Министрерсто образования и молодежной политики Свердловской области государственное автономное профессиональное образовательное учреждение Свердловской области «Алапаевский многопрофильный техникум»

ДИПЛОМНЫЙ ПРОЕКТ

**Разработка автоматизированной системы «Бронирование билетов»**

|  |  |
| --- | --- |
| Нормоконтролер  \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) (И.О.Фамилия) | Исполнитель: Жуков Данил Алексеевич  Студент 403 ИСП группы  Очной формы обучения  Специальность 09.02.07 Информационные системы и программирование  Руководитель: Загайнов Денис Сергеевич |

Допускается к защите «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Зам. директора по УПР \_\_\_\_\_\_\_\_\_\_\_\_ Е.В. Попова

Алапаевск

2024

Оглавление

**ВВЕДЕНИЕ**

В век информационных технологий, бронирование билетов в сети Интернет имеет большой спрос. Все больше и больше люди стараются перейти на электронные билеты, за счет их удобности, эргономичности.

Электронный билет — это электронный документ, удостоверяющий договор перевозки между пассажиром и компанией, осуществляющей перевозку пассажиров. В отличие от бумажного билета, электронный билет представляет собой цифровую запись в базе данных компании.

От привычных бумажных билетов, электронные имеют ряд преимуществ:

1. Удобность. Электронный билет можно скачать и хранить в памяти телефона, хранить в облаке или распечатать, для привычного использования
2. Гибкость. Электронный билет можно забронировать на определенную дату, место. За счет этого клиенту проще сориентироваться на подходящую для него дату, подстроить свои планы.
3. Время. На оформление электронного билета потребуется всего пара минут, что зачастую очень положительно сказывается на лояльности клиентов.
4. Способ оплаты. Оплатить электронный билет можно практически любой платежной системой.

Объектом исследования данной темы является автоматизированная система «Бронирование билетов»

Предмет исследования – автоматизированная система, которая будет использоваться для бронирования билетов

Целью данного дипломного проекта стало, разработать автоматизированную систему, с помощью которой будет осуществляться процесс бронирования билетов

Задачи:

1. Провести анализ предметной области
2. Определить требования к функциональным возможностям автоматизированной системы
3. Разработать программное приложение
4. Провести ряд тестирований на выявление несоответствий и устранить возникшие ошибки

**ГЛАВА 1. АНАЛИЗ ПРЕДМЕТОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ НА РАЗРАЬОТКУ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ «БРОНИРОВАНЕ БИЛЕТОВ»**

* 1. Техническое задание на разработку автоматизированной системы «Бронирование билетов»

*Общие сведения*

*Название:* «Бронирование билетов»

*Шифр:* АИС – автоматизированная информационная система, ГАПОУ СО «АМТ» - Государственное автономное профессиональное общеобразовательное учреждение Свердловской области «Алапаевский многопрофильный техникум»

*Документы, на основании которых создается АИС:* приказ о дипломном проектировании, в соответствие с учебным планом

*Заказчик:*

Организация: ГАПОУ СО «АМТ»

Адрес: 624601, Свердловская область, г. Алапаевск, ул. Ленина, д.11

*Разработчик:*

Организация: студент 403 ИСП группы, ГАПОУ СО «АМТ»

Адрес: 624601, Свердловская область, г. Алапаевск, ул. Ленина, д.11

*Сроки разработки:* с 15.01.2024 – 01.06.2024

*Финансирование:* не финансируется

*Цель и назначение автоматизированной системы:*

*Цель:* разработать систему, которая будет эксплуатироваться при прохождении демонстрационных экзаменов и в роли тренажера при подготовке

*Назначение:* автоматизированная система «Бронирование билетов», разрабатывается как приложение-тренажер, которое поможет студентам профильного направления, подготовиться к демонстрационному экзамену, а также преподавателям и мастерам подготовить и провести экзамен.

Характеристика объектов автоматизации:

*Требования к автоматизированной системе:*

*Требования к видам обеспечения:*

Перед установкой АИС «Бронирование билетов», предварительно необходимо установить (обновить) последнюю версию Microsoft SQL Server.

Также для редактирования данных, непосредственно в самой базе данных, необходимо наличие последней версии Microsoft SQL Server Management Studio

*Технические требования:*

Операционная система: Windows 10/11 x64 – разрядности

Процессор: intel Core I3 6100 3.7 ГГц / AMD Ryzen 3 1200 3.1 ГГц / или выше

Видеокарта: GeForce GT 1030 2 ГБ видеопамяти / AMD RX 550 2 ГБ видеопамяти / или выше

Монитор: разрешение экрана не менее 1920x1080 / или выше

ОЗУ: не менее 4 ГБ

Место на диске: не менее 2 гб

*Общие технические требования:*

*Численность и квалификация персонала, максимальное количество пользователей АИС:* численность и квалификация персонала – не предусмотрена или может быть установлена по усмотрению организации. Численность пользователей АИС – неограниченна, но при большом количестве одновременной эксплуатации могут возникнуть некоторые задержки в отзывчивости приложения

Требования к надежности:

Требования по безопасности АИС: безопасность АИС подразумевает создание резервной копии, каждый день, после эксплуатации АИС.

Требования к транспортабельности: при транспортировке АИС необходимо провести перенос базы данных и программной оболочки на внешний носитель (Flash USB, Внешний HDD и т.д.), перед этим провести резервное копирование АИС.

*Состав и содержание работ по созданию автоматизированной системы:*

Таблица 1. Этапы работ над проектом

|  |  |  |
| --- | --- | --- |
| Наименование работы | Сроки выполнения | Выполняющее лицо |
| Установка требований к АИС | 3 дня | Жуков Данил Алексеевич |
| Разработка технического задания | 3 дня | Жуков Данил Алексеевич |
| Согласование технического задания | 3 день | Жуков Данил Алексеевич |
| Разработка базы данных | 20 дней | Жуков Данил Алексеевич |
| Разработка графического интерфейса на базу данных | 35 дней | Жуков Данил Алексеевич |
| Внедрение базы данных в разработанный графический интерфейс | 15 дней | Жуков Данил Алексеевич |
| Тестирование продукта | 2 дня | Жуков Данил Алексеевич |
| Согласование с заказчиком поправок | 1 день | Жуков Данил Алексеевич |
| Внесение поправок | 5 дней | Жуков Данил Алексеевич |
| Итоговое тестирование | 2 день | Жуков Данил Алексеевич |
| Передача АИС заказчику / приемка продукта | 1 день | Жуков Данил Алексеевич |
| Итого: | 90 дней (3 месяца) | |

*Требования к документированию:*

В ходе разработки АИС должны так же быть разработаны следующие документы:

1. Техническое задние на АИС «Бронирование билетов»
2. Руководство оператора на АИС «Бронирование билетов»
3. Руководство системного администратора на АИС «Бронирование билетов»

*Порядок контроля и приемки:*

Порядок контроля и приемки осуществляется непосредственно вместе с заказчиком, производятся все необходимые тестовые сценарии. После чего эксплуатацию производит независимая единица и выдвигает свое мнение по поводу эксплуатации АИС.

По завершению приемки, создается документ, заверяющие принятие / не принятие АИС в эксплуатацию этой организацией.

*Источники разработки:*

Межгосударственный стандарт, ГОСТ 34602 – 2020 – «Техническое задание на создание автоматизированной системы» / Москва, Российский институт стандартизации 2021

* 1. Обзор существующих решений

Анализируя сеть Интернет, на аналогичные программные решения, все они являются автоматизированной системой – сайтом, на котором предоставляется услуга бронирования билетов.

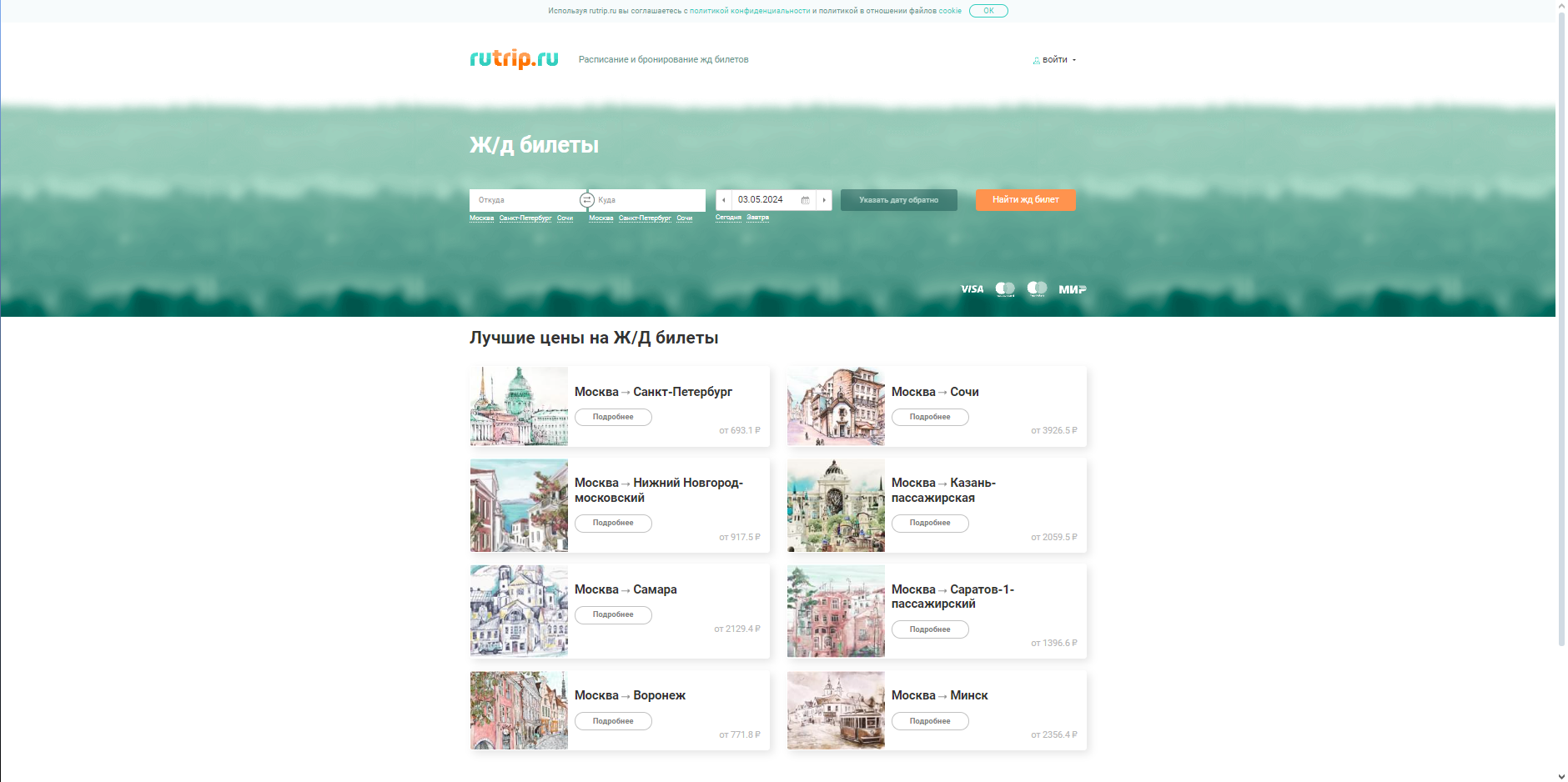
Среди большого множества приложений можно выделить некоторые программные решения:

«Расписание и бронирование жд билетов» / rutrip.ru / https:// rutrip.ru

«Яндекс.Расписания» / Yandex.Train / https://sul.su/Wk1

«РЖД» / rzd.ru / https://www.rzd.ru

*«Расписание и бронирование жд билетов» / rutrip.ru*

Рисунок 1 – Главное окно rutrip

При заходе на сайт, встречает окно, в котором необходимо указать направление сообщения, по которому необходимо купить(забронировать) билет, указать дату. Так же представлен блок лучших цен на билеты. (Рис.1)

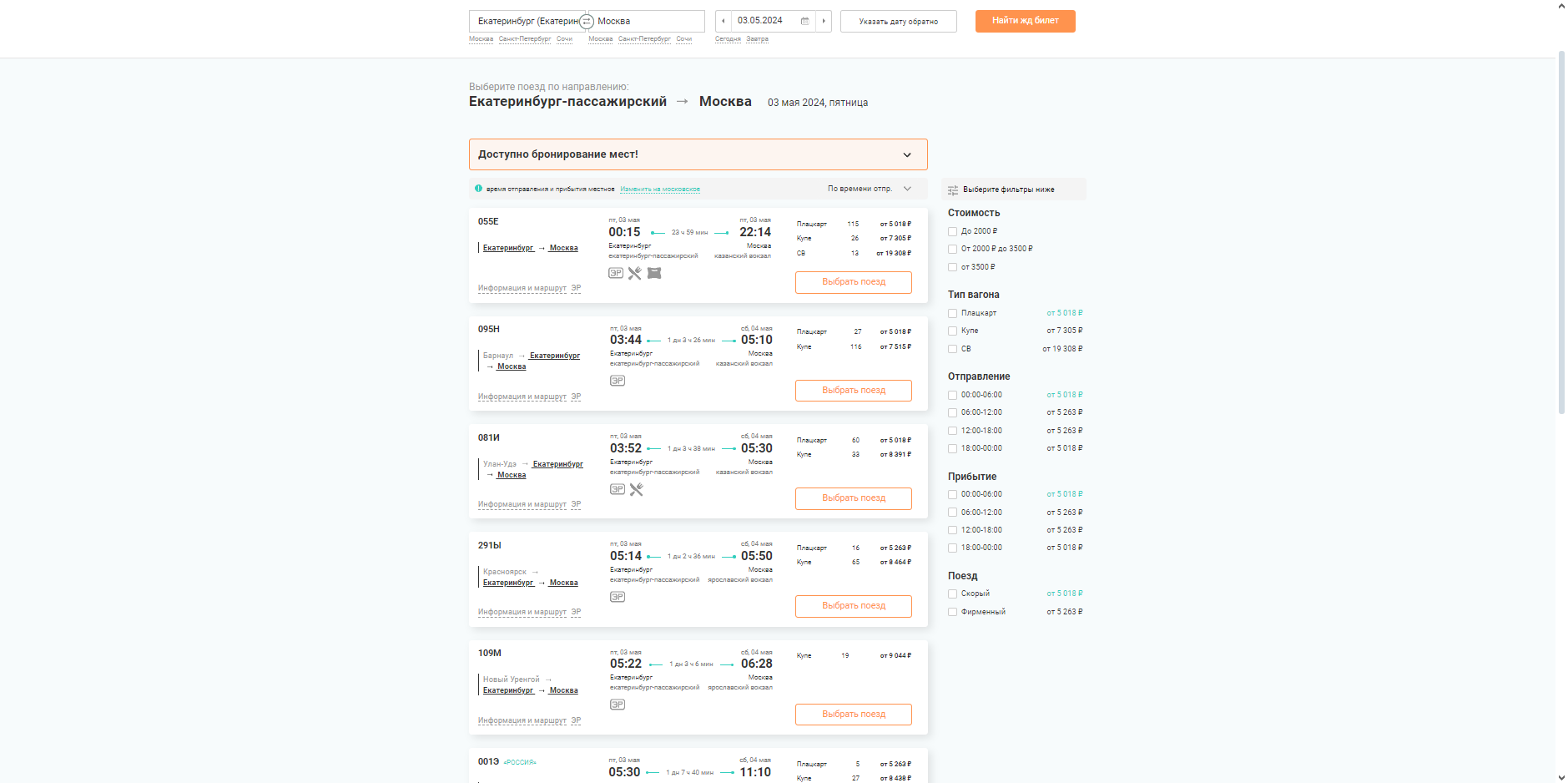
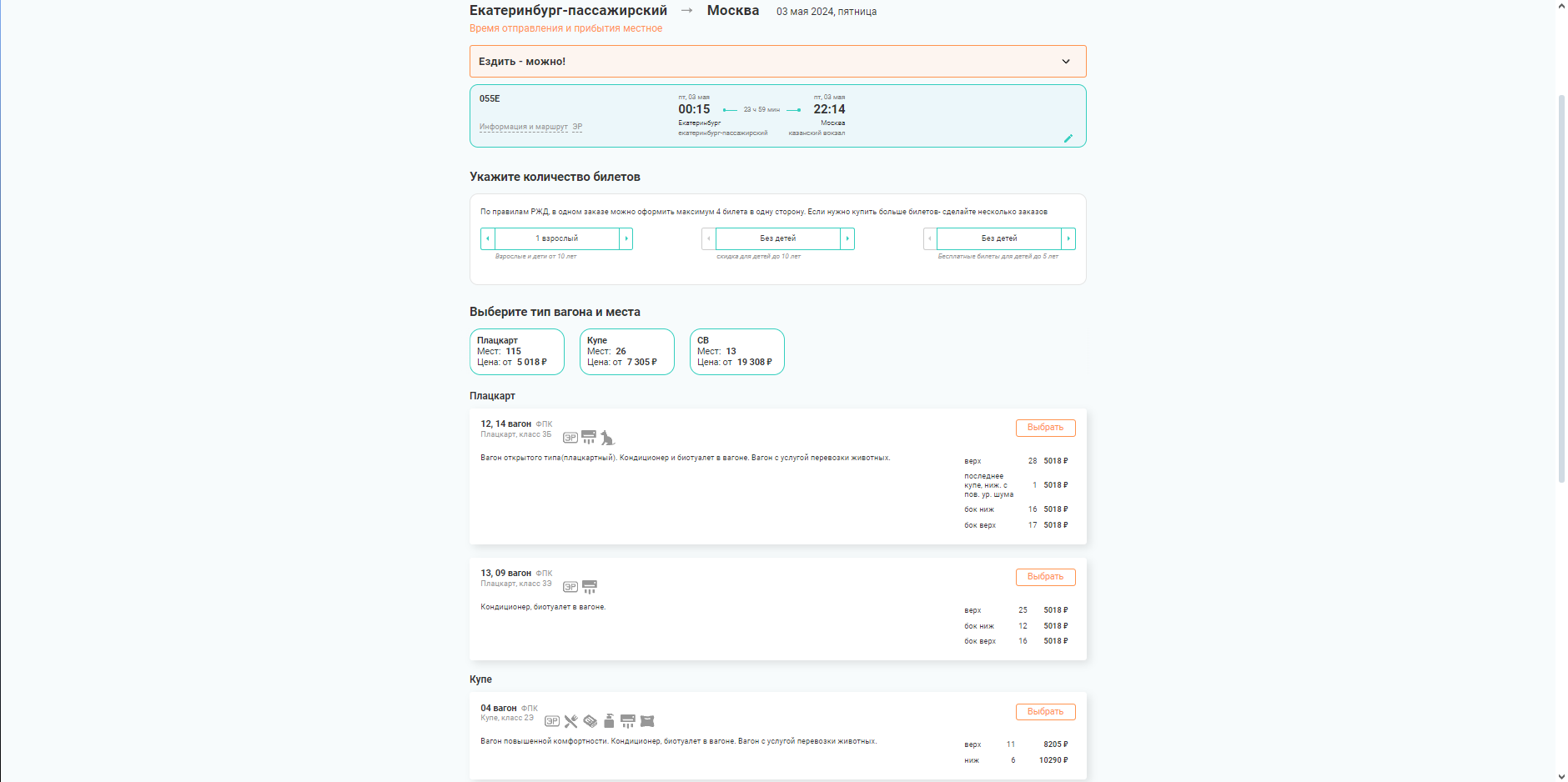
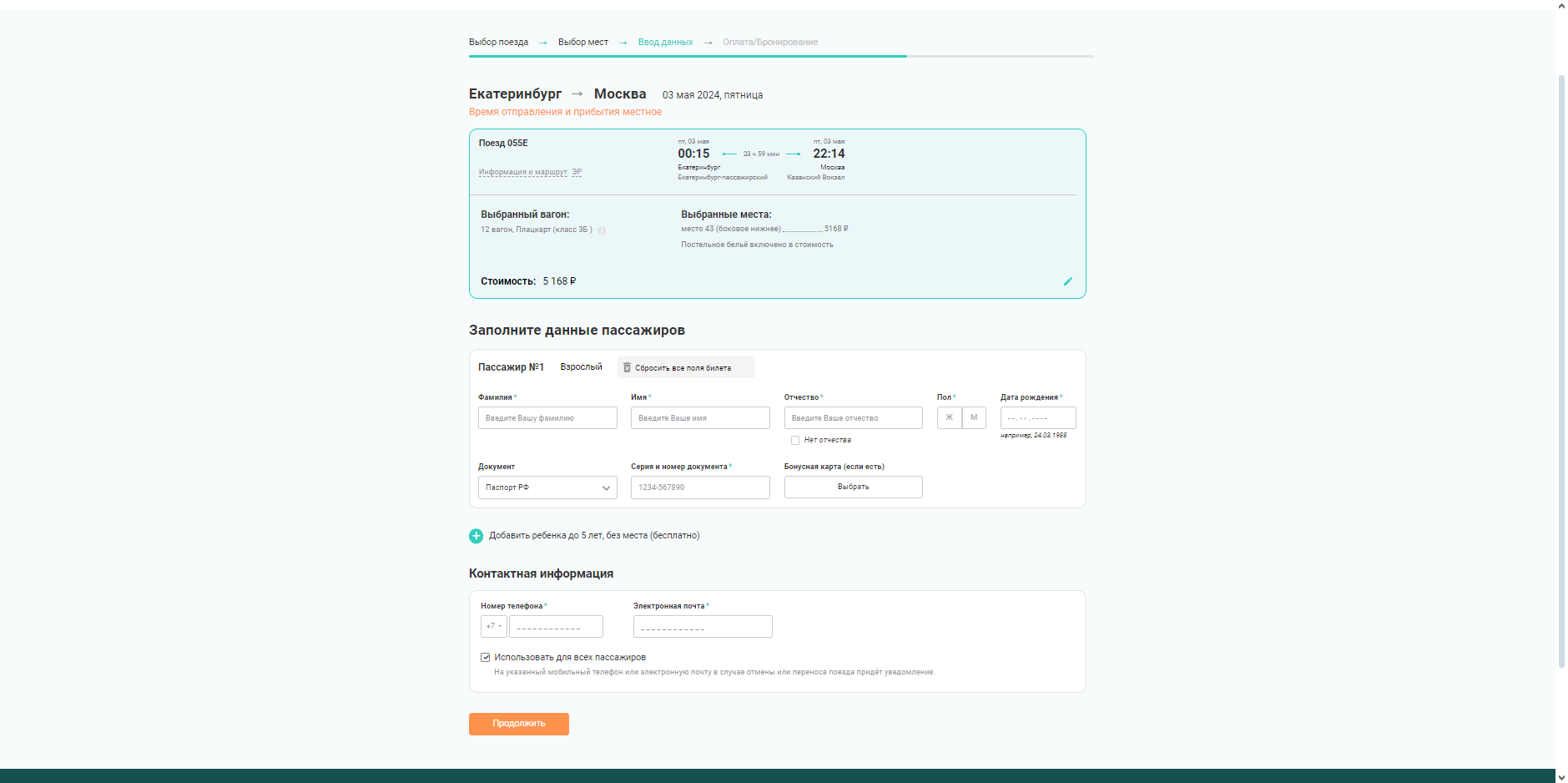


Рисунок 2 – Окно выбора сообщения

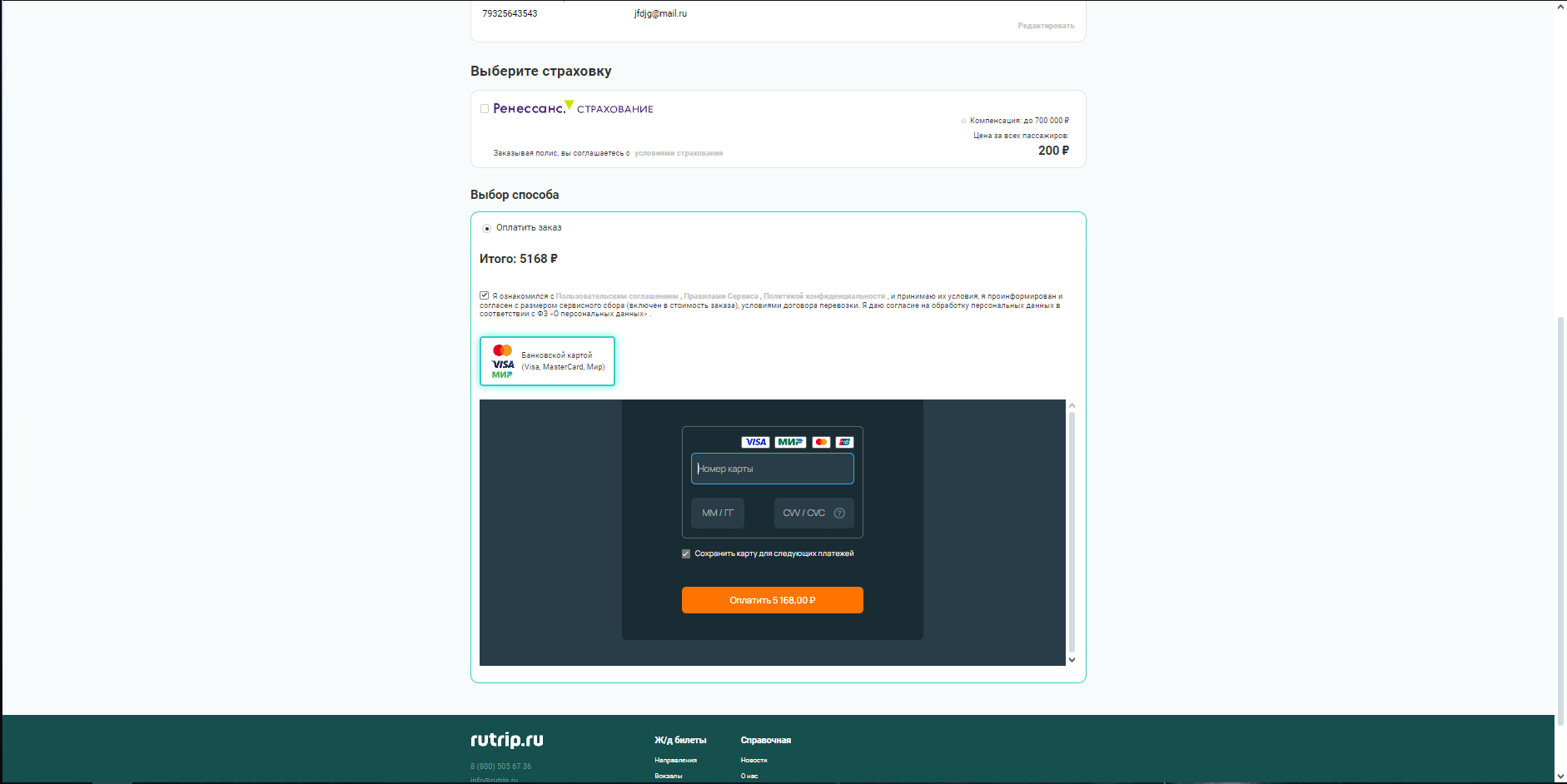
После выбора сообщения, открывается окно выбора предоставленных рейсов. На карточках сообщения предоставлена информация, которая поможет сориентироваться клиенту по необходимым ему критериям. (Рис. 2)

Рисунок 3 – Выбор вагона и места

После того, как был выбран подходящий рейс, появляется окно оформления. Необходимо указать количество билетов, тип вагона и место. (Рис. 3)

Рисунок 4 – Заполнение данных

Следующим шагом идет заполнение данных о пассажире: ФИО, дата рождения, пол, серия и номер паспорта, контактная информация. Так же дублируется информация о рейсе. (Рис.4)

Рисунок 5 – Оплата билета

После указания данных идет оплата. Оплата принимается банковскими картами нескольких платежных систем.

*«Яндекс.Расписания» / Yandex.Train*

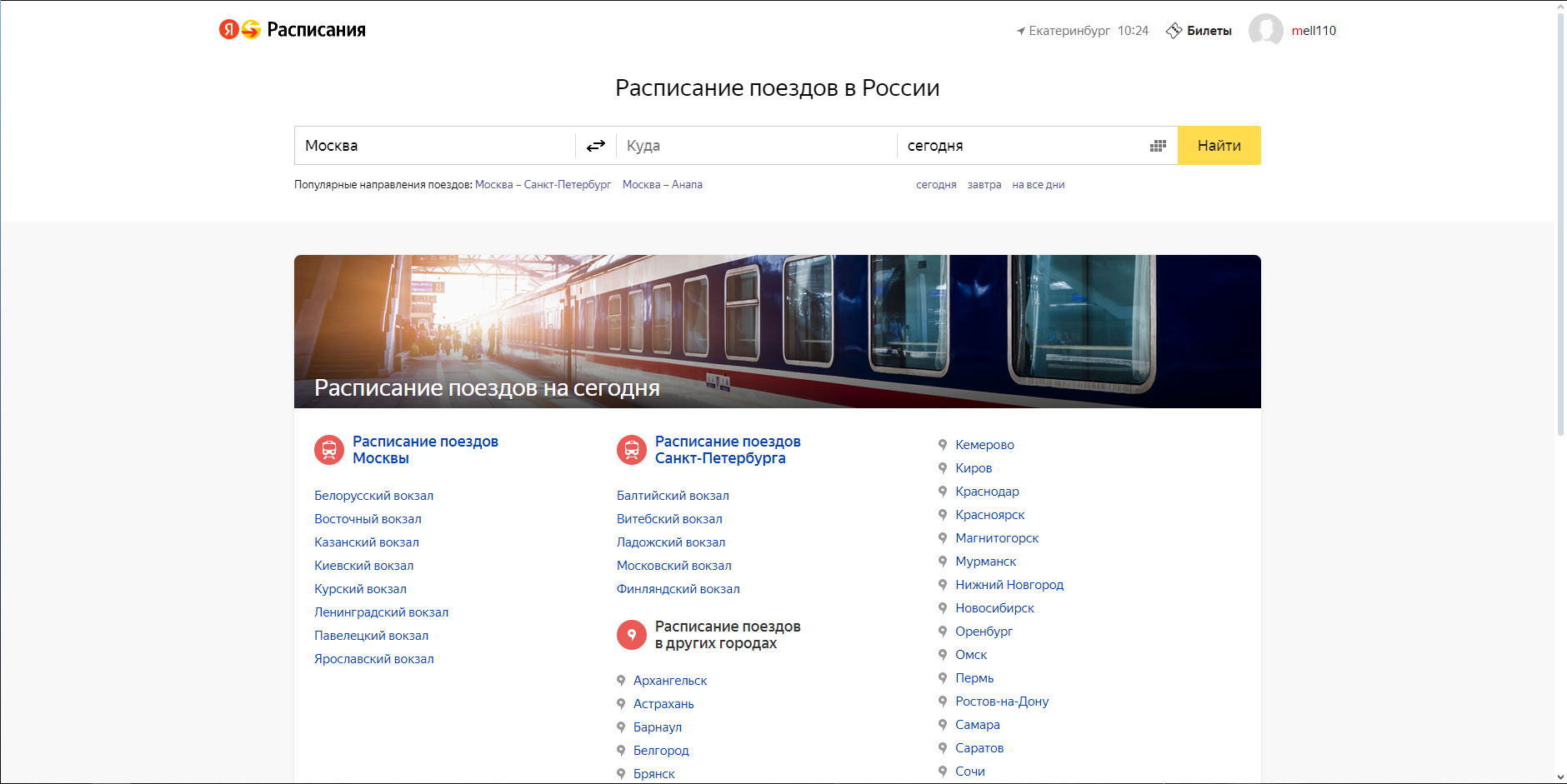
**

Рисунок 6 – Главное окно Яндекс.Расписания

Сайт предоставляет возможность оформить билет, по указанному сообщению. Необходимо указать направление сообщения, дату. (Рис. 6)

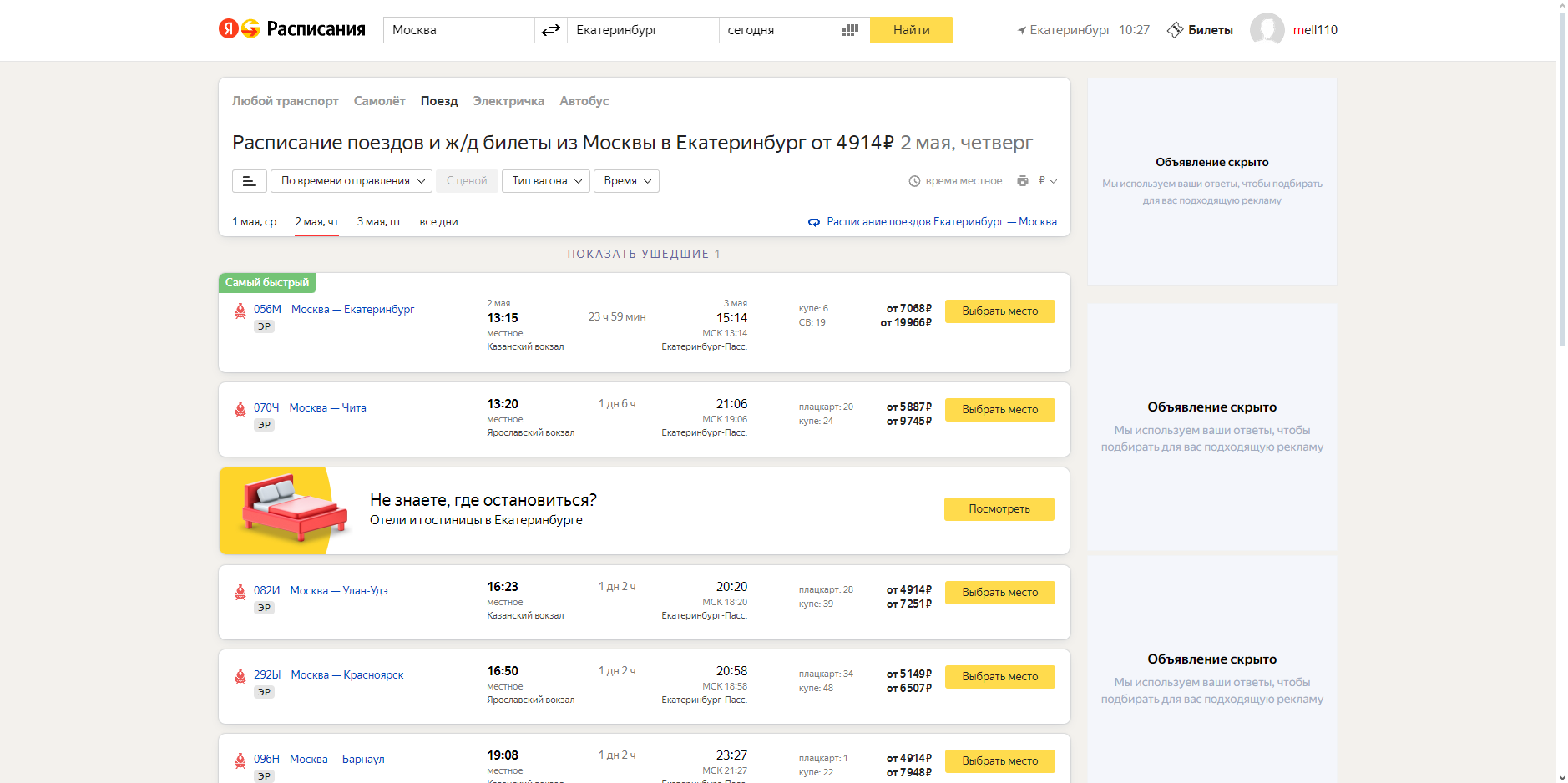
**

Рисунок 7 – Окно выбора сообщения

Выбрав сообщение, открывается окно выбора предоставленных рейсов. На карточках сообщения предоставлена информация, которая поможет сориентироваться клиенту по необходимым ему критериям (Рис. 7)

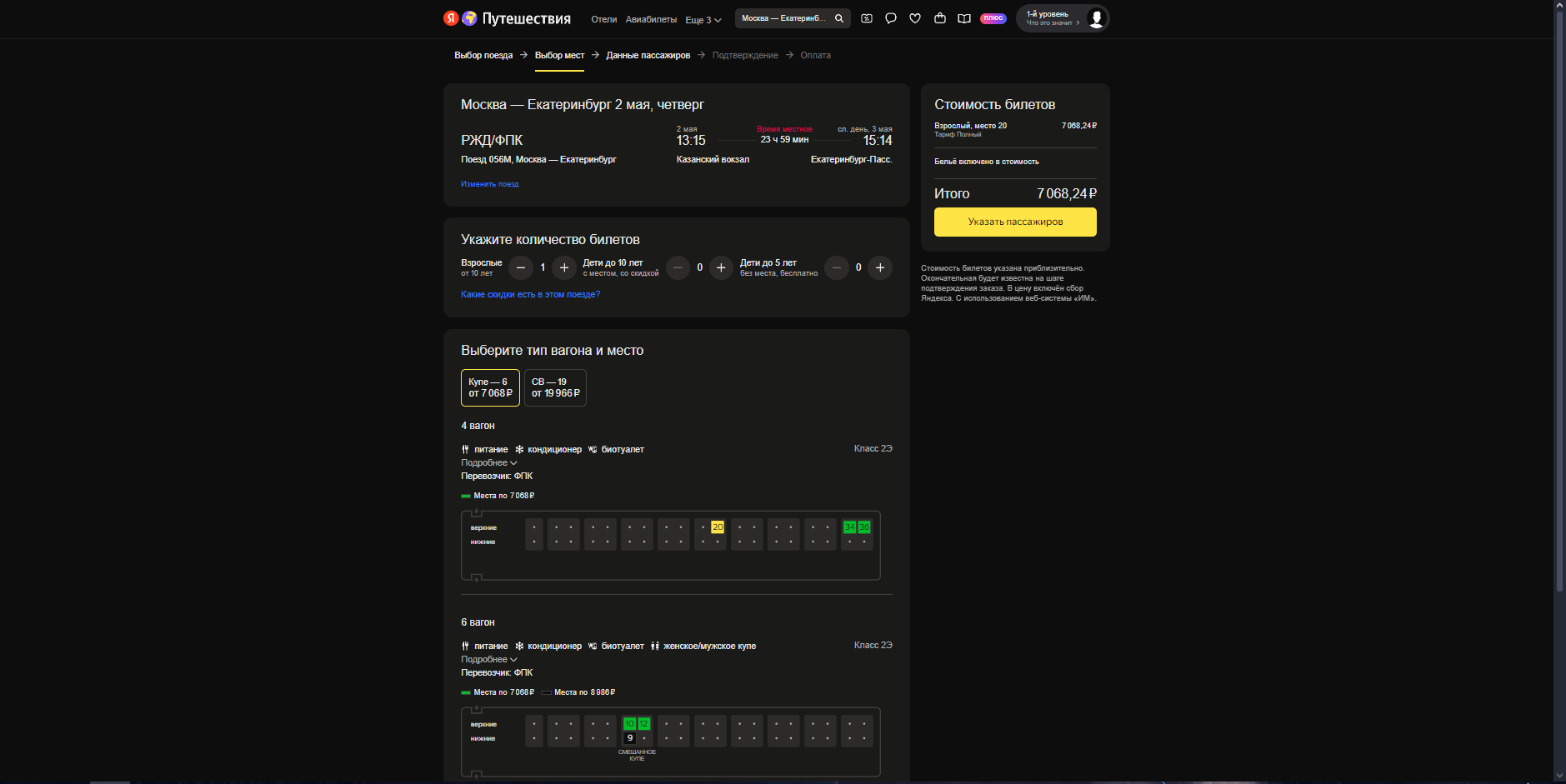
**

Рисунок 8 – Выбор вагона и места

После того, как был выбран подходящий рейс, появляется окно оформления. Необходимо указать количество билетов, тип вагона и место. (Рис. 8)

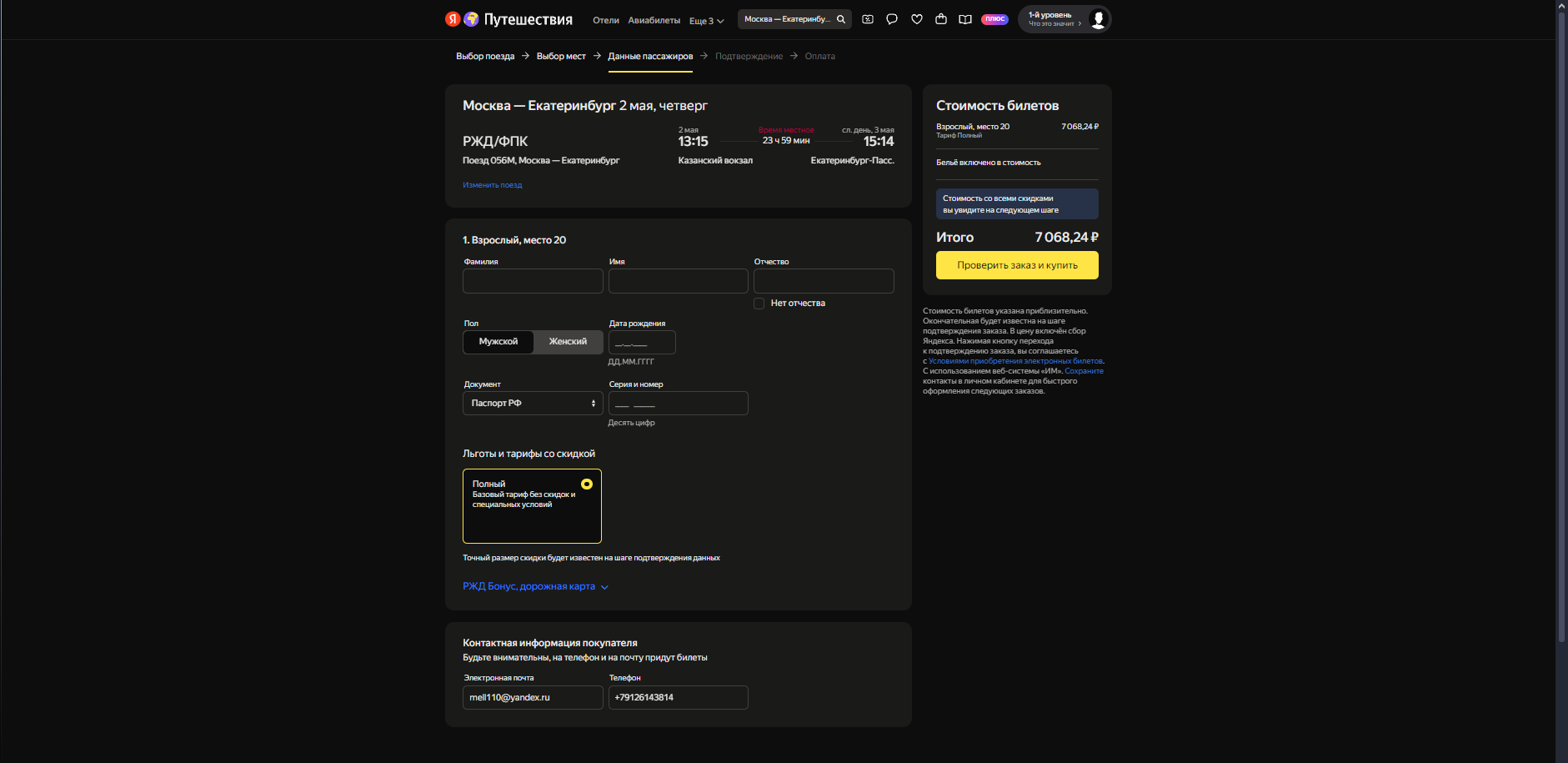


Рисунок 9 – Заполнение данных

Следующим шагом идет заполнение данных о пассажире: ФИО, дата рождения, пол, серия и номер паспорта, контактная информация. Так же дублируется информация о рейсе. (Рис. 9)

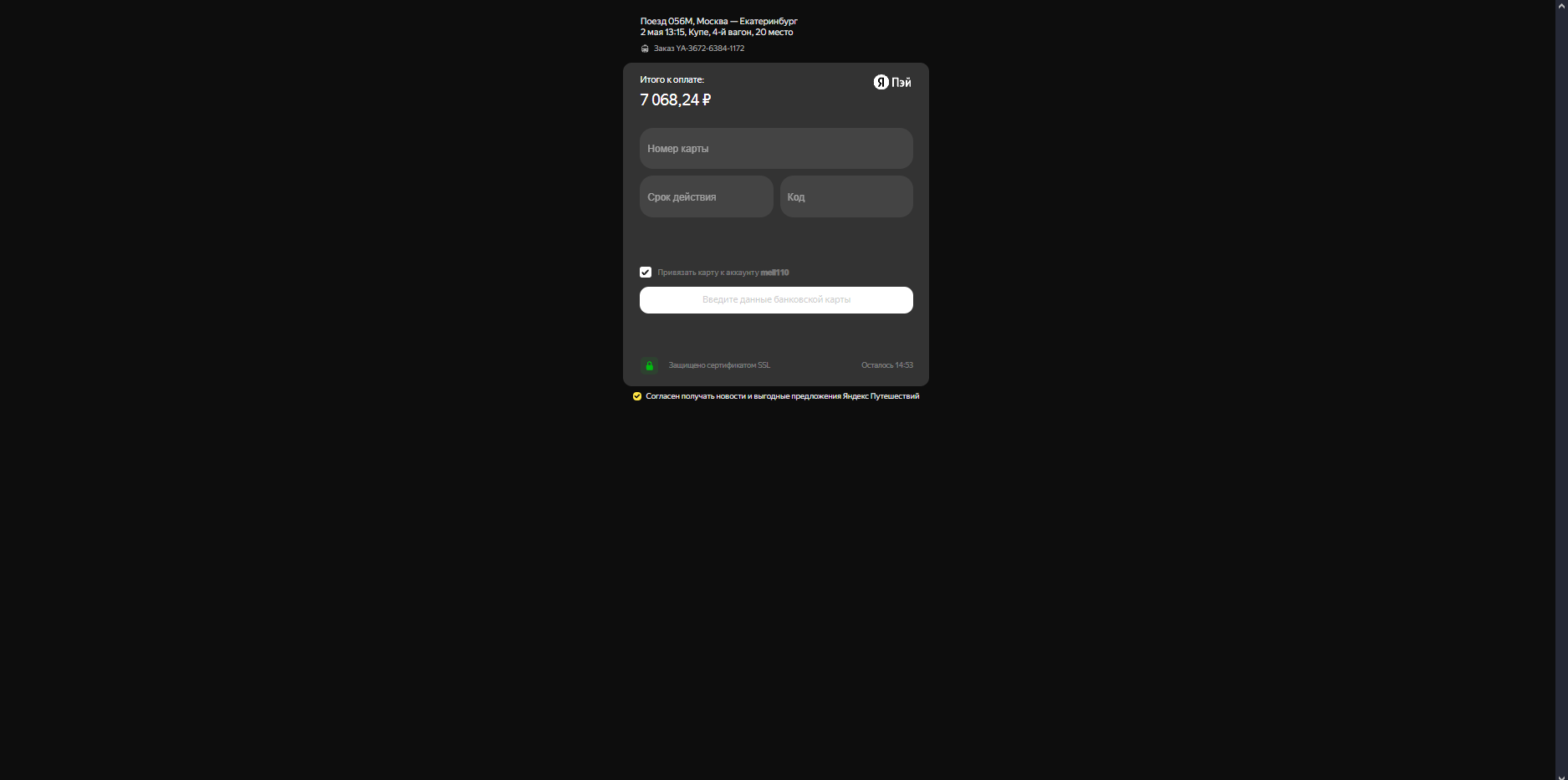


Рисунок 10 – Оплата билета

«РЖД» / rzd.ru

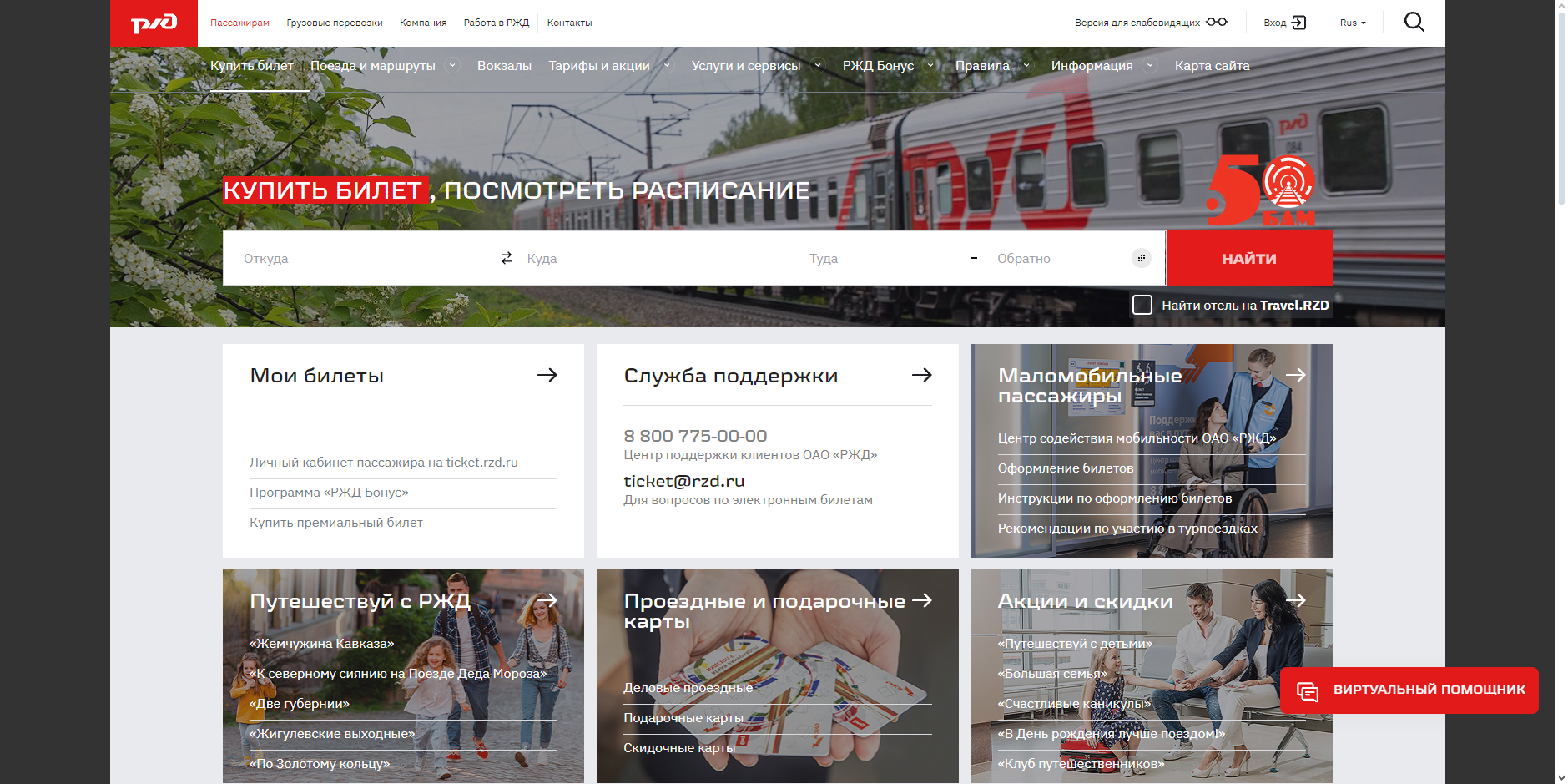


Рисунок 11 – Главное окно «РЖД»

При заходе на сайт, встречает окно, в котором необходимо указать направление сообщения, по которому необходимо купить(забронировать) билет, указать дату. (Рис. 11)

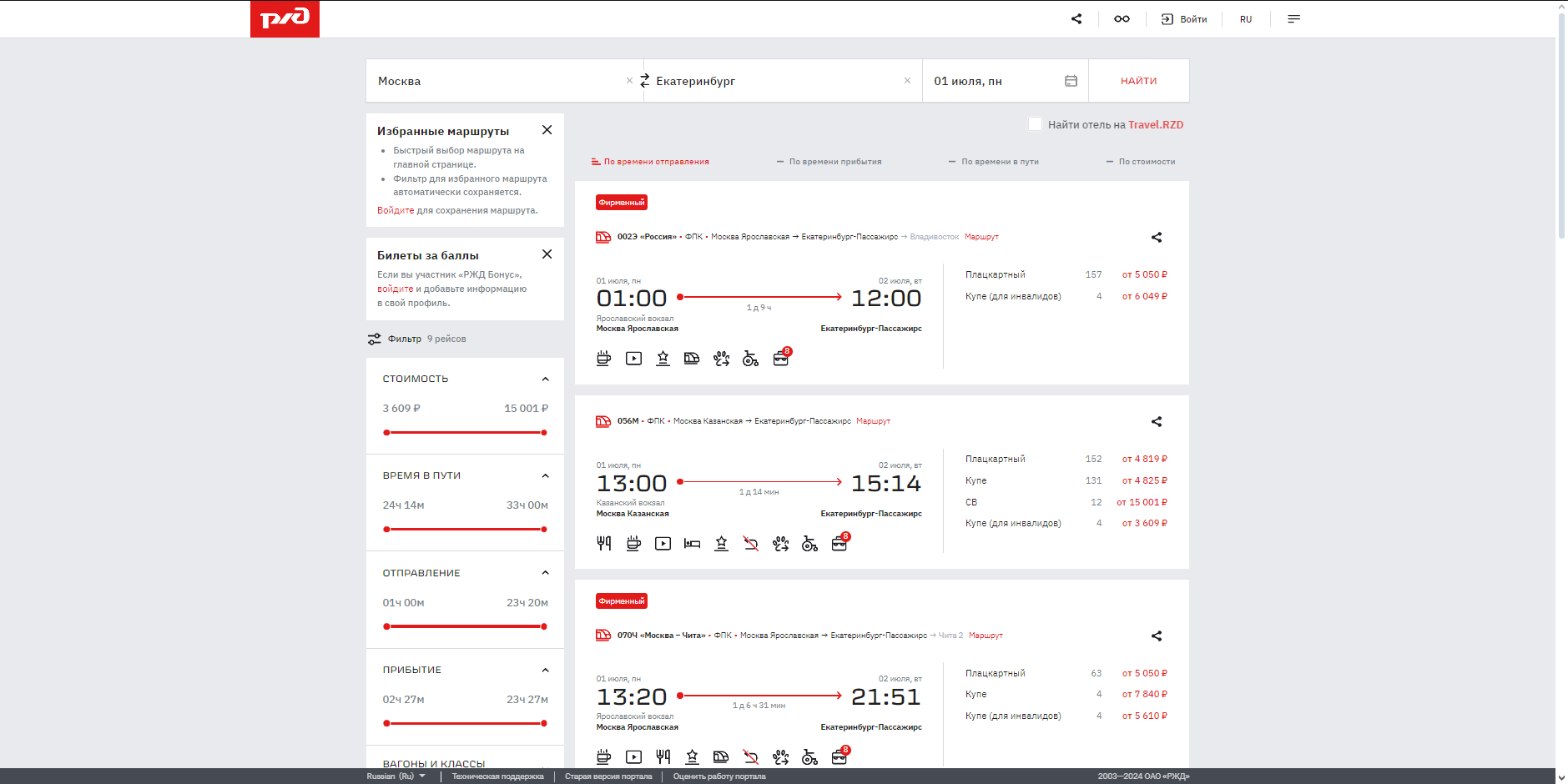


Рисунок 12 – Окно выбора сообщения

После выбора сообщения, открывается окно выбора предоставленных рейсов. На карточках сообщения предоставлена информация, которая поможет сориентироваться клиенту по необходимым ему критериям. (Рис. 12)



Рисунок 13 – Выбор вагона и места

После того, как был выбран подходящий рейс, появляется окно оформления. Необходимо вагон и место. (Рис. 13)

Подводя итог рассмотренных аналогичных решений, можно сделать вывод, что они имеют почти одинаковую структуру. У всех приятный интерфейс, удобен для использования, имеются версии для слабовидящих.

* 1. Обзор и обоснование выбора программного обеспечения для разработки автоматизированной системы «Бронирование билетов»

Перед началом разработки автоматизированной системы «Бронирование билетов» было необходимо проанализировать доступные программные обеспечения.

При анализе сети Интернет, были выделены следующие языки программирования, с помощью которых можно осуществить программный интерфейс: C#, C++, Python.

*Язык программирования C#:*

C# — современный объектно-ориентированный и типизированный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET.

C# применяется для разработки самых разнообразных программ, включая desktop- и веб-приложения, мобильные приложения, игры на базе Unity, облачные приложения на Microsoft Azure, приложения для работы с базами данных, интеграции с другими системами и создания инструментов разработки. Это очень гибкий язык программирования, который используется на разных платформах. Рассмотрим самые популярные сферы его применения.

*К плюсам языка относятся:*

1. Объектно-ориентированность. Полностью ориентирован на объекты, что позволяет программистам разрабатывать модульный и масштабируемый код.
2. Совместимость. Язык программирования разработан нацелен на платформу .NET, что обеспечивает высокую совместимость с Windows-приложениями.
3. Огромная стандартная библиотека. Поставляется с обширной стандартной библиотекой классов (FCL). Она включает в себя множество инструментов для работы с файлами, сетью, базами данных и т. д.
4. Кроссплатформенность. Начиная с версии C# 5.0 и появлением .NET Core, приложения на C# могут работать на разных платформах, включая Windows, Linux и macOS.
5. Интеграция с Visual Studio. Microsoft предоставляет среду разработки Visual Studio для работы с C#, которая облегчает создание, отладку и тестирование приложений.

*К минусам языка относятся:*

1. Невысокая скорость. При открытии программы на C# код сначала адаптируется под конкретное аппаратное обеспечение, а уже потом исполняется. Таким образом, скорость загрузки становится значительно ниже. При первом запуске интерфейсы на базе С# могут даже подтормаживать.
2. Слабое взаимодействие с железом. С# является высокоуровневым языком, поэтому он редко используется для разработки, предполагающей полное взаимодействие с железом (игровые движки, операционные системы, авиационный софт и т.д.).

*Язык программирование C++*

C++ - широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также компьютерных игр.

*К плюсам языка относятся:*

1. Поддерживаются различные стили и технологии программирования, включая традиционное директивное программирование, ООП, обобщённое программирование, метапрограммирование (шаблоны, макросы).
2. Кроссплатформенность: стандарт языка накладывает минимальные требования на ЭВМ для запуска скомпилированных программ. Для определения реальных свойств системы выполнения в стандартной библиотеке присутствуют соответствующие возможности.
3. Эффективность. Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы.

*К недостаткам языка относятся:*

1. Одна из основных проблем в C ++ заключается в том, что, когда информация указывает на эквивалентную вещь из двух разных отправных точек, это вызывает серьезную проблему, программа на C ++ по-прежнему будет иметь проблемы с кодированием.
2. Программа на C ++ сложна в течение очень большой высокоуровневой программы C ++ обычно используется для приложений, зависящих от платформы, для реальной ОС или платформы обычно выбирается набор библиотек.
3. C ++ - сложный и мощный язык, изучение и освоение которого может потребовать времени, особенно новичкам в программировании.

*Язык программирования Python:*

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

*К плюсам языка относятся:*

1. Низкий порог вхождения. Синтаксис Python схож с английским языком, который стал международным стандартом общения для разработчиков по всему миру. Это упрощает взаимодействие со сложными системами, а также даёт чёткое представление о взаимосвязи всех элементов кода между собой. Изучение Python может стать базой для «быстрого старта» для большинства начинающих программистов.
2. Лёгкость чтения. Python невероятно легко читать, поэтому у программистов обычно не возникает проблем с пониманием кода, написанного их коллегами. Это делает общение между разработчиками в рамках одного проекта намного более эффективным. А наличие большое числа IDE для разработки веб-приложений на Python делает совместную работу еще проще.
3. Хорошая визуализация. Представление данных в интуитивно понятном формате в Python достигается с помощью различных графиков и диаграмм. Компании, занимающиеся веб-разработкой, используют библиотеки Python с возможностью визуализации данных (например, Matplotlib), чтобы создавать чёткие и простые для понимания неспециалистов отчеты.
4. Одним из главных преимуществ языка Python является наличие у него большого числа фреймворков, упрощающих процесс разработки. Большинство фреймворков Python имеют четкую специализацию, в зависимости от типа и масштаба выполняемых с их помощью задач.

*К недостаткам языка относятся:*

1. Многопроцессорность — важная часть написания приложения. Python поддерживает многопроцессорность, но из-за отсутствия прямой поддержки многопоточности (задачи выполняются параллельно в один поток), он может быть не таким гибким или удобным, как другие языки.
2. Python — это язык, известный гибкостью подходов к типизации данных. Эта же динамическая типизация приводит к повышенному потреблению памяти. Поэтому Python будет неидеальным выбором для задач, интенсивно использующих память.
3. Python не требует, чтобы программисты определяли тип переменной, поскольку этот язык использует динамическую типизацию, которая упрощает и ускоряет написание кода. Это может привести к критическим ошибкам и дефектам, поскольку типы переменных не определены явно. Чтобы устранить эту проблему, разработчики должны запускать дополнительные тесты для выявления и исправления ошибок во время выполнения.

Для реализации базы данных, в которой будут храниться данные для взаимодействия с приложением, были выделены следующие СУБД: SQlite, MySql, Sql Server Management Studio (SSMS).

*База данных SQlite:*

SQLite — это быстрая и легкая встраиваемая однофайловая СУБД на языке C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы.

К плюсам СУБД относятся

*Преимущества SQlite:*

1. Высокая скорость. Благодаря особенностям архитектуры SQLite работает быстро, особенно на чтение. Компоненты СУБД встроены в приложение и вызываются в том же процессе. Поэтому доступ к ним быстрее, чем при взаимодействии между разными процессами.
2. Хранение данных в одном файле. База данных состоит из табличных записей, связей между ними, индексов и других компонентов. В SQLite они хранятся в едином файле (database file), который находится на том же устройстве, что и программа. Чтобы при работе не возникало ошибок, файл блокируется для сторонних процессов перед записью.
3. Минимализм. Создатели SQLite пользуются принципом «минимального полного набора». Из всех возможностей SQL в ней есть наиболее нужные. Поэтому SQLite отличают малый размер, простота решений и легкость администрирования.
4. Кроссплатформенность. СУБД подходит для UNIX-подобных систем, MacOS и Windows.

*Недостатки SQlite:*

1. Ограниченная поддержка типов данных. SQLite поддерживает только четыре типа данных, которые реализованы в SQL
2. Отсутствие хранимых процедур. Так называются блоки кода на SQL, которые сохраняются в базу данных. Хранимые процедуры можно вызывать как отдельные функции, и это удобно, если нужно последовательно выполнить несколько однотипных действий. Но SQLite их не поддерживает из-за особенностей архитектуры.
3. Ограничения в применении. Отсутствие сервера — преимущество и недостаток одновременно. Без сервера возможности СУБД меньше. Например, к одной базе не смогут обращаться несколько разных устройств.

MySql — это одна из систем управления базами данных (Databases Management System) или просто СУБД. Простыми словами, программа, с помощью которой создают и контролируют БД. Она предназначена для работы с базами реляционного типа и использует для взаимодействия с ними язык стандартизированных запросов (SQL).

*Преимущества MySql:*

1. Гибкость. Система легко настраивается под индивидуальные потребности и особенности конкретного сайта.
2. Универсальность. MySQL работает на ОС семейств Windows, Linux, Unix, Solaris и других.
3. Высокая производительность. Со средними и небольшими базами MySQL работает быстрее, чем другие популярные решения.
4. Безопасность. MySQL обеспечивает высокий уровень безопасности при хранении и передаче данных.

*Недостатки MySql:*

1. Медленное развитие. Система бесплатная, поэтому обновления выходят редко.
2. Недостаток функций. В MySQL по умолчанию меньше функций и возможностей, чем в других СУБД.
3. Снижение производительности при работе с большими проектами. Система отлично работает на небольших и средних базах данных.

Sql Server Management Studio (SSMS) – это бесплатная графическая среда для управления инфраструктурой SQL Server, разработанная компанией Microsoft. С помощью SSMS Вы можете разрабатывать базы данных, выполнять инструкции T-SQL, а также администрировать Microsoft SQL Server.

Среда SQL Server Management Studio является полнофункциональным инструментом для работы с Microsoft SQL Server, который предназначен как для разработчиков, так и для администраторов SQL Server.

*Преимущества SSMS:*

1. Масштабирование системы. Взаимодействовать с ней можно как на простых ноутбуках, так и на ПК с мощным процессором, который способен обрабатывать большой объем запросов.
2. Размер страниц – до 8 Кб. Данные извлекаются быстро, а сложную информацию удобнее хранить. Система обрабатывает транзакции в интерактивном режиме, есть динамическая блокировка.
3. Автоматизация рутинных административных задач. Например, управление блокировками и памятью, редактура размеров файлов. В программе продуманы настройки, можно создавать профили пользователей.
4. Удобный поиск. Его можно осуществлять по фразам, словам, тексту либо создавать ключевые индексы.

*Недостатки SSMS:*

1. Среда реализована только под Windows
2. Инструмент достаточно тяжелый
3. Большая часть функционала не требуется рядовым разработчикам или администраторам

Среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки.

*Преимущества Visual Studio:*

1. Nuget. Система управления пакетами для платформ разработки Microsoft, в первую очередь библиотек .NET Framework. Управляется .NET Foundation. Удобная установка библиотек в любой .Net проект.
2. Git Manager. Встроенный менеджер контроля версий. Изначально работал только с Team Foundation Server. Сейчас можно подключить Team Explorer (Название менеджера) к любому репозиторию. Присутствуют все необходимые функции для работы с git без запросов.
3. Test Manager. Встроенный менеджер тестов. После создания теста можно с помощью специального окна запускать и настраивать тесты.
4. IntelliSense. Технология автодополнения Microsoft. Дописывает название функции при вводе начальных букв. Кроме прямого назначения, IntelliSense используется для доступа к документации и для устранения неоднозначности в именах переменных, функций и методов, используя рефлексию.

*Недостатки Visual Studio:*

1. Новичку будет просто невозможно самостоятельно разобраться с Visual Studio без прохождения специальных курсов и чтения литературы. Это продукт скорее для опытных разработчиков, обращающих внимание на качество редактора и функции тестирования.

В конечном итоге, выбор был остановлен на:

1. Язык программирования – C#, за счет его возможностей и более широкого набора библиотек и возможностей, а также его совместимость с Windows – приложениями.
2. База данных – Microsoft SQL Server Management Studio, удобный интерфейс, большой выбор типов данных, расширяемость
3. Среда разработки – Visual Studio, встроенный менеджер пакетов (NuGet), за счет которого просто импортируются необходимые библиотеки, интегрированная связь с GitHub, подсказывание синтаксиса, при написании кода.

Таким образом, подводя итог главы, была проанализирована предметная область, поставлены конкретные задачи, для достижения цели, были выявлены аналоги программных приложений, описан его функционал и выявлены плюсы и минусы.

Были проанализированы языки программирования, актуальные СУБД, выделены их достоинства и недостатки. Выбрана среда программирования, так же обоснована и предоставлена с плюсами и минусами в использовании. Так же выделены и изучены аналогичные решения, которые помогут при проектировании автоматизированной информационной системы.

**ГЛАВА 2. ПРОЕКТИРОВАНИЕ И СОЗДАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ «БРОНИРОВАНЕ БИЛЕТОВ»**

1. **Описание этапов проектирования и разработки автоматизированной информационной системы «Бронирование билетов»**

Электронный билет — это электронный документ, удостоверяющий договор перевозки между пассажиром и компанией, осуществляющей перевозку пассажиров. В отличие от бумажного билета, электронный билет представляет собой цифровую запись в базе данных компании.

В ходе анализа предметной области и чернового проектирования, была создана структурная схема будущего программного приложения, показывающая функциональное назначение программы.

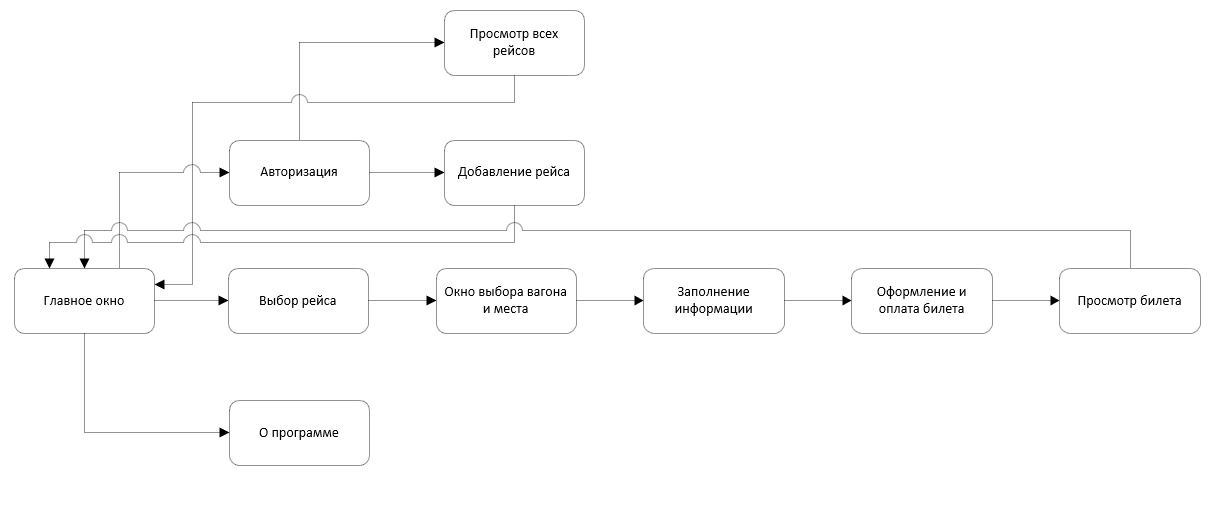


Рисунок 14 – Структурная схема приложения

Разработка интерфейса программного приложения

Интерфейс — это все, что помогает людям управлять устройствами и программами с помощью голоса, нажатий, жестов, с помощью командной строки, и даже силой мысли (последние изобретения).

Интерфейс нужен для того, чтобы дать команду программе или устройству на их «языке», получить ответ в форме текста, звука, изображения, информацию об ошибках, сбоях и т. д.

В качестве инструментов взаимодействия с системами UI могут выступать:

1. текстовые поля;
2. кнопки и галочки;
3. выпадающие списки;
4. всплывающие подсказки;
5. переключатели;
6. элементы меню программы или сайта;
7. и многое другое.

Для разработки программного интерфейса были рассмотрены наиболее актуальные виды:

1. Командная строка - способ взаимодействия между человеком и компьютером путём отправки компьютеру команд, представляющих собой последовательность символов.
2. Графический интерфейс – это вид пользовательского интерфейса, где элементы интерфейса исполнены на экране в виде графических изображений, элементов управления и др.
3. По типу управления (жестовый, голосовой, тактильный) – это технология взаимодействия, с помощью которой можно управлять устройствами, которые «понимают» жесты (движения тела). Используется в ноутбуках, смартфонах и планшетах. Бывают двух типов: контактный (тактильный) и бесконтактный.

В итоге, для реализации программного приложения был выбран графический интерфейс, так как, по сравнению с иными видами он более актуален и распространен.

Преимущества графического интерфейса:

1. Графический интерфейс более дружественный и приятный для глаз;
2. GUI обеспечивает легкий доступ к функциям системы и приложений;
3. GUI может использоваться с очень небольшими знаниями, что полезно для начинающих пользователей;
4. GUI обеспечивает визуально богатый вывод и информацию относительно CLI;

После анализа всех аспектов, был создан собственный интерфейс программного приложения.

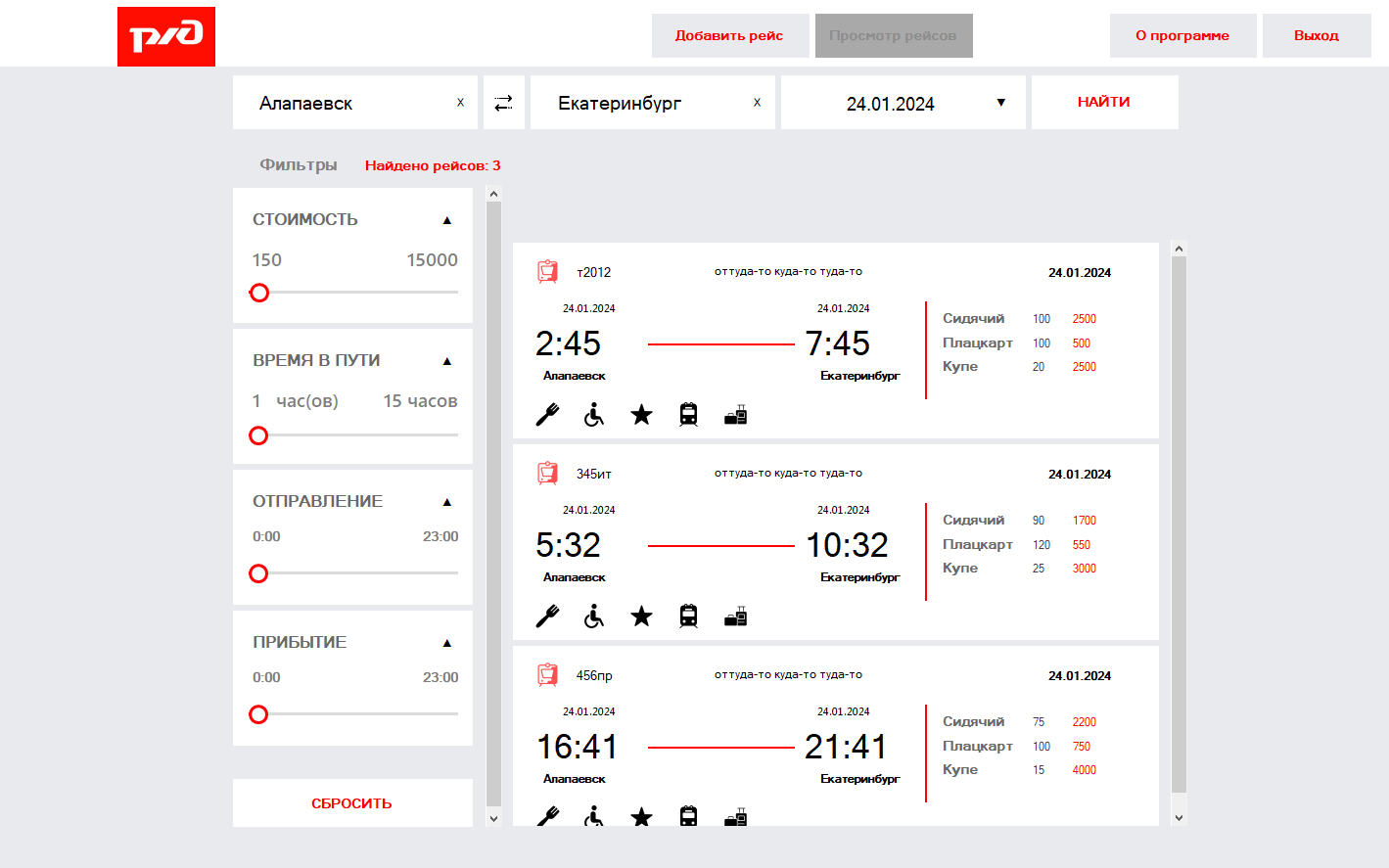


Рисунок 15 – Главное окно программного приложения

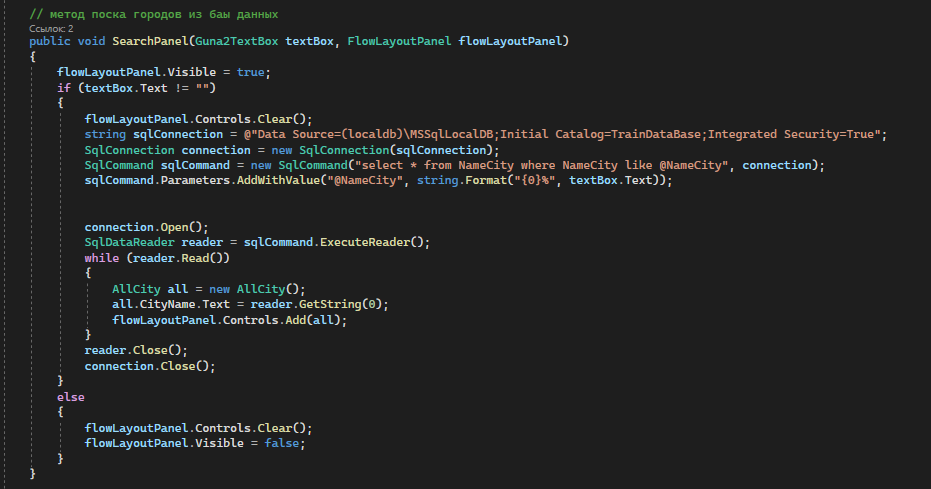


Рисунок 16 – Логика заполнения панели с городами следования

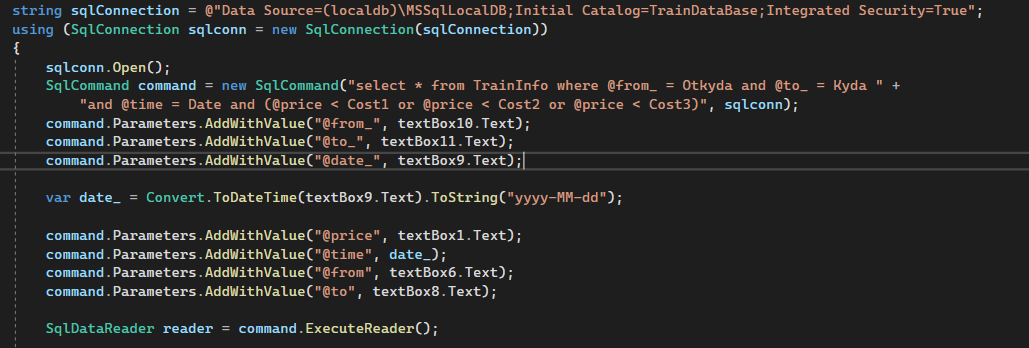


Рисунок 17 – Логика заполнения карточки рейса (1 часть)



Рисунок 18 – Логика заполнения карточки рейса (2 часть)

При нажатии на кнопку «НАЙТИ» производится подключение к базе данных, создается команда через класс *«SqlCommand»*, с помощью которой программа может обратиться к базе данных и получить из нее необходимую информацию. Далее, после того как данные были получены, создается экземпляр карточки *«UserControl1»* (Рис. 19), после чего в него записываются полученные данные и заполняются в панель на главном окне. Каждому рейсу присваивается своя собственная карточка.



Рисунок 19 – Карточка рейса

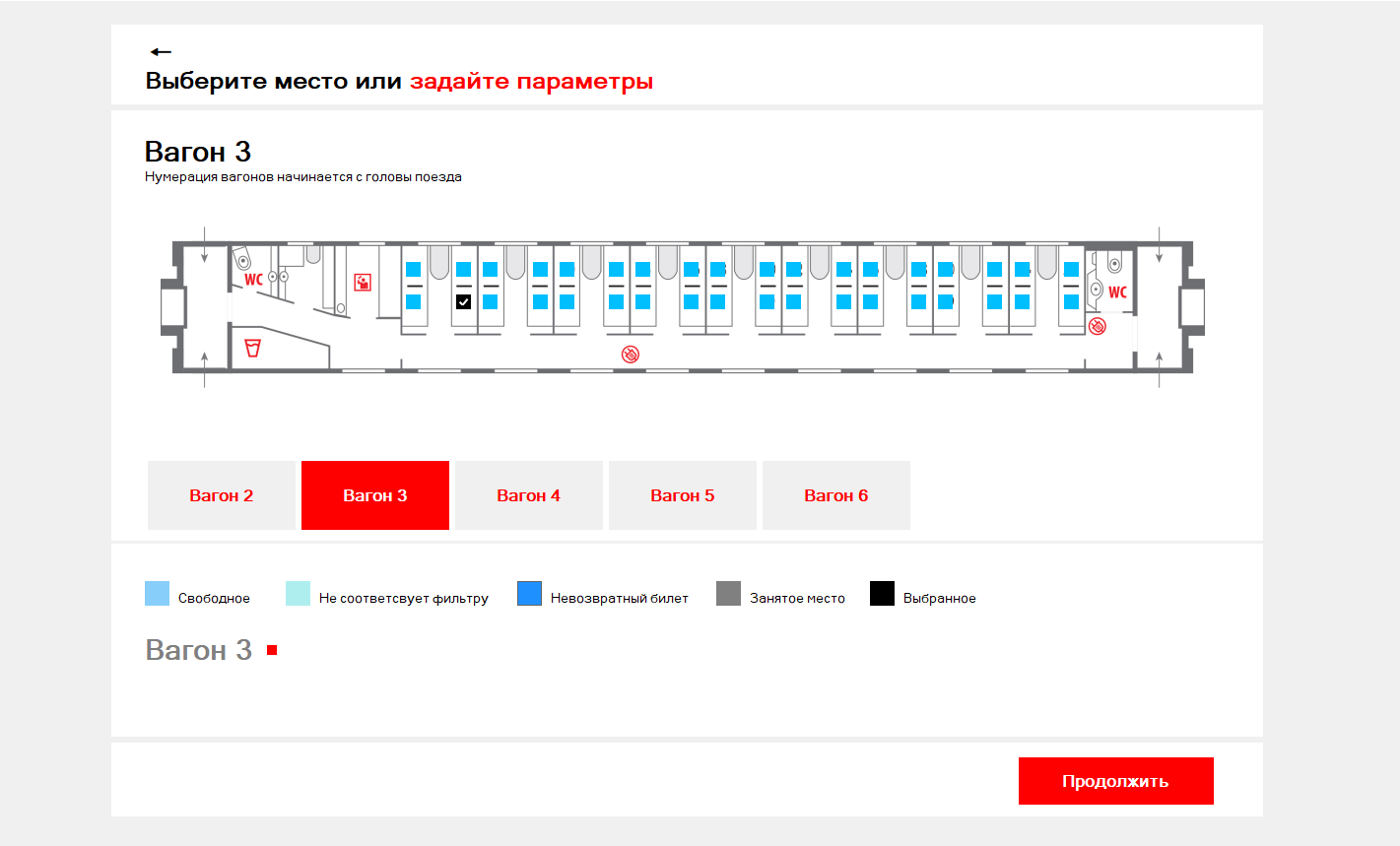
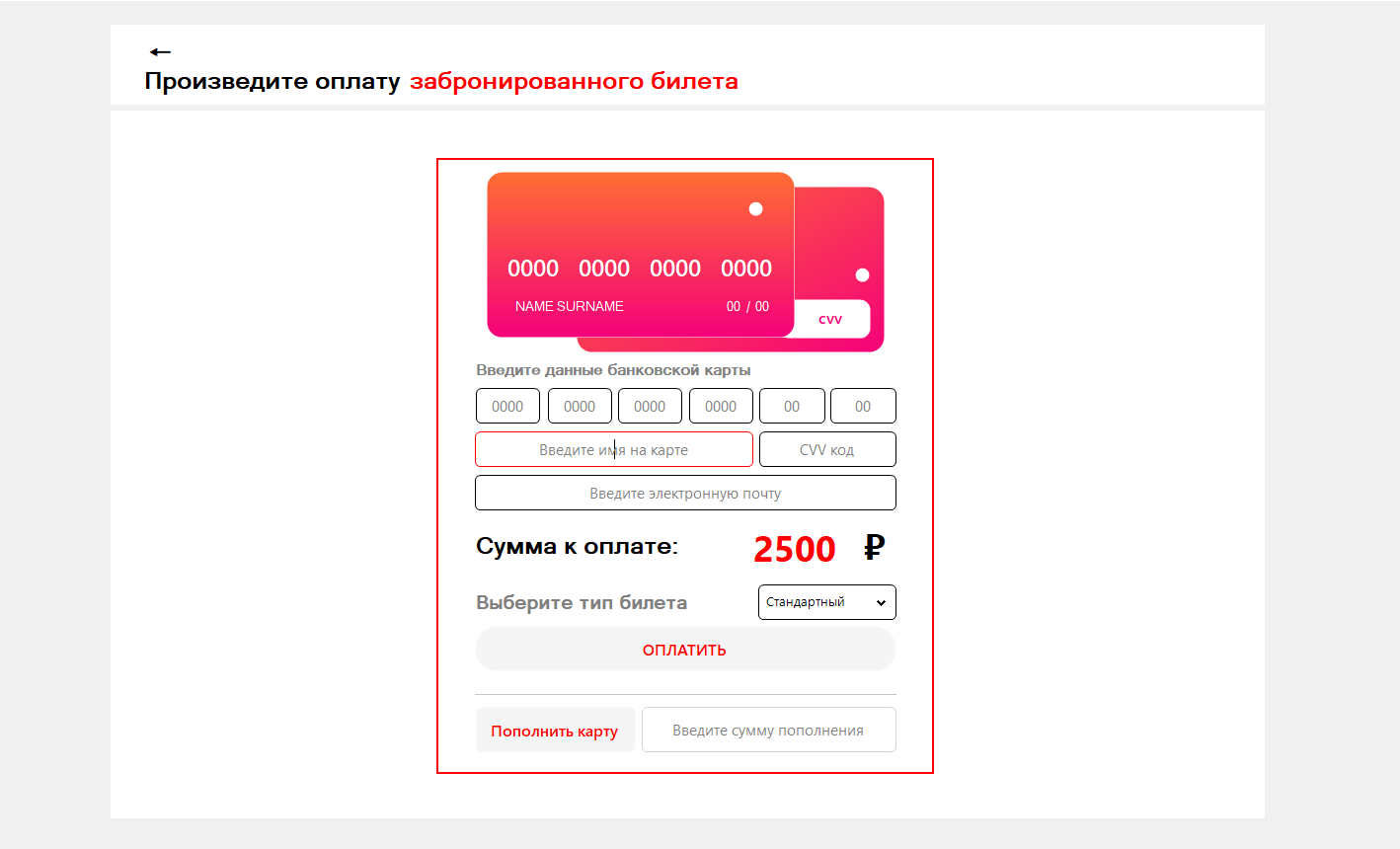


Рисунок 20 – Окно выбора вагона и места

При выборе вагона, меняется графическое отображение визуального просмотра вагона, тем самым, демонстрируется, какие места свободны.

  
Рисунок 21 – Окно оплаты билета

Окно оплаты билета содержит поле панели, в которую программным способом вносится интерфейс оплаты. При внесении в первых 4 цифр в поле банковской карты, будет автоматически определена платежная система (Рис. 21)

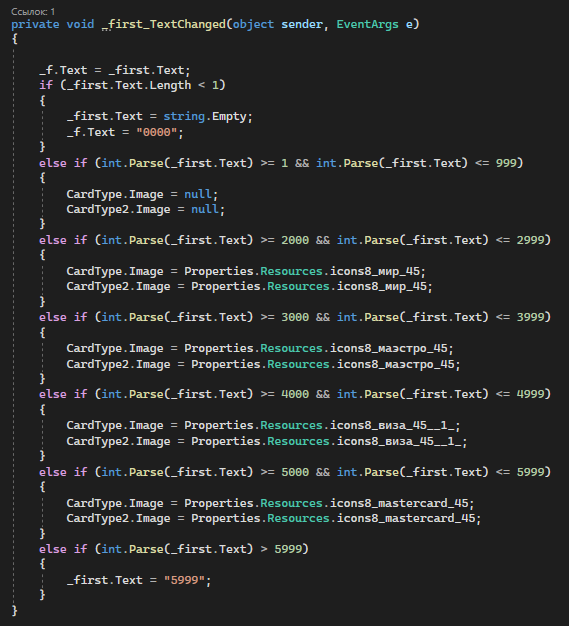


Рисунок 22 – Логика определения платежной системы банковской карты

При внесении 4 цифр в поле, проверяется диапазон чисел, например, платежная система «МИР» имеет код от 2000 до 2999, где первые 2 цифры – код платежной системы, вторые 2 цифры – подразделение банка. После проверки введенных чисел, поставляется необходимая картинка (Рис. 23)

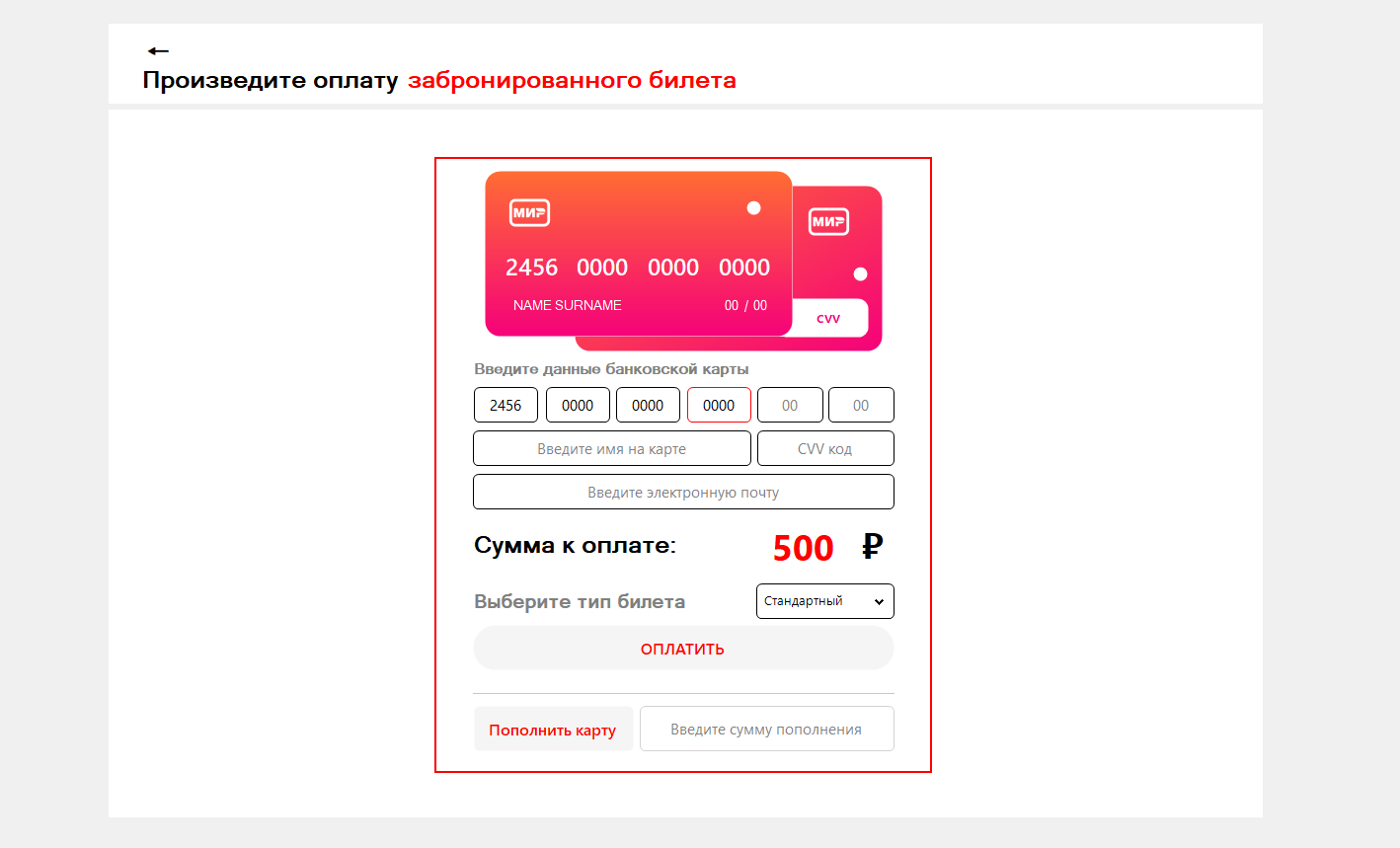


Рисунок 23 – Демонстрация проверки определения платежной системы



Рисунок 24 – Окно ожидания оформления билета

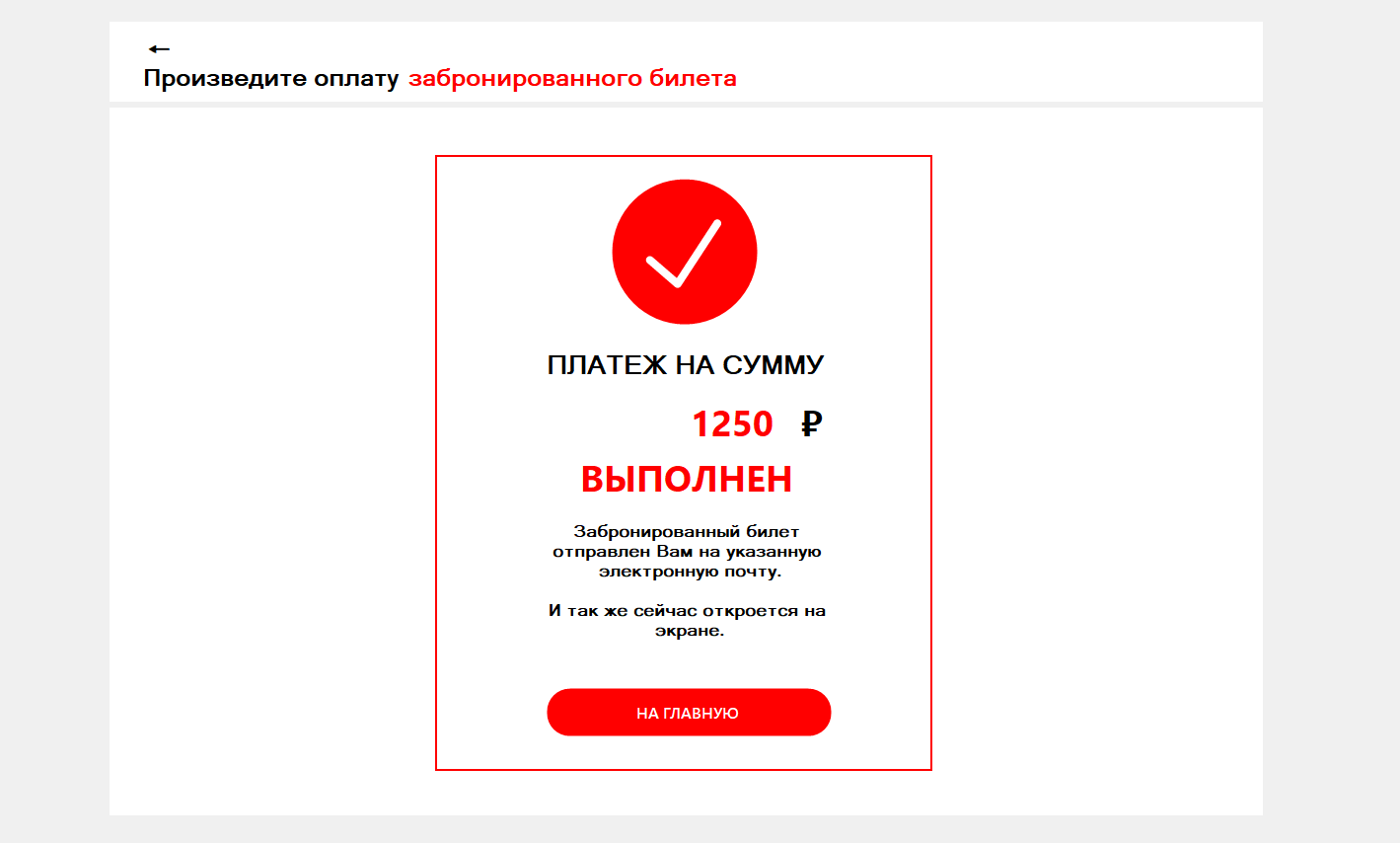


Рисунок 25 – Окно подтверждения выполненного платежа

После выполнения оформления билета, программа заполняет созданный в Office Word шаблон (Рис. 25)

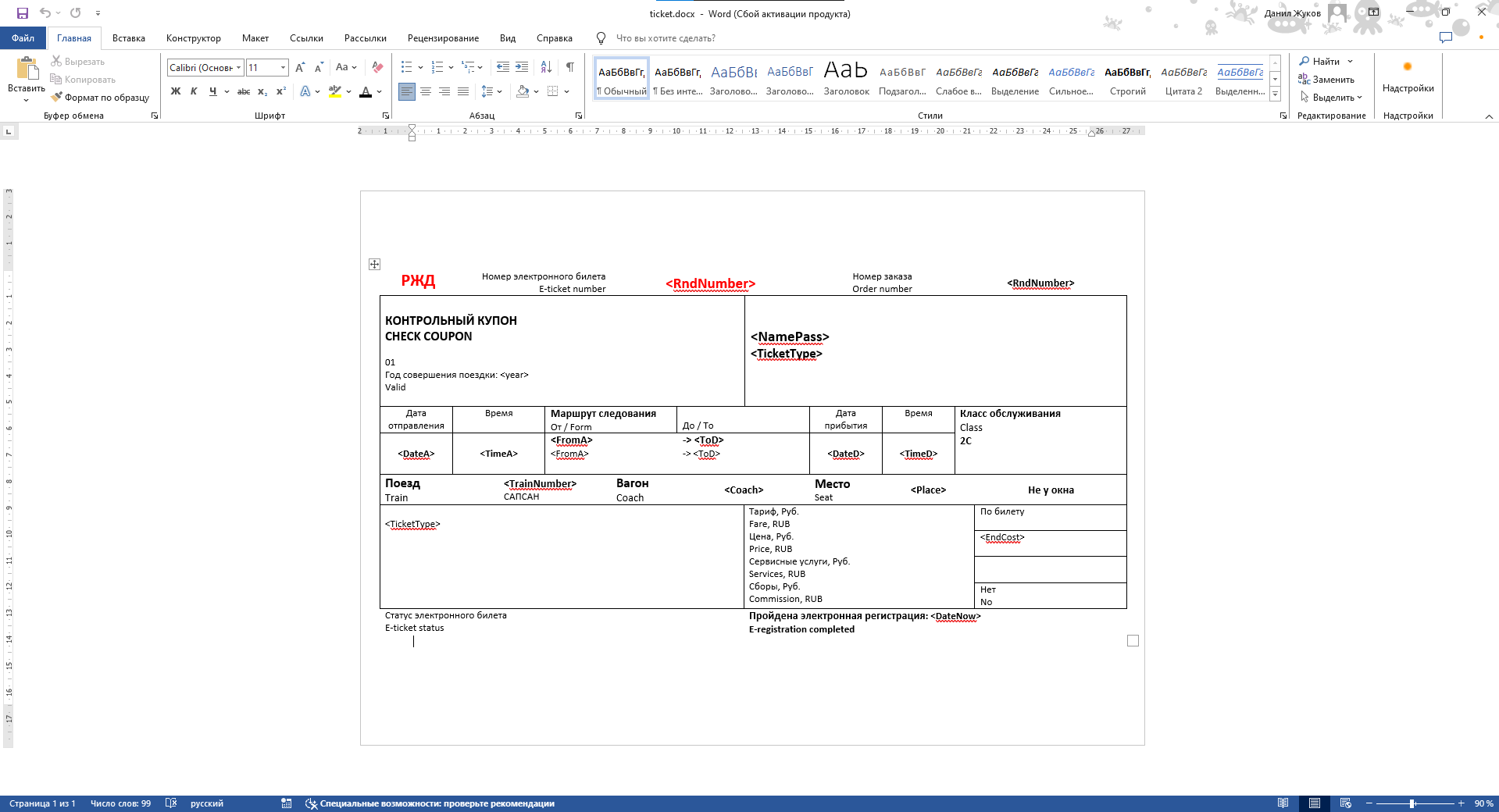


Рисунок 25 – Шаблон билета

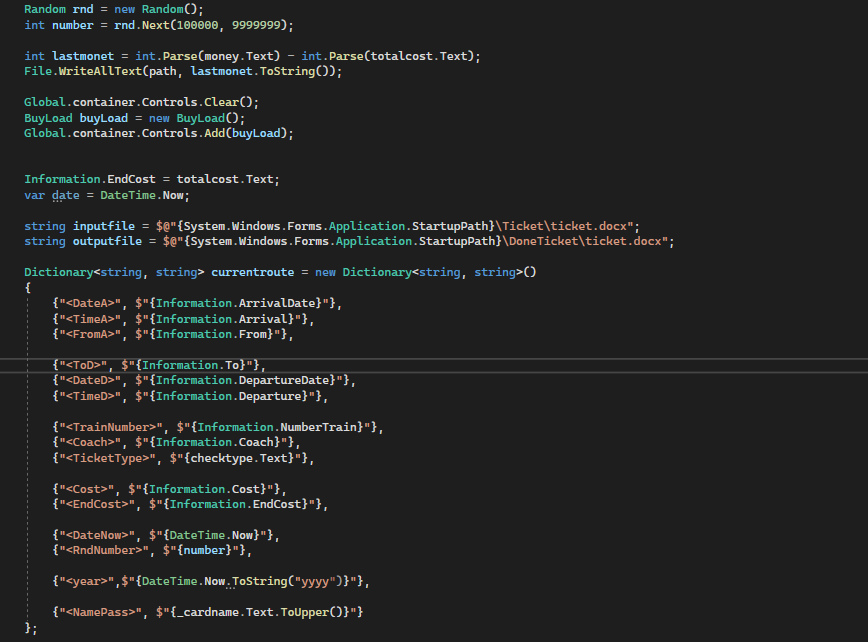


Рисунок 26 – Заполнение шаблона билета (1 часть)

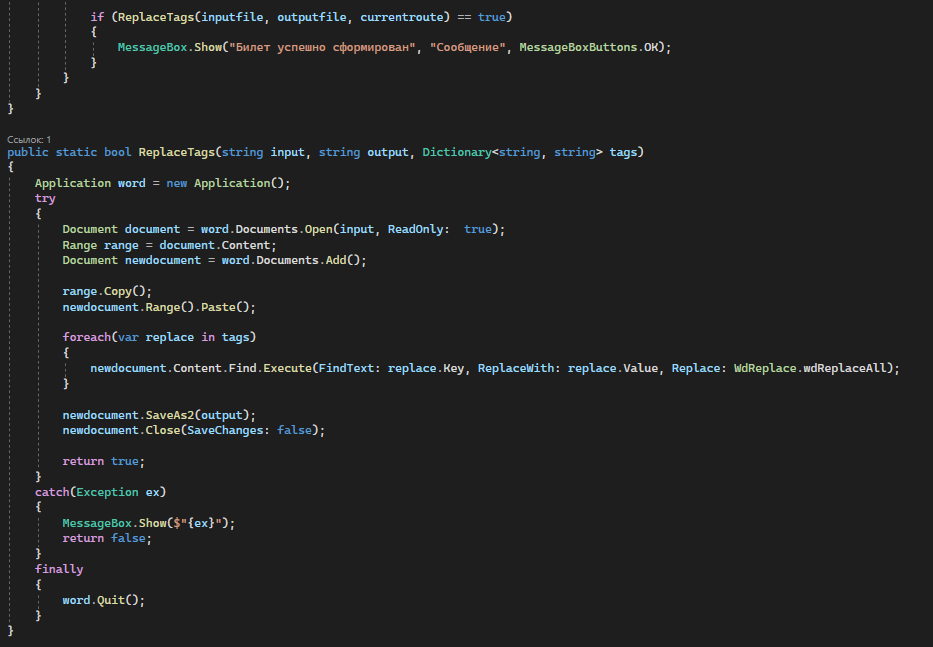


Рисунок 27 – Заполнение шаблона билета (2 часть)

При заполнении шаблона создается словарь, в который записаны подлежащие замене атрибуты в формате – <АТРИБУТ>. После чего, программа открывает шаблон файла, создает его копию и заполняет данными, с заменой атрибутов, после чего, сохраняет по указанному пути и задает название. (Рис. 28)

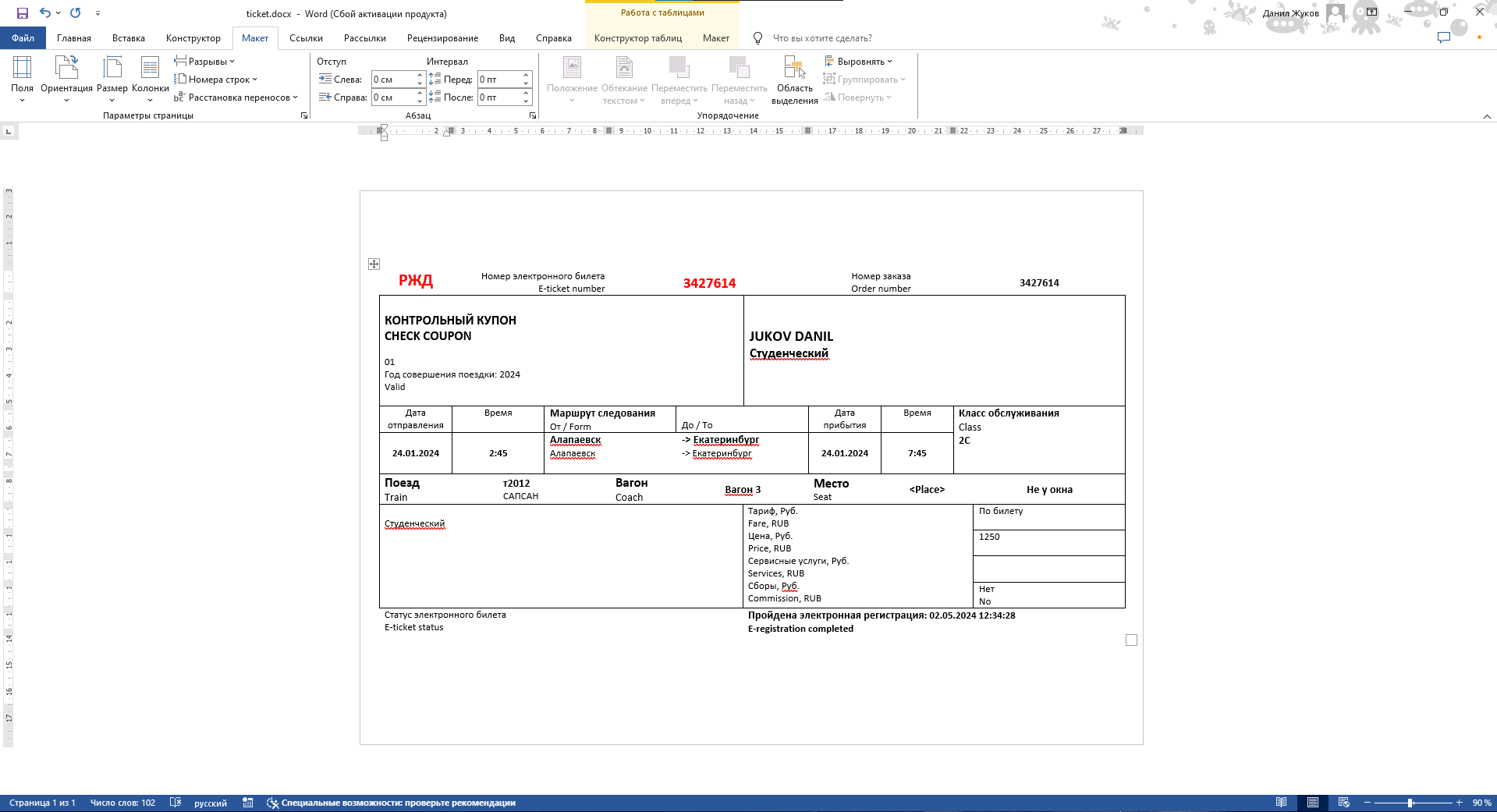


Рисунок 28 – Пример заполненного шаблона билета

Далее готовый билет можно распечатать, отправить на почту, хранить в электронном виде, конвертировать в pdf и т.д.

Так же, если необходимого рейса не оказалось в базе данных, его можно добавить через конструктор (Рис. 30), предварительно пройдя этап авторизации. (Рис. 29).

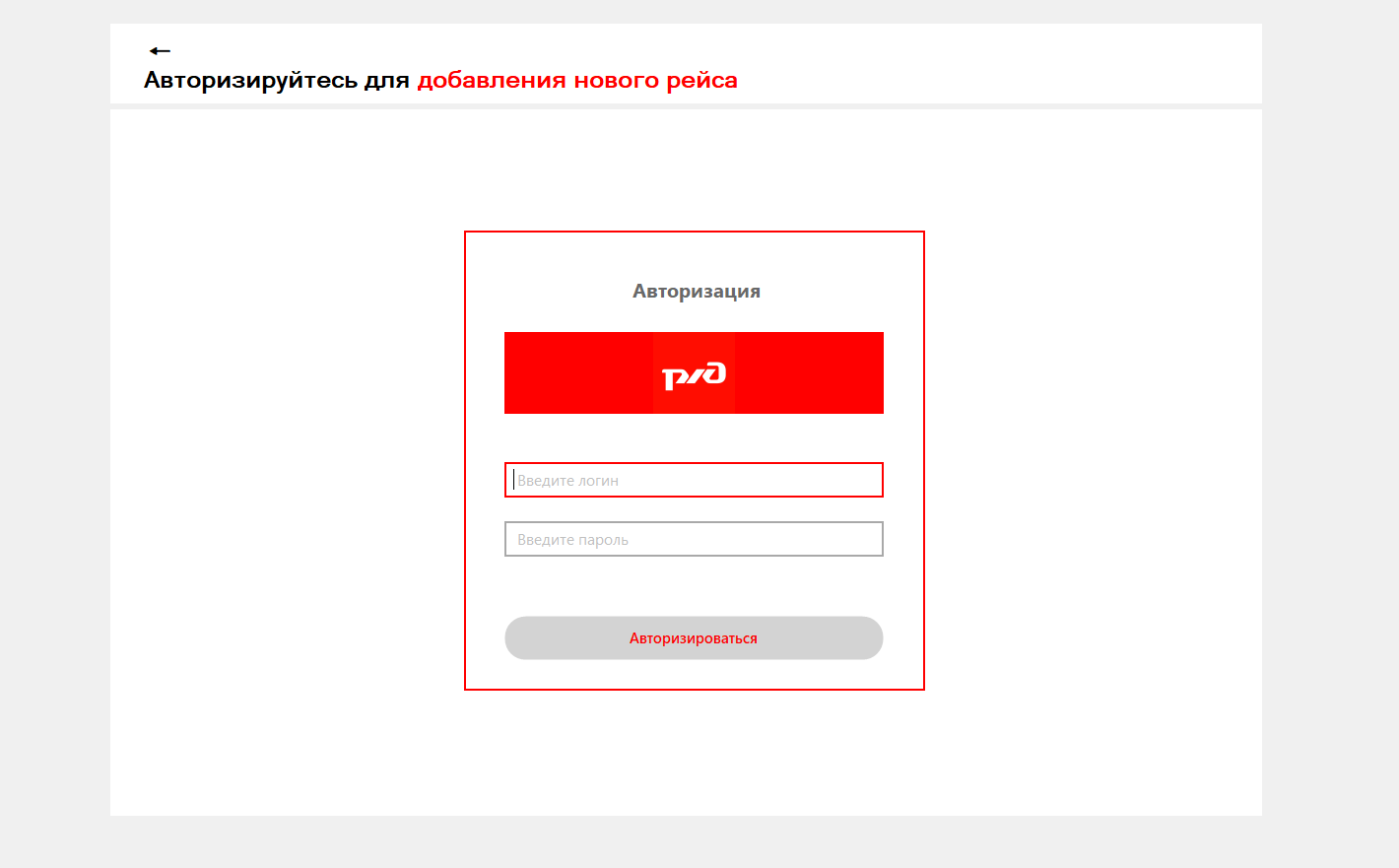


Рисунок 29 – Окно авторизации, для добавления рейса

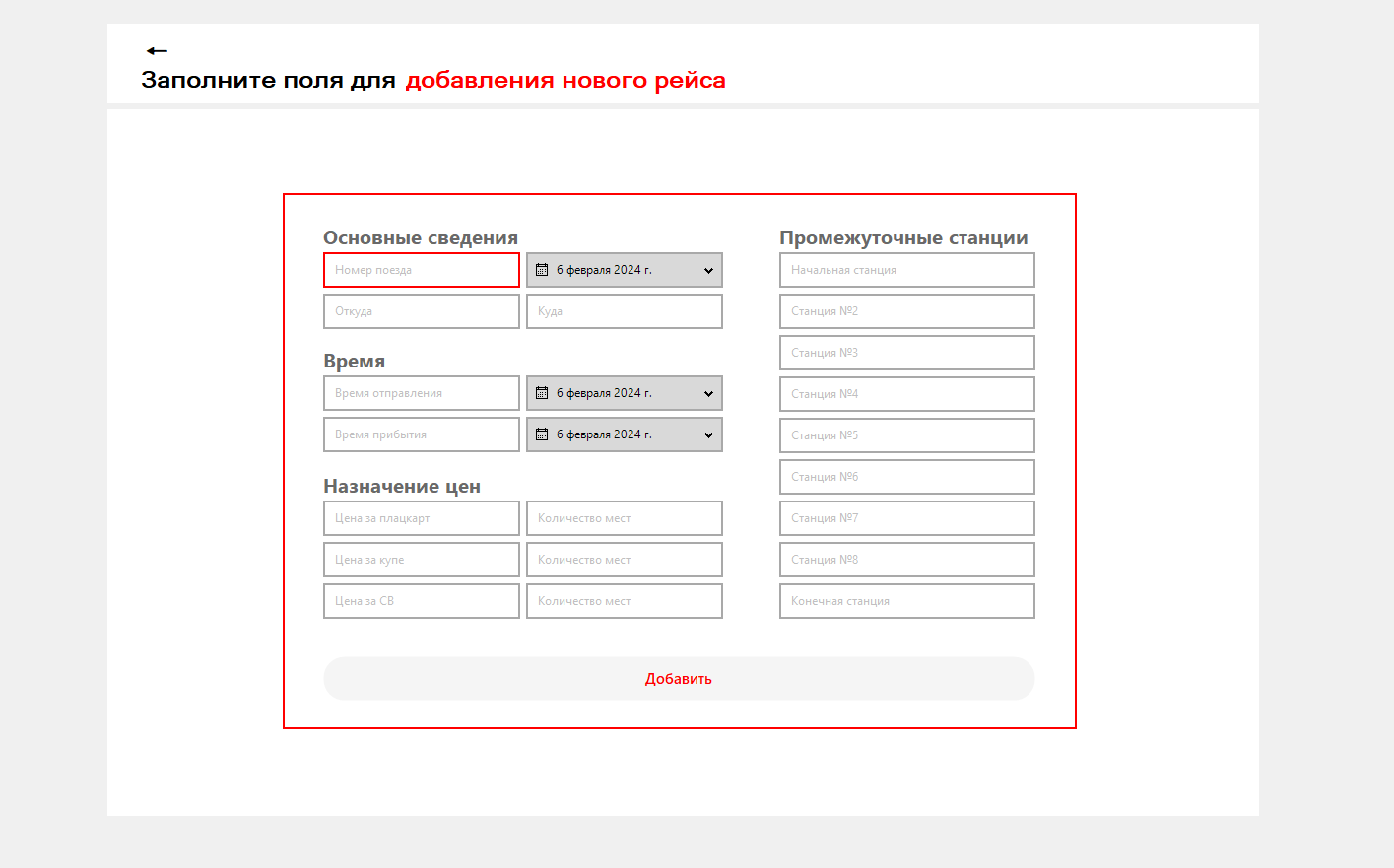


Рисунок 30 – Окно конструктора добавления рейса

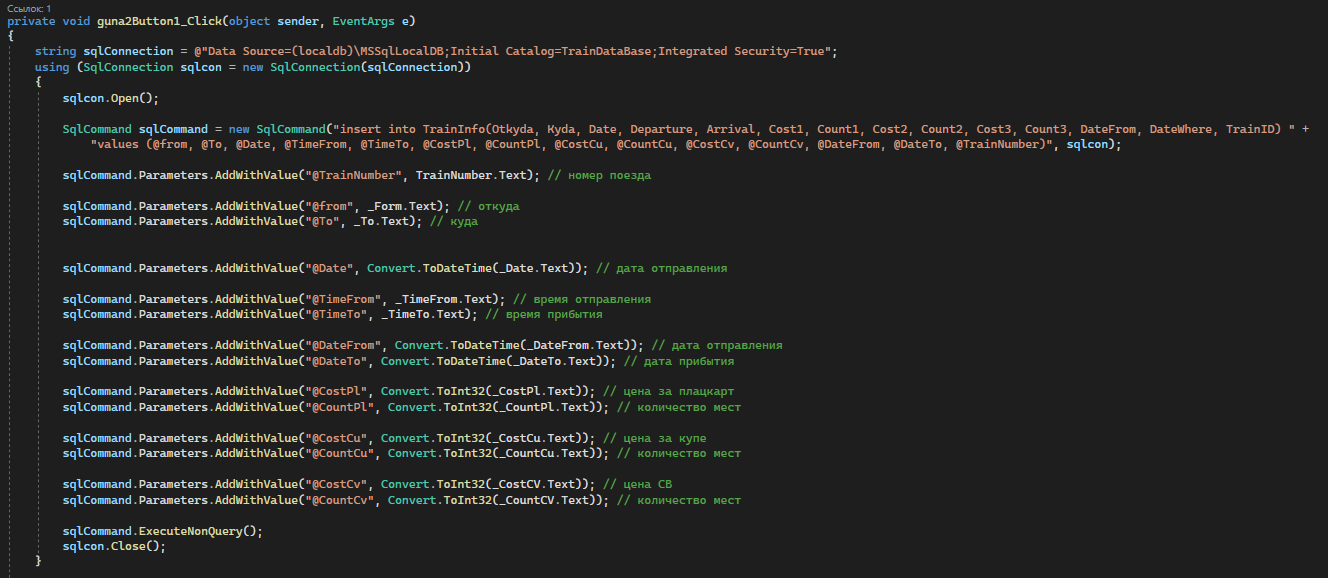


Рисунок 31 – Логика добавления нового рейса в базу данных

Производится подключение к базе данных, создается команда, с помощью которой база данных заполнит таблицу данными, которые в свою очередь буду получены из текстовых полей, который будут заполнены Рис. 31). После чего будут заполнены промежуточные станции, которые находятся на пути сообщения (Рис. 32).

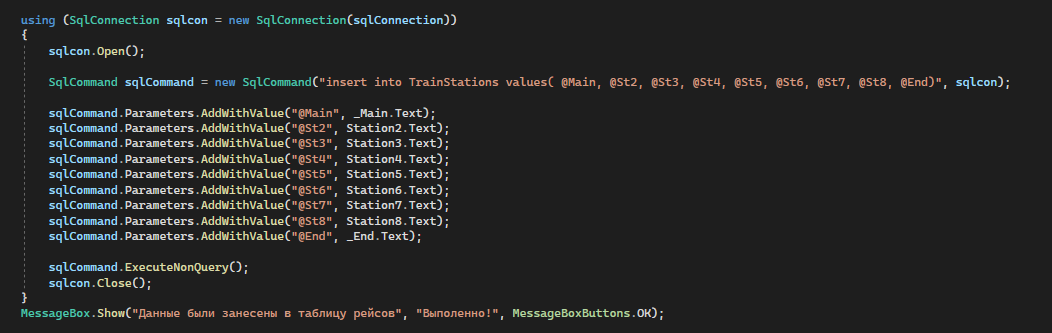


Рисунок 32 – Логика добавления станций в базу данных

Так же в приложении реализована возможность просмотра всех внесенных в базу данных рейсов (Рис. 33)

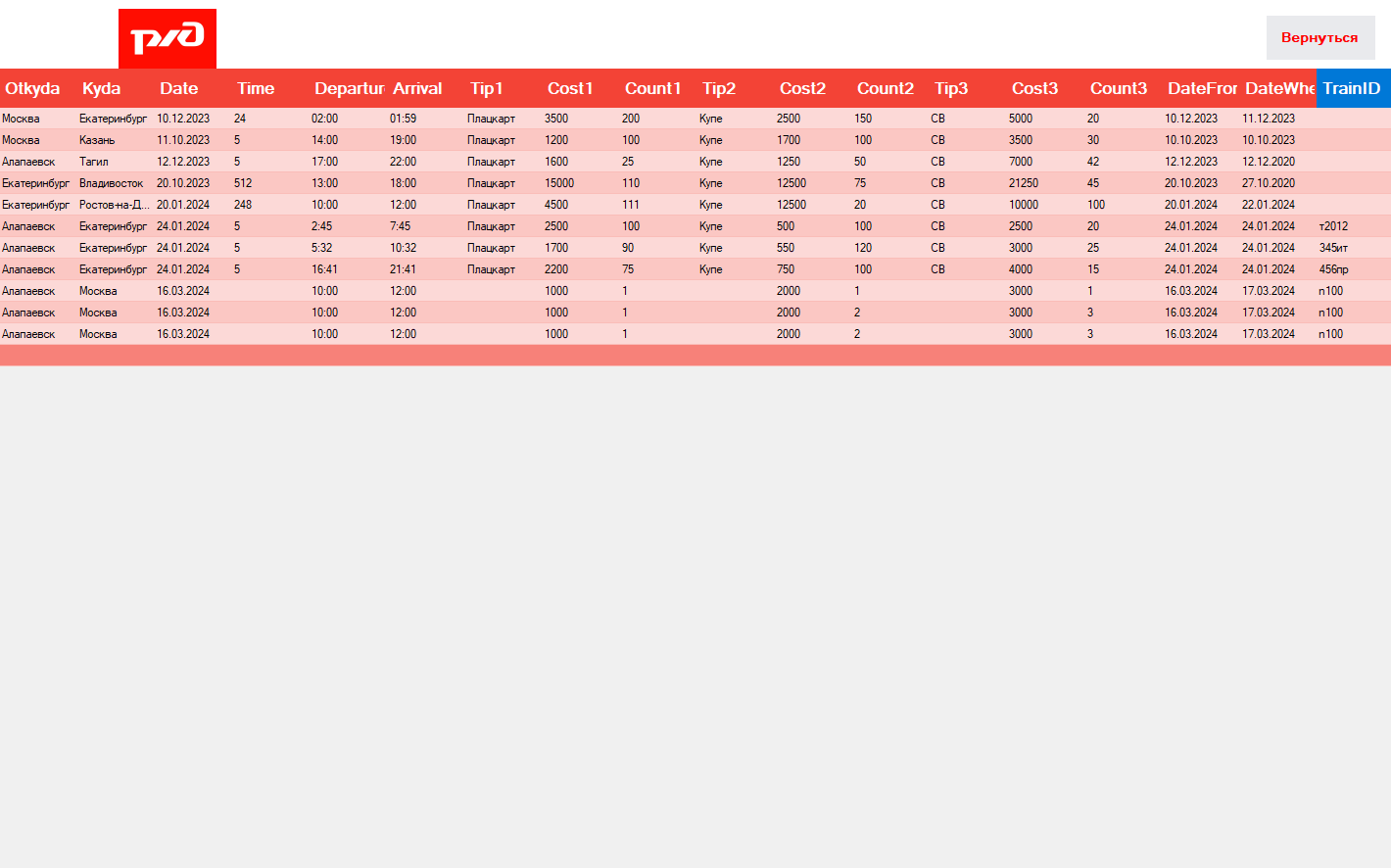


Рисунок 33 – Окно просмотра всех рейсов базы данных

При загрузке окна, производится подключение к базе данных. Создается запрос, который необходимо выполнить. Создается экземпляр класса *«SqlDataAdapter»* - класс, который обращается к базе данных и выполняет вышеописанный запрос. После чего создается виртуальная таблица с помощью класса *«DataTable»* в которою записываются данные, полученные при выполнении команды адаптером. В конечном итоге, заполняется таблица, которая находится в окне программы. (Рис. 34)

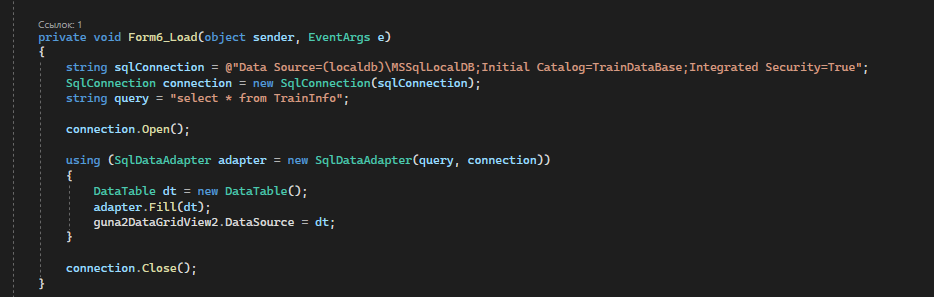


Рисунок 34 – Логика заполнения данных таблицы из базы данных

1. **Тестирование и отладка программного приложения**

Таблица 2. Тестирование встроенных функций

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Проводимое действие | Предполагаемый результат | Итоговый результат |
| 1 | Поиск рейса с установленной фильтрацией | Корректное применение фильтров при поиске подходящего рейса | Корректное применение фильтров при поиске подходящего рейса |
| 2 | Добавление рейса с помощью конструктора, интегрированного в программу | Успешное добавление рейса в таблицу базы данных | Добавление рейса в таблицу базы данных |
| 3 | Оформление билета на несколько пассажирский мест | Правильное вычисление цены за оформленные билеты | Цена за оформленные билеты вычисляется корректно |
| 4 | Поиск рейса без начальной и конечной станций | Программа сообщит о том, что необходимо указать начальную и конечную станции | Предупреждение о том, что станции не выбраны, выделение полей для заполнения |
| 5 | Проверка правильности определения платежной системы, при оформлении билета | Корректное отображение платежных систем по уникальному номеру карты | Платежные системы отображаются корректно |
| 6 | Оплата билета при недостаточном количестве денежных средств на банковской карте | Сообщение о том, что недостаточно средств | Сообщение о том, что недостаточно денежных средств для совершения операции |