Министерство образования и молодежной политики

Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Специальность 09.02.07: «Информационные системы и программирование»

Разработка информационно-поисковой системы для бронирования авиабилетов

**Пояснительная записка**

к курсовому проекту

КР-ПР-41-01-2022-ПЗ

Разработал:

Студент гр. ПР-41 \_\_\_\_\_\_\_\_\_\_\_\_\_\_/М.А. Варганов

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_/С.И. Овчинникова

2022

Министерство образования и молодёжной политики Свердловской области

ГАПОУ СО Екатеринбургский колледж транспортного строительства

Специальность 090207 «Информационные системы и программирование»

**ЗАДАНИЕ**

**на курсовой проект**

**Курс IV**

**Студенту:**

**Фамилия** Варганов

**Имя** Матвей

**Отчество** АлексеевичГруппа ПР-41

Тема курсового проекта: Разработка информационно-поисковой системы

для бронирования авиабилетов

**Курсовой проект должен содержать следующие разделы:**

Титульные листы

Задание на курсовой проект

Содержание

Введение

1. Описание предметной области. Постановка задачи
2. Выбор средств/методологии проектирования. Выбор СУБД
3. Построение инфологической (концептуальной) модели предметной области
4. Проектирование логической структуры БД
5. Формулировка ограничений целостности

5.1 Выявление полного перечня ограниченной целостности, присущего данной предметной области

5.2 Определение перечня ограничений целостности, которые будут контролироваться в данном проекте

5.3 Выбор способа реализации контроля целостности для каждого из ограничений

1. Проектирование физической структуры базы данных
2. Организация ввода данных в БД
3. Организация корректировки БД
4. Реализация запросов пользователя, получение отчетов
5. Разработка пользовательского интерфейса
6. Реализация проекта в среде конкретной СУБД

Условные сокращения

Заключение

Список литературы

Приложения

Приложение А Основные таблицы БД

Приложение Б Схема межтабличных связей

Приложение В Интерфейсные формы диалога пользователей с БД

Приложение Г Образцы отчетов пользователей

Приложение Д Образцы запросов пользователей к БД

Дата выдачи\_\_23 сентября 2022\_\_\_

Дата окончания\_\_\_\_5 декабря 2022\_\_

Преподаватель Овчинникова С.И

Содержание

Введение

В современных условиях, человек вынужден работать с гигантскими объемами информации. В связи с этим разработка программных продуктов, служащих для автоматизированного учета, весьма актуальна.

Системы обязаны представлять собой мощные средства, способные обрабатывать гигантские потоки данных высокой структурной сложности за минимум затраченного времени, обеспечивая дружественный диалог с пользователем Целью данной выпускной квалификационной работы является создание автоматизированной информационной системы, осуществляющей продажи авиабилетов. Разработка подобной системы весьма актуальна на данный момент. В современном мире самолеты являются не только самым быстрым видом транспорта, но и самым безопасным, в связи с этим авиаперелеты пользуются весьма высокой популярностью. Вследствие этого продаваемые на рейсы билеты востребованы и с высокой вероятностью найдут своего покупателя, при условии, что авиакомпания обеспечила клиенту полноценный доступ к нужной ему информации. Это и есть задача, решаемая современными автоматизированными информационными системами. Существует множество подобных разработок, позволяющих авиакомпаниям реализовывать авиабилеты, а пользователям приобретать их. Однако, зачастую, функциональность таких систем либо весьма ограничена, либо предоставляет достаточное количество информации, жертвуя дружественностью к пользователю.

Для реализации поставленной задачи были выбраны следующие средства разработки: MS Visual Studio 2022, MS SQL Server 2019, веб-платформа Angular, и языки программирования TypeScript, C#, SQL.

# 1 Описание предметной области

Целью данной курсовой работы является создание программного средства, осуществляющего поиск и бронирование авиабилетов. Авиакомпания занимается авиаперевозками пассажиров. Также она устанавливает маршруты полетов. Рейсы осуществляются по установленным маршрутам согласно расписанию. На каждый рейс существует определенное количество билетов. Продажа билета пользователю осуществляется при отправке запроса на бронирование, при условии, что данный билет до сих пор есть в наличии. Приобретя билет, пользователь предоставляет информацию о себе и становится пассажиром.

Система создаётся для обслуживания пользователей, бронирующих билеты на рейсы и осуществляющих поиск.

Абсолютно каждая авиакомпания использует определенную систему дистрибуции. Наиболее развитые используют GDS (глобальные дистрибьюторские системы, которые формируются из основных международных компьютерных систем резервирования). В итоге сервисы продаж авиабилетов при поиске информации пользуются ресурсами глобальных дистрибьюторских систем. Однако доступ к GDS является не бесплатным, поэтому в роли дистрибутивной системы для разрабатываемого продукта будет выступать БД, созданная MS SQL Server 2019.

В БД должна храниться информация о:

\* рейсах;

\* билетах;

\* клиентах;

\* аэропортах;

\* авиакомпаниях;

\* тарифах;

\* самолётах;

\* городах;

\* контактной информацией пользователя;

Разрабатываемая информационная система предназначена для продажи авиабилетов и упрощения доступа к нужной информации. Наличие данной разработки улучшает организационную работу авиаперевозчика за счёт отсутствия бумажной документации, поиск и систематизация которой занимали бы очень большое количество времени.

# 2 Выбор средств/методологии проектирования. Выбор СУБД

Для разработки информационной системы были выбраны язык программирования С#, MS Visual Studio 2022, язык программирования базы данных и запросов к ней SQL и система управления базами данных Microsoft SQL Server 2019, язык программирования TypeScript для разработки веб-интерфейса в фреймворке Angular.

Язык C# является объектно-ориентированным языком программирования. Разработан в 1998—2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET и впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270. Компилятор с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств, вроде Visual Studio.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников — языков C++, Java, Delphi, Модуля и Smalltalk — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++).

Среда разработки Microsoft Visual Studio 2022 — это набор инструментов и средств, предназначенных для помощи разработчикам программ любого уровня квалификации в решении сложных задач и создания новаторских решений. Роль Visual Studio заключается в том, чтобы улучшить процесс разработки и упростить разработку высокоэффективных программ. В Visual Studio содержатся инструменты для всех этапов разработки программного обеспечения (разработка, тестирование, развертывание, интеграция и управления). Visual Studio разрабатывается таким образом, чтобы обеспечить высокую надежность и совместимость. Visual Studio обладает удачным сочетанием безопасности, масштабируемости и взаимодействия.

SQL (англ. Structured Query Language — «язык структурированных запросов») – универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных. SQL основывается на исчислении кортежей. Язык SQL ориентирован на операции с данными, представленными в виде логически взаимосвязанных совокупностей таблиц-отношений. Важнейшая особенность его структур – ориентация на конечный результат обработки данных, а не на процедуру этой обработки. Язык SQL сам определяет, где находятся данные, индексы и даже какие наиболее эффективные последовательности операций следует использовать для получения результата, а потому указывать эти детали в запросе к базе данных не требуется. Основные особенности SQL:

\* независимость от конкретной СУБД – несмотря на наличие диалектов и различий в синтаксисе, в большинстве своём тексты SQL-запросов, содержащие DDL и DML, могут быть достаточно легко перенесены из одной СУБД в другую. Существуют системы, разработчики которых изначально ориентировались на применение по меньшей мере нескольких СУБД (например: система электронного документооборота Documentum может работать как с Oracle, так и с Microsoft SQL Server и IBM DB2). Естественно, что при применении некоторых специфичных для реализации возможностей такой переносимости добиться уже очень трудно;

\* наличие стандартов и набора тестов для выявления совместимости и соответствия конкретной реализации SQL общепринятому стандарту только способствует «стабилизации» языка. Правда, стоит обратить внимание, что сам по себе стандарт местами чересчур формализован и раздут в размерах (например, Core-часть стандарта SQL:2003 представляет собой более 1300 страниц текста);

\* декларативность – с помощью SQL программист описывает только то, какие данные нужно извлечь или модифицировать. То, каким образом это сделать, решает СУБД непосредственно при обработке SQL-запроса. Однако не стоит думать, что это полностью универсальный принцип — программист описывает набор данных для выборки или модификации, однако ему при этом полезно представлять, как СУБД будет разбирать текст его запроса. Чем сложнее сконструирован запрос, тем больше он допускает вариантов написания, различных по скорости выполнения, но одинаковых по итоговому набору данных;

\* реляционная основа языка – SQL является языком реляционных БД, поэтому он стал популярным тогда, когда получила широкое распространение реляционная модель представления данных. Табличная структура реляционной БД хорошо понятна, а потому язык SQL прост для изучения.

Microsoft SQL Server — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями.

На базе Microsoft SQL Server 2019 могут быть построены решения для компаний малого, среднего и крупного бизнеса. SQL Server 2019 выпускается в двух основных редакциях Standard и Enterprise. На основе последней создана также редакция для разработчиков Developer Edition, лицензия на которую позволяет разрабатывать и тестировать системы и приложения.

Enterprise-версия системы SQL Server 2019 представляет собой комплексную платформу, которая позволяет работать даже с самыми требовательными корпоративными OLTP-системами и хранилищами данных. Она обладает значительной масштабируемостью, возможностью создавать громадные хранилища данных, продвинутыми средствами анализа и усиленной безопасностью, что позволяет использовать ее как основу для критически 23 важных бизнес-приложений. Эта редакция позволяет консолидировать серверы и выполнять большое число OLTP-операций и крупные отчеты. Редакцию Microsoft SQL Server 2019 Enterprise характеризуют:

\* высокий уровень доступности – непрерывность бизнес-процессов обеспечивается благодаря защите данных от дорогостоящих человеческих ошибок и максимальному уменьшению сроков аварийного восстановления;

\* производительность и масштабируемость – инфраструктура на основе SQL Server 2019 Enterprise позволяет справиться с любыми пиковыми нагрузками;

\* безопасность – вопросы конфиденциальности, а также соответствия нормативным требованиям решаются с помощью встроенных средств защиты от несанкционированного доступа;

\* управляемость – автоматические диагностика, калибровка и настройка инфраструктуры позволяют управлять огромными объемами данных, значительно сократив издержки на управление и обслуживание;

\* бизнес-аналитика. SQL Server 2019 Enterprise помогает легко собрать и проанализировать большие объемы данных из хранилищ или киосков.

Кроме редакций Enterprise и Standard, существуют специализированные редакции SQL Server 2019, одна из которых Express. Редакция Express — также доступная для бесплатной загрузки, эта редакция идеальна для обучения и создания настольных и небольших серверных приложений

Angular представляет фреймворк от компании Google для создания клиентских приложений. Прежде всего он нацелен на разработку SPA-решений (Single Page Application), то есть одностраничных приложений. В этом плане Angular является наследником другого фреймворка AngularJS. В то же время Angular это не новая версия AngularJS, а принципиально новый фреймворк.

Angular предоставляет такую функциональность, как двустороннее связывание, позволяющее динамически изменять данные в одном месте интерфейса при изменении данных модели в другом, шаблоны, маршрутизация и так далее.

Одной из ключевых особенностей Angular является то, что он использует в качестве языка программирования TypeScript.

Преимущества Angular заключается в обилии возможностей, он помогает привязывать компоненты приложения друг к другу, передавать данные, анимировать интерфейсы и пр. Для простых проектов его функциональность может быть избыточной, но для сложных SPA-приложений она незаменима.

Фреймворк позволяет создавать не только веб-приложения. С его помощью можно писать код, который может быть адаптирован под другую среду. Например, приложение сможет работать в мобильной или десктопной операционной системе. С помощью Angular можно создать даже приложение для дополненной реальности.

Разработчики Angular — сотрудники Google, а поддержка большой корпорации помогает фреймворку развиваться. При этом благодаря свободной лицензии и открытому исходному коду развивать его могут и сторонние разработчики.

3 Построение концептуальной модели предметной области

Был проведён подробный анализ предметной области, и на основе результатов этого анализа были выделены следующие сущности:

\* Регистрация;

\* Рейс;

\* Аэропорт;

\* Самолёт;

\* Авиалиния;

\* Город;

\* Страна;

\* Клиент;

\* Паспорт;

\* Контактные данные;

\* Тариф.

В процессе функционирования сущности взаимодействуют друг с другом. В концептуальной модели взаимодействие между сущностями выражается с помощью связей, основными из которых являются следующие:

Связь Регистрация – Рейс (Рисунок 3.1). Показывает, что регистрация содержит рейс. Связь многие к одному, т.к. многие регистрации могут содержать один рейс.



Рис. 3.1 – Связь Регистрация – Рейс

Связь Регистрация – Клиент (Рис. 3.2). Показывает, что регистрация содержит клиента. Связь 1 к М, так как регистрация может содержать множество клиентов.



Рис. 3.2 – Регистрация – Клиент

Связь Регистрация – Контактные данные (Рис. 3.3), демонстрирует, что регистрация содержит множество контактных данных.



Рис. 3.3 – Регистрация – Контактные данные

Связь Рейс - Самолёт (Рисунок. 3.4), показывает, что на один рейс может быть только один самолёт.

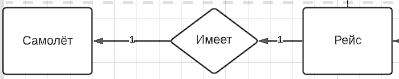


Рис. 3.4 – Связь Рейс - Самолёт

Связь Авиакомпания – Тариф (Рис. 3.5). У каждой авиакомпании есть свой список тарифов, поэтому связь 1 ко многим.



Рис. 3.5 – Авиакомпания - Тариф

Ранее были рассмотрены некоторые связи с итоговой таблицей Регистрация, далее будет рассмотрена вся модель. Совместив связи между сущностями, и добавив дополнительные была получена следующая модель (Рис. 3.6).

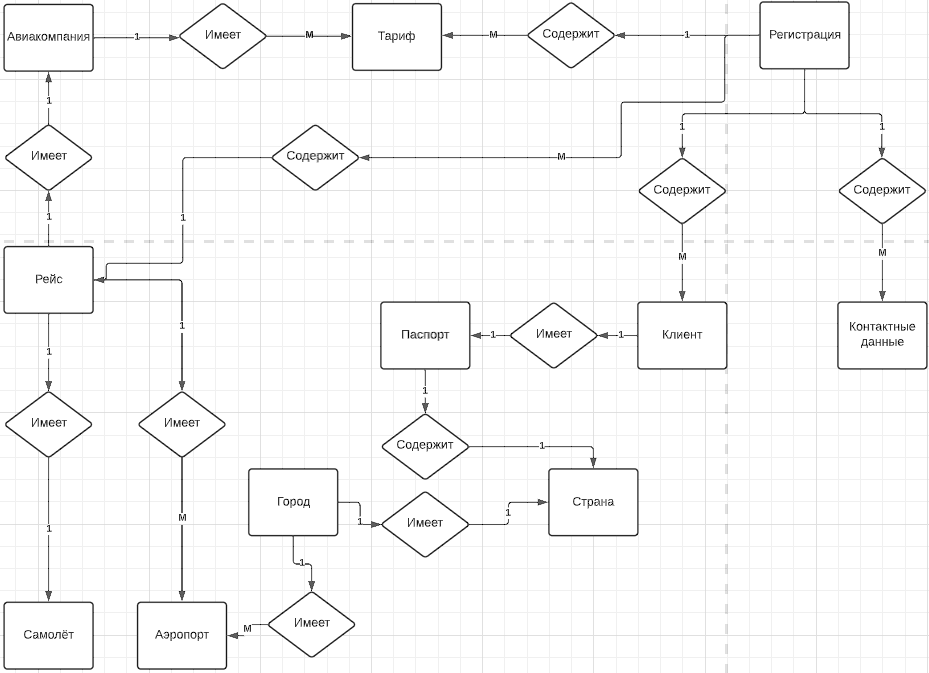


Рис. 3.6 – Модель предметной области

Рассмотрим эту модель подробнее. Так как система предназначена для бронирования авиабилетов, вся связи ведут к основной таблице регистрация, там хранится вся информация о бронировании билета, но основана она на информации о рейсе. Таблица рейс имеет самолёт, на котором будет проходить перелёт. Так как любой рейс осуществляет авиакомпания, они связаны, а у авиакомпании есть свой список тарифов, отражающих уровень комфорта перелёта. Так же рейс имеет несколько аэропортов, аэропорт отбытия и прибытия, логично отметить, что аэропорт связан с местоположением - городом и страной.

При регистрации на рейс клиент выбирает понравившейся ему маршрут из сущности Рейс, если его всё устраивает, то он вписывает личные данные, которые включают в себя паспорт, привязанный к сущности Клиент, у клиента может быть только один паспорт, поэтому связь один к одному, так как клиент имеет гражданство, то его паспорт привязан к стране. Контактные данные представляют из себя данные, по которым можно будет связаться с клиентом.

# 4 Проектирование логической структуры БД

На основе сущностей концептуальной модели предметной области были сформированы следующие таблицы:

Таблица 4.1 – атрибуты сущности Страна

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| Код страны | Число |
| Название | Текст |

Атрибуты таблицы:

\* Код страны – Уникальный идентификатор страны, не может повторяться и является первичным ключом;

\* Название – Название страны.

Таблица 4.2 – атрибуты сущности Город

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| Код города | Число |
| Название | Текст |
| Код страны | Число |

Атрибуты таблицы:

\* Код города – Уникальный идентификатор города, не может повторяться и является первичным ключом;

\* Название – Название города;

\* Код страны – Ключ страны, к которому привязан город.

Таблица 4.3 – атрибуты сущности Паспорт

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| Код паспорта | Число |
| Дата рождения | Дата |
| Код гражданства | Число |
| Номер и серия | Текст |
| Код страны выдачи | Число |
| Дата окончания действительности | Дата |

Атрибуты таблицы:

\* Код паспорта – Уникальный идентификатор паспорта, не может повторяться и является первичным ключом;

\* Дата рождения – Дата рождения, не может повторяться и является первичным ключом;

\* Код гражданства – Ключ страны гражданства, к которому привязан паспорт;

\* Номер и серия паспорта – Серия и номер паспорта

\* Код страны выдачи – Ключ страны выдачи, к которому привязан паспорт;

\* Дата окончания действительности – Дата, до которой годен документ.

Таблица 4.4 – атрибуты сущности Клиент

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| Код клиента | Число |
| Имя | Текст |
| Фамилия | Текст |
| Отчество | Текст |
| Код паспорта | Число |

Атрибуты таблицы:

\* Код клиента – Уникальный идентификатор клиента, не может повторяться и является первичным ключом;

\* Имя – Имя клиента;

\* Фамилия – Фамилия клиента;

\* Отчество – Отчество клиента;

\* Код паспорта – Ключ паспорта, к которому привязан клиент.

Таблица 4.5 - атрибуты сущности Аэропорт

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| Код аэропорта | Число |
| Код РФ | Текст |
| Название | Текст |
| Код города | Число |

Атрибуты таблицы:

\* Код аэропорта – Уникальный идентификатор аэропорта, не может повторяться и является первичным ключом;

\* Код РФ – Код аэропорта РФ;

\* Название – Название аэропорта;

\* Код города – Ключ города, которому привязан аэропорт;

Таблица 4.6 – атрибуты сущности Авиалиния

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| Код авиалинии | Число |
| Название | Текст |
| Картинка | Текст |

Атрибуты таблицы:

\* Код авиалинии – Уникальный идентификатор авиалинии, не может повторяться и является первичным ключом;

\* Название – Название аэропорта;

\* Картинка – Картинка авиалинии, хранит путь к картинке.

Таблица 4.7 – атрибуты сущности Самолёт

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| Код самолёта | Число |
| Модель | Текст |
| Бизнес места | Число |
| Эконом места | Число |
| Картинка мест | Текст |
| Описание | Текст |

Атрибуты таблицы:

\* Код самолёта – Уникальный идентификатор самолёта, не может повторяться и является первичным ключом;

\* Модель – Модель самолёта;

\* Бизнес места – Количество мест в бизнес классе;

\* Эконом места – Количество мест в эконом классе;

\* Картинка мест – Картинка схемы мест в самолёте, хранит путь к картинке;

\* Описание – Количество мест в эконом классе.

Таблица 4.8 – атрибуты сущности Рейс

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| Код рейса | Число |
| Код авиалинии | Число |
| Дата начала | Дата и время |
| Дата окончания | Дата и время |
| Цена | Число |
| Название рейса | Текст |
| Время в полёте | Текст |
| Код самолёта | Число |
| Место выхода на посадку | Текст |
| Аэропорт отправления | Число |
| Аэропорт прибытия | Число |

Атрибуты таблицы:

\* Код рейса – Уникальный идентификатор рейса, не может повторяться и является первичным ключом;

\* Код авиалинии – Ключ авиалинии, к которой привязан рейс;

\* Дата начала – Время вылета самолёта и аэропорта отбытия;

\* Дата окончания – Время, по которому самолёт прибудет в аэропорт прибытия;

\* Цена – Цена за рейс, без наценки;

\* Название рейса – Код рейса;

\* Время в полёте – Время в минутах, которое займёт перелёт;

\* Код самолёта – Ключ самолёта, к которому привязан рейс;

\* Место выхода на посадку – Место выхода на посадку в аэропорту;

\* Аэропорт отправления – Ключ аэропорта прибытия, к которому привязан рейс;

\* Аэропорт прибытия – Ключ аэропорта прибытия, к которому привязан рейс.

Таблица 4.9 – атрибуты сущности Тариф

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| Код тарифа | Число |
| Код авиалинии | Число |
| Название | Текст |
| Ручная кладь | Логический |
| Багаж | Логический |
| Питание | Логический |
| Обмен | Логический |
| Возврат | Логический |
| Наценка | Число |
| Страховка | Логический |
| Выбор места | Логический |
| Зал повышенного комфорта | Логический |
| Приоритетная посадка | Логический |

Атрибуты таблицы:

\* Код тарифа – Уникальный идентификатор тарифа, не может повторяться и является первичным ключом;

\* Код авиалинии – Ключ авиалинии, к которому привязан тариф;

\* Название – Название тарифа;

\* Ручная кладь – Наличие возможности провозки ручного багажа;

\* Багаж – Наличие возможности провозки багажа;

\* Питание – Наличие питания во время полёта;

\* Обмен – Наличие возможности обмена билета;

\* Возврат – Наличие возможности возврата билета;

\* Наценка – Наценка тарифа на цену за рейс;

\* Страховка – Наличие страховки;

\* Выбор места – Наличие возможности выбрать место в самолёте самостоятельно;

\* Зал повышенного комфорта – Наличие возможности провести время ожидания самолёта в зале повышенной комфортности;

\* Приоритетная посадка – Наличие приоритетной посадки в самолёт.

Таблица 4.10 – атрибуты сущности Контактная информация

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| Код информации | Число |
| Адрес электронной почты | Текст |
| Номер телефона | Текст |

Атрибуты таблицы:

\* Код информации – Уникальный идентификатор контактной информации, не может повторяться и является первичным ключом;

\* Адрес электронной почты – Электронная почта, на которую будет зарегистрирован рейс;

\* Номер телефона – Номер телефона, на который будет зарегистрирован рейс.

Таблица 4.11 – атрибуты сущности Регистрация

|  |  |
| --- | --- |
| Имя столбца | Тип данных |
| Код регистрации | Число |
| Код рейса | Число |
| Код бронирования | Текст |
| Код клиента | Число |
| Код контактной информации | Число |
| Код тарифа | Число |
| Место в самолёте | Текст |
| Итоговая цена | Число |
| Зарегистрирован | Логический |

Атрибуты таблицы:

\* Код регистрации – Уникальный идентификатор регистрации, не может повторяться и является первичным ключом;

\* Код рейса – Ключ рейса, привязанный к регистрации;

\* Код бронирования – Ключ бронирования, по которому будет проходить регистрация и просмотр билета;

\* Код клиента – Ключ клиента, привязанный к регистрации;

\* Код контактной информации – Ключ контактной информации, привязанный к регистрации;

\* Код тарифа – Ключ тарифа, привязанный к регистрации;

\* Место – Место пассажира в самолёте;

\* Итоговая цена – Конечная цена с наценкой;

\* Зарегистрирован – Наличие регистрации на рейс клиентом, перед посадкой клиент должен обязательно зарегистрироваться.

5 Формулировка ограничений целостности

Целостность – это актуальность и непротиворечивость информации, соответствие имеющейся информации в базе данных внутренней логике и структуре, а также защищенность от несанкционированных изменений и разрушений.

Ограничение целостности – это некоторое утверждение или правило, которое описывает допустимые значения отдельных информационных единиц и связей между ними. Они делятся на:

\* явные;

\* неявные.

За соответствие ограничениям целостности следит СУБД в процессе своего функционирования. Она проверяет ограничения целостности во всех случаях, когда они могут быть нарушены (например, при редактировании данных).

5.1 Выявление полного перечня ограниченной целостности, присущего данной предметной области

В процессе проектирования для предметной области были выделены следующие ограничения целостности:

\* Код бронирования может повторяться только в том случаем, если при регистрации были выбраны рейсы из точки A в точку B, и обратно из точки B в точку A.

\* Код аэропорта РФ и название рейса не могут повторяться.

\* Номер паспорта может состоять только из цифр.

\* Фамилия, имя и отчество могут состоять только из латинских букв и исключать цифры или специальные символы.

\* Время, проведенное в полете, состоит только из цифр, и имеет тип int.

\* При бронировании билета, дата выдачи паспорта не может быть меньше чем текущая дата, иначе документ не действителен.

\* Номер телефона клиента может содержать только цифры и его длина не может быть меньше и не больше 11 чисел.

\* Фамилия, имя, дата рождения, дата окончания действительности паспорта, номер и серия паспорта, страна выдачи, электронная почта, номер телефона, место в самолёте при регистрации на рейс клиентом не могут быть пустыми.

5.2 Определение перечня ограничений целостности, которые будут контролироваться в данном проекте

В этом проекте будут учитываться ограничения целостности, связанные с ограничением максимальной длины значения, с уникальностью значений и с взаимоисключающими значениями.

5.3 Выбор способа реализация контроля целостности для каждого из ограничений

Все ограничения целостности будут реализованы с использованием стандартных функций СУБД:

\* Ограничения поля по длине достигается выбором типа данных с ограничением максимальной длины;

\* Уникальность значений достигается использованием команды UNIQUE в настройках поля таблицы, либо назначением поля первичным ключом, который не может повторяться;

\* Ситуация, когда 2 поля таблицы взаимосвязаны так, что значение в 1 поле исключает значение во 2 поле, решается использованием команды CHECK, которая будет проверять поля, и при взаимоисключающих значениях не будет заносить изменения в таблицу.

6 Проектирование физической структуры базы данных

**Физическая модель базы данных** — это модель данных, которая определяет, каким образом представляются данные, и содержит все детали, необходимые СУБД для создания базы данных. Физическая структура определяет, тип и свойства данных, которые будут записаны в память компьютера.

Опишем схемы для основных таблиц из логической модели данных. Схема сущности будет содержать полное описание атрибутов-характеристик этой сущности и их свойств на языке целевой СУБД, с учетом связей между сущностями и ограничениями целостности.

Таблица 5.1 - Регистрация:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код регистрации** | **Числовой** | **4** |
| **Код рейса** | **Числовой** | **4** |
| **Код бронирования** | **Текстовый** | **16** |
| **Код клиента** | **Числовой** | **4** |
| **Код контактных данных** | **Числовой** | **4** |
| **Код тарифа** | **Числовой** | **4** |
| Место в самолёте | Текстовый | 8 |
| Конечная цена | Денежный | 8 |
| Зарегистрирован | Логический | 1 |

Таблица 5.2 - Клиент:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код клиента** | **Числовой** | **4** |
| Имя | Текстовый | 40 |
| Фамилия | Текстовый | 40 |
| Отчество | Текстовый | 40 |
| **Код паспорта** | **Числовой** | **4** |

Таблица 5.3 - Тариф:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код тарифа** | **Числовой** | **4** |
| **Код авиалинии** | **Числовой** | **4** |
| Название | Текстовый | 40 |
| Ручной багаж | Логический | 1 |
| Багаж | Логический | 1 |
| Питание | Логический | 1 |
| Обмен | Логический | 1 |
| Возврат | Логический | 1 |
| Наценка | Числовой | 4 |
| Страховка | Логический | 1 |
| Выбор места | Логический | 1 |
| Зал повышенного комфорта | Логический | 1 |
| Приоритет посадки | Логический | 1 |

Таблица 5.4 - Рейс:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код рейса** | **Числовой** | **4** |
| **Код авиалинии** | **Числовой** | **4** |
| Начало рейса | Дата и время | 8 |
| Конец рейса | Дата и время | 8 |
| Цена | Денежный | 8 |
| **Код рейса РФ** | **Текстовый** | **6** |
| Время в полёте | Числовой | 4 |
| Код самолёта | Числовой | 4 |
| Место выхода на посадку | Текстовый | 20 |
| Код аэропорт отбытия | Числовой | 4 |
| Код аэропорт прибытия | Числовой | 4 |

Таблица 5.5 - Контактные данные:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код** | **Числовой** | **4** |
| Электронная почта | Текстовый | 200 |
| Номер телефона | Текстовый | 22 |

Таблица 5.6 - Паспорт:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код** | **Текстовый** | **4** |
| Дата рождения | Дата | 3 |
| **Код гражданства** | **Числовой** | **4** |
| Номер и серия | Текстовый | 20 |
| **Код страны выдачи** | **Числовой** | **4** |
| Дата окончания действительности | Дата | 3 |

Таблица 5.7 - Город:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код города** | **Числовой** | **4** |
| Название | Текстовый | 40 |
| **Код страны** | **Числовой** | **4** |

Таблица 5.8 - Страна:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код страны** | **Числовой** | **4** |
| Название | Текстовый | 40 |

Таблица 5.9 - Аэропорт:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код аэропорта** | **Числовой** | **4** |
| Код РФ | Текстовый | 6 |
| Название | Текстовый | 40 |
| **Код города** | **Числовой** | **4** |

Таблица 5.10 - Авиалинии:

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код авиалинии** | **Числовой** | **4** |
| Название | Текстовый | 50 |
| Каринка | Текстовый | 400 |

Таблица 5.11 - Самолёт

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Макс. вес в байтах |
| **Код самолёта** | **Числовой** | **4** |
| Модель | Текстовый | 40 |
| Бизнес места | Числовой | 4 |
| Эконом места | Числовой | 4 |
| Картинка мест | Текстовый | 400 |
| Описание | Текстовый | 8000 |

7 Организация ввода данных в БД

Работа с базой данных будет проходит через страницу сайт, которая будет бронировать клиента на рейс. Пример ввода данных в базу:

На (Рис. 7.1 – Таблица паспортных данных до добавления) представлена таблица паспорта, открытая в СУБД. Запомним, что она включает 4 элемента.

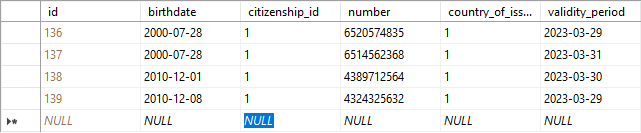


Рис 7.1 – Таблица паспортных данных до добавления

Далее представлена таблица (Рис 7.2 – Таблица контактных данных до добавления) контактных данных, которая включает в себя также 4 элемента.

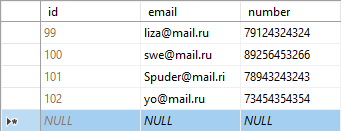


Рис 7.2 – Таблица контактных данных до добавления

Рассмотрим таблицу (Рис 7.3 – Таблица клиента до добавления данных) клиента, которая включает в себя 2 элемента.



Рис 7.3 – Таблица клиента до добавления данных

Данные с вышеуказанных трёх таблиц привязаны к таблице Регистрация (Рис. 7.4 Таблица регистрации до добавления данных), которая приведена ниже.

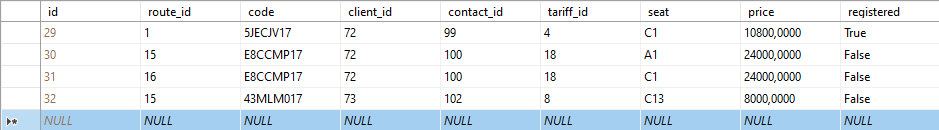


Рис. 7.4 Таблица регистрации до добавления данных

Перейдем на страницу сайта (Рис 7.5 – Бронирование билета) бронирования билета на рейс.

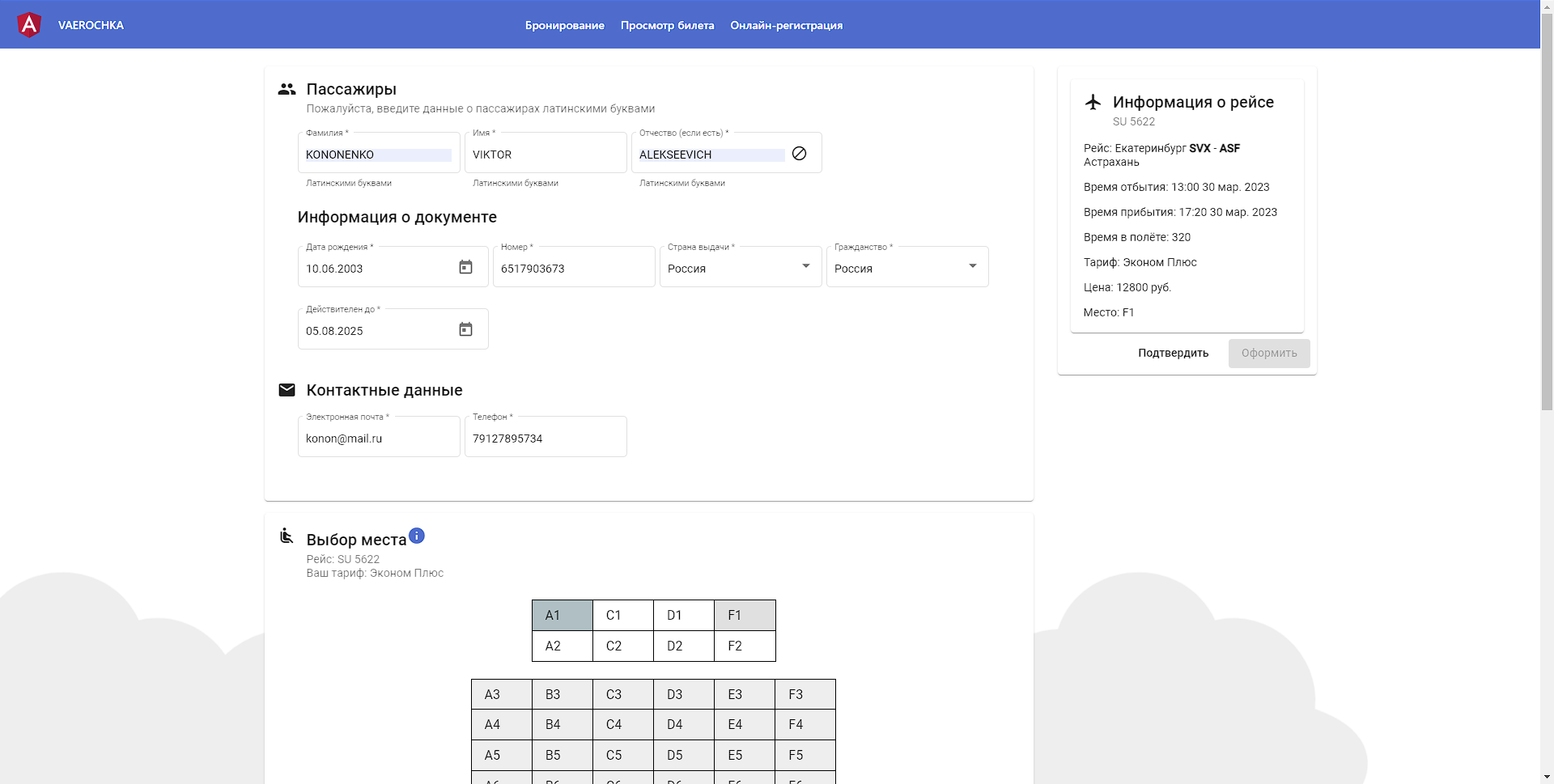


Рис 7.5 – Бронирование билета

После бронирования билета данные в таблице контактной информации (Рис 7.5 – Таблица контактных данных после добавления).

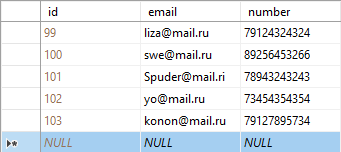


Рис 7.5 – Таблица контактных данных после добавления)

В таблице паспортных данных (Рис 7.6 – Таблица паспортных данных после добавления).

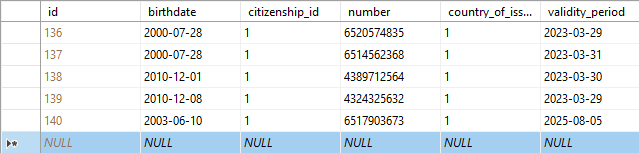


Рис 7.6 – Таблица паспортных данных

В таблице Клиент (Рис 7.7 – Таблица Клиент после добавления).



Рис 7.7 – Таблица Клиент после добавления

И так же в таблицу Регистрация добавились данные (Рис 7.8 – Таблица Регистрация).

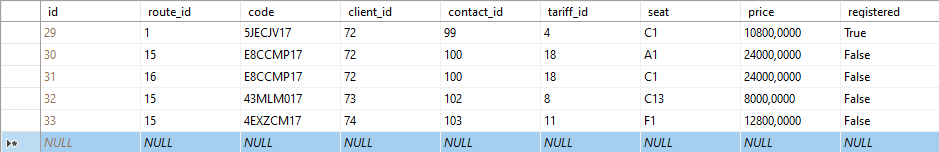


Рис 7.8 – Таблица Регистрация

8 Организация корректировки БД

Корректировка данных происходит через сайт, ниже это продемонстрировано на примере регистрации на рейс (Рис 8.1 – Таблица Регистрация до обновления). Логическое поле «regitered» в строке с кодом регистрации 33 имеет значение «false».

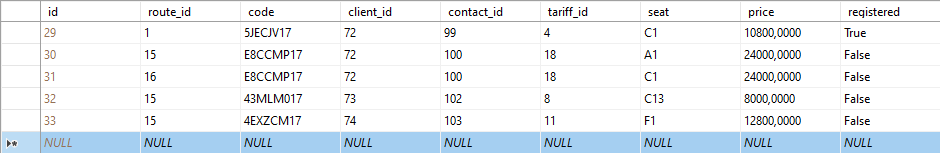


Рис 8.1 – Таблица Регистрация до обновления

После того как мы попадем на страницу регистрации на рейс (Рис 8.2 – Страница регистрации после регистрации) и зарегистрируемся.

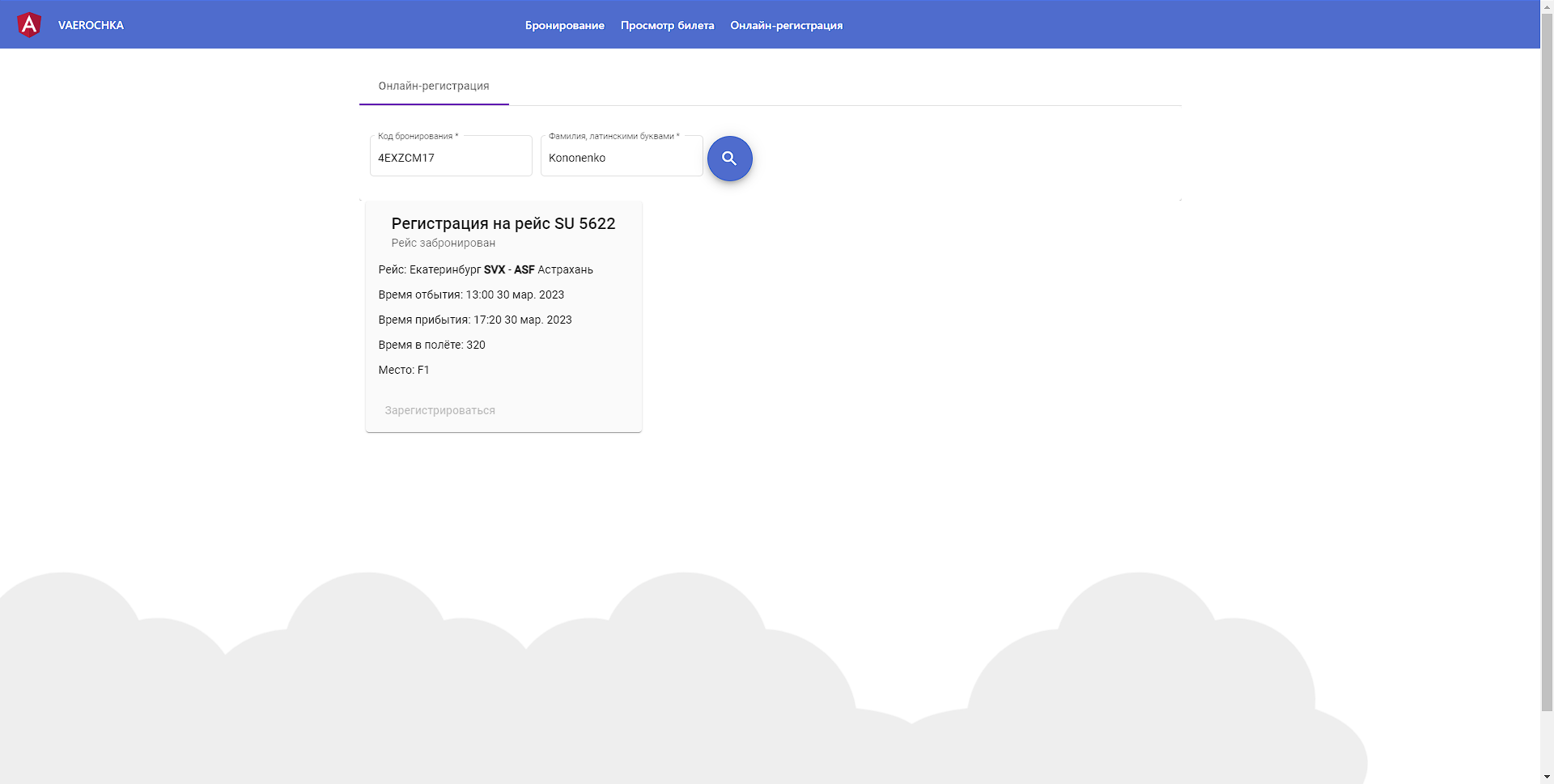


Рис 8.2 – Страница регистрации после регистрации

То данные в таблице Регистрация (Рис. 8.3 – Таблица Регистрация после обновления) так же поменяются, и теперь клиент зарегистрирован на рейс.

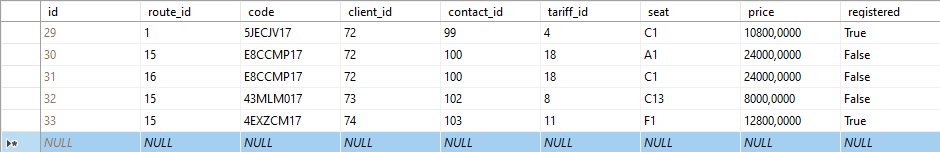


Рис. 8.3 – Таблица Регистрация после обновления

9 Реализация запросов пользователя, получение отчётов

Запросы пользователей также осуществляются через сайт. В качестве первого примера рассмотрим запрос от пользователя на имеющиеся рейсы по заданным данным, такие как: город прибытия, отбытия и датой отбытия (Рис 9.1 – Страница поиска рейса).

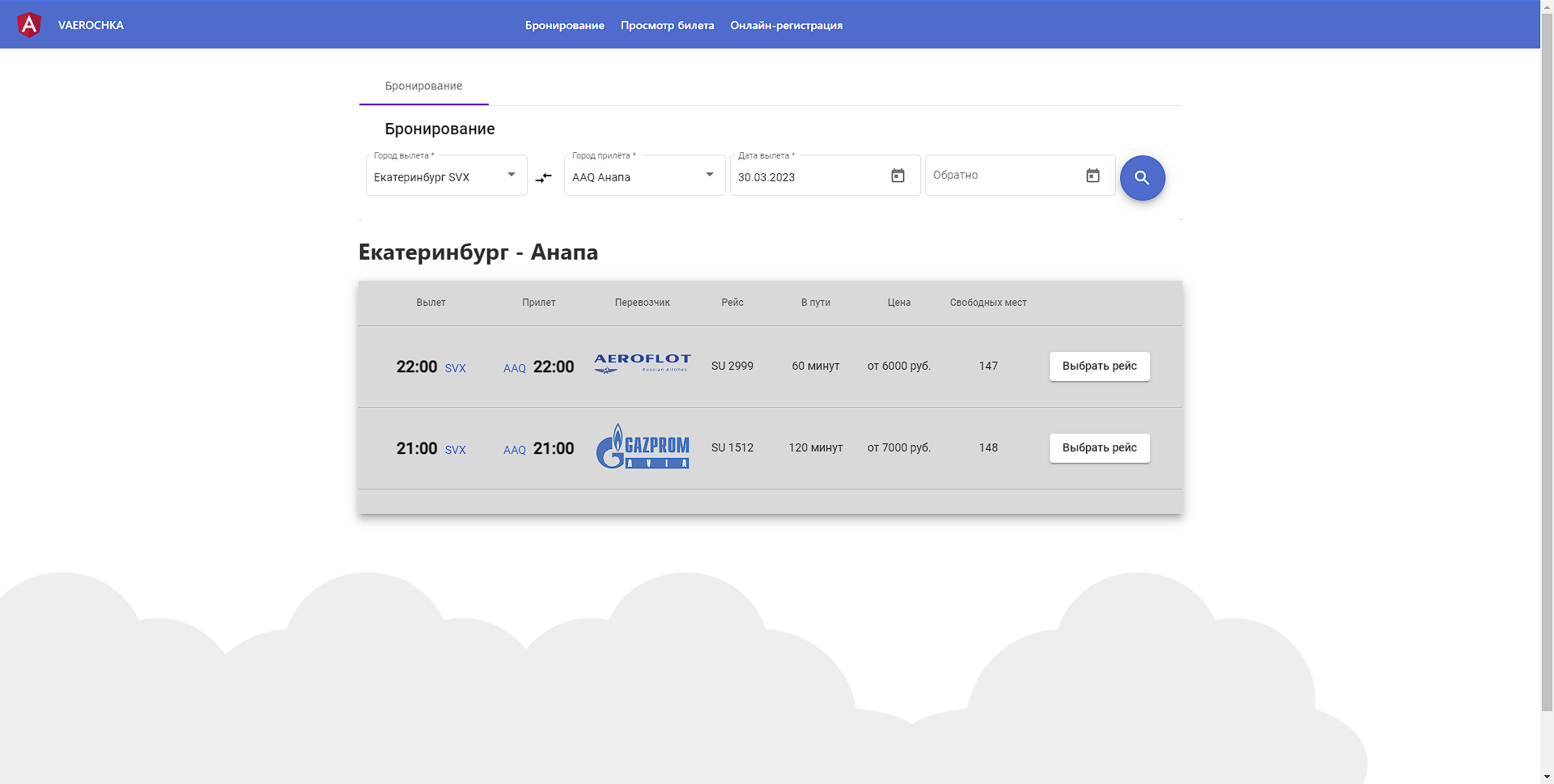


Рис 9.1 – Страница поиска рейса

Второй запрос похож на прошлый, только к заданным данным добавляется ещё одна дата отбытия, будут выведены дополнительные рейсы обратно в первоначальный город отбытия по заданной дате (Рис 9.2 – Страница поиска рейсов).

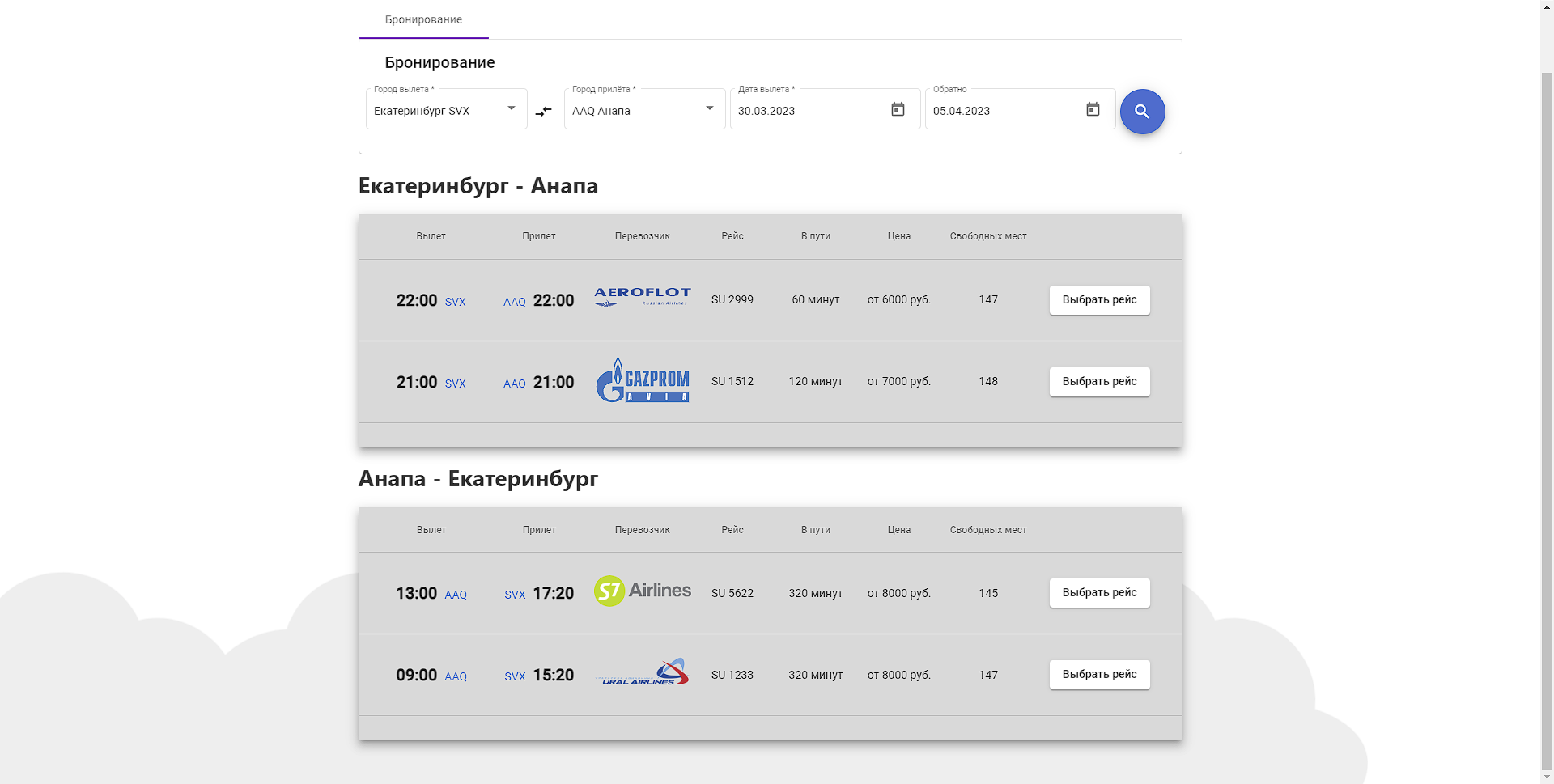


Рис 9.2 – Страница поиска рейсов

Третий запрос на тарифы, предложенные от авиалинии на рейс (Рис 9.3 – Страница выбора рейса).

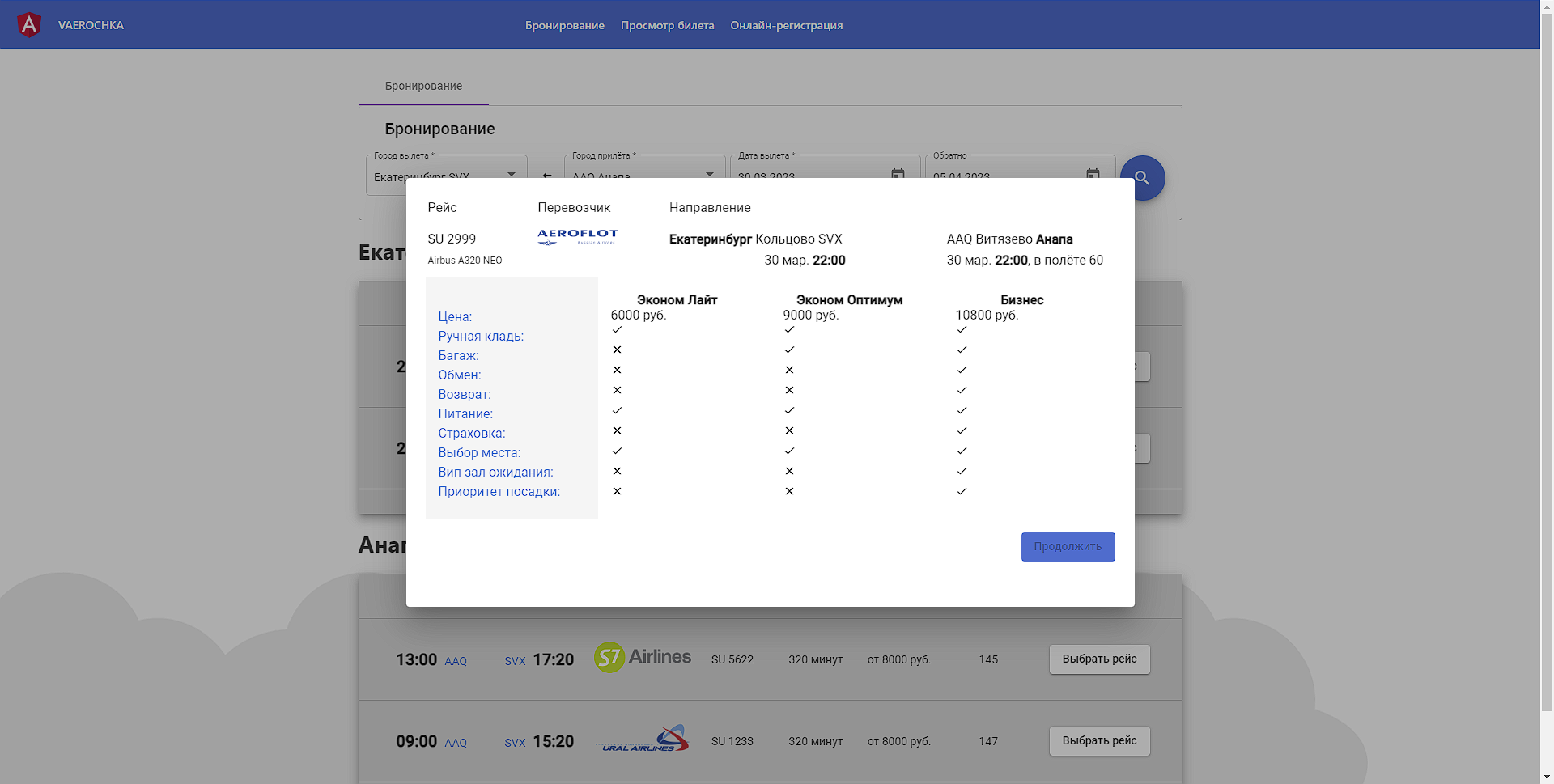


Рис 9.3 – Страница выбора рейса

Четвёртый запрос на информацию о рейсе, по коду бронирования и фамилии выведется таблица с информацией о рейсе (Рис. 9.4 – Страница с информацией на зарегистрированный рейс).

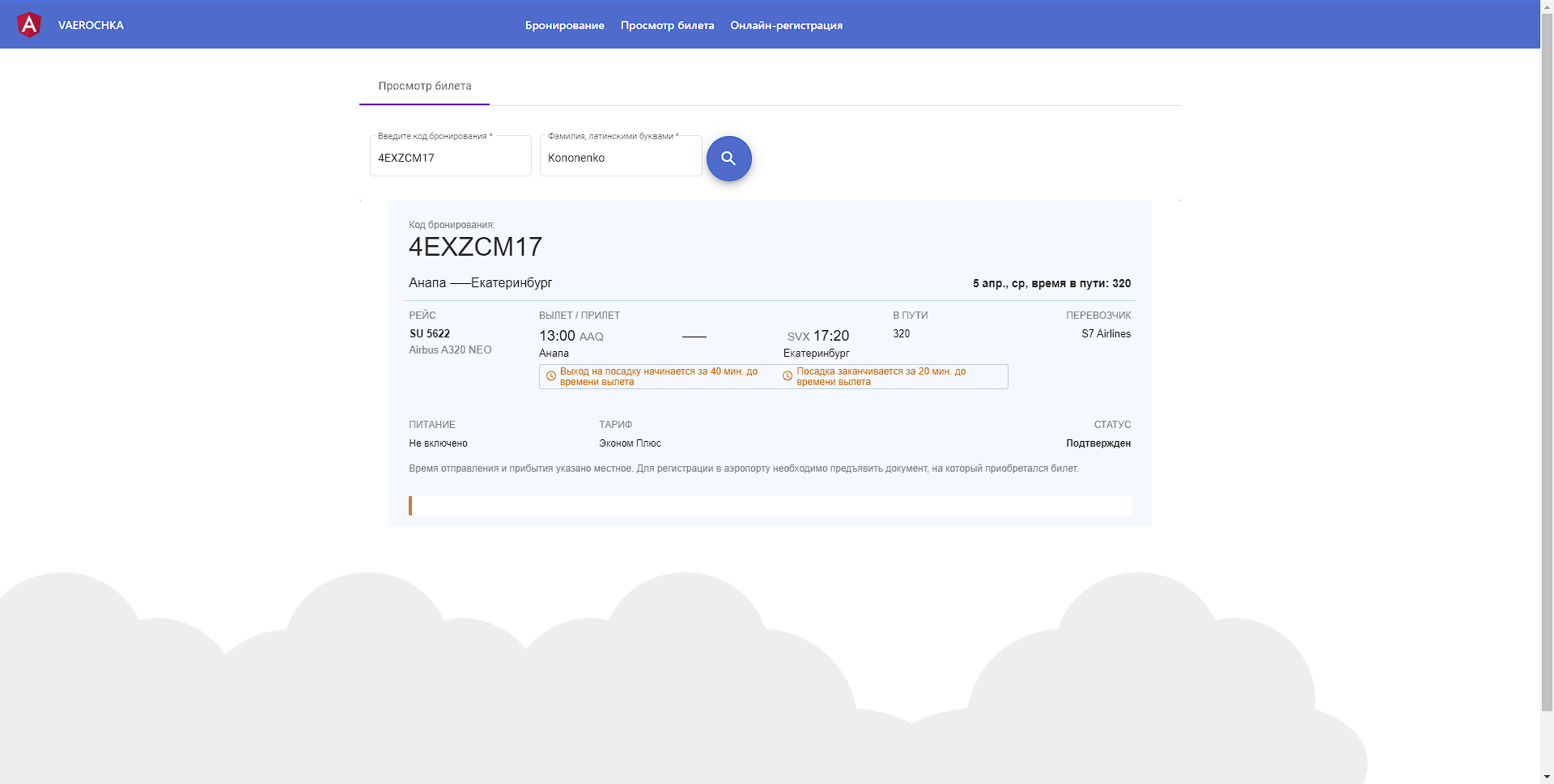


Рис. 9.4 – Страница с информацией на зарегистрированный рейс

**10 Разработка пользовательского интерфейса**

Пользовательским интерфейсом служит сайт, написанный для работы с базой данных, описанной в разделах выше. Первоначальной страницей (Рис 10.1 – Страница с рейсами) является страница, на которой пользователь, введя предложенные данные получит подходящие рейсы и данные о них.

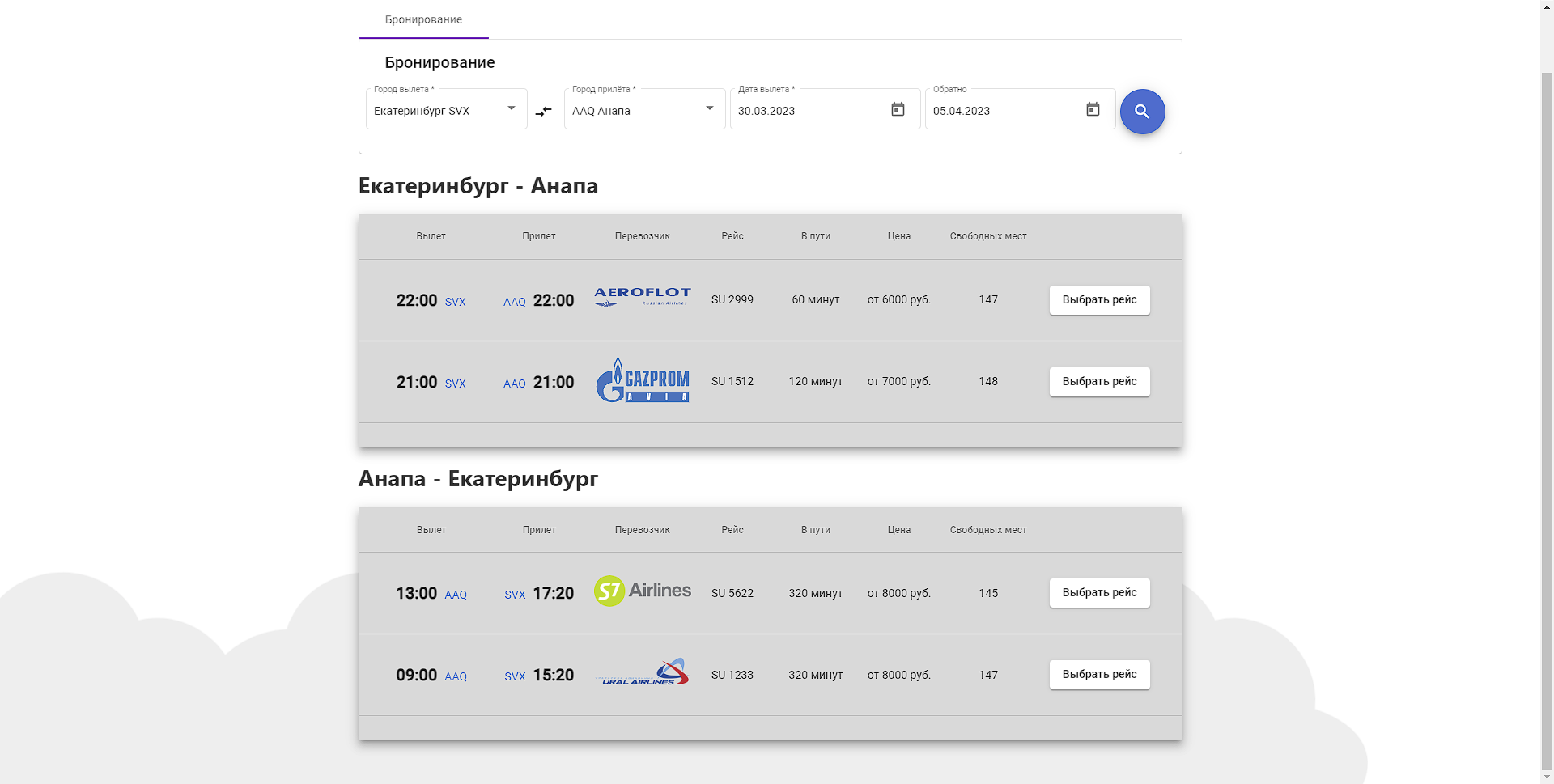


Рис 10.1 – Страница с рейсами

После того как вывелись подходящие рейсы клиент выбирает нужный ему и попадает на модальное окно выбора тарифа (Рис. 10.2 - Модальное окно с тарифами).

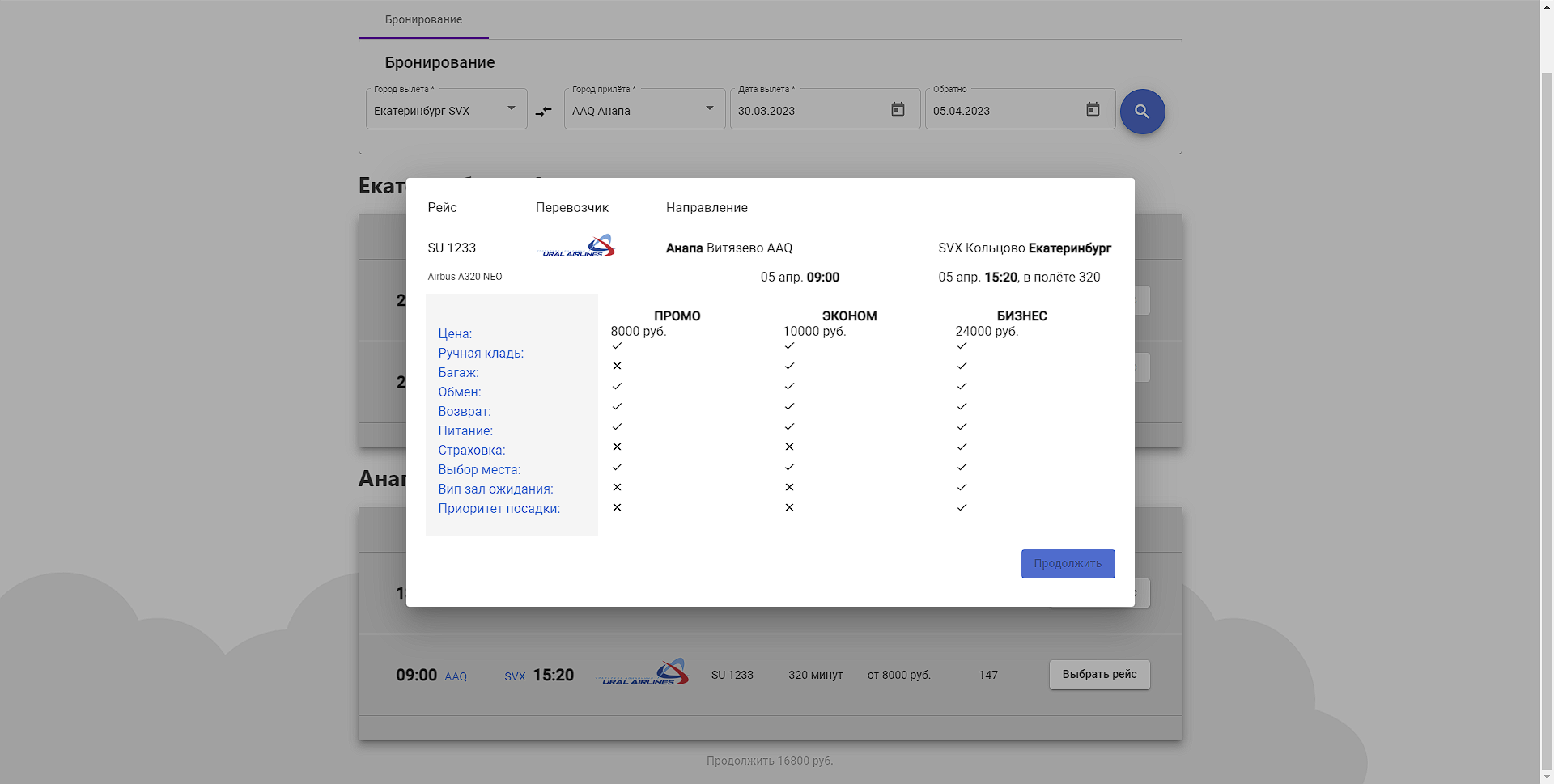


Рис. 10.2 - Модальное окно с тарифами

После выбора тарифов и нажав на кнопку продолжить, клиент попадает на страницу регистрации (Рис. 10.3 – Страница регистрации), где ему предложено вписать личный данные, обязательные для регистрации на рейс, и выбрать места в самолёте на для каждого рейса.

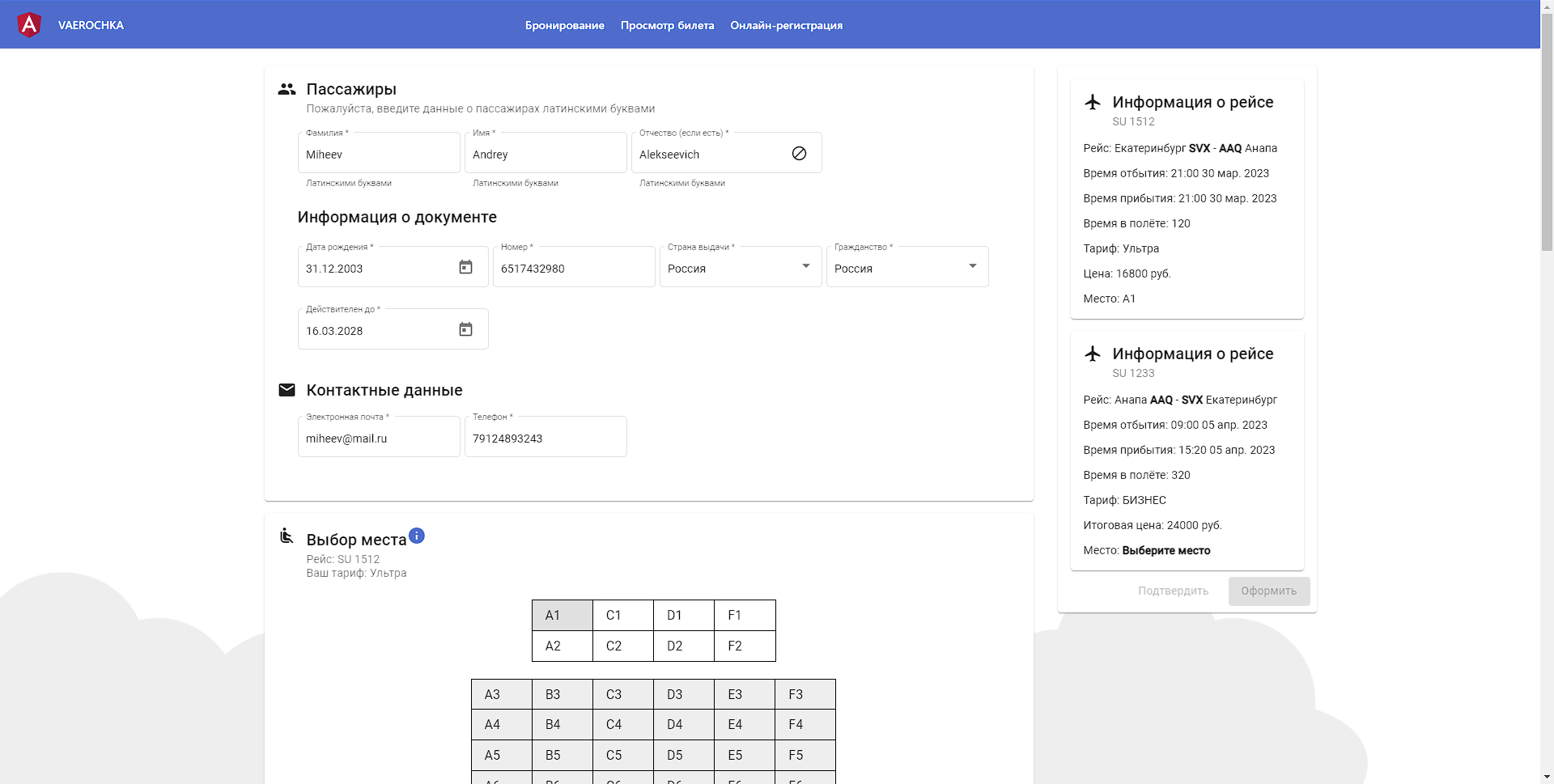


Рис. 10.3 – Страница регистрации

Также, если схема с выбором места клиенту не понятна, он может открыть модальное окно (Рис. 10.4 – Модальное окно с местами в самолёте), в котором показана картинка с местами в самолёте.

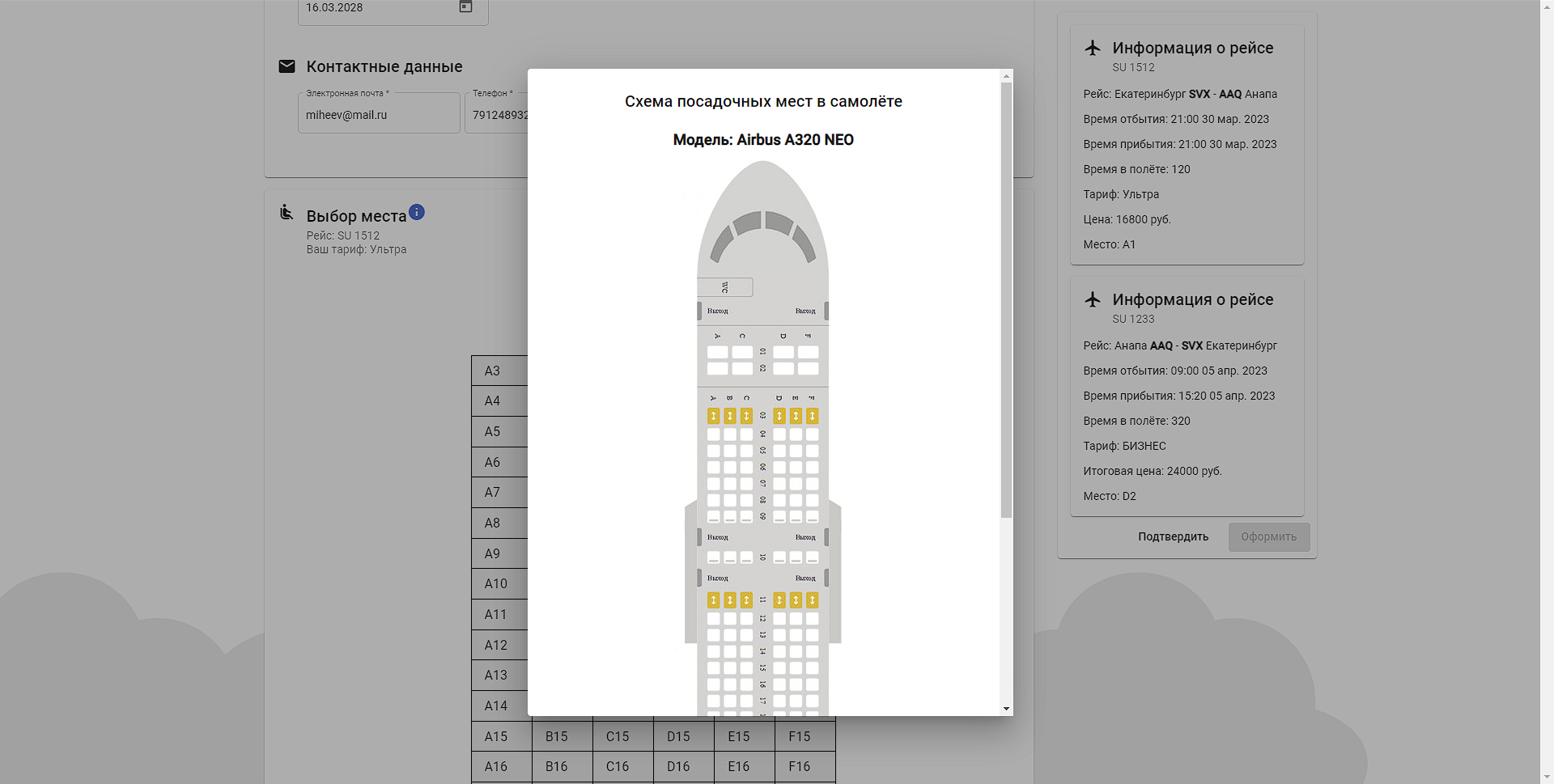


Рис. 10.4 – Модальное окно с местами в самолёте

После того, как клиент введет все данные, и выберет места в самолёте, ему нужно подтвердить данные, и нажать кнопку оформить, далее выведется модальное окно с оплатой (Рис 10.5 – Модальное окно оплаты), на нем предложено ввести данные карточки, по которой будет произведена оплата, подтвердить и оплатить. Данная процедура ничего не имеет общего с настоящей оплатой.

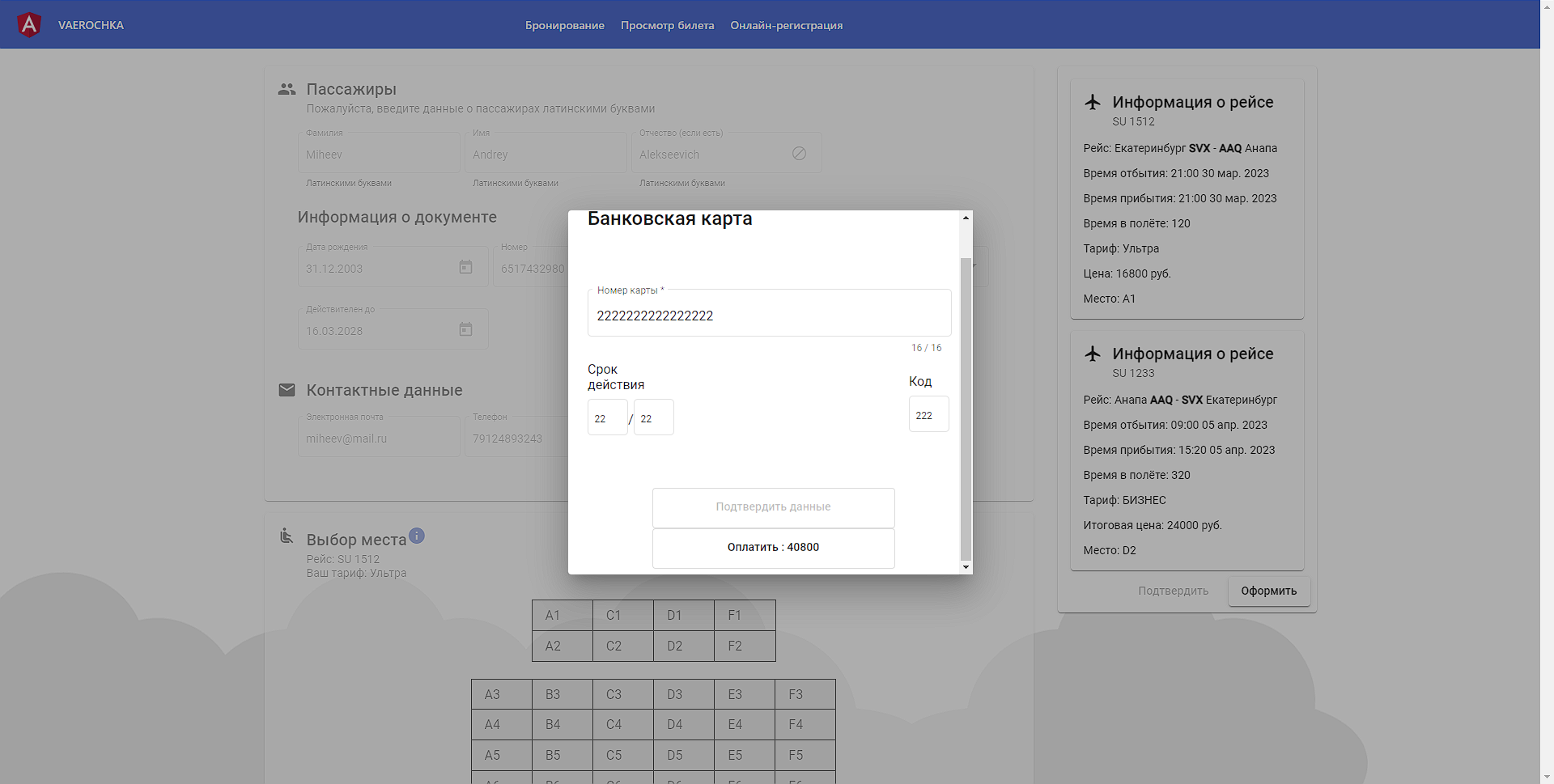


Рис 10.5 – Модальное окно оплаты

Далее, после оплаты, выведется страница с билетами на рейс (Рис 10.6 – Страница билета после оплаты), в которых указана подробная информация.

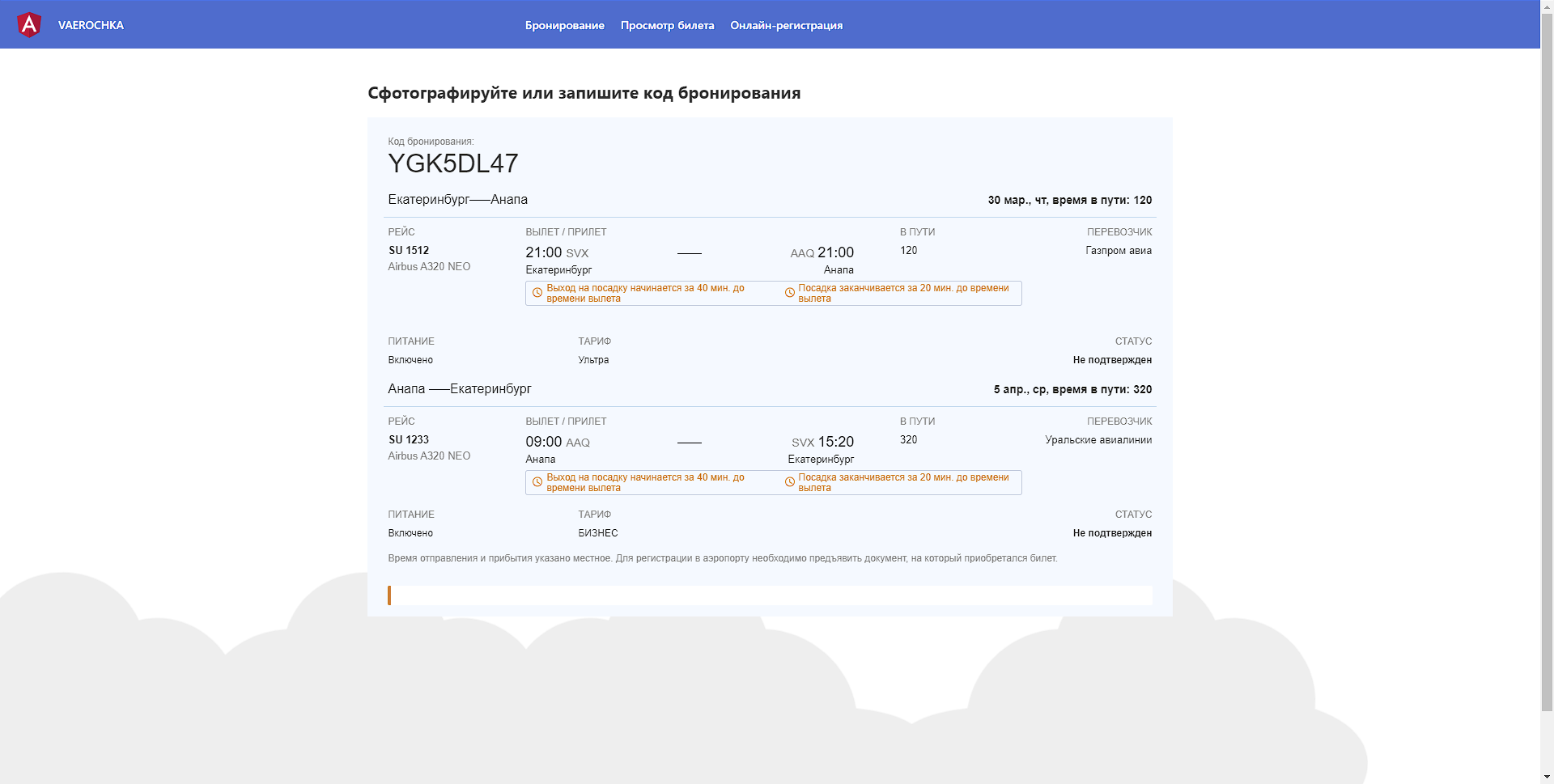


Рис 10.6 – Страница билета после оплаты

Так же на странице выше выведен код бронирования, по нему клиент может посмотреть подробную информацию о рейсах, или же произвести онлайн регистрацию на сайте.

Если клиенту нужно посмотреть информацию о рейсах повторно, то он должен перейти на вкладку просмотра билета, которая расположена в шапке сайта, и после того как он вписал код бронирования, фамилию, на которую была осуществлена регистрация рейса, то при нажатии поиск выведется подробная информация о рейсах (Рис 10.7 – Страница с подробной информации о рейсе).

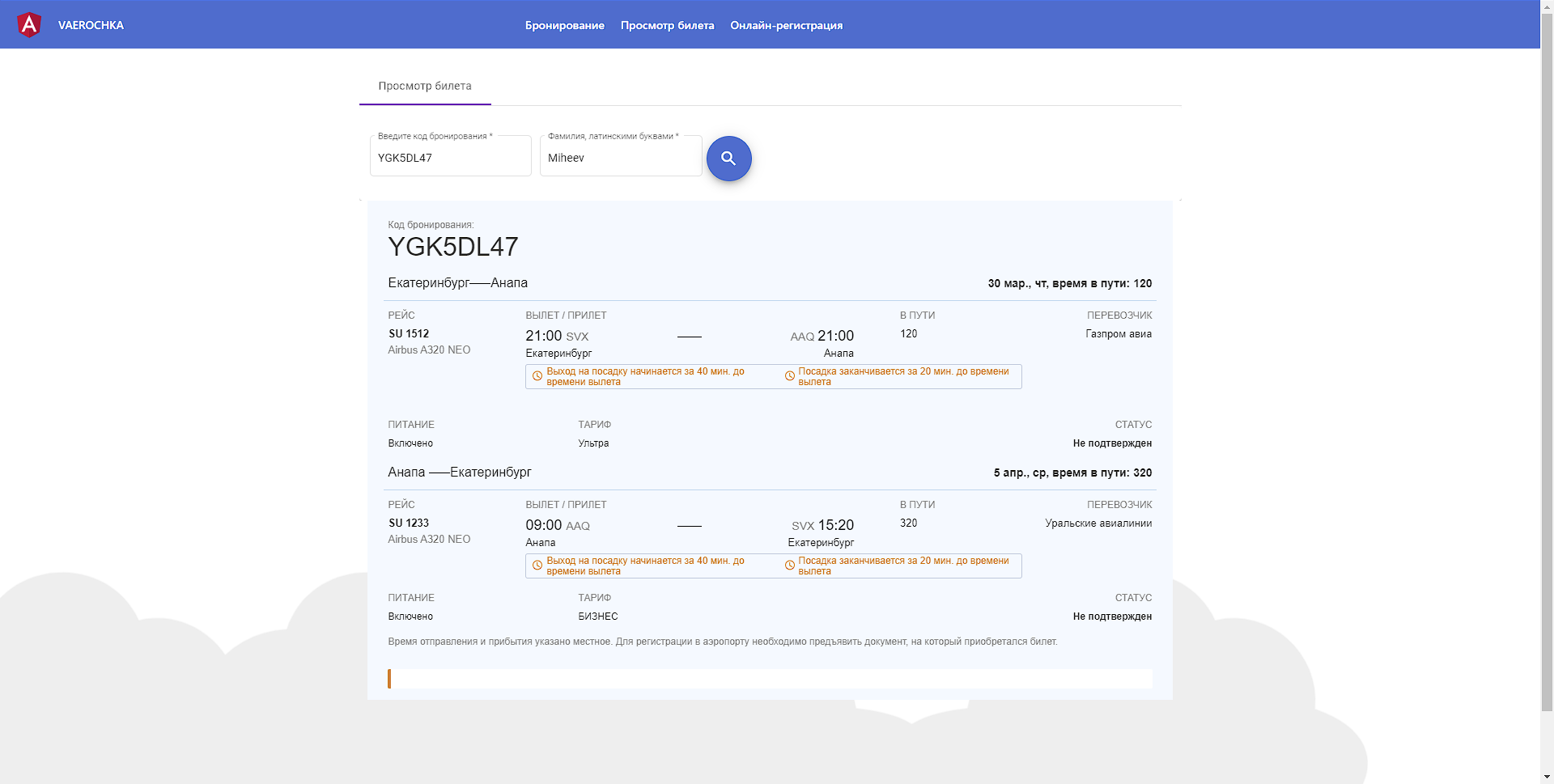


Рис 10.7 – Страница с подробной информации о рейсе

Для того, чтобы осуществить регистрацию онлайн на сайте, нужно перейти на вкладку Онлайн-регистрация в шапке профиля, продолжится ввести так же код бронирования и фамилию, после отправки которых, будут выведены рейсы (Рис. 10.8 – Страница онлайн-регистрации), на которые клиент может зарегистрироваться. Регистрация может быть не доступна, т.к. она начинается за 24 часа до начала рейса, или же клиент уже зарегистрирован.

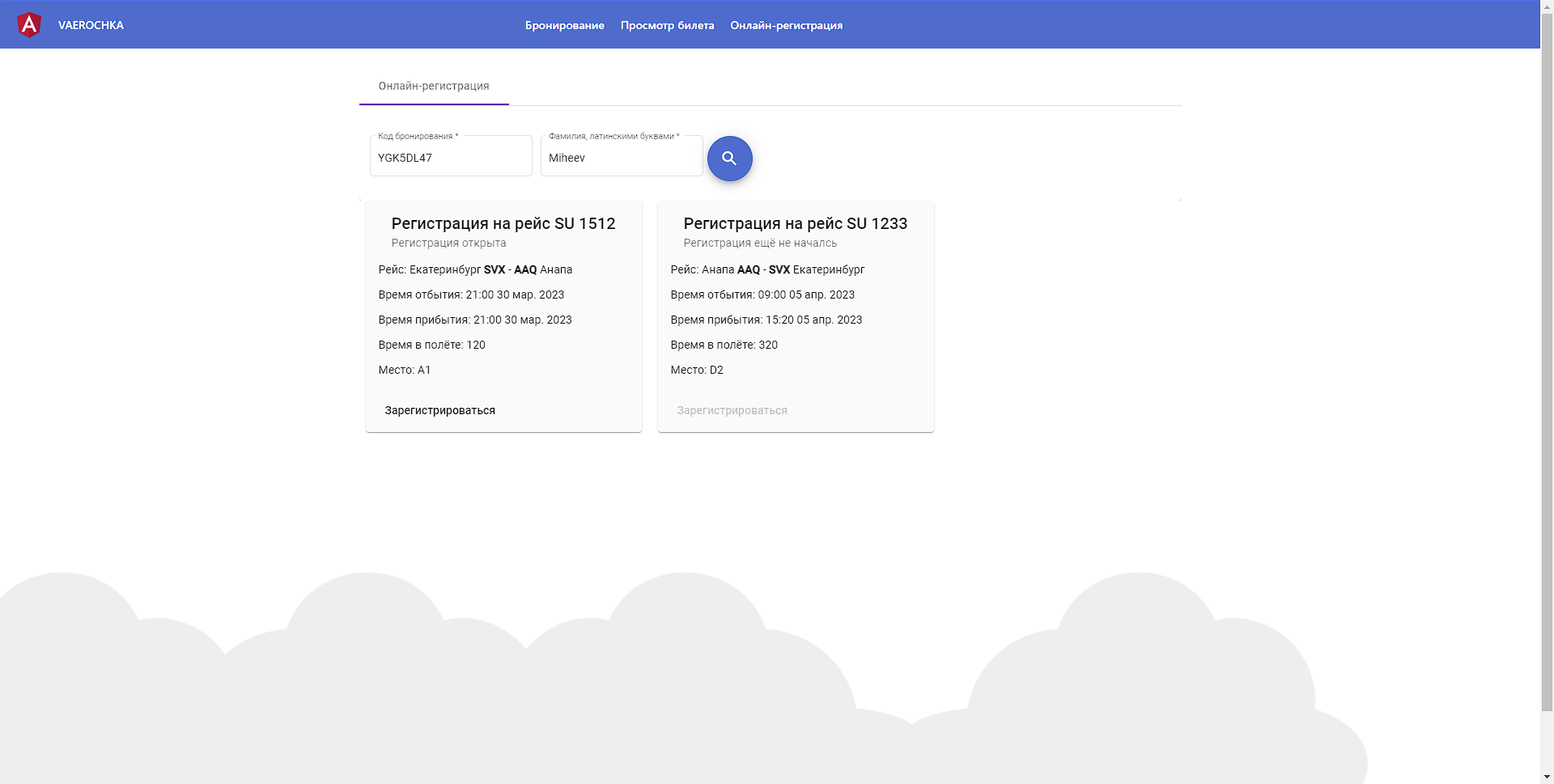


Рис. 10.8 – Страница онлайн-регистрации

11 Реализация проекта в среде конкретной СУБД

Реализация проекта разработки базы данных в среде СУБД MS SQL является многоступенчатым трудоемким процессом, основные этапы которого были подробно рассмотрены выше. Был реализован контроль целостности для каждого из ограничений, выявленных нами ранее. С помощью SQL были построены все необходимые для удовлетворения информационных потребностей пользователей БД запросы.

База данных была написана через интерфейс MS SQL Management Studio. Он представляет собой набор удобных инструментов для работы с SQL сервером, также позволяя создавать базы данных (Рис 11.1 - Окно создания базы данных).

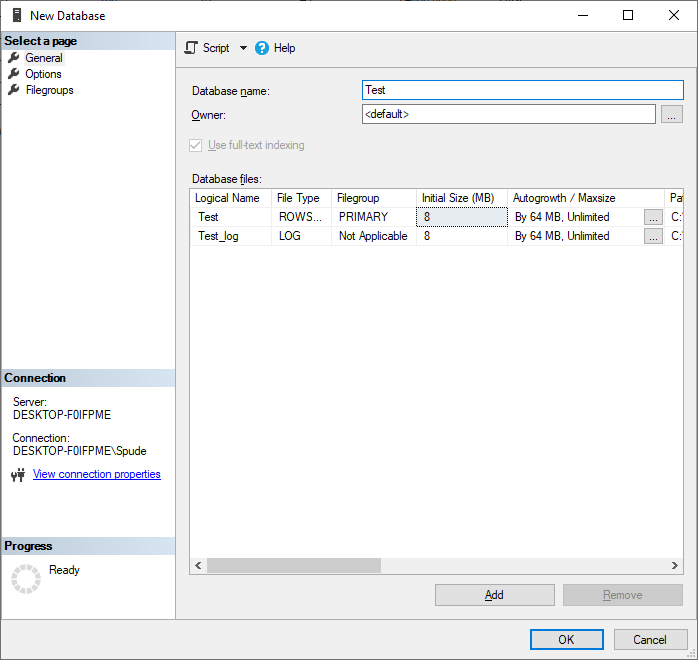


Рис 11.1 - Окно создания базы данных

После создания базы данных MS SQL Management Studio позволяет создавать таблицы (Рис 11.2 – Форма создания таблицы), в которых можно вписать наименования столбцов, их тип, и может ли значение быть пустым.

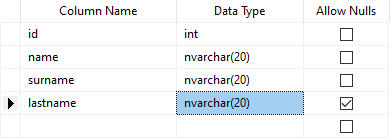


Рис 11.2 – Форма создания таблицы

Так же MS SQL Management Studio имеет возможность редактирования значений в таблице (Рис 11.3 – Окно редактирования таблицы).

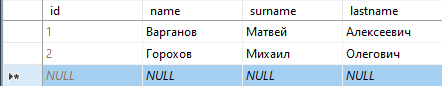


Рис 11.3 – Окно редактирования таблицы).

Следующим важнейшим этапом реализации проекта была разработка пользовательского интерфейса, сайта. Были созданы все необходимые страницы для работы с базой данных.

Условные сокращения

Код РФ – это трехбуквенный код, основанный на названии аэропорта.

Заключение

В ходе работы над данным проектом была изучена предметная область. В процессе анализа которой была составлена ER модель, вошедшая в основу спроектированной логической структуры базы данных. Исследовав популярные СУБД была выбрана одна из лучших, в которой реализована логическая модель базы. Для работы с базой данных разработан сайт, позволяющий с ней работать.

Проект несомненно можно улучшить в будущем. Можно расширить систему для бронирования не только билетов, но и железнодорожных поездок.

Разработанный проект намного облегчает процесс бронирования авиабилетов, как для клиентов, так и для сотрудников, ускорив процесс подбора рейса и регистрации, а также просмотра информации о забронированных рейсах.

Список литературы

\* Шилдс У. «SQL: быстрое погружение» 2022 С. 410.

\* Введение в Angular (<https://metanit.com/web/angular2/1.1.php>)

\* Сборка Angular приложений и другие ресурсы (<https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Angular_building>)

\* Введение в Web API (<https://metanit.com/sharp/aspnet5/23.1.php>)

**ПРИЛОЖЕНИЯ**

Приложение А Основные таблицы БД