разработки

В соответствии с поставленной задачей необходимо разработать компоненты Web-представительства «сервис по поиску попутчиков». Современный инструментарий разработчика предлагает большой выбор средств для реализации такого рода проектов. Прежде чем приступить к разработке, были рассмотрены существующие среды и средства для разработки Web-приложений.

1.2.1 Язык гипертекстовой разметки *HTML*

HTML или язык гипертекстовой разметки — это стандартный язык разметки, используемый для создания веб-страниц и приложений. Это основа всех веб-страниц в Интернете.

Таблица 1.1 – Плюсы и минусы *HTML*

Плюсы	Минусы
Простота в освоении и использовании	Ограниченная интерактивность и функциональность без дополнительных языков сценариев, таких как <i>JavaScript</i>
Поддерживается всеми браузерами	Может потребоваться много времени для создания сложных макетов и дизайнов
Обеспечивает структуру и	Может быть сложно поддерживать

Плюсы	Минусы
семантику веб-контента	согласованность и доступность на разных устройствах и в разных браузерах
Легко интегрируется с другими технологиями	При неправильном использовании могут возникнуть уязвимости безопасности

1.2.2 Язык стилей CSS

CSS или каскадные таблицы стилей — это язык стилей, используемый для управления представлением и макетом веб-страниц. Он используется для определения внешнего вида *HTML*-элементов, включая их размер, цвет, шрифт, интервалы и другие визуальные характеристики.

Таблица 1.2 – Плюсы и минусы CSS

Плюсы	Минусы
Разделение задач: CSS позволяет четко разделить содержимое и представление, упрощая поддержку и обновление веб-сайтов	Совместимость с браузерами: разные браузеры могут интерпретировать CSS немного по-разному, что приводит к несоответствиям во внешнем виде
Согласованность: определяя стили в одном месте, CSS может гарантировать, что все страницы на веб-сайте будут выглядеть одинаково	Кривая обучения: CSS может быть трудным для изучения, особенно для новичков, которые могут испытывать трудности с синтаксисом и макетом
Гибкость: CSS предоставляет широкий спектр вариантов стилей, что позволяет создавать сложные макеты и дизайны	Специфика: CSS может быть сложным и требовать глубокого понимания специфики селекторов и правил наследования
Эффективность: CSS может уменьшить количество кода,	Обслуживание: CSS со временем может усложниться, особенно если на веб-

Плюсы	Минусы
необходимого для оформления веб-сайта, что может ускорить загрузку страниц	сайте много страниц или над проектом работает несколько разработчиков

1.2.3 Язык программирования JS

JavaScript (JS) — это высокоуровневый интерпретируемый язык программирования, который обычно используется для создания интерактивных веб-страниц и динамических пользовательских интерфейсов. Это позволяет разработчикам добавлять функциональные возможности на свои веб-сайты и веб-приложения, включая анимацию, проверку форм и другие динамические элементы. *JavaScript* часто называют клиентским языком, что означает, что он работает на компьютере пользователя, а не на сервере. Он также используется в серверных средах с помощью таких технологий, как *Node.js*. *JS* — очень популярный язык, который поддерживается всеми современными веб-браузерами.

Таблица 1.3 – Плюсы и минусы *JS*

Плюсы	Минусы
Он лёгок в изучении и реализации	Он может быть подвержен ошибкам если не закодирован должным образом
Он поддерживается всеми основными веб-браузерами	Он может быть уязвим для угроз безопасности, если не защищен должным образом
Он универсален и может использоваться для самых разных целей	Он может вызвать проблемы с производительностью, если не оптимизировано должным образом
Он постоянно развивается и	Это не серверный язык

Плюсы	Минусы
улучшается с новыми обновлениями и функциями	программирования, поэтому он не может самостоятельно выполнять сложные серверные операции

1.2.4 Язык программирования *PHP*

PHP (препроцессор гипертекста) — это серверный язык сценариев, используемый для веб-разработки. Он имеет открытый исходный код, бесплатен и широко используется. Он может быть легко встроен в *HTML* и использоваться для создания динамических веб-страниц, обработки и хранения данных из веб-форм и взаимодействия с базами данных. Он также используется для создания систем управления контентом, веб-сайтов электронной коммерции и веб-приложений.

Таблица 1.4 – Плюсы и минусы *PHP*

Плюсы	Минусы
Может быть легко встроен в <i>HTML</i>	Не так быстр, как другие языки, такие как <i>Java</i> или <i>Python</i>
Кроссплатформенная совместимость	Уязвимости безопасности могут возникнуть при неправильном обращении
Большая поддержка сообщества	Несовместимые соглашения об именах и документации
Огромное количество доступных фреймворков и библиотек	Может привести к коду, который трудно поддерживать и масштабировать
Поддерживает различные БД	Плохо справляется с многопоточностью

1.2.5 Система управления базами данных *MySQL*

MySQL — это система управления реляционными базами данных (*RDBMS*) с открытым исходным кодом, которая широко используется для управления и хранения данных в различных приложениях. Она написана на

языках программирования C и C++ и является одной из самых популярных СУБД, используемых в веб-приложениях, благодаря совместимости с различными операционными системами и масштабируемости. MySQL использует SQL (язык структурированных запросов) для взаимодействия с базой данных и известен своей способностью эффективно обрабатывать большие объемы данных. Он также обладает высокой степенью безопасности и предоставляет различные функции, такие как управление пользователями, резервное копирование и восстановление, а также репликация для избыточности данных. Кроме того, MySQL имеет обширное сообщество разработчиков, которые вносят свой вклад в его развитие и обеспечивают поддержку пользователей.

Таблица 1.5 – Плюсы и минусы MySQL

Плюсы	Минусы
Открытый исходный код: MySQL бесплатна для использования и широко поддерживается сообществом разработчиков.	Ограниченные возможности: по сравнению с другими базами данных MySQL имеет ограниченные возможности.
Высокая производительность: MySQL может обрабатывать большие объемы данных и известен высокой скоростью обработки.	Безопасность: MySQL имеет репутацию менее безопасной по сравнению с другими базами данных.
Простота использования: MySQL имеет удобный интерфейс, в котором легко ориентироваться и использовать.	Ограниченная поддержка: MySQL имеет ограниченную систему поддержки по сравнению с другими коммерческими базами данных.
Масштабируемость: MySQL можно легко масштабировать для удовлетворения потребностей	нет

Плюсы	Минусы
растущего бизнеса.	
Доступность: <i>MySQL</i> доступен для нескольких платформ, включая <i>Windows</i> , <i>Linux</i> и <i>Mac OS</i> .	нет

1.3 Выбор инструментов разработки

1.3.1 Выбор среды разработки

В качестве среды разработки было решено выбрать *Visual Studio Code*. *VS Code* — это легкий редактор кода с открытым исходным кодом, разработанный *Microsoft*. Он предлагает множество преимуществ, таких как:

- простота использования. *VS Code* легко установить и использовать, что делает его доступным как для начинающих, так и для опытных разработчиков;
- расширяемость. *VS Code* поддерживает широкий спектр расширений, которые можно установить и настроить для расширения возможностей в соответствии с конкретными потребностями разработки;
- навигация по коду. *VS Code* имеет мощную функцию поиска, которую можно использовать для быстрой и эффективной навигации по кодовым базам;
- отладка. *VS Code* поставляется со встроенными инструментами отладки, которые легко настроить и использовать;
- совместная работа. *VS Code* обеспечивает эффективную совместную работу над кодовыми базами благодаря таким функциям, как интеграция с *Git* и возможность совместного использования;
- многоязычная поддержка. *VS Code* поддерживает несколько языков программирования, включая *JavaScript*, *Python*, *C++* и многие другие;

- межплатформенная поддержка. *VS Code* доступен на нескольких платформах, включая *Windows*, *Mac* и *Linux*. 1.5.2 Выбор платформы для реализации графического интерфейса.

1.3.2 Выбор вспомогательных библиотек

В качестве вспомогательной библиотеки *JS* был выбран *JQuery*.

JQuery — это облегченная библиотека *JS*, упрощающая процесс манипулирования HTML-документами, обработки событий и создания анимации. Он предлагает разработчикам простой в использовании *API*, который позволяет им быстро выполнять сложные задачи с минимальным кодом.

JQuery предлагает несколько преимуществ, в том числе улучшенную кросс-браузерную совместимость, сокращение времени разработки и уменьшение сложности кода. Он также предоставляет широкий спектр плагинов и виджетов, которые можно легко интегрировать в проекты, что делает его универсальным инструментом для создания сложных веб-приложений.

Благодаря интуитивно понятному синтаксису и мощным функциям *JQuery* стала одной из самых популярных используемых сегодня библиотек *JS*. *JQuery* широко используется разработчиками для создания динамических и интерактивных веб-страниц, что делает его важным инструментом для всех, кто занимается веб-разработкой.

1.3.3 Выбор ПО для тестирования

Open Server - это пакет программного обеспечения, предназначенный для разработки и локального тестирования веб-приложений на платформе *Windows*. Он включает в себя все необходимые компоненты, такие как веб-сервер *Apache*, сервер баз данных *MySQL*, интерпретатор *PHP* и другие инструменты. Одной из функций *Open Server* является предустановленный *phpMyAdmin*, который обеспечивает удобное управление базами данных

MySQL. Вот некоторые особенности и функциональность *Open Server* и *phpMyAdmin*:

- установка и настройка. *Open Server* обеспечивает простую установку и настройку необходимых компонентов, что делает его доступным даже для новичков в области веб-разработки. Он предлагает удобный мастер установки, который позволяет выбрать нужные компоненты и параметры;

- веб-сервер *Apache*. *Open Server* включает в себя веб-сервер *Apache*, который является одним из самых популярных веб-серверов в мире. Он обеспечивает обработку *HTTP*-запросов и раздачу веб-страниц пользователям;

- сервер баз данных *MySQL*. *Open Server* поставляется с сервером баз данных *MySQL*, который позволяет создавать и управлять базами данных, таблицами, индексами и другими аспектами данных. *MySQL* является одной из самых распространенных реляционных баз данных, используемых в веб-разработке;

- интерпретатор *PHP*. *Open Server* включает в себя интерпретатор *PHP*, который является одним из самых популярных языков программирования для веб-разработки. *PHP* позволяет создавать динамические веб-страницы, обрабатывать формы, взаимодействовать с базами данных и многое другое;

- *PhpMyAdmin*. *Open Server* предоставляет предустановленный *phpMyAdmin*, который обеспечивает удобный веб-интерфейс для управления базами данных *MySQL*. С его помощью вы можете создавать и удалять базы данных, таблицы, выполнять *SQL*-запросы, просматривать и редактировать данные и многое другое;

- управление хостами и виртуальными хостами. *Open Server* позволяет создавать и управлять хостами и виртуальными хостами для веб-приложений. Это позволяет разрабатывать и тестировать приложения в разных средах, включая локальный сервер;

1.4 Архитектура приложения

1.4.1 Паттерн MVC

Паттерн - повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста. Всего существует три группы паттернов, которые приведены ниже на структурной диаграмме.

На рисунке 1.5 показаны основные паттерны проектирования.

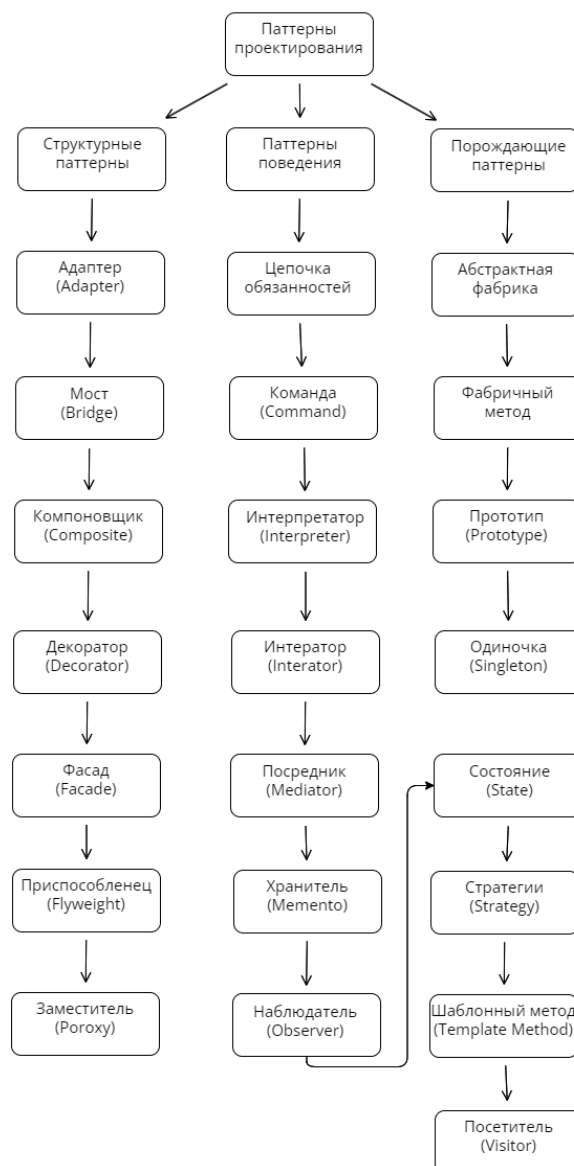


Рисунок 1.5 – Группы паттернов

Структурные паттерны связаны с организацией кода таким образом, чтобы его было легко читать, понимать и модифицировать. Они предоставляют решения для общих проблем с композицией объектов и структурами классов. Примеры структурных паттернов включают паттерн адаптера, который позволяет несовместимым классам работать вместе, выступая в качестве моста между ними, и паттерн декоратора, который добавляет функциональность к существующим объектам, не изменяя их структуру.

Порождающие паттерны используются для создания объектов и классов таким образом, чтобы они были гибкими, расширяемыми и пригодными для повторного использования. Они предоставляют решения для проблем с созданием и инициализацией объектов. Примеры порождающих паттернов включают паттерн *Factory*, который отделяет создание объекта от остального кода, и паттерн *Singleton*, который гарантирует, что существует только один экземпляр класса.

Поведенческие паттерны связаны с взаимодействием и общением между объектами и классами. Они предоставляют решения для общих проблем с поведением объекта и коммуникацией. Примеры поведенческих паттернов включают паттерн *Observer*, который позволяет объектам уведомляться об изменениях в других объектах, и паттерн *Command*, который инкапсулирует запрос в виде объекта, позволяя передавать его и выполнять позже.

В случае разработки веб-сервиса по поиску попутчиков является паттерн *MVC*.

Паттерн *Model-View-Controller* (*MVC*) — это архитектура программного обеспечения, которая обычно используется при разработке пользовательских интерфейсов. Он разделяет программное обеспечение на три компонента: модель, представление и контроллер. Эти компоненты предназначены для совместной работы, чтобы обеспечить удобный

интерфейс, делая разработку программного обеспечения более эффективной и простой в обслуживании.

Структура паттерна *MVC* представлена на рисунке 1.6

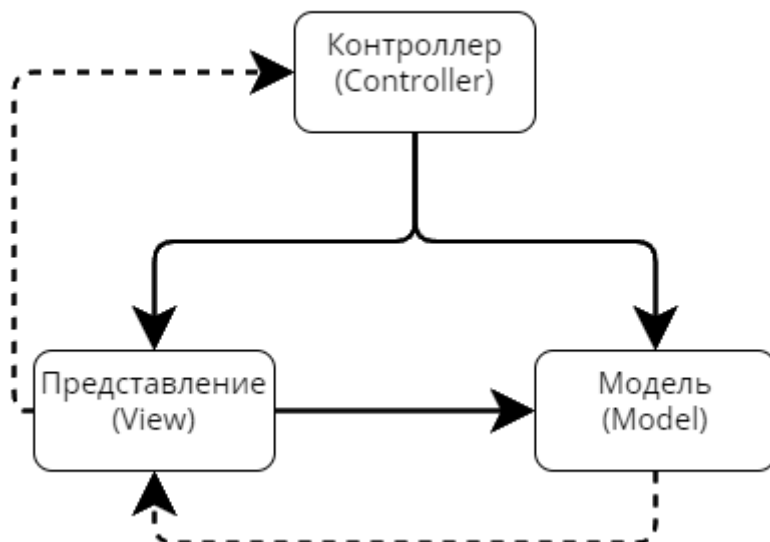


Рисунок 1.6 – Структура паттерна проектирования *MVC*

Модель отвечает за управление данными и бизнес-логикой приложения. Это ядро приложения, поскольку оно представляет данные и функции, с которыми работает приложение. Модель получает входные данные от контроллера и обновляет представление всякий раз, когда происходят изменения. Он также обновляет контроллер последними данными и уведомляет представление о любых изменениях.

Представление отвечает за отображение данных в пользовательском интерфейсе. Он отображает данные из модели и позволяет пользователю взаимодействовать с приложением. Представление получает ввод от пользователя и отправляет его контроллеру. Он также прослушивает обновления от модели и соответствующим образом обновляет себя.

Контроллер отвечает за управление потоком информации между Моделью и Представлением. Он действует как посредник между ними, обрабатывая пользовательский ввод и соответствующим образом обновляя

модель и представление. Он получает запросы от представления и обновляет модель новыми данными. Он также обновляет представление с изменениями из модели. На рисунке 1.7 показана диаграмма взаимодействия выбранного паттерна.

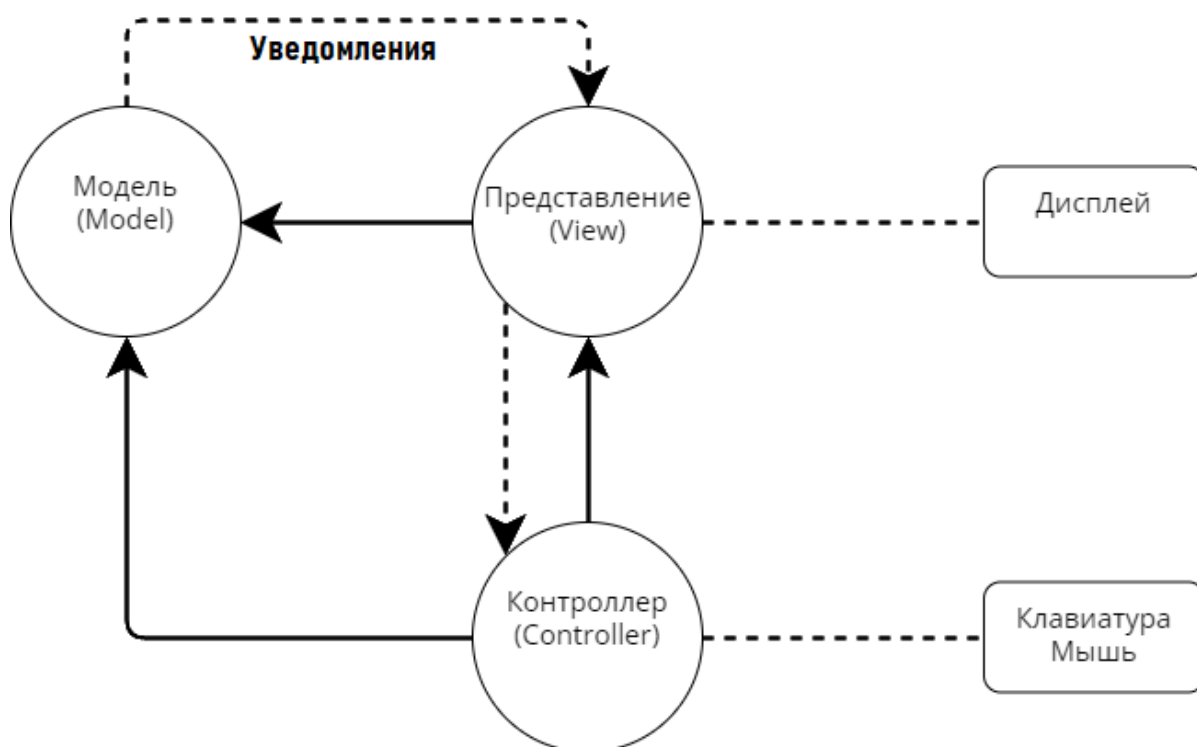


Рисунок 1.7 – Диаграмма взаимодействия MVC

Преимущества использования паттерна MVC многочисленны. Он помогает программистам разделить различные аспекты приложения, упрощая обслуживание и обновление программного обеспечения. Модель можно тестировать независимо от представления и контроллера, что означает, что изменения можно вносить, не затрагивая пользовательский интерфейс. Представление и контроллер также можно изменить, не затрагивая модель. Это упрощает добавление новых функций и функций в программное обеспечение.

1.4.2 Обзор веб-сервера Apache

В последние годы Интернет прочно вошел во все сферы человеческой жизни и деятельности, став доступным для Министерства обороны США,

государственных и коммерческих организаций, частных лиц и любых других пользователей. Сегодня многие считают, что Интернет ограничивается лишь веб-браузерами и WWW. Однако WWW, в отличие от других протоколов, позволяет не только передавать файлы и текстовые сообщения, но и показывать картинки, передавать звук и видео в режиме реального времени. Основным протоколом для WWW является *HTTP (Hypertext Transfer Protocol)*, который определяет обмен данными по принципу "запрос-ответ". Клиент отправляет запрос серверу, содержащий информацию о типе запроса, заголовке запроса и самом запросе. В ответ сервер отправляет клиенту код состояния, заголовок ответа и саму информацию.

На сегодняшний день существует множество веб-серверов, но одним из наиболее популярных является Apache, разработанный *Apache Software Foundation (ASF)*. Apache является самым распространенным веб-сервером в мире и имеет версии для различных платформ и операционных систем, включая *Windows NT*. Его преимущества включают надежность, высокую производительность и обширный набор функций и модулей. Однако наиболее заметной особенностью Apache является его свободное распространение и открытый код, что позволяет быстро исправлять ошибки и вносить изменения в программный код.

Apache поддерживает кросс-платформенность, протоколы *HTTP/1* и *HTTP/2*, имеет модульную структуру, обеспечивает защиту и запись log-файлов. Кроме того, Apache предлагает множество дополнительных модулей, расширяющих его функциональность. Благодаря свободному распространению и открытому коду, можно создавать высокоэффективные и безопасные модули, что делает Apache непревзойденным среди коммерческих серверов.

Однако, несмотря на все его достоинства, у Apache есть один недостаток: отсутствие удобной программы установки и управления. При установке необходимо работать с командной строкой и иногда вносить

изменения в конфигурационные файлы под *UNIX*. Этот нетипичный подход, хотя и неудобный на первый взгляд, не делает *Apache* менее популярным, и он является главным конкурентом на рынке веб-серверов.

Apache обладает множеством функций, включая поддержку различных операционных систем, создание нескольких *log*-файлов, работу как сервис или программа под *Windows NT* и запуск из *inetd* под *UNIX*, поддержку *Windows CGI* и *HTTP* (включая *HTTP PUT*), автоматический ответ при изменении документа, поддержку *Microsoft ISAPI*, возможность запрета доступа с конкретных адресов и к конкретным документам, запрет запуска *CGI* скриптов, настройку без перезапуска сервера, полный исходный код в поставке, поддержку других протоколов (*ftp*, *telnet*), настройку пользовательских директорий и наличие модуля *proxy*.

Данные от *Webcompare* (<http://webcompare.internet.com>) и *Swatch* (<http://serverwatch.internet.com>) показывают, что *Apache* занимает 58,85% рынка веб-серверов, имеет более 3,5 миллионов пользователей и получает высокие оценки за надежность, производительность, техническую поддержку и относительную простоту использования.

1.5 Разработка приложения

1.5.1 Разработка фронтенда

Визуальная часть сайта включает в себя все элементы, которые видит пользователь при посещении сайта: дизайн, цветовую схему, шрифты, изображения, видео, анимацию, кнопки, формы, меню навигации и т.д. Она важна для создания привлекательного и удобного интерфейса, который позволяет пользователям быстро и легко находить нужную информацию и выполнять действия на сайте.

Графический интерфейс должен предоставлять возможности для взаимодействия с сервисом в соответствии *UML* диаграммы использования представленной на рисунке 1.8.

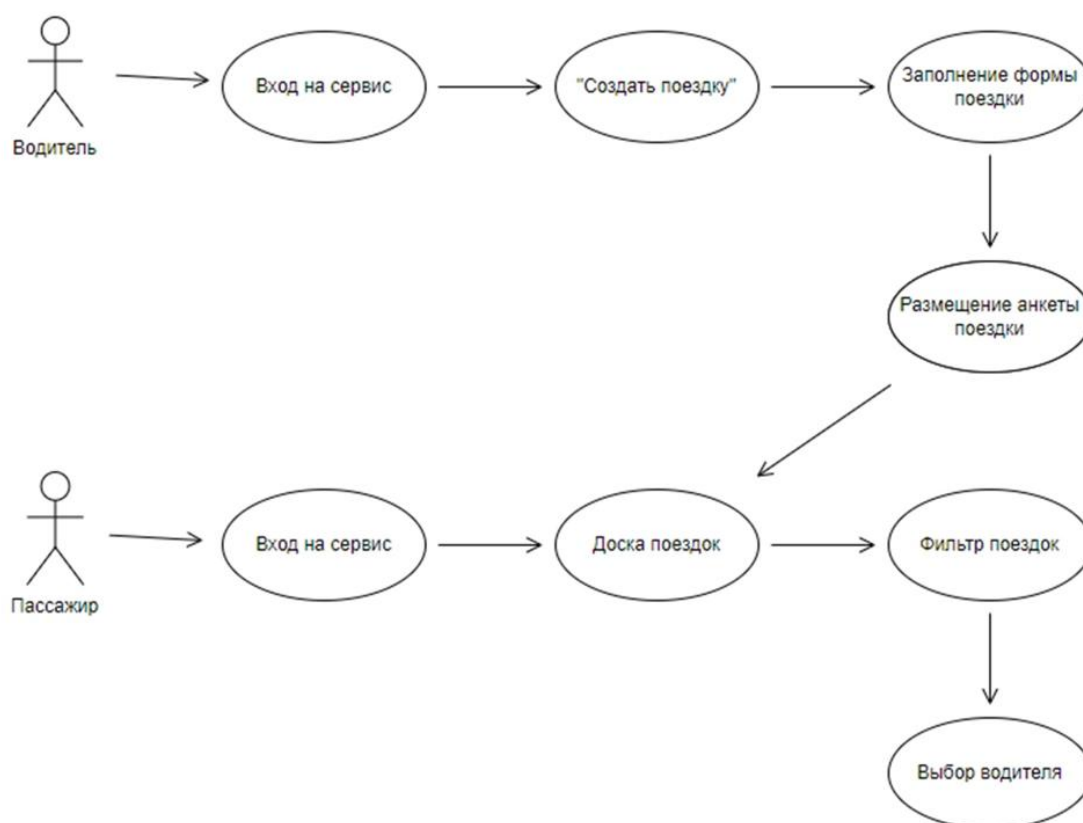


Рисунок 1.8– *UML* – диаграмма вариантов использования веб-сервиса

Для реализации визуальной части проекта были использованы *HTML* и *CSS*.

1.5.1.1 Разметка макета страниц

Для разметки страницы был использован *HTML*. *HTML* состоит из набора тегов, определяющих структуру и содержание веб-страницы. В проекте были использованы следующие теги:

- "<html>". Этот тег используется для определения начала и конца *HTML*-документа. Обычно он размещается в начале файла, а закрывающий тег — в конце файла;
- "<head>". Этот тег используется для определения раздела заголовка *HTML*-документа. Он содержит информацию о документе, такую как заголовок, метатеги и ссылки на внешние ресурсы;

- "<title>". Этот тег используется для определения заголовка веб-страницы. Он отображается в строке заголовка браузера, а также используется поисковыми системами для идентификации страницы;

- "<body>". Этот тег используется для определения основного содержимого веб-страницы. Он содержит все видимые элементы, такие как текст, изображения и видео. В нем и будет содержаться весь контент страниц;

- от "<h1>" до "<h6>". Эти теги используются для определения заголовков и подзаголовков. "<h1>" — самый большой заголовок, а "<h6>" самый маленький;

- "<p>". Этот тег используется для определения абзацев текста. Это наиболее часто используемый тег для текстового контента.

- "<a>". Этот тег используется для создания гиперссылок. Его можно использовать для ссылки на другие веб-страницы, адреса электронной почты или файлы на том же сервере;

- "". Этот тег используется для отображения изображений на веб-странице. Для этого требуется исходный атрибут, указывающий расположение файла изображения;

- "" и "". Эти теги используются для создания нумерованных списков. "" определяет список, а "" определяет каждый элемент в списке;

- "" и "". Эти теги используются для создания упорядоченных списков. "" определяет список, а "" определяет каждый элемент в списке;

- "<table>". Этот тег используется для создания таблиц на веб-странице. Для определения структуры таблицы требуется использование тегов "<tr>" (строка таблицы) и "<td>" (данные таблицы);

- "<form>". Этот тег используется для создания форм на веб-странице. Он позволяет пользователям вводить данные, такие как текст, флажки и переключатели;

- "<input>". Этот тег используется внутри тега "<form>" для создания полей ввода. Его можно использовать для создания текстовых полей, флажков, переключателей и кнопок отправки;

- "<select>" и "<option>". Эти теги используются внутри тега "<form>" для создания выпадающих меню. "<select>" определяет меню, а "<option>" определяет каждый элемент в меню;

- "<div>". Этот тег используется для создания разделов или подразделов на веб-странице. Он обычно используется для целей макета. Он является основой для построения качественного макета;

- "". Этот тег похож на "<div>", но используется для небольших разделов или подразделов на веб-странице;

- "
". Этот тег используется для создания разрывов строк в тексте. Он обычно используется для разделения абзацев;

- "<hr>". Этот тег используется для создания горизонтальных линий на веб-странице. Он обычно используется для разделения секций;

- "" и "". Эти теги используются для определения текста, который должен быть выделен. "" выделяет текст жирным шрифтом, а "" выделяет текст курсивом;

- "<meta>". Этот тег используется для предоставления метаданных о веб-странице, таких как ключевые слова, описания и информация об авторе. Он помещается в раздел "<head>" документа.

Это лишь некоторые из многих тегов *HTML*, доступных для использования веб-разработчиками. Правильное использование этих тегов необходимо для создания доступных, хорошо структурированных и удобных для поисковых систем веб-сайтов.

1.5.1.2 Разработка стилей страниц

В качестве языка оформления страницы был выбран CSS. В качестве тегов для оформления страницы были выбраны:

Теги селектора используются для выбора элементов *HTML* и применения к ним стилей. Существует несколько типов тегов селектора, в том числе:

- селектор элементов выбирает все экземпляры *HTML*-элемента и применяет к ним стили. Например, если необходимо стилизовать все абзацы на своем веб-сайте, нужно использовать селектор элементов следующим образом: *p {styles};*

- селектор *ID* выбирает элемент *HTML* по его атрибуту *ID* и применяет к нему стили. Селекторы идентификаторов уникальны и могут использоваться только один раз на странице. Например, если необходимо задать стиль определенному элементу *div*, нужно использовать селектор *ID* следующим образом: *#div {styles};*

- селектор класса выбирает элемент *HTML* по его атрибуту класса и применяет к нему стили. Селекторы классов можно использовать несколько раз на странице. Например, если необходимо стилизовать все элементы с помощью класса «кнопка», нужно использовать селектор класса следующим образом: *.button {styles}.*

Блочная модель - это концепция, используемая для описания макета *HTML*-элементов. Он состоит из четырех частей: поля, границы, заполнения и содержимого. CSS предлагает несколько тегов для управления размером, интервалом и внешним видом этих элементов:

- теги ширины и высоты используются для управления размером элемента. Например, если необходимо установить ширину изображения равной 500 пикселям, нужно использовать тег ширины следующим образом: *img { width: 500px; };*

- теги поля используются для создания пространства вокруг элемента. Например, если необходимо создать поле в 20 пикселей вокруг изображения, нужно использовать следующий тег поля: *img { margin: 20px; }*;

- теги границы используются для создания рамки вокруг элемента. Например, если необходимо создать сплошную черную рамку вокруг элемента *div*, нужно использовать тег границы следующим образом: *div { border: 1px solid black; }*;

- теги заполнения используются для создания пространства между содержимым элемента и его границей. Например, если необходимо создать отступ в 10 пикселей внутри элемента *div*, то нужно использовать следующий тег заполнения: *div { padding: 10px; }*.

Текстовые теги используются для управления внешним видом текста на веб-сайте. В CSS доступно несколько текстовых тегов, в том числе:

- тег *Font-family* используется для установки семейства шрифтов элемента. Например, если необходимо использовать шрифт *Arial* для всего текста на своем веб-сайте, то нужно использовать тег семейства шрифтов следующим образом: *body { font-family: Arial, sans-serif; }*;

- тег *Font-size* используется для установки размера шрифта. Например, если необходимо установить размер шрифта всего текста на веб-сайте равным 16 пикселям, то нужно использовать тег *font-size* следующим образом: *body { font-size: 16px; }*;

- тег *Text-align* используется для выравнивания текста внутри элемента. Например, если необходимо центрировать весь текст в элементе *div*, то нужно использовать тег *text-align* следующим образом: *div { text-align: center; }*;

- тег *Text-decoration* используется для украшения текста, например подчеркивания или зачеркивания. Например, если необходимо добавить подчеркивание ко всем ссылкам на вашем веб-сайте, то нужно использовать тег *text-decoration* следующим образом: *a { text-decoration: underline; }*.

Теги макета используются для управления макетом и расположением элементов на веб-сайте. В CSS доступно несколько тегов макета, в том числе:

- тег отображения используется для управления отображением элемента. Например, если необходимо отобразить элемент как элемент блока, то нужно использовать тег *display* следующим образом: *div { display: block; }*;

- плавающий тег используется для позиционирования элемента слева или справа от его контейнера. Например, если необходимо разместить изображение справа от элемента *div*, то нужно тег *float* следующим образом: *img { float: right; }*;

- тег положения используется для позиционирования элемента относительно его родительского элемента. Например, если необходимо расположить элемент *div* на расстоянии 50 пикселей от верхнего края родительского элемента, то нужно использовать тег *position* следующим образом: *div { position: relative; top: 50 px; }*.

Так же, помимо прямой отсылки на *html* тег, можно использовать классы. Классы задаются в начальном теге в качестве его продолжения, например: *<div class="line"></div>*, где “*line*” произвольное название класса. После создания класса можно к нему обратиться в *css* файле как *.line{}*.

На рисунке 1.9 показан итог верстки главной страницы веб-сервиса.



Рисунок 1.9 – Итог готовой страницы с использованием *html* и *css*

1.5.1.3 Разработка модального окна

1.5.1.3.1 Структура модального окна

Структура модального окна выглядит следующим образом:

- "`<div class="modal-content">`" - контейнер для содержимого модального окна;
- "`<span class="close">×</span>`" - кнопка закрытия модального окна, обозначена символом "×";
- "`<form class="box" id="formx" action="javascript:void(null);" onsubmit="call()" method="post">`" - форма, содержащая поля для заполнения информации о поездке. Атрибуты *id*, *action*, *onsubmit*, *method* определяют идентификатор формы, действия при отправке формы, функцию обратного вызова и метод отправки данных соответственно;

- внутри формы есть несколько блоков "`<div class='box1'>`", которые группируют связанные поля;
- в каждом блоке "`<div class='box1'>`" есть два подблока "`<div class='box1-1'>`" и "`<div class='box1-2'>`", которые содержат поля для ввода различных данных о поездке;
- последний блок "`<div class='social-menu'>`" содержит кнопку и выпадающее меню со ссылками на социальные сети для связи. При нажатии на кнопку отображается выпадающий список "`<div class='drop-content'>`", в котором содержатся ссылки и поля для ввода социальных сетей;
- "`<textarea id='form-textarea' type='text' name='details' maxlength='100' cols='30' rows='10' placeholder='Детали поездки'></textarea>`" - текстовое поле для ввода деталей поездки;
- "`<input type='submit'>`" - кнопка отправки формы;
- "`<div id='results'></div>`" - контейнер для отображения результатов отправки формы;

Данная разметка представляет только внешний вид модального окна и его элементов. Логика открытия, закрытия и обработки данных формы реализуется с помощью *JavaScript*.

1.5.1.3.2 Функциональность модального окна

JavaScript код на отвечает за функциональность модального окна. Он связывает модальное окно с определенными элементами на странице и определяет поведение при их взаимодействии.

- `var modal = document.getElementById('myModal');` - получает модальное окно по его идентификатору и сохраняет ссылку на него в переменной `modal`;
- `var btn1 = document.getElementById("myBtn1");` и `var btn2 = document.getElementById("myBtn2");` - получают кнопки, которые будут открывать модальное окно, по их идентификаторам и сохраняют ссылки на них в переменных `btn1` и `btn2`;

- `var span = document.getElementsByClassName('close')[0];` - получает ссылку на элемент с классом "close" (кнопка закрытия) и сохраняет ее в переменной span;

- `function OpenModal() { ... }` - определяет функцию *OpenModal*, которая будет вызываться при открытии модального окна. Внутри функции устанавливается свойство *display* модального окна на "block", что делает его видимым;

- `span.onclick = function() { ... }` - назначает обработчик события "click" на кнопку закрытия. При клике на кнопку устанавливается свойство *display* модального окна на "none", скрывая его;

- `window.onclick = function(event) { ... }` - назначает обработчик события "click" на окно браузера. При клике вне модального окна (на любой другой области страницы) проверяется, является ли цель события модальным окном. Если нет, то свойство *display* модального окна устанавливается на "none", скрывая его;

- `btn1.onclick = OpenModal;` и `btn2.onclick = OpenModal;` - назначают обработчики события "click" на кнопки *btn1* и *btn2*. При клике на эти кнопки будет вызываться функция *OpenModal*, открывая модальное окно.

Таким образом, при клике на кнопки *btn1* и *btn2* модальное окно будет отображаться, а при клике на кнопку закрытия или вне модального окна, оно будет скрываться.

1.5.2 Разработка бэкенда

1.5.2.1 Разработка Базы Данных

Для создания базы данных будем использовать приложение *Open Server*, в котором есть встроенный *PHPMyAdmin*:

- запускаем *OpenServer* и убеждаемся, что веб-сервер и *MySQL* сервер работают. Понять это можно по зеленой иконке флажка;

- открываем браузер и вводим в адресной строке «<http://localhost/phpmyadmin>». Это откроет *phpMyAdmin* в браузере.

- после загрузки страницы *phpMyAdmin* можно увидеть список баз данных в левой панели;

- в верхней части страницы нужно найти и нажать на вкладку "*Базы данных*". В некоторых версиях *OpenServer* она также может называться "*Базы данных MySQL*";

- на странице "*Базы данных*" находим раздел "*Создать базу данных*". Вводим желаемое имя базы данных в поле "*Название базы данных*" и выбираем кодировку и сортировку;

- нажимаем кнопку "*Создать*" или "*Create*", чтобы создать новую базу данных;

- после создания базы данных она появится в списке баз данных на странице "*Базы данных*".

Структура созданной базы данных выглядит следующим образом:

```
CREATE TABLE "poputchik2" (  
  "id" int NOT NULL,  
  "name" varchar(10) NOT NULL,  
  "number" varchar(20) NOT NULL,  
  "otkuda" varchar(20) NOT NULL,  
  "kuda" varchar(20) NOT NULL,  
  "date" date NOT NULL,  
  "time" varchar(6) NOT NULL,  
  "details" varchar(100) NOT NULL,  
  "price"      varchar(100)      CHARACTER SET utf8mb3 COLLATE  
utf8mb3_general_ci NOT NULL,  
  "place"      varchar(100)      CHARACTER SET utf8mb3 COLLATE  
utf8mb3_general_ci NOT NULL,  
  "vk"         varchar(100)      CHARACTER SET utf8mb3 COLLATE  
utf8mb3_general_ci NOT NULL,  
  "whatsapp"   varchar(100)      CHARACTER SET utf8mb3 COLLATE  
utf8mb3_general_ci NOT NULL,
```

```

"telegram"    varchar(100)    CHARACTER    SET    utf8mb3    COLLATE
utf8mb3_general_ci NOT NULL,
"odnoklassniki" varchar(100)    CHARACTER    SET    utf8mb3    COLLATE
utf8mb3_general_ci NOT NULL,
"viber"       varchar(100)    CHARACTER    SET    utf8mb3    COLLATE
utf8mb3_general_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

```

Таблица 1.6 – Структура таблицы MySQL

Имя	Тип	Сравнение
<i>id</i>	<i>int</i>	
<i>name</i>	<i>varchar(10)</i>	<i>utf8mb3_general_ci</i>
<i>number</i>	<i>varchar(20)</i>	<i>utf8mb3_general_ci</i>
<i>otkuda</i>	<i>varchar(20)</i>	<i>utf8mb3_general_ci</i>
<i>kuda</i>	<i>varchar(20)</i>	<i>utf8mb3_general_ci</i>
<i>date</i>	<i>data</i>	
<i>time</i>	<i>time</i>	
<i>details</i>	<i>varchar(100)</i>	<i>utf8mb3_general_ci</i>
<i>price</i>	<i>varchar(5)</i>	<i>utf8mb3_general_ci</i>
<i>place</i>	<i>varchar(10)</i>	<i>utf8mb3_general_ci</i>
<i>vk</i>	<i>varchar(100)</i>	<i>utf8mb3_general_ci</i>
<i>whatsapp</i>	<i>varchar(100)</i>	<i>utf8mb3_general_ci</i>
<i>telegram</i>	<i>varchar(100)</i>	<i>utf8mb3_general_ci</i>
<i>odnoklassniki</i>	<i>varchar(100)</i>	<i>utf8mb3_general_ci</i>
<i>viber</i>	<i>varchar(100)</i>	<i>utf8mb3_general_ci</i>

1.5.2.2 Отправка данных из модального окна на БД

Для того чтобы отправлять данные из модального окна на БД был использован *PHP* код.

Созданный код является скриптом на языке программирования *PHP*, который выполняет отправку данных из формы на базу данных *MySQL*. На веб-странице есть *HTML*-форма, из которой пользователь вводит данные, и после отправки формы данные обрабатываются этим скриптом.

Вначале задаются переменные, которые получают значения из массива *\$_POST*, содержащего данные, отправленные методом *POST* из *HTML*-формы.

Затем устанавливается соединение с базой данных *MySQL*, используя объект *mysqli*. Если соединение не удалось установить, выводится сообщение об ошибке.

После этого формируется *SQL*-запрос для вставки данных в таблицу *poputchik2* с использованием значениями переменных. Запрос выполняется с помощью метода *query* объекта *mysqli*.

Если запрос выполнен успешно, выводится сообщение "Данные отправлены!". В противном случае выводится сообщение об ошибке, содержащее текст запроса и сообщение об ошибке соединения.

Наконец, соединение с базой данных закрывается с помощью метода *close*.

Поступившие данные в *PHPMyAdmin* представлены на рисунке 5.6.

←→	id	name	number	otkuda	kuda	date	time	details	price	place	vk	whatsapp	telegram	odnoklassniki	viber
☐	50	Михаил	+7 (896) 980-52-54	Пенза	Сургут	2023-06-08	16:00	666	1800	4	https://vk.com/mnyrkov				

Рисунок 5.6 – Данные, отправленные из формы на БД

Поступившие данные на языке *MySQL* выглядят следующим образом:

```
"INSERT INTO "poputchik2" ("id", "name", "number", "otkuda",
"kuda", "date", "time", "details", "price", "place", "vk",
"whatsapp", "telegram", "odnoklassniki", "viber") VALUES (50,
'Михаил', '+7 (896) 980-52-54', 'Пенза', 'Сургут', '2023-06-08',
'16:00', '666', '1800', '4', 'https://vk.com/mnyrkov', '', '',
'', '');
```

1.5.2.3 Вывод содержимого из базы данных на страницу

Для вывода содержимого базы данных используется *PHP*. Скрипт будет отвечать за отображение информации из базы данных на веб-странице. Его можно разбить его по частям:

- открывающий тег `<?php` и начало цикла *while*: Этот код указывает, что следующий блок кода будет на языке *PHP*, и начинается цикл *while*, который будет выполняться до тех пор, пока в базе данных есть результаты для получения;

- блок кода внутри цикла: В этом блоке кода происходит отображение информации из базы данных. Каждая запись из базы данных отображается в виде блока `<div class="post">`, содержащего информацию о попутчике;

- поля информации: Каждое поле информации из базы данных, такое как имя, откуда, дата, цена и т.д., отображается в соответствующем блоке `<div>`. Значения полей берутся из массива *\$information*, который содержит данные текущей записи;

- дополнительные детали и ссылки на социальные сети: В конце блока кода отображаются дополнительные детали попутчика и ссылки на его профили в социальных сетях (*VK*, *WhatsApp*, *Telegram*, *Одноклассники*, *Viber*). Значения ссылок также берутся из массива *\$information*;

- закрывающий тег `<?php` и завершение цикла: Этот код завершает блок кода на языке *PHP* и закрывает цикл *while*.

1.5.2.4 Удаление устаревших данных из базы данных

Для удаления устаревших записей из базы данных был разработан скрипт на языке *PHP*, который работает следующим образом:

- подключение к базе данных: В этом коде устанавливается соединение с базой данных *MySQL*, используя параметры сервера (*\$servername*), имя пользователя (*\$username*), пароль (*\$password*) и имя базы данных (*\$dbname*). Функция *mysqli_connect()* создает соединение с базой данных;

- проверка соединения: С помощью условной конструкции *if* проверяется, удалось ли установить соединение с базой данных. Если соединение не удалось установить, то выводится сообщение об ошибке с помощью функции *die()*;

- удаление устаревших записей: В данном коде выполняется SQL-запрос для удаления устаревших записей из таблицы *poputchik2*. SQL-запрос *DELETE FROM poputchik2 WHERE date < CURRENT_DATE* удаляет все записи, у которых поле *date* имеет значение меньше текущей даты;

- выполнение SQL-запроса: С помощью функции *mysqli_query()* выполняется SQL-запрос на удаление устаревших записей. Результат выполнения запроса сохраняется в переменную *\$deleteDate_connect*.

Таким образом, данный код устанавливает соединение с базой данных, выполняет SQL-запрос на удаление устаревших записей и удаляет их из таблицы *poputchik2*. На рисунке 1.10 представлена UML – диаграмма классов.

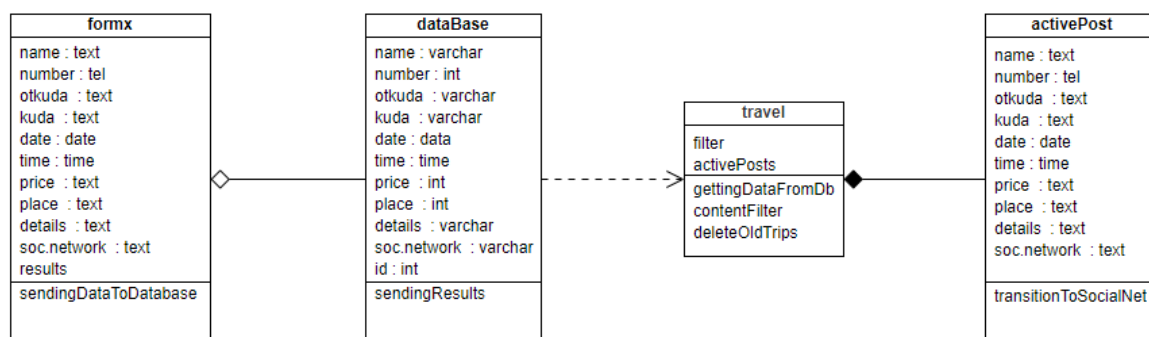


Рисунок 1.10– UML – диаграмма классов

1.5.2.5 Разработка поиска постов

Код поиска представляет собой обработчик события "*click*" для элемента с идентификатором "*search-btn*" внутри функции *\$(document).ready()*. Функция *\$(document).ready()* гарантирует, что код будет выполнен только после полной загрузки DOM-дерева страницы.

Когда происходит клик по элементу с идентификатором "search-btn", выполняется функция обратного вызова. Внутри нее происходит следующее:

- получается значение текстового поля с идентификатором "search-input" и приводится к нижнему регистру с помощью метода `.toLowerCase()`. Значение сохраняется в переменной `searchVal`;
- используя метод `.each()`, перебираются все элементы с классом "post" на странице;
- для каждого элемента, получается текстовое содержимое с помощью метода `.text()` и приводится к нижнему регистру. Значение сохраняется в переменной `postText`;
- с помощью метода `.indexOf(searchVal)` проверяется, содержит ли `postText` строку `searchVal`. Если вхождение не найдено (равно -1), то элемент скрывается с помощью метода `.hide()`. В противном случае, элемент отображается с помощью метода `.show()`.

Таким образом, код реализует поиск введенного пользователем значения в текстовых элементах с классом "post" и скрывает/отображает соответствующие элементы в зависимости от результатов поиска.

На рисунке 1.11 представлена UML-диаграмма последовательностей.

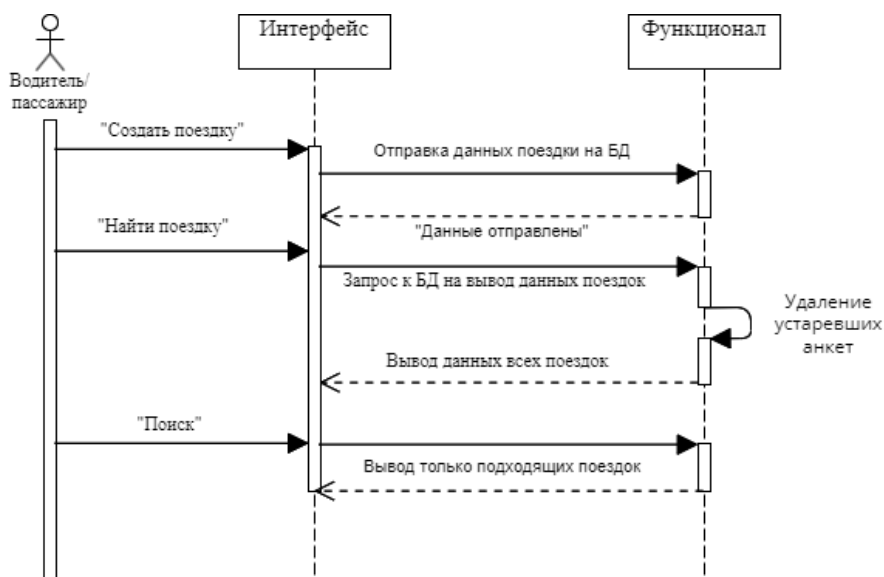


Рисунок 1.11 – UML – диаграмма последовательностей

1.5.3 Размещение сайта на хостинг

1.5.3.1 Что такое хостинг

Хостинг (или веб-хостинг) - это услуга предоставления места на сервере, где вы можете разместить свой веб-сайт, чтобы он был доступен для просмотра и взаимодействия через интернет. Хостинг-провайдеры предоставляют серверное пространство и ресурсы, необходимые для работы веб-сайта, такие как процессорное время, оперативная память, пропускная способность и хранилище данных.

При создании веб-сайта, все его файлы, включая *HTML*, *CSS*, *JavaScript*, изображения и другие медиафайлы, должны быть размещены на сервере, чтобы пользователи могли получить к ним доступ через интернет. Хостинг-провайдеры обеспечивают эту возможность, предоставляя серверное пространство и связанные услуги.

Основные аспекты хостинга включают:

- серверное пространство. Хостинг-провайдеры предоставляют сайту определенное количество дискового пространства на своих серверах, где владелец может хранить файлы своего веб-сайта;
- пропускная способность. Это относится к количеству данных, которые могут передаваться между веб-сайтом и его посетителями за определенный период времени. Выбор плана хостинга с соответствующей пропускной способностью важен для обеспечения доступности вашего сайта при большом количестве посетителей;
- доменное имя. Часто хостинг-провайдеры также предлагают услуги регистрации доменных имен, которые являются уникальными адресами веб-сайта в сети. Разработчик может зарегистрировать доменное имя и связать его с хостинг-аккаунтом;
- управление сервером. Хорошие хостинг-провайдеры обычно предоставляют панель управления, которая позволяет управлять файлами,

базами данных, почтовыми ящиками и другими настройками хостинг-аккаунта;

- безопасность и резервное копирование. Хостинг-провайдеры обычно предоставляют средства для обеспечения безопасности веб-сайта, такие как защита от вредоносных программ и резервное копирование данных, чтобы предотвратить потерю информации в случае сбоя.

Существуют различные типы хостинга, включая общий (*shared*) хостинг, виртуальный выделенный (*VPS*) хостинг, выделенный хостинг и облачный хостинг, каждый из которых имеет свои особенности и уровень ресурсов. Выбор подходящего типа хостинга зависит от требований к веб-сайту и ожидаемого трафика.

1.5.3.2 Размещение веб-сервиса на хостинг

В качестве хостинг-провайдера был выбран сервис *TimeWeb*. В общих чертах процесс установки выглядит следующим образом:

- регистрируемся на сайте *TimeWeb* и приобретаем хостинг-аккаунт, выбрав подходящий план, который поддерживает *MySQL*;

- получаем доступ к панели управления нашим хостинг-аккаунтом. *TimeWeb* предоставляет панель управления под названием "*TimeWeb Control Panel*";

- создаем базу данных *MySQL*. В панели управления находим раздел, связанный с базами данных или *MySQL*. Создаем новую базу данных, указав ее имя, пользователя и пароль для доступа к базе данных;

- настраиваем подключение к базе данных *MySQL*. Вначале импортируем нашу базу данных, а затем в каждом файле, который работает с базой данных, меняем имя пользователя, пароль и т.д.;

- проверка работы сайта. После завершения установки и настройки сайта и базы данных, убеждаемся, что все функционирует должным образом. Проверяем доступность сайта, перейдя по его *URL*-адресу. Убеждаемся, что

веб-страницы отображаются корректно и взаимодействуют с базой данных MySQL.

1.6 Описание веб-сервиса

1.6.1 Руководство пользователя

Приложение предлагает наличие таких функций, как:

- создание анкеты поездки в специальной форме;
- общая доска с уже созданными поездками;
- поиск нужной поездки через специальное поле.

На главной странице мы наблюдаем две кнопки: «Создать поездку» и «Найти поездку». На рисунке 1.12 представлена главная страница веб-сервиса.

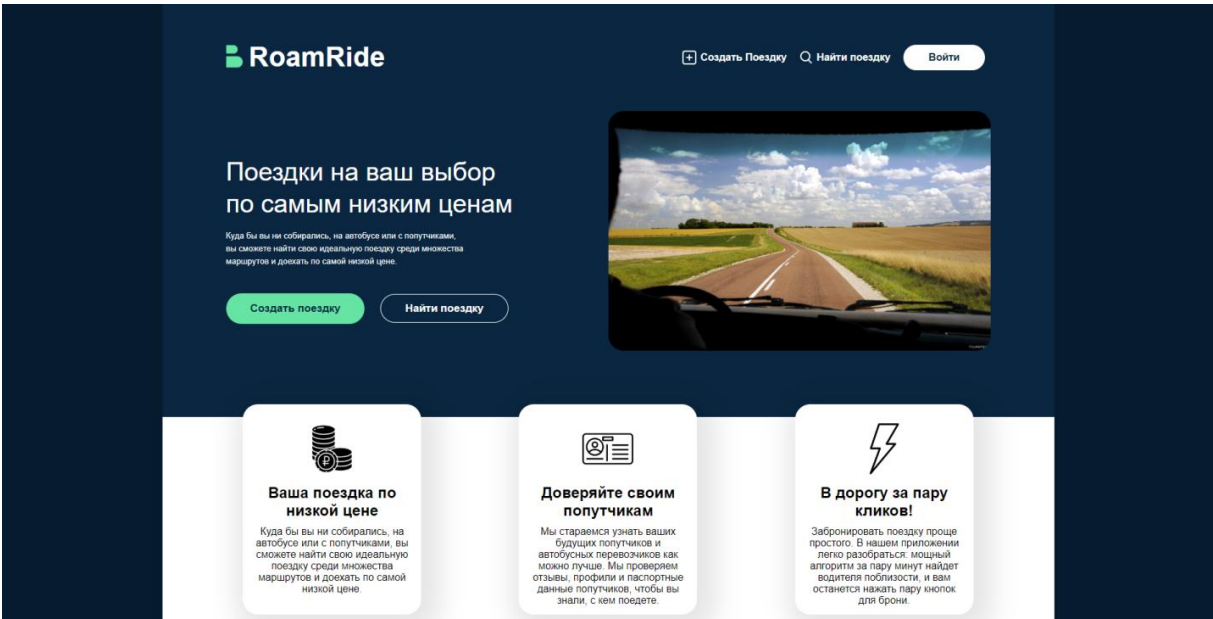


Рисунок 1.12 – Главная страница

Если пользователь является водителем, то при нажатии на кнопку «Создать поездку» перед ним всплывает модальное окно, в котором он может заполнить следующую форму. На рисунке 1.13 представлена форма создания поездки.

RoamRide

УКАЖИТЕ ДЕТАЛИ СОЗДАВАЕМОЙ ПОЕЗДКИ

Ваше имя: _____ Откуда: _____

Номер телефона: _____ Куда: _____

82, 86, 2823 _____ Цена одного билета: _____

--- 1 --- _____ Количество мест: _____

Ссылка на соц. сеть для связи ▼

Детали поездки

Отправить

Рисунок 1.13 – Форма создания поездки

Помимо стандартных пунктов по типу «Имя», «Откуда-куда» и др., водитель так же может указать ссылку на свою страницу в какой-либо социальной сети. На рисунке 1.14 представлена доска активных поездок.

RoamRide

УКАЖИТЕ ДЕТАЛИ СОЗДАВАЕМОЙ ПОЕЗДКИ

Артеи: _____ Пена: _____

+7 (896) 990-52-54 _____ Москва: _____

85, 86, 2823 _____ 1800 _____

20:00 _____ 2 _____

Ссылка на соц. сеть для связи ▼

Ваша поездка по низкой цене

В дорогу за пару кликов!

Рисунок 1.14 – Форма создания поездки и соц. сети водителя

После полного заполнения формы водитель нажимает кнопку «Отправить», после чего он может наблюдать под кнопкой надпись «Данные отправлены». На рисунке 1.15 представлен возврат результата отправки.

Рисунок 1.15 – Отправка данных

После этого данные поступают на БД, представленную на рисунке 1.16.

<div>←→</div>			id	name	number	otkuda	kuda	date	time	details	price	place	vk	whatsapp	telegram	odnoklassniki	viber
☐	<div><div>✎</div>Изменить</div>	<div><div>📄</div>Копировать</div>	<div><div>🗑</div>Удалить</div>	44	Пантелей	+7 (896) 980-52-54	Пенза	Саратов	2023-06-03	14:00	123	1200	2				
☐	<div><div>✎</div>Изменить</div>	<div><div>📄</div>Копировать</div>	<div><div>🗑</div>Удалить</div>	45	Пантелей	+7 (896) 980-52-54	Пенза	Саратов	2023-06-03	14:00	123	1200	2	https://vk.com/mnyrkov	https://vk.com/mnyrkov		https://vk.com/mnyrkov
☐	<div><div>✎</div>Изменить</div>	<div><div>📄</div>Копировать</div>	<div><div>🗑</div>Удалить</div>	50	Михаил	+7 (896) 980-52-54	Пенза	Сургут	2023-06-08	16:00	666	1800	4	https://vk.com/mnyrkov			
☐	<div><div>✎</div>Изменить</div>	<div><div>📄</div>Копировать</div>	<div><div>🗑</div>Удалить</div>	51	Артём	+7 (896) 980-52-54	Пенза	Саратов	2023-06-02	22:30	234	1000	2	https://vk.com/mnyrkov			
☐	<div><div>✎</div>Изменить</div>	<div><div>📄</div>Копировать</div>	<div><div>🗑</div>Удалить</div>	52	Михаил	+7 (896) 980-52-54	Екатеринбург	Томск	2023-06-05	11:40	11111	1500	3	https://vk.com/mnyrkov			
☐	<div><div>✎</div>Изменить</div>	<div><div>📄</div>Копировать</div>	<div><div>🗑</div>Удалить</div>	53	Артём	+7 (896) 980-52-54	Пенза	Москва	2023-06-05	20:00	Kway	1800	2	https://vk.com/mnyrkov			

Рисунок 1.16 – Таблица БД

А затем уже с БД на общую доску, где пассажир, ищущий водителя, может это посмотреть. На рисунке 1.17 представлена доска активных поездок.

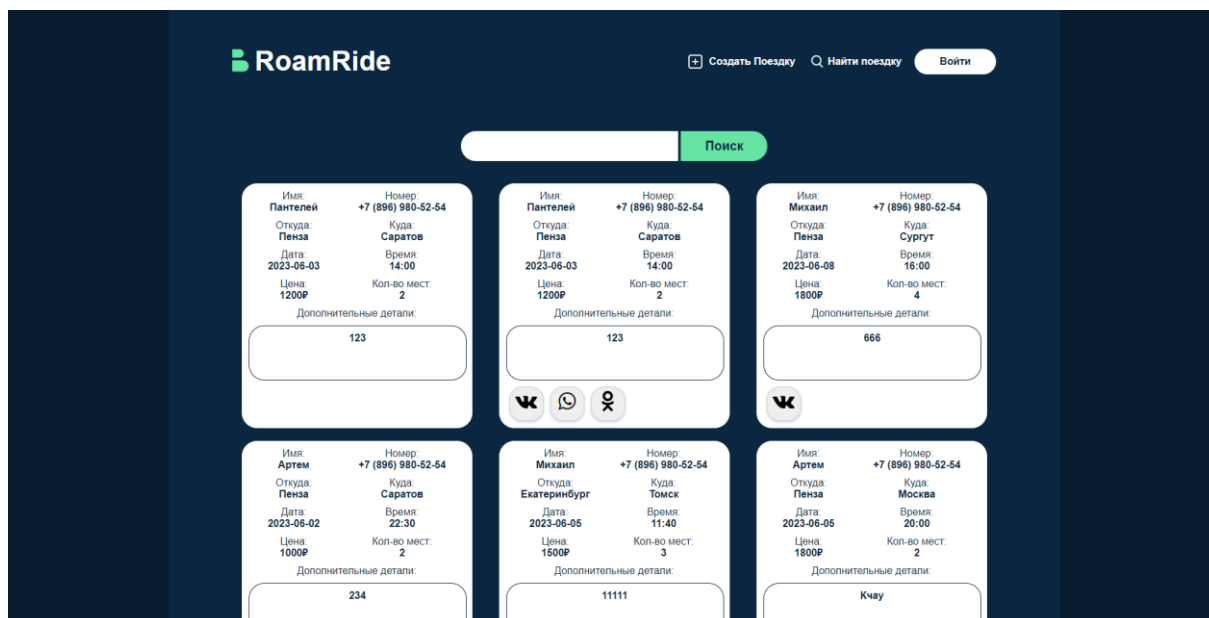


Рисунок 1.17 – Доска активных поездок

При необходимости можно воспользоваться фильтром и найти интересующую поездку. На рисунке 1.18 представлен фильтр поездок.

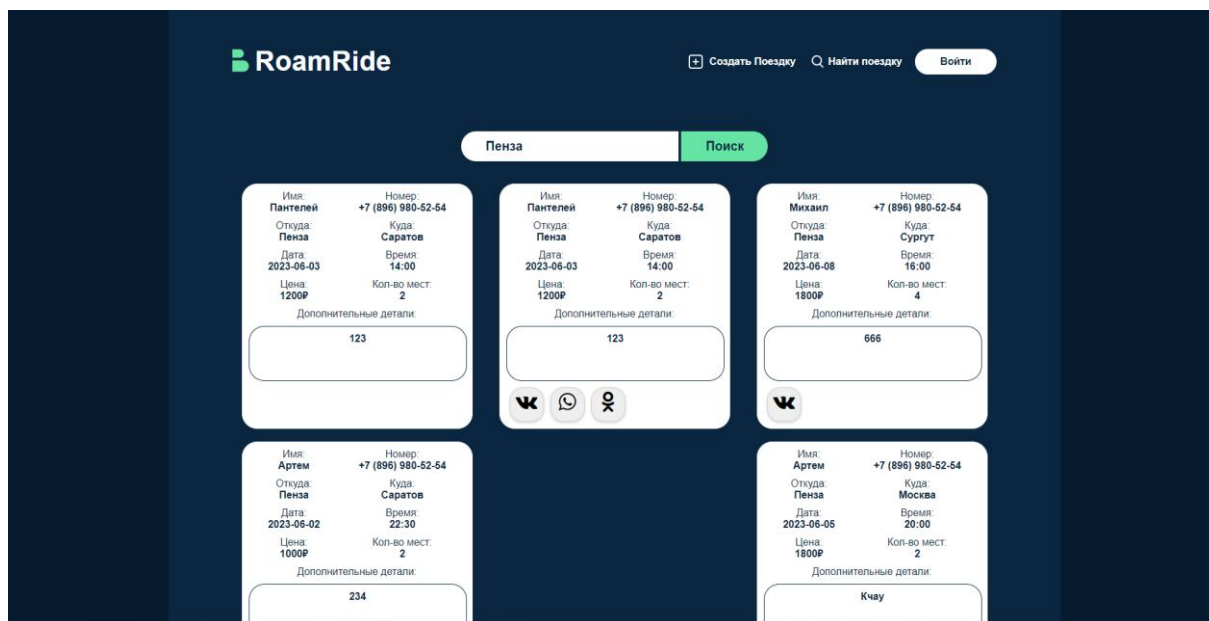


Рисунок 1.18 – Фильтр объявлений

После чего можно нажать по иконке под деталями поездки, тем самым перейдя на страницу водителя в социальной сети.

2 Безопасность жизнедеятельности

Современная жизнь человека тесно связана с информационными технологиями, но несмотря на множество преимуществ использования компьютера, есть и некоторые недостатки. Один из самых значительных недостатков заключается в его влиянии на здоровье. Компьютер может стать источником различных заболеваний, связанных с электромагнитным излучением, напряжением на глаза, дыхательными органами, а также с сидячим образом жизни, который негативно сказывается на мышцах и суставах. Вот некоторые факторы риска:

- повышенная температура некоторых компонентов компьютера и в результате повышенная температура в рабочем помещении;
- недостаточный уровень влажности воздуха на рабочем месте;
- высокая нагрузка на зрение и психоэмоциональное состояние;
- риск получения удара электрическим током или статическим электричеством;
- недостаточное освещение рабочей зоны;
- монотонность работы и другие факторы.

Для минимизации этих рисков существуют определенные требования к организации рабочего места:

- необходимо обеспечить естественное освещение, ориентированное на север или северо-запад, чтобы избежать солнечных бликов на экранах компьютеров, мешающих восприятию информации. Интенсивность солнечного света должна регулироваться с помощью штор, жалюзи или других устройств. В помещениях без естественного света разрешается организация рабочих мест только при наличии расчетов, доказывающих достаточное освещение таких комнат;
- рабочее место сотрудника, использующего персональный монитор, должно иметь площадь не менее 4,5 квадратных метров;

- при отделке помещений разрешается использовать только безопасные материалы с соответствующими санитарно-эпидемиологическими заключениями;

- необходимо поддерживать оптимальный микроклимат с нормальным уровнем влажности, температуры и скорости циркуляции воздуха. Также следует проводить ежедневную влажную уборку, контролировать содержание вредных веществ в воздухе, уровень шума, вибрацию и интенсивность электромагнитных полей;

- рабочее место должно быть оборудовано специальной мебелью, регулируемой по высоте и другим параметрам.

					ПГУ 1.090301.15.001 ПЗ	Лист
						49
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В ходе выполнения выпускной квалификационной работы был разработан веб-сервис по поиску попутчиков.

Разработанное веб-приложение предоставляет удобный веб-интерфейс для визуализации всей необходимой информации по поиску попутчиков.

Была разработана структура базы данных для последующего создания интернет магазина. Данная структура была реализована в формате реляционной СУБД.

Результатом работы веб-сервиса будет предоставление пользователю таблицы с информацией о попутчиках, подходящих для совместной поездки.

Одним из достоинств разработанного приложения является наличие удобного и интуитивно понятного для пользователя графического интерфейса.

Выполненные отладка и тестирование доказали работоспособность и корректность работы приложения.

В дальнейшем можно усовершенствовать разработанное веб-приложение и добавить возможность удаление анкеты поездки водителем, а так же возможность ее редактирования. Улучшить графический интерфейс, добавив самые популярные поездки и их стоимость, а так же добавить дополнительную информацию о сервисе. Так же можно улучшить поле поиска, добавив отдельные поля для каждой категории поездки и добавить фильтр по стоимости на требуемый пассажиру маршрут.

Список использованных источников

1. W3Schools: [Электронный ресурс]. URL: <https://www.w3schools.com/> (дата обращения: 26.04.2022)
2. Официальная документация PHP: [Электронный ресурс]. URL: <https://www.php.net/docs.php> (дата обращения: 30.04.2022)
3. Официальная документация MySQL: [Электронный ресурс]. URL: <https://dev.mysql.com/doc/> (дата обращения: 30.04.2022)
4. Дэвид Флэнаган. "JavaScript: Подробное руководство" Издательство "Питер", 2020.
5. Джон Дакетт. "HTML и CSS: Дизайн и создание веб-сайтов" Издательство "Вильямс", 2011.
6. Джон Дакетт. "JavaScript и JQuery: Интерактивная разработка веб-приложений" Издательство "Вильямс", 2014.
7. Ларри Ульман. "PHP и MySQL для создания динамических веб-сайтов" Издательство "Питер", 2017.
8. Люк Веллинг и Лаура Томсон. "Разработка веб-приложений на PHP и MySQL" Издательство "Аддисон-Вэсли", 2016.
9. Робин Никсон. "Изучаем PHP, MySQL и JavaScript: с примерами кода, CSS и HTML5" Издательство "Питер", 2014.
10. Эрик Мейер. "CSS: Подробное руководство" Издательство "Питер", 2017.

Приложение А

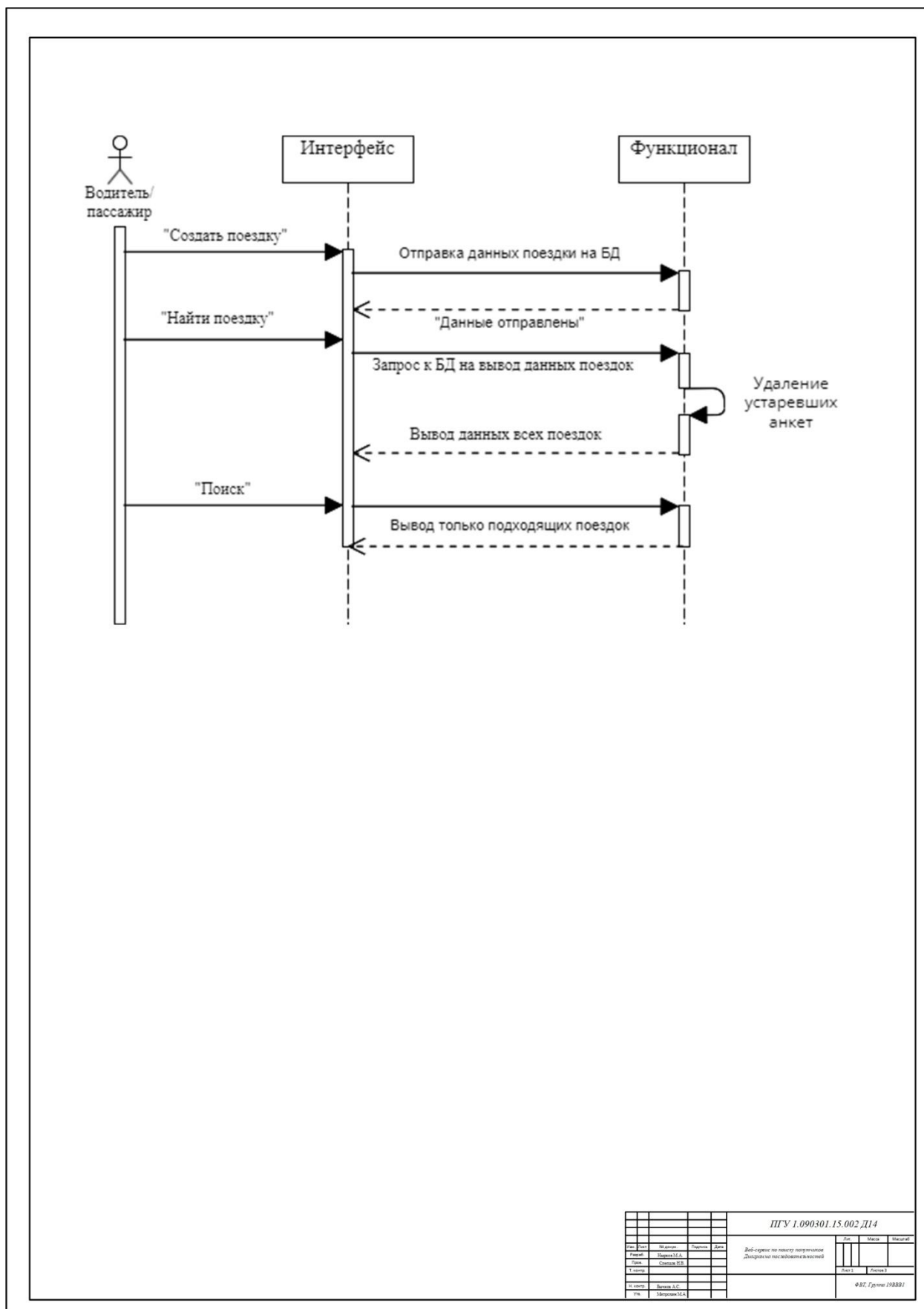
(обязательное)

Веб-сервис по поиску попутчиков

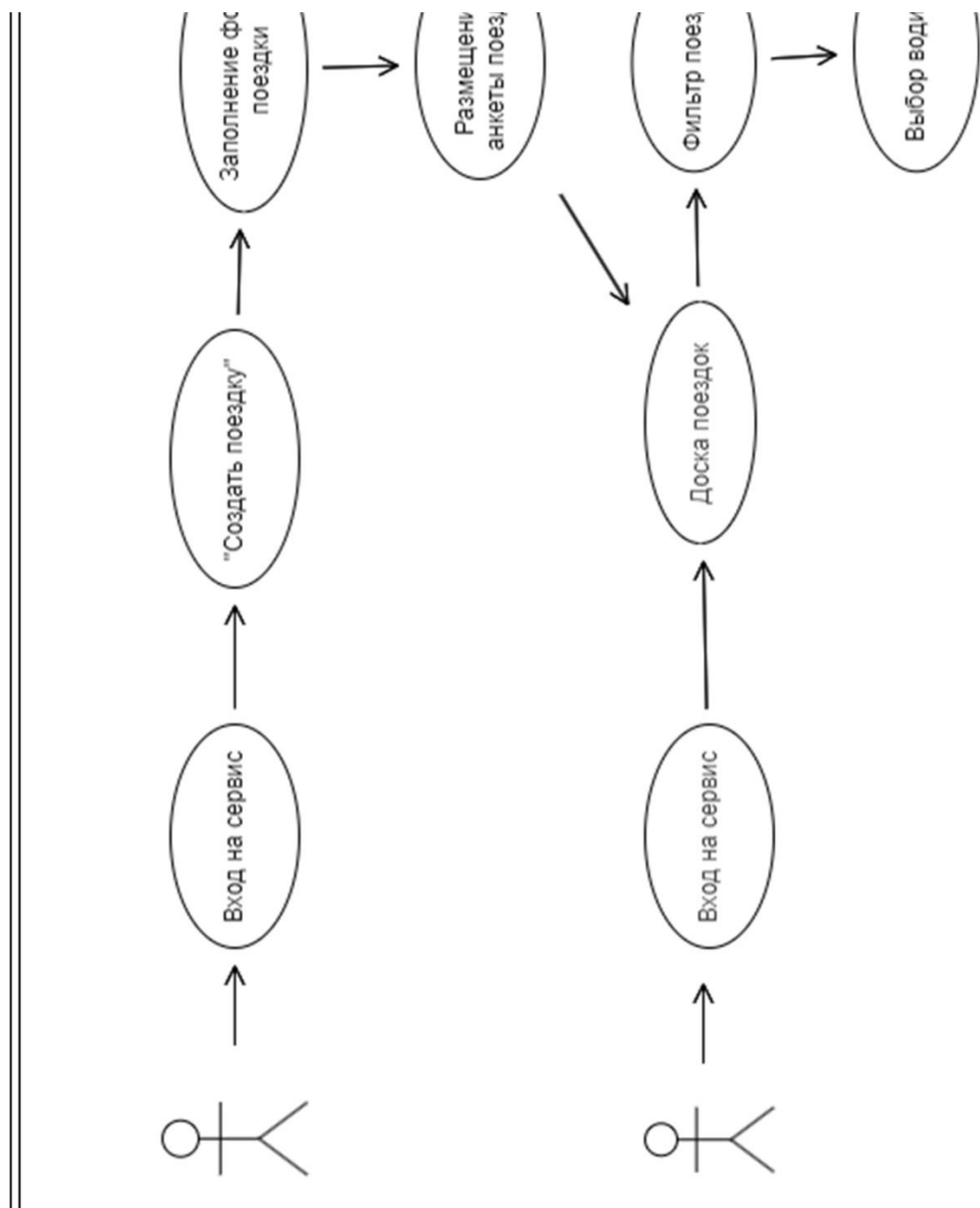
UML диаграммы

					ПГУ 1.090301.15.001 ПЗ	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

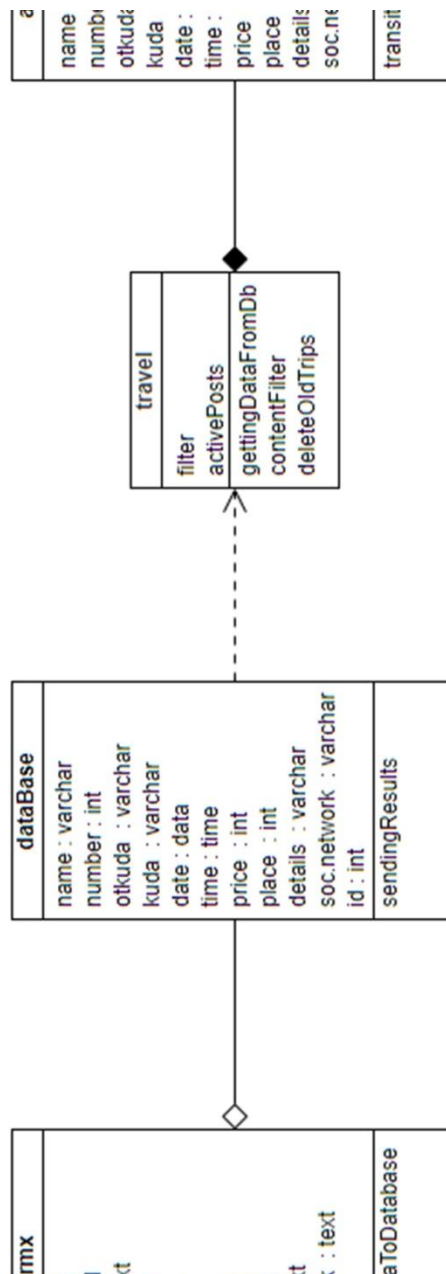
Приложение А.1 – UML – диаграмма последовательностей



Приложение А.2 – UML – диаграмма прецедентов



Приложение А.3 – UML – диаграмма классов



Приложение Б

(обязательное)

Веб-сервис по поиску попутчиков

Листинг программы

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ПГУ 1.090301.15.001 ПЗ

Лист

55

Файл «index.html»

```
<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <link rel="stylesheet" href="style.css">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com"
    crossorigin>

    <link
    href="https://fonts.googleapis.com/css2?family=Roboto:wght@4
    00;700&display=swap"

    rel="stylesheet">

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-
    awesome/4.7.0/css/font-awesome.css"

    rel="stylesheet">

    <title>Попутчик.PNZ</title>

</head>

<body>

    <div class="header">

        <div class="header-line">

            <a href="index.html" class="logo-up">

                <div class="header-logo">

                    <br>

                </div>

                <div class="logo-name">

                    <p>RoamRide</p>

                </div>

            </a>

        </div>

    </div>

</body>

</html>
```

```

<div class="nav">

<i class="gg-add-r"></i>

<button id="myBtn1" class="nav-item1">Создать
Поездку</button>

<i class="gg-search"></i>

<a href="search.php" class="nav-item2">Найти поездку</a>

<a href="#" class="log-in">Войти</a>

</div>

</div>

<div class="header-down">

<div>

<div class="header-title">

Поездки на ваш выбор <br> по самым низким ценам

</div>

<div class="header-title2">Куда бы вы ни собирались, на
автобусе или с попутчиками,

<br>вы сможете найти свою идеальную поездку среди множества

<br>маршрутов и доехать по самой низкой цене.

</div>

<div class="btn">

<button id="myBtn2" class="btn-create">Создать
поездку</button>

<a href="search.php" class="btn-search">Найти поездку</a>

</div>

</div>

<div>



</div>

</div>

<div id="myModal" class="modal">

<div class="modal-content">

```

```

<span class="close">&times;</span>

<form class="box" id="formx" action="javascript:void(null);"
onsubmit="call()" method="post">

<h1>Укажите детали создаваемой поездки</h1>

<div class="box1">

<div class="box1-1">

<input type="text" name="name" maxlength="10" required
placeholder="Ваше имя">

<input type="tel" name="number" maxlength="20" class="phone-
mask" required placeholder="Номер телефона">

</div>

<div class="box1-2">

<input type="text" name="otkuda" maxlength="20" required
placeholder="Откуда">

<input type="text" name="kuda" maxlength="20" required
placeholder="Куда">

</div>

</div>

<div class="box1">

<div class="box1-1">

<input type="date" name="date" id="davaToday" required>

<input type="time" name="time" required>

</div>

<div class="box1-2">

<input type="text" name="price" required placeholder="Цена
одного билета">

<input type="text" name="place" required
placeholder="Количество мест">

</div>

</div>

<div class="social-menu">

<button class="drop-btn">Ссылка на соц. сеть для связи
&#9660;</button>

```

```

<div class="drop-content">

<div class="box2">

<div class="box2-1">

<div class="social vk">

<a href="#"><i class="fa fa-vk fa-2x"></i></a>

<input type="text" name="vk">

</div>

<div class="social whatsapp">

<a href="#"><i class="fa fa-whatsapp fa-2x"></i></a>

<input type="text" name="whatsapp">

</div>

</div>

<div class="box2-2">

<div class="social telegram">

<a href="#"><i class="fa fa-paper-plane fa-2x"></i></a>

<input type="text" name="telegram">

</div>

<div class="social odnoklassniki">

<a href="#"><i class="fa fa-odnoklassniki fa-2x"></i></a>

<input type="text" name="odnoklassniki">

</div>

<div class="social viber" title="Ham Viber">

<a href="#">

<svg role="img" viewBox="0 0 512 512">

<path fill="currentColor" d="M444 49.9C431.3 38.2 379.9.9
265.3.4c0 0-135.1-8.1-200.9 52.3C27.8 89.3 14.9 143 13.5
209.5c-1.4 66.5-3.1 191.1 117 224.9h.1l-.1 51.6s-.8 20.9 13
25.1c16.6 5.2 26.4-10.7 42.3-27.8 8.7-9.4 20.7-23.2 29.8-
33.7 82.2 6.9 145.3-8.9 152.5-11.2 16.6-5.4 110.5-17.4
125.7-142 15.8-128.6-7.6-209.8-49.8-246.5zM457.9 287c-12.9
104-89 110.6-103 115.1-6 1.9-61.5 15.7-131.2 11.2 0 0-52
62.7-68.2 79-5.3 5.3-11.1 4.8-11-5.7 0-6.9.4-85.7.4-85.7-.1
0-.1 0 0 0-101.8-28.2-95.8-134.3-94.7-189.8 1.1-55.5 11.6-
101 42.6-131.6 55.7-50.5 170.4-43 170.4-43 96.9.4 143.3 29.6

```

154.1 39.4 35.7 30.6 53.9 103.8 40.6 211.1zm-139-80.8с.4
8.6-12.5 9.2-12.9.6-1.1-22-11.4-32.7-32.6-33.9-8.6-.5-7.8-
13.4.7-12.9 27.9 1.5 43.4 17.5 44.8 46.2zm20.3 11.3с1-42.4-
25.5-75.6-75.8-79.3-8.5-.6-7.6-13.5.9-12.9 58 4.2 88.9 44.1
87.8 92.5-.1 8.6-13.1 8.2-12.9-.3zm47 13.4с.1 8.6-12.9 8.7-
12.9.1-.6-81.5-54.9-125.9-120.8-126.4-8.5-.1-8.5-12.9 0-12.9
73.7.5 133 51.4 133.7 139.2zM374.9 329v.2с-10.8 19-31 40-
51.8 33.31-.2-.3с-21.1-5.9-70.8-31.5-102.2-56.5-16.2-12.8-
31-27.9-42.4-42.4-10.3-12.9-20.7-28.2-30.8-46.6-21.3-38.5-
26-55.7-26-55.7-6.7-20.8 14.2-41 33.3-51.8h.2с9.2-4.8 18-3.2
23.9 3.9 0 0 12.4 14.8 17.7 22.1 5 6.8 11.7 17.7 15.2 23.8
6.1 10.9 2.3 22-3.7 26.6l-12 9.6с-6.1 4.9-5.3 14-5.3 14s17.8
67.3 84.3 84.3с0 0 9.1.8 14-5.319.6-12с4.6-6 15.7-9.8 26.6-
3.7 14.7 8.3 33.4 21.2 45.8 32.9 7 5.7 8.6 14.4 3.8 23.6z"/>

</svg>

<input type="text" name="viber">

</div>

</div>

</div>

</div>

</div>

<textarea id="form-textarea" type="text" name="details"
maxlength="100" cols="30" rows="10" placeholder="Детали
поездки"></textarea>

<input type="submit">

<div id="results"></div>

</form>

</div>

</div>

</div>

<div class="cards">

<div class="cards-holder">

<div class="card">

<div class="card-image">


```

</div>

<div class="card-title">

Ваша поездка по низкой цене

</div>

<div class="card-desc">

Куда бы вы ни собирались, на автобусе или с попутчиками,
вы сможете найти свою идеальную поездку среди множества
маршрутов и доехать по самой низкой цене.

</div>

</div>

<div class="card">

<div class="card-image">



</div>

<div class="card-title">

Доверяйте своим попутчикам

</div>

<div class="card-desc">

Мы стараемся узнать ваших будущих попутчиков и автобусных
перевозчиков как можно лучше. Мы проверяем отзывы, профили
и паспортные данные попутчиков, чтобы вы знали, с кем
поедете.

</div>

</div>

<div class="card">

<div class="card-image">



</div>

<div class="card-title">

В дорогу за пару кликов!

```

```

</div>

<div class="card-desc">

Забронировать поездку проще простого. В нашем приложении
легко

разобраться: мощный алгоритм за пару минут найдет водителя
поблизости, и вам останется нажать пару кнопок для брони.

</div>

</div>

</div>

</div>

<script src="js/script.js"></script>

<script>document.getElementById('davaToday').valueAsDate =
new Date();</script>

<script type="text/javascript"
src="//code.jquery.com/jquery-1.8.3.min.js"></script>

<script type="text/javascript" language="javascript">

function call() {

var msg = $('#formx').serialize();

$.ajax({

type: 'post',

url: 'post.php',

data: msg,

success: function(data) {

$('#results').html(data);

},

error:function (xhr, str) {

alert('Возникла ошибка: ' + xhr.responseCode);

}

});

}

</script>

```

```

<script src="js/maska-nomera.js" type="text/javascript"
></script>

<script>

$('.phone-mask').mask('+7 (999) 999-99-99');

$.fn.setCursorPosition = function(pos) {
    if ($(this).get(0).setSelectionRange) {
        $(this).get(0).setSelectionRange(pos, pos);
    } else if ($(this).get(0).createTextRange) {
        var range = $(this).get(0).createTextRange();
        range.collapse(true);
        range.moveEnd('character', pos);
        range.moveStart('character', pos);
        range.select();
    }
};

$('input[type="tel"]').click(function(){
    $(this).setCursorPosition(4);
});

</script>

</body>

</html>

```

Файл «search.php»

```
<?php
include "post2.php";

$results = mysqli_query($connect, "SELECT * FROM "poputchik2");

?>

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <link rel="stylesheet" href="search-style.css">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com"
    crossorigin>

    <link
    href="https://fonts.googleapis.com/css2?family=Roboto:wght@4
    00;700&display=swap"

    rel="stylesheet">

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-
    awesome/4.7.0/css/font-awesome.css"

    rel="stylesheet">

    <title>Попутчик.PNZ</title>

</head>

<body>

    <div class="header">

        <div class="header-line">

            <a href="index.html" class="logo-up">

                <div class="header-logo">

                    <br>
```

```



</div>

<div class="logo-name">

<p>RoamRide</p>

</div>

</a>

<div class="nav">

<i class="gg-add-r"></i>

<a href="#" id="nav-item">Создать Поездку</a>

<i class="gg-search"></i>

<a href="#" id="nav-item">Найти поездку</a>

<a href="#" class="log-in">Войти</a>

</div>

</div>

<div class="posts-search">

<div class="search-content">

<input type="text" id="search-input" class="search-input">

<button id="search-btn" class="search-button">Поиск</button>

</div>

<div class="posts">

<?php

while($information = mysqli_fetch_assoc($results))

{

?>

<div class="post">

<div class="post-full-info">

```

```

<div class="info">

<div class="post-name">

Имя: <br> <p><?php echo $information['name']; ?></p>

</div>

<div class="post-otkuda">

Откуда: <br> <p><?php echo $information['otkuda']; ?></p>

</div>

<div class="post-date">

Дата: <br> <p><?php echo $information['date']; ?></p>

</div>

<div class="post-price">

Цена: <br> <p><?php echo $information['price']; ?>₽</p>

</div>

</div>

<div class="info">

<div class="post-number">

Номер: <br> <p><?php echo $information['number']; ?></p>

</div>

<div class="post-kuda">

Куда: <br> <p><?php echo $information['kuda']; ?></p>

</div>

<div class="post-time">

Время: <br> <p><?php echo $information['time']; ?></p>

</div>

<div class="post-place">

Кол-во мест: <br> <p><?php echo $information['place'];
?></p>

</div>

```

```

</div>

</div>

<div class="info">

Дополнительные детали:

<div class="post-details">

<p><?php echo $information['details']; ?></p>

</div>

<div class="social-network">

<div class="social vk">

<a href="<?php echo $information['vk']; ?>" target="_blank"
title="Ссылка на VK"><i class="fa fa-vk fa-2x"></i></a>

</div>

<div class="social whatsapp">

<a href="<?php echo $information['whatsapp']; ?>"
target="_blank" title="Ссылка на Whatsapp"><i class="fa fa-
whatsapp fa-2x"></i></a>

</div>

<div class="social telegram">

<a href="<?php echo $information['telegram']; ?>"
target="_blank" title="Ссылка на Telegram"><i class="fa fa-
paper-plane fa-2x"></i></a>

</div>

<div class="social odnoklassniki">

<a href="<?php echo $information['odnoklassniki']; ?>"
target="_blank" title="Ссылка на Одноклассники"><i class="fa
fa-odnoklassniki fa-2x"></i></a>

</div>

<div class="social viber" title="Нам Viber">

<a href="<?php echo $information['viber']; ?>"
target="_blank" title="Ссылка на Viber">

<svg role="img" viewBox="0 0 512 512">

```

```

<path fill="currentColor" d="M444 49.9C431.3 38.2 379.9.9
265.3.4c0 0-135.1-8.1-200.9 52.3C27.8 89.3 14.9 143 13.5
209.5c-1.4 66.5-3.1 191.1 117 224.9h.11-.1 51.6s-.8 20.9 13
25.1c16.6 5.2 26.4-10.7 42.3-27.8 8.7-9.4 20.7-23.2 29.8-
33.7 82.2 6.9 145.3-8.9 152.5-11.2 16.6-5.4 110.5-17.4
125.7-142 15.8-128.6-7.6-209.8-49.8-246.5zM457.9 287c-12.9
104-89 110.6-103 115.1-6 1.9-61.5 15.7-131.2 11.2 0 0-52
62.7-68.2 79-5.3 5.3-11.1 4.8-11-5.7 0-6.9.4-85.7.4-85.7-.1
0-.1 0 0 0-101.8-28.2-95.8-134.3-94.7-189.8 1.1-55.5 11.6-
101 42.6-131.6 55.7-50.5 170.4-43 170.4-43 96.9.4 143.3 29.6
154.1 39.4 35.7 30.6 53.9 103.8 40.6 211.1zm-139-80.8c.4
8.6-12.5 9.2-12.9.6-1.1-22-11.4-32.7-32.6-33.9-8.6-.5-7.8-
13.4.7-12.9 27.9 1.5 43.4 17.5 44.8 46.2zm20.3 11.3c1-42.4-
25.5-75.6-75.8-79.3-8.5-.6-7.6-13.5.9-12.9 58 4.2 88.9 44.1
87.8 92.5-.1 8.6-13.1 8.2-12.9-.3zm47 13.4c.1 8.6-12.9 8.7-
12.9.1-.6-81.5-54.9-125.9-120.8-126.4-8.5-.1-8.5-12.9 0-12.9
73.7.5 133 51.4 133.7 139.2zM374.9 329v.2c-10.8 19-31 40-
51.8 33.31-.2-.3c-21.1-5.9-70.8-31.5-102.2-56.5-16.2-12.8-
31-27.9-42.4-42.4-10.3-12.9-20.7-28.2-30.8-46.6-21.3-38.5-
26-55.7-26-55.7-6.7-20.8 14.2-41 33.3-51.8h.2c9.2-4.8 18-3.2
23.9 3.9 0 0 12.4 14.8 17.7 22.1 5 6.8 11.7 17.7 15.2 23.8
6.1 10.9 2.3 22-3.7 26.61-12 9.6c-6.1 4.9-5.3 14-5.3 14s17.8
67.3 84.3 84.3c0 0 9.1.8 14-5.319.6-12c4.6-6 15.7-9.8 26.6-
3.7 14.7 8.3 33.4 21.2 45.8 32.9 7 5.7 8.6 14.4 3.8 23.6z"/>

</svg>

</a>

</div>

</div>

</div>

</div>

</div>

<?php
}

?>

</div>

</div>

</div>

```

```

<script type="text/javascript"
src="//code.jquery.com/jquery-1.8.3.min.js"></script>

<script src="js/search.js" type="text/javascript"></script>

<style>

.social.vk:has(a[href=""]),
.social.whatsapp:has(a[href=""]),

.social.telegram:has(a[href=""]),.social.odnoklassniki:has(a
[href=""]),

.social.viber:has(a[href=""]){

display: none;

}

</style>

</body>

</html>

```

					ПГУ 1.090301.15.001 ПЗ	Лист
						69
Изм.	Лист	№ докум.	Подпись	Дата		

Файл «style.css»

```
body{
    margin: 0;
    font-family: 'Monrope', sans-serif;
    background-color: #071b2e;
}
p{
    margin: 0;
    padding: 0;
}
.header{
    background-color: #0A2640;
    width: 1400px;
    height: auto;
    margin: 0px auto;
    color: white;
}
.logo-up{
    text-decoration: none;
    display: flex;
    align-items: center;
    color: white;
}
.header-line{
    margin: 0px 100px;
    padding: 60px 0px;
    display: flex;
```

					ПГУ 1.090301.15.001 ПЗ	Лист
						70
Изм.	Лист	№ докум.	Подпись	Дата		

```

        align-items: center;
    }

    .logo-name{
        font-weight: 700;
        font-size: 43.25px;
        margin-left: 10px;
    }

    .nav{
        display: flex;
        align-items: center;
        padding-left: 467px;
    }

    .nav-item1{
        text-decoration: none;
        color: white;
        font-weight: 600;
        font-size: 16px;
        cursor: pointer;
        transition: color 0.3s linear;
        display: inline;
        background: none;
        border: none;
    }

    .nav-item2{
        text-decoration: none;
        color: white;

```

```

    font-weight: 600;

    font-size: 16px;

    margin: 0px 10px;

    transition: color 0.3s linear;

    display: inline;
}

.log-in{

    padding: 8px 40px;

    text-decoration: none;

    font-weight: 700;

    font-size: 16px;

    line-height: 24px;

    margin-left: 10px;

    color: #0A2640;

    background-color: white;

    border-radius: 24px;

    transition: background-color 0.3s linear;
}

.nav-item2:hover,.nav-item1:hover{

    color: #65E4A3;
}

.log-in:hover{

    background-color: #65E4A3;
}

.gg-add-r {

    box-sizing: border-box;

    position: relative;

```

```

display: block;

width: 22px;

height: 22px;

border: 2px solid;

transform: scale(var(--ggs,1));

border-radius: 4px
}

.gg-add-r::after,
.gg-add-r::before {
    content: "";
    display: block;
    box-sizing: border-box;
    position: absolute;
    width: 10px;
    height: 2px;
    background: currentColor;
    border-radius: 5px;
    top: 8px;
    left: 4px
}

.gg-add-r::after {
    width: 2px;
    height: 10px;
    top: 4px;
    left: 8px
}

}

.gg-search {

```

```

    box-sizing: border-box;

    position: relative;

    display: block;

    transform: scale(var(--ggs,1));

    width: 16px;

    height: 16px;

    border: 2px solid;

    border-radius: 100%;

    margin-left: -4px;

    margin-top: -4px
}

.gg-search{

    margin-left: 15px;

}

.gg-search::after {

    content: "";

    display: block;

    box-sizing: border-box;

    position: absolute;

    border-radius: 3px;

    width: 2px;

    height: 8px;

    background: currentColor;

    transform: rotate(-45deg);

    top: 10px;

    left: 12px

}

```

```

.header-down{
    margin: 0px 100px;
    padding-bottom: 100px;
    display: flex;
    justify-content: space-between;
}

.auto-img{
    width: 600px;
    height: auto;
    border-radius: 24px;
}

.header-title{
    font-weight: 400;
    font-size: 40px;
    line-height: 50px;
    padding-bottom: 16px;
    padding-top: 70px;
}

.header-title2{
    font-weight: 400;
    font-size: 13px;
    line-height: 20px;
}

.btn{
    margin-top: 40px;
}

.btn-create{

```

```

padding: 12px 38px;

text-decoration: none;

font-weight: 700;

font-size: 17px;

line-height: 24px;

color: #0A2640;

background-color: #65E4A3;

border-radius: 24px;

transition: background-color 0.5s linear;

margin-right: 20px;

cursor: pointer;

border: none;

}

.btn-create:hover{

    background-color: #0DBBFC;

}

.btn-search{

    padding: 12px 38px;

    text-decoration: none;

    font-weight: 700;

    font-size: 17px;

    line-height: 24px;

    color: white;

    border-radius: 24px;

    border: 1px solid;

    transition: background-color 0.5s linear;

    transition: color 0.3s linear;

```

```

}

.btn-search:hover{

    background-color: #0DBBFC;

    color: #0A2640;

    border: none;

}

.cards-holder{

    display: flex;

    justify-content: space-around;

    background-color: white;

    padding: 20px 50px;

}

.cards{

    width: 1400px;

    margin: 0px auto;

}

.card{

    width: 250px;

    height: 300px;

    box-shadow: 10px 10px 40px 20px #0000001A;

    padding: 15px;

    margin: -40px 0px 0px 0px;

    border-radius: 24px;

    background-color: white;

}

.card-image{

    display: flex;

```

```

        justify-content: center;

        margin: 15px 0px;
    }

    .card-title{

        display: flex;

        text-align: center;

        font-weight: 700;

        font-size: 24px;

        margin-bottom: 10px;
    }

    .card-desc{

        display: flex;

        text-align: center;

        font-weight: 400;

        font-size: 16px;
    }

    .card-img{

        width: 80px;

        height: 80px;
    }

    .modal{

        display: none;

        position: fixed;

        z-index: 1;

        left: 0;

        top: 0;

        width: 100%;
    }

```

```

        height: 100%;

        overflow: auto;

        background-color: rgba(0, 0, 0, 0.8);
    }

    .modal-content{

        background-color: #071b2e;

        margin: 3% auto;

        padding: 5px;

        width: 600px;

        color: #0A2640;

        border-radius: 24px;
    }

    .close{

        color: #aaa;

        float: right;

        font-size: 28px;

        font-weight: bold;

        cursor: pointer;

        padding: 15px;
    }

    .box{

        padding: 15px;

        top: 50%;

        left: 50%;

        color: #0A2640;

        background-color: #0A2640;

        text-align: center;
    }

```

```

        border-radius: 24px;
    }
    .box1, .box2{
        justify-content: space-between;
    }
    .box h1{
        color: white;
        text-transform: uppercase;
        font-weight: 500;
    }
    .box1 input[type = "text"],.box input[type = "tel"],
    .box input[type = "date"],.box input[type = "time"],
    .box input[type = "number"]{
        border:0;
        background-color: none;
        font-size: 14px;
        /*display: block;*/
        margin: 10px auto;
        text-align: center;
        border: 2px solid black;
        padding: 14px 10px;
        width: 250px;
        outline: none;
        color: #0A2640;
        border-radius: 24px;
        transition: 0.25s;
        height: 20px;
    }

```

```

}

.box textarea[type = "text"]{

    border:0;

    background-color: none;

    font-size: 14px;

    /*display: block;*/

    margin: 10px auto;

    text-align: center;

    border: 2px solid black;

    padding: 14px 10px;

    width: 530px;

    outline: none;

    color: #0A2640;

    border-radius: 24px;

    transition: 0.25s;

    height: 200px;

}

/*.box1 input[type = "text"]:focus,

.box input[type = "tel"]:focus,

.box textarea[type = "text"]:focus,

.box input[type = "tel"]:focus

{

    width: 320px;

    border-color: #65E4A3;

}*/

.box input[type = "submit"]{

    border:0;

```

```

background-color: none;

font-size: 16px;

font-weight: 600;

display: block;

margin: 10px auto;

text-align: center;

border: 2px solid #65E4A3;

padding: 14px 10px;

width: 200px;

outline: none;

color: #0A2640;

border-radius: 24px;

transition: 0.25s;

cursor: pointer;

}

.box input[type = "submit"]:hover{

    background-color: #65E4A3;

}

.box label{

    color: white;

    font-size: 20px;

    font-weight: 500;

}

.box1{

    display: flex;

}

#results{

```

```

        color: white;
    }
    .drop-btn {
        background-color: #65E4A3;
        color: #0A2640;
        padding: 16px;
        font-size: 16px;
        border: none;
        border-radius: 24px;
        font-size: 16px;
        font-weight: 600;
        margin: 10px 0;
    }
    .social-menu {
        position: relative;
        display: inline-block;
    }
    .drop-content {
        display: none;
    }
    .drop-content a:hover {background-color: #ddd;}
    .social-menu:hover .drop-content {display: block;}
    .social-menu:hover .drop-btn {background-color: #4dac7b;}
    .drop-content input{
        border: 0;
        font-size: 14px;
        display: block;
    }

```

```

margin: 5px auto;

text-align: center;

border: 2px solid black;

padding: 14px 10px;

width: 250px;

outline: none;

color: #0A2640;

border-radius: 0 24px 24px 0;

transition: 0.25s;

height: 20px;

}

@import url(https://maxcdn.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css);

.social a {

text-align: center;

width: 37px;

height: 34px;

float: left;

background: #fff;

border: 1px solid #ccc;

box-shadow: 0 2px 4px rgba(0,0,0,0.15), inset 0 0 50px
rgba(0,0,0,0.1);

border-radius: 24px 0 0 24px;

margin: 7px 0 0 0;

padding: 6px;

color: #000;

}

.vk a:hover {

```

```

        background: #5d84ae;

        color: #fff;
    }

    .whatsapp a:hover {

        background: #50b154;

        color: #fff;
    }

    .telegram a:hover {

        background: #249bd7;

        color: #fff;
    }

    .viber a:hover{

        background: #7b519d;

        color: #fff
    }

    .odnoklassniki a:hover {

        background: #f93;

        color: #fff;
    }

    .social-network{

        margin-top: 10px;
    }

    .social{

        display: inline-block;
    }

```

Файл «search-style.css»

```
body{
    margin: 0;
    font-family: 'Monrope', sans-serif;
    background-color: #071b2e;
}
p{
    margin: 0;
    padding: 0;
}
.header{
    background-color: #0A2640;
    width: 1400px;
    height: 100%;
    margin: 0px auto;
    color: white;
}
.logo-up{
    text-decoration: none;
    display: flex;
    align-items: center;
    color: white;
}
.header-line{
    margin: 0px 100px;
    padding: 60px 0px;
    display: flex;
```

					ПГУ 1.090301.15.001 ПЗ	Лист
						86
Изм.	Лист	№ докум.	Подпись	Дата		

```

        align-items: center;
    }
    .logo-name{
        font-weight: 700;
        font-size: 43.25px;
        margin-left: 10px;
    }
    .nav{
        display: flex;
        align-items: center;
        padding-left: 467px;
    }
    #nav-item{
        text-decoration: none;
        color: white;
        font-weight: 600;
        font-size: 16px;
        margin: 0px 10px;
        transition: color 0.3s linear;
    }
    .log-in{
        padding: 8px 40px;
        text-decoration: none;
        font-weight: 700;
        font-size: 16px;
        line-height: 24px;
        margin-left: 10px;
    }

```

```

        color: #0A2640;

        background-color: white;

        border-radius: 24px;

        transition: background-color 0.3s linear;
    }

    #nav-item:hover{

        color: #65E4A3;

    }

    .log-in:hover{

        background-color: #65E4A3;

    }

    .gg-add-r {

        box-sizing: border-box;

        position: relative;

        display: block;

        width: 22px;

        height: 22px;

        border: 2px solid;

        transform: scale(var(--ggs,1));

        border-radius: 4px

    }

    .gg-add-r::after,
    .gg-add-r::before {

        content: "";

        display: block;

        box-sizing: border-box;

        position: absolute;

```

```

width: 10px;

height: 2px;

background: currentColor;

border-radius: 5px;

top: 8px;

left: 4px
}

.gg-add-r::after {

width: 2px;

height: 10px;

top: 4px;

left: 8px

}

.gg-search {

box-sizing: border-box;

position: relative;

display: block;

transform: scale(var(--ggs,1));

width: 16px;

height: 16px;

border: 2px solid;

border-radius: 100%;

margin-left: -4px;

margin-top: -4px

}

.gg-search{

margin-left: 15px;

```

```

}

.gg-search::after {
    content: "";
    display: block;
    box-sizing: border-box;
    position: absolute;
    border-radius: 3px;
    width: 2px;
    height: 8px;
    background: currentColor;
    transform: rotate(-45deg);
    top: 10px;
    left: 12px
}

.hide{
    display: none;
}

.search{
    text-align: center;
}

.posts{
    display: flex;
    flex-wrap: wrap;
    margin: 0 100px;
    justify-content: space-between;
}

```

```

.post{
    background-color: white;
    border: 1px solid white;
    border-radius: 24px;
    margin: 10px 15px;
    padding: 10px;
    color: #0A2640;
    width: 340px;
    text-align: center;
}

.post-full-info{
    display: flex;
    justify-content: space-around;
}

.post-name, .post-number,
.post-otkuda, .post-kuda,
.post-date, .post-time,
.post-price, .post-place{
    margin-bottom: 10px;
}

.post-details{
    width: 340px;
    height: 84px;
    border: 1px solid #0A2640;
    border-radius: 24px;
    margin-top: 10px;
}

```

```

.post-details p{
    margin-top: 10px;
}

.info p {
    font-weight: 600;
    font-size: 16px;
}

.h1{
    text-align: center;
    font-size: 35px;
}

.search-content{
    width: 1400px;
    text-align: center;
    position: sticky;
    top: 0;
    padding: 25px 0;
    background-color: #0A2640;
}

.search-input{
    padding: 12px 38px;
    text-decoration: none;
    font-weight: 700;
    font-size: 20px;
    color: #0A2640;
    background-color: white;
    border-radius: 24px 0 0 24px;
}

```

```

        border: none;
    }

    .search-button{
        padding: 12px 38px;
        text-decoration: none;
        font-weight: 700;
        font-size: 20px;
        color: #0A2640;
        background-color: #65E4A3;
        border-radius: 0 24px 24px 0;
        transition: background-color 0.5s linear;
        cursor: pointer;
        border: none;
    }

    .search-button:hover{
        background-color: #0DBBFC;
    }

    @import url(https://maxcdn.bootstrapcdn.com/font-
    awesome/4.7.0/css/font-awesome.min.css);

    .social a {
        text-align: center;
        width: 40px;
        height: 40px;
        float: left;
        background: #fff;
        border: 1px solid #ccc;
        box-shadow: 0 2px 4px rgba(0,0,0,0.15), inset 0 0 50px
        rgba(0,0,0,0.1);
    }

```

```

border-radius: 20px;

margin: 0 5px 0 5px;

padding: 6px;

color: #000;

}

.vk a:hover {

background: #5d84ae;

color: #fff;

}

.whatsapp a:hover {

background: #50b154;

color: #fff;

}

.telegram a:hover {

background: #249bd7;

color: #fff;

}

.viber a:hover{

background: #7b519d;

color: #fff

}

.odnoklassniki a:hover {

background: #f93;

color: #fff;

}

.social-network{

margin-top: 10px;}

```

Файл «script.js»

```
var modal = document.getElementById('myModal');  
var btn1 = document.getElementById("myBtn1");  
var btn2 = document.getElementById("myBtn2");  
var span = document.getElementsByClassName('close')[0];  
  
function OpenModal() {  
    modal.style.display = "block";  
}  
span.onclick = function(){  
    modal.style.display = "none";  
}  
window.onclick = function(event){  
    if(event.target == modal){  
        modal.style.display = "none";  
    }  
}  
  
btn1.onclick = OpenModal;  
btn2.onclick = OpenModal;
```

Файл «search.js»

```
$(document).ready(function() {  
    $("#search-btn").click(function() {  
        var searchVal = $("#search-input").val().toLowerCase();  
        $(".post").each(function() {  
            var postText = $(this).text().toLowerCase();  
            if (postText.indexOf(searchVal) == -1) {  
                $(this).hide();  
            } else {  
                $(this).show();  
            }  
        });  
    });  
});
```

					ПГУ 1.090301.15.001 ПЗ	Лист
						96
Изм.	Лист	№ докум.	Подпись	Дата		

Файл «*post.php*»

```
<?php

$name = $_POST['name'];

$number = $_POST['number'];

$otkuda = $_POST['otkuda'];

$kuda = $_POST['kuda'];

$date = $_POST['date'];

$time = $_POST['time'];

$details = $_POST['details'];

$price = $_POST['price'];

$place = $_POST['place'];

$vk = $_POST['vk'];

$whatsapp = $_POST['whatsapp'];

$telegram = $_POST['telegram'];

$odnoklassniki = $_POST['odnoklassniki'];

$viber = $_POST['viber'];


$servername = "localhost";
$username = "root";
$password = "";
$dbname = "BASE";


$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connect failed: " . $conn->connect_error);
}
```

```

$sql = "INSERT INTO poputchik2 (name, number, otkuda, kuda,
time, date, details, price, place, vk, whatsapp, telegram,
odnoklassniki, viber)

VALUES('$name','$number','$otkuda','$kuda','$time','$date','$det
ails','$price','$place','$vk','$whatsapp','$telegram','$odnoklas
sniki','$viber')";

if($conn->query($sql) === TRUE ) {

    echo "Данные отправлены!";

} else {

    echo "Error" . $sql . "<br>" . $conn->error;

}

$conn->close();

?>

```

Файл «post2.php»

```
<?php
```

```
$servername = "localhost";
```

```
$username = "root";
```

```
$password = "";
```

```
$dbname = "BASE";
```

```
$connect = mysqli_connect($servername, $username, $password,  
$dbname);
```

```
if ($conn->connect_error) {
```

```
    die("Connect failed: " . $conn->connect_error);
```

```
}
```

```
$deleteDate = "DELETE FROM "poputchik2" WHERE "date" <  
CURRENT_DATE;";
```

```
$deleteDate_connect = mysqli_query($connect, $deleteDate);
```

```
?>
```

					ПГУ 1.090301.15.001 ПЗ	Лист
						99
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение В

(обязательное)

Веб-сервис по поиску попутчиков

Презентация

					ПГУ 1.090301.15.001 ПЗ	Лист
						100
Изм.	Лист	№ докум.	Подпись	Дата		

Тема работы: «Web-сервис по поиску попутчиков»

Автор работы: ст. гр. 19ВВВ1 Нырков М.А.
Научный руководитель: к.т.н. доцент каф. ВТ Слепцов Н.В.

Пенза 2023

ЦЕЛЬ И ЗАДАЧИ

Цель работы - разработать приложение «Веб-сервис по поиску попутчиков», позволяющее пассажирам находить выгодные поездки, а водителям находить пассажиров для экономии.

Задачи:

- Провести анализ предметной области;
- Разработать графический интерфейс;
- Разработать и реализовать структуру базы данных;
- Разработать программную часть для работы с базой данных;
- Произвести тестирование разработанного веб-приложения.

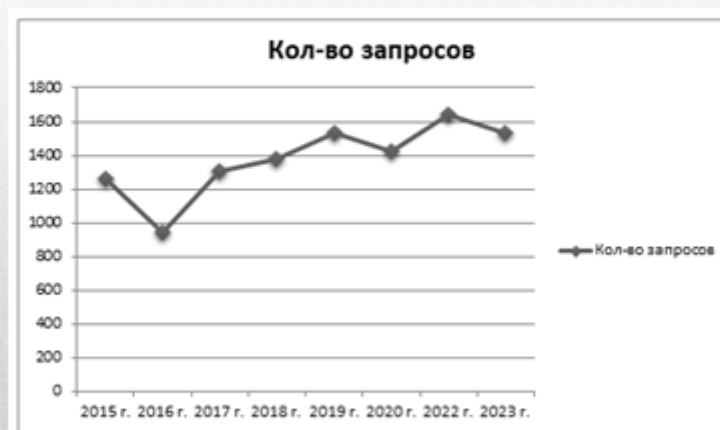
Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.15.001 ПЗ

Лист

101

График популярности запросов



Используемые технологии



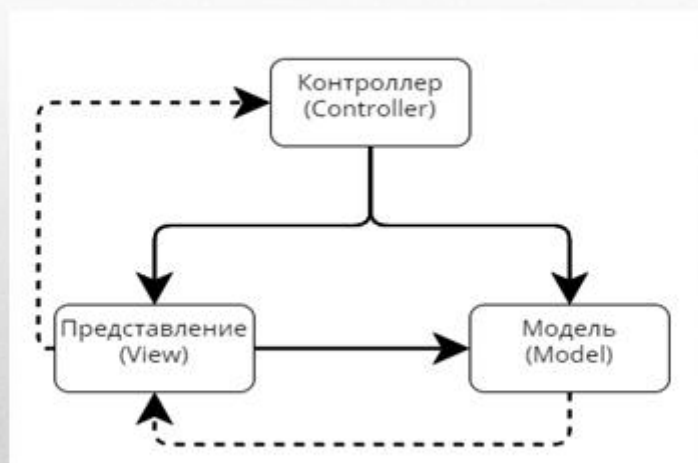
Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.15.001 ПЗ

Лист

102

Архитектура веб-сервиса



Главная страница



Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.15.001 ПЗ

Лист

103

Интерфейс создания поездки

Данные из формы

id	name	number	status	date	time	side	price	url
1	14. Водитель	+7 (900) 800-02-04	Водитель	2023-09-15	18:00	Стор	1000	https://roamride.ru/
2	15. Пасса	+7 (900) 800-02-04	Пасса	2023-09-15	18:00	Стор	1000	https://roamride.ru/
3	16. Водитель	+7 (900) 800-02-04	Водитель	2023-09-15	18:00	Стор	1000	https://roamride.ru/
4	17. Пасса	+7 (900) 800-02-04	Пасса	2023-09-15	18:00	Стор	1000	https://roamride.ru/

Страница для поиска поездки

Создать поездку Найти поездку Войти

Поиск

<p>Имя: Михаил</p> <p>Пол: Муж</p> <p>Страна: Россия</p> <p>Дата: 2023-06-12</p> <p>Время: 11:40</p> <p>Цена: 140000</p> <p>Кол-во мест: 2</p> <p>Дополнительные детали</p> <p>123</p> <p></p>	<p>Имя: Андрей</p> <p>Пол: Муж</p> <p>Страна: Россия</p> <p>Дата: 2023-06-12</p> <p>Время: 12:40</p> <p>Цена: 140000</p> <p>Кол-во мест: 2</p> <p>Дополнительные детали</p> <p>3214</p> <p></p>	<p>Имя: Артём</p> <p>Пол: Муж</p> <p>Страна: Россия</p> <p>Дата: 2023-06-16</p> <p>Время: 18:30</p> <p>Цена: 120000</p> <p>Кол-во мест: 2</p> <p>Дополнительные детали</p> <p>34042</p> <p></p>
<p>Имя: Павел</p> <p>Пол: Муж</p> <p>Страна: Россия</p> <p>Дата: 2023-06-16</p> <p>Время: 20:30</p> <p>Цена: 100000</p> <p>Кол-во мест: 1</p> <p>Дополнительные детали</p> <p>822123</p> <p></p>	<p>Имя: Артём</p> <p>Пол: Муж</p> <p>Страна: Россия</p> <p>Дата: 2023-06-16</p> <p>Время: 12:18</p> <p>Цена: 200000</p> <p>Кол-во мест: 2</p> <p>Дополнительные детали</p> <p>2200000</p> <p></p>	<p>Имя: Екатерина</p> <p>Пол: Жен</p> <p>Страна: Россия</p> <p>Дата: 2023-06-16</p> <p>Время: 18:00</p> <p>Цена: 200000</p> <p>Кол-во мест: 2</p> <p>Дополнительные детали</p> <p>821022</p> <p></p>

Поиск нужной поездки

Создать поездку Найти поездку Войти

Поиск

<p>Имя: Михаил</p> <p>Пол: Муж</p> <p>Страна: Россия</p> <p>Дата: 2023-06-12</p> <p>Время: 11:40</p> <p>Цена: 140000</p> <p>Кол-во мест: 2</p> <p>Дополнительные детали</p> <p>123</p> <p></p>	<p>Имя: Артём</p> <p>Пол: Муж</p> <p>Страна: Россия</p> <p>Дата: 2023-06-16</p> <p>Время: 12:18</p> <p>Цена: 200000</p> <p>Кол-во мест: 2</p> <p>Дополнительные детали</p> <p>2200000</p> <p></p>	<p>Имя: Екатерина</p> <p>Пол: Жен</p> <p>Страна: Россия</p> <p>Дата: 2023-06-16</p> <p>Время: 18:00</p> <p>Цена: 200000</p> <p>Кол-во мест: 2</p> <p>Дополнительные детали</p> <p>821022</p> <p></p>
<p>Имя: Михаил</p> <p>Пол: Муж</p> <p>Страна: Россия</p> <p>Дата: 2023-06-11</p> <p>Время: 18:00</p> <p>Цена: 100000</p> <p>Кол-во мест: 2</p> <p>Дополнительные детали</p> <p>42434</p> <p></p>		

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.15.001 ПЗ

Лист

105

ЗАКЛЮЧЕНИЕ

В результате проделанной работы был реализован web-сервис на базе классического паттерна модель-вид-контроллер, что позволяет легко масштабировать приложение. Задачи, поставленные на дипломный проект выполнены.

А так же следует отметить, что выбран популярный инструментарий для разработки: PHP и MySQL.

Данный набор инструментов позволяет быстро собрать команду разработки, т.к. данные инструменты являются популярными среди разработчиков.

Спасибо за внимание!