

# ОГЛАВЛЕНИЕ

СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ .....	8
ВВЕДЕНИЕ .....	9
1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ .....	10
1.1. Общая характеристика предметной области .....	10
1.2. Определение объекта и предмета исследования .....	12
1.3. Обзор существующих разработок .....	13
1.3.1. Система АЛС-АРС Московского метрополитена .....	13
1.3.2. Система управления транспортом города Москвы .....	14
1.3.3. Телеавтоматическая система «Старт» .....	15
1.3.4. Система сигнализации АЛСН .....	16
1.3.5. Результаты анализа существующих разработок .....	16
1.4. Постановка задачи на разработку ИС .....	17
1.5. Техническое задание .....	19
1.5.1. Общие сведения .....	19
1.5.2. Назначение и цели создания системы .....	21
1.5.3. Характеристики объекта автоматизации .....	21
1.5.4. Требования к системе .....	21
1.5.5. Состав и содержание работ по созданию и развертыванию системы .....	23
1.5.6. Порядок контроля и приемки системы .....	23
1.5.7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие .....	23
1.5.8. Требования к документированию .....	24
1.5.9. Источники разработки ТЗ .....	24
Выводы по разделу .....	24
2. АНАЛИТИЧЕСКИЙ РАЗДЕЛ .....	25
2.1. Информационное обеспечение ИС .....	25
2.1.1. Информационная модель и ее описание .....	25
2.1.2. Нормативно-справочная и входная информация .....	27
2.1.3. Результатная информация .....	27

2.2. Определение основных требований к ИС.....	27
2.2.1. Требования к функциональным характеристикам.....	28
2.2.2. Требования к надежности.....	29
2.2.3. Условия эксплуатации.....	30
2.2.4. Требования к составу и параметрам аппаратных средств.....	30
2.2.5. Требования к информационной совместимости.....	31
Выводы по разделу.....	31
3. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ.....	32
3.1. Обоснование выбора средств разработки.....	32
3.1.1. Выбор локального сервера.....	32
3.1.2. Выбор среды разработки.....	33
3.1.3. Выбор СУБД.....	34
3.2. Описание реализации клиентской и серверной частей ИС.....	34
3.3. Описание реализации базы данных ИС.....	35
3.4. Обеспечение информационной безопасности при эксплуатации ИС.....	36
3.5. Описание работы пользователей с ИС.....	37
Выводы по разделу.....	39
4. ЭКОНОМИЧЕСКИЙ РАЗДЕЛ.....	40
4.1. Организация и планирование работ по теме.....	40
4.1.1. Организация работ.....	41
4.1.2. График проведения работ.....	42
4.2. Расчет стоимости проведения работ по теме.....	43
4.2.1. Материалы, покупные изделия и полуфабрикаты.....	43
4.2.2. Специальное оборудование.....	44
4.2.3. Основная заработная плата.....	44
4.2.4. Дополнительная заработная плата.....	44
4.2.5. Страховые отчисления.....	44
4.2.6. Командировочные расходы.....	45
4.2.7. Накладные расходы.....	45
4.2.8. Прочие расходы.....	45

4.2.9. Подсчет себестоимости проекта.....	45
Выводы по разделу.....	46
ЗАКЛЮЧЕНИЕ .....	47
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	48
ПРИЛОЖЕНИЯ.....	52
Приложение А — Логическая модель базы данных.....	53
Приложение Б — Листинг программных компонентов.....	54
Приложение В — Презентация к ВКР.....	58

## СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

- ИС — информационная система;
- БД — база данных;
- ИИ — искусственный интеллект;
- АЛС-АРС — автоматическая локомотивная сигнализация с автоматическим регулированием скорости;
- СЦБ — устройства сигнализации, централизации и блокировки;
- АЛСН — автоматическая локомотивная сигнализация непрерывного действия;
- ДАУ-АРС — система АРС с ДАУ (ДАУ — дублирующее автономное устройство);
- ЭПК — электропневматический клапан;
- ПО — программное обеспечение;
- УКБМ — устройство контроля бдительности машиниста;
- ГОСТ — государственный стандарт;
- ЕСПД — единая система программной документации;
- ГБ — гигабайт;
- ОС — операционная система;
- ВКР — выпускная квалификационная работа;
- ТЗ — техническое задание;
- СУБД — система управления базами данных;
- MS — Microsoft (компания «Майкрософт»);
- SSMS — интегрированная среда SQL Server Management Studio;
- ОЗП — основная заработная плата;
- ДЗП — дополнительная заработная плата;
- ПР — прочие расходы.

## **ВВЕДЕНИЕ**

Темой настоящей выпускной квалификационной работы является «Проектирование информационной системы управления общественным транспортом». Актуальность темы обусловлена стремительным развитием общественного транспорта в стране и повышением нагрузок на него, а также существующими проблемами его управления. Причинами возникновения существующих проблем управления является слабая цифровизация и автоматизация процессов. Большинство процессов управления общественным транспортом практически не автоматизированы. Особенно это касается городов, где транспорт практически не развивался со времен распада СССР. Вытекающими последствиями являются высокие убытки и неудобство его использования, что вынуждает большое количество людей пересаживаться на личные автомобили. Это сильно повышает загруженность улиц и дорог городов.

Цифровизация и автоматизация некоторых процессов управления общественным транспортом должны не только увеличить его доступность и снизить загруженность городских улиц и дорог, но и простимулировать обновление парка транспортных средств.

# **1. ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ**

## **1.1. Общая характеристика предметной области**

Информационная система управления общественным транспортом (далее — ИС управления транспортом) рассматривается под углом двух предметных областей: раздела статистики, занимающегося исследованием процессов урбанистики и социального поведения в общем и общественным транспортом в частности, а также цифровизации процессов управления транспортными средствами и инфраструктурой. Далее проведен анализ каждой из предметных областей. Анализ начинается с изучения определений и трактовок, относящихся к ним (предметным областям) понятий.

Для того, чтобы найти связь между городскими процессами и цифровыми технологиями и данными, можно ознакомиться со следующими утверждениями и определениями, относящимися к современным городам.

Цифровой (умный) город: «умный город — это город, в котором цифровые технологии и инновационные системы расширяют и увеличивают возможности решения городских проблем и улучшают показатели городской жизни» [1];

Горожанин как генератор данных: «В поисках информации и просто находясь на связи, горожане автоматически становятся генератором данных. Они оставляют цифровые следы в виде интернет-запросов, маршрутов передвижения, персональных данных... Горожане сегодня — единица данных» [1];

Новая эпоха урбанизма — сетевой урбанизм: «сейчас начинается новая эпоха, когда урбанизм, учитывающий данные, все больше дополняется и замещается сетевым урбанизмом, основанном на данных» [1];

Большие данные и их переизбыток в связи с массовой цифровизацией городских процессов: «Ситуация относительно недостатка данных сменяется их переизбытком. Это в первую относится к оперативным городским данным, поскольку традиционная инфраструктура города, включая транспорт (шоссе,

железные дороги, автобусные маршруты, а также транспортные средства)...» [1];

Зачем использовать данные в управлении городскими процессами: «Один из ключевых доводов в пользу подхода к муниципальному управлению, основанного на работе с непрерывными потоками данных, заключается в том, что он предоставляет прочную фактическую основу для принятия решений, системного контроля и формирования политического курса...» [1].

Эффективно действующий общественный транспорт позволяет перевозить большое количество пассажиров, тем самым уменьшая нагрузку на транспортные артерии города. Большое количество автомобилей бросает вызов городскому транспорту. Поэтому он должен отвечать всем современным тенденциям и стимулировать жителей предпочитать его личным автомобилям.

Цифровизация в транспорте: «Транспорт в первую очередь ощущает на себе тенденцию по цифровизации технологий, т.е. это объективная необходимость проводить автоматизацию управления транспортом, повышать надежность транспортной системы и т.д... Транспортное средство должно иметь персональную идентификацию в Интернете и находиться под контролем программного обеспечения.» [2];

Что такое современный общественный транспорт?: «Современный общественный транспорт города это: автобусы, трамваи и городская электричка. Городской общественный транспорт — это разновидность пассажирского транспорта как отрасли, представляющей услуги по перевозке людей по маршрутам, которые перевозчик заранее устанавливает.» [2].

Объединить все современные урбанистические и градостроительные тенденции и цифровые технологии можно, создав многофункциональную информационную систему, при помощи которой можно будет автоматизировать процессы управления общественным транспортом от принятия решения об отправлении транспортного средства на маршрут до

переключения светофоров для беспрепятственного движения городских транспортных средств.

## **1.2. Определение объекта и предмета исследования**

В соответствии с результатами анализа предметной области можно определить объект исследования. Объектом исследования является управление общественным транспортом и объектами транспортной инфраструктуры. Предметом исследования является информационная система управления общественным транспортом, которая должна автоматически осуществлять процессы управления. Стоит отметить, что выше описанный объект является всего лишь некой абстракцией. В результате разработки данная абстрактная система должна получить определенную форму исполнения — реализацию. Данная система предназначена для регионов и муниципалитетов, в связи с чем должна соответствовать всем государственным стандартам и требованиям. Также важным аспектом является то, что системе необходимо быть универсальной: ее работа должна быть одинаково эффективна независимо от места развертывания, что выставляет к ней дополнительные требования. Ядром системы должен быть сервер с базой данных всех моделей, структур и происходящих событий. Это позволит объединить различные процессы во едино и обеспечить их эффективное взаимодействие. Так как информационная система управления общественным транспортом рассматривается под углом двух предметных областей, при ее разработке должны рассматриваться:

1. Современные урбанистические, градостроительные и транспортные тенденции, в число которых входит: создание магистральных маршрутов общественного транспорта, интеграция железных дорог в общую транспортную модель города, безбарьерная среда и т.д.;

2. Взаимная интеграция различных существующих систем управления транспортными средствами, объектами транспортной инфраструктуры, в том числе дорожной сигнализации;



3. Исследование процессов происходящих в городской среде в целом и в общественном транспорте в частности, а также рассмотрение возможности их автоматизации и цифровизации;

4. Программные инструменты реализации — актуальные языки программирования, алгоритмы и паттерны, используемые при создании различных программных компонентов, серверов и баз данных (БД), инструменты анализа больших данных а также инструменты реализующие машинное обучение и создание искусственного интеллекта (ИИ).

### **1.3. Обзор существующих разработок**

Был выполнен анализ некоторых наиболее эффективных систем, которые отвечают за тот или иной транспортный процесс. К таким системам относятся: №1 — Система АЛС-АРС [3] и устройства СЦБ [4], действующие в Московском метрополитене; №2 — Система контроля за соблюдением расписания и за обстановкой на дорогах города (центр управления транспортом города Москвы); №3 — Телеавтоматическая система управления транспортом «Старт»; №4 — Система АЛСН [5] и устройства СЦБ [4], использующиеся на железных дорогах. Указанные выше системы предназначены для выполнения различных задач, однако могут быть взаимно интегрированы.

#### **1.3.1. Система АЛС-АРС Московского метрополитена**

Система АЛС-АРС предназначена для повышения безопасности движения поездов метрополитена. АЛС-АРС отвечает за управление локомотивной сигнализацией. Она следит за соблюдением скоростного режима, способна автоматически снизить скорость поезда или полностью остановить его. Система передает информацию о максимально допустимой скорости движения на участке в данный момент времени на пульт, находящийся в кабине машиниста.

На линиях метро, на которых система АЛС-АРС является основным средством сигнализации при движении поездов, она работает совместно с

дублирующими автономными устройствами (ДАУ-АРС). В такой ситуации у машиниста в кабине имеется информация о скоростном режиме не только на текущем (основная сигнализация), но и на следующем (предупредительная сигнализация). Для передачи информации о допустимой скорости движения используются частотные кодовые сигналы, подаваемые на путевой рельс. Каждой частоте соответствует своя максимальная скорость движения: 75 Гц — 80 (90) км/ч; 125 Гц — 70 (75) км/ч; 225 Гц — 40 км/ч; 275 Гц — сигнал остановки; 325 Гц — может иметь разное значение в зависимости от установленной системы. Данная частота может передавать следующую информацию: направление движения с максимальной скоростью 40 км/ч, «равенство скоростей»: скорость напереди лежащем участке больше или равна скорости на текущем. Указателем в кабине управления электропоездом подаются сигналы: разрешено движение со скоростью не выше указанной; «немедленно остановитесь»; нет частоты или отсутствие частоты — «немедленно остановитесь». Еще одной важной составляющей является автостоп. Автостоп — это комплекс устройств на электропоезде и пути, который в случае проезда машинистом запрещающего сигнала производит экстренное торможение поезда. На пути находится специальное устройство — путевая скоба, которая при запрещающем сигнале светофора находится в поднятом положении. Электропоезда оборудованы срывным клапаном, рамка которого механически контактирует с поднятой путевой скобой, что приводит к срыву ЭПК и экстренному торможению электропоезда.

### **1.3.2. Система управления транспортом города Москвы**

Система контроля за соблюдением расписания и за обстановкой на дорогах города предназначена для упрощения процессов управления общественным транспортом и дорожным движением в таком большом городе как Москва. Она позволяет следить за соблюдением компаниями-перевозчиками установленного расписания, за общей обстановкой на дорогах города и быстро реагировать на внештатные ситуации.

Принцип работы системы довольно банален: на городских улицах установлено огромное количество дорожных камер видеонаблюдения. На всех маршрутных транспортных средствах установлено несколько камер а также устройства спутниковой геолокации. В центре управления транспортом города Москвы ежедневно дежурят более восьмидесяти сотрудников. В их обязанности входит контроль за соблюдением установленного для маршрутных транспортных средств расписания, за дорожной ситуацией в городе, а также реагирование на сообщения о дорожно-транспортных происшествиях и поломках транспортных средств. На данный момент нет открытой информации о том, собираются ли большие данные по перемещению, скорости и пассажиропотоку на различных маршрутах общественного транспорта, однако можно предположить сбор данных имеет место быть, так как эти данные могут помочь в проектировании новых маршрутов и расписания движения, а также улучшении уже существующих.

### **1.3.3. Телеавтоматическая система «Старт»**

Применение телеавтоматической системы управления транспортом «Старт» обеспечивает увеличение эффективности использования дорожно-уличной сети, снижение задержек транспорта на перекрестках а также повышение скорости сообщения и безопасности движения.

Прикладное ПО включает в себя клиент-серверные приложения, клиент и БД. Данное обеспечение позволяет управлять дорожной сигнализацией, которая содержит большое количество светофоров, камерами наружного видеонаблюдения, установленным на улицах города и другими устройствами. Система предоставляет возможность контролировать остановку на дорогах: загруженность дорог, дорожно-транспортные происшествия и т.д. Операторы системы через ее клиентскую часть могут дистанционно управлять устройствами дорожной инфраструктуры, тем самым разгружая улицы города. Например, оператор может организовать так называемую «зеленую волну»: светофоры на загруженном участке дороге будут поочередно переключаться

на разрешающий сигнал, что позволит большому потоку транспортных средств быстро проехать загруженный участок дороги.

#### **1.3.4. Система сигнализации АЛСН**

АЛСН — система сигнализации на рельсовом транспорте непрерывного действия. Она передает сигнальные показания на пост управления подвижного состава. Отличительная особенность системы непрерывного действия в том, что информация о сигнале светофора поступает непрерывно.

На железнодорожных маршрутах России и стран СНГ применяются два разных способа кодирования сигналов АЛСН: импульсный числовой и частотный. Импульсный числовой сигнал передает серию импульсов переменного тока (при таком методе число и длительность этих импульсов). Частотный сигнал, в свою очередь, непрерывно передает переменный ток (при таком методе информацию несет частота). Например такой принцип действия применяется в системе АЛС-АРС Московского метрополитена. На наземных железных дорогах в большинстве случаев применяется импульсное кодирование. Локомотивный светофор в системе АЛСН поддерживает пять сигнальных показаний: зеленый огонь обозначается тремя импульсами в кодовом цикле; желтый огонь — два импульса; одновременно красный и желтый огонь — один импульс; красный огонь — при отсутствии импульсов после одновременно красного и желтого; белый огонь — при отсутствии импульсов после зеленого или желтого. Для увеличения уровня безопасности движения поездов по железной дороге в АЛСН предусмотрены автостоп, устройства контроля скорости, а также контроля бдительности машиниста. Контроль бдительности машиниста проводится с помощью специального устройства — УКБМ [6].

#### **1.3.5. Результаты анализа существующих разработок**

В результате анализа можно выявить полезный функционал существующих систем. Данный функционал может быть использован при разработке. Например, опыт разработки и эксплуатации систем АЛС можно

использовать для организации движения наземного рельсового транспорта. В системе контроля за соблюдением расписания и за обстановкой на дорогах города большинство решений принимается ее операторами. Необходимо автоматизировать часть этих процессов. Это повысит эффективность работы общественного транспорта. Телеавтоматическая система «Старт» используется для управления транспортными потоками на самых загруженных участках города. Подобный функционал стоит использовать на всех улицах и дорогах города и интегрировать процессы управления транспортными потоками с процессами управления общественным транспортом.

Стоит отметить, что рассматриваемые системы реализуют лишь часть необходимых задач в различных узких областях. В связи с этим не корректно сравнивать из друг с другом. Что касается системы АЛС-АРС Московского метрополитена, то она является частным случаем АЛС. Об этом уже было написано в принципе работы системы АЛСН.

Исследуемые системы на данный момент мало распространены в нашей стране, поэтому еще не существует определенной, единственно верной стратегии по созданию ИС управления общественного транспорта, которую можно было бы развернуть и эффективно использовать на всем просторе нашей родины.

#### **1.4. Постановка задачи на разработку ИС**

Разрабатываемая информационная система в первую очередь должна автоматизировать часть процессов управления общественным транспортом. Некоторые задачи, которые на данным момент лежат на плечах операторов, должны выполняться автоматически. Для этих целей можно использовать алгоритмы машинного обучения. Именно поэтому очень важно собирать данные со всех возможных транспортных объектов. Со временем они могут увеличить эффективность работы общественного транспорта. Для автоматизации процессов управления необходимо создать цифровую

транспортную модель. Эта модель должна представлять собой цифровой представление всех транспортных объектов, их параметров и функций. Цифровые представления объектов должны отражать их текущее состояние (для транспортного средства — это его скорость, местоположение, отставание от графика или его опережение, для светофора — это текущая фаза, то через сколько она будет переключена, для остановки — это то, сколько пассажиров на ней садится на тот или иной маршрут, в какой момент времени это происходит и т.п.) Через цифровые представления транспортных объектов будет производиться управление ими. Например, в случае, если транспортное средство отстает от графика, на пульт водителя будет передаваться сообщение о том, что необходимо ускориться, светофоры на его пути, по возможности, будут переключаться на разрешающий сигнал и т.д.

Ниже описан функционал, которым должна обладать разрабатываемая информационная система:

1. Возможность создания цифровой транспортной модели населенного пункта, которая будет включать в себя все маршруты, транспортные средства и управляемые инфраструктурные объекты (светофоры, системы АЛС и т.д.);
2. Сбор данных со всех указанных выше объектов, для их последующего изучения, обработки с целью улучшения существующей транспортной модели;
3. Развертывание транспортной модели для автоматического управления;
4. Хранение собираемых больших данных;
5. Возможность управления цифровой транспортной моделью дистанционно человеком.

Из всего сказанного выше, задачи, решаемые системой, целесообразно разделить:

- На задачи, относящиеся к проектированию транспортной модели: система должна предоставлять возможность цифровизации существующей транспортной модели того или иного населенного пункта;
- На задачи, относящиеся к развертыванию модели и ее управлению: система должна автоматизировать процессы управления транспортом города, а также предоставить возможность дистанционного управления, теми процессами которые на данный момент не возможно автоматизировать.

Также при создании системы важно обратить внимание на то, что все ее части должны быть взаимно согласованы и не препятствовать работе друг друга. Особое внимание стоит обратить на сбор, первичную обработку и хранение больших данных, которые будут собираться после развертывания системы.

## **1.5. Техническое задание**

Для написания технического задания разрабатываемого проекта был выбран ГОСТ 34.602-2020 [16], потому что этот ГОСТ больше подходит для создания ТЗ ИС управления общественным транспортом. Техническое задание написанное по нему будет содержать такие важные пункты как «Состав и содержание работ по созданию системы», «Характеристика объектов автоматизации» а также «Требования к системе».

### **1.5.1. Общие сведения**

#### **Полное наименование системы:**

Информационная система управления общественным транспортом.

#### **Шифр темы или шифр (номер) договора:**

Не устанавливается.

#### **Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты:**

Не устанавливаются.

#### **Перечень документов, на основании которых создается система:**

- Документация и руководства к платформы .NET Core;

- Документация и руководства к языку программирования C#;
- Документация и руководства к платформе ASP.NET;
- Документация и руководства к фреймворку Microsoft Entity Framework Core;
- Документация и руководства к языку запросов T-SQL;
- Документация и руководства к платформе WPF;
- Руководство по использованию многофункциональной среды разработки Microsoft Visual Studio;
- Руководство по использованию интегрированной среды управления Microsoft SQL Server Management Studio;

Обоснование выбора среды разработки, платформ и фреймворков приводится в технологическом разделе.

#### **Плановые сроки начала и окончания работ по созданию системы:**

- Дата начала разработки системы: 10 февраля 2023 года;
- Плановый срок разработки системы: 3 месяца;
- Плановый срок разработки системы: 10 мая 2023 года.

Календарный график реализации всех поставленных задач приведен в экономическом разделе.

#### **Сведения об источниках и порядке финансирования работ:**

Все сведения, касающиеся финансирования работ, приведены в экономическом разделе.

#### **Порядок оформления и предъявления заказчику результатов работ по созданию системы (ее частей), по изготовлению и наладке отдельных средств (технических, программных, информационных) и программно-технических (программно-методических) комплексов системы:**

Работы по проектированию и реализации разрабатываемой системы выполняются поэтапно в соответствии с календарным планом проекта. Календарный план проекта и порядок выполнения работ приведены в экономическом разделе.



## **1.5.2. Назначение и цели создания системы**

### **Назначение разрабатываемой системы:**

Разрабатываемая система предназначена для управления общественным транспортном (транспортными средствами и объектами транспортной и дорожной инфраструктуры).

### **Цели создания системы:**

Цель разработки — создание системы, которая автоматизирует процессы управления общественным транспортом и создаст новые возможности для улучшения доступности транспорта и удобства его использования.

## **1.5.3. Характеристики объекта автоматизации**

### **Краткие сведения об объекте автоматизации или ссылки на документы, содержащие такую информацию:**

Автоматизация заключается в цифровизации процессов управления транспортными средствами и объектами транспортной инфраструктуры. Система позволит улучшить качество работы общественного транспорта.

### **Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды:**

Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды не рассматриваются. Требования не устанавливаются.

## **1.5.4. Требования к системе**

### **Требования к системе в целом:**

Разрабатываемая система должна иметь трехуровневую архитектуру. В систему предлагается выделить следующие функциональные подсистемы:

- Подсистема проектирования;
- Подсистема управления;
- Подсистема развертывания.

### **Требования к функциям (задачам), выполняемым системой:**

В таблице ниже описаны требования к функциям подсистемы проектирования транспортной модели (Таблица 1.1).

Таблица 1.1 — Функции подсистемы проектирования транспортной модели.

Функция	Задача
Создание транспортной модели.	Создавать проект транспортной модели населенного пункта для последующего добавления на нее транспортных объектов и дальнейшего ее развертывания.
Создание маршрутов и их трассировок.	Через пользовательский интерфейс предоставлять возможность добавлять маршруты транспорта в разрабатываемую модель.
Добавление инфраструктурных объектов.	Через пользовательский интерфейс предоставлять возможность добавлять объекты транспортной инфраструктуры в разрабатываемую модель.
Добавление транспортных средств.	Через пользовательский интерфейс предоставлять возможность добавлять транспортные средства в разрабатываемую модель.
Экспорт транспортной модели для последующего ее (модели) развертывания.	Через пользовательский интерфейс предоставлять возможность экспортировать транспортную модель для развертывания на сервере и эксплуатации.

В таблице ниже описаны требования к функциям подсистемы управления транспортной модели (Таблица 1.2).

Таблица 1.2 — Функции подсистемы управления транспортной моделью.

Функция	Задача
Выгрузка транспортной модели на сервер информационной системы.	Выгружать экспортированную транспортную модель на сервер информационной системы и инициировать запуск ее функционирования.
Отслеживание текущей обстановки через развернутую транспортную модель.	Через пользовательский интерфейс предоставлять возможность отслеживания текущего состояния транспортной модели.
Отслеживание статистики в реальном времени.	Через пользовательский интерфейс предоставлять возможность просмотра статистики загруженности маршрутов в целом, загруженность конкретных остановок и участков.
Вынужденное управление транспортными объектами.	Через пользовательский интерфейс предоставлять возможность посылать сообщения на объекты или менять их состояние в случае необходимости.

В таблице ниже описаны требования к функциям подсистемы развертывания транспортной модели (Таблица 1.3).

Таблица 1.3 — Функции подсистемы развертывания транспортной модели.

Функция	Задача
Работа хранилища с базой данных информационной системы.	Реализовывать взаимодействие с базой данных: добавление, обновление и удаление записей в базе данных.
Сбор статистики транспортных объектов для улучшения транспортно модели и обучения.	Реализовывать сбор статистики с транспортных средств и объектов транспортной инфраструктуры для исследования и обучения моделей машинного обучения.
Управление транспортными объектами при помощи моделей машинного обучения.	Реализовывать управление транспортными объектами при помощи алгоритмов машинного обучения, на основе данных собираемой статистики.
Управление транспортными объектами при помощи алгоритмов, не относящимся к машинному обучения.	Реализовывать управление транспортными объектами при помощи математических и логических алгоритмов, которые не относятся к алгоритмам машинного обучения.
Предоставлять доступ к данным клиентам ИС через API.	Реализовывать доступ ко всем необходимым данным клиентам информационной системы через веб-API.

### Требования к видам обеспечения:

Требования к видам обеспечения не устанавливаются.

#### 1.5.5. Состав и содержание работ по созданию и развертыванию системы

Работы по созданию системы выполняется в три этапа:

- Разработка эскизного проекта;
- Разработка технического проекта;
- Реализация программного обеспечения.

Конкретные сроки выполнения проекта, стадий и этапов разработки информационной системы определяются и описываются в экономическом разделе ВКР.

#### 1.5.6. Порядок контроля и приемки системы

Требования к порядку контроля и приемки системы не устанавливаются.

#### 1.5.7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

Информационная системы разрабатывается в образовательных целях. Ввод системы в действие на данный момент времени не планируется. Требования не устанавливаются.

### **1.5.8. Требования к документированию**

Требования к документированию выставляются в соответствии с требованиями к ВКР. Другие требования не устанавливаются.

### **1.5.9. Источники разработки ТЗ**

Настоящее техническое задание разработано на основе следующих документов и информационных материалов:

- ГОСТ 34.602-2020 [16] «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы».
- ГОСТ 24.701-86 [17] «Надежность автоматизированных систем управления».
- ГОСТ 15150-69 [18] «Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды».

### **Выводы по разделу**

В данном разделе была описана общая характеристика предметной области, определены объект и предмет исследования, выполнен обзор уже существующих разработок. Была поставлена задача на разработку информационной системы и ее подсистем, а также было разработано техническое задание. На основе этих данных можно подробнее описать принцип работы системы, описать ее информационно обеспечение, а также определить основные требования к ней.

## **2. АНАЛИТИЧЕСКИЙ РАЗДЕЛ**

### **2.1. Информационное обеспечение ИС**

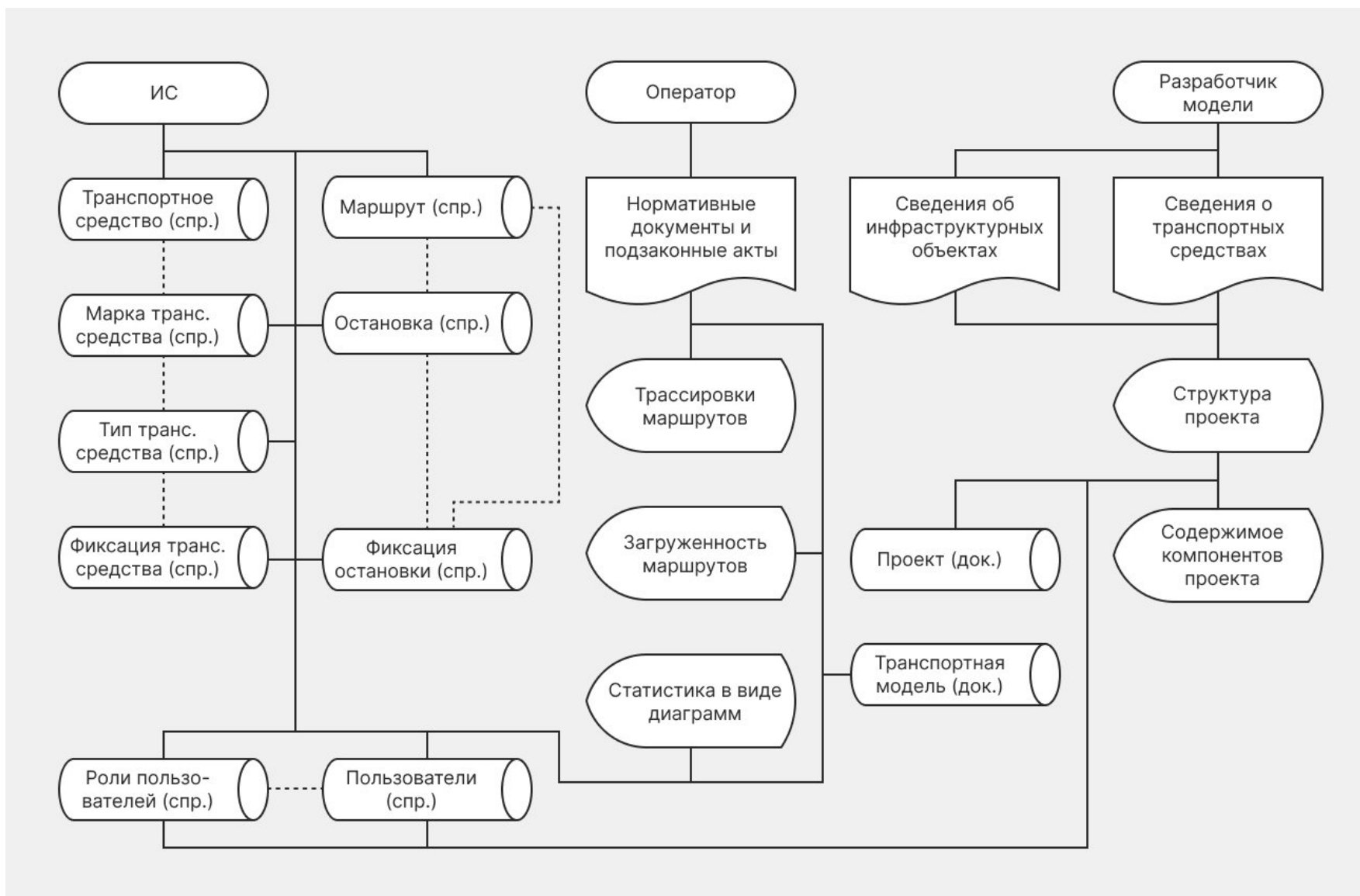
#### **2.1.1. Информационная модель и ее описание**

Информационная модель предназначена для описания потоков информации в разрабатываемой информационной системе. В ней отображается вся информация, необходимая для решения поставленной задачи. Для каждого потока в информационной модели указывается источник, пути преобразования, а также приемник. Модель отражает как состав и способы сбора исходной информации, так и распределении ее (информации) по различным задачам.

Информационная модель будет представлена в виде схемы данных (ГОСТ 19.701-90) [19]. Схема данных состоит из следующего набора компонентов:

- Символ данных (также может обозначать вид носителя данных);
- Символ процесса, который следует выполнить над данными;
- Символ линий, которые обозначают потоки данных между процессами и / или носителям данных
- Специальные символы, которые используются для упрощения процессов написания и чтения схемы;

Графическое представление информационной модели, созданной согласно предметной области изображена на рисунке 2.1.



**Рисунок 2.1 — Графическое представление информационной модели.**

### **2.1.2. Нормативно-справочная и входная информация**

Входной информацией для разрабатываемой ИС управления транспортом являются сведения о транспортных средствах (автобусах, трамваях и т.п.), существующие маршруты общественного транспорта (типы, трассировки), объекты инфраструктуры (депо, устройства дорожной сигнализации, устройства контроля занятости участков рельсового транспорта), фиксации состояния описанных выше объектов в определенный момент времени. Эти данные вносятся в систему разработчиками транспортной модели населенного пункта, в время ее создания.

### **2.1.3. Результатная информация**

Результатная информация — это графическое представления цифровой транспортной модели, состояния ее объектов для операторов информационной системы, а также команды управления, подаваемые ИС и ее операторами этим объектам. В результате выполнения процессов управления происходит принятие решений по изменению состояния транспортных объектов и сохранения фиксаций их состояний. Устройства, управляющие объектами, отправляют запросы на сервер информационной системы и получают в ответ управляющие сигналы. Данные фиксаций состояний отправляемые на сервер, должны быть сохранены в базе данных. Они предназначены для формирования статистики в виде различных графиков и цифровых значений.

## **2.2. Определение основных требований к ИС**

Разрабатываемая ИС с технической точки зрения является прикладным программным продуктом, который создается для государственных учреждений, занимающихся управлением и развитием общественного транспорта в том или ином регионе России. Из этого следует, что требования, предъявляемые к разрабатываемой системе, должны соответствовать отечественных стандартов программного обеспечения и программной документации. По этой причине определение требований, предъявляемых к

информационной системе управления общественным транспортом, будут соответствовать ГОСТ-19.001-77 ЕСПД [7].

### **2.2.1. Требования к функциональным характеристикам**

ИС управления транспортом должна реализовывать следующий функционал для проектирования цифровой транспортной модели:

- Функции для создания транспортной модели населенного пункта, которая будет включать в себя управляемые инфраструктурные объекты: Цифровая транспортная модель — модель, которая будет являться виртуальным интерфейсом, выступающим в качестве «фасада» между физическим объектом (светофором, стрелкой; транспортным средством) и субъектом — оператором;
- Функции для создания транспортной модели населенного пункта, которая будет включать маршруты передвижения различных транспортных средств: В транспортной модели маршруты передвижения транспортных средств должны быть представлены в виде трассировок по реальным координатам. Маршруты также включают такие объекты как остановки, рельсовое полотно (рельсовый транспорт), устройства управления рельсовым транспортом и т.д.;
- Функции экспорта созданного проекта, для последующего развертывания: Модель будет приводится к формату, в котором будет развертываться. Данный формат предназначен только для управления (проект проектирования транспортной модели не удаляется). Проектирование и улучшение модели можно продолжать даже после ее развертывания. После завершения проектирования транспортная модель выгружается повторно.

ИС управления транспортом должна реализовывать следующий функционал для управления транспортной моделью:

- Функционал развертывания транспортной модели: Выгруженная версия транспортной модели представляет собой иерархию экземпляров типов (классов или структур), которые содержат в себе данные о транспортных



объектах и предоставлять функции для взаимодействия с ними. Развертывание подразумевает извлечение данных из файла модели на сервер;

- Функционал управления и транспортной моделью и ее объектам в режиме реального времени: Управление транспортной моделью будет осуществляться операторами, через клиентское ПО с пользовательским интерфейсом. Данное программное обеспечение будет взаимодействовать физическими транспортными объектами, через сервер информационной системы (не напрямую). Именно для этого сервер должен поддерживать функционал развертывания модели;

- Функционал сбора данных с объектов транспортной модели для последующего ее улучшения: Сервер информационной системы должен реализовывать функционал сбора данных с транспортных объектов. Например, скорость прохождения участков дороги транспортными средствами в то или иное время суток, пассажиропоток на различных маршрутах общественного транспорта и т.п.;

Функционал формирования статистики в режиме реального времени на основе собираемых данных: Сервер должен использовать технологии анализа больших данных, в том числе машинное обучение, для последующего улучшения транспортной модели.

### **2.2.2. Требования к надежности**

Для корректной работы ИС управления транспортом устанавливаются следующие требования к надежности:

- Сервер, на котором будет развернута система, должен быть обеспечен современной антивирусной защитой и защитой от атак;
- Должен быть обеспечен вывод сообщений о возникших ошибках с кодом ошибки или другим содержимым;
- Сохранение всех возникших ошибок, для последующего изучения администраторами с целью устранения их причин.

### **2.2.3. Условия эксплуатации**

Условия эксплуатации аппаратного обеспечения, на котором будет развернута система и ее база данных, определяются учреждением, которое является конечным пользователем системы. Учреждение должно уполномочить профессионально обученных лиц, которые будут работать с системой и администрировать ее. Такие сотрудники должны уметь работать с устройствами ЭВМ и разбираться в специфике ИС.

### **2.2.4. Требования к составу и параметрам аппаратных средств**

К устройствам, на которых будет производится работа с системой а также ее администрирование, выставляются следующие требования к программно-аппаратному обеспечению:

- Процессоры средней производительности с архитектурой x86-64 [14]. Например, процессоры линейки Intel Core I3, Intel Core I5 или линейки AMD Ryzen 3, AMD Ryzen 5;
- Свободное место на жестком диске более 5 ГБ;
- Общий объем устройств оперативной памяти должен составлять не менее 4 ГБ;
- Высокоскоростное Интернет-соединение: адаптер WI-FI и порт Ethernet;
- Операционная система Windows версии 10 или более поздней версии, либо ОС семейства Linux с версией ядра 5 или выше.

К устройствам, на которых будет проведено развертывание серверной части ИС управления транспортом выставляются следующие требования к программно-аппаратному обеспечению:

- Высокопроизводительные серверные процессоры. Например, процессоры линейки Intel Xeon Gold, Intel Xeon Silver или Intel Xeon Bronze;
- Высокоскоростная серверная оперативная память типа DDR4;
- Использование дисковых массивов RAID [8] для повышения уровня сохранности данных;

- Операционная система Windows Server 2019 [9] или более поздней версии, либо ОС семейства Linux с версией ядра 5 или выше.

### **2.2.5. Требования к информационной совместимости**

При разработке ИС управления транспортом должны быть применены:

- Один или несколько высокоуровневых языков программирования, поддерживаемых платформой .NET [13] и .NET Core [10]. Например C# [13], F# или Visual Basic;
- Entity Framework [15] для организации хранилища;
- Язык запросов к серверу Microsoft SQL Server [11] и его базам данных — язык запросов T-SQL [12].

### **Выводы по разделу**

В данном разделе было описано информационное обеспечение разрабатываемой системы: информационная модель и ее описание, нормативно-справочная и входная информация, а также результатная информация. Были определены основные требования к информационной системе: требования к функциональным характеристикам, требования к надежности и условия эксплуатации, требования к составу и параметрам аппаратных средств и требования к информационной совместимости. На основе этих данных можно перейти к проектированию ИС: обосновать выбор средств разработки, описать реализацию клиентской части, базы данных. Определить требования к обеспечению информационной безопасности во время эксплуатации системы, а также описать работу пользователей с ней.

## 3. ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ

### 3.1. Обоснование выбора средств разработки

В процессе разработки программных продуктов очень важную роль играет совокупность инструментов разработки, формирующих окружение. Среда разработки — один из самых важных инструментов. Она помогает в написании и отладки программного кода, тестирования и публикации.

Для создания клиент-серверных систем необходимы следующие инструменты:

- Локальный сервер;
- Редактор кода (среда разработки);
- Система управления базами данными (СУБД).

#### 3.1.1. Выбор локального сервера

На данный момент существуют различные системы, выполняющие задачи локального сервера. Каждая из таких систем имеет свои преимущества и недостатки. В таблице ниже рассмотрены самые распространенные решения (Таблица 3.1).

Таблица 3.1 — Описание самых распространенных локальных серверов.

Достоинства	Недостатки
<b>Apache</b> — это свободное программное обеспечение для размещения веб-сервера.	
Доступность, гибкость настройки, кроссплатформенность, совместимость, масштабируемость, поддержка пользователей, функциональность.	Производительность, излишний функционал, сложная конфигурация повышает уязвимость, синтаксис файлов конфигурации.
<b>Nginx</b> создавался в качестве альтернативы Apache с целью устранить ограниченность производительности.	
Бесплатный доступ, стабильность, надежность, высокая частота обновления, гибкость за счет модульной структуры, кроссплатформенность, поддержка пользователей.	Снижение производительности на сайтах с большим трафиком, излишний функционал, сложная конфигурация повышает уязвимость.
<b>IIS</b> — веб-сервер, разработанный компанией Microsoft, который занимает второе место на рынке после Apache.	
Тесная интеграция с другими технологиями и продуктами компании, использование возможностей платформы .NET, удобное взаимодействие с настольными приложениями, широкий выбор CMS-систем.	Неполный набор возможностей веб-сервера: отсутствует поддержка SSL, URL rewriting rules, собственных настроек безопасности.

Так как программная часть информационной системы реализуется на языке программирования C#, то в качестве локального веб-сервера был выбран IIS [20] (рекомендуемый компанией Microsoft для своих платформ), так как он очень удобен в использовании при развертывания приложений платформы .NET.

### 3.1.2. Выбор среды разработки

Для разработки программных продуктов на платформе .NET чаще всего используют следующие среды разработки и редакторы кода: Microsoft Visual Studio, Microsoft Visual Studio Code, JetBrains Rider. В таблице ниже описано данное программное обеспечение (Таблица 3.2).

Таблица 3.2 — Описание подходящих средств разработки.

Название	Описание	Доступность
Microsoft Visual Studio	Microsoft Visual Studio — линейка продуктов компании Microsoft, которые включают интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данное программное обеспечение подходит для разработки различных типов приложений на большинстве современных языков программирования.	Есть бесплатная версия ПО
Microsoft Visual Studio Code	Microsoft Visual Studio Code — текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга.	Бесплатное ПО
JetBrains Rider	JetBrains Rider — кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Поддерживаются языки программирования C#, VB.NET и F#.	Платное ПО (пробный 30-дневный период)

Для разработки программного продукта выбрана среда Microsoft Visual Studio (Microsoft Visual Studio Community 2022 Preview) в связи со спецификой разрабатываемой системы. Данная среда разработки имеет большое количество шаблонов различных проектов на платформу .NET, что даст быстрый старт в разработке информационной системы.

### **3.1.3. Выбор СУБД**

Выбор системы управления базами данных [21] и сервера база — это очень важный аспект подготовки к разработке информационной системы. На данный момент существует большое количество различных систем со своими достоинствами и недостатками. Самым удобным и эффективным вариантом для проектов реализуемых на платформе .NET будет выбор сервера базы данных Microsoft SQL Server. Компания Microsoft предоставляет большое количество компонентов для интеграции баз данные в разрабатываемое приложение.

Для управления сервером Microsoft SQL Server и базами данных размещаемых на нем отлично подойдет ПО Microsoft SQL Server Management Studio (MS SSMS). Это интегрированная среда для управления любой инфраструктурой SQL, от SQL Server до баз данных SQL Azure. SSMS предоставляет средства для настройки, наблюдения и администрирования экземпляров SQL Server и баз данных.

## **3.2. Описание реализации клиентской и серверной частей ИС**

Разрабатываемая система состоит из трех подсистем:

- Подсистема проектирования транспортной модели — подсистема, предназначенная для разработки транспортной модели, включающей объекты транспортной инфраструктуры, транспортные средства, а также маршруты и их трассировки;
- Подсистема управления транспортной моделью — подсистема, предназначенная для управления созданной транспортной моделью, анализа собираемой статистики, а также принятия решений в случае возникновения внештатных ситуаций;
- Подсистема развертывания транспортной модели — подсистема, на которой развертывается цифровая транспортная модель (Приложение Б.2, Б.3). Данная подсистема является сервером. В ней организуется сбор статистики и автоматическое принятие решений в режиме реального времени.

Подсистема управления взаимодействует с этой подсистемой напрямую, через отправку запросов (Приложение Б.4, Б.5).

Также в разрабатываемой системе предусматривается так называемое ядро. Ядро не является полнофункциональной подсистемой. Оно реализует общий функционал, относящийся к двум или всем трем подсистемам.

Структурная схема ИС изображена на рисунке 3.1.

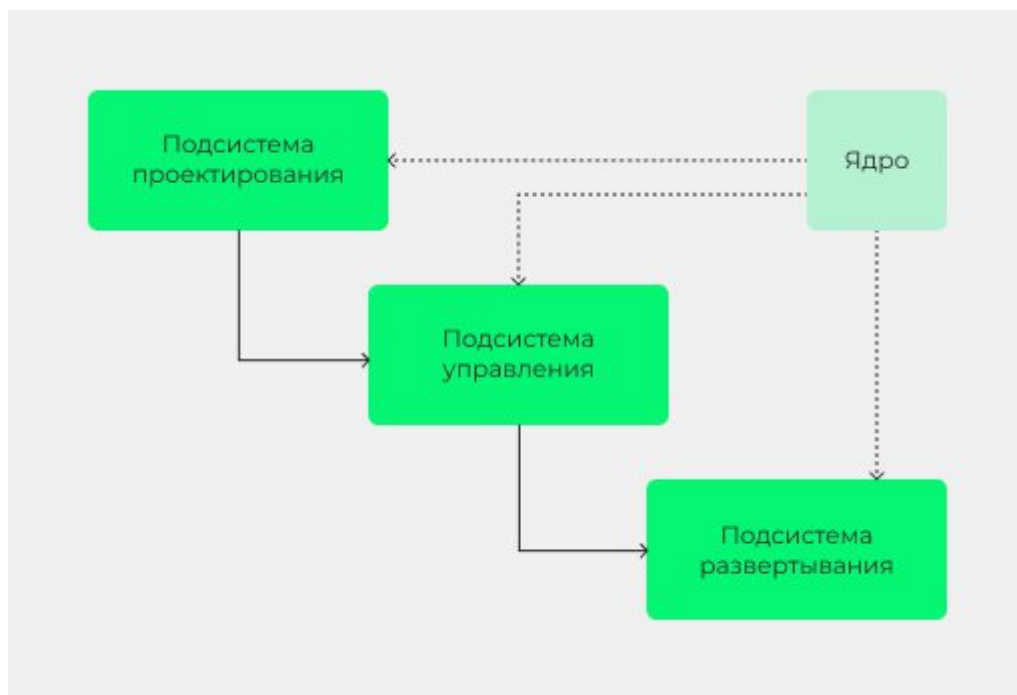


Рисунок 3.1 — Структурная схема ИС.

### 3.3. Описание реализации базы данных ИС

При проектировании базы данных создаются ее физическая и логические модели. Для создания физической модели базы данных используются UML-диаграммы [22]. После того, как физическая модель была создана, создается логическая. Для создания логической модели используется диаграмма классов (Приложение А). Диаграмма классов создается в среде разработки Microsoft Visual Studio.

Для создания базы данных на основе описанных классов используется Entity Framework. Для установки пакетов используется менеджер пакетов NuGet В проект хранилища устанавливаются следующие пакеты:

- Microsoft.EntityFrameworkCore — основной пакет;

- Microsoft.EntityFrameworkCore.SqlServer — дополнительный пакет для взаимодействия сервером Microsoft SQL Server;
- Microsoft.EntityFrameworkCore.Tools — дополнительный пакет, предоставляющий функционал работы через консоль диспетчера пакетов.

Самым простым вариантом создания таблиц в базе данных являются миграции. Этот вариант позволяет не использовать язык запросов T-SQL напрямую. Перед выполнением миграции необходимо описать контекст базы данных (Приложение Б.1, Б.3), содержащий наборы всех сущностей (классов). После того как контекст создан необходимо выполнить две команды:

- Команда, создающая миграцию;
- Команда, обновляющая базу данных (если БД не создана, то она создается на основе первой миграции).

Миграция представляет собой сгенерированный код C#, который в процессе транслируется в код на языке запросов T-SQL. Команды выполняемые для создания базы данных приведены в ниже (Листинг 3.1).

*Листинг 3.1 — Команды создания миграции и обновления БД.*

```
Add-Migration Initial
...
Update-Database
...
```

После того как база данных создана можно посмотреть результат. Для этого необходимо открыть окно обозревателя объектов SQL Server. На локальном сервере будет присоединена созданная база данных.

### **3.4. Обеспечение информационной безопасности при эксплуатации ИС**

Для обеспечения информационной безопасности разрабатываемой системы [23] необходимо использовать современную защиту от вирусов и атак. Предлагается использование Kaspersky Endpoint Security для бизнеса. Данное программное обеспечение выполняет следующие функции:



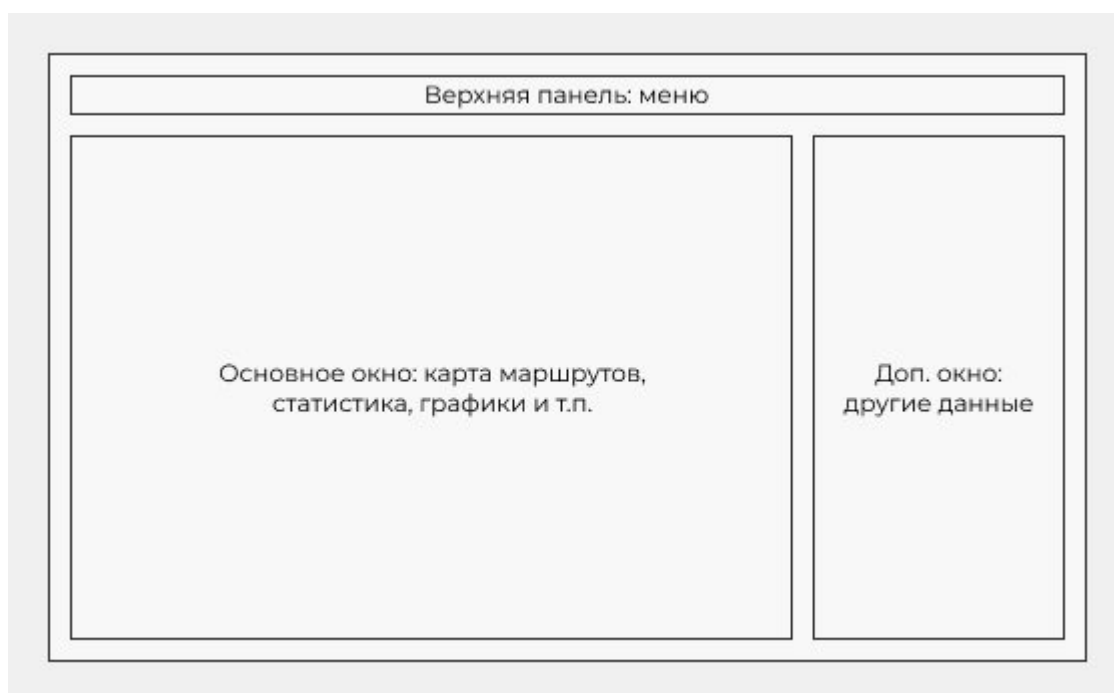
- Обнаружение и исправление уязвимостей для уменьшения поверхности атаки;
- Экономия времени за счет автоматизации задач развертывания операционной системы и программного обеспечения;
- Централизованное управление системой защиты из единой облачной или веб-консоли;
- Шифрование данных для предотвращения утечки данных с потерянного устройства.

### **3.5. Описание работы пользователей с ИС**

Пользователи информационной системы делятся на три категории:

- Разработчики транспортной модели — пользователи, которые занимаются разработкой и улучшением транспортной модели для последующего ее развертывания в подсистеме развертывания. Разработчики выполняют свои задачи через пользовательский интерфейс подсистемы проектирования транспортной модели;
- Администраторы — пользователи, которые занимаются управлением транспортной моделью, принятием решений в случае нештатных ситуаций и контролем за корректностью работы системы через пользовательский интерфейс подсистемы управления;
- Инфраструктурные объекты и транспортные средства — это неодушевленные пользователи системы. Они должны взаимодействовать с ИС и выполнять действия на основе принятых ею решений.

Разработчики и администраторы (операторы) информационной системы будут взаимодействовать с ней через пользовательский интерфейс подсистем. Интерфейс представляет собой набор выпадающих меню, вкладок и окон. Пример макета пользовательского интерфейса подсистем проектирования и управления изображен на рисунке 3.2.



**Рисунок 3.2 — Макет пользовательского интерфейса.**

О принципе взаимодействия инфраструктурных объектов и транспортных средств с ИС стоит поговорить подробнее. Сам принцип взаимодействия довольно тривиален. Устройства должны отправлять запросы на сервер ИС и реагировать на решения об изменении из состояний. Например, светофор работает в штатном режиме и переключает фазы. В определенный момент времени системой было принято решение, что светофору необходимо открыть разрешающий сигнал приближающемуся трамваю (чтобы тот проехал светофор без остановки и смог соблюсти график). Проблема заключается в том, что на дорогах и улицах страны огромное количество различных управляющих устройств, часть из которых может не поддерживать такой принцип взаимодействия. Это касается и других устройств (устройства управления наружным освещением, системы АЛС и т.п.).

В качестве решения рассматривается вариант применения промежуточного программно-аппаратного устройства, которое будет агрегировать все виды локальных устройств. Данное устройство должно взаимодействовать с сервером ИС и отправлять локальным устройствам сигналы в таком формате, который они поддерживают. Схема такого взаимодействия приведена на рисунке 3.3.

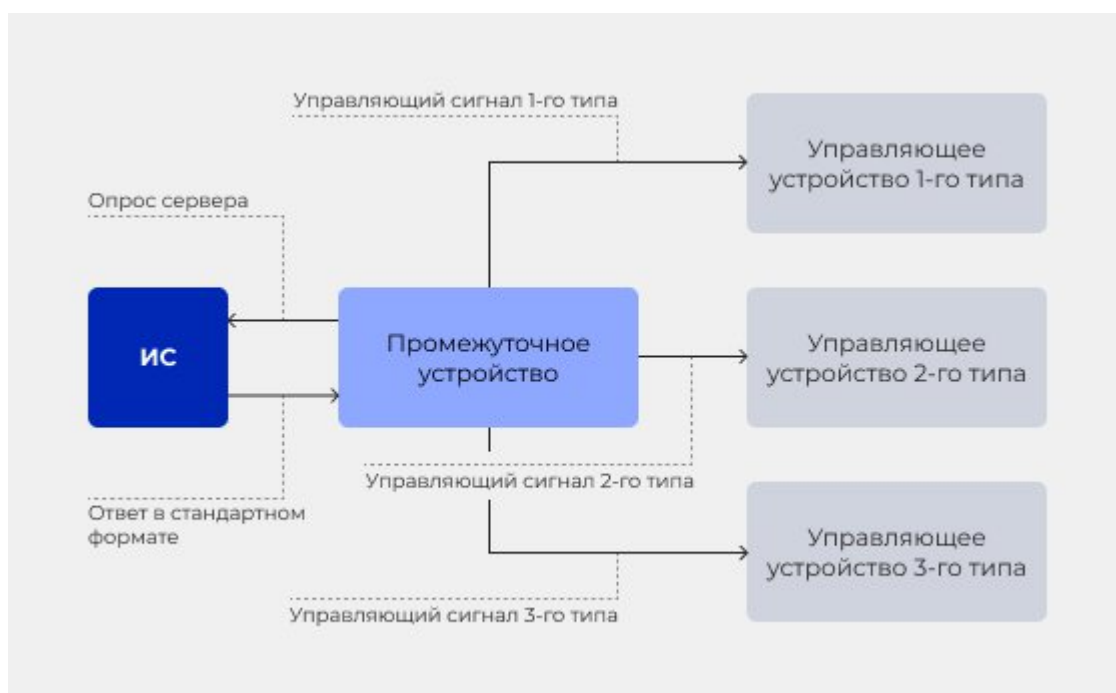


Рисунок 3.3 — Схема взаимодействия управляющих устройств с ИС.

## Выводы по разделу

В данном разделе было выполнено обоснование выбора средств разработки: локального сервера, среды разработки, системы управления базами данных. Было выполнено описание реализации клиентской части информационной системы, базы данных. Были описаны обеспечение информационной безопасности а также работа пользователя с ИС. На основе этих данных можно составить календарный план проекта, а также провести расчет стоимости реализации проекта.

## 4. ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

### 4.1. Организация и планирование работ по теме

В текущей выпускной квалификационной работе (ВКР) принимают участие 3 лица:

1. Руководитель ВКР — преподаватель, который курирует работу студента. В его задачи входят: правильная постановка задачи, контроль за всеми этапами работы над ВКР, корректировка и оценка работы в целом.

2. Консультант по экономической части ВКР (далее консультант) — преподаватель кафедры экономики. Его задача — консультирование по всем вопросам, связанным с экономическим разделом ВКР.

3. Разработчик — студент. В его обязанности входит реализация всех поставленных задач ВКР, а именно создание проектной документации к ВКР, создание самого проекта, а также его тестирование.

Состав участников работы изображен на организационной схеме, на рисунке 4.1.



Рисунок 4.1 — Организационная схема.

#### 4.1.1. Организация работ

Общий срок выполнения ВКР составляет 90 дней. План реализации проекта ВКР представлен в таблице 4.1.

Таблица 4.1 — План реализации проекта ВКР.

№	Название этапа	Исполнитель	Трудоемкость, чел/дни	Продолжительность работ, дни
<b>1</b>	<b>Техническое задание</b>			5
1.1	Разработка и утверждение технического задания.	Руководитель	5	
		Разработчик	5	
<b>2</b>	<b>Технические предложения</b>			7
2.1	Обсуждение технических предложений.	Руководитель	7	
		Консультант	1	
		Разработчик	7	
<b>3</b>	<b>Эскизный проект</b>			16
3.1	Анализ исходных данных и требований.	Разработчик	9	
3.2	Постановка задачи.	Консультант	1	
3.3	Разработка общего описания алгоритма функционирования.	Руководитель	2	
		Разработчик	7	
<b>4</b>	<b>Технический проект</b>			15
4.1	Определение формы представления входных и выходных данных.	Руководитель	2	
		Разработчик	5	
4.2	Разработка структуры программы и логической структуры базы данных.	Руководитель	2	
		Консультант	1	
		Разработчик	10	
<b>5</b>	<b>Реализация</b>			47
5.1	Реализация программных компонентов системы.	Разработчик	47	

#### 4.1.2. График проведения работ

Календарный график исполнения всех выше описанных работ представлен ниже в виде таблицы (Таблица 4.2).

Таблица 4.2 — Представление графика исполнения работ (диаграмма Ганта).

Этапы\Дни	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90
1	0 5																	
2		5 12																
3			12 28															
4						28 43												
5									43 90									

## 4.2. Расчет стоимости проведения работ по теме

С учетом специфики выпускной квалификационной работы в сфере информационных технологий, консультантом по экономическому разделу ВКР были выделены следующие статьи расчета себестоимости проекта:

1. Материалы, покупные изделия и полуфабрикаты.
2. Специальное оборудование.
3. Основная заработная плата.
4. Дополнительная заработная плата.
5. Страховые отчисления.
6. Командировочные расходы.
7. Накладные расходы.
8. Прочие расходы.

### 4.2.1. Материалы, покупные изделия и полуфабрикаты

В данной статье расходов описывается стоимость материалов, покупных изделий и полуфабрикатов, а также прочих материальных ценностей. Первым делом рассчитываются суммарная стоимость всех материальных ценностей, затем транспортные расходы в размере 15% процентов от посчитанных ранее затрат на материальные ценности. После высчитывается общая сумма расходов. Стоимость материальных ценностей, которые необходимы для реализации выпускной квалификационной работы приведена в таблице ниже (Таблица 4.3).

Таблица 4.3 — Себестоимость материальных ценностей.

№	Наименование материальной ценности	Количество	Цена единицы, руб	Суммарная стоимость, руб
1	USB FLASH-накопитель объёмом 16 гигабайт	1 шт.	450	450
2	Лазерный принтер	1 шт.	11 400	11 400
3	Бумага формата А4	1 пачка	290	290
4	Набор из черных и синих ручек	1 шт.	600	600
5	Набор карандашей	1 шт.	300	300
Итоговая стоимость материалов:				13 040

Продолжение Таблицы 4.3.

<b>Транспортные расходы:</b>	1956
<b>Итоговая стоимость:</b>	14 996

#### 4.2.2. Специальное оборудование

В текущей ВКР не предусматривается использование специального оборудования, поэтому затрат на него нет.

#### 4.2.3. Основная заработная плата

Данная статья расходов включает в себя оплату труда всех участников ВКР на каждом этапе выполнения работы с учетом трудоемкости, месячного оклада и оплаты труда за день. В среднем на один рабочий месяц выпадает 8 выходных дней (без учета праздничных дней), поэтому дневная оплата рассчитывается как месячная деленная на 22. Основная заработная плата (ОЗП) участников ВКР приведена в таблице ниже (Таблица 4.4).

Таблица 4.4 — Основная заработная плата.

№ этапа	Исполнитель	Месячный оклад, руб	Трудо-емкость, чел/дни	Оплата за день, руб.	Оплата за этап, руб.
1	Руководитель	80 000	4	3 636	14 544
	Разработчик	2615	7	119	833
2	Руководитель	80 000	3	3 636	10 908
	Разработчик	2615	7	119	833
3	Руководитель	80 000	3	3 636	10 908
	Разработчик	2615	3	119	357
4	Руководитель	80 000	3	3 636	10 908
	Разработчик	2615	10	119	1 190
5	Руководитель	80 000	3	3 636	10 908
	Разработчик	2615	3	119	357
<b>Итого общей заработной платы:</b>					61 746

#### 4.2.4. Дополнительная заработная плата

В текущей ВКР не предусматривается дополнительная заработная плата (ДЗП) для ее участников.

#### 4.2.5. Страховые отчисления

Отчисления на социальные нужды [24] (страховые выплаты, далее СВ) отчисляются 30% из фонда оплаты труда, который состоит из основной



заработной платы. Страховые выплаты рассчитываются по формуле ниже (4.1):

$$CB = (ОЗП + ДЗП) * 30\% = 61\,746 * 0,3 = 20\,323 \text{ руб. } 80 \text{ коп.} \quad (4.1)$$

#### 4.2.6. Командировочные расходы

В текущей ВКР не предусматриваются командировочные и прочие подобные расходы, так как командировок в процессе выполнения не предусматривается.

#### 4.2.7. Накладные расходы

В текущей ВКР не предусматриваются накладные расходы, так как разрабатываемая система не является частью инфраструктуры вуза.

#### 4.2.8. Прочие расходы

В данной работе прочие расходы (ПР) приведены как расходы на потраченное при работе с ВКР машинное время компьютера и интернет соединения в течении 90 рабочих дней по 8 часов в день с учётом того, что тариф стоимости электроэнергии 4,16 рублей за кВт\*ч, средний компьютер за час потребляет 0,25 кВт\*ч, а стоимость интернета 600 руб./месяц. Прочие расходы рассчитываются по формуле ниже (4.2):

$$ПР = 90 * 8 * 4,16 * 0,25 + 600 = 1348 \text{ руб. } 80 \text{ коп.} \quad (4.2)$$

#### 4.2.9. Подсчет себестоимости проекта

Полная себестоимость проекта рассчитывается как общая стоимость по всем 9 статьям расходов, которые были описаны выше. Полная себестоимость ВКР и стоимость каждого пункта расходов приведены в таблице ниже (Таблица 4.5).

Таблица 4.5 — Полная себестоимость ВКР и величины расходов.

№ статьи	Номенклатура	Затраты, руб.
1	Материалы, покупные изделия и полуфабрикаты.	14 996
2	Специальное оборудование.	—
3	Основная заработная плата.	61 746
4	Дополнительная заработная плата.	—
5	Страховые отчисления.	20 323,8
6	Командировочные расходы.	—

*Продолжение Таблицы 4.5.*

7	Накладные расходы.	—
8	Прочие расходы.	1348,8
<b>Полная себестоимость проекта:</b>		<b>98 414,6</b>

Так как информационная система управления общественным транспортом разрабатывается для использования в учреждениях относящихся к органам исполнительной власти, то договорная цена, а также прибыль не рассматриваются и не рассчитываются.

### **Выводы по разделу**

В данном разделе было выполнено описание организации и планирования работ над текущей ВКР. Была создана организационная схема включающая всех участников. Была составлена диаграмма Ганта, включающая все этапы выполнения проекта. Также был выполнен расчет себестоимости выполнения проекта по девяти статьям расходов.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были решены следующие задачи:

- Проведен анализ предметной области. Определены объект и предмет исследования;
- Проведен обзор и анализ существующих решений, установлены частичные аналоги, выявлены их достоинства и недостатки;
- Поставлена задача на разработку информационной системы;
- Составлено техническое задание по государственному стандарту ГОСТ 34.602-2020;
- Описано информационное обеспечение информационной системы: информационная модель и ее описание, нормативно-справочная и входная информация, результатная информация;
- Определены основные требования к информационной системе: требования к функциональным характеристикам, требования к надежности, условия эксплуатации, требования к составу и параметрам аппаратных средств, требования к информационной совместимости;
- Обоснован выбор средств разработки: выбор локального сервера, выбор сред разработки, выбор системы управления базами данных;
- Описана реализация клиентской части информационной системы;
- Описана реализация базы данных информационной системы;
- Рассмотрено обеспечение информационной безопасности при эксплуатации информационной системы;
- Описана работа пользователей с информационной системой;
- Были выполнены организация и планирования работ по теме ВКР;
- Был выполнен расчет стоимости проведения работ по теме ВКР.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сети города: Люди. Технологии. Власти / Под редакцией Е. Лапиной-Карастюк, О. Запорожец, А. Возьянова; пер с англ. К. Гусаровой, А. Возняка, О. Запорожец, М.-В. Моррис, Е. Лапиной-Карастюк. — М.: Новое литературное образование, 2021 — 576 с.: ил. (Серия STUDIA URBANISIA).
2. Штырхунова Наталья Александровна, Голубкина Ксения Вячеславовна, Абрамян Сюзанна Кароевна ВОЗМОЖНОСТИ ЦИФРОВИЗАЦИИ НА ТРАНСПОРТЕ (НА ПРИМЕРЕ ГОРОДСКОГО ПАССАЖИРСКОГО ТРАНСПОРТА) // Гуманитарные, социально-экономические и общественные науки. 2021. №4-2.
3. Ефанов Д.В., Жуков С.А., Симаков Е.В., Пентегов Д.А., Феклистова М.В. ТЕХНОЛОГИИ ПЕРЕКЛЮЧЕНИЯ СРЕДСТВ АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ ДВИЖЕНИЯ ПОЕЗДОВ В МЕТРОПОЛИТЕНАХ ПРИ ВНЕДРЕНИИ КОМПЛЕКСНЫХ СИСТЕМ УПРАВЛЕНИЯ // Автоматика на транспорте. 2022. №1.
4. Семенов Н.А., Ачитаев Н.Н., Гаранин А.Е. РЕАЛИЗАЦИЯ СИСТЕМЫ ДИАГНОСТИКИ И МОНИТОРИНГА СОСТОЯНИЙ УСТРОЙСТВ СЦБ В УЧЕБНОМ ПРОЦЕССЕ КРИЖТ ИРГУПС // Вестник науки. 2022. №12 (57).
5. Кокурин И.М. РЕШЕНИЯ ЗАДАЧ ИНТЕРВАЛЬНОГО РАЗГРАНИЧЕНИЯ ПОЕЗДОВ // Автоматика на транспорте. 2021. №3.
6. Антонов А. А., Красавин А. В., Новиков А. С., Сиразетдинова А. Д. Оценка факторов, влияющих на безопасность ведения поезда // Наука и современность. 2010. №4-1.
7. ГОСТ 19001-77 Единая система программной документации. Общие положения. — Введ. 01.01.1980 — 6 с. (дата обращения: 01.04.2023).
8. Плахина Екатерина Александровна К ВОПРОСУ О ЗАЩИТЕ ДАННЫХ НА НОСИТЕЛЯХ ИНФОРМАЦИИ // Известия ТулГУ. Технические науки. 2021.

9. Болотов Александр Сергеевич, Болотова Татьяна Павловна  
Файловые системы Windows Server 2012 // Вестник ГУУ. 2013. №1.
10. Гагарин В.Ю., Вагнер А.В., Тропченко А.А. ОСОБЕННОСТИ  
МИГРАЦИИ ТЕХНИЧЕСКИХ ПРОЕКТОВ С МИКРОСЕРВИСНОЙ  
АРХИТЕКТУРОЙ И ИСПОЛЬЗОВАНИЕМ GRPC В КАЧЕСТВЕ  
ПРОТОКОЛА МЕЖСЕРВИСНОГО ВЗАИМОДЕЙСТВИЯ С  
ПЛАТФОРМЫ .NET CORE 3 НА ПЛАТФОРМУ .NET 6 // МНИЖ. 2022. №4-  
1 (118).
11. Быковская Елизавета Феликсовна, Бугакова Татьяна Юрьевна  
РАЗРАБОТКА БАЗЫ ДАННЫХ УЧЕТА НАУЧНЫХ ДОСТИЖЕНИЙ  
ОБУЧАЮЩИХСЯ СРЕДСТВАМИ MICROSOFT SQL SERVER // Интерэкспо  
Гео-Сибирь. 2021. №2.
12. Поддубная Л.В. Реализация алгоритмов на графах средствами  
реляционной алгебры // Символ науки. 2016. №6-1.
13. Джеффри Рихтер CLR via C#. Программирование на платформе  
Microsoft .NET Framework 4.5 на языке C#. 4-е изд. — СПб.: Питер, 2013 —  
896 с.: (Серия Мастер-класс).
14. Гузарев Антон Сергеевич МЕТОД ОЦЕНКИ  
ПРОИЗВОДИТЕЛЬНОСТИ ВСТРАИВАЕМЫХ МОДУЛЕЙ НА ОСНОВЕ  
X86-64 ПРОЦЕССОРОВ // Техника средств связи. 2018. №4 (144).
15. Кобылкин Д.С., Юсупова О.В. ПЕРСПЕКТИВА ПРИМЕНЕНИЯ  
СРЕДСТВ ENTITY FRAMEWORK ПРИ РАЗРАБОТКЕ СИСТЕМ  
РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ НА ЯЗЫКЕ C# // Символ  
науки. 2021. №2.
16. ГОСТ 34.602-2020 Информационные технологии. Комплекс  
стандартов на автоматизированные системы. Техническое задание на создание  
автоматизированной системы. — Введ. 01.01.2020 — 12 с. (дата обращения:  
01.04.2023).

17. ГОСТ 24.701-86 Единая система стандартов автоматизированных систем управления. Надежность автоматизированных систем управления. Основные положения. — Введ. 01.07.1987 — 12 с. (дата обращения: 01.04.2023).

18. ГОСТ 11550-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. — Введ. 01.01.1971 — 72 с. (дата обращения: 01.04.2023).

19. ГОСТ 19.101-77 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. — Введ. 01.01.1992 — 24 с. (дата обращения: 01.04.2023).

20. Татаров Максим Олегович, Пугачёв Алексей Юрьевич, Пушкин Константин Александрович ПРИМЕНЕНИЕ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ УЧЕБНОЙ И НАУЧНОЙ ДЕЯТЕЛЬНОСТИ В ВОЕННОЙ ОБРАЗОВАТЕЛЬНОЙ ОРГАНИЗАЦИИ // Военная мысль. 2021. №1.

21. Басов А.С. СРАВНЕНИЕ СОВРЕМЕННЫХ СУБД // Вестник науки. 2020. №7 (28).

22. Макеева О.В., Сартаков М.В., Чернов Е.А. МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ С ПОМОЩЬЮ UML // Инновации и инвестиции. 2021. №9.

23. Добрышин М.М. НАПРАВЛЕНИЯ РАЗВИТИЯ И ЗАДАЧИ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ АДАПТИВНЫХ СИСТЕМ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ // Экономика и качество систем связи. 2022. №3 (25).

24. Сурков Леонид Петрович, Сычугова Ирина Моисеевна, Булохова Татьяна Александровна Систематизация процесса планирования и разработка алгоритма расчета отчислений на социальные нужды в подразделениях холдинга РЖД // Современные технологии. Системный анализ. Моделирование. 2014. №1 (41).

## **ПРИЛОЖЕНИЯ**

Приложение А — Логическая модель базы данных.

Приложение Б — Листинг программных компонентов.

Приложение В — Презентация к ВКР.



## Приложение А — Логическая модель базы данных

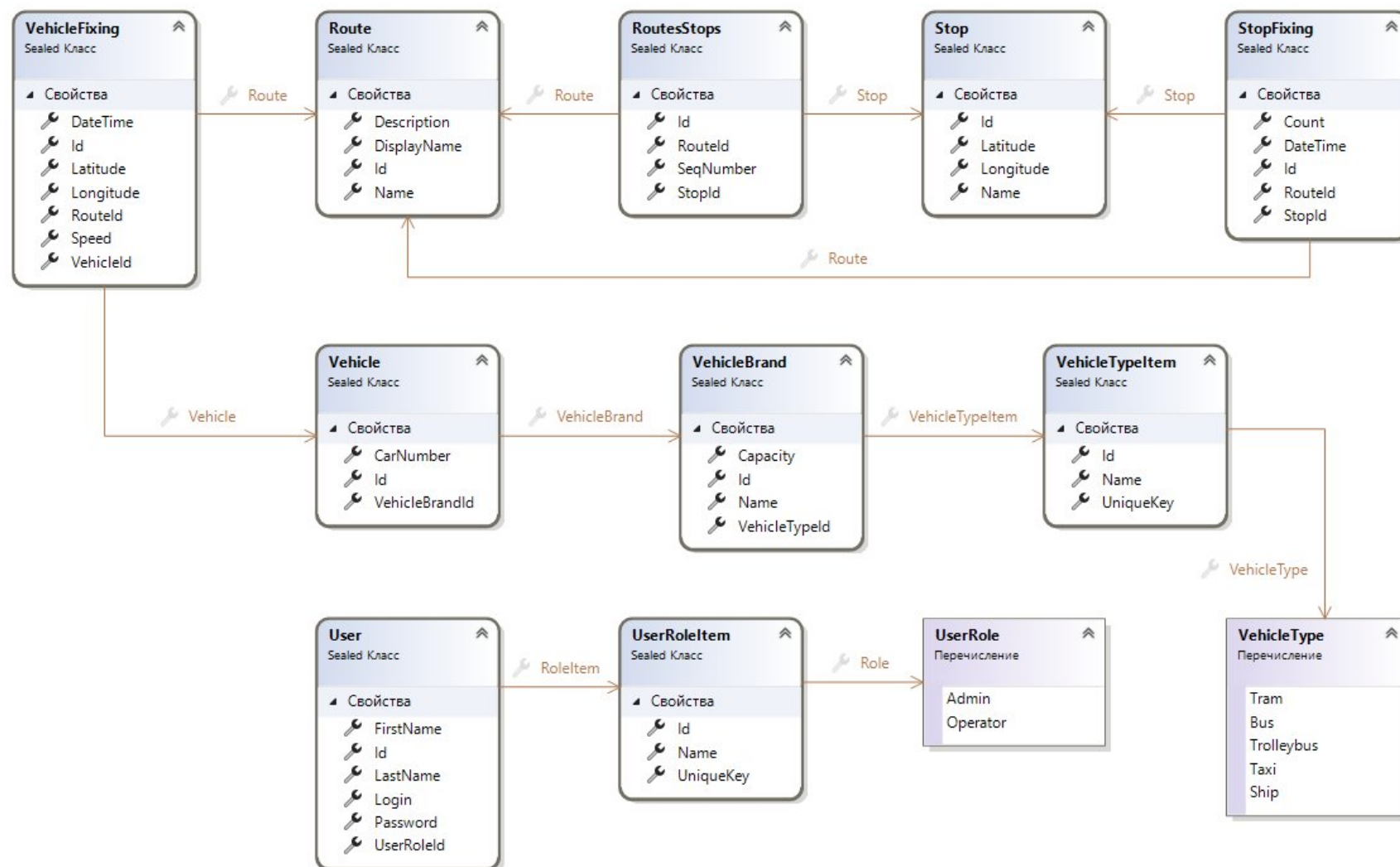


Рисунок А.1 — Логическая модель базы данных.

## Приложение Б — Листинг программных компонентов

*Листинг Б.1 — Интерфейс контекста базы данных и его реализация.*

```
using Microsoft.EntityFrameworkCore;
using TransportManagement.Core.DataAccess.DomainModels;

namespace TransportManagement.Core.DataAccess.Context
{
    public interface IDatabaseContext : IDisposableable
    {
        public DbSet<Route>? Routes { get; set; }
        public DbSet<RoutesStops>? RoutesStops { get; set; }
        public DbSet<Stop>? Stops { get; set; }
        public DbSet<StopFixing>? StopFixings { get; set; }
        public DbSet<User>? Users { get; set; }
        public DbSet<UserRoleItem>? UserRoleItems { get; set; }
        public DbSet<Vehicle>? Vehicles { get; set; }
        public DbSet<VehicleBrand>? VehicleBrands { get; set; }
        public DbSet<VehicleFixing>? VehicleFixings { get; set; }
        public DbSet<VehicleTypeItem>? VehicleTypeItems { get; set; }

        public int SaveChanges();
        public Task<int> SaveChangesAsync(CancellationToken cancellationToken = default);
        public DbSet<TEntity> Set<TEntity>() where TEntity : class;
    }

    public class DatabaseContext : DbContext, IDatabaseContext
    {
        public DatabaseContext(DbContextOptions options) : base(options)
        {
            return;
        }

        public DbSet<Route>? Routes { get; set; }
        public DbSet<RoutesStops>? RoutesStops { get; set; }
        public DbSet<Stop>? Stops { get; set; }
        public DbSet<StopFixing>? StopFixings { get; set; }
        public DbSet<User>? Users { get; set; }
        public DbSet<UserRoleItem>? UserRoleItems { get; set; }
        public DbSet<Vehicle>? Vehicles { get; set; }
        public DbSet<VehicleBrand>? VehicleBrands { get; set; }
        public DbSet<VehicleFixing>? VehicleFixings { get; set; }
        public DbSet<VehicleTypeItem>? VehicleTypeItems { get; set; }
    }
}
```

*Листинг Б.2 — Интерфейс транспортной модели.*

```
using TransportManagement.Core.DataAccess.DeploymentModels.Deployment.Traces;
using TransportManagement.Core.DataAccess.DomainModels;

namespace TransportManagement.Server.Services
{
    public interface ITransportProjectService
    {
        public string ProjectFilePath { get; }
    }
}
```

*Продолжение Листинга Б.2.*

```
public bool IsProjectExists { get; }

public List<Core.DataAccess.DomainModels.Route> GetRoutes();
public Trace? GetRouteTrace(int routeId);
public List<Stop>? GetRouteStops(int routeId);
public List<Vehicle>? GetVehicles(Predicate<Vehicle> predicate);
public List<StopFixing>? GetPassengersTraffic(int routeId);
public List<VehicleFixing>? GetVehicleFixings(int routeId);
public bool SaveStopFixing(StopFixing stopFixing);
public bool SaveVehicleFixing(VehicleFixing vehicleFixing);
    }
}
```

*Листинг Б.3 — Внедрение зависимостей.*

```
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.EntityFrameworkCore;
using Microsoft.IdentityModel.Tokens;
using TransportManagement.Core.DataAccess.ClientServerModels.Authentication;
using TransportManagement.Core.DataAccess.Context;
using TransportManagement.Server.Services;

namespace TransportManagement.Server.Extensions
{
    public static class DependencyInjection
    {
        public static void AddCustomServices(this IServiceCollection services,
            ConfigurationManager configManager)
        {
            AddDbContext(services, configManager);
            AddAuthentication(services);

            AddScoped(services);
        }

        private static void AddDbContext(IServiceCollection services,
            ConfigurationManager configManager)
        {
            _ = services.AddDbContext<IDatabaseContext, DatabaseContext>
                ((DbContextOptionsBuilder optionsBuilder) =>
                {
                    string? connectionString = configManager
                        .GetConnectionString("DefaultConnection");
                    optionsBuilder.UseSqlServer(connectionString);
                });
        }

        private static void AddAuthentication(IServiceCollection services)
        {
            _ = services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
                .AddJwtBearer(options =>
                {
                    options.TokenValidationParameters = new TokenValidationParameters
                    {
                        ValidateIssuer = true,
                        ValidIssuer = AuthOptions.Issuer,
                        ValidateAudience = true,
                        ValidAudience = AuthOptions.Audience,
                        ValidateIssuerSigningKey = true,
                    }
                });
        }
    }
}
```

*Продолжение Листинга Б.3.*

```
        IssuerSigningKey = AuthOptions.IssuerSigningKey
    };
    });
}

private static void AddScoped(IServiceCollection services)
{
    services.AddScoped<ITransportProjectService,
    TransportProjectService>();
}
}
```

*Листинг Б.4 — Отправка запросов на сервер.*

```
#region Получение данных.
private async Task<HttpResponseMessage?> SendHttpGet(Uri uri)
{
    try
    {
        return await httpClient.GetAsync(uri);
    }
    catch (HttpRequestException)
    {
        MessageBox.Show("Не удалось подключиться к серверу", "Ошибка подключения", MessageBoxButton.OK, MessageBoxImage.Error);
        return default;
    }
}

private async Task<T?> ReadResponseContent<T>(HttpResponseMessage? responseMessage)
{
    try
    {
        if (responseMessage is not null && responseMessage.IsSuccessStatusCode)
            return await responseMessage.Content.ReadFromJsonAsync<T?>();
        else
            return default;
    }
    catch (AggregateException)
    {
        MessageBox.Show("Сервер не обработал запрос", "Ошибка подключения", MessageBoxButton.OK, MessageBoxImage.Error);
        return default;
    }
}
}
#endregion
```

*Листинг Б.5 — Обработка запросов на сервере.*

```
using Microsoft.AspNetCore.Mvc;
using TransportManagement.Core.DataAccess.ClientServerModels;
using TransportManagement.Server.Services;

namespace TransportManagement.Server.Controllers.TransportModel
{
    [ApiController]
    [RequireHttps]
    [Route("api/project")]
    public class ProjectDataController : Controller
    {

```

```
private readonly ITransportProjectService _transportProjectService;

public ProjectDataController(ITransportProjectService service)
{
    _transportProjectService = service;
}

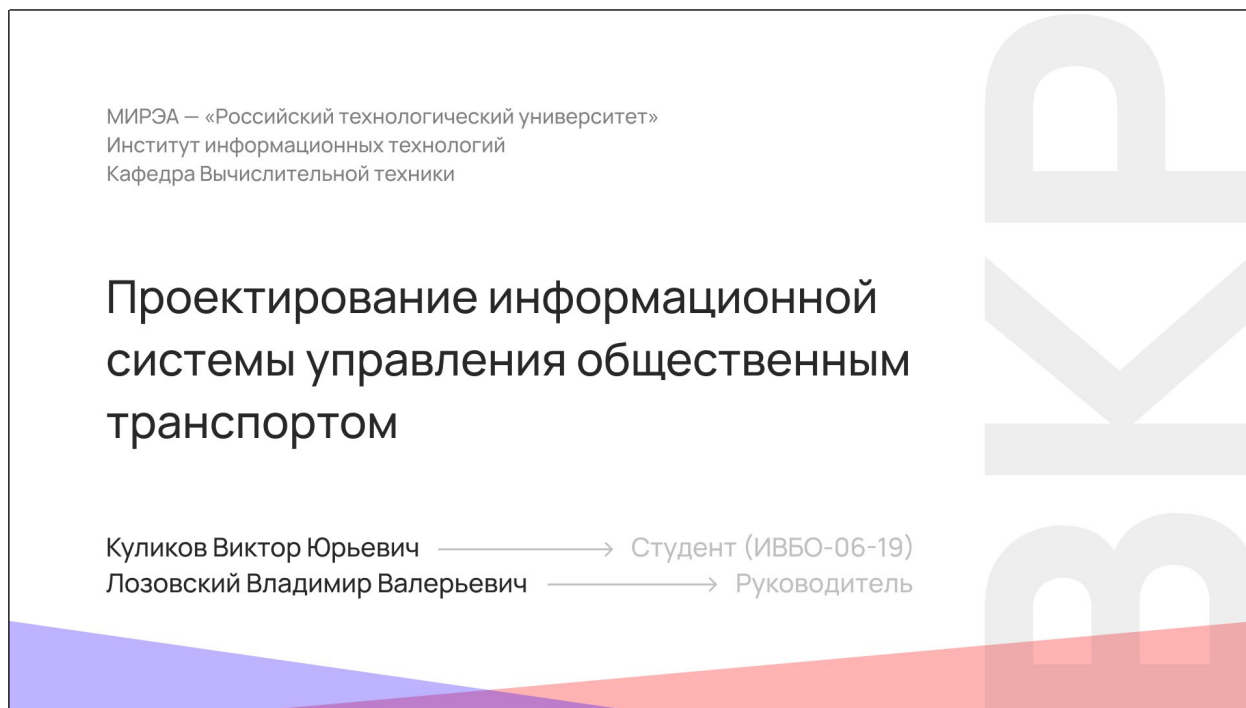
[Authorize]
[HttpGet("exists")]
public IActionResult GetState()
{
    if (_transportProjectService.IsProjectExists)
        return new StatusCodeResult(204);
    else
        return NotFound(ErrorModel.MissingProjectFile);
}

[Authorize]
[HttpGet("routes")]
public IActionResult GetRoutes()
{
    if (_transportProjectService.IsProjectExists)
    {
        var routes = _transportProjectService.GetRoutes();
        return Ok(routes);
    }
    else
    {
        return NotFound(ErrorModel.MissingProjectFile);
    }
}

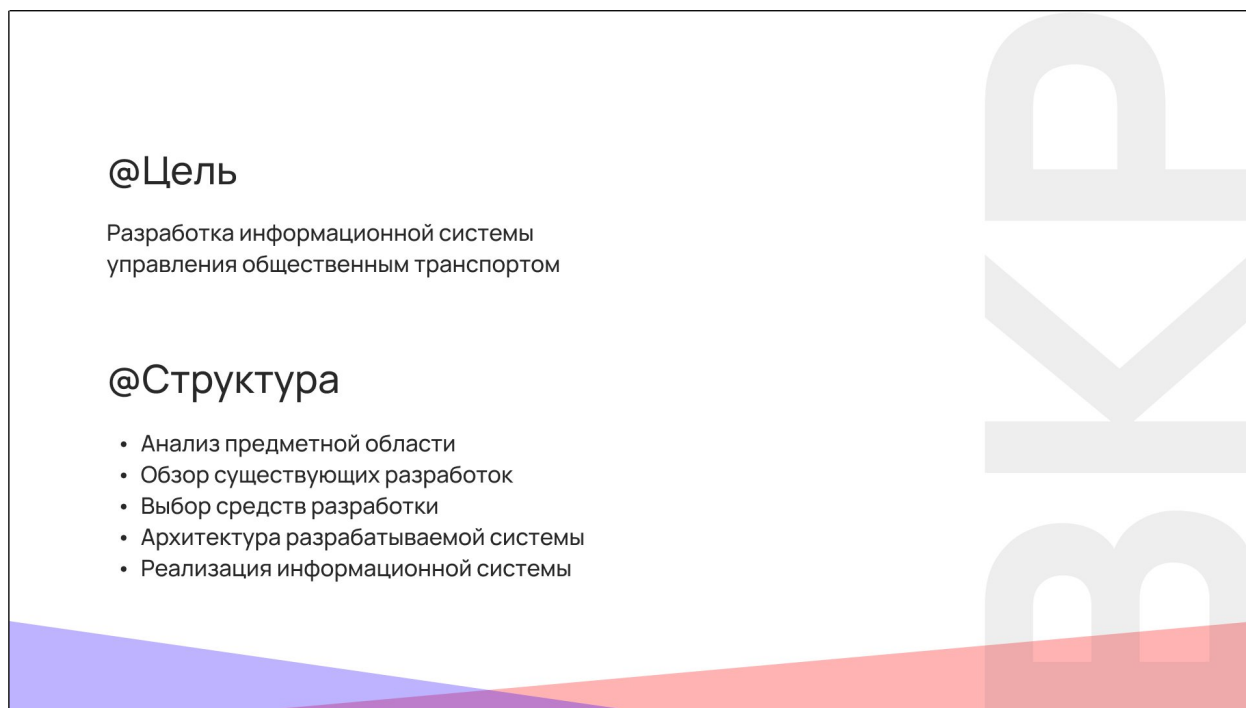
[Authorize]
[HttpGet("routes/{routeId}/trace")]
public IActionResult GetRouteTrace(int routeId)
{
    if (_transportProjectService.IsProjectExists)
    {
        var trace = _transportProjectService.GetRouteTrace(routeId);
        return trace is not null ? Ok(trace) :
            NotFound(ErrorModel.MissingNodeInProjectFile);
    }
    else
    {
        return NotFound(ErrorModel.MissingProjectFile);
    }
}

...
}
}
```

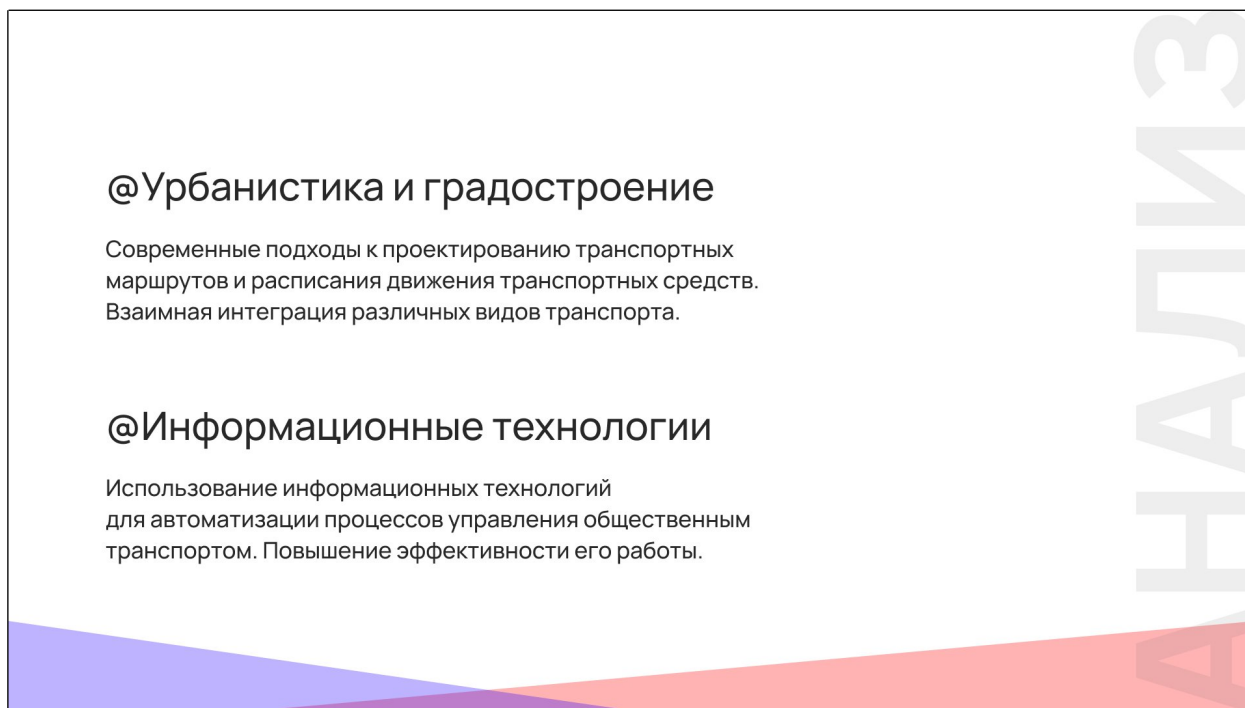
## Приложение В — Презентация к ВКР



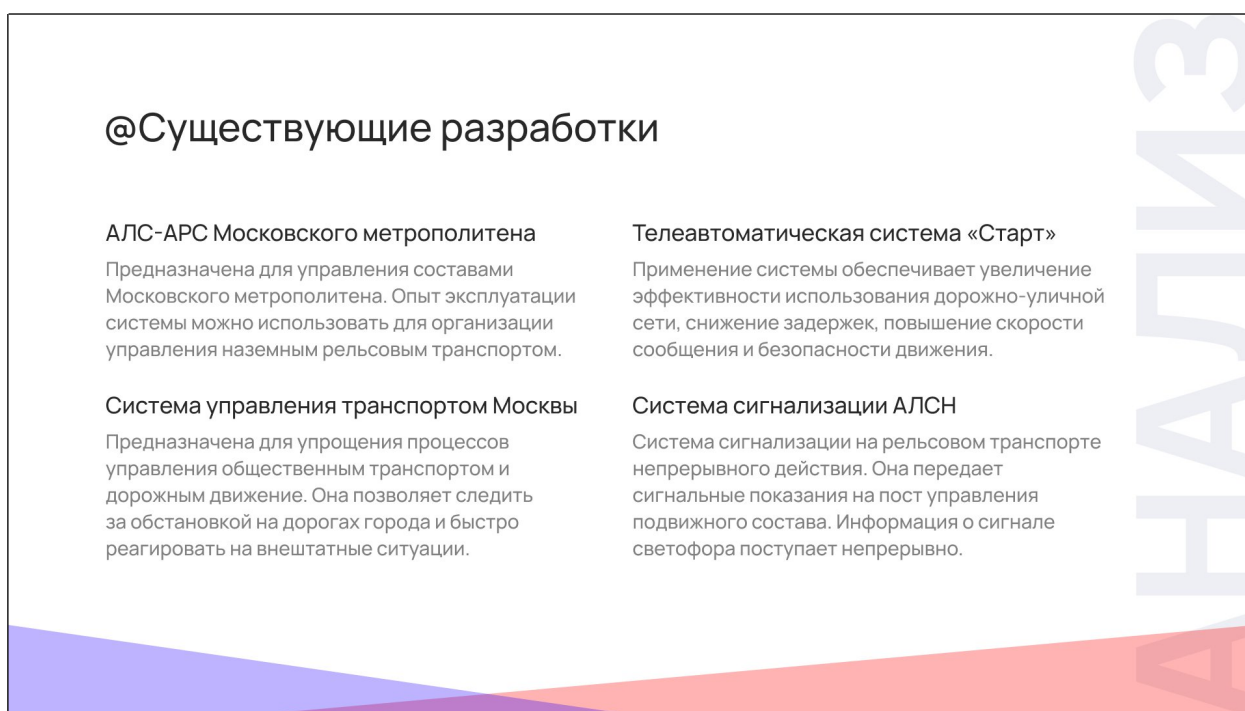
**Рисунок В.1 — Титульный слайд.**



**Рисунок В.2 — Цели и задачи.**



**Рисунок В.3 — Анализ предметной области.**



**Рисунок В.4 — Обзор существующих разработки.**

## @Выбор средств разработки

### Среда разработки

Microsoft Visual Studio Community 2022 Preview

### Локальный сервер

IIS — Internet Information Services

### Сервер базы данных

Microsoft SQL Server

### Система управления базами данных

Microsoft SQL Server Management Studio

### Платформа и язык программирования

.NET Core 7.0 / C# (Си-Шарп)

### Платформа серверной части

ASP.NET Core — WebApi

### Платформа клиентской части

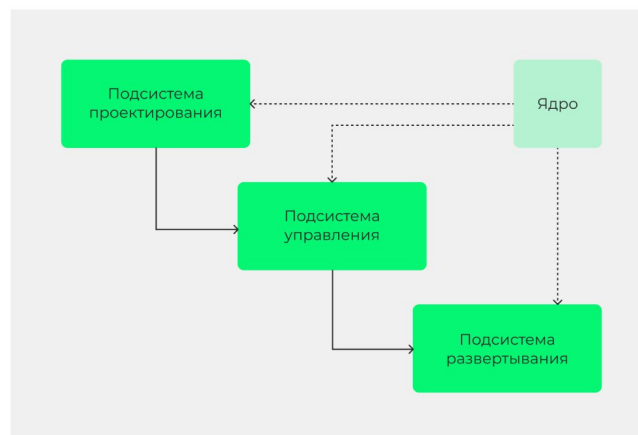
WPF (Windows Presentation Foudation)

### Фреймворк взаимодействия с БД

Entity Framework Core 7.0

**Рисунок В.5 — Выбор средств разработки.**

## @Архитектура системы



**Рисунок В.6 — Архитектура системы 1.**



## @Архитектура системы

### Подсистема проектирования

Предназначена для разработки транспортной модели, включающей объекты транспортной инфраструктуры, транспортные средства, а также маршруты и их трассировки.

### Подсистема управления

Предназначенная для управления разработанной цифровой транспортной моделью, анализа собираемой статистикой, а также принятия решений в случае возникновения внештатных ситуаций.

### Подсистема развертывания

Предназначена для развертывания цифровой транспортной модели. Данная подсистема является сервером. В ней организуется сбор статистики и автоматическое принятие решений.

\* Цифровая транспортная модель является цифровым представлением всех транспортных объектов. Эти представления должны отражать их текущее (объектов) состояние. Через цифровые представления транспортных объектов будет производиться управление ими.

**Рисунок В.7 — Архитектура системы 2.**

## @Реализация системы

### Создание базы данных

Для реализации базы данных устанавливаются следующие пакеты:

- Microsoft.EntityFrameworkCore
- Microsoft.EntityFrameworkCore.SqlServer
- Microsoft.EntityFrameworkCore.Tools

Далее описываются модели (классы), а также контекст базы данных приложения. После этого создается миграция и обновляется база данных. Она присоединяется к серверу базы данных.

### Цифровая транспортная модель

Доступ цифровой транспортной модели осуществляется через API-контроллеры в подсистеме развертывания. Эти контроллеры обрабатывают запросы поступающие от клиента.

### Администрирование модели

Клиентское ПО получает доступ к данным цифровой транспортной модели, а также ее статистике при помощи отправки HTTP-запросов. Для отправки запросов на сервер используется клиент HttpClient.

**Рисунок В.8 — Реализация системы 1.**

## @Реализация системы



Рисунок В.9 — Реализация системы 2.

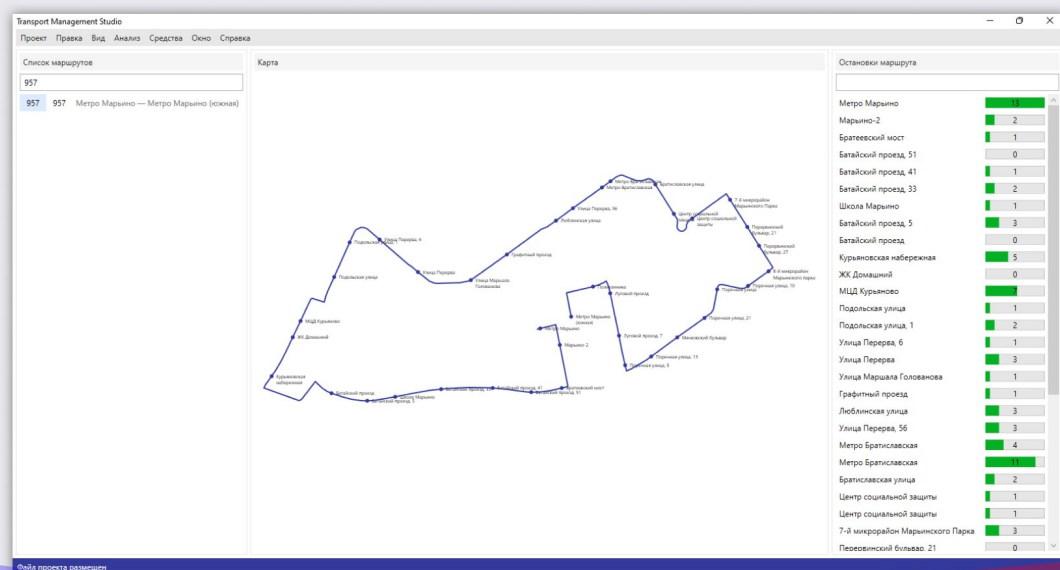


Рисунок В.10 — Снимок экрана.

## @Спасибо за внимание

—> Вы можете задать вопросы по презентации,  
а также по материалам не вошедшим в нее.

**Рисунок В.11 — Спасибо за внимание.**