

Минобрнауки России

Юго-Западный государственный университет

Кафедра программной инженерии

ОТЧЕТ

о преддипломной (производственной) практике

наименование вида и типа практики

на (в) ООО «МЦОБ. Онлайн-сервисы»

наименование предприятия, организации, учреждения

Студента 4 курса, группы ПО-016

курса, группы

Гуляева Даниила Александровича

фамилия, имя, отчество

Руководитель практики от
предприятия, организации,
учреждения

Оценка

директор

должность, звание, степень

Куркина А. В.

фамилия и. о.

подпись, дата

Руководитель практики от
университета

Оценка

к.т.н. доцент

должность, звание, степень

Чаплыгин А. А.

фамилия и. о.

подпись, дата

Члены комиссии

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

подпись, дата

фамилия и. о.

Курск 2024 г.

СОДЕРЖАНИЕ

1	Анализ предметной области	3
1.1	Общественный транспорт и его значимость в городской жизни	3
1.2	Основные преимущества и недостатки автобусов как вида общественного транспорта	4
1.3	Основные преимущества и недостатки маршрутного такси как вида общественного транспорта	5
1.4	Основные преимущества и недостатки трамваев как вида общественного транспорта	7
1.5	Основные преимущества и недостатки метро как вида общественного транспорта	8
1.6	Основные преимущества и недостатки поездов как вида общественного транспорта	10
1.7	Основные преимущества и недостатки троллейбусов как вида общественного транспорта	11
1.8	Основные преимущества и недостатки такси как вида общественного транспорта	13
2	Техническое задание	15
2.1	Основание для разработки	15
2.2	Цель и назначение разработки	15
2.3	Требования пользователя к интерфейсу	16
2.4	Моделирование вариантов использования	19
2.4.1	Вариант использования «Просмотр рейтинга водителей»	20
2.4.2	Вариант использования «Добавление новых водителей»	21
2.4.3	Вариант использования «Регистрация аккаунта»	21
2.4.4	Вариант использования «Авторизация пользователя»	21
2.4.5	Вариант использования «Отправка обращений»	22
2.4.6	Вариант использования «Модерация обращений»	22
2.4.7	Вариант использования «Просмотр информации в личном кабинете»	23

2.5 Требования к оформлению документации	23
3 Технический проект	24
3.1 Общие сведения о программной системе	24
3.2 Проектирование архитектуры программной системы	25
3.2.1 Выбор архитектурного стиля и паттернов проектирования	25
3.2.2 Описание REST API микросервисов	26
3.2.3 Структура базы данных	30
3.2.3.1 Схема данных	32
3.2.3.2 Описание схемы данных	33
3.2.4 Описание микросервисов	39
3.2.5 Архитектура сервисов	41
3.2.5.1 Класс «CategoriesOfDriverSLicense»	41
3.2.5.2 Класс «Driver»	41
3.2.5.3 Класс «DriverStatus»	42
3.2.5.4 Класс «EmployeeStatus»	43
3.2.5.5 Класс «Flight»	43
3.2.5.6 Класс «FlightsStatus»	44
3.2.5.7 Класс «Passenger»	45
3.2.5.8 Класс «Rating»	45
3.2.5.9 Класс «RouteStop»	46
3.2.5.10Класс «Staff»	47
3.2.5.11Класс «Stop»	48
3.2.5.12Класс «UrbanRoute»	48
3.2.5.13Класс «User»	49
3.2.5.14Класс «Vehicle»	50
3.2.5.15Класс «EmployeeId»	51
3.2.5.16Класс «DriverDto»	52
3.2.5.17Класс «PassengerDto»	53
3.3 Обоснование выбора технологии проектирования	54
3.3.1 Описание используемых технологий и языков программирования	54

3.3.2	Язык программирования JavaScript	54
3.3.3	Язык программирования C#	55
3.3.4	HTML	55
3.3.5	React	56
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	57

1 Анализ предметной области

1.1 Общественный транспорт и его значимость в городской жизни

Перевозка пассажиров по городу представляет собой важное и ответственное звено в инфраструктуре. Ежедневно сотни тысяч горожан воспользуются общественным транспортом, чтобы достичь своих назначенных мест. Городской транспорт обеспечивает плавное передвижение к местам работы, учебы, медицинских учреждений и другим объектам, необходимым для выполнения повседневных дел.

В арсенале общественного транспорта имеется ряд различных средств передвижения, каждое из которых обладает своими особенностями. Среди них можно отметить:

- Автобусы – широко распространенный вид транспорта, следующий по установленным маршрутам и останавливающийся на остановках для посадки и высадки пассажиров.
- Маршрутные такси (маршрутки) – пассажирские автобусы, следующие по установленным маршрутам с более гибким графиком и маршрутом, чем обычные автобусы.
- Трамваи – электрические транспортные средства, движущиеся по рельсам по определенным маршрутам.
- Метро – подземный или надземный железнодорожный транспорт с несколькими линиями и станциями, обеспечивающий быстрое и эффективное передвижение в городе.
- Поезда – обеспечивающие связь между городами и имеющие различные маршруты и расписания.
- Троллейбусы – электрические транспортные средства, движущиеся по улицам города и питающиеся электричеством от воздушной контактной сети.
- Такси – предоставляющие индивидуальные или групповые поездки на короткие и средние расстояния в городе или за его пределами.

Разнообразие городского транспорта позволяет каждому горожанину выбрать оптимальный способ передвижения в соответствии с его потребностями и предпочтениями. Важно отметить, что услуги перевозки пассажиров должны быть организованы качественно и безопасно как для самих пассажиров, так и для других участников дорожного движения. Для эффективного управления различными видами транспорта учреждены соответствующие министерства и ведомства. Управление транспортной системой представляет собой комплекс мероприятий, нацеленных на координацию деятельности транспортных элементов как внутри системы, так и в контексте внешней среды. Эффективное управление транспортными средствами требует обладания знанием правил дорожного движения, выполнения налоговых обязательств, а также определения структуры платных и бесплатных участков дорожной инфраструктуры и учета особенностей движения при перевозке большого числа пассажиров, среди прочего. Эти аспекты определяют правила использования городского пассажирского транспорта.

1.2 Основные преимущества и недостатки автобусов как вида общественного транспорта

Автобусы являются ключевым элементом общественного транспорта во многих городах по всему миру. Они представляют собой пассажирские автотранспортные средства, предназначенные для перевозки людей по определенным маршрутам и остановкам в городских и пригородных районах.

Характеристика:

- Комфортные условия: в современных автобусах предусмотрены комфортные сиденья, освещение, системы вентиляции и отопления для обеспечения удобства пассажиров во время поездки.
- Системы безопасности: автобусы обычно оснащены системами безопасности, такими как системы контроля стабильности, подушки безопасности и системы наблюдения за пассажирами.

- Информационные системы: многие автобусы оснащены информационными табло, которые сообщают пассажирам информацию о текущем местоположении, следующей остановке и времени прибытия.

К преимуществам автобусов можно отнести:

- Гибкость маршрутов: автобусы имеют возможность адаптироваться к изменениям спроса и трафика, что позволяет оптимизировать маршруты и удовлетворить потребности пассажиров в различных районах города.

- Доступность для широкого круга пользователей: автобусы являются доступным средством транспорта для всех категорий горожан, включая маломобильные группы населения, людей с ограниченными физическими возможностями и тех, кто не владеет собственным автотранспортом.

- Равномерное покрытие городских районов: благодаря распределению маршрутов, автобусы обеспечивают равномерное покрытие городских районов, что позволяет жителям области легко добираться до различных объектов, таких как магазины, учебные заведения и медицинские учреждения.

К недостаткам автобусов можно отнести:

- Ограниченная скорость передвижения: в отличие от некоторых других видов общественного транспорта, таких как метро или трамваи, автобусы могут быть подвержены задержкам из-за дорожного движения, что может снизить скорость передвижения и увеличить время в пути.

- Перегруженность в пиковые часы: в часы пик автобусы часто бывают переполнены, что приводит к дискомфорту для пассажиров и увеличивает время ожидания на остановках. Это также может привести к неравномерности в обслуживании, особенно в плотно населенных районах города.

1.3 Основные преимущества и недостатки маршрутного такси как вида общественного транспорта

Маршрутные такси, или маршрутки, представляют собой небольшие пассажирские автобусы или микроавтобусы, которые следуют по установленным маршрутам. В отличие от стандартных городских автобусов, маршрутки обладают большей гибкостью как в графике движения, так и в марш-

рутах. Они могут более оперативно адаптироваться к изменяющимся условиям дорожного движения и потребностям пассажиров. Маршрутные такси часто останавливаются по требованию пассажиров, что делает их особенно удобными для передвижения по городским улицам, где плотность населения высокая и потребность в частых остановках велика. Благодаря своей маневренности и меньшему размеру, они способны обслуживать маршруты, которые могут быть недоступны для больших автобусов, обеспечивая доступ к отдаленным и труднодоступным районам города.

К преимуществам маршрутного такси, как вида общественного транспорта, можно отнести:

- Гибкость маршрутов: маршрутные такси предлагают более гибкий график и маршрут, чем обычные автобусы, что обеспечивает большую гибкость в передвижении по городу и позволяет достичь более отдаленных районов.
- Быстрое перемещение: за счет отсутствия остановок на каждой остановке, маршрутные такси могут обеспечить более быстрое перемещение пассажиров, особенно в случае коротких поездок.
- Доступность для малых групп: маршрутные такси могут быть удобным вариантом для малых групп пассажиров, так как они обычно имеют меньшую вместимость и могут быть арендованы для частных поездок или экскурсий.

К недостаткам маршрутного такси можно отнести:

- Ограниченная вместимость: в отличие от общественных автобусов, маршрутные такси обычно имеют меньшую вместимость, что может привести к перегруженности и дискомфорту в часы пик или при большом количестве пассажиров.
- Неустойчивость маршрутов: из-за более гибкого графика и маршрута, маршрутные такси могут быть менее надежными в плане регулярности движения, особенно в случае изменений трафика или неблагоприятных погодных условий.

- Ограниченный охват маршрутов: маршрутные такси могут не охватывать все районы города или работать только на определенных маршрутах, что может снизить их удобство для некоторых пассажиров.

1.4 Основные преимущества и недостатки трамваев как вида общественного транспорта

Трамваи представляют собой надежное и эффективное средство общественного транспорта, которое функционирует за счет электрической энергии и движется по специальным рельсовым путям. Благодаря своей способности обрабатывать большие потоки пассажиров, трамваи являются неотъемлемой частью транспортной инфраструктуры многих городов. Они обладают высокой проходимостью, что позволяет им эффективно справляться с городскими заторами и обеспечивать своевременную доставку пассажиров.

Современные трамваи оборудованы передовыми системами безопасности, включая автоматические тормозные системы, средства обнаружения препятствий и аварийные сигнальные устройства. Это обеспечивает высокий уровень безопасности как для пассажиров, так и для других участников дорожного движения. Комфорт пассажиров также находится на высоком уровне: в трамваях предусмотрены удобные сиденья, системы климат-контроля, информационные табло и широкие двери для удобного входа и выхода. Некоторые модели также оснащены местами для инвалидных колясок и детских колясок, что делает их доступными для всех категорий граждан.

Кроме того, трамваи являются экологически чистым видом транспорта, так как работают на электричестве и не выбрасывают вредные вещества в атмосферу. Это способствует улучшению качества воздуха в городе и снижению уровня шумового загрязнения. В целом, трамваи сочетают в себе эффективность, безопасность и экологичность, играя важную роль в обеспечении устойчивого городского транспорта.

К преимуществам трамваев, как вида общественного транспорта, можно отнести:

- Экологическая чистота: трамваи являются экологически чистым видом транспорта, так как работают на электричестве, что способствует снижению выбросов вредных веществ и улучшению качества воздуха в городе.

- Стабильность маршрутов: трамваи обычно следуют по фиксированным маршрутам и расписаниям, что обеспечивает стабильность в расписании движения и предсказуемость для пассажиров.

- Высокая вместимость: благодаря большим вместительным способностям трамваи могут перевозить большое количество пассажиров за один рейс, что делает их эффективным средством общественного транспорта в плотно населенных районах.

К недостаткам трамваев относится:

- Ограниченная мобильность: трамваи ограничены своими рельсовыми путями, что делает их менее гибкими и мобильными по сравнению с другими видами транспорта, особенно в случае необходимости изменения маршрутов.

- Ограниченное покрытие городских районов: в некоторых городах трамваи охватывают лишь ограниченное количество районов, что может создавать неудобства для жителей районов, не попадающих в зону покрытия.

1.5 Основные преимущества и недостатки метро как вида общественного транспорта

Метрополитен является ключевым элементом городской инфраструктуры, обеспечивающим быструю и надежную перевозку горожан. Благодаря своей независимости от наземного дорожного движения, метро позволяет пассажирам избегать пробок и других задержек, характерных для уличного транспорта. Это особенно важно в час пик, когда городские дороги перегружены, а спрос на транспортные услуги значительно возрастает.

Современные системы метрополитена оснащены передовыми технологиями и оборудованием, обеспечивающими высокий уровень безопасности и комфорта для пассажиров. В вагонах метро обычно установлены системы климат-контроля, информационные дисплеи, обеспечивающие актуаль-

ную информацию о маршруте и следующей станции, а также системы видеонаблюдения для обеспечения безопасности. Кроме того, на станциях метро предусмотрены удобства для маломобильных граждан, такие как лифты, эскалаторы и пандусы.

Метрополитен также является экологически чистым видом транспорта, поскольку поезда работают на электричестве и не выбрасывают вредных веществ в атмосферу. Это способствует снижению уровня загрязнения воздуха в городе и поддержанию экологического баланса.

Станции метро часто расположены в стратегически важных районах города, таких как деловые центры, жилые районы и туристические достопримечательности, что делает метрополитен удобным и доступным видом транспорта для широкой аудитории. Благодаря высокой частоте движения поездов, пассажиры могут рассчитывать на минимальное время ожидания, что делает метро одним из самых эффективных способов передвижения по городу.

Таким образом, метрополитен не только обеспечивает быструю и надежную перевозку горожан, но и играет важную роль в улучшении качества городской среды и поддержании устойчивого развития городов.

К преимуществам метро, как вида общественного транспорта, можно отнести:

- Высокая скорость передвижения: метрополитен обеспечивает быструю перевозку пассажиров благодаря использованию высокоскоростных поездов и отсутствию пробок на путях.
- Высокая пропускная способность: метро способно перевозить большие объемы пассажиров за короткие промежутки времени, что особенно важно в крупных городах с высокой плотностью населения.
- Стабильность маршрутов: линии метро обычно имеют фиксированные маршруты, что обеспечивает стабильность и предсказуемость для пассажиров.
- Экономия времени: использование метро позволяет избежать пробок и сократить время в пути, что делает его предпочтительным средством передвижения для тех, кто стремится сэкономить время.

К недостаткам метро относится:

- Ограниченная география: метро доступно не во всех районах города, что ограничивает его доступность для некоторых жителей и посетителей.
- Высокие затраты на строительство и обслуживание: создание и поддержание метрополитена требует значительных инвестиций в инфраструктуру и техническое обслуживание, что может привести к высоким операционным расходам.
- Перегруженность в часы пик: в пиковые часы метро может быть переполнено, что может вызвать дискомфорт и стресс у пассажиров, особенно на станциях пересадок.

1.6 Основные преимущества и недостатки поездов как вида общественного транспорта

Представленный вид транспорта является традиционным выбором для перевозки пассажиров на средние и дальние расстояния. Характеризуется относительной экологической чистотой и высоким уровнем надежности и безопасности, что делает его предпочтительным средством передвижения по сравнению с другими видами транспорта, такими как автобусы. Возможным недостатком может быть высокая стоимость проезда в дальнем следовании, а также относительно невысокая скорость передвижения по сравнению с авиационным транспортом.

В городских условиях часто используются пригородные электрички и, в некоторых случаях, монорельсовый транспорт. Билеты на электрички доступны по относительно низким ценам. Однако ограниченное количество остановок и маршрутов в городской зоне может ограничить удобство использования электричек для перемещений внутри города. Тем не менее, они остаются оптимальным выбором для пригородных поездок.

К преимуществам поездов, как вида общественного транспорта, можно отнести:

- Быстрое перемещение между городами: поезда предоставляют быструю и эффективную перевозку пассажиров на дальние расстояния, что делает

их привлекательным выбором для путешествий между городами и регионами.

- Комфортные условия для пассажиров: обычно поезда оснащены комфортабельными сиденьями, обширными вагонами и различными услугами, такими как кафе-вагоны, развлекательные зоны и Wi-Fi, что делает поездки более приятными и комфортными для пассажиров.

- Экологическая эффективность: использование некоторых видов поездов в качестве общественного транспорта способствует снижению выбросов вредных веществ в атмосферу, что благоприятно сказывается на окружающей среде и общественном здоровье.

К недостаткам поездов относятся:

- Ограниченное покрытие маршрутов: несмотря на преимущества в дальних перевозках, поезда обычно предлагают ограниченное количество маршрутов, что может затруднить доступ к некоторым районам или городам.

- Высокая стоимость билетов: поездка на поезде может быть дороже по сравнению с другими видами транспорта, особенно на некоторых премиальных или дальних маршрутах, что может ограничить доступность этого вида транспорта для некоторых пассажиров.

- Зависимость от расписания: поезда следуют строго определенным расписаниям, и их доступность ограничена по времени. Это может создавать неудобства для пассажиров, особенно если они не могут выезжать в определенное время.

1.7 Основные преимущества и недостатки троллейбусов как вида общественного транспорта

Троллейбусы представляют собой электрические транспортные средства, которые движутся по улицам города, питаясь электричеством от воздушной контактной сети, установленной над дорогами. Эти транспортные средства играют важную роль в городской транспортной системе, сочетая экологическую чистоту с эффективностью перевозок.

Троллейбусы оснащены надежными системами энергопитания, которые обеспечивают стабильное и бесперебойное движение по установленным маршрутам. Благодаря использованию электричества, троллейбусы не выбрасывают вредных веществ в атмосферу, что способствует улучшению качества воздуха в городских районах и снижению уровня шумового загрязнения.

Кроме того, троллейбусы часто оснащены системами видеонаблюдения и безопасного выхода, что повышает уровень безопасности как для пассажиров, так и для водителей. Это особенно важно в условиях интенсивного городского движения, где безопасность является приоритетом.

Одним из значительных особенностей троллейбусов является их маневренность и способность работать на участках дорог с крутыми подъемами и спусками, где другие виды транспорта могут испытывать трудности. Это делает троллейбусы идеальным выбором для городов с холмистым рельефом.

К преимуществам троллейбусов, как вида общественного транспорта, можно отнести:

- Экологическая чистота: троллейбусы являются экологически чистым видом транспорта, поскольку они работают на электричестве, что снижает выбросы вредных веществ и улучшает качество воздуха в городе.

- Электромобильность: зависимость от электричества делает троллейбусы более стабильными в условиях колебаний цен на топливо и снижает эксплуатационные расходы по сравнению с автобусами, работающими на дизеле или бензине.

- Надежность и стабильность маршрутов: троллейбусы, как и трамваи, движутся по фиксированным маршрутам, что обеспечивает стабильное и надежное обслуживание пассажиров в течение дня.

К недостаткам троллейбусов относится:

- Ограниченная мобильность: троллейбусы зависят от наличия воздушной контактной сети, что делает их менее мобильными по сравнению с автобусами. Они могут быть ограничены в перемещении по городу из-за нехватки инфраструктуры.

- Уязвимость к погодным условиям: плохие погодные условия, такие как сильный ветер, грозы или снегопады, могут создать проблемы для троллейбусов, так как воздушная контактная сеть может быть повреждена или нарушена.

- Ограниченная гибкость маршрутов: троллейбусы могут быть ограничены в своей способности изменять маршруты или быстро адаптироваться к изменяющимся потребностям пассажиров из-за необходимости следовать по фиксированным линиям и зависимости от инфраструктуры.

Таким образом, троллейбусы не только обеспечивают комфортные и безопасные условия для пассажиров, но и способствуют устойчивому развитию городского транспорта за счет использования экологически чистых технологий. Их интеграция в транспортную систему города позволяет эффективно решать проблемы транспортного сообщения и улучшать качество жизни горожан.

1.8 Основные преимущества и недостатки такси как вида общественного транспорта

Такси представляет собой услугу пассажирского транспорта, оказываемую частными лицами или специализированными компаниями. Эти услуги играют важную роль в транспортной системе города, предлагая пассажирам удобный и быстрый способ передвижения. Оно дополняет другие виды общественного транспорта, особенно в случаях, когда требуется быстрая и прямая поездка.

Характеристики такси включают в себя индивидуальные или групповые поездки на короткие и средние расстояния как в пределах города, так и за его пределами. Такси доступно круглосуточно, что делает его незаменимым средством передвижения в ночное время или в случаях, когда общественный транспорт недоступен.

К преимуществам такси, как вида общественного транспорта, можно отнести:

- Индивидуальная доступность: такси предоставляет индивидуальные поездки, адаптирующиеся к потребностям каждого пассажира, позволяя им выбирать время отправления, маршрут и комфортность поездки.

- Быстрое перемещение: такси обычно обеспечивают быструю и удобную доставку пассажиров к их месту назначения, особенно на короткие и средние расстояния, что позволяет сэкономить время в сравнении с общественным транспортом или личным автомобилем.

- Гибкий график работы: такси доступны в любое время суток, что делает их удобным средством передвижения как днем, так и ночью, а также в условиях праздников или чрезвычайных ситуаций.

К недостаткам такси относятся:

- Высокие стоимость услуг: в сравнении с общественным транспортом или собственным автомобилем, стоимость поездок на такси может быть значительно выше, особенно на длинные расстояния или в периоды пикового спроса.

- Неопределенность в стоимости и маршруте: стоимость поездки на такси и маршрут могут варьироваться в зависимости от множества факторов, таких как трафик, погодные условия и политика тарифов таксиста или компании, что может привести к неожиданным расходам для пассажиров.

- Ограниченная доступность: в отдаленных или малонаселенных районах такси могут быть менее доступными или вовсе отсутствовать, что делает их менее привлекательным вариантом для путешествий в таких местах.

2 Техническое задание

2.1 Основание для разработки

На основании актуальности проблемы, связанной с частыми нарушениями правил дорожного движения, недостаточным качеством предоставления услуг общественного транспорта и другими аспектами, возникшими в сфере общественного транспорта, было принято решение о разработке веб-приложения для сбора и анализа информации об инцидентах на дорогах общего пользования.

2.2 Цель и назначение разработки

Функциональное предназначение разрабатываемого веб-приложения заключается в сборе, анализе и обработке информации, поступающей от пассажиров и граждан, касающейся инцидентов с общественным транспортом. Целью является устранение текущих недочетов, связанных с инцидентами в сфере общественного транспорта.

На текущем этапе разработки приложение будет заниматься обработкой обращений, связанных с инцидентами в городском общественном транспорте безрельсового типа, таких как автобусы, маршрутные такси и троллейбусы. В дальнейшем планируется расширение функционала на все виды общественного транспорта.

Веб-приложение будет обеспечивать эффективную и систематизированную обработку инцидентов, что позволит соответствующим службам и ответственным лицам оперативно реагировать на возникающие проблемы и сокращать их негативное воздействие на пассажиров. В дополнение к этому, приложение будет предоставлять гражданам удобную платформу для обратной связи, способствуя увеличению прозрачности и уровня доверия к деятельности транспортных служб.

Согласно поставленной цели, были сформулированы следующие задачи:

1. Разработка информационного веб-приложения.

2. Создание базы данных.
3. Внедрение функционала авторизации и регистрации.
4. Реализация функционала для создания и отображения новостной ленты.
5. Осуществление системы загрузки обращений.

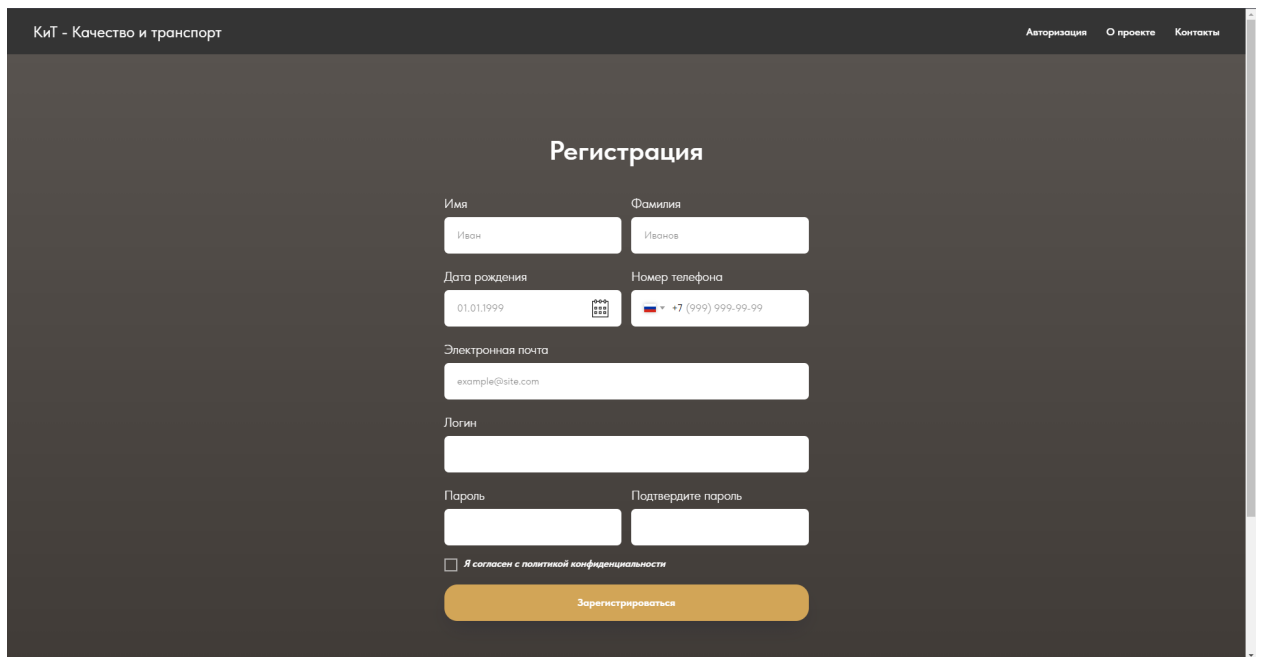
В результате выполнения поставленных задач будет создано полноценное информационное веб-приложение, включающее надежную базу данных для хранения, обработки и анализа информации. Пользователи смогут регистрироваться в системе, а также получать доступ к ленте новостей для оперативного информирования о событиях в области транспорта и дорожного движения. Дополнительно, приложение будет оборудовано системой обратной связи, что позволит улучшить взаимодействие между пользователями и транспортными службами.

2.3 Требования пользователя к интерфейсу

Веб-приложение должно содержать следующие компоненты:

- форма регистрации;
- форма авторизации;
- форма для добавления обращения;
- раздел «Новостная лента», представляющий собой список обращений, успешно прошедших модерацию;
- отображение рейтинга водителя;
- личный кабинет пользователя;
- административная панель.

На рисунках 2.1-2.4 представлены макеты интерфейса пользователя программного продукта.



The registration form is titled "Регистрация" and is set against a dark gray background. It includes input fields for "Имя" (Name) with the placeholder "Иван", "Фамилия" (Surname) with "Иванов", "Дата рождения" (Date of birth) with "01.01.1999", and "Номер телефона" (Phone number) with "+7 (999) 999-99-99". There is also a field for "Электронная почта" (Email) with "example@site.com", a "Логин" (Login) field, and fields for "Пароль" (Password) and "Подтвердите пароль" (Confirm password). A checkbox labeled "Я согласен с политикой конфиденциальности" (I agree with the privacy policy) is located below the password fields. A large orange button labeled "Зарегистрироваться" (Register) is at the bottom.

КиТ - Качество и транспорт

Авторизация О проекте Контакты

Регистрация

Имя: Иван

Фамилия: Иванов

Дата рождения: 01.01.1999

Номер телефона: +7 (999) 999-99-99

Электронная почта: example@site.com

Логин:

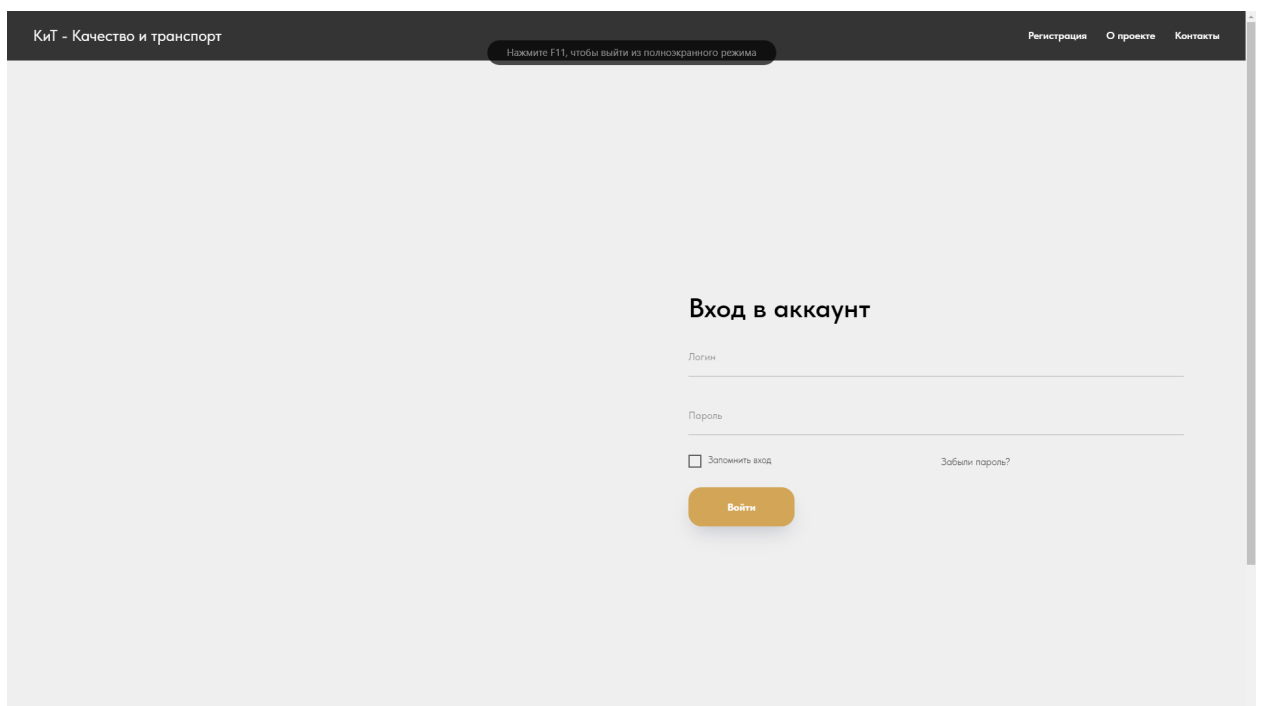
Пароль:

Подтвердите пароль:

☐ Я согласен с политикой конфиденциальности

Зарегистрироваться

Рисунок 2.1 – Макет формы регистрации



The login form is titled "Вход в аккаунт" and is set against a light gray background. It includes input fields for "Логин" (Login) and "Пароль" (Password). Below the password field is a checkbox labeled "Запомнить вход" (Remember login) and a link "Забыли пароль?" (Forgot password?). A large orange button labeled "Войти" (Login) is at the bottom. The header includes the site name "КиТ - Качество и транспорт" and navigation links "Регистрация", "О проекте", and "Контакты". A dark bar with the text "Нажмите F11, чтобы выйти из полноэкранного режима" (Press F11 to exit full-screen mode) is also present.

КиТ - Качество и транспорт

Регистрация О проекте Контакты

Нажмите F11, чтобы выйти из полноэкранного режима

Вход в аккаунт

Логин:

Пароль:

☐ Запомнить вход

[Забыли пароль?](#)

Войти

Рисунок 2.2 – Макет формы авторизации

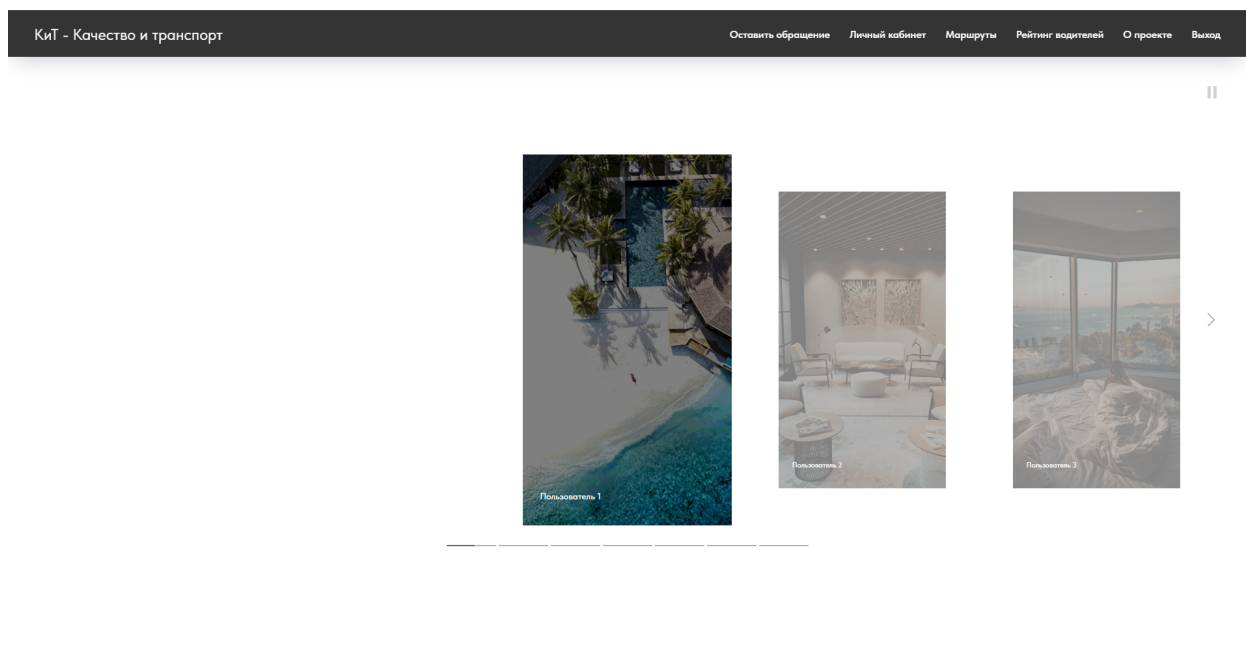


Рисунок 2.3 – Раздел «Новостная лента»

Рисунок 2.4 – Раздел «Отправка обращения»

2.4 Моделирование вариантов использования

Для разрабатываемого веб-приложения была реализована диаграмма прецедентов, которая способствует физической разработке и детальному анализу взаимосвязей объектов. Для построения диаграммы вариантов использования применяется унифицированный язык визуального моделирования UML.

Диаграмма вариантов использования описывает функциональное назначение разрабатываемой системы, то есть показывает, что система будет делать в процессе своего функционирования. Она представляет собой исходное концептуальное представление системы в процессе проектирования и разработки. В проектируемой системе прецеденты представляют собой действия, предоставляемые системой актерам или сущностям, взаимодействующим с системой. Актером является сущность, взаимодействующая с системой извне, будь то человек или техническое устройство. Прецедент описывает набор действий, которые система выполняет для актера. На основании анализа предметной области в программе должны быть реализованы следующие прецеденты:

1. Регистрация аккаунта.
2. Авторизация пользователя.
3. Отправка обращений.
4. Публикация обращений.
5. Просмотр информации в личном кабинете.
6. Модерация обращений.

На рисунке 2.5 представлены функциональные требования к системе в виде диаграммы прецедентов нотации UML.

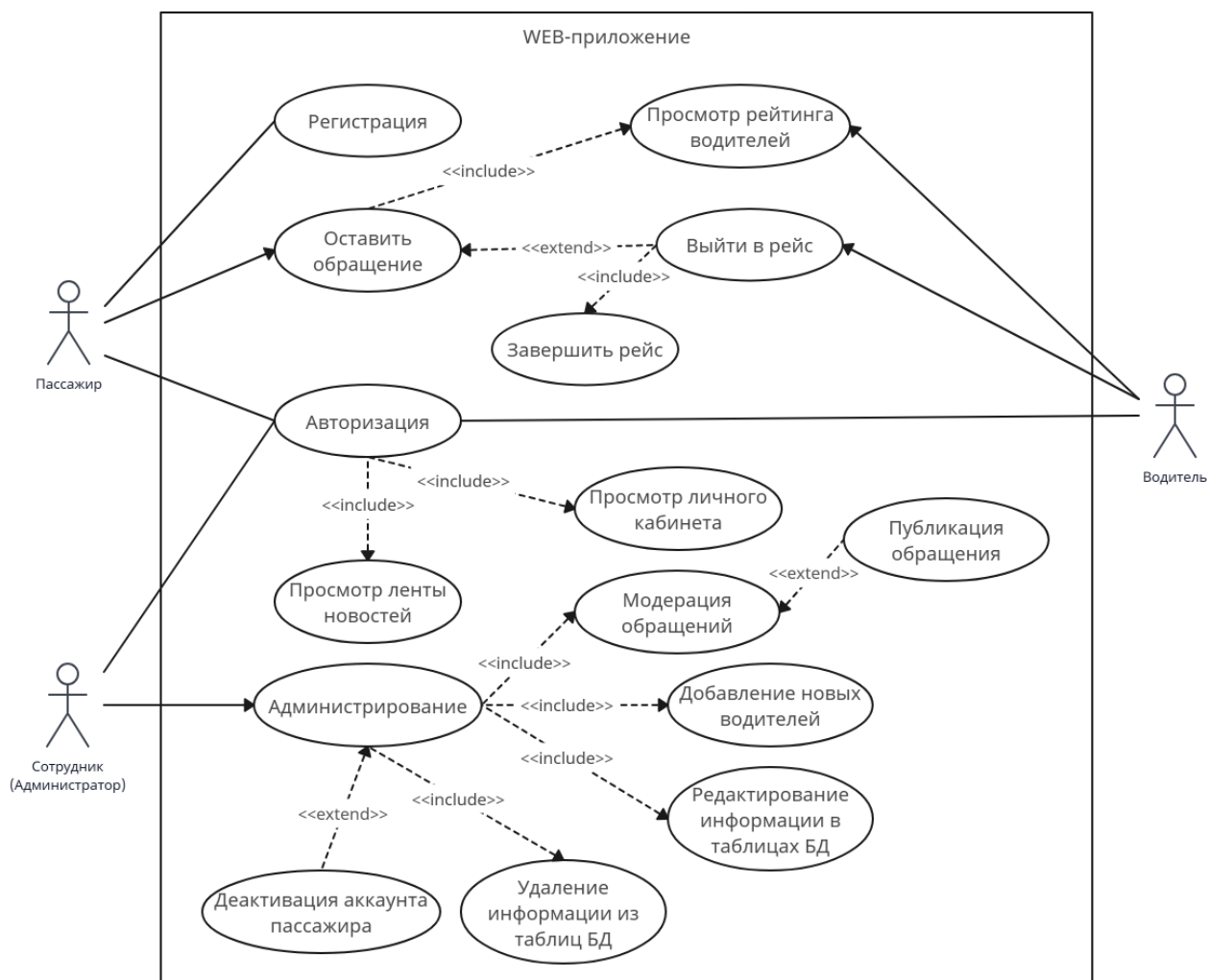


Рисунок 2.5 – Диаграмма прецедентов

2.4.1 Вариант использования «Просмотр рейтинга водителей»

Заинтересованные лица и их требования: Пользователи, желающие ознакомиться с рейтингом водителей.

Предусловие: Пользователь заходит в систему под своим аккаунтом.

Постусловие: Пользователь просматривает рейтинг водителей. Основной успешный сценарий:

Основной успешный сценарий:

1. Пользователь заходит в раздел «Рейтинг водителей».
2. Система отображает рейтинговую таблицу водителей, выводя основную информацию: ФИО водителя, номер маршрута, общая оценка.

2.4.2 Вариант использования «Добавление новых водителей»

Заинтересованные лица и их требования: Пользователи, которые несут ответственность за создание аккаунтов новых водителей.

Предусловие: Пользователь вошел в систему под своим аккаунтом, имеющим расширенные права доступа.

Постусловие: Пользователь создал профиль нового водителя.

Основной успешный сценарий:

1. Пользователь заходит в панель администратора.
2. Пользователь переходит в раздел «Добавить нового водителя».
3. Пользователь корректно заполняет все поля формы регистрации.
4. Пользователь нажимает на кнопку «Сохранить».
5. Система создает и открывает аккаунт нового водителя.

2.4.3 Вариант использования «Регистрация аккаунта»

Заинтересованные лица и их требования: Пользователи, желающие получить доступ к веб-приложению.

Предусловие: Пользователь открывает страницу регистрации.

Постусловие: Пользователь имеет аккаунт в системе.

Основной успешный сценарий:

1. Пользователь заходит на страницу регистрации.
2. Пользователь корректно заполняет все поля формы регистрации.
3. Пользователь нажимает на кнопку «Зарегистрироваться».
4. Система создает аккаунт пользователя.

2.4.4 Вариант использования «Авторизация пользователя»

Заинтересованные лица и их требования: Пользователи, желающие получить доступ к веб-приложению.

Предусловие: Пользователь открывает страницу авторизации.

Постусловие: Пользователь попадает на главную страницу.

Основной успешный сценарий:

1. Пользователь заходит на страницу авторизации.
2. Пользователь корректно вводит логин и пароль от своего аккаунта в соответствующих полях формы.
3. Пользователь нажимает на кнопку «Войти».
4. Система загружает главную страницу.

2.4.5 Вариант использования «Отправка обращений»

Заинтересованные лица и их требования: Пользователи, желающие оставить обращение о случившемся инциденте.

Предусловие: Пользователь открывает раздел «Оставить обращение».

Постусловие: Пользователь отправляет обращение в систему.

Основной успешный сценарий:

1. Пользователь заходит в раздел «Оставить обращение».
2. Пользователь заполняет все необходимые поля.
3. Пользователь нажимает на кнопку «Отправить».
4. Система создает новое обращение и отправляет его на проверку администратору.

2.4.6 Вариант использования «Модерация обращений»

Заинтересованные лица и их требования: Пользователи, которые несут ответственность за корректность контента обращений.

Предусловие: Пользователь вошел в систему под своим аккаунтом, имеющим расширенные права доступа.

Постусловие: Пользователь допускает обращение для публикации.

Основной успешный сценарий:

1. Пользователь заходит в раздел «Новые обращения».
2. Пользователь проверяет правильность введенных данных, а также содержание обращения на корректность и достоверность.
3. Пользователь нажимает на кнопку «Опубликовать».
4. Система показывает обращение на главной странице с целью ознакомления для других пользователей.

2.4.7 Вариант использования «Просмотр информации в личном кабинете»

Заинтересованные лица и их требования: Пользователи, желающие ознакомиться или дополнить информацию о себе в личном кабинете.

Предусловие: Пользователь заходит в систему под своим аккаунтом.

Постусловие: Пользователь просматривает или дополняет информацию о себе.

Основной успешный сценарий:

1. Пользователь заходит в раздел «Личный кабинет».
2. Пользователь просматривает имеющуюся информацию и/или добавляет новую информацию.
3. Пользователь нажимает на кнопку «Сохранить» при добавлении новых данных.
4. Система выводит сообщение «Данные сохранены», при добавлении новых данных.

2.5 Требования к оформлению документации

Разработка программной документации и программного изделия должна производиться согласно ГОСТ 19.102-77 и ГОСТ 34.601-90. Единая система программной документации.

3 Технический проект

3.1 Общие сведения о программной системе

Необходимо спроектировать и разработать веб-приложение для освещения инцидентов, связанных с общественным транспортом, такими как автобусы, троллейбусы и маршрутные такси. Это веб-приложение предоставит пользователям возможность просматривать посты и видеозаписи, содержащие информацию об инцидентах с общественным транспортом.

Для доступа к просмотру или оставлению обращений пользователи должны будут зарегистрироваться или войти в систему. Это обеспечит контроль над контентом и позволит отслеживать активность каждого пользователя. Пользователи смогут оставлять свои обращения о случившихся инцидентах, заполняя форму обратной связи, где они будут описывать произошедшее событие. Зарегистрированные пользователи смогут просматривать все доступные посты и видеозаписи, связанные с инцидентами, которые будут отображаться в формате новостной ленты. Все оставленные обращения будут проходить обязательную модерацию, в ходе которой тексты будут проверяться на отсутствие неподобающего контента. Только обращения, прошедшие модерацию, будут публиковаться в ленте.

Администраторы получат расширенные права доступа, позволяющие им добавлять новых водителей в базу данных, редактировать существующую информацию, а также управлять обращениями. Административная панель предоставит все необходимые инструменты для выполнения этих задач. В приложении будет предусмотрена функция отображения рейтинга водителей, что позволит пассажирам оценивать их работу и оставлять обратную связь. Пользователи также будут иметь доступ к личному кабинету, где они смогут управлять своей информацией, просматривать свои обращения и взаимодействовать с приложением.

Это веб-приложение предназначено для освещения инцидентов, связанных с общественным городским транспортом, и будет служить платформой для обмена информацией между пассажирами и ответственными орга-

нами. Основная цель системы — предоставлять актуальные данные о происшествиях, чтобы ответственные службы могли оперативно реагировать на возникающие проблемы и улучшать качество обслуживания общественного транспорта.

3.2 Проектирование архитектуры программной системы

3.2.1 Выбор архитектурного стиля и паттернов проектирования

Архитектурный стиль и паттерны проектирования имеют ключевое значение при создании высокопроизводительных, масштабируемых и безопасных систем, особенно в контексте разработки REST API. REST API (Representational State Transfer Application Programming Interface) представляет собой архитектурный подход, который основывается на принципах унифицированных интерфейсов и передачи состояния между клиентом и сервером. В процессе разработки REST API критически важно выбрать правильный архитектурный стиль и использовать соответствующие паттерны проектирования, чтобы обеспечить эффективную работу системы и удовлетворение требований пользователей.

Клиент-серверная архитектура является одним из фундаментальных аспектов разработки REST API. Этот архитектурный стиль разделяет систему на две основные части: клиентскую, которая отправляет запросы на сервер, и серверную, которая обрабатывает эти запросы и возвращает результаты. Такая структура позволяет создавать масштабируемые и гибкие системы, способные обрабатывать большие объемы запросов от множества клиентов одновременно.

Одним из важных паттернов проектирования для REST API является паттерн Facade, который позволяет создать единый интерфейс для взаимодействия с комплексной системой, скрывая сложные детали реализации и предоставляя простой и интуитивно понятный интерфейс для пользователей. Использование паттерна Facade повышает модульность и гибкость REST API, облегчая его использование и сопровождение.

Также значимым паттерном является Адаптер. В контексте REST API, паттерн Адаптер позволяет преобразовывать данные из одного формата в другой, что обеспечивает совместимость между различными системами и источниками данных. Например, если данные поступают в формате, который необходимо преобразовать для работы с REST API, Адаптер справляется с этой задачей, гарантируя единообразие данных в системе.

Для обеспечения безопасности данных в REST API широко используются протокол HTTPS и технология ODBC (Open Database Connectivity). Протокол HTTPS обеспечивает защищенное соединение между клиентом и сервером, шифруя данные и предотвращая их несанкционированный доступ или изменение. Технология ODBC предоставляет универсальный интерфейс для взаимодействия с различными базами данных, обеспечивая безопасную и эффективную работу с данными в REST API.

Архитектура системы представлена на рисунке 3.1.

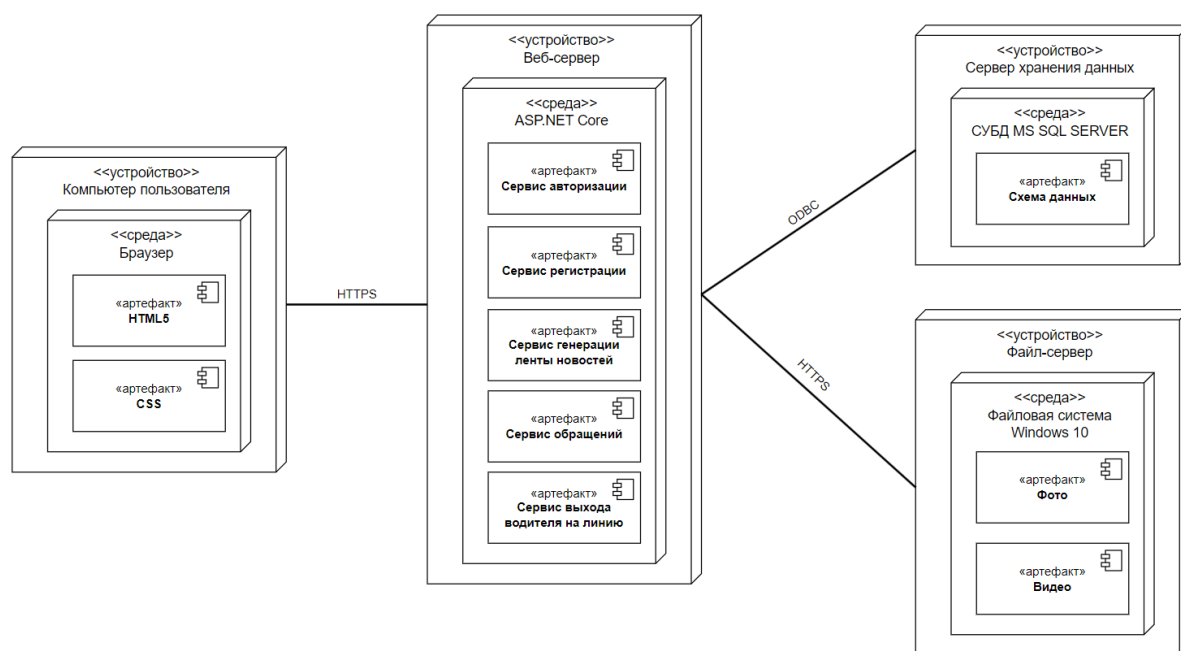


Рисунок 3.1 – Архитектура программной системы

3.2.2 Описание REST API микросервисов

Таблица 3.1 – Описание методов для работы с новостной лентой

HTTP-метод	Описание	Входные параметры	Выходные параметры
GET /api/News Generator Controller	Получение всех записей из таблицы	-	Ratings: List<object>
GET /api/ Employee GET /api/News Generator Controller/:id	Получение конкретной записи из таблицы	RatingId: integer	Rating: object
GET /api/News Generator Controller /noPublished	Получение записей, которые еще не были опубликованы	PublishedEntry: bool	Ratings: List<object>
POST /api/News Generator Controllerd	Добавление новой записи	Data: object	Rating: object
PUT /api/News Generator Controller/:id	Обновление записи в таблице	RatingId: integer, RatingUp: object	Rating: object
DELETE /api/News Generator Controller/:id	Удаление записи из таблицы	RatingId: integer	-

Таблица 3.2 – Описание методов для работы авторизации и регистрации

HTTP-метод	Описание	Входные параметры	Выходные параметры
POST /api/ Autorisation Controller/ registration	Регистрация нового пассажира	Data: object	Passenger: object
POST /api/ Autorisation Controller/: login	Авторизация пользователя	Login: string, Password: string	UserId: integer, RoleInTheSystem: string

Таблица 3.3 – Описание методов для работы с маршрутами

HTTP-метод	Описание	Входные параметры	Выходные параметры
POST /api/ UrbanRoute Controller	Добавление нового маршрута	Data: object	UrbanRoute: object
GET /api/ UrbanRoute Controller/:id	Получение нужного маршрута	RouteId: integer	UrbanRoute: object
GET /api/ UrbanRoute Controller	Получение всех маршрутов	-	UrbanRoutes: List<object>
PUT /api/ UrbanRoute Controller/:id	Обновление данных о маршруте	RouteId: integer, UrbanRouteUp: object	UrbanRoute: object
DELETE /api/ UrbanRoute Controller/:id	Удаление маршрута	RouteId: integer	-

Таблица 3.4 – Описание методов для работы с водителями

HTTP-метод	Описание	Входные параметры	Выходные параметры
POST /api/Driver Controller	Добавление данных о водителе	Data: object	Driver: object
GET /api/Driver Controller/:id	Получение данных о водителе	UserId: integer	Driver: object
PUT /api/Driver Controller/:id	Обновление данных водителя	UserId: integer, DriverUp: object	Driver: object
DELETE /api/Driver Controller/:id	Удаление данных о водителе	UserId: integer	-

Таблица 3.5 – Описание методов для работы с пассажирами

HTTP-метод	Описание	Входные параметры	Выходные параметры
GET /api/Passenger Controller/:id	Получение данных о пассажире	UserId: integer	Passenger: object
PUT /api/Passenger Controller/:id	Обновление данных пассажира	UserId: integer, PassengerUp: object	Passenger: object
DELETE /api/Passenger Controller/:id	Удаление данных о пассажире	UserId: integer	-

Таблица 3.6 – Описание методов для работы с сотрудниками

HTTP-метод	Описание	Входные параметры	Выходные параметры
POST /api/EmployeeController	Добавление нового сотрудника	Data: object	Employee: object
GET /api/EmployeeController/:id	Получение данных о сотруднике	UserId: integer	Employee: object
PUT /api/EmployeeController/:id	Обновление данных сотрудника	UserId: integer, EmployeeUp: object	Employee: object
DELETE /api/EmployeeController/:id	Удаление данных о сотруднике	UserId: integer	-

3.2.3 Структура базы данных

В качестве системы управления базами данных была выбрана реляционная СУБД Microsoft SQL Server. Она предназначена для хранения и управления данными, а также для выполнения различных задач по их обработке, анализу и управлению. SQL Server используется в корпоративных приложениях, веб-приложениях и других системах, где требуется надежное и масштабируемое хранилище данных.

Microsoft SQL Server обладает рядом преимуществ, которые делают его оптимальным выбором для управления базами данных в различных проектах. К ключевым преимуществам можно отнести:

- Высокая производительность. SQL Server оптимизирован для высокой производительности как для транзакционных рабочих нагрузок, так и для аналитических запросов. Он включает в себя технологии, такие как in-

memory processing и параллельное выполнение запросов, что позволяет обрабатывать большие объемы данных эффективно.

- **Безопасность данных.** SQL Server предоставляет обширные возможности для обеспечения безопасности данных. В него встроены функции шифрования данных, аутентификации, управления доступом на основе ролей (RBAC), а также средства для аудита и мониторинга безопасности.

- **Высокая доступность и отказоустойчивость.** Существует несколько функций для обеспечения высокой доступности и отказоустойчивости, включая Always On Availability Groups, репликацию и зеркалирование баз данных. Эти функции помогают минимизировать время простоя и обеспечивают непрерывный доступ к данным.

- **Масштабируемость.** SQL Server может масштабироваться как вертикально, так и горизонтально. Он поддерживает работу на разных уровнях — от небольших приложений до крупных корпоративных систем с большими объемами данных и высокой нагрузкой.

- **Обширная поддержка и документация.** Microsoft предоставляет обширную документацию, поддержку и ресурсы для обучения. Это включает в себя онлайн-документацию, форумы, обучающие курсы и официальные сертификации, что помогает администраторам и разработчикам быстро освоить и эффективно использовать SQL Server.

Несмотря на многочисленные преимущества, Microsoft SQL Server имеет и некоторые недостатки, которые могут быть значимыми в зависимости от конкретных требований и условий эксплуатации. К недостаткам можно отнести:

- **Обновления и совместимость.** Обновления SQL Server могут вызывать затруднения и иногда могут создавать проблемы с совместимостью приложений, особенно если они зависят от определенных версий или функций базы данных. Это требует тщательного планирования и тестирования перед внедрением обновлений.

- **Ограниченная кроссплатформенность.** Хотя в последние годы Microsoft сделала шаги в сторону кроссплатформенности (например, выпу-

стила версии SQL Server для Linux), основная платформа и многие функции SQL Server всё ещё оптимизированы для работы на Windows.

- Требования к аппаратному обеспечению. Для достижения максимальной производительности SQL Server требует значительных ресурсов аппаратного обеспечения, таких как процессор, память и дисковое пространство.

3.2.3.1 Схема данных

На основании анализа предметной области и технического задания была разработана база данных, предназначенная для хранения и обработки хранящейся информации. Схема данных представлена на рисунке 3.2.

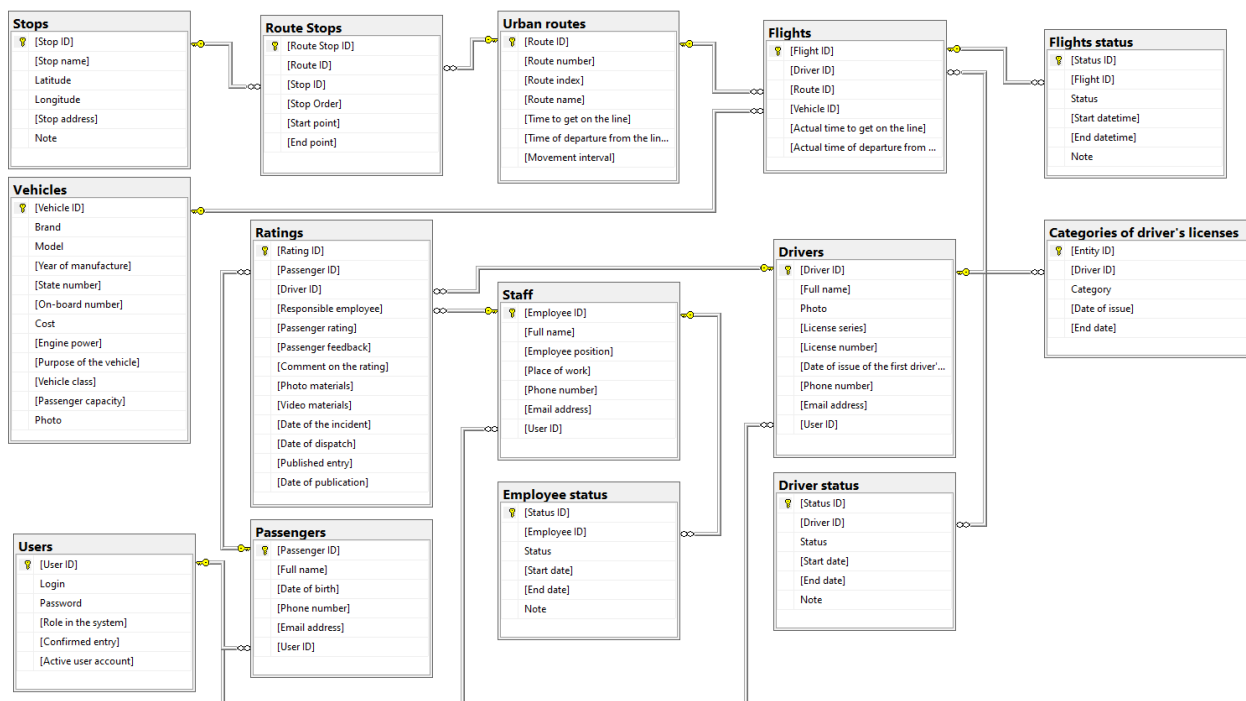


Рисунок 3.2 – Схема данных

Все сущности базы данных приведены к третьей нормальной форме, что означает, что каждая таблица удовлетворяет требованиям нормализации:

1. Отсутствие дублирования данных. В каждой таблице данные не дублируются, что помогает избежать избыточности и обеспечивает экономное использование ресурсов.

2. Зависимость атрибутов от первичного ключа. Все неключевые атрибуты в таблице зависят только от первичного ключа, а не от других неключевых атрибутов. Это гарантирует, что каждая таблица содержит данные, относящиеся к одному логическому объекту или концепции.

3. Устранение транзитивных зависимостей. Если атрибуты зависят друг от друга через другой атрибут, такие зависимости устранены. Это обеспечивает целостность данных и упрощает процесс обновления и удаления данных.

3.2.3.2 Описание схемы данных

Ниже приведено полное описание структуры базы данных, включая все таблицы и их атрибуты.

Таблица 3.7 – Описание таблицы «Users»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	User ID	INTEGER	
	*	Login	NVARCHAR	100
	*	Password	NVARCHAR	100
	*	Role in the system	NVARCHAR	15
	◦	Confirmed entry	BIT	
	◦	Active user account	BIT	

Таблица 3.8 – Описание таблицы «Drivers»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Driver ID	INTEGER	
	*	Full name	NVARCHAR	100

Продолжение таблицы 3.8

Key Type	Optionality	Column Name	Data Type	Size
	*	Photo	NVARCHAR	MAX
	*	License series	NVARCHAR	5
	*	License number	NVARCHAR	6
	*	Date of issue of the first driver's license	DATE	
	*	Phone number	NVARCHAR	20
	*	Email address	NVARCHAR	150

Таблица 3.9 – Описание таблицы «Categories of driver's license»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Entity ID	INTEGER	
fk	*	Driver ID	INTEGER	
	*	Category	NVARCHAR	50
	*	Date of issue	DATE	
	*	End date	DATE	

Таблица 3.10 – Описание таблицы «Driver status»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Status ID	INTEGER	
fk	*	Driver ID	INTEGER	
	*	Status	NVARCHAR	50
	*	Start date	DATE	
	o	End date	DATE	
	o	Note	NVARCHAR	500

Таблица 3.11 – Описание таблицы «Staff»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Employee ID	INTEGER	
	*	Full name	NVARCHAR	100
	*	Employee position	NVARCHAR	75
	*	Place of work	NVARCHAR	150
	*	Phone number	NVARCHAR	20
	*	Email address	NVARCHAR	150
fk	*	User ID	INTEGER	

Таблица 3.12 – Описание таблицы «Employee status»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Status ID	INTEGER	
fk	*	Employee ID	INTEGER	
	*	Employee position	NVARCHAR	75
	*	Status	NVARCHAR	50
	*	Start date	DATE	
	o	End date	DATE	

Таблица 3.13 – Описание таблицы «Passengers»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Passenger ID	INTEGER	
	*	Full name	NVARCHAR	100
	*	Date of birth	DATE	
	*	Phone number	NVARCHAR	20
	*	Email address	NVARCHAR	150

Продолжение таблицы 3.13

Key Type	Optionality	Column Name	Data Type	Size
fk	*	User ID	INTEGER	

Таблица 3.14 – Описание таблицы «Vehicles»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Vehicles ID	INTEGER	
	*	Brand	NVARCHAR	100
	*	Model	NVARCHAR	100
	*	Year of manufacture	DATE	
	*	State number	NVARCHAR	15
	*	On-board number	NVARCHAR	5
	o	Cost	MONEY	
	*	Engine power	INTEGER	
	*	Purpose of the vehicle	NVARCHAR	100
	*	Vehicle class	NVARCHAR	100
	*	Passenger capacity	INTEGER	
	*	Photo	NVARCHAR	MAX

Таблица 3.15 – Описание таблицы «Urban routes»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Route ID	INTEGER	
	*	Route number	INTEGER	
	o	Route index	NVARCHAR	1

Продолжение таблицы 3.15

Key Type	Optionality	Column Name	Data Type	Size
	*	Route name	NVARCHAR	100
	*	Time to get on the line	TIME	
	*	Time of departure from the line	TIME	
	◦	Movement interval	INTEGER	

Таблица 3.16 – Описание таблицы «Stops»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Stops ID	INTEGER	
	*	Stop name	NVARCHAR	100
	*	Latitude	DECIMAL	9,6
	*	Longitude	DECIMAL	9,6
	◦	Stop address	NVARCHAR	150
	◦	Note	NVARCHAR	500

Таблица 3.17 – Описание таблицы «Route stops»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Route stops ID	INTEGER	
fk	*	Route ID	INTEGER	
fk	*	Stop ID	INTEGER	
	*	Stop Order	INTEGER	9,6
	◦	Start point	BIT	
	◦	End point	BIT	

Таблица 3.18 – Описание таблицы «Flights»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Flights ID	INTEGER	
fk	*	Driver ID	INTEGER	
fk	*	Route ID	INTEGER	
fk	*	Vehicle ID	INTEGER	
	◦	Actual time to get on the line	DATETIME	
	◦	Actual time of departure from the line	DATETIME	

Таблица 3.19 – Описание таблицы «Flights status»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Status ID	INTEGER	
fk	*	Flight ID	INTEGER	
	*	Status	NVARCHAR	50
	◦	Start datetime	DATETIME	
	◦	End datetime	DATETIME	
	◦	Note	NVARCHAR	500

Таблица 3.20 – Описание таблицы «Ratings»

Key Type	Optionality	Column Name	Data Type	Size
pk	*	Rating ID	INTEGER	
fk	*	Passenger ID	INTEGER	
fk	*	Driver ID	INTEGER	
fk	*	Responsible employee ID	INTEGER	

Продолжение таблицы 3.20

Key Type	Optionality	Column Name	Data Type	Size
	*	Passenger rating	NUMERIC	10,2
	*	Passenger feedback	NVARCHAR	100
	◦	Comment on the rating	NVARCHAR	250
	◦	Photo materials	NVARCHAR	MAX
	◦	Video materials	NVARCHAR	MAX
	*	Date of the incident	DATE	
	*	Date of dispatch	DATE	
	◦	Published entry	BIT	
	◦	Date of publication	DATE	

3.2.4 Описание микросервисов

Микросервисы играют ключевую роль в разрабатываемом веб-приложении, поскольку они обеспечивают модульную архитектуру, гибкость и масштабируемость системы. Каждый микросервис представляет собой отдельную функциональную единицу, специализирующуюся на определенной задаче, что позволяет легко разрабатывать, развертывать и масштабировать приложение.

1. Микросервис работы с базой данных через панель администратора (управление базой данных).

Назначение: Управление базой данных администратором системы.

Функции: Добавление новых водителей в базу данных, редактирование и удаление информации, мониторинг активности пользователей, управление обращениями и отчетами.

Преимущества: Обеспечивает администратору удобные инструменты для управления данными, что повышает эффективность администрирования и поддержания актуальности данных.

2. Микросервис регистрации.

Назначение: Обеспечение удобного процесса регистрации новых пользователей в системе.

Функции: Обработка регистрационных данных, проверка уникальности пользователей, отправка подтверждений по электронной почте.

Преимущества: Облегчает процесс регистрации, повышая удобство для пользователей и увеличивая количество участников, активно использующих приложение.

3. Микросервис обработки обращений.

Назначение: Обработка обращений от граждан, связанных с инцидентами на общественном транспорте.

Функции: Прием обращений, проверка корректности данных, направление обращений на модерацию администратору.

Преимущества: Обеспечивает структурированное и оперативное управление обращениями, что способствует быстрому реагированию на проблемы и улучшению качества обслуживания пассажиров.

4. Микросервис аутентификации и авторизации пользователей.

Назначение: Обеспечение безопасного доступа пользователей к веб-приложению.

Функции: Управление регистрацией новых пользователей, проверка учетных данных при входе, назначение ролей и прав доступа.

Преимущества: Гарантирует, что только авторизованные пользователи могут оставлять обращения и получать доступ к персонализированным

функциям приложения, обеспечивая безопасность данных и предотвращение несанкционированного доступа.

Эти микросервисы интегрированы для создания эффективной и надежной системы управления инцидентами в общественном транспорте. Они обеспечивают структурированный подход к обработке обращений, защиту данных, гибкость управления и удобство для конечных пользователей, что в конечном итоге способствует улучшению качества обслуживания и повышению доверия к транспортным службам.

3.2.5 Архитектура сервисов

3.2.5.1 Класс «CategoriesOfDriverSLicense»

«CategoriesOfDriverSLicense» - класс, используется для получения подробной информации о лицензиях водителя.

Таблица 3.21 – Свойства класса "CategoriesOfDriverSLicense"

Свойство	Тип	Обязательное	Описание
EntityId	int	true	Уникальный идентификатор
DriverId	int	true	Идентификатор водителя
Category	string	true	Категория ВУ
DateOfIssue	DateTime	true	Дата получения категории
EndDate	DateTime	true	Дата окончания действия категории

3.2.5.2 Класс «Driver»

«Driver» - класс, содержащий в себе данные о водителе.

Таблица 3.22 – Свойства класса "Driver"

Свойство	Тип	Обязательное	Описание
DriverId	int	true	Идентификатор водителя
FullName	string	true	ФИО водителя
Photo	string	true	Путь к файлу с фото водителя
LicenseSeries	string	true	Серия ВУ
LicenseNumber	string	true	Номер ВУ
DateOfIssueOfTheFirstDriverSLicense	DateTime	true	Дата выдачи первого ВУ
PhoneNumber	string	true	Номер телефона
EmailAddress	string	true	Электронная почта
UserId	int	true	Идентификатор пользователя

3.2.5.3 Класс «DriverStatus»

«DriverStatus» - класс, который используется для получения сведений о статусе водителя (может быть уволен, либо находится на больничном).

Таблица 3.23 – Свойства класса "DriverStatus"

Свойство	Тип	Обязательное	Описание
StatusId	int	true	Уникальный идентификатор
DriverId	int	true	Идентификатор водителя
Status	string	true	Его статус
StartDate	DateTime	true	Начальная дата
EndDate	DateTime	false	Конечная дата

Таблица 3.23 – Продолжение таблицы 3.23

Свойство	Тип	Обязательное	Описание
Note	string	false	Примечание

3.2.5.4 Класс «EmployeeStatus»

«EmployeeStatus» - подобный классу «DriverStatus», содержит информацию о статусе сотрудников.

Таблица 3.24 – Свойства класса "EmployeeStatus"

Свойство	Тип	Обязательное	Описание
StatusId	int	true	Уникальный идентификатор
EmployeeId	int	true	Идентификатор сотрудника
Status	string	true	Его статус
StartDate	DateTime	true	Начальная дата
EndDate	DateTime	false	Конечная дата
Note	string	false	Примечание

3.2.5.5 Класс «Flight»

«Flight» - содержит информацию о рейсах совершаемых водителями по заданным маршрутам.

Таблица 3.25 – Свойства класса "Flight"

Свойство	Тип	Обязательное	Описание
FlightId	int	true	Уникальный идентификатор
DriverId	int	true	Идентификатор водителя

Таблица 3.25 – Продолжение таблицы 3.25

Свойство	Тип	Обязательное	Описание
RouteId	int	true	Идентификатор маршрута
VehicleId	int	true	Идентификатор автобуса
ActualTimeToGetOnTheLine	DateTime	false	Время выхода на линию
ActualTimeOfDepartureFromTheLine	DateTime	false	Время схода с линии

3.2.5.6 Класс «FlightsStatus»

«FlightsStatus» - используется для получения сведений о статусе рейса.

Таблица 3.26 – Свойства класса "FlightsStatus"

Свойство	Тип	Обязательное	Описание
StatusId	int	true	Уникальный идентификатор
FlightId	int	true	Идентификатор рейса
Status	string	true	Статус рейса
StartDatetime	DateTime	false	Начальная дата и время
EndDatetime	DateTime	false	Конечная дата и время
Note	string	false	Примечание

3.2.5.7 Класс «Passenger»

«Passenger» - применяется для работы с данными пассажиров.

Таблица 3.27 – Свойства класса "Passenger"

Свойство	Тип	Обязательное	Описание
PassengerId	int	true	Уникальный идентификатор
FullName	string	true	ФИО пассажира
DateOfBirth	DateTime	true	Дата рождения
PhoneNumber	string	true	Номер телефона
EmailAddress	string	true	Электронная почта
UserId	int	true	Идентификатор пользователя

3.2.5.8 Класс «Rating»

«Rating» - используется для сбора информации и формирования рейтинга водителей на основе отзывов пассажиров.

Таблица 3.28 – Свойства класса "Rating"

Свойство	Тип	Обязательное	Описание
RatingId	int	true	Уникальный идентификатор
PassengerId	int	true	Идентификатор пассажира
DriverId	int	true	Идентификатор водителя
Responsible Employee	int	true	Идентификатор ответственного сотрудника

Таблица 3.28 – Продолжение таблицы 3.28

Свойство	Тип	Обязательное	Описание
Passenger Rating	decimal	true	Оценка от пассажира
Passenger Feedback	string	true	Обращение пассажира
CommentOnTheRating	string	false	Дополнительные сведения
PhotoMaterials	string	false	Путь к файлу с фотоматериалом
VideoMaterials	string	false	Путь к файлу с видеоматериалом
DateOfTheIncident	DateTime	true	Дата произошедшего инцидента
DateOfDispatcht	DateTime	true	Дата отправки обращения
PublishedEntry	bool	false	Было ли опубликовано обращение
DateOfPublication	DateTime	false	Дата публикации

3.2.5.9 Класс «RouteStop»

«RouteStop» - содержит информацию об остановках на конкретном маршруте.

Таблица 3.29 – Свойства класса "RouteStop"

Свойство	Тип	Обязательное	Описание
RouteStopId	int	true	Уникальный идентификатор
RouteId	int	true	Идентификатор маршрута
StopId	int	true	Идентификатор остановки
StopOrder	int	true	Порядок следования остановок
StartPoint	bool	false	Является ли остановка начальной точкой маршрута
EndPoint	bool	false	Является ли остановка конечной точкой маршрута

3.2.5.10 Класс «Staff»

«Staff» - класс, содержащий в себе данные о сотруднике.

Таблица 3.30 – Свойства класса "Staff"

Свойство	Тип	Обязательное	Описание
EmployeeId	int	true	Уникальный идентификатор
FullName	string	true	ФИО сотрудника
EmployeePosition	string	true	Занимаемая должность
PlaceOfWork	string	true	Место работы

Таблица 3.30 – Продолжение таблицы 3.30

Свойство	Тип	Обязательное	Описание
PhoneNumber	string	true	Номер телефона
EmailAddress	string	true	Электронная почта
UserId	int	true	Идентификатор пользователя

3.2.5.11 Класс «Stop»

«Stop» - используется для получения и добавления информации об остановках в городе.

Таблица 3.31 – Свойства класса "Stop"

Свойство	Тип	Обязательное	Описание
StopId	int	true	Уникальный идентификатор
StopName	string	true	Название остановки
Latitude	decimal	true	Широта
Longitude	decimal	true	Долгота
StopAddress	string	false	Адрес остановки
Note	string	false	Примечание

3.2.5.12 Класс «UrbanRoute»

«UrbanRoute» - содержит данные о городских маршрутах.

Таблица 3.32 – Свойства класса "UrbanRoute"

Свойство	Тип	Обязательное	Описание
RouteId	int	true	Уникальный идентификатор
RouteNumber	int	true	Номер маршрута
RouteIndex	string	false	Индекс маршрута
RouteName	string	true	Наименование маршрута
TimeToGet OnTheLine	TimeSpan	true	Время начала обслуживания маршрута
TimeOfDeparture FromTheLine	TimeSpan	true	Время окончания обслуживания маршрута
MovementInterval	int	true	Интервал движения ТС

3.2.5.13 Класс «User»

«User» - класс, содержащий в себе данные пользователей веб-приложения.

Таблица 3.33 – Свойства класса "User"

Свойство	Тип	Обязательное	Описание
UserId	int	true	Уникальный идентификатор
Login	string	true	Логин пользователя
Password	string	true	Пароль пользователя

Таблица 3.33 – Продолжение таблицы 3.33

Свойство	Тип	Обязательное	Описание
RoleInTheSystem	string	true	Роль в системе
ConfirmedEntry	bool	false	Подтвержден ли аккаунт пользователя
ActiveUser Account	bool	false	Активен ли аккаунт пользователя

3.2.5.14 Класс «Vehicle»

«Vehicle» - содержит данные о маршрутном транспортном средстве.

Таблица 3.34 – Свойства класса "Vehicle"

Свойство	Тип	Обязательное	Описание
VehicleId	int	true	Уникальный идентификатор
Brand	string	true	Марка ТС
Model	string	true	Модель ТС
YearOf Manufacture	DateTime	true	Год производства
StateNumber	string	true	Государственный регистрационный знак
OnBoardNumber	string	true	Бортовой номер ТС
Cost	decimal	false	Стоимость ТС
EnginePower	int	true	Мощность двигателя в лошадиных силах

Таблица 3.34 – Продолжение таблицы 3.34

Свойство	Тип	Обязательное	Описание
PurposeOfTheVehicle	string	true	Цель назначения ТС
VehicleClass	string	true	Класс ТС
PassengerCapacity	int	true	Вместимость пассажиров
Photo	string	true	Путь к файлу с фото ТС

3.2.5.15 Класс «EmployeeId»

«EmployeeId» - содержит в себе данные пользователя системы и информацию о сотруднике.

Таблица 3.35 – Свойства класса "EmployeeId"

Свойство	Тип	Обязательное	Описание
EmployeeId	int	true	Уникальный идентификатор
FullName	string	true	ФИО сотрудника
EmployeePosition	string	true	Занимаемая должность
PlaceOfWork	string	true	Место работы
PhoneNumber	string	true	Номер телефона
EmailAddress	string	true	Электронная почта
Login	string	true	Логин пользователя
Password	string	true	Пароль пользователя

Таблица 3.35 – Продолжение таблицы 3.35

Свойство	Тип	Обязательное	Описание
RoleIn TheSystem	string	true	Роль в системе
ConfirmedEntry	bool	false	Подтвержден ли аккаунт пользо- вателя
ActiveUser Account	bool	false	Активен ли акка- унт пользователя

3.2.5.16 Класс «DriverDto»

«DriverDto» - содержит в себе данные пользователя системы и информацию о водителе.

Таблица 3.36 – Свойства класса "DriverDto"

Свойство	Тип	Обязательное	Описание
DriverId	int	true	Идентификатор водителя
FullName	string	true	ФИО водителя
Photo	string	true	Путь к файлу с фото водителя
PhoneNumber	string	true	Номер телефона
EmailAddress	string	true	Электронная почта
Login	string	true	Логин пользова- теля
Password	string	true	Пароль пользо- вателя
RoleIn TheSystem	string	true	Роль в системе

Таблица 3.36 – Продолжение таблицы 3.36

Свойство	Тип	Обязательное	Описание
ConfirmedEntry	bool	false	Подтвержден ли аккаунт пользователя
ActiveUser Account	bool	false	Активен ли аккаунт пользователя

3.2.5.17 Класс «PassengerDto»

«PassengerDto» - содержит всю информацию о пассажире, включая пользовательские данные.

Таблица 3.37 – Свойства класса "PassengerDto"

Свойство	Тип	Обязательное	Описание
PassengerId	int	true	Уникальный идентификатор
FullName	string	true	ФИО пассажира
DateOfBirth	DateTime	true	Дата рождения
PhoneNumber	string	true	Номер телефона
EmailAddress	string	true	Электронная почта
Login	string	true	Логин пользователя
Password	string	true	Пароль пользователя
RoleIn TheSystem	string	true	Роль в системе
ConfirmedEntry	bool	false	Подтвержден ли аккаунт пользователя

Таблица 3.37 – Продолжение таблицы 3.37

Свойство	Тип	Обязательное	Описание
ActiveUser Account	bool	false	Активен ли аккаунт пользователя

3.3 Обоснование выбора технологии проектирования

В настоящее время информационный рынок, предоставляющий программные решения в выбранной сфере, предлагает множество продуктов, которые позволяют успешно разработать веб-приложение и достигнуть поставленной цели.

3.3.1 Описание используемых технологий и языков программирования

В процессе разработки веб-приложения используются различные программные средства и языки программирования, каждый из которых применяется для выполнения специфических задач, необходимых для реализации проекта.

3.3.2 Язык программирования JavaScript

Выбор языка программирования JavaScript для разработки микросервисов также обосновывается несколькими ключевыми преимуществами этого языка. Прежде всего, JavaScript является одним из самых популярных языков программирования, широко используемым в веб-разработке. Его широкое распространение обусловлено его универсальностью и возможностью использования как на стороне клиента (браузер), так и на стороне сервера (Node.js).

Второе важное преимущество JavaScript - его высокая гибкость и динамичность. JavaScript позволяет быстро создавать интерактивные и динамические пользовательские интерфейсы, что особенно важно для микросервисов, предоставляющих данные веб-приложениям.

Третье преимущество JavaScript - его асинхронная природа. JavaScript поддерживает асинхронное программирование, что позволяет эффективно обрабатывать большие объемы запросов и взаимодействовать с базами данных и другими внешними ресурсами без блокировки выполнения программы.

Наконец, JavaScript обладает обширной экосистемой библиотек и фреймворков, таких как Express.js для серверной разработки, React.js и Vue.js для разработки пользовательских интерфейсов, что делает его привлекательным выбором для создания микросервисных архитектур.

В целом, JavaScript представляет собой мощный инструмент для разработки микросервисов, обеспечивая высокую гибкость, производительность и возможность масштабирования системы.

3.3.3 Язык программирования C#

Выбор языка программирования C# для разработки микросервисов обусловлен несколькими ключевыми преимуществами этого языка. Прежде всего, C# известен своей высокой производительностью, что обеспечивает быстрое выполнение программы. Второе важное преимущество - масштабируемость. С помощью C# легко создавать системы, способные масштабироваться в соответствии с растущими потребностями. И наконец, кросс-платформенность делает C# универсальным языком программирования, что позволяет разрабатывать приложения, работающие на различных операционных системах. Все эти факторы совместно обеспечивают быструю разработку надежной и масштабируемой системы при использовании C#.

3.3.4 HTML

HTML является стандартным языком разметки, широко используемым для создания структуры веб-страниц. Выбор HTML для разработки микросервисов также имеет свои преимущества.

Во-первых, HTML является основным строительным блоком веб-страниц и обеспечивает их структуру и семантику. Это позволяет легко со-

здавать пользовательские интерфейсы и представления для микросервисов, делая их интуитивно понятными и удобными для пользователей.

Во-вторых, HTML легко интегрируется с другими технологиями веб-разработки, такими как CSS для стилизации и JavaScript для создания интерактивных элементов. Это позволяет создавать более сложные и функциональные веб-приложения на основе микросервисной архитектуры.

Третье преимущество HTML - его доступность и поддержка во всех современных браузерах. Это обеспечивает широкий охват аудитории и упрощает развертывание и поддержку веб-приложений на основе микросервисов.

Наконец, HTML обладает простым и интуитивно понятным синтаксисом, что делает его доступным даже для новичков в веб-разработке. Это упрощает процесс создания и поддержки микросервисов, особенно при работе в команде.

В целом, HTML является важным инструментом для создания пользовательских интерфейсов и представлений в микросервисной архитектуре. Его простота, доступность и гибкость делают его привлекательным выбором для разработки веб-приложений любого уровня сложности.

3.3.5 React

React - это JavaScript-библиотека, разработанная для создания пользовательских интерфейсов веб-приложений. Выбор React для разработки микросервисов также имеет свои преимущества.

Во-первых, React предоставляет удобный и эффективный способ создания динамических пользовательских интерфейсов. Он использует компонентный подход, позволяя разбивать пользовательский интерфейс на множество небольших и переиспользуемых компонентов. Это делает код более чистым, модульным и легко поддерживаемым.

Во-вторых, React обеспечивает высокую производительность благодаря виртуальному DOM. Виртуальный DOM позволяет React оптимизировать обновление пользовательского интерфейса, перерисовывая только те компоненты, которые действительно изменились. Это делает React идеальным вы-

бором для создания микросервисов с высокой производительностью и быстрой отрисовкой.

Третье преимущество React - его обширная экосистема. Существует множество дополнительных библиотек и инструментов, таких как Redux для управления состоянием, React Router для навигации в приложении, и многие другие, которые облегчают разработку микросервисов на основе React.

React представляет собой мощный инструмент для разработки микросервисов, обеспечивая высокую производительность, модульность и широкие возможности для создания динамичных пользовательских интерфейсов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лукин, В.Н. Введение в проектирование баз данных. [Текст] / В.Н. Лукин. - М.: Вузовская книга, 2015. - 144 с. – ISBN 978-5-507-48747-9. – Текст : непосредственный.
2. Фримен, А. Практикум по программированию на JavaScript / А. Фримен. – Москва : Вильямс, 2013. – 960 с. – ISBN 978-5-8459-1799-7. – Текст : непосредственный.
3. Клеппман, М. Высоконагруженные приложения. Программирование, масштабирование, поддержка / М. Клеппман. – Санкт-Петербург : Питер, 2018. – 640 с. – ISBN 978-5-44-610512-0. – Текст : непосредственный.
4. Веру, Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – Санкт-Петербург : Питер, 2016. – 336 с. – ISBN 978-5-496-02082-4. – Текст : непосредственный.
5. Мартин, Р. Чистый код. Создание, анализ и рефакторинг / Р. Мартин. – Санкт-Петербург : Питер, 2020. – 464 с. – ISBN 978-5-4461-0960-9. – Текст : непосредственный.
6. Баланов, А. Построение микросервисной архитектуры и разработка высоконагруженных приложений. Учебное пособие / А. Баланов. – Москва : Лань, 2024. – 244 с. – ISBN 978-5-507-48747-9. – Текст : непосредственный.
7. Дэккетт, Д. HTML и CSS. Разработка и создание веб-сайтов / Д. Дэккетт. – Москва : Эксмо, 2014. – 480 с. – ISBN 978-5-699-64193-2. – Текст : непосредственный.
8. Макфарланд, Д. Большая книга CSS / Д. Макфарланд. – Санкт-Петербург : Питер, 2012. – 560 с. – ISBN 978-5-496-02080-0. – Текст : непосредственный.
9. Порселло, Б. React. Современные шаблоны для разработки приложений / Б. Порселло. – Санкт-Петербург : Питер, 2022. – 320 с. – ISBN 978-5-4461-1492-4. – Текст : непосредственный.

10. Титтел, Э. HTML5 и CSS3 для чайников / Э. Титтел, К. Минник. – Москва : Вильямс, 2016 – 400 с. – ISBN 978-1-118-65720-1. – Текст : непосредственный.