Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет название

Кафедра название

Дисциплина «Название дисциплины»

|  |
| --- |
| «К ЗАЩИТЕ ДОПУСТИТЬ» |
| Руководитель курсового проекта  канд.техн.наук, доцент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Б. Иванова |
| \_\_\_.\_\_\_\_.2025 |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовому проекту

на тему:

«Проектирование и разработка программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок»

БГУИР КП 1-40 05 01-10 022 ПЗ

|  |
| --- |
| Выполнил студент группы 0123456  ДЕНИС Матвей Сергеевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |
| Курсовой проект представлен на проверку 21.12.2024  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |

Минск 2024

РЕФЕРАТ

БГУИР КП 1-40 05 01-10 022 ПЗ

**Денис, М.С.** Проектирование и разработка программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок : пояснительная записка дипломной работы / М.С. Денис. – Минск : БГУИР, 2024. – с.

Пояснительная записка с., 48 рис., 22 табл., 23 источника, 2 приложения

ВЕБ-ПРИЛОЖЕНИЕ, АВТОМАТИЗАЦИЯ, МЕЖДУГОРОДНИЕ ПАССАЖИРСКИЕ ПЕРЕВОЗКИ, АРХИТЕКТУРА ПРОГРАММНОГО СРЕДТСВА, ПРОЕКТИРОВАНИЕ, ТЕСТИРОВАНИЕ ФУНКЦИОНАЛЬНОСТИ

*Цель проектирования*: разработка программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок.

*Методология проведения работы*: для решения поставленных задач использовались принципы системного подхода, моделирования предметной области с использованием графической нотации и стандартов, теории баз данных, принципы и подходы к проектированию оптимальной архитектуры программного обеспечения, современные практики написания эффективного программного кода и методы тестирования функциональности программного обеспечения.

*Результаты работы*: выполнен анализ предметной области и сравнительный анализ программных средств управления очередью, разработана функциональная модель предметной области в нотации *BPMN* в состояниях *AS IS* и *TO BE*, разработана спецификация функциональных требований, разработана архитектура программного средства, разработана база данных, разработаны диаграммы классов, описывающие статические и динамические аспекты поведения программного средства, разработаны алгоритмы, реализующие бизнес-логику программного средства, реализован и протестирован проект программного средства, составлено руководство по развертыванию и применению программного средства.

*Область применения результатов*: результаты работы могут применяться при организации управления междугородними пассажирскими перевозками.

**ЗАДАНИЕ**

**по курсовому проекту**

Содержание

[Введение 5](#_Toc188025756)

[1 Анализ литературных исследований и программных решений 7](#_Toc188025757)

[1.1 Описание и анализ предметной области 7](#_Toc188025758)

[1.2 Обзор функциональности аналогов программного средства 9](#_Toc188025759)

[2 Моделирование предметной области и разработка требований к программному средству 15](#_Toc188025760)

[2.1 Анализ и формализация бизнес-процессов предметной области 15](#_Toc188025761)

[2.2 Анализ требований к разрабатываемому программному средству и разработка их спецификаций 17](#_Toc188025762)

[2.3 Образ предлагаемого решения 34](#_Toc188025763)

[3 Проектирование и разработка программного средства 37](#_Toc188025764)

[3.1 Архитектурные решения 37](#_Toc188025765)

[3.2 Обоснование выбора компонентов и технологий для реализации программного средства 40](#_Toc188025766)

[3.3 Проектирование и разработка пользовательского интерфейса 42](#_Toc188025767)

[3.4 Разработка модели данных 46](#_Toc188025768)

[3.5 Описание статических и динамических аспектов поведения программных объектов 49](#_Toc188025769)

[3.6 Разработка и описание алгоритмов, реализующих бизнес-логику разрабатываемого программного средства 54](#_Toc188025770)

[3.7 Разработка программного средства 57](#_Toc188025771)

[3.8 Механизмы обеспечения информационной безопасности 62](#_Toc188025772)

[4 Тестирование и проверка работоспособности программного средства 63](#_Toc188025773)

[5 Руководство по установке (развертыванию) и использованию программного средства 66](#_Toc188025774)

[5.1 Руководство по установке (развертыванию) программного средства 66](#_Toc188025775)

[5.2 Руководство пользователя 68](#_Toc188025776)

[Заключение 73](#_Toc188025777)

Введение

Современный пассажирский транспорт является важнейшей системой, обеспечивающей экономическое развитие городов и социальное благополучие населения. Несмотря на стремительное увеличение количества личного автотранспорта, по оценкам многих специалистов, общественный транспорт обеспечивает около 80% поездок граждан. Активное внедрение цифровых технологий в транспортной отрасли повышает интенсивность конкуренции на рынке и задает новые стандарты как клиентского опыта, так и экономической эффективности автотранспортного предприятия. В связи с этим совершенствование инструментов и методов управления становится не просто целесообразным вложением в будущее бизнеса, но и необходимым условием его конкурентоспособности.

Актуальность дипломной работы заключается в необходимости разработки современных средств и методов, позволяющих автоматизировать деятельность диспетчера междугородных пассажирских перевозок автотранспортного предприятия.

Практическая значимость курсовой работы заключается в разработке и внедрении в деятельность автотранспортного предприятия средств автоматизации, позволяющих повысить эффективность деятельности диспетчера междугородных пассажирских перевозок.

Объект исследования – деятельность диспетчера междугородных пассажирских перевозок.

Предмет исследования – программное средство автоматизации рабочего места диспетчера междугородних пассажирских перевозок.

Цель курсового проекта – разработка программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок.

Для достижения поставленной цели решаются следующие задачи:

* проанализировать предметную область и существующие аналоги;
* выполнить моделирование предметной области и разработать спецификацию к программному средству;
* осуществить проектирование и реализацию программного средства;
* протестировать программное средство;
* разработать руководство по установке и использованию программного средства.

Курсовой проект состоит из введения, пяти глав, заключения, списка использованных источников, приложений и ведомости.

В первом разделе работы проводится анализ потребностей автотранспортного предприятия в средствах автоматизации деятельности диспетчера междугородних пассажирских перевозок, а также сравнительный анализ аналогов – существующих решений в области диспетчеризации пассажирских автотранспортных перевозок.

Во втором разделе работы выполняется моделирование предметной области, результатом чего являются функциональная модель предметной области в нотации *BPMN* в состояниях *AS IS* и *TO BE*, диаграмма вариантов использования *UML*, функциональная спецификация программного средства и образ предполагаемого решения.

Третий раздел посвящен проектированию и разработке программного средства, в ходе которого определяются архитектура программного средства и средства и методы разработки, разрабатываются база данных, описываются статически и динамические аспекты поведения программных объектов в виде диаграмм *UML*, разрабатываются алгоритмы и код программного средства.

В четвертом разделе приводятся результаты тестирования программного средства и проверяется его работоспособность.

В пятом разделе описываются действия, связанные с развертыванием и использованием программного средства.

В приложении приведены фрагменты программного кода, реализующего бизнес-логику программного средства.

Курсовой проект выполнен самостоятельно, проверен в в системе «Антиплагиат» [1]. Процент оригинальности составляет 90%. Цитирования обозначены ссылками на публикации, указанными в «Списке использованных источников№. Скриншот приведен на рисунке А.1 Приложения А.

# Анализ литературных исследований и программных решений

## Описание и анализ предметной области

Разрабатываемое программного средство предназначено для автоматизации деятельности диспетчера междугородных пассажирских перевозок автотранспортного предприятия.

Предметная область относится к организации пассажирских перевозок, под которой понимается деятельность специалистов – технологов пассажирских перевозок – по изучению спроса на перевозки, разработке прогрессивных форм удовлетворения этого спроса, рационализации маршрутной системы, упорядочению обслуживания пассажиров (включая продажу билетов и сбор проездной платы), обоснованному решению задач технологии перевозок пассажиров, диспетчеризации перевозок, учету и контролю за перевозочной деятельностью и качеством обслуживания пассажиров [2].

Услуги междугородних пассажирских перевозок оказываются автотранспортным предприятием с применением автобусного сообщения. Автобусный парк при этом включает автобусы трех классов (типов):

* автобусы малого класса имеют длину до 8 метров;
* автобусы среднего класса имеют длину до 10 метров;
* автобусы большого класса могут достигать длины 12 метров.

Пассажировместимость автобусов колеблется от 20 до 50 сидячих пасажиромест.

Междугородные перевозки автотранспортным предприятием построены на базе маршрутного принципа, который основан на совпадении интересов достаточно большого числа пассажиров и позволяет организовывать регулярные автобусные маршруты . Поездки по маршрутам практически удовлетворяют транспортные потребности подавляющего числа пассажиров при приемлемых тарифах на оказываемые услуги [3].

Сущность маршрутной технологии перевозок пассажиров состоит в организации движения подвижного состава по неизменному пути следования в виде последовательности повторяющихся циклов транспортировки – рейсов. Основные принципы маршрутной технологии:

* определенность маршрута и стабильность его трассы;
* регулярность движения транспортных средств по маршруту и преимущественная организация движения по расписанию;
* совпадение интересов пассажиров, пользующихся маршрутом, выраженное в соответствии пассажирских корреспонденций и трассы маршрута;
* предварительное, до начала движения, оформление маршрутной документации и обустройство маршрута;
* контроль за работой транспортных средств на маршруте и осуществление диспетчерского управления [4].

Таким образом, маршрут – это установленный и оборудованный путь следования подвижного состава, выполняющего регулярные перевозки.

Движение автобуса на маршруте считается регулярным, если автобус отправился в рейс точно по расписанию, своевременно проследовал все промежуточные контрольные пункты и прибыл на конечный пункт по расписанию с учетом допустимых отклонений.

Общая организация и контроль работы водителей при работе на маршрутах возлагается на диспетчера.

Деятельность диспетчера междугородных пассажирских перевозок включает следующие виды работ:

* организовывать и контролировать работу водителей и выполнение ими сменного плана и задания по перевозкам;
* инструктировать водителей об условиях и особенностях перевозок на маршрутах, уделяя при этом особое внимание состоянию дорог, особенностям дорожного движения на отдельных участках в конкретных метеорологических условиях;
* обеспечивать взаимодействие со всеми участниками перевозочного процесса с целью его оптимизации;
* координировать работу автомобильного парка, принимать меры по включению резервных автомобилей в дорожное движение на маршруте взамен преждевременно сошедших с маршрута по техническим или другим причинам, оперативному переключению автомобилей с маршрута на маршрут, на другой путь следования в связи с ремонтом дорог
* заполнять, выдавать и принимать путевые листы и другие документы, отражающие выполненную водителями работу, проверять правильность их оформления; рассчитывать в путевых листах соответствующие технико-эксплуатационные показатели;
* выдавать плановые задания, регистрировать задания и заявки на перевозки;
* составлять оперативные сводки и рапорты о работе и происшествиях за смену;
* проверять правильность оформления документов по выполненным перевозкам;
* контролировать соблюдение водителями транспортной дисциплины; организовывать в необходимых случаях оказание своевременной технической помощи подвижному составу на линии.

Цель разработки программного средства организация возможности оперативной регистрации и получения информации, применяемой диспетчером междугородних пассажирских перевозок в своей деятельности.

Достижению поставленной цели служит решение следующих задач:

* формирование базы нормативно-справочной информации, используемой диспетчером междугородних пассажирских перевозок;
* обеспечение возможности регистрации оперативной информации о плановых рейсах и назначенных водителях и транспортных средствах;
* обеспечение возможности формирования информации для пассажиров и водителей о назначенных рейсах.

Общее описание проблемы, для решения которой необходима разработка программного средства, приведено в таблице 1.1.

Таблица 1.1 – Определение проблемы предметной области

|  |  |
| --- | --- |
| Элемент | Описание |
| Проблема | Отсутствие оперативного доступа к данным о маршрутным междугородним перевозкам |
| воздействует на | * персонал автотранспортного предприятия, оказывающего услуги междугородних пассажирских перевозок * граждан, пользующихся услугами междугородних пассажирских перевозок |
| результатом чего является | * вероятность неоказания услуги * отсутствие информации о междугородних рейсах |
| Выигрыш от | создания автоматизированного рабочего места диспетчера междугородних автоматических перевозок |
| Может состоять в следующем | * формирование информации для пассажиров о расписании междугородних автобусов * формирование водителей о назначенных рейсах * возможность оперативного изменения данных о междугородних рейсах |

Таким образом, в рамках курсового проекта рассматривается автоматизация формирования ежедневного расписания, назначения водителей и транспортных средств на рейсы, формирования информации для информирования пассажиров и водителей о назначенных рейсах.

## Обзор функциональности аналогов программного средства

На сегодняшний день на рынке представлен ряд решений, позволяющих автоматизировать деятельность диспетчера междугородних пассажирских перевозок.

Аналоги программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок приведены в таблице 1.2.

Таблица 1.2 – Аналоги программного средства

|  |  |  |  |
| --- | --- | --- | --- |
| Название аналога | Компания-разработчик | *URL* официального сайта компании-  разработчика | *URL* источников для обзора |
| 1 | 2 | 3 | 4 |
| *ST Passenger Pro Client / Web* | СТ Технолоджи | *https://space-team.com/* | *https://space-team.com/ monitoring\_transporta/software/ passenger/ST\_Passenger\_Pro\_Client/* |

Окончание таблицы 1.2

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Авибус: Управление автовокзалами | ООО «Авибус» | *https://atp.avibus.pro/* | *https://atp.avibus.pro/about\_system.php* |
| *Transport Manager* | ООО «Стократ» | *https://transport-manager.ru/* | *https://transport-manager.ru/o\_programe/* |
| *StarBus CRM* | ООО «СтарБас» | *https://star-bus.ru/* | *https://partner.star-bus.ru/crm* |

*ST Passenger Pro Client / Web* – клиентское программное обеспечение системы для автоматизации перевозочного процесса и контроля над ним, получения и обработки аналитических данных о работе наземного пассажирского транспорта на автотранспортных предприятиях и в центральных диспетчерских службах городов и регионов [5].

Пример веб-версии пользовательского интерфейса *ST Passenger Pro Client / Web* приведен на рисунке 1.1.

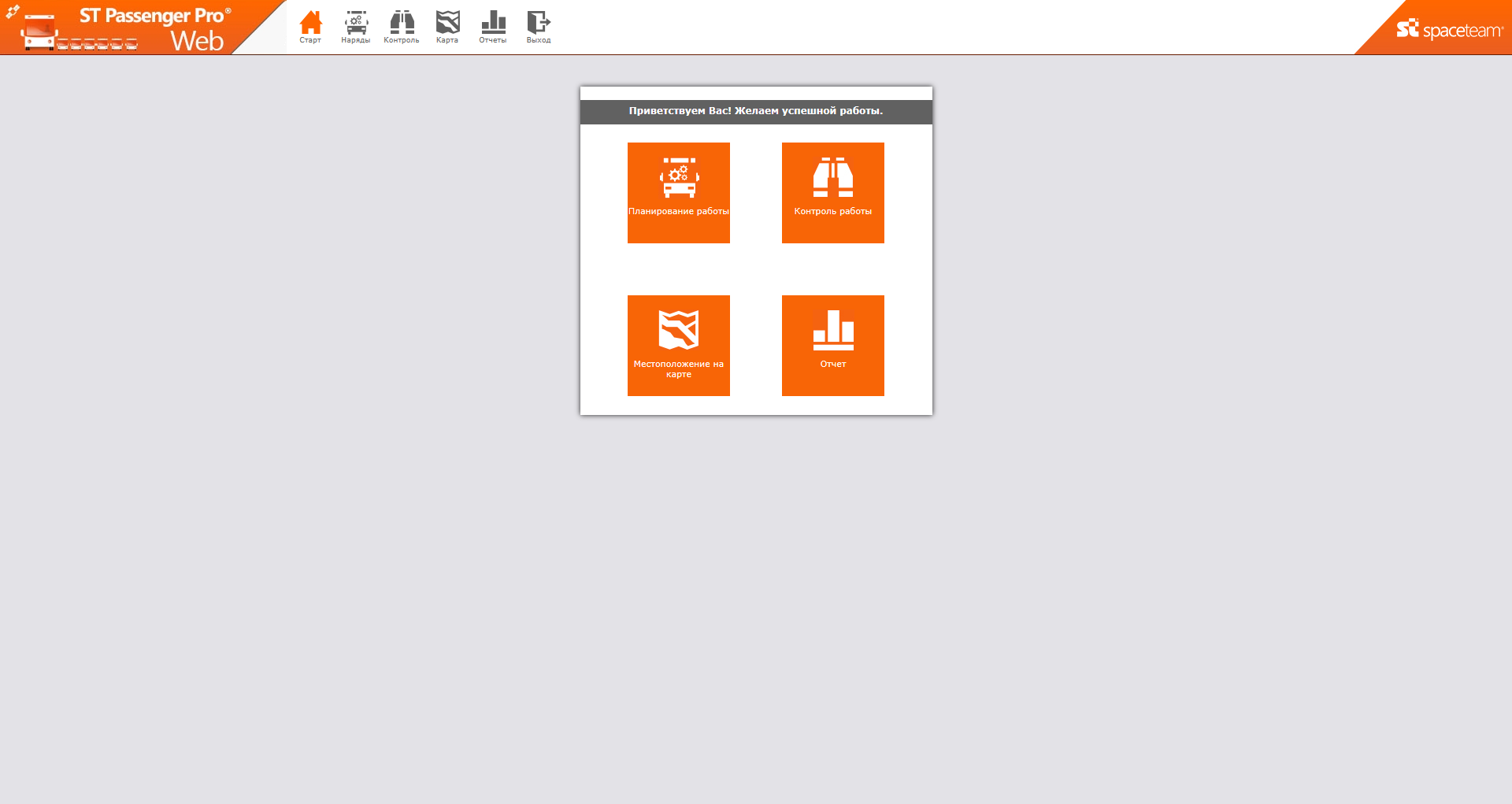


Рисунок 1.1 – Интерфейс *ST Passenger Pro Client / Web*

Программное обеспечение *ST Passenger Pro Client / Web* применяется для автоматизации рабочих мест диспетчеров, начальников служб эксплуатации, начальников служб безопасности, руководителей автотранспортных предприятий и диспетчерских служб в составе комплексной системы *ST Passenger Pro*, которая обеспечивает самые востребованные функции пассажироперевозок – диспетчеризация, контроль, учет и анализ работы пассажирского транспорта.

«Авибус: Управление автовокзалами» представляет собой автоматизированную систему управления на базе платформы «1С: Предприятие» для автовокзалов, автостанций и перевозчиков [6].

Пример пользовательского интерфейса «Авибус: Управление автовокзалами» приведен на рисунке 1.2.



Рисунок 1.2 – Интерфейс «Авибус: Управление автовокзалами»

Система имеет интерфейс платформы «1С: Предприятие», настроенный под выполнение задач автотранспортного предприятия. Сама система и дополнительные сервисы и компоненты закрывают все потребности автовокзала любого масштаба и позволяют предоставить пассажирам уже привычный для них уровень сервиса, автоматизировать рутинные задачи сотрудников и организовать работу собственных интернет-представительств и офлайн-агентов продажи билетов.

Программа управления транспортом *Transport Manager* – это современное и эффективное решение для оптимизации работы пассажирского автопарка, позволяющая управлять транспортом любого типа и размера, обеспечивая полный контроль над его эксплуатацией [7].

Пример пользовательского интерфейса *Transport Manager* приведен на рисунке 1.2.



Рисунок 1.3 – Интерфейс *Transport Manager*

Программа включает в себя широкий спектр функций, которые позволяют максимально эффективно использовать ваш транспорт:

* мониторинг и контроль за работой транспорта;
* оптимизация маршрутов и расписания движения;
* анализ затрат на эксплуатацию транспорта;
* оперативное реагирование на любые изменения в работе транспорта;
* автоматическое формирование отчетов и аналитика.

Облачная система *StarBus CRM* предназначена для автоматизации междугородных автобусных перевозок [8].

Пример интерфейса *StarBus CRM* приведен на рисунке 1.4.



Рисунок 1.4 – Интерфейс *StarBus CRM*

Сравнение функциональности аналогов приведено в таблице 1.3.

Таблица 1.3 – Функциональность аналогов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Аналог (разработчик) | Назначение аналога | Функциональность аналога | Стоимость | *URL* |
| 1 | 2 | 3 | 4 | 5 |
| *ST Passenger Pro Client / Web* (СТ Технолоджи) | Автоматизация рабочих мест диспетчеров, начальников служб эксплуатации, руководителей автотранспортных предприятий и диспетчерских служб | Запись в очередь. Вызов клиента / повторный заказ. Переадресация на другое окно / услугу. Приостановление обслуживания. Смена статуса окна | СТ Технолоджи | *https://space-team.com/* |
| Авибус: Управление автовокзалами (ООО «Авибус») | Комплексная система автоматизации автовокзала | Регистрация при помощи штрих-кода. Закрепление окна за видом транспорта. Управление воротами. Планирование маршрута. Информирование водителей | ООО «Авибус» | *https://atp.avibus.pro/* |
| *Transport Manager* (ООО «Стократ») | Управление автомобиль­ными очередями | Планирование прибытия автотранспорта на территорию. Идентификация автотранспорта.  Управление парковочным пространством.  Оперативный контроль | ООО «Стократ» | *https://transport-manager.ru/* |
| *StarBus CRM* (ООО «СтарБас») | Автоматизация междугородних автобусных перевозок | Регистрация водителей. Информирование водителей. Диспетчеризация очереди. Формирование аналитики. | ООО «СтарБас» | *https://star-bus.ru/* |

Все рассмотренные программные средства реализую базовую функциональность, но не во всех реализован доступ к функциям по ролям, адаптивный веб-интерфейс, ни одно программное средство не покрывает требуемый автотранспортному предприятию функционал.

Предполагаемая функциональность программного средства с учетом ролей пользователей приведена в таблице 1.4.

Таблица 1.4 – Предполагаемая функциональность программного средства

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Функция | Администратор | Диспетчер | Пассажир | Водитель |
| Авторизация | да | да | нет | нет |
| Управление пользователями | да | нет | нет | нет |
| Регистрация рейсов | да | да | нет | нет |
| Регистрация водителей и ТС на рейсы | да | да | нет | нет |
| Формирование расписания на дату | да | да | да | нет |
| Формирование сменного задания | да | да | нет | да |

**Выводы к разделу 1**. Существующие программные средства автоматизации управления междугородними пассажирскими перевозками, представленные на рынке, не адаптированы к фактическим потребностям автотранспортного предприятия, не удовлетворяют требуемым критериям и потребуют существенных затрат на настройку и модернизацию программного средства в соответствии с нуждами автотранспортного предприятия. Следовательно, актуальной является задача разработки нового программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок, удовлетворяющего всем критериям, приведенным в таблице 1.4.

# Моделирование предметной области и разработка требований к программному средству

## Анализ и формализация бизнес-процессов предметной области

В рамках курсового проектирования разработаны модели бизнес-процессов предметной области в нотации *BPMN* на двух уровнях представления: «как есть» (*AS IS*) и «как будет» (*TO BE*).

Модель *AS IS* описывает деятельность диспетчера междугородних пассажирских перевозок в том виде, который соответствует взаимодействию участников процесса до внедрения программного средства, разрабатываемого в курсовом проекте.

Схема бизнес-процесса диспетчеризации междугородних пассажирских перевозок в состоянии *AS IS* приведена на рисунке 2.1.



Рисунок 2.1 – Схема бизнес-процесса в состоянии *AS IS*

В состоянии *AS IS* процесс диспетчеризации междугородних пассажирских перевозок осуществляется в следующем порядке:

* диспетчер регистрирует рейсы на определенную дату (на основании нормативно-справочной информации);
* диспетчер регистрирует водителей и транспортные средства на рейсы (на основании нормативно-справочной информации);
* пассажир, имея потребность в получении расписания, формирует соответствующий запрос и направляет его диспетчеру;
* водитель, имея потребность в получении сменного задания, формирует соответствующий запрос и направляет его диспетчеру;
* диспетчер, получив запрос, формирует соответствующий ответ и направляет его запросившему.

В ходе обследования предприятия выяснено, что процесс диспетчеризации междугородних пассажирских перевозок автоматизирован частично и имеет следующие недостатки:

* информация о расписании формируется по запросу и требует непосредственного участия диспетчера в формировании ответа;
* информация о сменных нарядах формируется по запросу и требует непосредственного участия диспетчера в формировании ответа.

Это приводит к повышению трудоемкости деятельности и невозможности параллельной обработки запросов. Таким образом, имеется потребность в реорганизации процесса управления отгрузкой готовой продукции.

Модель *TO BE* описывает процесс диспетчеризации междугородних пассажирских перевозок в том виде, который соответствует взаимодействию участников процесса после внедрения программного средства, разрабатываемого в курсовом проекте. В модели *TO BE* устранены недостатки, присущие процессу в состоянии «как есть».

Схема бизнес-процесса диспетчеризации междугородних пассажирских перевозок в состоянии *TO BE* приведена на рисунке 2.2.



Рисунок 2.2 – Схема бизнес-процесса в состоянии *TO BE*

В состоянии *TO BE* процесс диспетчеризации междугородних пассажирских перевозок осуществляется в следующем порядке:

* диспетчер регистрирует рейсы на определенную дату (на основании нормативно-справочной информации) в хранилище данных рейсов и нарядов;
* диспетчер регистрирует водителей и транспортные средства на рейсы (на основании нормативно-справочной информации) в хранилище данных рейсов и нарядов;
* при поступлении запроса от пассажира расписание формируется без участия диспетчера на основании данных хранилища рейсов и нарядов;
* при поступлении запроса от водителя сменное задание формируется без участия диспетчера на основании данных хранилища рейсов и нарядов.

Исходя из описанного процесса можно сделать вывод, что оптимизация деятельности диспетчера междугородних пассажирских перевозок и повышение ее эффективности могут быть достигнуты путем полной автоматизации процесса. При помощи программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок можно ускорить и упростить регистрацию междугородних пассажирских рейсов на дату, водителей и транспортных средств на эти рейсы, формирование расписания на эти рейсы и сменных нарядов по запросам, а также уменьшить вероятность возникновения ошибок.

## Анализ требований к разрабатываемому программному средству и разработка их спецификаций

### **Документ о концепции и границах**. Данный документ определяет границы проекта, которые используются как основа для последующей разработки программного средства.

**Исходные данные**. В настоящее время диспетчеризация междугородних пассажирских перевозок на автотранспортном предприятии автоматизирована частично: информация о расписании и информация о сменных нарядах формируется по запросу и требует непосредственного участия диспетчера в формировании ответа. Это приводит к повышению трудоемкости деятельности и невозможности параллельной обработки запросов.

**Возможности бизнеса**. Для ускорения и упрощения регистрации междугородних пассажирских рейсов на дату, водителей и транспортных средств на эти рейсы, формирования расписания на эти рейсы и сменных нарядов по запросам, а также уменьшения вероятность возникновения ошибок, принято решение о разработке и внедрении в деятельность предприятия программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок.

**Бизнес-цели**. Выделены следующие бизнес-цели проекта разработки и внедрения программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок:

*BO*-1 Автоматизировать процесс диспетчеризации междугородних пассажирских перевозок в части регистрации рейсов на дату, а также водителей и транспортных средств на эти рейсы.

*BO*-2 Реализовать возможность получения пассажирами оперативной информации о расписании на определенную дату без участия диспетчера.

*BO*-3 Реализовать возможность получения водителями оперативной информации о сменном задании без участия диспетчера.

**Видение решения**. Для работников автотранспортного предприятия, которые осуществляют регистрацию рейсов междугородних автобусов, а также регистрацию водителей и транспортных средств на рейсы, этот продукт автоматизации рабочего места диспетчера междугородних пассажирских перевозок является веб-приложением, которое позволяет автоматизировать рутинные операции. В отличие от основных конкурентов, наш продукт позволит регистрировать междугородние пассажирские рейсы на дату, водителей и транспортные средства на эти рейсы (наряды), а также получать в оперативном режиме пассажирам – расписание рейсов, водителям – сменное задание.

**Бизнес-правила**. Перечень правил, которые необходимо соблюсти при разработке программного средства приведены в таблице 2.1.

Таблица 2.1 – Бизнес-правила

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Иденти­фикатор | Определение правила | Тип правила | Статическое или динамическое | Источник |
| 1 | 2 | 3 | 4 | 5 |
| *BR-*1 | Рейсы регистрируются на определенную дату по определенному маршруту | Факт | Статическое | Руково­дитель |
| *BR-*2 | Водитель и транспортное средство регистрируются на определенный рейс нарядом | Факт | Статическое | Руково­дитель |
| *BR*-3 | Водитель и транспортное средство не могут быть назначены на рейсы, дата и время начала и окончания которых пересекаются | Ограничение | Динамическое | Руково­дитель |
| *BR-*4 | Доступ к регистрации данных рейсов и нарядов имеют только авторизованные пользователи | Ограничение | Статическое | Руководитель |
| *BR*-5 | Редактировать пользователей может только администратор | Ограничение | Статическое | Руково­дитель |

Окончание таблицы 2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| *BR*-6 | Данные зарегистрированных рейсов являются основанием для формирования расписания | Факт | Статическое | Руково­дитель |
| *BR*-7 | Доступ к формированию расписания имеют все пользователи | Ограничение | Статическое | Руково­дитель |
| *BR*-8 | Данные нарядов являются основанием для формирования сменного задания | Факт | Статическое | Руково­дитель |
| *BR-*9 | Доступ к формированию сменного задания доступен авторизованным пользователям | Ограничение | Статическое | Руково­дитель |
| *BR-*10 | Доступ к формированию сменного задания неавторизованным пользователям предоставляется только при указании табельного номера | Ограничение | Статическое | Руково­дитель |

**Рамки и ограничения проекта**. Выделены следующие основные функции программного обеспечения:

*FE*-1 Авторизация пользователей.

*FE*-2 Регистрация рейсов по маршрутам на указанную дату, время.

*FE*-3 Закрепление маршрута за водителем.

*FE*-4 Закрепление транспортного средства за водителем.

*FE*-5 Регистрация водителя и транспортного средства на рейс (наряд).

*FE*-6 Формирование расписания рейсов на выбранную дату.

*FE*-7 Формирование сменного задания для выбранного водителя за выбранный период.

Выделены следующие ограничения проекта:

*LI*-1 Нормативно-справочная информация, включающая вспомогательные данные, формируется за рамками программного средства и хранится в общей базе данных. К такой информации относятся данные персонала (водителей), данные транспортных средств, данные междугородних автобусных маршрутов, данные начальных и конечных пунктов маршрутов.

*LI*-2 Водители назначаются только на рейсы закрепленных за ними маршрутов. За одним водителем может быть закреплено несколько маршрутов. Маршрут может быть закреплен за несколькими водителями.

*LI*-3 Транспортные средства назначаются только на рейсы водителей, за которыми закреплены транспортные средства. За одним водителем может быть закреплено несколько транспортных средств. Транспортное средство может быть закреплено за несколькими водителями.

**Профили заинтересованных лиц**. Список заинтересованных лиц приведен в таблице 2.2.

Таблица 2.2 – Профили заинтересованных лиц

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заинтересованное лицо | Основная ценность | Отношение | Основные интересы | Ограничения |
| Руководство компании | Увеличить производительность труда работников; сократить затраты | Полная поддержка | Экономия расходов должна превысить затраты на разработку и эксплуатацию | Не определены |
| Работники | Более эффективное использование рабочего времени | Озабоченность возможным сокращением персонала | Сохранение рабочих мест и уровня заработной платы | Необходимость обучения работников |
| Пассажиры | Оперативное получение информации, экономия времени, удобство | Положительное отношение | Простота использования, скорость получения и достоверность информации | Необходимость доступа к Интернет |

### **Пользовательские требования**. Данный раздел описывает задачи, которые пользователь может выполнять с помощью программного средства.

**Диаграмма вариантов использования**. Пользовательские требования к программному средству представлены диаграммой вариантов использования, которая приведена на рисунке 2.3.



Рисунок 2.3 – Диаграмма вариантов использования

**Описание действующих лиц**. Описание действующих лиц – пользователей программного средства приведено в таблице 2.3.

Таблица 2.3 – Описание действующих лиц

|  |  |  |
| --- | --- | --- |
| Класс пользователей | Описание | Варианты использования |
| Диспетчеры | Штатные работники предприятия, отвечающие за диспетчеризацию междугородних пассажирских перевозок | *UC*-1 Авторизоваться  *UC*-2 Регистрировать рейсы  *UC*-3 Закрепить маршрут за водителем  *UC*-4 Открепить маршрут от водителя  *UC*-5 Закрепить транспортное средство за водителем  *UC*-6 Открепить транспортное средство от водителя  *UC*-7 Регистрировать наряды  *UC*-8 Сформировать сменное задание  *UC*-9 Сформировать расписание |
| Администраторы | Штатный системный администратор предприятия | *UC*-10 Редактировать пользователей  Все варианты использования диспетчера |
| Водители | Внешние пользователи (штатные работники), участвующие в оказании услуг пассажирских перевозок | *UC*-8 Сформировать сменное задание |
| Пассажиры | Внешние пользователи – потребители услуг пассажирских перевозок | *UC*-9 Сформировать расписание |

**Описание вариантов использования**. Описание вариантов использования приведено в таблице 2.4.

Таблица 2.4 – Описание вариантов использования

|  |  |
| --- | --- |
| Параметр | Описание |
| 1 | 2 |
| Наименование | ***UC*-1 Авторизоваться** |
| Действующее лицо | Диспетчер, Менеджер, Администратор |
| Краткое описание | Получение доступа к системе |
| Входные условия | 1 Пользователь не авторизован в системе  2 Выведена главная страница |
| Основной поток действий | 1 Пользователь выбирает ссылку «Войти»  2 Пользователь вводит логин и пароль  3 Пользователь подтверждает операцию  4 Система авторизует пользователя, выводит стартовую страницу |
| Исключения при выполнении | На шаге 4 основного потока  1 Если логин, пароль некорректны, система выводит сообщение об ошибке, возврат к шагу 2 основного потока |
| Выходные условия | 1 Если выполнен успешно, пользователь авторизован в системе  2 В противном случае состояние системы не изменяется |

Продолжение таблицы 2.4

|  |  |
| --- | --- |
| 1 | 2 |
| Наименование | ***UC*-2 Регистрировать рейсы** |
| Действующее лицо | Диспетчер, Администратор |
| Краткое описание | Регистрация рейсов на определенную дату |
| Входные условия | 1 Пользователь авторизован как диспетчер или администратор |
| Основной поток действий | 1 Пользователь выбирает пункт меню «Рейсы»  2 Система выводит список рейсов  3 Пользователь выбирает операцию (добавить, изменить, удалить)  4 Система выводит форму операции  5 Пользователь вводит данные (при необходимости) и подтверждает операцию (Сохранить / Удалить)  6 Система сохраняет данные, выводит список рейсов, переход к шагу 3 |
| Альтернативный поток действий | На шаге 3 основного потока  1 Пользователь указывает условия отбора и нажимает кнопку «Применить»  2 Система выводит список рейсов в соответствии с условиями отбора, переход к шагу 3 основного потока |
| Исключения при выполнении | На шаге 6 основного потока  1 Если возникла ошибка проверки данных / допустимости операции, система сообщает пользователю о причине ошибки, возврат к шагу 5 основного потока |
| Выходные условия | 1 Данные рейсов зарегистрированы |
| Наименование | ***UC*-3 Закрепить маршрут за водителем** |
| Действующее лицо | Диспетчер, Администратор |
| Краткое описание | Регистрация связи водителя и маршрута |
| Входные условия | 1 Пользователь авторизован как диспетчер или администратор |
| Основной поток действий | 1 Пользователь выбирает пункт меню «Водители»  2 Система выводит список водителей  3 Пользователь выбирает пункт подменю «Маршруты водителя»  4 Система выводит список маршрутов водителя  5 Пользователь выбирает операцию «Добавить»  6 Система выводит форму операции  7 Пользователь выбирает маршрут и подтверждает операцию  8 Система сохраняет данные, выводит список маршрутов водителя, переход к шагу 5 |
| Альтернативный поток действий | На шаге 3 основного потока  1 Пользователь указывает условия отбора и нажимает кнопку «Применить»  2 Система выводит список водителей в соответствии с условиями отбора, переход к шагу 3 основного потока |
| Исключения при выполнении | На шаге 8 основного потока  1 Если возникла ошибка проверки данных / допустимости операции, система сообщает пользователю о причине ошибки, возврат к шагу 7 основного потока |
| Выходные условия | 1 Связь водителя и маршрута создана |

Продолжение таблицы 2.4

|  |  |
| --- | --- |
| 1 | 2 |
| Наименование | ***UC*-4 Открепить маршрут от водителя** |
| Действующее лицо | Диспетчер, Администратор |
| Краткое описание | Удаление связи водителя и маршрута |
| Входные условия | 1 Пользователь авторизован как диспетчер или администратор |
| Основной поток действий | 1 Пользователь выбирает пункт меню «Водители»  2 Система выводит список водителей  3 Пользователь выбирает пункт подменю «Маршруты водителя»  4 Система выводит список маршрутов водителя  5 Пользователь выбирает операцию «Удалить»  6 Система выводит форму подтверждения операции  7 Пользователь подтверждает операцию  8 Система сохраняет данные, выводит список маршрутов водителя, переход к шагу 5 |
| Альтернативный поток действий | На шаге 3 основного потока  1 Пользователь указывает условия отбора и нажимает кнопку «Применить»  2 Система выводит список водителей в соответствии с условиями отбора, переход к шагу 3 основного потока |
| Исключения при выполнении | На шаге 8 основного потока  1 Если возникла ошибка проверки данных / допустимости операции, система сообщает пользователю о причине ошибки, возврат к шагу 7 основного потока |
| Выходные условия | 1 Связь водителя и маршрута удалена |
| Наименование | ***UC*-5 Закрепить транспортное средство за водителем** |
| Действующее лицо | Диспетчер, Администратор |
| Краткое описание | Регистрация связи водителя и транспортного средства |
| Входные условия | 1 Пользователь авторизован как диспетчер или администратор |
| Основной поток действий | 1 Пользователь выбирает пункт меню «Водители»  2 Система выводит список водителей  3 Пользователь выбирает пункт подменю «ТС водителя»  4 Система выводит список транспортных средств водителя  5 Пользователь выбирает операцию «Добавить»  6 Система выводит форму операции  7 Пользователь выбирает транспортное средство и подтверждает операцию  8 Система сохраняет данные, выводит список транспортных средств водителя, переход к шагу 5 |
| Альтернативный поток действий | На шаге 3 основного потока  1 Пользователь указывает условия отбора и нажимает кнопку «Применить»  2 Система выводит список транспортных средств в соответствии с условиями отбора, переход к шагу 3 основного потока |
| Исключения при выполнении | На шаге 8 основного потока  1 Если возникла ошибка проверки данных / допустимости операции, система сообщает пользователю о причине ошибки, возврат к шагу 7 основного потока |
| Выходные условия | 1 Связь водителя и транспортного средства создана |

Продолжение таблицы 2.4

|  |  |
| --- | --- |
| 1 | 2 |
| Наименование | ***UC*-6 Открепить транспортное средство от водителя** |
| Действующее лицо | Диспетчер, Администратор |
| Краткое описание | Удаление связи водителя и транспортного средства |
| Входные условия | 1 Пользователь авторизован как диспетчер или администратор |
| Основной поток действий | 1 Пользователь выбирает пункт меню «Водители»  2 Система выводит список водителей  3 Пользователь выбирает пункт подменю «ТС водителя»  4 Система выводит список транспортных средств водителя  5 Пользователь выбирает операцию «Удалить»  6 Система выводит форму подтверждения операции  7 Пользователь подтверждает операцию  8 Система сохраняет данные, выводит список транспортных средств водителя, переход к шагу 5 |
| Альтернативный поток действий | На шаге 3 основного потока  1 Пользователь указывает условия отбора и нажимает кнопку «Применить»  2 Система выводит список транспортных средств в соответствии с условиями отбора, переход к шагу 3 основного потока |
| Исключения при выполнении | На шаге 8 основного потока  1 Если возникла ошибка проверки данных / допустимости операции, система сообщает пользователю о причине ошибки, возврат к шагу 7 основного потока |
| Выходные условия | 1 Связь водителя и транспортного средства удалена |
| Наименование | ***UC*-7 Регистрировать наряды** |
| Действующее лицо | Диспетчер, Администратор |
| Краткое описание | Регистрация нарядов (водителей и транспортных средств) на рейс |
| Входные условия | 1 Пользователь авторизован как диспетчер или администратор |
| Основной поток действий | 1 Пользователь выбирает пункт меню «Рейсы»  2 Система выводит список рейсов  3 Пользователь выбирает рейс и операцию (добавить наряд, изменить наряд, удалить наряд)  4 Система выводит форму операции  5 Пользователь вводит данные (при необходимости) и подтверждает операцию (Сохранить / Удалить)  6 Система сохраняет данные, выводит список рейсов, переход к шагу 3 |
| Альтернативный поток действий | На шаге 3 основного потока  1 Пользователь указывает условия отбора и нажимает кнопку «Применить»  2 Система выводит список рейсов в соответствии с условиями отбора, переход к шагу 3 основного потока |
| Исключения при выполнении | На шаге 6 основного потока  1 Если возникла ошибка проверки данных / допустимости операции, система сообщает пользователю о причине ошибки, возврат к шагу 5 основного потока |
| Выходные условия | 1 Данные нарядов зарегистрированы |

Продолжение таблицы 2.4

|  |  |
| --- | --- |
| 1 | 2 |
| Наименование | ***UC*-8 Сформировать сменное задание** |
| Действующее лицо | Диспетчер, Администратор, Водитель |
| Краткое описание | Формирование сменного задания на водителя (список нарядов) за выбранный период |
| Входные условия | 1 Пользователь авторизован как диспетчер или администратор  2 (ИЛИ) Пользователь не авторизован |
| Основной поток действий | 1 Пользователь выбирает пункт меню «Сменное задание»  2 Система выводит форму отчета «Сменное задание»  3 Пользователь указывает условия отбора и нажимает кнопку «Сформировать»  4 Система выводит список нарядов выбранного водителя за указанный период, переход к шагу 3 |
| Альтернативный поток действий | На шаге 4 основного потока  1 Если возникла ошибка проверки данных / допустимости операции, система сообщает пользователю о причине ошибки, возврат к шагу 3 основного потока |
| Выходные условия | 1 Сменное задание сформировано |
| Наименование | ***UC*-9 Сформировать расписание** |
| Действующее лицо | Диспетчер, Администратор, Водитель, Пассажир |
| Краткое описание | Формирование расписания (список рейсов) за выбранную дату |
| Входные условия | 1 Пользователь авторизован как диспетчер или администратор  2 (ИЛИ) Пользователь не авторизован |
| Основной поток действий | 1 Пользователь выбирает пункт меню «Расписание»  2 Система выводит форму отчета «Расписание»  3 Пользователь указывает дату и нажимает кнопку «Сформировать»  4 Система выводит список рейсов за указанный период, переход к шагу 3 |
| Исключения при выполнении | На шаге 4 основного потока  1 Если возникла ошибка проверки данных / допустимости операции, система сообщает пользователю о причине ошибки, возврат к шагу 3 основного потока |
| Выходные условия | 1 Расписание сформировано |
| Наименование | ***UC*-10 Редактировать пользователей** |
| Действующее лицо | Администратор |
| Краткое описание | Редактирование данных справочника пользователей |
| Входные условия | 1 Пользователь авторизован как администратор |
| Основной поток действий | 1 Пользователь выбирает пункт меню «Пользователи»  2 Система выводит список пользователей  3 Пользователь выбирает операцию (добавить, изменить, удалить)  4 Система выводит форму операции  5 Пользователь вводит данные (при необходимости) и подтверждает операцию (Сохранить / Удалить)  6 Система сохраняет данные, выводит список пользователей, переход к шагу 3 |

Окончание таблицы 2.4

|  |  |
| --- | --- |
| 1 | 2 |
| ***UC*-10 Редактировать пользователей** | |
| Альтернативный поток действий | На шаге 3 основного потока  1 Пользователь указывает условия отбора и нажимает кнопку «Применить»  2 Система выводит список пользователей в соответствии с условиями отбора, переход к шагу 3 основного потока |
| Исключения при выполнении | На шаге 6 основного потока  1 Если возникла ошибка проверки данных / допустимости операции, система сообщает пользователю о причине ошибки, возврат к шагу 5 основного потока |
| Выходные условия | 1 Данные справочника пользователей отредактированы |

### **Спецификация требований к программному средству**. Данный раздел описывает требования, предъявляемые к программному средству.

**Функциональное назначение**. Программное средство предназначено для автоматизации деятельности диспетчера междугородних пассажирских перевозок.

**Контекстная диаграмма**. Границы проекта представлены контекстной диаграммой, приведенной на рисунке 2.4.



Рисунок 2.4 – Контекстная диаграмма

**Функциональные требования**. Функциональные требования, предъявляемые к программному средству, приведены в таблице 2.5.

Таблица 2.5 – Функциональные требования

|  |  |  |
| --- | --- | --- |
| Пользовательское требование | Функциональное требование | Описание требования |
| 1 | 2 | 3 |
| ***UC*-1 Авторизоваться** | ***FR*-1 Авторизация**  .Отображение  .Запрос  .Ответ | **Получение доступа к функциям системы**  Выводится, если авторизация не выполнена.  При авторизации запрашивается логин и пароль пользователя.  При корректных данных выполняется переход на главную страницу, при некорректных – выводится сообщение об ошибке авторизации. |
| ***UC*-2 Регистрировать рейсы** | ***FR*-2 Регистрация рейсов**  .Доступ  .Отображение  .Отбор  .Добавление  .Изменение  .Удаление | **Функция регистрации междугородних рейсов пассажирского автотранспорта**  Функция доступна диспетчерам и администраторам, другие пользователи не видят пункт меню и формы.  Функция включает формы списка рейсов, добавления рейса, изменения рейса, удаления рейса. Форма списка включает основные данные, предусматривает поля для указания условий отбора и кнопки для перехода к операциям добавления, изменения, удаления рейса.  Данные списка выводятся в соответствии с отбором. Условия отбора сохраняются и восстанавливаются при следующем переходе к списку рейсов.  При добавлении и изменении выполняется проверка на полноту данных и соответствие требованиям бизнеса. При нарушении – запрет сохранения.  При удалении выполняется проверка целостности данных. При нарушении – запрет удаления. |
| ***UC*-3 Закрепить маршрут за водителем**  ***UC*-4 Открепить маршрут от водителя** | ***FR*-3 Закрепление маршрута**  .Доступ  .Отображение  .Отбор | **Функция закрепления маршрута за водителем**  Функция доступна диспетчерам и администраторам, другие пользователи не видят пункт меню и формы.  Функция включает формы списка водителей, списка маршрутов водителя, добавления маршрута водителя, удаления маршрута водителя. Форма списка водителей включает основные данные водителя, предусматривает поля для указания условий отбора и кнопку для перехода к списку маршрутов водителя. Форма списка маршрутов водителя включает данные водителя, перечень маршрутов, кнопки для добавления и удаления маршрута.  Данные списка выводятся в соответствии с отбором. Условия отбора сохраняются и восстанавливаются при следующем переходе к списку водителей. |

Продолжение таблицы 2.5

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
|  | .Добавление  .Изменение  .Удаление | При добавлении выполняется проверка на полноту данных и соответствие требованиям бизнеса. При нарушении – запрет сохранения.  При удалении выполняется проверка целостности данных. При нарушении – запрет удаления. |
| ***UC*-5 Закрепить транспортное средство за водителем**  ***UC*-6 Открепить транспортное средство от водителя** | ***FR*-4 Закрепление транспортного средства**  .Доступ  .Отображение  .Отбор  .Добавление  .Изменение  .Удаление | **Функция закрепления транспортного средства за водителем**  Функция доступна диспетчерам и администраторам, другие пользователи не видят пункт меню и формы.  Функция включает формы списка водителей, списка транспортных средств водителя, добавления транспортного средства водителя, удаления транспортного средства водителя. Форма списка водителей включает основные данные водителя, предусматривает поля для указания условий отбора и кнопку для перехода к списку транспортных средств водителя. Форма списка транспортных средств водителя включает данные водителя, перечень транспортных средств, кнопки для добавления и удаления транспортного средства.  Данные списка выводятся в соответствии с отбором. Условия отбора сохраняются и восстанавливаются при следующем переходе к списку водителей.  При добавлении выполняется проверка на полноту данных и соответствие требованиям бизнеса. При нарушении – запрет сохранения.  При удалении выполняется проверка целостности данных. При нарушении – запрет удаления. |
| ***UC*-7 Регистрировать наряды** | ***FR*-5 Регистрация нарядов**  .Доступ  .Отображение  .Отбор  .Добавление  .Изменение  .Удаление | **Функция регистрации нарядов на рейсы**  Функция доступна диспетчерам и администраторам, другие пользователи не видят пункт меню и формы.  Функция включает формы списка рейсов, добавления наряда, изменения наряда, удаления наряда. Форма списка включает основные данные, предусматривает поля для указания условий отбора и кнопки для перехода к операциям, в том числе, добавления наряда, изменения наряда, удаления наряда.  Данные списка выводятся в соответствии с отбором. Условия отбора сохраняются и восстанавливаются при следующем переходе к списку рейсов.  При добавлении и изменении выполняется проверка на полноту данных и соответствие требованиям бизнеса. При нарушении – запрет сохранения.  При удалении выполняется проверка целостности данных. При нарушении – запрет удаления. |

Окончание таблицы 2.5

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| ***UC*-8 Сформировать сменное задание** | ***FR*-6 Формирование сменного задания**  .Доступ  .Отображение | **Функция формирования сменного задания**  Функция доступна всем пользователям  Функция включает форму сменного задания, включающую поля для указания условий отбора.  Данные сменного задания выводятся в соответствии с отбором при условии корректного указания табельного номера водителя. |
| ***UC*-9 Сформировать расписание** | ***FR*-7 Формирование расписания**  .Доступ  .Отображение | **Функция формирования сменного задания**  Функция доступна всем пользователям  Функция включает форму расписания, включающую поля для указания даты отправления.  Данные расписания выводятся на указанную дату. |
| ***UC*-10 Редактировать пользователей** | ***FR*-8 Редактирование пользователей**  .Доступ  .Отображение  .Отбор  .Добавление  .Изменение  .Удаление | **Функция редактирования пользователей системы с назначением прав доступа**  Функция доступна только администраторам, другие пользователи не видят пункт меню и формы.  Функция включает формы списка, добавления, изменения, удаления. Форма списка включает основные данные, предусматривает поля для указания условий отбора и кнопки для перехода к операциям добавления, изменения, удаления.  Данные списка выводятся в соответствии с отбором. Условия отбора сохраняются и восстанавливаются при следующем переходе к списку пользователей.  При добавлении и изменении пользователя выполняется проверка на полноту данных и уникальность телефона. Пароль вводится с подтверждением. Администратор не может изменить свою роль. При нарушении – запрет сохранения.  При удалении выполняется проверка целостности данных. Администратор не может удалить свою запись. При нарушении – запрет удаления. |

**Нефункциональные требования**. Нефункциональные требования, включая требования к внешним интерфейсам, юзабилити и производительности, приведены в таблице 2.6.

Таблица 2.6 – Нефункциональные требования

|  |  |
| --- | --- |
| Идентификатор требования | Описание требования |
| 1 | 2 |
| Пользовательские интерфейсы | |
| *UI*-1 | Приложение должно иметь адаптивный интерфейс, подходящий для использования на мониторах стационарных компьютеров, ноутбуках, мобильных устройствах |

Окончание таблицы 2.6

|  |  |
| --- | --- |
| 1 | 2 |
| Пользовательские интерфейсы | |
| *UI*-2 | При разработке интерфейса необходимо использовать традиционные для веб-приложений средства управления (меню, кнопки, списки) |
| *UI*-3 | Программное средство должно быть удобным в использовании, иметь интуитивно понятный интерфейс |
| *UI*-4 | Пользовательский интерфейс программного средства должен быть эргономичным и учитывать возможности человеческого восприятия и моторики |
| *UI*-5 | Пользовательский интерфейс необходимо выполнить в лаконичном минималистическом стиле с акцентом на функциональность |
| Интерфейсы программного обеспечения | |
| *SI*-1 | Программное средство должно обеспечивать передачу идентификатора маршрута, его дату и время при регистрации рейса. |
| *SI*-2 | Программное средство должно обеспечивать передачу идентификатора рейса, идентификатора водителя, идентификатора транспортного средства, идентификатора пользователя при регистрации наряда. |
| *SI*-3 | Программное средство должно обеспечивать передачу идентификатора водителя, идентификатора маршрута или идентификатора транспортного средства при закреплении за водителем маршрута или транспортного средства соответственно. |
| *SI*-4 | Программное средство должно обеспечивать передачу даты отправления при формировании расписания рейсов. |
| *SI*-5 | Программное средство должно обеспечивать передачу периода и идентификатора водителя при формировании сменного задания. |
| Коммуникационные интерфейсы | |
| *CI*-1 | Программное средство должно выводить пользователю текстовое сообщение в случае возникновения исключений и ошибок при обработке данных |
| Требования к юзабилити | |
| *USE*-1 | Программное средство должно позволять получить пассажиру расписание всех рейсов всех маршрутов на указанную дату одним запросом. |
| *USE*-2 | Программное средство должно позволять получить водителю список всех нарядов в сменном задании одним запросом. |
| Требования к производительности | |
| *PER*-1 | Программное средство должно обслуживать до 100 пользователей одновременно |
| *PER*-2 | Все веб-страницы, генерируемые программным средством, должны полностью загружаться не более чем за 3 секунды после запроса их по интернет-подключению со скоростью 10 Мбит/сек. |
| *PER*-3 | Система должна выводить пользователю результат операции в среднем за 3 секунды и не более чем через 6 секунд после отправки запроса. |

**Требования к данным** представлены логической моделью и словарем данных.

**Логическая модель данных**. Объекты и наборы данных, которые будет обрабатывать программное средство, а также связи между ними, приведены на рисунке 2.5.



Рисунок 2.5 – Логическая модель данных

**Словарь данных**. Описание структуры данных и атрибутов приведено в таблице 2.7.

Таблица 2.7 – Словарь данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Элемент данных | Описание | Структура или тип данных | Длина | Значе­ния |
| 1 | 2 | 3 | 4 | 5 |
| Роль | Данные роли | Код + Наименование + Обозначение | 40 | 1 + Администратор + admin  2 + Диспетчер + dispatcher |
| Пользователь | Данные пользователей | Код + Логин + Пароль + Имя + Код роли + Действует | 203 |  |
| Персонал | Данные водителей | Код + Имя + Табельный номер + Действует | 131 |  |
| Тип ТС | Данные типов транспортных средств | Код + Наименование | 60 | 1 + Малый автобус  2 + Средний автобус  3 + Большой автобус |

Окончание таблицы 2.7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Населенный пункт | Данные населенных пунктов | Код + Наименование | 60 | 1 + Минск  2 + Брест  3 + Гомель  4 + Витебск  5 + Орша  6 + Могилев |
| Маршрут | Данные междугородних пассажирских маршрутов | Код + Наименование + Номер маршрута + Код типа ТС + Код места отправления + Код места прибытия + Действует | 166 |  |
| Маршрут персонала | Данные маршрутов, закрепленных за водителями | Код водителя + Код маршрута | 20 |  |
| Рейс | Данные назначенных рейсов | Код + Дата-время отправления + Код маршрута + Длительность + Действует | 41 |  |
| Транспортное средство | Данные транспортных средств | Код + Наименование + Код типа ТС + Модель + Госномер + Пассажировместимость + Действует | 173 |  |
| Транспортное средство персонала | Данные транспортных средство, закрепленных за водителями | Код водителя + Код ТС | 20 |  |
| Наряд | Данные нарядов | Код + Код рейса + Код ТС + Код водителя+ Код пользователя | 50 |  |

**Отчеты**. Программное средство формирует отчеты, описание которых приведено в таблице 2.8.

Таблица 2.8 – Отчеты, формируемые программным средством

|  |  |
| --- | --- |
| Параметр | Описание |
| 1 | 2 |
| Идентификатор отчета | *RPT*-1 |
| Заголовок отчета | Расписание междугородних пассажирских рейсов |
| Цель отчета | Вывод информации междугородним пассажирским рейсам на указанную дату отправления |
| Пользователи отчета | Пассажиры, работники автотранспортного предприятия |
| Источник данных | База данных программного средства |
| Частота и использование | Отчет генерируется по запросу. Данные в отчете статичны. Отчет отображается в веб-браузере. Можно распечатать, если устройство поддерживает печать. |
| Время доступа | Не более 6 секунд после запроса |

Окончание таблицы 2.6

|  |  |
| --- | --- |
| 1 | 2 |
| Параметры | Дата отправления |
| Визуальный макет | Таблица |
| Колонтитулы | Нет |
| Тело отчета | Заголовок: наименование и дата отправления  Поля группировки: номер и наименование маршрута  Поля и заголовки столбцов: время отправления, время прибытия, время в пути, состояние (по плану, в пути, завершен, отменен) |
| Интерактивность | Отсутствует |
| Ограничения безопасности | Доступен всем пользователям |
| Идентификатор отчета | *RPT*-2 |
| Заголовок отчета | Сменное задание |
| Цель отчета | Вывод списка нарядов водителя за выбранный период |
| Пользователи отчета | Работники автотранспортного предприятия |
| Источник данных | База данных программного средства |
| Частота использования | Отчет генерируется по запросу. Данные в отчете статичны. Отчет отображается в веб-браузере. Можно распечатать, если устройство поддерживает печать. |
| Время доступа | Не более 6 секунд после запроса |
| Параметры | Табельный номер водителя, период |
| Визуальный макет | Таблица |
| Колонтитулы | Нет |
| Тело отчета | Заголовок: наименование  Поля и заголовки столбцов: номер наряда, дата-время отправления, дата-время прибытия, время в пути, маршрут, транспортное средство, кто назначил |
| Интерактивность | Отсутствует |
| Ограничения безопасности | Доступен при корректном указании табельного номера водителя |

**Системные атрибуты программного обеспечения**, приведенные в таблице 2.9, включают требования к надежности, доступности, безопасности, удобству сопровождения / эксплуатации.

Таблица 2.9 – Системные атрибуты программного обеспечения

|  |  |  |
| --- | --- | --- |
| Идентификатор требования | | Описание требования |
| 1 | | 2 |
| Надежность | | |
| *ROB*-1 | Программное средство должно соответствовать высокому уровню надежности, при котором вероятность сбоя при обращении к основным функциям не превышает 1 % | |
| *ROB*-2 | Для программного средства должно быть настроено резервное копирование данных (как самой программы, так и базы данных) с возможностью восстановления | |

Окончание таблицы 2.9

|  |  |
| --- | --- |
| 1 | 2 |
| Доступность | |
| *AVL*-3 | Программное средство должно быть доступно 99% времени по схеме «24 на 7» за исключением времени планового обслуживания |
| Безопасность | |
| *SEC*-1 | Доступ к защищенным функциям программного средства должны иметь только авторизованные пользователи |
| *SEC*-2 | Пароли пользователей должны храниться в базе данных в зашифрованном виде, не предусматривающем декодирования (в виде хеша) |
| *SEC*-3 | Необходимо организовать защищенный обмен данными между браузером и серверной частями программного средства с использованием средств *SSL* |
| *SEC*-4 | Для защиты программного средства необходимо использовать средства антивирусной защиты и брандмауэров |
| Удобство сопровождения / эксплуатации | |
| *EXP*-1 | Программное средство должно быть реализовано на платформе .*NET Core* 8 |
| *EXP*-2 | Все данные программного средства, за исключением данных сессий пользователей, должны храниться в базе данных, находящейся под управлением СУБД *Microsoft SQL Server 2019* |
| *EXP*-3 | Для реализации использовать фреймворк *ASP.NET Core* и технологию *Entity Framework Core* |
| *EXP*-4 | Программное средство должно быть реализовано на языке программирования *C*# |

**Верификация**. Необходимо реализовать верификацию пользователя при выполнении операций с программным средством. Также необходимо реализовать валидацию данных, вводимых пользователем в формы до выполнения операций с базой данных.

## Образ предлагаемого решения

**Бизнес-проблемы**. В ходе проведенного обследования были выявлены следующие бизнес-проблемы:

* отсутствует единая информационная база, объединяющая автоматизированные рабочие места работников автотранспортного предприятия;
* низкая оперативность получения пассажирами информации о расписании междугородних рейсов;
* информация о расписании формируется по запросу и требует непосредственного участия диспетчера в формировании ответа;
* низкая оперативность получения водителями информации о рейсах, назначенных им на определенную дату;
* информация о сменных нарядах формируется по запросу и требует непосредственного участия диспетчера в формировании ответа.

Указанные бизнес-проблемы предлагается решить путем разработки программного средства, позволяющего:

* облегчить деятельность работников компании в условиях увеличивающегося потока информации;
* своевременно регистрировать междугородние пассажирские рейсы и отменять их при необходимости;
* своевременно регистрировать водителей и транспортные средства на рейсы и заменять их;
* в оперативном режиме получать информацию о расписании рейсов и назначениях водителей.

**Назначение программного средства**. Основными задачами программного средства являются регистрация междугородних пассажирских рейсов на дату, водителей и транспортных средств на эти рейсы, формирование расписания на эти рейсы и сменных нарядов по запросам.

Программное средство автоматизации рабочего места диспетчера междугородних пассажирских перевозок – это автоматизированная система, представленная в виде клиент-серверного приложения, включающего веб-сайт, веб-сервис и базу данных под управлением СУБД, размещенные на веб-сервере. Функциональная архитектура программного решения приведена на рисунке 2.4.



Рисунок 2.6 – Функциональная архитектура решения

**Цели создания**. Программное средство автоматизации рабочего места диспетчера междугородних пассажирских перевозок создается в следующих целях:

* автоматизация процесса;
* повышение эффективности деятельности диспетчеров междугородних пассажирских перевозок;
* предоставление возможности пассажирам и водителям получать необходимую информацию самостоятельно без обращения к диспетчеру.

Тогда необходимо разработать программное средство, позволяющее решать следующие задачи:

* закрепление маршрутов и транспортных средств за водителями;
* регистрация междугородних пассажирских рейсов;
* регистрация нарядов на рейсы;
* формирование расписания на указанную дату;
* формирование сменного задания водителя на указанный период.

Программное средство должно быть реализовано с использованием объектно-ориентированного языка программирования в виде клиент-серверного веб-приложения. Данные необходимо хранить в реляционной базе данных. Необходимо предусмотреть шифрование и резервное копирование данных. Пароли пользователей должны храниться в зашифрованном виде.

**Выводы по разделу 2**. Исследование текущего состояния бизнес-процесса диспетчеризации междугородних пассажирских перевозок на автотранспортном предприятии позволило выявить ряд недостатков при его организации: процесс автоматизирован частично, информация о расписании и информация о сменных нарядах формируется по запросу и требует непосредственного участия диспетчера в формировании ответа, формирование носит неоперативный характер. Определены особенности функционирования бизнес-процесса после внедрения программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок, разработана диаграмма вариантов использования и их описание, определены функциональные и нефункциональные требования к программному средству, выполнена постановка задачи.

# Проектирование и разработка программного средства

## Архитектурные решения

Разрабатываемая система представляет собой веб-приложение. Для веб-приложений характерна трехуровневая архитектура:

* клиентская часть, обеспечивающая интерфейс пользователя и взаимодействующий с серверной частью приложения;
* серверная часть, обрабатывающая запросы от клиентской части, реализующая бизнес-логику и взаимодействующая с базой данных;
* сервер баз данных, обрабатывающий запросы от серверной части к базе данных, реализующей хранение данных.

Архитектура программного средства представлена в виде модели *C*4.

Диаграмма, иллюстрирующая контекстный уровень программного средства, приведена на рисунке 3.1.

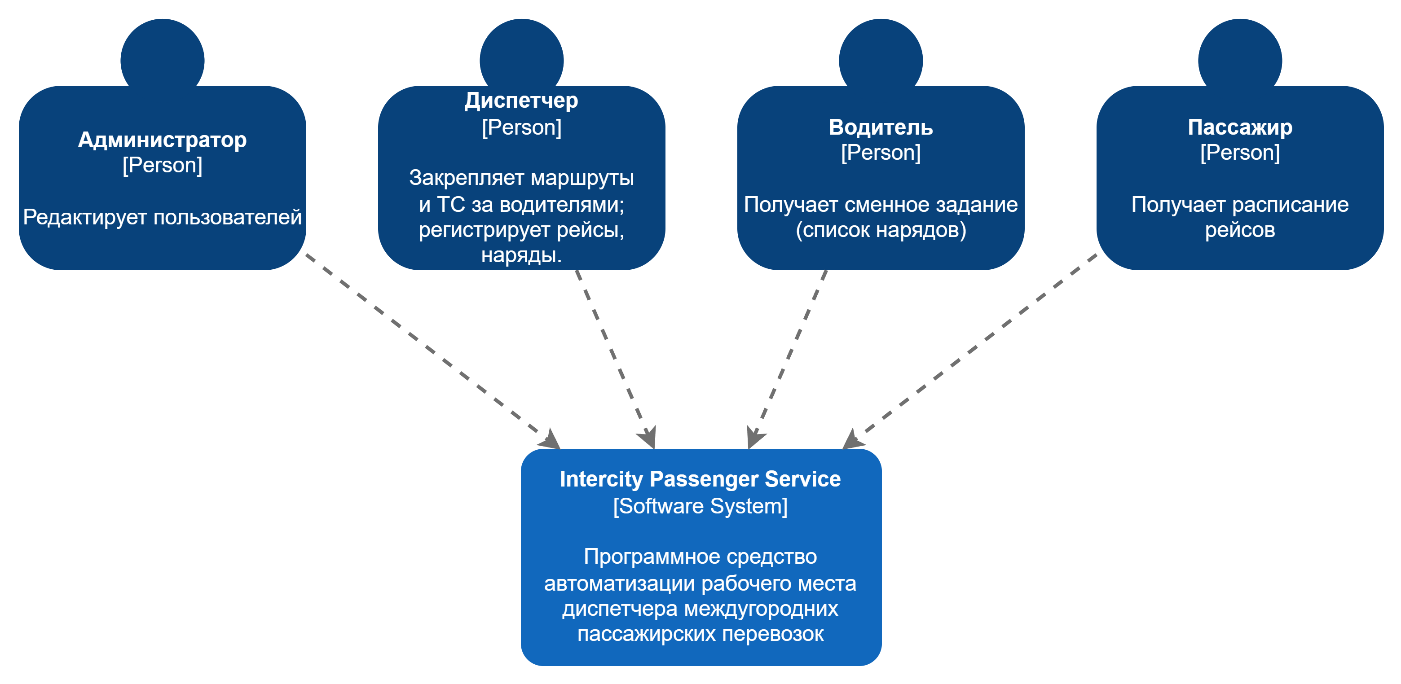


Рисунок 3.1 – Контекстный уровень программного средства

На контекстном уровне представлены:

* администратор – работник предприятия, редактирующий пользователей;
* диспетчер – работник предприятия, закрепляющий маршруты и транспортные средства за водителями, регистрирующий рейсы и наряды (основной пользователь программного средства);
* водитель – работник предприятия, получающий сменное задание (список нарядов);
* пассажир – потребитель услуги междугородних пассажирских перевозок, получающий расписание рейсов;
* *Intercity Passenger Service* – программное средство автоматизации рабочего места диспетчера междугородних пассажирских перевозок.

Диаграмма, иллюстрирующая контейнерный уровень программного средства, приведена на рисунке 3.2.

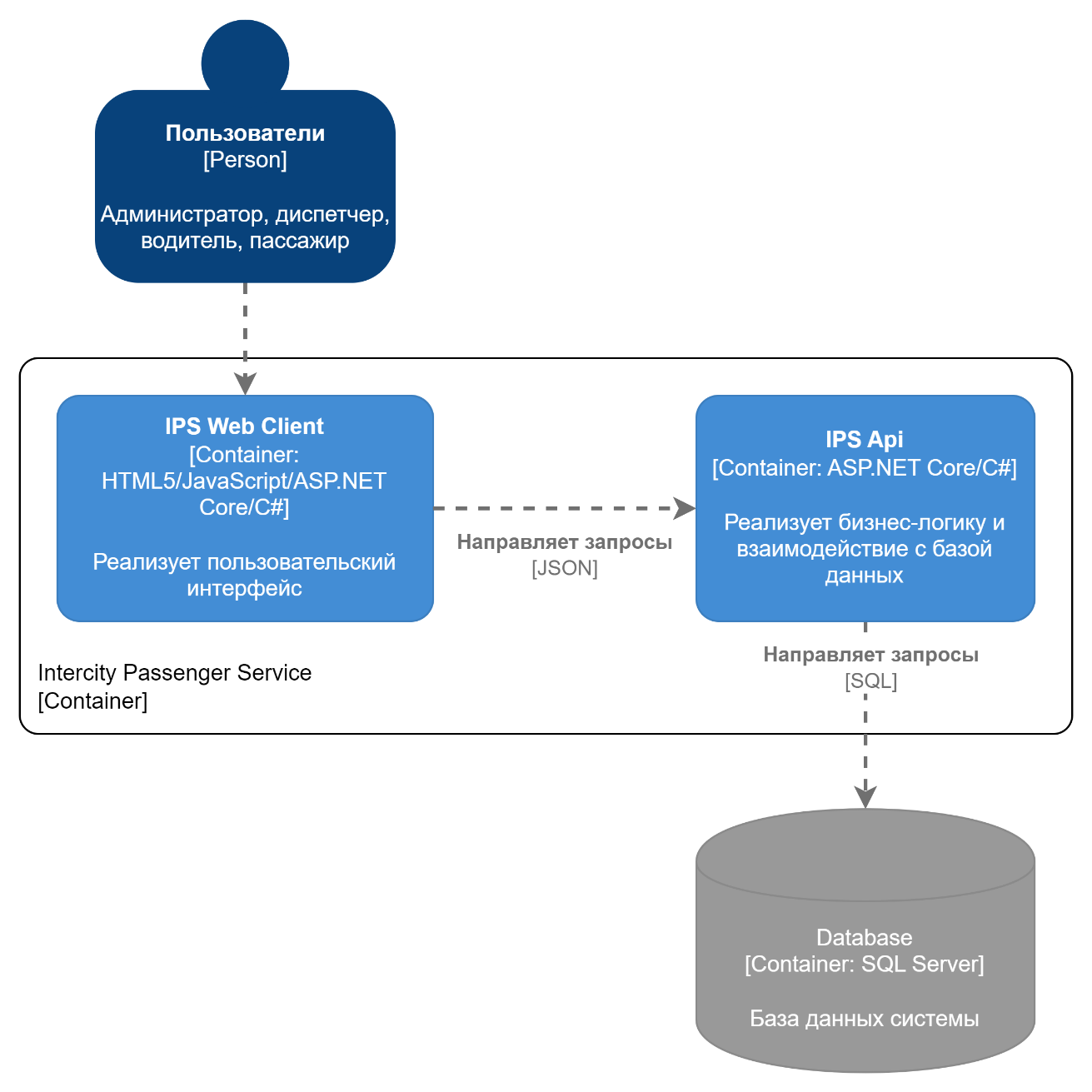


Рисунок 3.2 – Контейнерный уровень программного средства

На контейнерном уровне помимо пользователей представлены база данных формата *SQL Server* и контейнер программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок *Intercity Passenger Service*, включающий две подсистемы:

* *IPS Web Client* – подсистема, реализующая пользовательский веб-интерфейс, разработанная с использованием средств *HTML*5, *JavaScript*, *ASP.NET Core*, *C*#;
* *IPS Api* – подсистема веб-сервиса стандарта *OpenAPI*, реализующая бизнес-логику и взаимодействие с базой данных, разработанная с использованием средств *ASP.NET Core*, *C*#.

Диаграмма, иллюстрирующая компонентный уровень программного средства, приведена на рисунке 1.2.

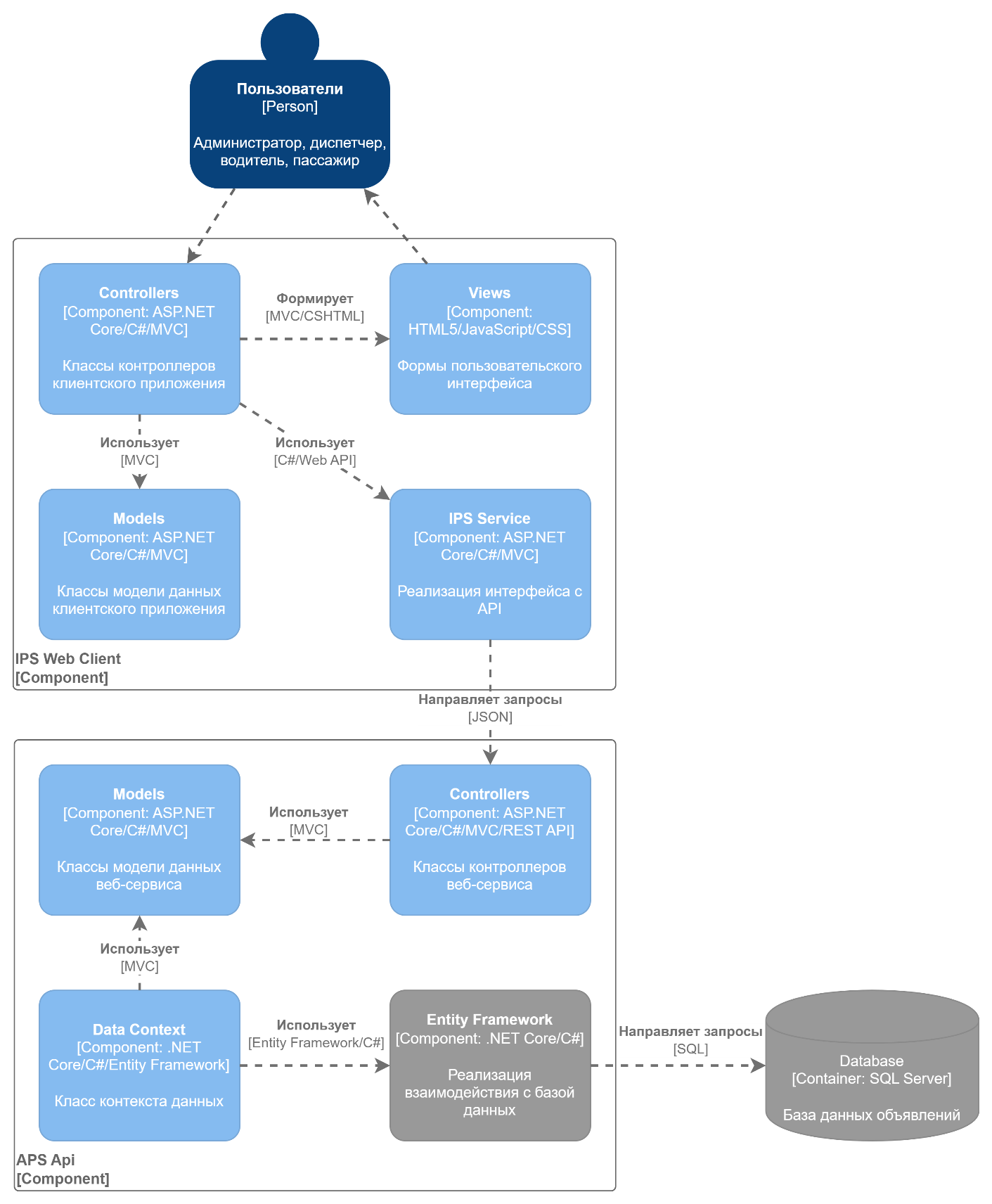


Рисунок 3.3 – Компонентный уровень программного средства

На компонентном уровне помимо пользователей представлены база данных формата *SQL Server* и компоненты подсистем *IPS Web Client* и *IPS Api*.

Подсистема *IPS Web Client* включает следующие компоненты:

* *Controllers* – классы контроллеров клиентского приложения;
* *Views* – формы пользовательского интерфейса;
* *Models* – классы модели данных клиентского приложения;
* *IPS Service* – реализация интерфейса с *API*.

Подсистема *IPS Api* включает следующие компоненты:

* *Controllers* – классы контроллеров веб-сервиса REST API;
* *Models* – классы модели данных веб-сервиса;
* *Data Context* – класс контекста данных;
* *Entity Framework* – вспомогательный компонент, реализующий взаимодействие с базой данных.

Последний уровень детализации модели *C*4 на уровне классов приведен в разделе 3.5.

## Обоснование выбора компонентов и технологий для реализации программного средства

Программное средство реализации онлайн-сервиса оценки и подбора автомобиля с пробегом необходимо разработать в виде веб-приложения. В настоящее время при разработке веб-приложений применяются следующие технологии:

* *Java Enterprise Edition* (*Java EE*) – серверная платформа, обеспечивающая функционирование модулей в соответствии со спецификацией *Java EE*;
* *.NET Framework* – программная платформа от компании *Microsoft*, предназначенная для создания настольных приложений и веб-приложений*;*
* *AMP* – комплекс серверного программного обеспечения, широко используемый в Интернет (*Apache* – веб-сервер, *MySQL* – СУБД, *PHP* – язык программирования, используемый для создания веб-приложений).

Сравнение приведенных платформ приведено в таблице 3.1.

Таблица 3.1 – Сравнение платформ разработки веб-приложений

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | *Java Platform* | *DOT.NET Core* | *AMP* |
| Кроссплатформенность | Да | Нет | Да |
| Документация на русском языке | Частично | Да | Да |
| Веб-сервер | *Apache Tomcat, GlassFish* | *IIS*, *Apache* | *Apache* |
| Сервер приложений | *Java EE* | Среда *.NET Core* | Среда *PHP* |
| СУБД | *Oracle, Sybase, MySQL* | *MS SQL, MySQL, Oracle* и др. | *MySQL* |
| Язык программирования | *Java* | *C#, VB, C++* | *PHP* |

На основании таблицы 3.1 в качестве платформы реализации выбрана *.NET Framework* по причине более высокой производительности и поддержки языков программирования. Для настоящего проекта выбрана технология *ASP.NET Core* по причине большей гибкости в реализации клиентской составляющей. Преимуществом *ASP.NET Core* в сравнении с традиционной *ASP.NET* является кроссплатформенность.

При реализации приложений на базе платформы *.NET* наиболее часто применяются следующие СУБД: *Microsoft SQL*, *MySQL*, *PostgreSQL*.

Сравнение приведенных СУБД приведено в таблице 3.2.

Таблица 3.2 – Сравнительный анализ СУБД

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | *Mirosoft SQL* | *MySQL* | *PostgreSQL* |
| Кроссплатформенность (*Windows*, *Linux*) | да / нет | да / да | да / да |
| Ограничение на использование *RAM, GB* | 64 | не ограничено | не ограничено |
| Ограничение на использование *CPU*, количество ядер | 4 | не ограничено | не ограничено |
| Лицензия | Пропиетарная | *GPL* | *GPL* |
| Скорость работы | Высокая | Высокая | Средняя |
| Поддержка в *Visual Studio* | да | нет | нет |

Таким образом, в сравнении с конкурентами СУБД *Microsoft SQL Server* имеет преимущество, так как поддерживается выбранной средой разработки, что определяет ее выбор для реализации проекта.

Сервер приложений предоставляет интерфейс *REST API*, реализованный на основе спецификации *OpenAPI* (*Swagger Specification*), которая представляет собой набор объектов *JSON* с определенной схемой, которая определяет их наименование, порядок и содержимое [26].

Для взаимодействия с базой данных серверное приложение использует технологию *Entity Framework Core* – простую, кроссплатформенную и расширяемую версию технологии доступа к данным *Entity Framework*. *Entity Framework Core* может использовать в качестве объектно-реляционного модуля сопоставления (*O/RM*), который позволяет работать с базой данных с помощью объектов *.NET Core* [27, 28].

Клиентское приложение реализовано на базе платформы *ASP.NET Core MVC*, предоставляющей основанный на шаблонах способ создания динамических веб-сайтов с разделением задач согласно паттерну проектирования *Model-View-Controller* [29].

Пользовательский веб-интерфейс реализован с помощью библиотеки *Bootstrap* [30].

Для разработки выбрана среда *Microsoft Visual Studio Community* 2022 – официальная среда разработки для платформы *.NET Core* от *Microsoft*.

Выбранная среда разработки позволяет разрабатывать проекты с использованием выбранных технологий:

* с использованием менеджера пакетов *NuGet* имеется возможность встраивания в проект актуальных версий дополнительных библиотек и фреймворков, в том числе: *EntityFramework*, *jquery*, *bootstrap*;
* имеется возможность разработки проектов *ASP.NET Core MVC*, *OpenApi* и проектов тестирования программных средств.

## Проектирование и разработка пользовательского интерфейса

Карта форм пользовательского интерфейса приведена на рисунке 3.4.



Рисунок 3.4 – Карта форм пользовательского интерфейса

Таким образом, программное средство включает следующие формы:

* стартовая страница;
* форма расписания рейсов;
* форма сменного задания;
* форма авторизации;
* формы списка, добавления, изменения, удаления пользователей;
* форма списка водителей;
* формы списка, добавления, удаления маршрутов водителя;
* формы списка, добавления, удаления транспортных средств водителя;
* формы списка, добавления, изменения, удаления рейсов;
* формы добавления, изменения, удаления нарядов.

Ниже приведены следующие планы расположения элементов пользовательского интерфейса:

* при ширине 992 px и более (рисунок 3.5);
* при ширине 575-991 px (рисунок 3.6);
* при ширине 574 px и менее (рисунок 3.7).



Рисунок 3.5 – Страница списка данных при ширине 992 px и более



Рисунок 3.6 – Страница списка данных при ширине 575-991 px



Рисунок 3.7 – Страница списка данных при ширине 574 px и менее

Пример дизайна пользовательского интерфейса приведен на рисунке 3.8.

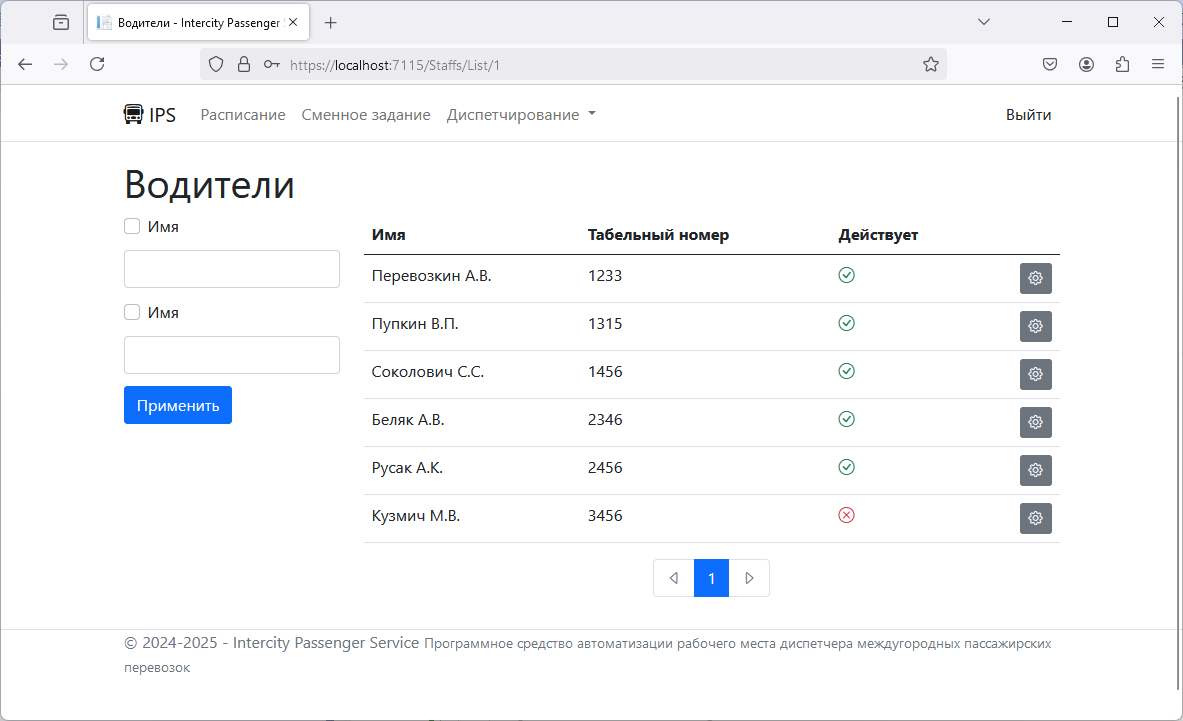


Рисунок 3.8 – Пример дизайна пользовательского интерфейса

Все формы программного средства имеют единый дизайн.

Диаграмма *User flow* логики действий диспетчера (основного пользователя) при работе с программным средством приведена на рисунке 3.9.



Рисунок 3.9 – User flow логики действий пользователя

Порядок действий пользователя зависит от результата, который ему необходимо получить, и способа его достижения:

* привязку маршрутов к водителю можно редактировать, выбрав список маршрутов водителя;
* привязку транспортных средств к водителю можно редактировать, выбрав список транспортных средств водителя;
* регистрировать рейсы можно, выбрав список рейсов;
* регистрировать наряды можно, выбрав рейс в списке рейсов.

## Разработка модели данных

Концептуальная модель данных приведена в разделе 2.3.3 на рисунке 2.5.

Схема логической модели базы данных, включающая сущности, их атрибуты и связи между ними в нотации *IDEF1.X*, приведена на рисунке 3.10. Связи между сущностями представлены в нотации *Crow’s Feet*.



Рисунок 3.10 – Схема логической модели базы данных

Сущности модели проектируемой базы данных приведены в таблице 3.3.

Таблица 3.3 – Сущности модели проектируемой базы данных

|  |  |
| --- | --- |
| Имя | Определение |
| *Role* (Роль) | Роли учетных записей пользователей |
| *User* (Пользователь) | Пользователи программного средства, имеющие доступ к защищенным функциям |
| *UserPreference* (Настройка пользователя) | Сериализованные данные настроек пользователя (отборы в списках) |
| *Place* (Населенный пункт) | Населенные пункты, являющиеся началом и окончанием междугородних пассажирских маршрутов |
| *TransportType* (Тип ТС) | Типы транспортных средств |
| *Staff* (Персонал) | Водители (работники предприятия) |
| *Transport* (Транспортное средство) | Автобусы, на которых осуществляются междугородние пассажирские перевозки |
| *StaffTransport* (ТС персонала) | Транспортные средства, привязанные к водителям |
| *Way* (Маршрут) | Междугородние пассажирские маршруты |
| *StaffWay* (Маршрут персонала) | Маршруты, привязанные к водителям |
| *WaySchedule* (Рейс) | Рейсы по маршрутам, назначенные на определенную дату-время |
| *Order* (Наряд) | Водители и транспортные средства, назначенные на конкретные рейсы |

Разработанная база данных нормализована:

* соответствует первой нормальной форме (1*NF*), т.к. все атрибуты атомарны;
* соответствует второй нормальной форме (2*NF*), т.к. находится в первой нормальное форме и каждый неключевой атрибут находится в полной функциональной зависимости от ключа;
* соответствует третьей нормальной форме (3*NF*), т.к. находится во второй нормальной форме и каждый неключевой атрибут находится в нетранзитивной зависимости от первичного ключа.

Описание базы данных приведено в таблице 3.4.

Таблица 3.4 – Описание таблиц базы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица | Атрибут сущности | Поле таблицы | Тип данных | Ключ |
| 1 | 2 | 3 | 4 | 5 |
| *Role* (Роль) | Код | *Id* | *int* | *PK* |
| Наименование | *Name* | *nvarchar*(20) |  |
| Обозначение | *Code* | *nvarchar*(10) | *U*1 |
| *User* (Пользователь) | Код | *Id* | *int* | *PK* |
| Логин | *Login* | *nvarchar*(50) |  |
| Пароль | *Password* | *nvarchar*(32) |  |
| Ф.И.О. | *Name* | *nvarchar*(100) |  |
| Код роли | *RoleId* | int | *FK*1 |
| Действует | *Enabled* | *bit* |  |

Окончание таблицы 4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 45 | 5 |
| *UserPreference* (Настройка пользователя) | Код | *Id* | *int* | *PK* |
| Код пользователя | *UserId* | *int* | *FK*1 |
| Настройка | *Preference* | *int* |  |
| Данные | *JSON* | *nvarchar(max)* |  |
| *Place* (Населенный пункт) | Код | *Id* | *int* | *PK* |
| Наименование | *Name* | *nvarchar*(50) |  |
| *TransportType* (Тип ТС) | Код | *Id* | *int* | *PK* |
| Наименование | *Name* | *nvarchar*(50) |  |
| *Staff* (Персонал) | Код | Id | *int* | *PK* |
| Ф.И.О. | *Name* | *nvarchar*(100) |  |
| Табельный номер | *StaffNumber* | *nvarchar*(20) |  |
| Действует | *Enabled* | *bit* |  |
| *Transport* (Транспортное средство) | Код | *Id* | int | *PK* |
| Наименование | *Name* | *nvarchar*(100) |  |
| Код типа ТС | *TransportTypeId* | *int* | *FK*1 |
| Модель | *Model* | *nvarchar*(100) |  |
| Госномер | *GosNumber* | *nvarchar*(100) |  |
| Пассажировместимость | *Capacity* | *int* |  |
| Действует | *Enabled* | *bit* |  |
| *StaffTransport* (ТС персонала) | Код персонала | *StaffId* | *int* | *PK, FK*1 |
| Код ТС | *TransportId* | *int* | *PK, FK*2 |
| *Way* (Маршрут) | Код | *Id* | *int* | PK | |
| Наименование | *Name* | *nvarchar*(115) |  | |
| Номер маршрута | *Model* | *nvarchar*(10) |  | |
| Код типа ТС | *TransportTypeId* | *int* | *FK*1 | |
| Код места отправления | *PlaceStartId* | *int* | *FK*2 | |
| Код места прибытия | *PlaceFinishId* | *int* | *FK*3 | |
| Действует | *Enabled* | *bit* |  | |
| *StaffWay* (Маршрут персонала) | Код персонала | *StaffId* | *int* | *PK, FK*1 | |
| Код маршрута | *WayId* | *int* | *PK, FK*2 | |
| *WaySchedule* (Рейс) | Код | *Id* | *int* | *PK* | |
| Дата-время отправления | *DateStart* | *datetime* |  | |
| Код маршрута | *WayId* | *int* | *FK*1 | |
| Длительность | *Duration* | *int* |  | |
| Действует | *Enabled* | *bit* |  | |
| *Order* (Наряд) | Код | *Id* | *int* | *PK* | |
| Код рейса | *WayScheduleId* | *int* | *FK*1 | |
| Код ТС | *TransportId* | *int* | *FK*2 | |
| Код водителя | *StaffId* | *int* | *FK*3 | |
| Код пользователя | *UserId* | *int* | *FK*4 | |

Схема физической модели данных, разработанная на основании логической модели данных приведена на рисунке 3.11.



Рисунок 3.11 – Схема физической модели базы данных

Приведенная схема показывает таблицы реляционной базы данных, их поля и типы данных, первичные и внешние ключи, а также связи между таблицами. Разработанная база данных соответствует третьей нормальной форме. На основании модели базы данных с помощью схемы *CodeFirst* технологии *EntityFramework* создана база данных программного средства.

## Описание статических и динамических аспектов поведения программных объектов

Базовая диаграмма классов с разделение на пакеты приведена на рисунке 3.12.



Рисунок 3.12 – Диаграмма пакетов

На диаграмме представлены следующие классы и пакеты:

* *IPSApi* – приложение веб-сервиса, реализующего обработку данных и предоставляющего интерфейс клиентской части по стандарту *OpenApi*;
* *IPSService* – класс интерфейса к сервису *OpenApi*;
* *IPSWebClient* – клиентское приложение.

Диаграмма классов веб-сервиса приведена на рисунке 3.13.



Рисунок 3.13 – Диаграмма классов веб-сервиса

Диаграмма классов клиентского приложения приведена на рисунке 3.14.



Рисунок 3.14 – Диаграмма классов контроллеров

Диаграмма деятельности для варианта использования «Закрепить маршрут за водителем» приведена на рисунке 3.15.



Рисунок 3.15 – Диаграмма деятельности варианта использования  
«Закрепить маршрут за водителем»

Деятельность варианта использования «Закрепить маршрут за водителем» осуществляется в следующем порядке:

* пользователь переходит к списку водителей;
* система выводит список водителей;
* пользователь переходит к списку маршрутов водителя;
* система выводит список маршрутов водителя;
* пользователь выбирает операцию «Добавить маршрут водителя»;
* система выводит форму добавления маршрута водителя;
* пользователь выбирает маршрут и подтверждает операцию;
* система обрабатывает запрос на добавление маршрута водителя и переходит выводу списка маршрутов водителя.

Диаграмма последовательности для варианта использования «Закрепить маршрут за водителем» приведена на рисунке 3.16.



Рисунок 3.16 – Диаграмма последовательности варианта использования  
«Закрепить маршрут за водителем»

Список сообщений между объектами приведен в таблице 3.5.

Таблица 3.5 – Описание сообщений между объектами

|  |  |  |  |
| --- | --- | --- | --- |
| Номер сообщения | Объект-отправитель | Объект-получатель | Имя сообщения |
| 1 | 2 | 3 | 4 |
|  | Диспетчер | *IPSWebClient. StaffsController* | *List* |
|  | *IPSWebClient. StaffsController* | *IPSService* | *GetStaffListAsync* |
|  | *IPSService* | *IPSApi.StaffsController* | *GetStaffsList* |
|  | *IPSApi.StaffsController* | *IPSContext* | *Staff.Query* |

Окончание таблицы 3.5

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
|  | Диспетчер | *IPSWebClient. StaffsController* | *Way* |
|  | *IPSWebClient. StaffsController* | *IPSService* | *GetStaffWayAsync* |
|  | *IPSService* | *IPSApi.StaffsController* | *GetStaffWay* |
|  | *IPSApi.StaffsController* | *IPSContext* | *StaffWay.Query* |
|  | Диспетчер | *IPSWebClient. StaffsController* | *WayCreate* |
|  | *IPSWebClient. StaffsController* | *IPSService* | *GetWaySelectAsync* |
|  | *IPSService* | *IPSApi.ActionController* | *GetWaySelect* |
|  | *IPSApi.ActionController* | *IPSContext* | *Way.Query* |
|  | Диспетчер | *IPSWebClient. StaffsController* | *WayCreate* |
|  | *IPSWebClient. StaffsController* | *IPSService* | *SetStaffWayAsync* |
|  | *IPSService* | *IPSApi.StaffsController* | *SetStaffWay* |
|  | *IPSApi.StaffsController* | *IPSContext* | *StaffWay.Query* |

Диаграмма состояний объекта с условиями отбора приведена на рисунке 7.1.



Рисунок 3.17 – Диаграмма состояний объекта с условиями подбора

Приведенная на рисунке 7.1 диаграмма отражает порядок изменения состояний объекта filter, содержащего условия отбора в списках:

* формирование объекта с данными «по умолчанию» – наступает, когда инициирован вывод формы отбора;
* ввод данных отбора пользователем – наступает, когда выведена форма отбора в форме списке или обнаружены некорректные данные в форме;
* проверка данных на корректность – наступает, когда пользователь подтвердил отправку формы с условиями отбора;
* ожидание ответа на запрос от веб-сервиса – наступает, когда условия отбора направлены в веб-сервис;
* корректировка данных отбора по результатам запроса – наступает, когда получен ответ от веб-сервиса.

Приведенным диаграммам соответствует большая часть функций программного средства.

## Разработка и описание алгоритмов, реализующих бизнес-логику разрабатываемого программного средства

Схема алгоритма формирования данных расписания веб-сервисом приведена на рисунке 3.18.

Приведенный алгоритм предполагает следующий порядок действий:

* получение дату из запроса;
* определение даты начала отбора рейсов;
* определение даты окончания отбора рейсов;
* определение пустого списка результата;
* получение из БД списка всех доступных маршрутов;
* начало цикла 1: для каждого маршрута из списка маршрутов;
* получение из БД рейсов маршрута за указанный период;
* если число рейсов больше нуля, добавление к списку результата объекта с данными маршрута и списком рейсов;
* завершение цикла 1;
* возврат списка результата.

Алгоритм реализуется методом *GetSchedule*() класса *WaySchedulesController* веб-сервиса:

public async Task<ActionResult<IEnumerable<ScheduleView>>> GetSchedule(DateTime date)

{

var dateAt = new DateTime(date.Year, date.Month, date.Day, 0, 0, 0);

var dateTo = new DateTime(date.Year, date.Month, date.Day, 23, 59, 59);

var list = new List<ScheduleView>();

var ways = await \_context.Ways.Include(p => p.TransportType).Where(p => p.Enabled).OrderBy(p => p.WayNumber).ToListAsync();

foreach (var way in ways)

{

var waySchedules = await \_context.WaySchedules

.Include(p => p.Way).Include( p => p.Way.PlaceStart).Include(p => p.Way.PlaceFinish)

.Where(p => p.WayId == way.Id && p.DateStart >= dateAt && p.DateStart <= dateTo).OrderBy(p => p.DateStart).ToListAsync();

if (waySchedules.Count > 0)

{

list.Add(new ScheduleView { Way = way, WaySchedules = waySchedules });

}

}

return list;

}



Рисунок 3.18 – Схема алгоритма формирования данных расписания

Схема алгоритма формирования данных сменного задания (список нарядов водителя за период) веб-сервисом приведена на рисунке 3.19.



Рисунок 3.19 – Схема алгоритма формирования данных сменного задания

Приведенный алгоритм предполагает следующий порядок действий:

* получение данных фильтра из запроса;
* определение даты начала отбора рейсов по фильтру;
* определение даты окончания отбора рейсов по фильтру;
* если дата окончания меньше даты начала, определение даты окончания отбора рейсов по дате начала;
* получение из БД списка нарядов водителя по отбору;
* возврат списка нарядов водителя.

Алгоритм реализуется методом *GetStaffOrders*() класса *StaffsController* веб-сервиса:

public async Task<ActionResult<IEnumerable<Order>>> GetStaffOrders(StaffOrdersFilter filter)

{

var dateAt = new DateTime(filter.DateAt.Year, filter.DateAt.Month, filter.DateAt.Day, 0, 0, 0);

var dateTo = new DateTime(filter.DateTo.Year, filter.DateTo.Month, filter.DateTo.Day, 23, 59, 59);

if (dateTo < dateAt) dateTo = new DateTime(dateAt.Year, dateAt.Month, dateAt.Day, 23, 59, 59);

var list = await \_context.Orders

.Include(p => p.Transport).Include(p => p.Transport.TransportType)

.Include(p => p.Staff)

.Include(p => p.WaySchedule).Include(p => p.WaySchedule.Way)

.Include(p => p.WaySchedule.Way.TransportType)

.Include(p => p.WaySchedule.Way.PlaceStart)

.Include(p => p.WaySchedule.Way.PlaceFinish)

.Include(p => p.User).Include(p => p.User.Role)

.Where(p => p.StaffId == filter.StaffId

&& p.WaySchedule.DateStart >= filter.DateAt

&& p.WaySchedule.DateStart <= filter.DateTo

&& p.WaySchedule.Enabled == true)

.OrderBy(p => p.WaySchedule.DateStart)

.ToListAsync();

return list;

}

Прочие алгоритмы программного средства реализованы аналогично.

## Разработка программного средства

Описание основных запросов к серверу и ответов на них приведено в таблице 3.6.

Таблица 3.6 – Описание запросов к серверу и ответов на них

|  |  |
| --- | --- |
| Параметр | Описание |
| 1 | 2 |
| Функциональное требование | ***FR*-1 Авторизация** |
| *URL* | *POST https://localhost:7158/api/Action/GetAuth* |

Продолжение таблицы 3.6

|  |  |
| --- | --- |
| 1 | 2 |
| Пример запроса | *POST /api/Action/GetAuth*  *host: localhost:7158*  *Content-Type: application/json*  *{*  *"Login": "admin",*  *"Password": "c4ca4238a0b923820dcc509a6f75849b"*  *}* |
| Пример ответа | *200 OK*  *Content-type: application/json; charset=utf-8*  *{*  *"id": 1,*  *"login": "admin",*  *"name": "Админов А.А.",*  *"role": {*  *"id": 1,*  *"name": "Администратор",*  *"code": "admin"*  *},*  *"enabled": true*  *}* |
| Функциональное требование | ***FR*-2 Регистрация рейсов** |
| *URL* | *PUT https://localhost:7158/api/WaySchedules/16* |
| Пример запроса | *PUT /api/WaySchedules/16*  *host: localhost:7158*  *Content-Type: application/json*  *{*  *"id": 16,*  *"dateStart": "2025-01-16T19:25:00",*  *"wayId": 4,*  *"duration": 170,*  *"enabled": true*  *}* |
| Пример ответа | *200 OK*  *Content-length: 0* |
| Функциональное требование | ***FR*-3 Закрепление маршрута** |
| *URL* | *POST https://localhost:7158/api/Staffs/SetStaffWay* |
| Пример запроса | *POST /apiStaffs/SetStaffWay*  *host: localhost:7158*  *Content-Type: application/json*  *{*  *"staffId": 1,*  *"wayId": 3*  *}* |
| Пример ответа | *200 OK*  *Content-length: 0* |

Продолжение таблицы 3.6

|  |  |
| --- | --- |
| 1 | 2 |
| Функциональное требование | ***FR*-4 Закрепление транспортного средства** |
| *URL* | *POST https://localhost:7158/api/Staffs/GetStaffTransport* |
| Пример запроса | *POST /api/Staffs/GetStaffTransport*  *host: localhost:7158*  *Content-Type: application/json*  *{*  *"staffId": 1*  *}* |
| Пример ответа | *200 OK*  *Content-type: application/json; charset=utf-8*  *[*  *{*  *"id": 2,*  *"name": "МАЗ-251 AB 5678-7",*  *"transportTypeId": 3,*  *"transportType": { … },*  *"model": "МАЗ-251",*  *"gosNumber": "AB 5678-7",*  *"capacity": 45,*  *"enabled": true*  *},*  *{*  *"id": 3,*  *"name": "МАЗ-231 AB 6789-7",*  *"transportTypeId": 3,*  *"transportType": { … },*  *"model": "МАЗ-231",*  *"gosNumber": "AB 6789-7",*  *"capacity": 51,*  *"enabled": true*  *}*  *]* |
| Функциональное требование | ***FR*-5 Регистрация нарядов** |
| *URL* | *POST https://localhost:7158/api/Orders* |
| Пример запроса | *POST /api/Orders*  *host: localhost:7158*  *Content-Type: application/json*  *{*  *"wayScheduleId": 10,*  *"transportId": 2,*  *"staffId": 2,*  *"userId": 3,*  *}* |
| Пример ответа | *200 OK*  *Content-type: application/json; charset=utf-8*  *10* |

Продолжение таблицы 3.6

|  |  |
| --- | --- |
| 1 | 2 |
| Функциональное требование | ***FR*-6 Формирование сменного задания** |
| *URL* | *POST https://localhost:7158/api/Staffs/GetStaffOrders* |
| Пример запроса | *POST / api/Staffs/GetStaffOrders*  *host: localhost:7158*  *Content-Type: application/json*  *{*  *"dateAt": "2025-01-15",*  *"dateTo": "2025-01-15",*  *"staffId": 1*  *}* |
| Пример ответа | *200 OK*  *Content-type: application/json; charset=utf-8*  *[*  *{*  *"id": 1,*  *"wayScheduleId": 1,*  *"waySchedule": {*  *"id": 1,*  *"dateStart": "2025-01-15T06:30:00",*  *"wayId": 1,*  *"way": {*  *"id": 1,*  *"name": "9102 Минск - Брест",*  *"wayNumber": "9102",*  *"transportTypeId": 3,*  *"transportType": { … },*  *"placeStartId": 1,*  *"placeStart": { … },*  *"placeFinishId": 2,*  *"placeFinish": { … },*  *"enabled": true*  *},*  *"duration": 295,*  *"enabled": true*  *},*  *"transportId": 1,*  *"transport": {*  *"id": 1,*  *"name": "МАЗ-251 AB 4567-7",*  *"transportTypeId": 3,*  *"transportType": { … },*  *"capacity": 45,*  *"enabled": true*  *},*  *…*  *}*  *]* |

Окончание таблицы 3.6

|  |  |
| --- | --- |
| 1 | 2 |
| Функциональное требование | ***FR*-7 Формирование расписания** |
| *URL* | *POST https://localhost:7158/api/WaySchedules/GetSchedule* |
| Пример запроса | *POST /api/WaySchedules/GetSchedule*  *host: localhost:7158*  *Content-Type: application/json*  *{*  *"date": "2025-01-15"*  *}* |
| Пример ответа | *200 OK*  *Content-type: application/json; charset=utf-8*  *[*  *{*  *"way": {*  *"id": 1,*  *"name": "9102 Минск - Брест",*  *"wayNumber": "9102",*  *"transportTypeId": 3,*  *"transportType": { … },*  *"placeStartId": 1,*  *"placeStart": { … },*  *"placeFinishId": 2,*  *"placeFinish": { … },*  *"enabled": true*  *},*  *"waySchedules": [*  *{*  *"id": 1,*  *"dateStart": "2025-01-15T06:30:00",*  *"wayId": 1,*  *"duration": 295,*  *"enabled": true*  *},*  *{*  *"id": 2,*  *"dateStart": "2025-01-15T12:10:00",*  *"wayId": 1,*  *"duration": 310,*  *"enabled": true*  *}*  *]*  *…*  *}*  *]* |

Примеры программного кода приведены в приложении Б.

## Механизмы обеспечения информационной безопасности

Программное средство предполагает использование следующих механизмов обеспечения информационной безопасности.

Доступ к функциям программного средства ограничен логином и паролем. Программное средство доступно по протоколу *https*.

Пароль в базе данных хранится в зашифрованном виде, не предусматривающем дешифрацию. Для шифрования применяется алгоритм *MD*5 (программный код функции приведен в приложении Б).

Пароль между веб-клиентом и веб-сервисом передается в зашифрованном виде.

**Выводы по разделу 3**. Выполнено проектирование и разработка программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок: спроектирована архитектура системы, выбраны компоненты и технологии для реализации, определен перечень форм программного средства и спроектированы макеты пользовательского интерфейса, разработаны схемы логической и физической моделей базы данных, реализована модель данных, разработаны диаграммы классов, разработаны диаграмма деятельности, диаграмма последовательности, диаграмма состояний иллюстрирующие типовое взаимодействие между объектами программного средства, разработаны алгоритмы, реализующие бизнес-логику программного средства, и программный код. Определены механизмы обеспечения информационной безопасности.

# Тестирование и проверка работоспособности программного средства

Тест-кейсы для проверки уровня базовых пользовательских требований приведены в таблице 4.1.

Таблица 4.1 – Тест-кейсы для проверки уровня базовых пользовательских требований

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор тест-кейса | Заглавие тест-кейса | Шаги тест-кейса | Ожидаемый результат |
| 1 | 2 | 3 | 4 |
| *TC*-1 | Авторизоваться | 1 Открыть клиентское приложение в веб-браузере  2 Выбрать ссылку «Войти»  3 Ввести логин, пароль, нажать кнопку «Войти» | 1 Вывод главной страницы  2 Вывод формы авторизации  3 Вывод главной страницы с пунктом меню «Диспетчеризация» или сообщения об ошибке авторизации |
| *TC*-2 | Регистрировать рейсы | 1 Выбрать пункт меню «Рейсы»  2 Указать условия отбора, нажать кнопку «Применить»  3 Нажать кнопку операции (добавить рейс, изменить рейс, удалить рейс)  4 Ввести данные рейса, подтвердить операцию | 1 Вывод формы списка рейсов  2 Вывод списка рейсов в соответствии с отбором  3 Вывод формы операции  4 Вывод обновленного списка рейсов или сообщения об ошибке операции |
| *TC*-3 | Закрепить маршрут за водителем | 1 Выбрать пункт меню «Водители»  2 Указать условия отбора, нажать кнопку «Применить»  3 Нажать кнопку «Маршруты водителя»  4 Нажать кнопку операции (добавить маршрут водителя)  5 Ввести данные маршрута водителя, подтвердить операцию | 1 Вывод формы списка водителей  2 Вывод списка водителей в соответствии с отбором  3 Вывод формы списка маршрутов водителя  4 Вывод формы операции  5 Вывод обновленного списка маршрутов водителя или сообщения об ошибке операции |

Продолжение таблицы 4.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| *TC*-4 | Открепить маршрут от водителя | 1 Выбрать пункт меню «Водители»  2 Указать условия отбора, нажать кнопку «Применить»  3 Нажать кнопку «Маршруты водителя»  4 Нажать кнопку операции (удалить маршрут водителя)  5 Подтвердить операцию | 1 Вывод формы списка водителей  2 Вывод списка водителей в соответствии с отбором  3 Вывод формы списка маршрутов водителя  4 Вывод формы операции  5 Вывод обновленного списка маршрутов водителя или сообщения об ошибке операции |
| *TC*-5 | Закрепить транспортное средство за водителем | 1 Выбрать пункт меню «Водители»  2 Указать условия отбора, нажать кнопку «Применить»  3 Нажать кнопку «ТС водителя»  4 Нажать кнопку операции (добавить ТС водителя)  5 Ввести данные ТС водителя, подтвердить операцию | 1 Вывод формы списка водителей  2 Вывод списка водителей в соответствии с отбором  3 Вывод формы списка ТС водителя  4 Вывод формы операции  5 Вывод обновленного списка ТС водителя или сообщения об ошибке операции |
| *TC*-6 | Открепить транспортное средство от водителя | 1 Выбрать пункт меню «Водители»  2 Указать условия отбора, нажать кнопку «Применить»  3 Нажать кнопку «ТС водителя»  4 Нажать кнопку операции (удалить ТС водителя)  5 Подтвердить операцию | 1 Вывод формы списка водителей  2 Вывод списка водителей в соответствии с отбором  3 Вывод формы списка ТС водителя  4 Вывод формы операции  5 Вывод обновленного списка ТС водителя или сообщения об ошибке операции |
| *TC*-7 | Регистрировать наряды | 1 Выбрать пункт меню «Рейсы»  2 Указать условия отбора, нажать кнопку «Применить»  3 Нажать кнопку операции (добавить наряд, изменить наряд, удалить наряд)  4 Ввести данные наряда, подтвердить операцию | 1 Вывод формы списка рейсов  2 Вывод списка рейсов в соответствии с отбором  3 Вывод формы операции  4 Вывод обновленного списка рейсов или сообщения об ошибке операции |

Окончание таблицы 4.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| *TC*-8 | Сформировать сменное задание | 1 Выбрать пункт меню «Сменное задание»  2 Указать период и табельный номер водителя, нажать кнопку «Сформировать» | 1 Вывод формы отчета «Сменное задание» без данных  2 Вывод данных отчета «Сменное задание» в соответствии с отбором или сообщения об ошибке |
| *TC*-9 | Сформировать расписание | 1 Выбрать пункт меню «Расписание»  2 Указать дату, нажать кнопку «Сформировать» | 1 Вывод формы отчета «Расписание» на текущую дату  2 Вывод данных отчета «Расписание» в соответствии с указанной датой |
| *TC*-10 | Редактировать пользователей | 1 Выбрать пункт меню «Пользователи»  2 Указать условия отбора, нажать кнопку «Применить»  3 Нажать кнопку операции (добавить, изменить, удалить)  4 Ввести данные пользователя, подтвердить операцию | 1 Вывод формы списка справочника пользователей  2 Вывод списка пользователей в соответствии с отбором  3 Вывод формы операции  4 Вывод обновленного списка пользователей или сообщения об ошибке операции |

Все тест-кейсы пройдены успешно.

**Выводы к разделу 4**. Все тест-кейсы программного средства пройдены успешно. Полученные в ходе тестирования результаты соответствуют ожидаемым. Все функциональные требования к программному средству выполняются. Следовательно, разработанное программное средство удовлетворяет требованиям задания на разработку.

# Руководство по установке (развертыванию) и использованию программного средства

## Руководство по установке (развертыванию) программного средства

Программное средство представлено в виде двух отдельных приложений:

* *IPSApi* – серверная часть программного средства;
* *IPSWebClient* – клиентская часть программного средства.

Настройки серверной части программного средства приведены в файле *IPSApi/appsettings.json*:

{

"ConnectionStrings": {

"DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=IPSApi7158;Trusted\_Connection=True;MultipleActiveResultSets=true"

},

"Urls": "https://localhost:7158;http://localhost:5294",

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*"

}

Параметр *ConnectionStrings* определяет настройки подключения к базе данных. В тестовом примере используется база данных *Microsoft SQL Server* 2019 в редакции *LocalDB* с именем *IPSApi7158*. База данных создается при первом обращении программного средства к базе данных. При промышленной эксплуатации в параметре *ConnectionStrings* необходимо указать сервер и имя рабочей базы данных, а также прочие параметры, в том числе параметры доступа.

Параметр *Urls* определяет адрес и порт, по которому запускается и доступна серверная часть (в тестовом варианте *https://localhost:7158*).

Настройки клиентской части программного средства приведены в файле *IPSWebClient/appsettings.json*:

{

"IPSApiUrl": "https://localhost:7158/",

"Urls": "https://localhost:7115;http://localhost:5013",

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*"

}

Параметр *IPSApiUrl* определяет адрес и порт, по которому доступна серверная часть приложения (в тестовом варианте *https://localhost:7158*).

Параметр *Urls* определяет адрес и порт, по которому запускается и доступна клиентская часть (в тестовом варианте *https://localhost:7115*).

Для запуска серверной части необходимо запустить файл *IPSApi\IPSApi.exe*. Скриншот окна с запущенной серверной частью приведен на рисунке 5.1.

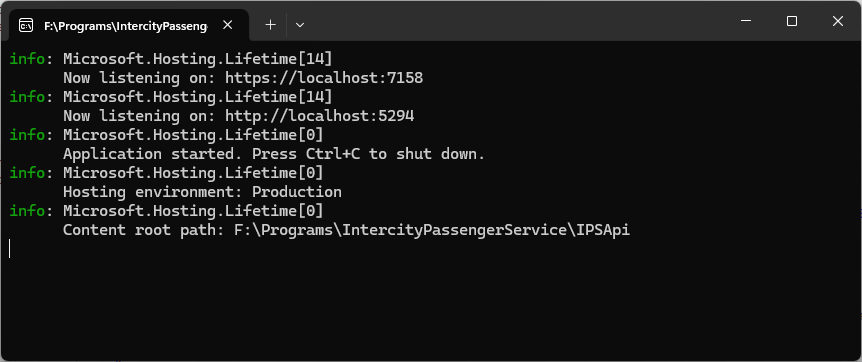


Рисунок 5.1 – Окно приложения серверной части

Для запуска клиентской части необходимо запустить файл *IPSWebClient\IPSWebClient.exe*. Скриншот окна с запущенной клиентской частью приведен на рисунке 5.2.

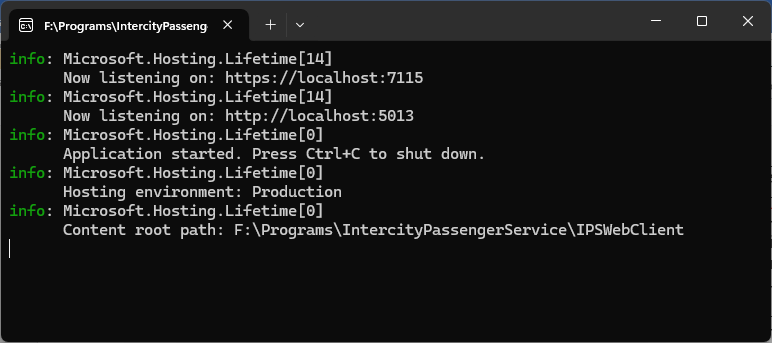


Рисунок 5.2 – Окно приложения клиентской части

Для доступа к клиентской части необходимо в адресной строке браузера указать адрес приложения (в тестовом варианте *https://localhost:7115*).

## Руководство пользователя

При переходе по адресу программного средства в веб-браузере открывается стартовая страница веб-приложения для неавторизованного пользователя (см. рисунок 5.3).

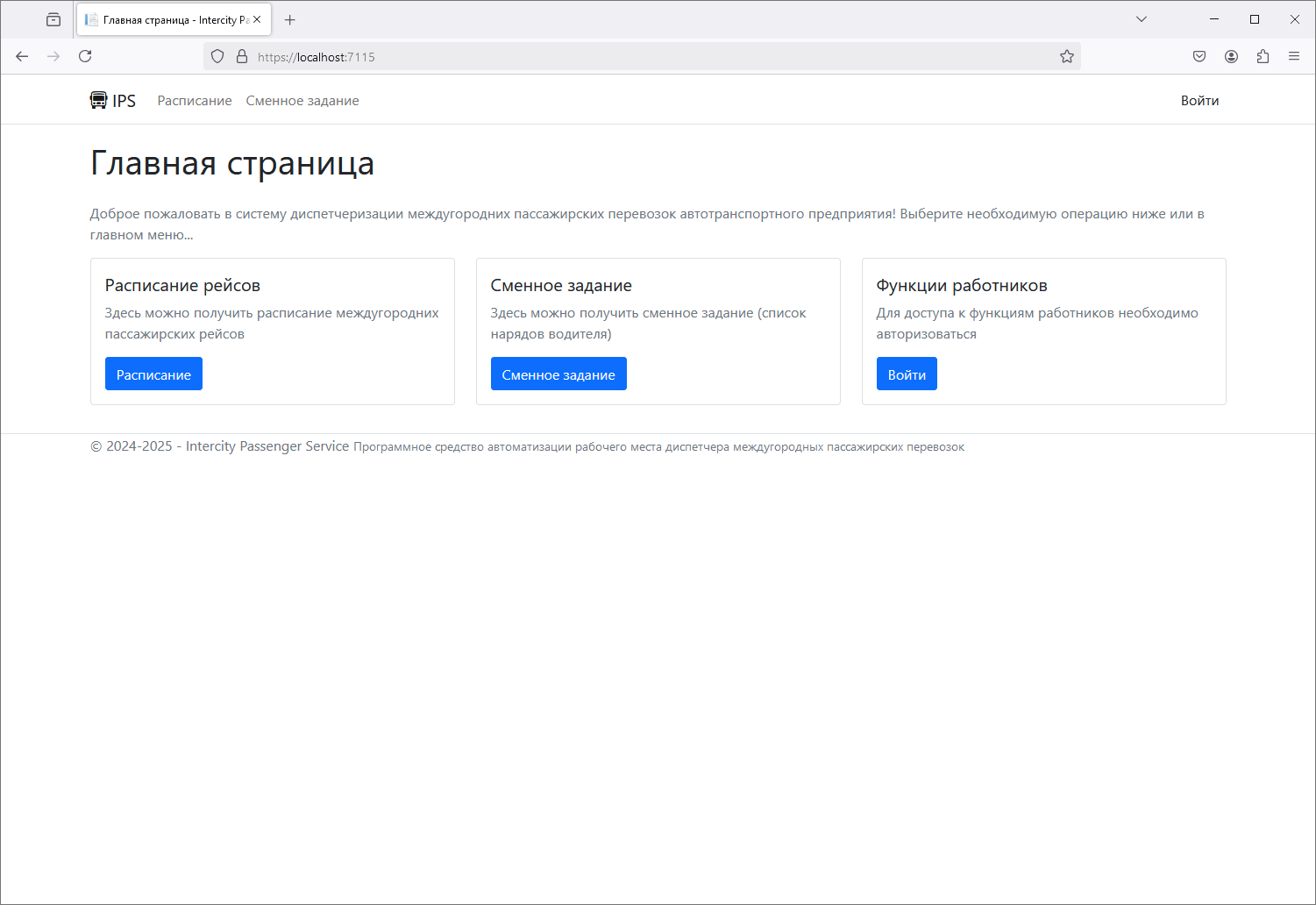


Рисунок 5.3 –Стартовая страница неавторизованного пользователя

Для формирования расписания необходимо нажать кнопку «Расписание» на стартовой странице или выбрать соответствующий пункт меню. Далее в форме необходимо указать дату и нажать кнопку «Сформировать» (см. рисунок 5.4).

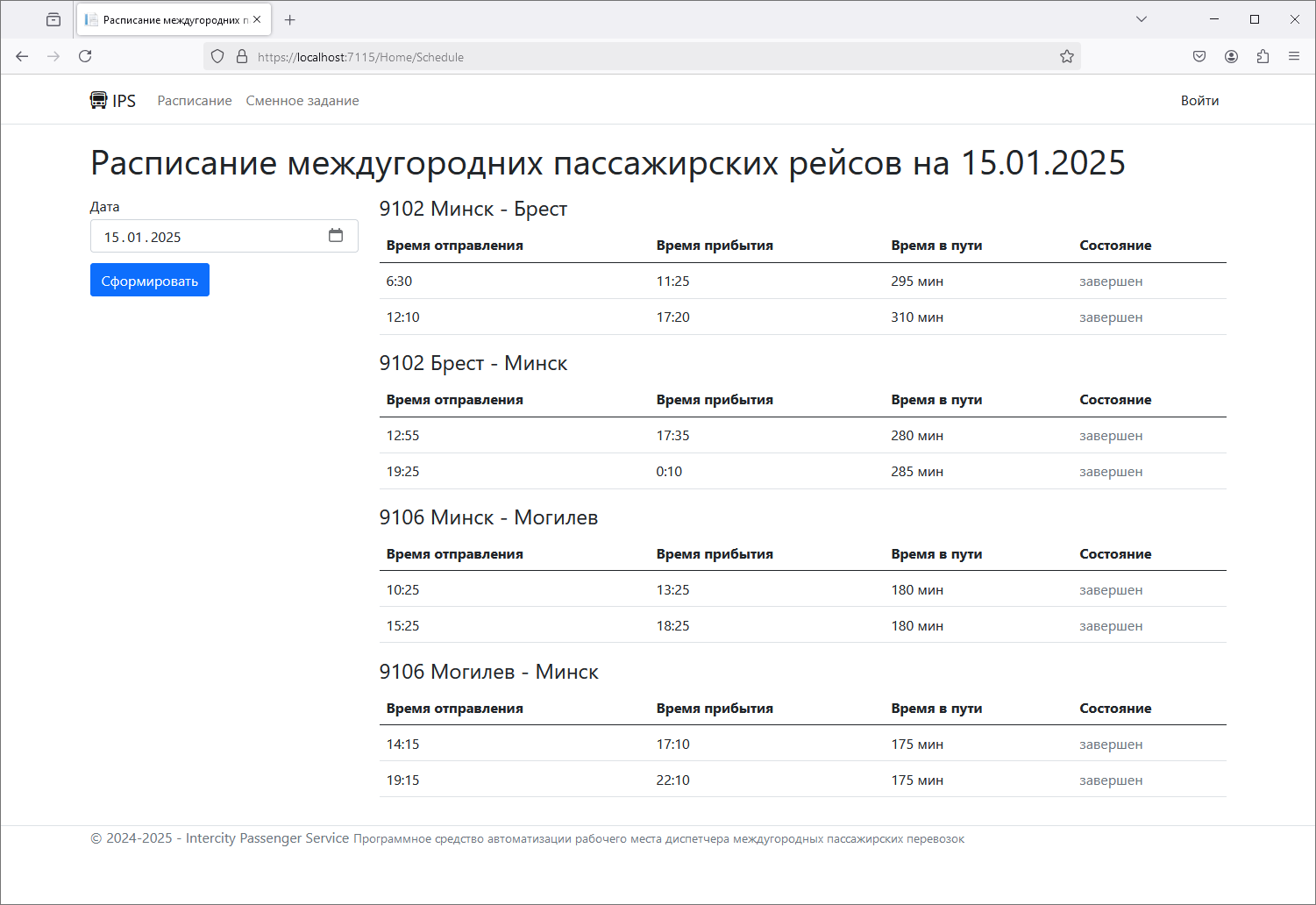


Рисунок 5.4 – Форма расписания

Для формирования сменного задания необходимо нажать кнопку «Сменное задание» на стартовой странице или выбрать соответствующий пункт меню. Далее в форме необходимо указать табельный номер, период и нажать кнопку «Сформировать» (см. рисунок 5.5).

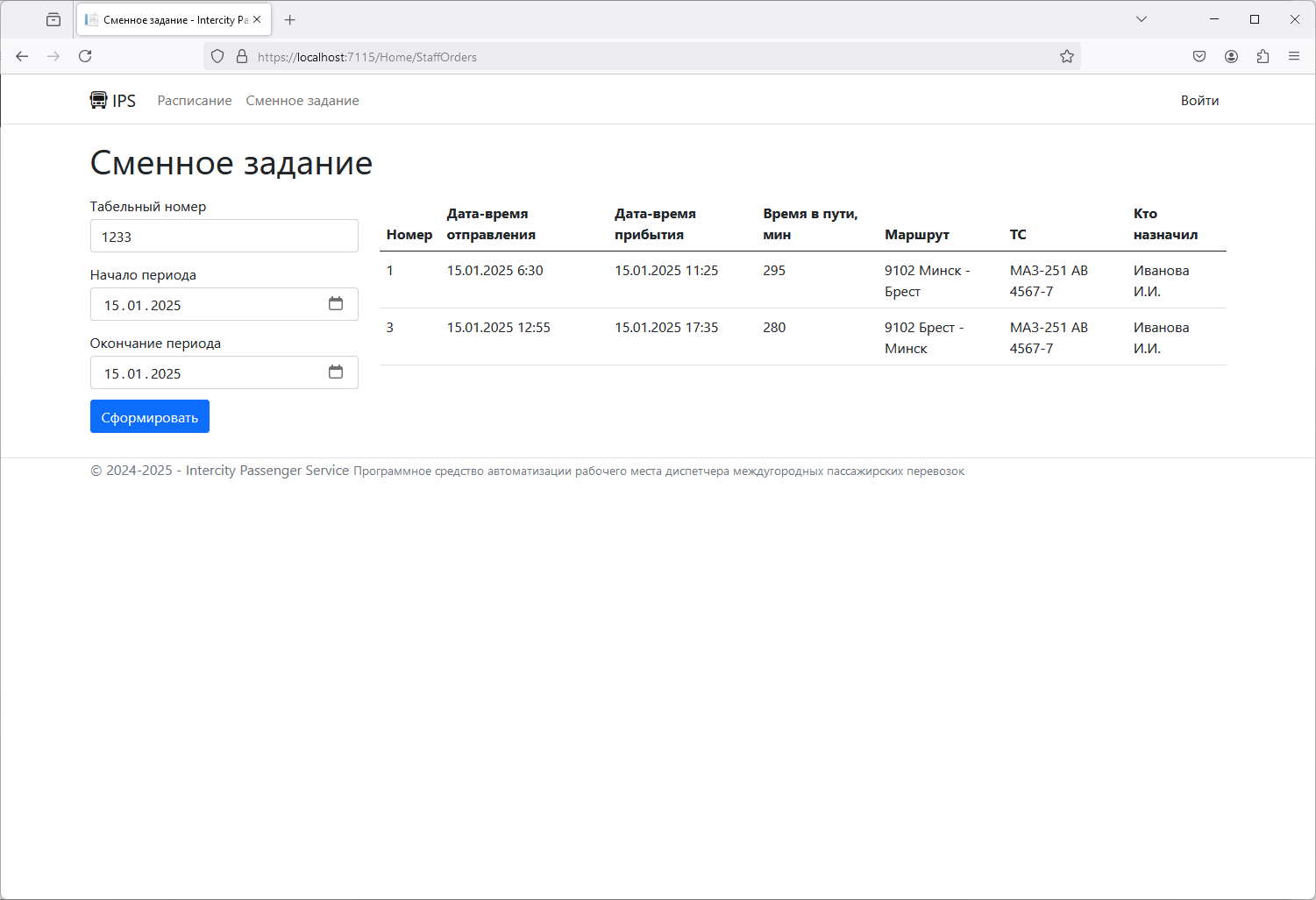


Рисунок 5.5 – Форма сменного задания

Для работников предприятия работа начинается с авторизации пользователя. Для авторизации необходимо нажать кнопку «Войти» на стартовой странице или выбрать соответствующий пункт меню. Форма авторизации приведена на рисунке 5.6. Для доступа к системе пользователь должен указать логин и пароль.

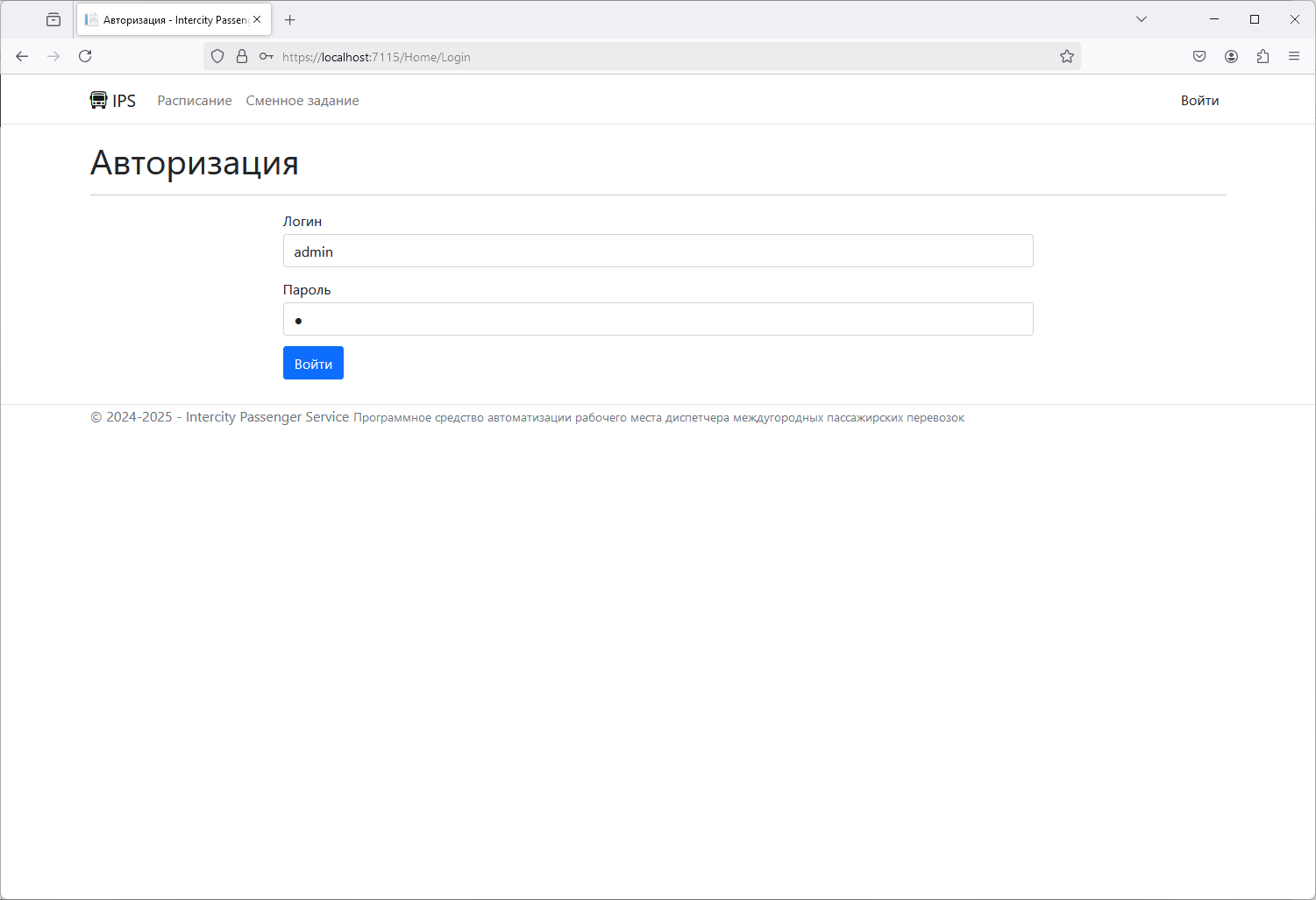


Рисунок 5.6 – Форма авторизации

При создании базы данных добавляется пользователь-администратор с полными правами (логин: *admin*, пароль: 1).

В случае успешной авторизации пользователь получает доступ к системе (см. рисунок 5.7).

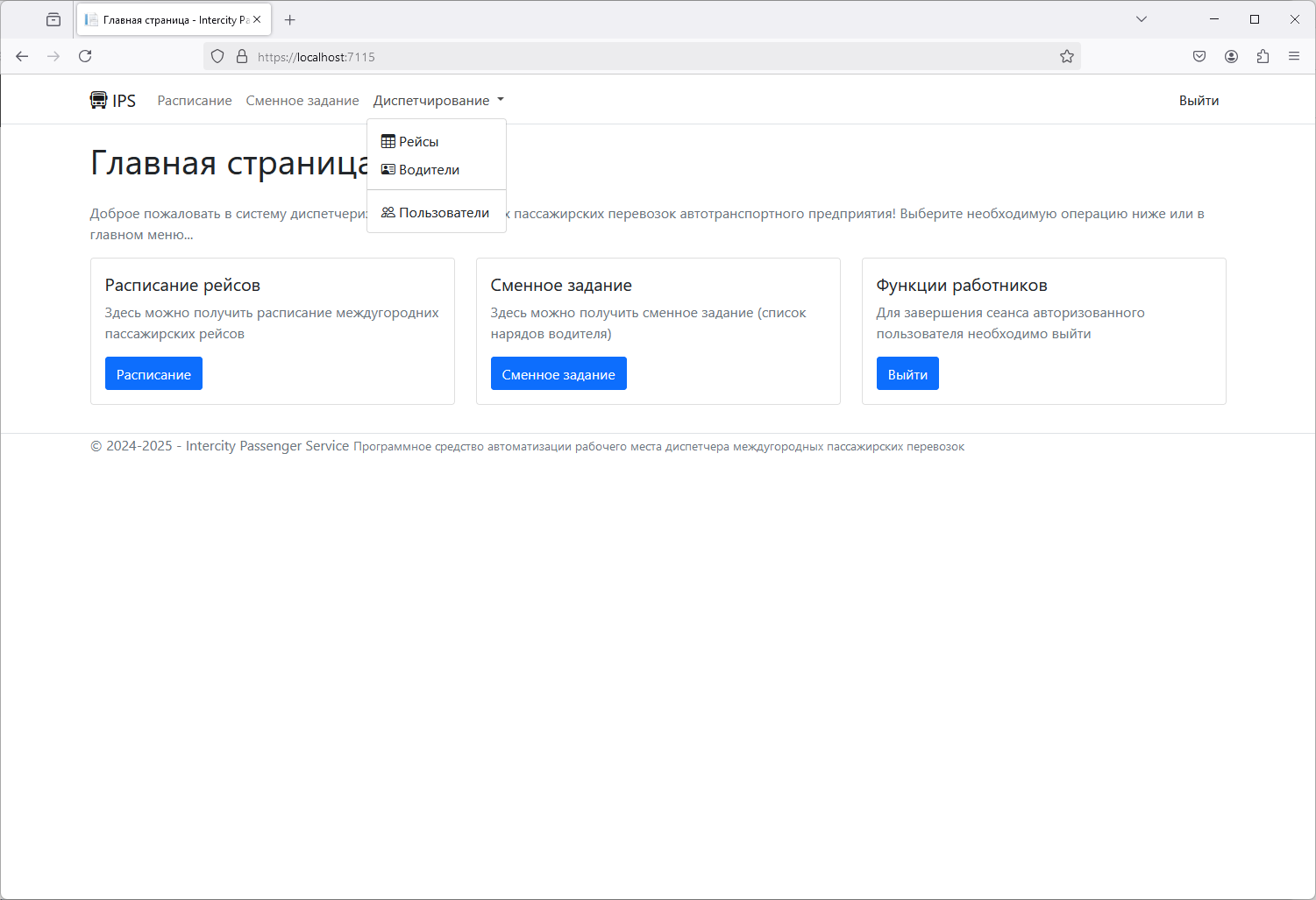


Рисунок 5.7 – Стартовая страница авторизованного пользователя

Редактирование справочников и регистрация документов доступны пользователям в зависимости от роли:

* администратору доступны все функции;
* диспетчеру доступны все функции, за исключением редактирования пользователей.

Редактирования справочников и регистрации документов рассмотрено на примере закрепления маршрутов за водителями.

Для редактирования данных необходимо выбрать пункт меню «Диспетчеризация» и необходимый подпункт: рейсы, водители, пользователи.

При выборе подпункта меню «Диспетчеризация – Водители» открывается форма списка водителей (см. рисунок 5.8).

Для отбора в списке необходимо в форме отбора выбрать параметры, по которым требуется отбор, указать их значения и нажать кнопку «Применить».

Для перехода к списку маршрутов водителя необходимо найти водителя в списке, нажать кнопку «Операции» и выбрать подпункт «Маршруты водителя». В результате открывается формы списка маршрутов водителя (см. рисунок 5.9). Кнопка «К списку» возвращает к списку водителей.

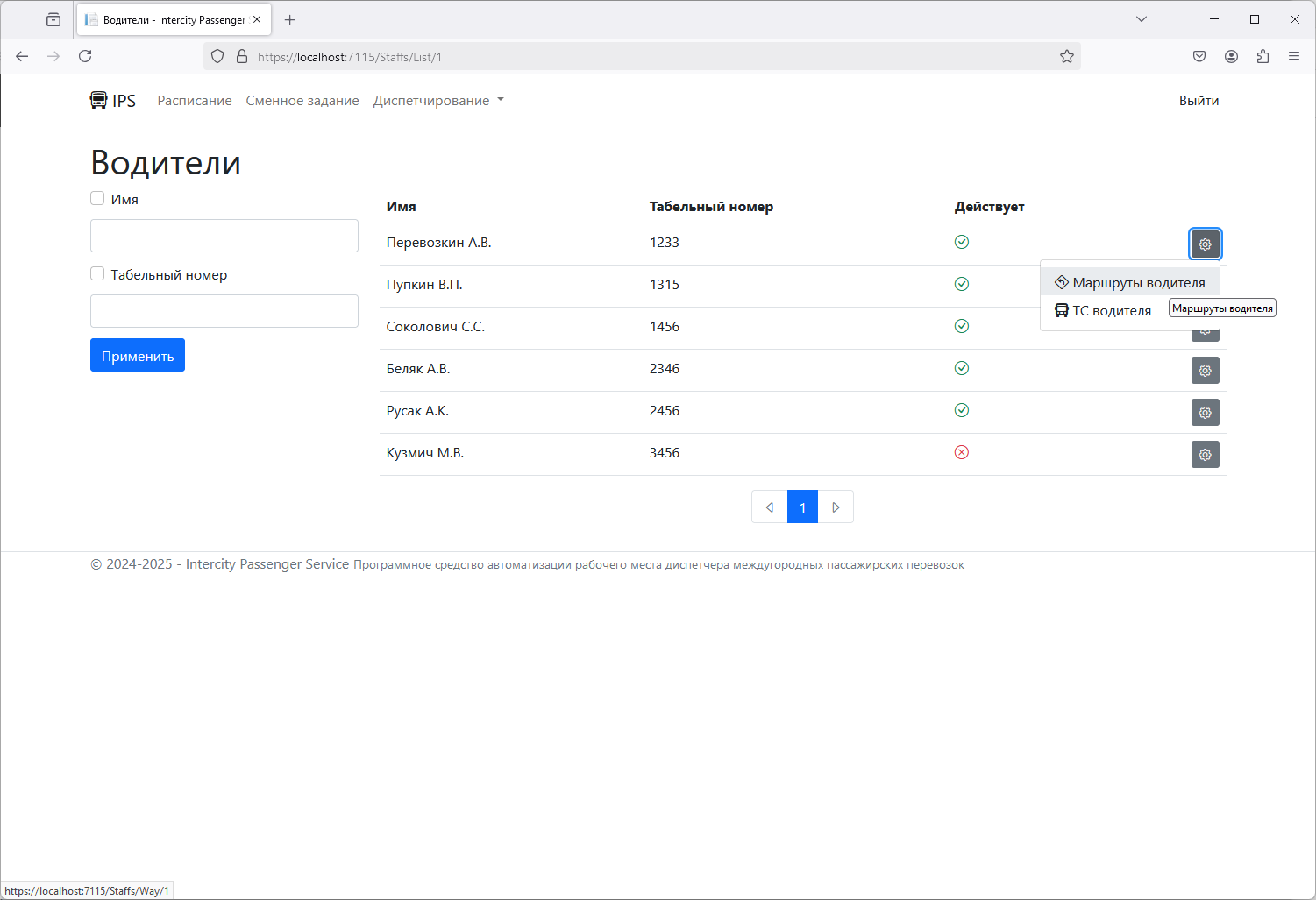


Рисунок 5.8 – Форма списка водителей

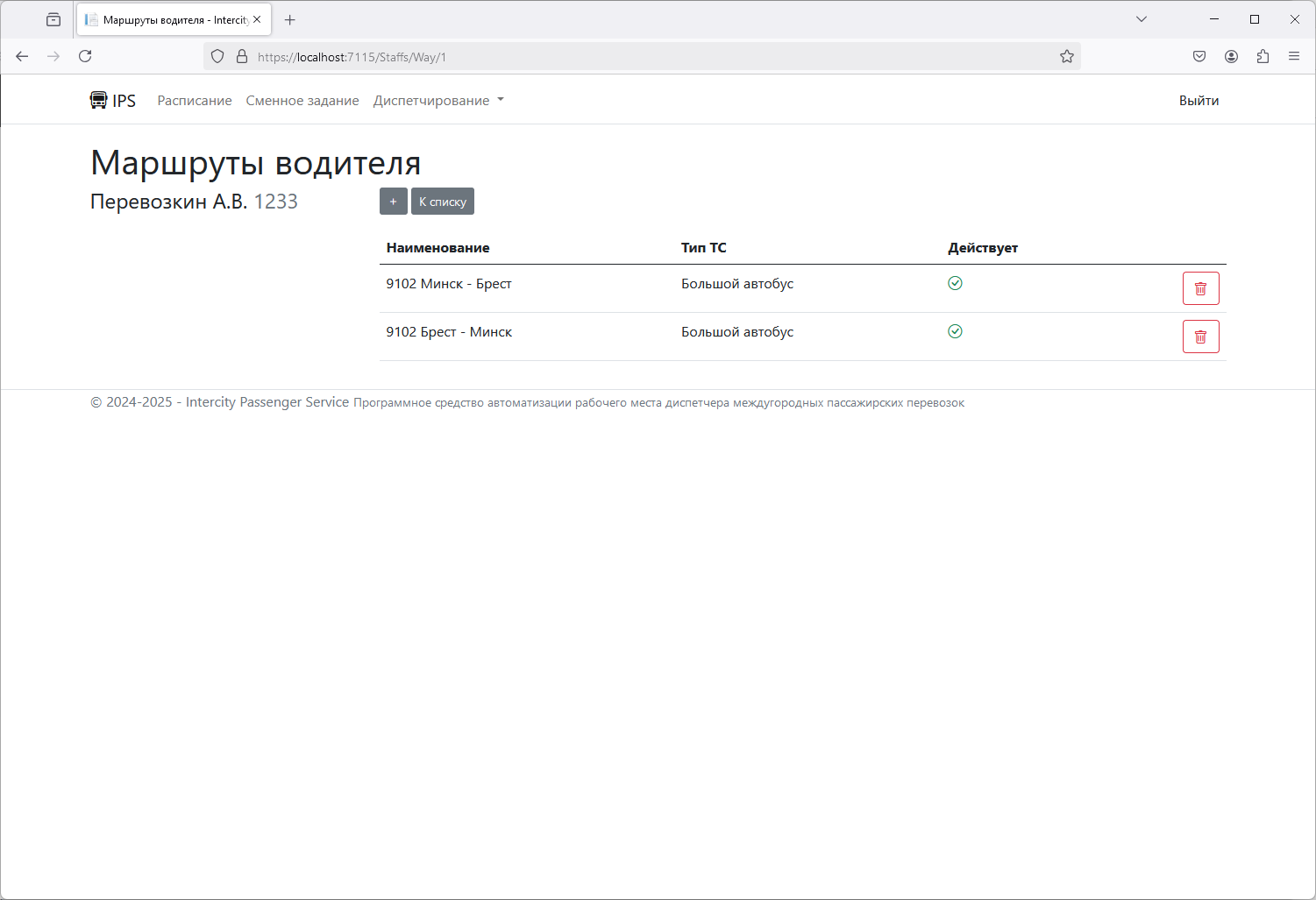


Рисунок 5.9 – Форма списка маршрутов водителя

Для добавления маршрута водителя необходимо в списке маршрутов водителя нажать кнопку «Добавить». В результате открывается форма добавления маршрута водителя (см. рисунок 5.10), в которой необходимо выбрать маршрут и нажать кнопку «Сохранить». Для отмены операции следует нажать кнопку «Отмена», система вернет пользователя к странице списка.

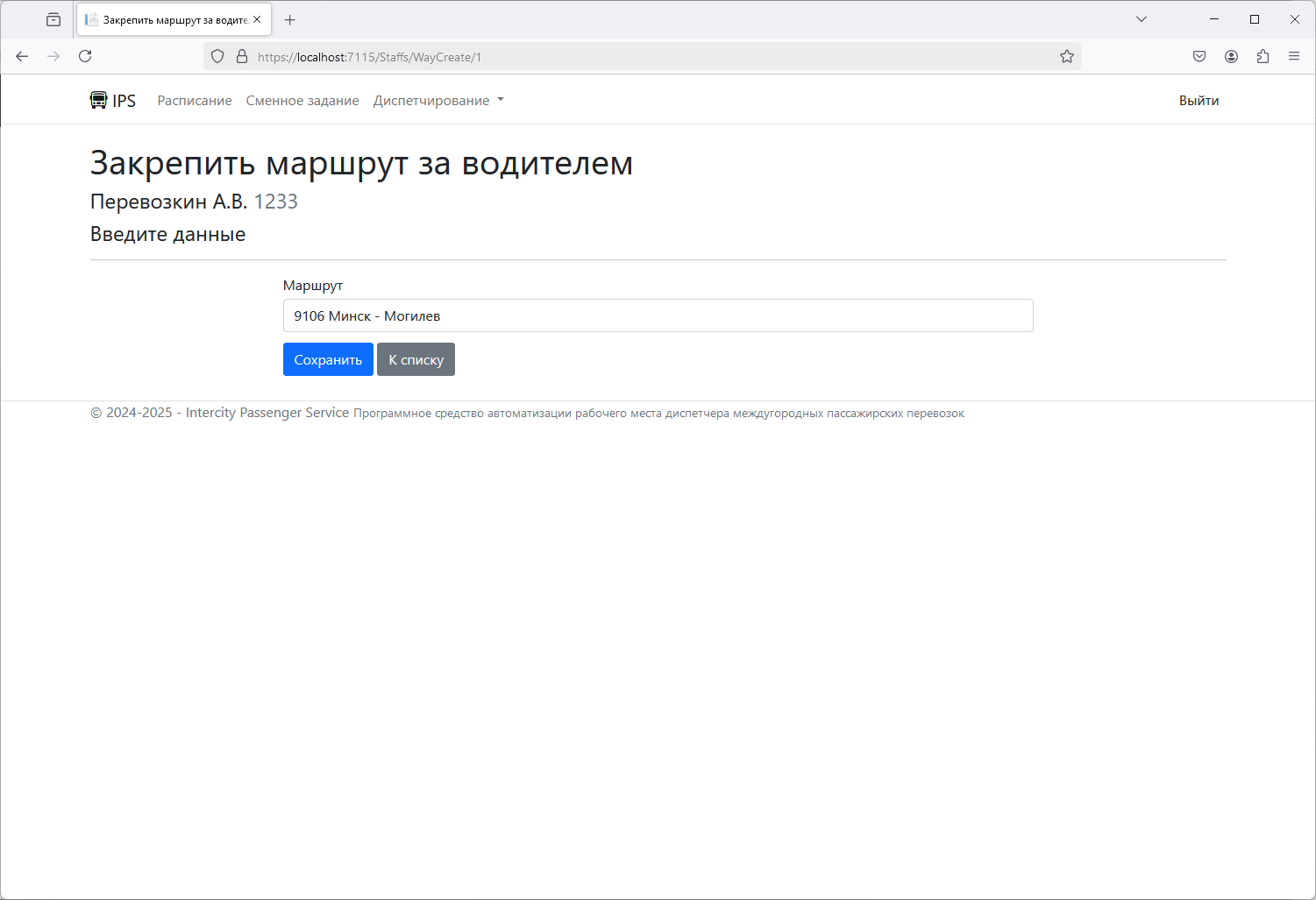


Рисунок 5.10 – Форма добавления маршрута водителя

Для удаления маршрута водителя необходимо найти в списке маршрут водителя и нажать кнопку «Удалить». В результате открывается форма удаления маршрута водителя (см. рисунок 5.11), в которой необходимо нажать кнопку «Удалить». Для отмены операции следует нажать кнопку «Отмена», система вернет пользователя к странице списка.

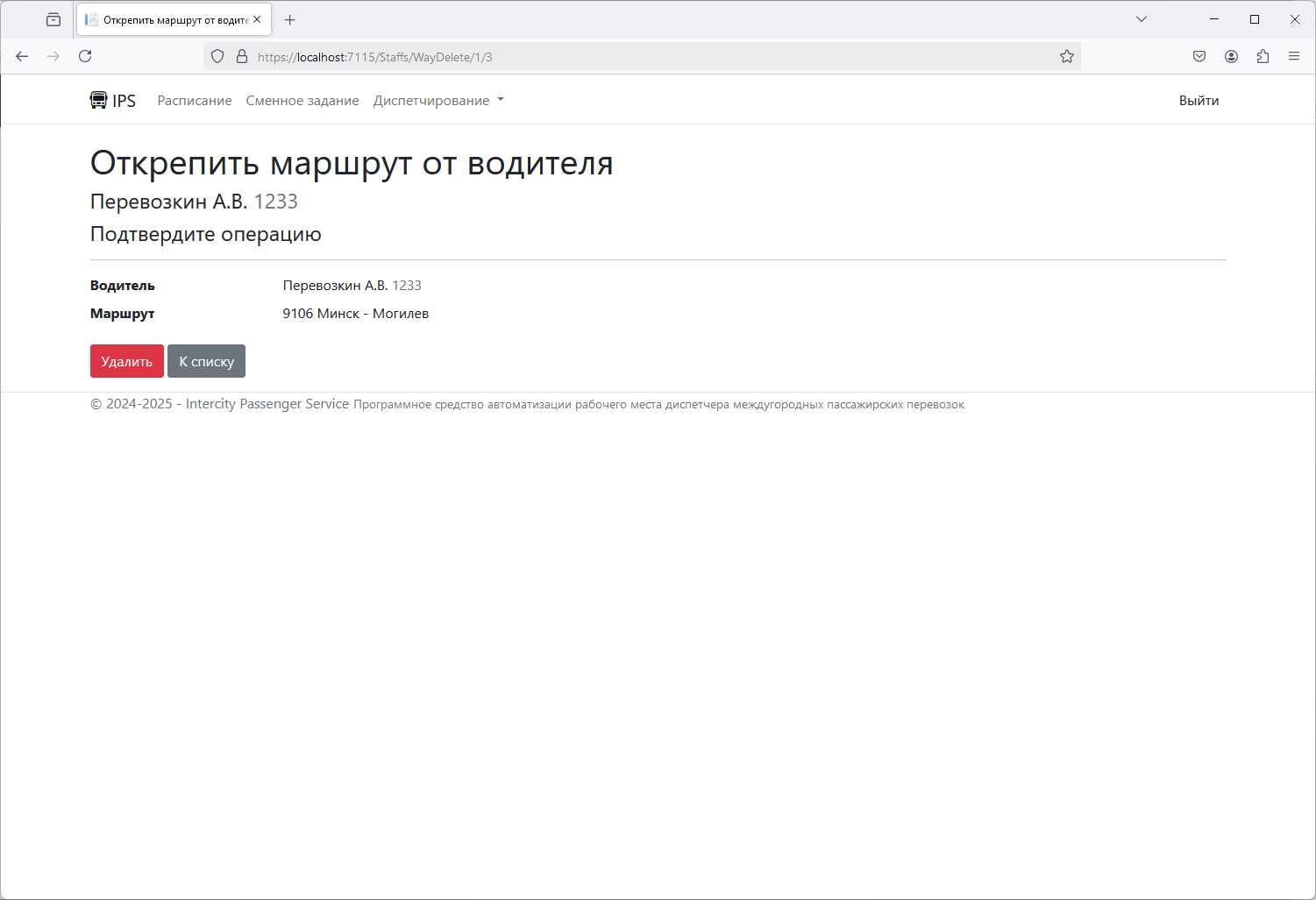


Рисунок 5.11 – Форма удаления маршрута водителя

Прочие операции авторизованного пользователя выполняются аналогично рассмотренным.

**Выводы к разделу 5**. Разработано руководство по развертыванию и использованию программного средства.

Заключение

В результате проведенного исследования получены следующие результаты:

1. Существующие программные средства автоматизации управления междугородними пассажирскими перевозками, представленные на рынке, не адаптированы к фактическим потребностям автотранспортного предприятия, не удовлетворяют требуемым критериям и потребуют существенных затрат на настройку и модернизацию программного средства в соответствии с нуждами автотранспортного предприятия. Следовательно, актуальной является задача разработки нового программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок, удовлетворяющего требуемым критериям.
2. Исследование текущего состояния бизнес-процесса диспетчеризации междугородних пассажирских перевозок на автотранспортном предприятии позволило выявить ряд недостатков при его организации: процесс автоматизирован частично, информация о расписании и информация о сменных нарядах формируется по запросу и требует непосредственного участия диспетчера в формировании ответа, формирование носит неоперативный характер. Определены особенности функционирования бизнес-процесса после внедрения программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок, разработана диаграмма вариантов использования и их описание, определены функциональные и нефункциональные требования к программному средству, выполнена постановка задачи.
3. Выполнено проектирование и разработка программного средства автоматизации рабочего места диспетчера междугородних пассажирских перевозок: спроектирована архитектура системы, выбраны компоненты и технологии для реализации, определен перечень форм программного средства и спроектированы макеты пользовательского интерфейса, разработаны схемы логической и физической моделей базы данных, реализована модель данных, разработаны диаграммы классов, разработаны диаграмма деятельности, диаграмма последовательности, диаграмма состояний иллюстрирующие типовое взаимодействие между объектами программного средства, разработаны алгоритмы, реализующие бизнес-логику программного средства, и программный код. Определены механизмы обеспечения информационной безопасности.

В результате разработано программное средство, реализующее следующие функции:

* закрепление маршрутов и транспортных средств за водителями;
* регистрация междугородних пассажирских рейсов;
* регистрация нарядов на рейсы;
* формирование расписания на указанную дату;
* формирование сменного задания водителя на указанный период.

Программное средство реализовано в виде веб-приложения на базе платформы *ASP.NET Core*. Для хранения данных используется СУБД *Microsoft SQL Server*.

1. Проведено тестирование программного средства. Все тест-кейсы программного средства пройдены успешно. Полученные в ходе тестирования результаты соответствуют ожидаемым. Все функциональные требования к программному средству выполняются. Следовательно, разработанное программное средство удовлетворяет требованиям задания на разработку.
2. Разработано руководство по развертыванию и использованию программного средства.

Программное средство в отличие от аналогов позволяет регистрировать междугородние пассажирские рейсы на дату, водителей и транспортные средства на эти рейсы (наряды), а также получать в оперативном режиме пассажирам – расписание рейсов, водителям – сменное задание.

Практическая значимость программного средства заключается в возможности его использования для диспетчеризации междугородних пассажирских перевозок.