МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

**КУРСОВАЯ РАБОТА**

по дисциплине БАЗЫ ДАННЫХ

на тему: Разработка фрагмента информационной системы

Ж/Дорожный вокзал

Студентки 3 курса АВТ-713 группы

Тягунова Виктория Валерьевна

Руководитель доцент

Астапчук В. А.

Новосибирск 2019

# **Реферат**

# 26 стр., 31 рис., 2 ист., 1 прилож.

Конечный пользователь, программа, информационная система, железнодорожный вокзал

Рассмотрен и описан процесс разработки программного обеспечения для создания отчетности. В качестве средства разработки выбрана среда программирования PyCharm, программное обеспечение для автоматизации развёртывания и управления приложениями Docker и фреймворк для веб-приложений на языке Python Django. Сформирован состав задач, решаемых системой. Разработан технологический процесс функционирования системы. Определены технические задачи. Описана структура программного обеспечения. Проведена разработка конфигурации, кодирование основных алгоритмов, отладка проектируемой системы. Описана программная реализация, пользовательский интерфейс и возможности расширения функций программы.

Содержание

[**Введение** 4](#_Toc28047113)

[**1 Формулировка задания** 5](#_Toc28047114)

[**2 Варианты использования** 6](#_Toc28047115)

[**3 ER-диаграмма** 9](#_Toc28047116)

[**4 Физическая схема** 10](#_Toc28047117)

[**5 Логическая структура данных** 11](#_Toc28047118)

[**6 Формы представления данных** 12](#_Toc28047119)

[**7 Формы выходных документов** 21](#_Toc28047120)

[**8 Пример запроса** 23](#_Toc28047122)

[**Заключение** 24](#_Toc28047123)

[**Приложение А** 25](#_Toc28047124)

[**Руководство пользователя** 25](#_Toc28047125)

[**Список литературы** 26](#_Toc28047126)

# **Введение**

Все чаще приходится решать задачи, связанные со сбором, обработкой, хранением больших объемов информации. Производить эти действия без использования специальных инструментов становится невозможно. Поэтому, если для решения наших задач нам необходимы знания об однотипных объектах или повторяющихся явлениях, нам стоит использовать базу данных. База данных (БД) – это структурированные знания об объектах.

БД помогает систематизировать и хранить информацию из определенной предметной области, облегчает доступ к данным, поиск и предоставление необходимых сведений.

В рамках работы будет разработана система «Железнодорожный вокзал», охватывающая наиболее общие функции работы. Для возможного дальнейшего развития программы и использования её на практике в случае необходимости нужно будет добавить некоторые функции в программу, дополнив существующую иерархическую структуру, либо даже реорганизовать имеющуюся.

Основной целью данной работы будет получение практических навыков проектирования БД и реализации ее под управлением СУБД MS SQL Server. Также, получение опыта управления данными БД, используя язык SQL.

# **1 Формулировка задания**

База данных должна содержать сведения о следующих объектах:

Поезда – номер поезда, станция назначения, станция отправления, категория, расписание (дата и время отправления, количество билетов по категориям - плацкартный, купейный, св, поездная бригада (фамилия, должность)), маршрут (промежуточная станция, дата и время прибытия, стоянка), задержка.

Выходные документы:

Справка о станциях пересадок по маршруту до указанной станции.  
Справка о наличии билетов на указанный поезд.  
Справка о поездных бригадах.

Бизнес-правила:

1) Поезда могут быть транзитными;

2) Сведения о поездной бригаде заполняются только для поездов, формируемых на местной станции.

# **2 Варианты использования**

В данном приложении администратор обладает всеми правами:

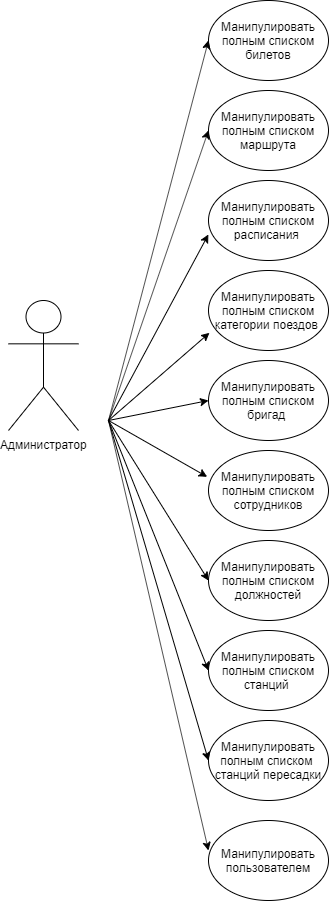
****

Рисунок. 1. Диаграмма вариантов использования системы «Железнодорожный вокзал» администратором

Начальник станции имеет следующие права:

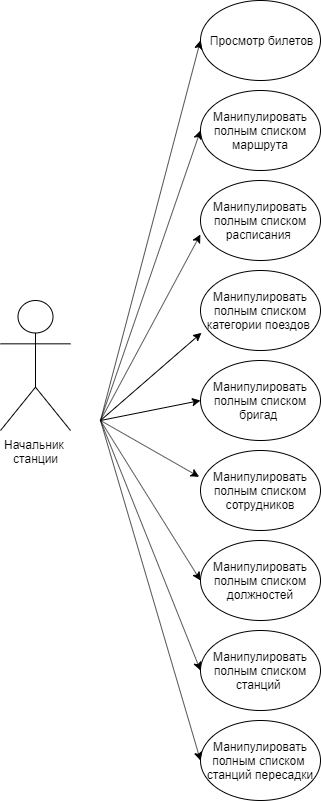


Рисунок. 2. Диаграмма вариантов использования системы «Железнодорожный вокзал» начальником станции

Менеджер обладает уже куда более скромным набором возможностей:

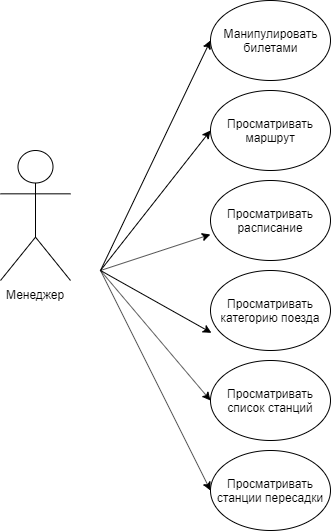


Рисунок. 3. Диаграмма вариантов использования системы «Железнодорожный вокзал» менеджером

# 

# **3 ER-диаграмма**

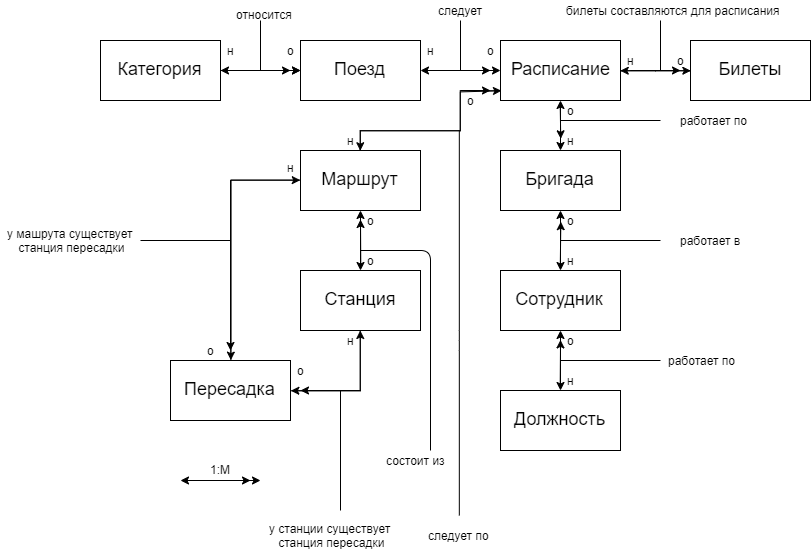


Рисунок 4. Концептуальная модель данных системы

# **4 Физическая схема**

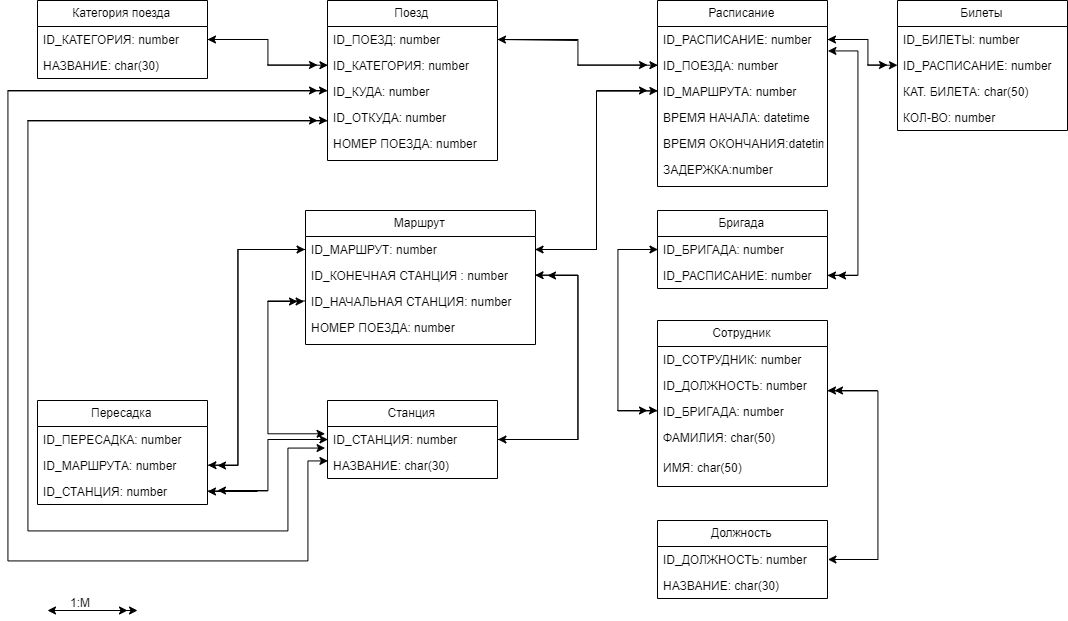


Рисунок 5. Физическая модель данных системы для СУБД

# **5 Логическая структура данных**

SELECT name FROM app\_station AS s

INNER JOIN app\_route\_transfer AS tr ON tr.station\_id = s.id

INNER JOIN app\_route AS r ON r.id = tr.route\_id

WHERE r.start\_station\_id = 1 AND r.finish\_station\_id = 2

SELECT SUM(t.count) FROM app\_ticket AS t

INNER JOIN app\_timetable AS tt ON tt.id = t.timetable\_id

INNER JOIN app\_train AS tr ON tr.id = tt.train\_id

WHERE tr.id = 1

SELECT \* FROM app\_train AS tr

INNER JOIN app\_timetable AS tt ON tt.train\_id = tr.id

INNER JOIN app\_route AS r ON r.id = tt.route\_id

INNER JOIN app\_station AS s\_finish ON s\_finish.id = r.finish\_station\_id

INNER JOIN app\_station AS s\_start ON s\_start.id = r.start\_station\_id

WHERE s\_start.id = 1 AND s\_finish.id = 2 AND tt.start\_time >= (now() - interval '2 week') AND tt.end\_time <= now()

# **6 Формы представления данных**

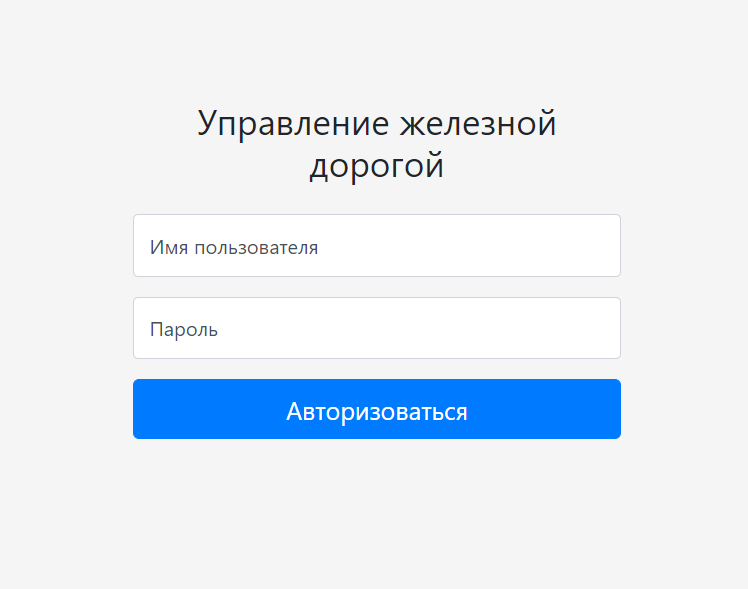


Рисунок 6. Вход в систему

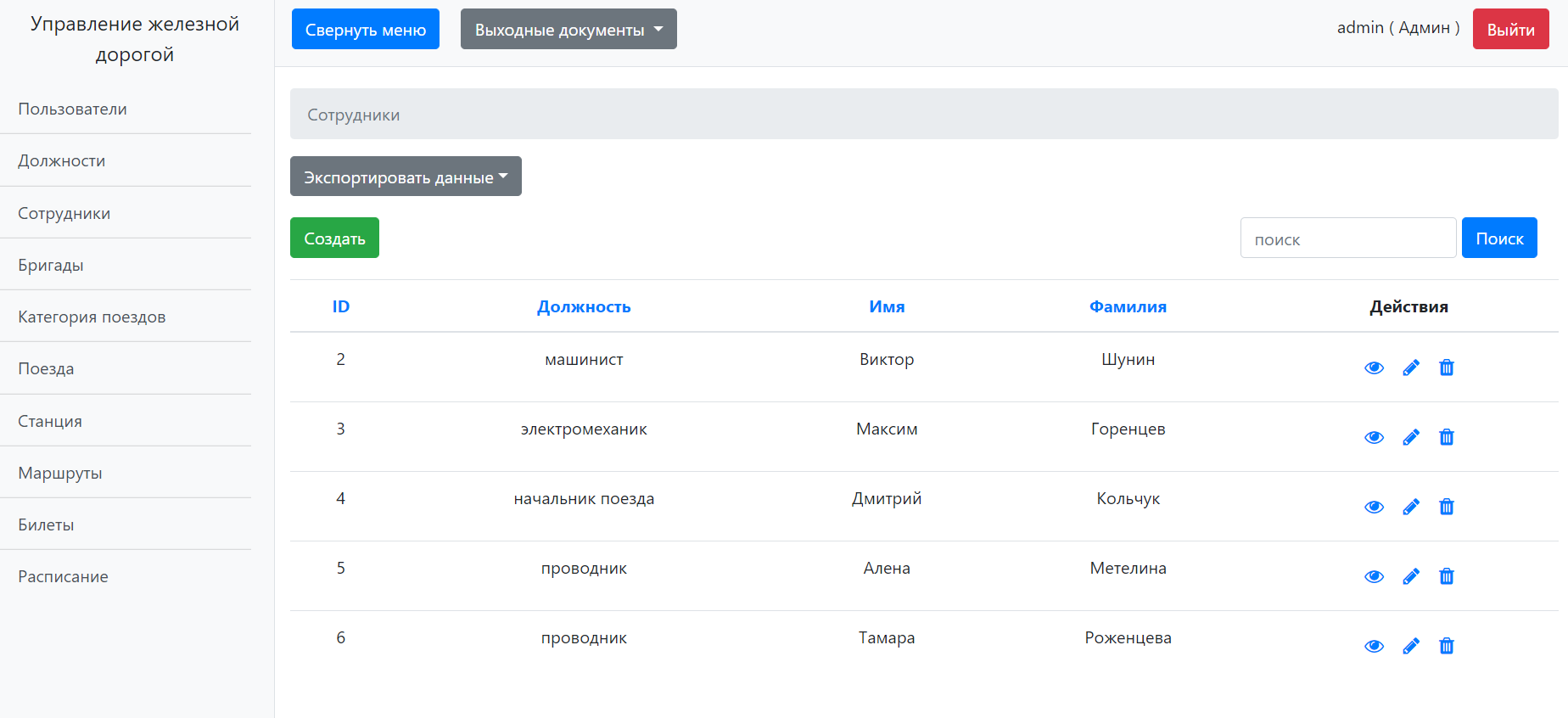


Рисунок 7. Страница для просмотра пользователей

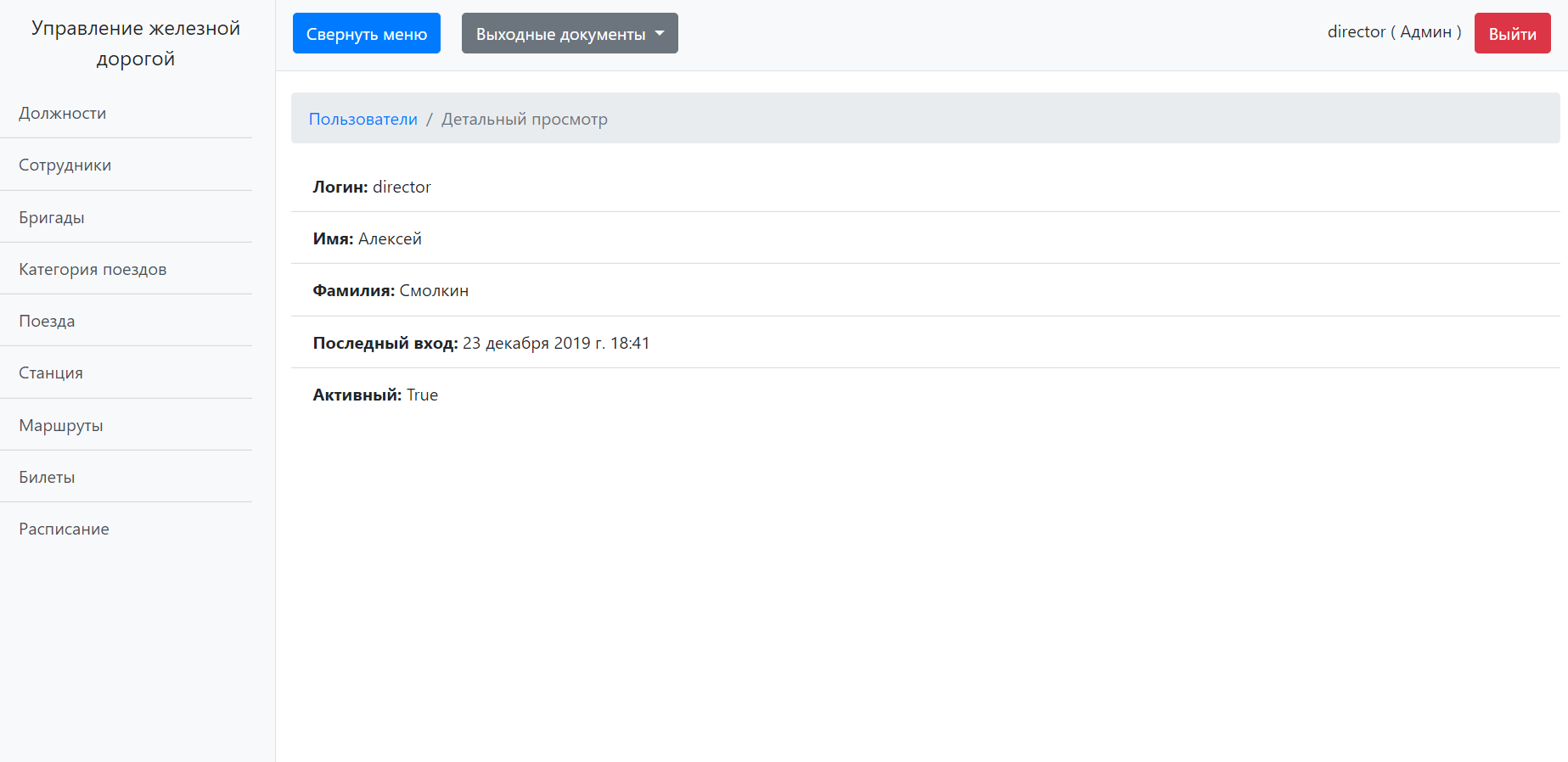


Рисунок 8. Страница для просмотра пользователя

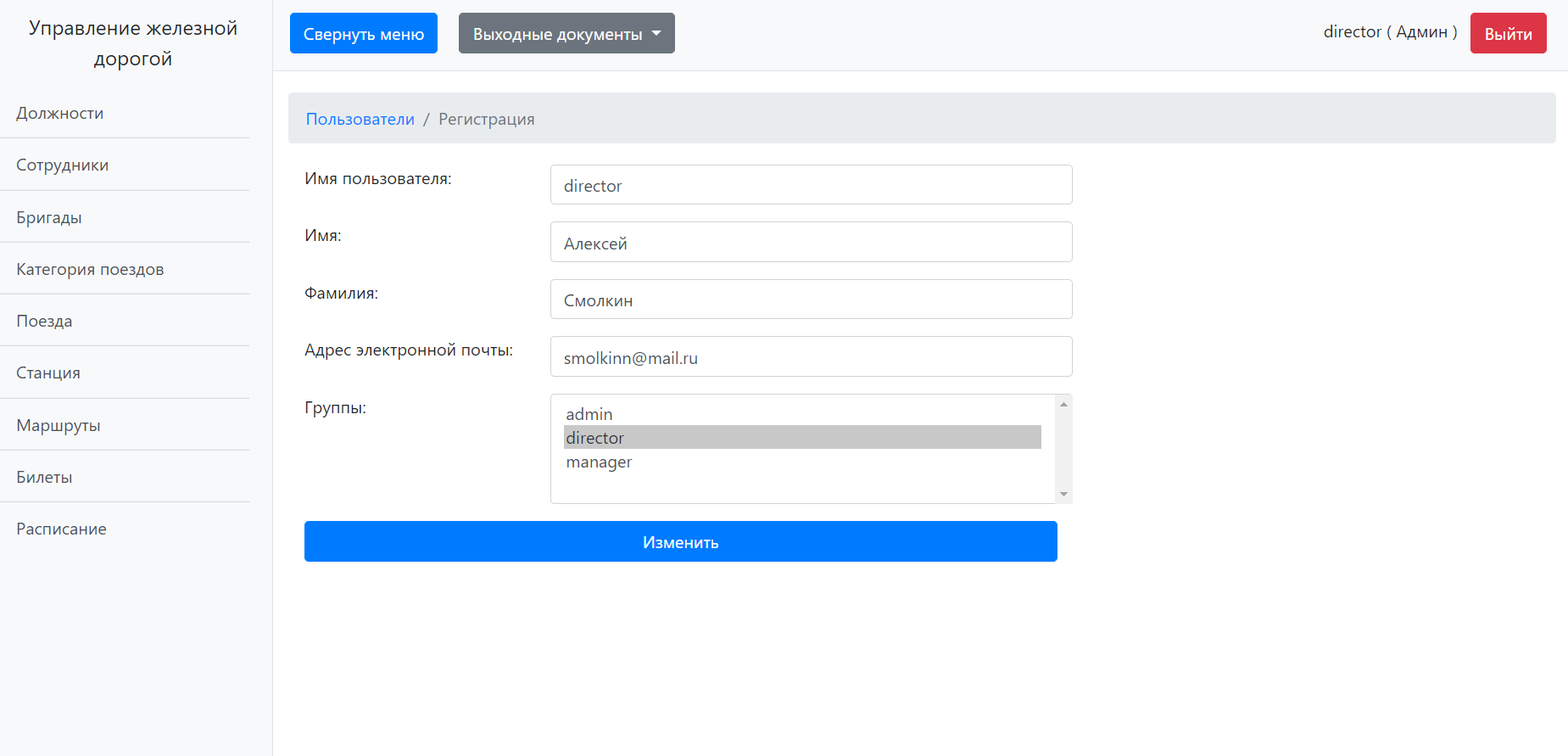


Рисунок 9. Страница для редактирования пользователя

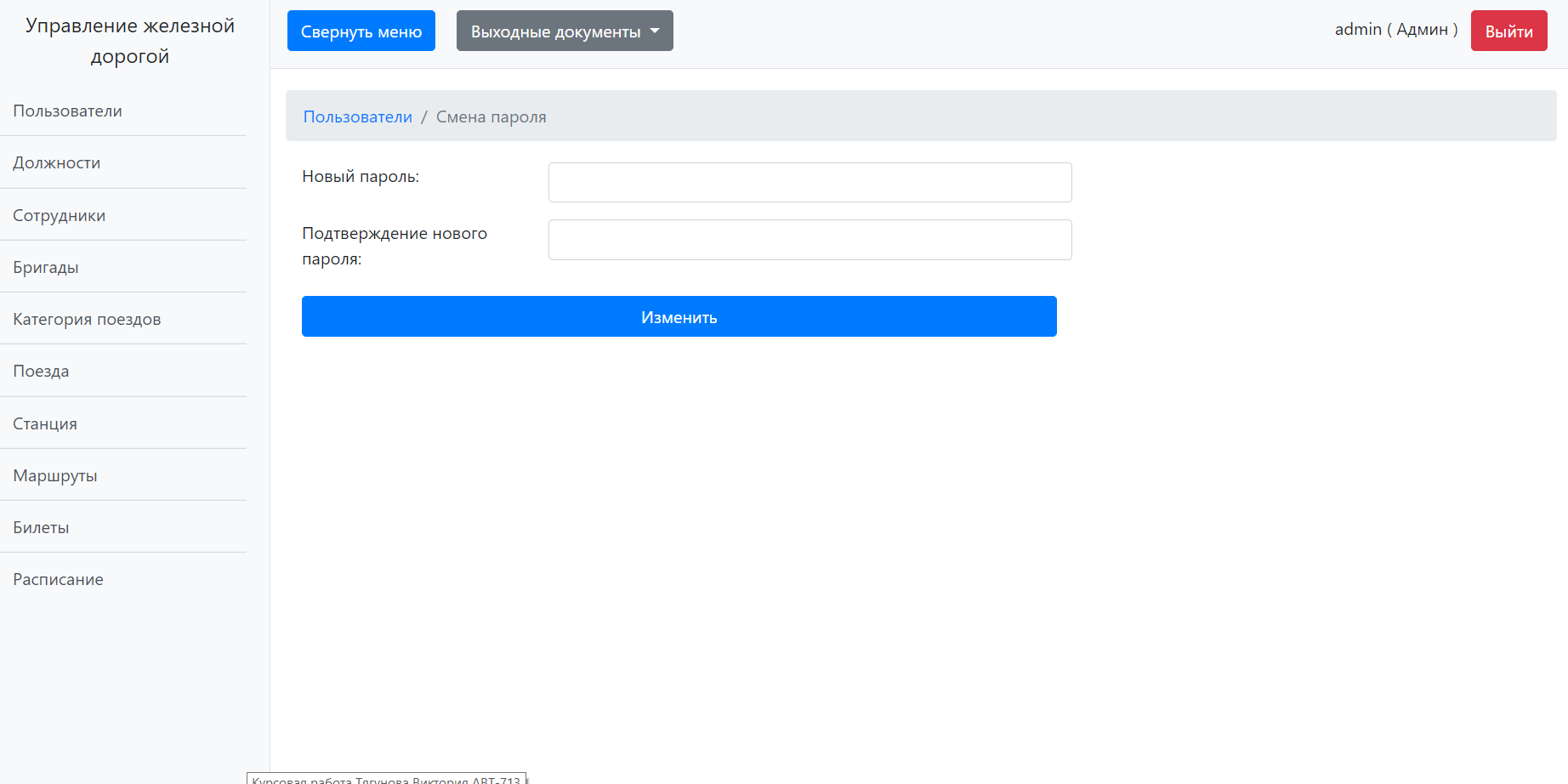


Рисунок 10. Страница для смены пароля

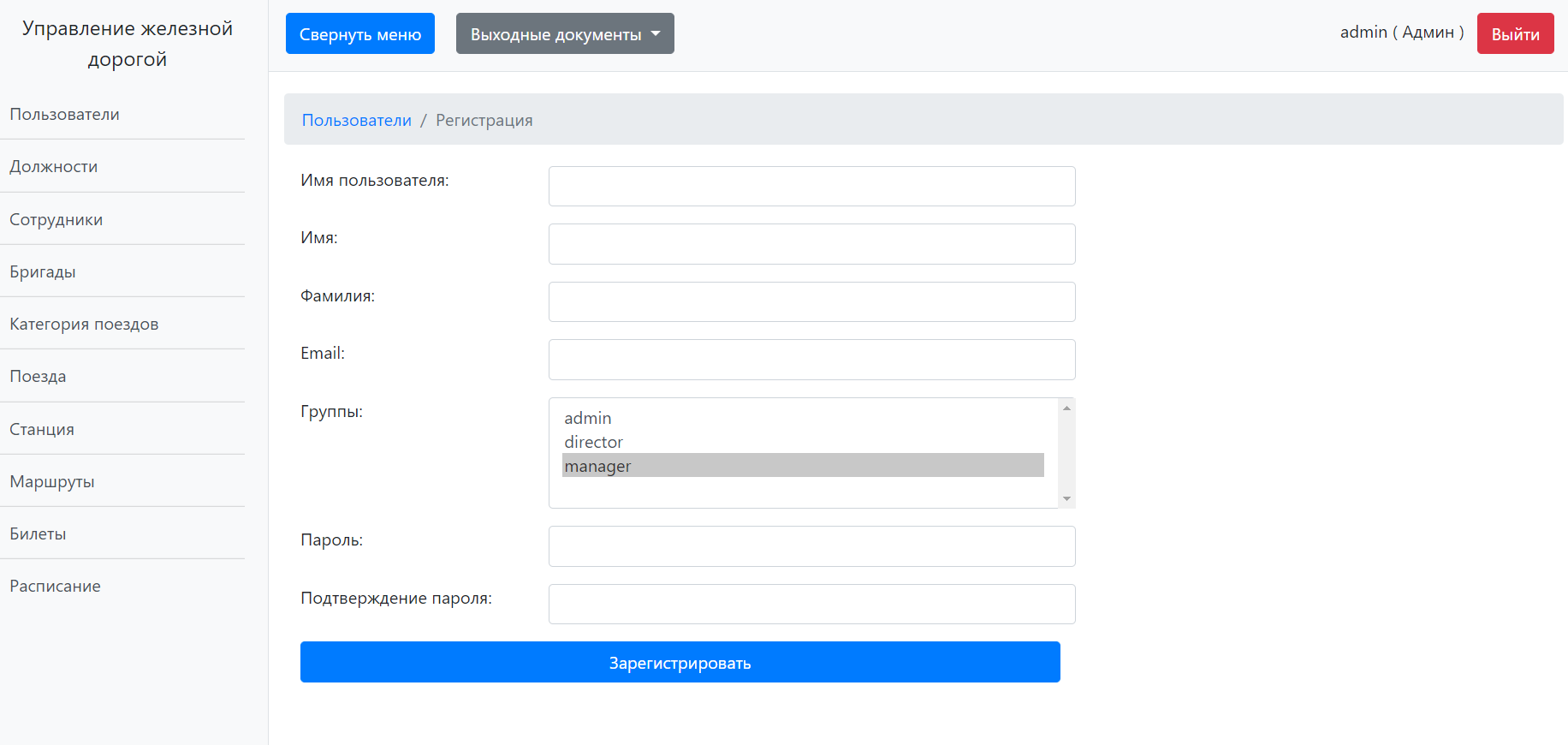


Рисунок 11. Страница для регистрации пользователя администратором

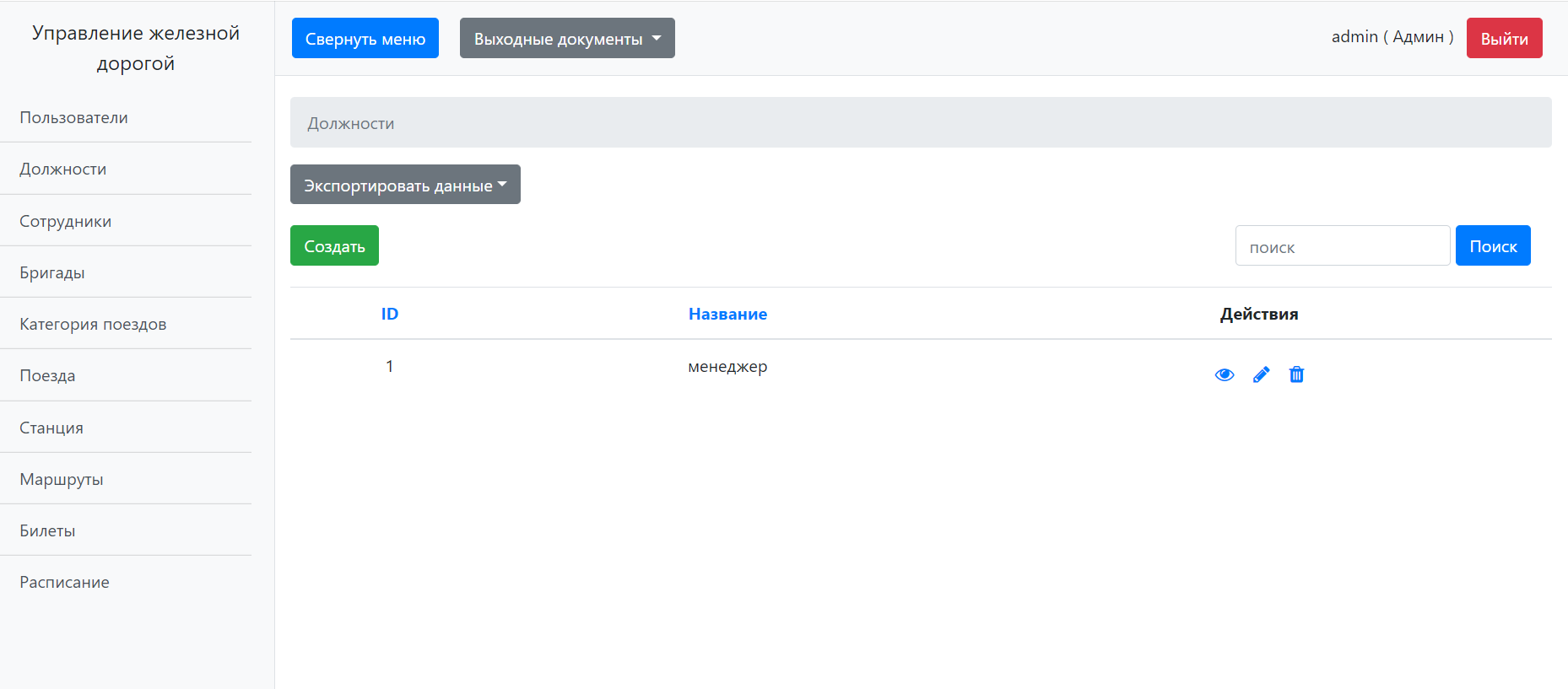


Рисунок 12. Страница для просмотра должностей

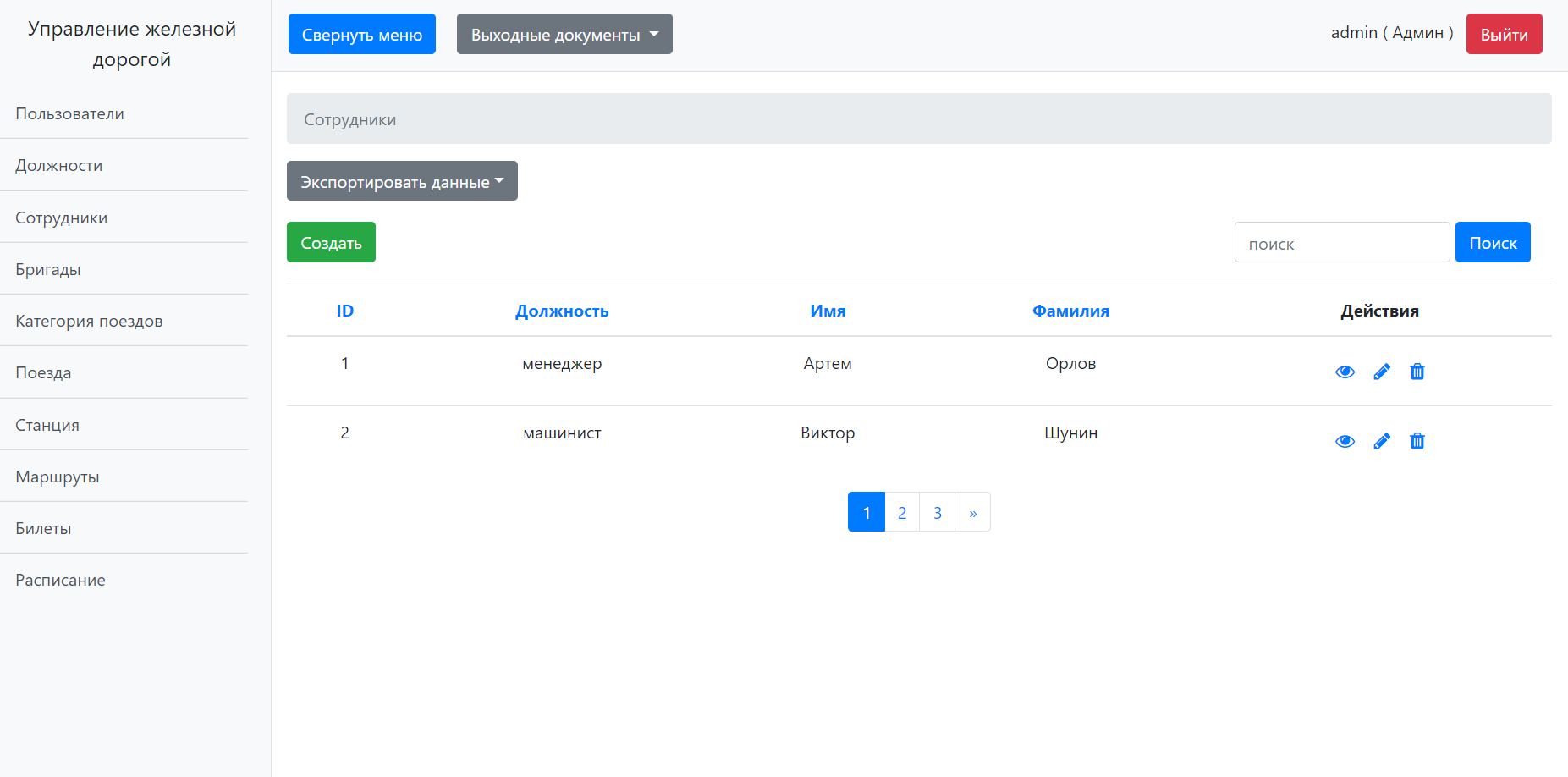


Рисунок 13. Страница для просмотра сотрудников

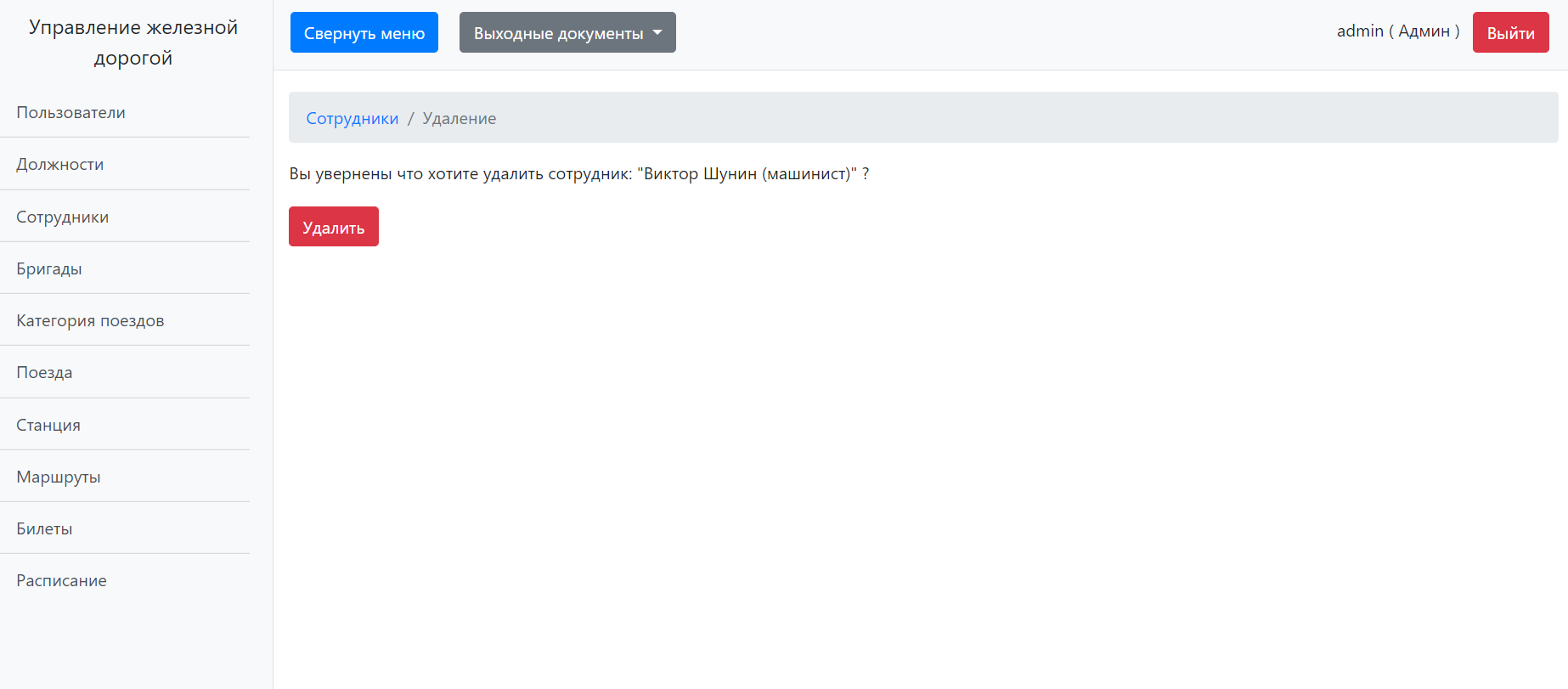


Рисунок 14. Страница подтверждения удаления сотрудника

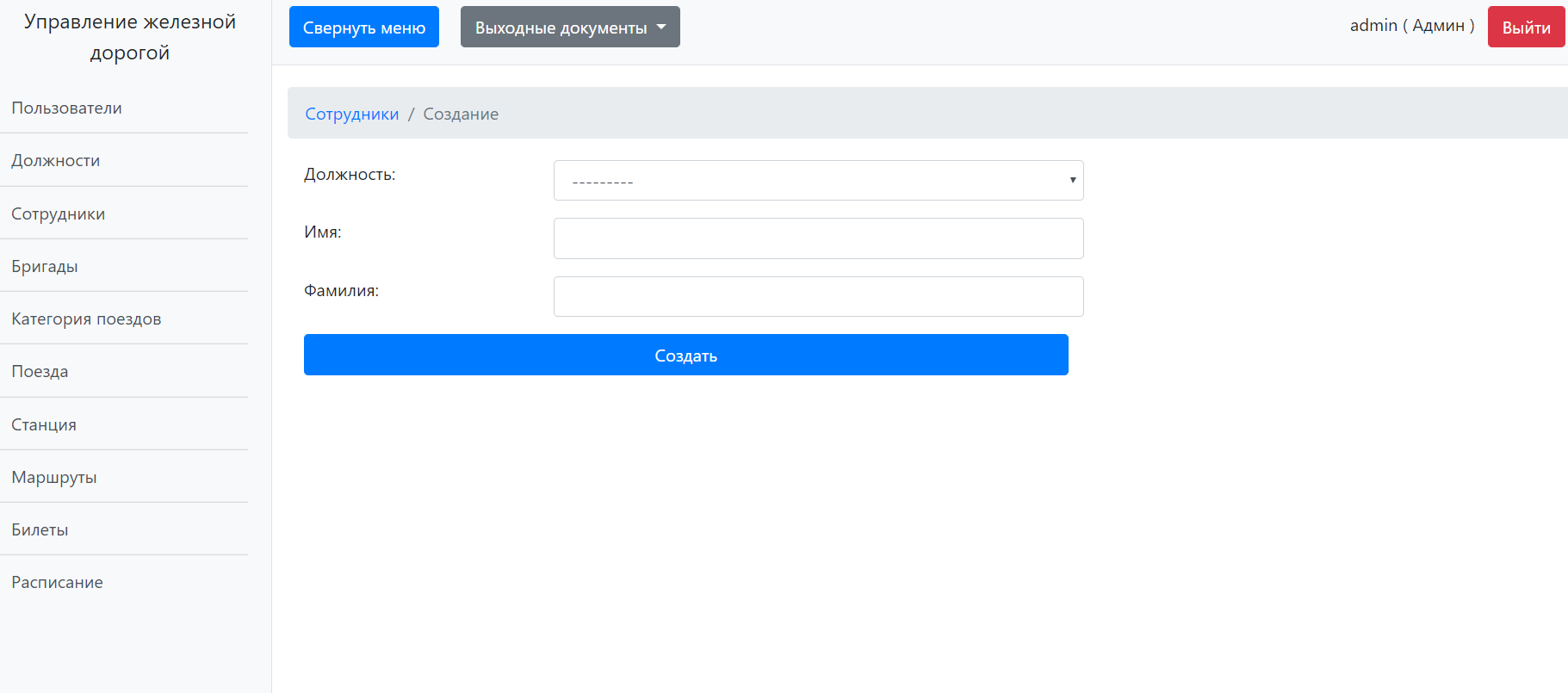


Рисунок 15. Страница для создания сотрудника

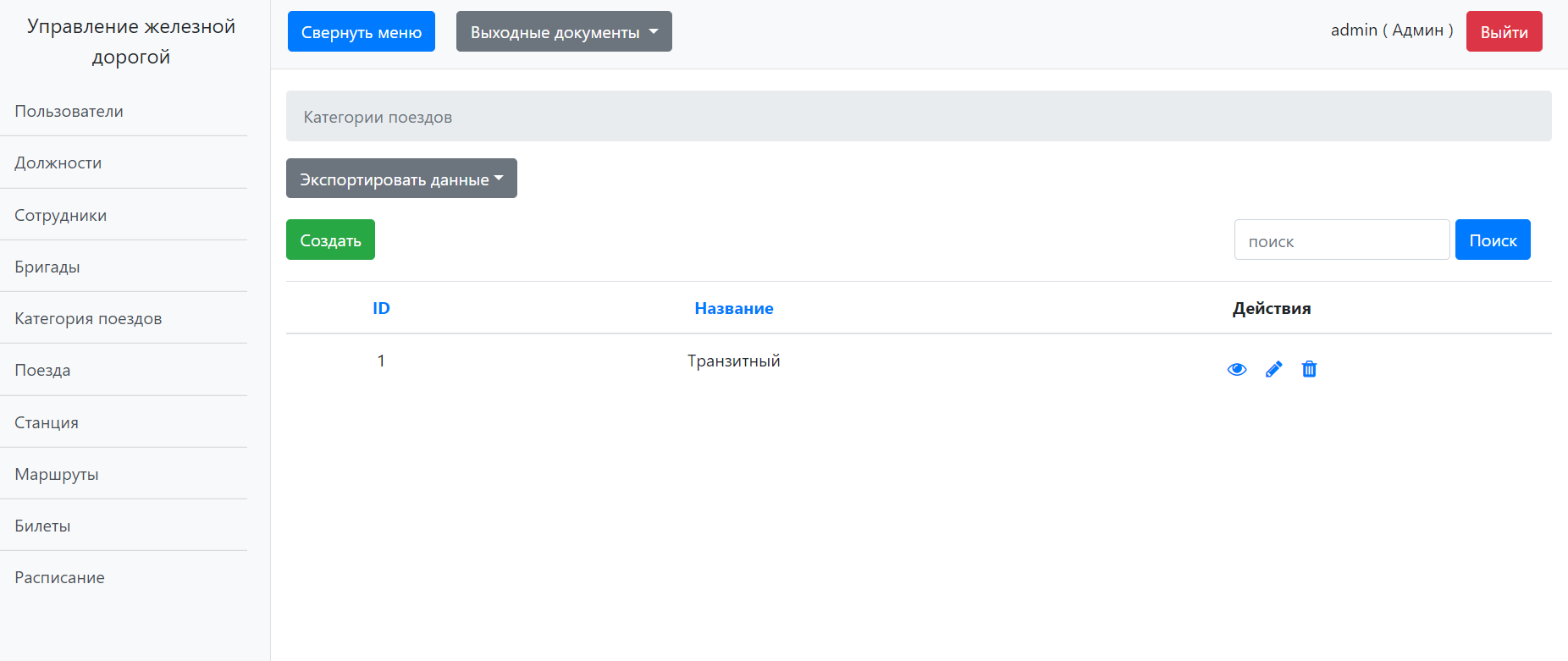


Рисунок 16. Страница для просмотра категорий поезда

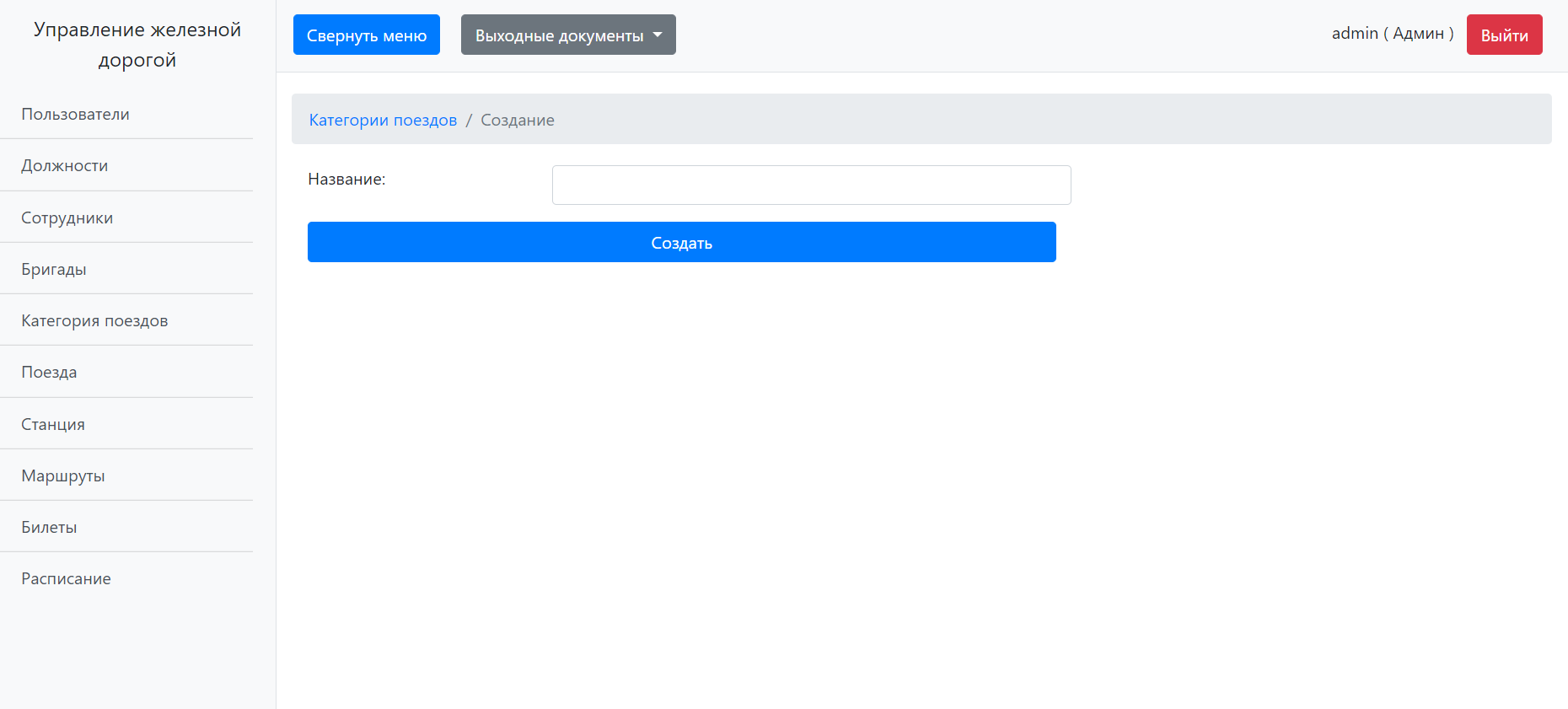


Рисунок 17. Страница для создания категории поезда

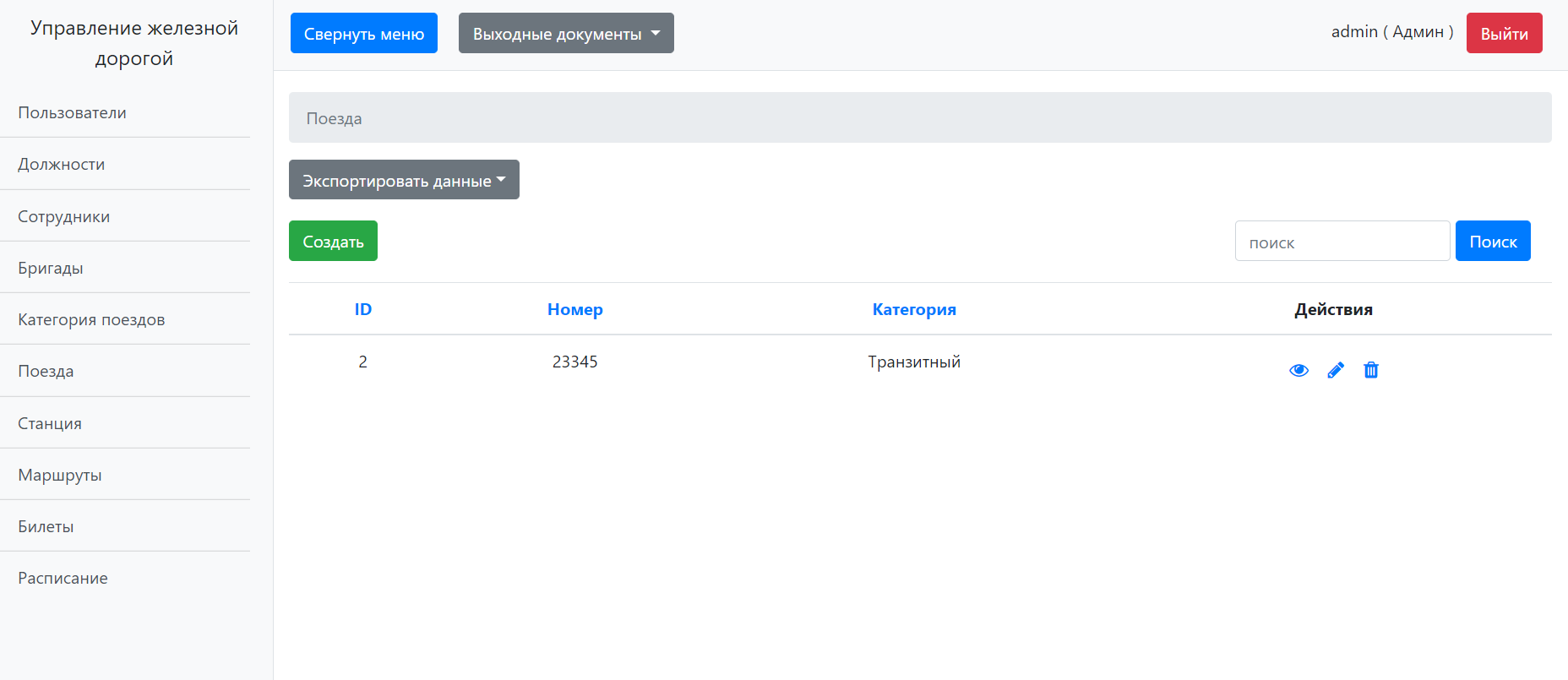


Рисунок 18. Страница для просмотра поездов

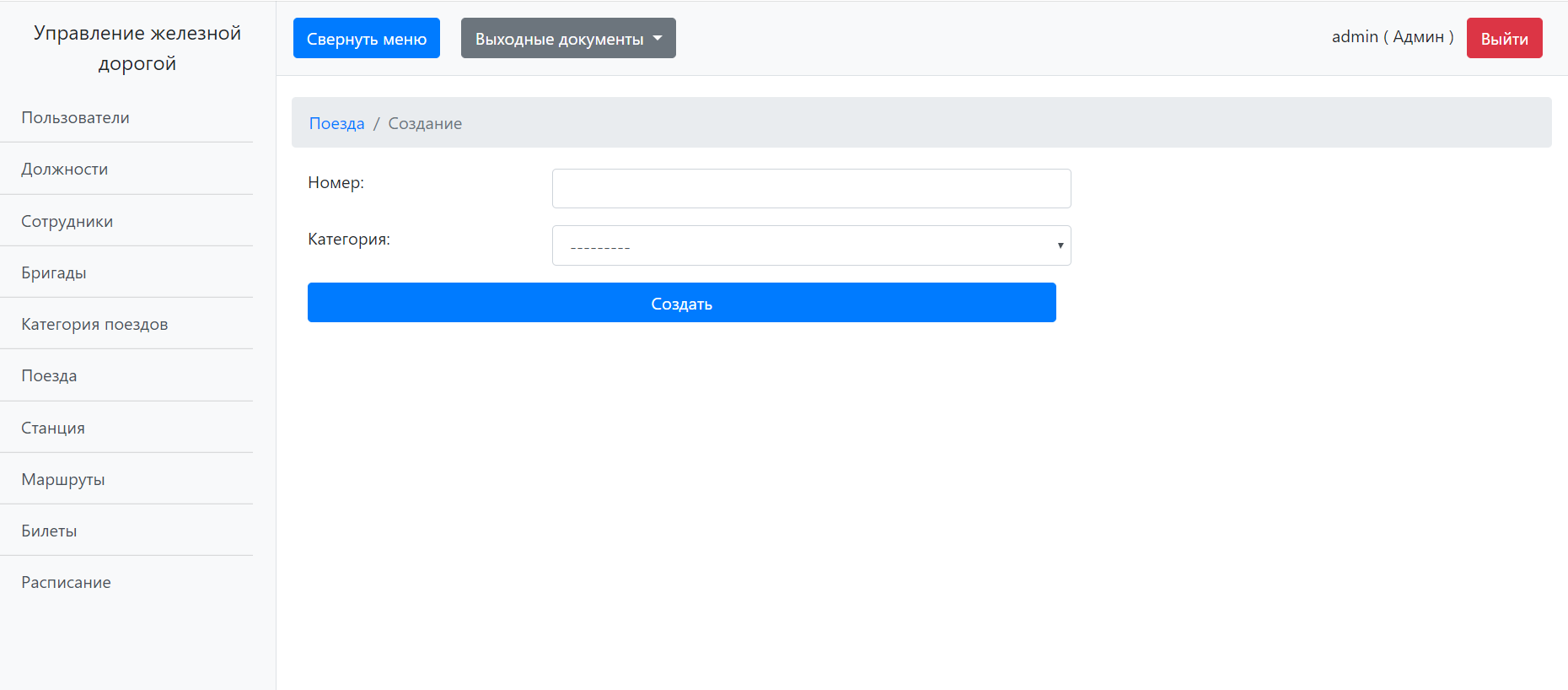


Рисунок 19. Страница для создания поезда

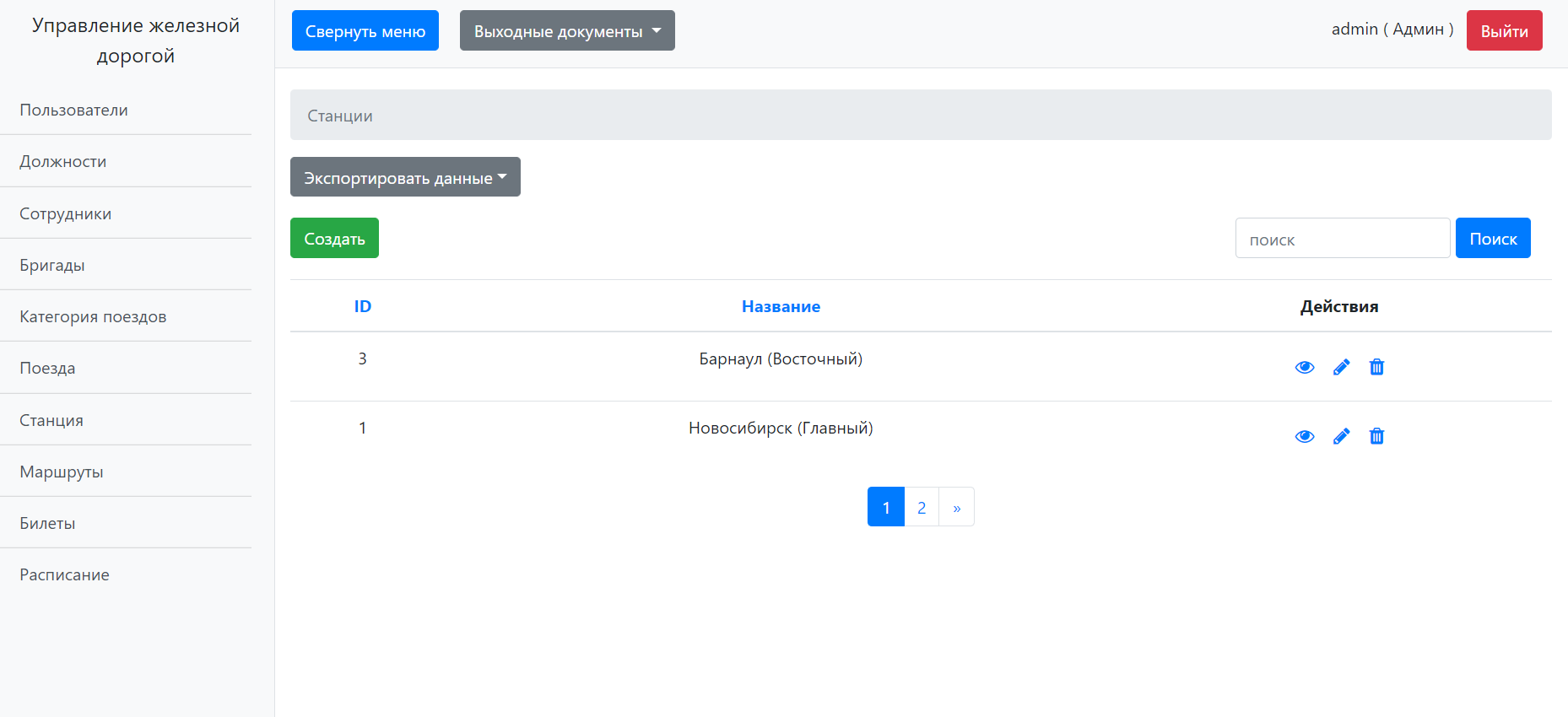


Рисунок 20. Страница для просмотра станций

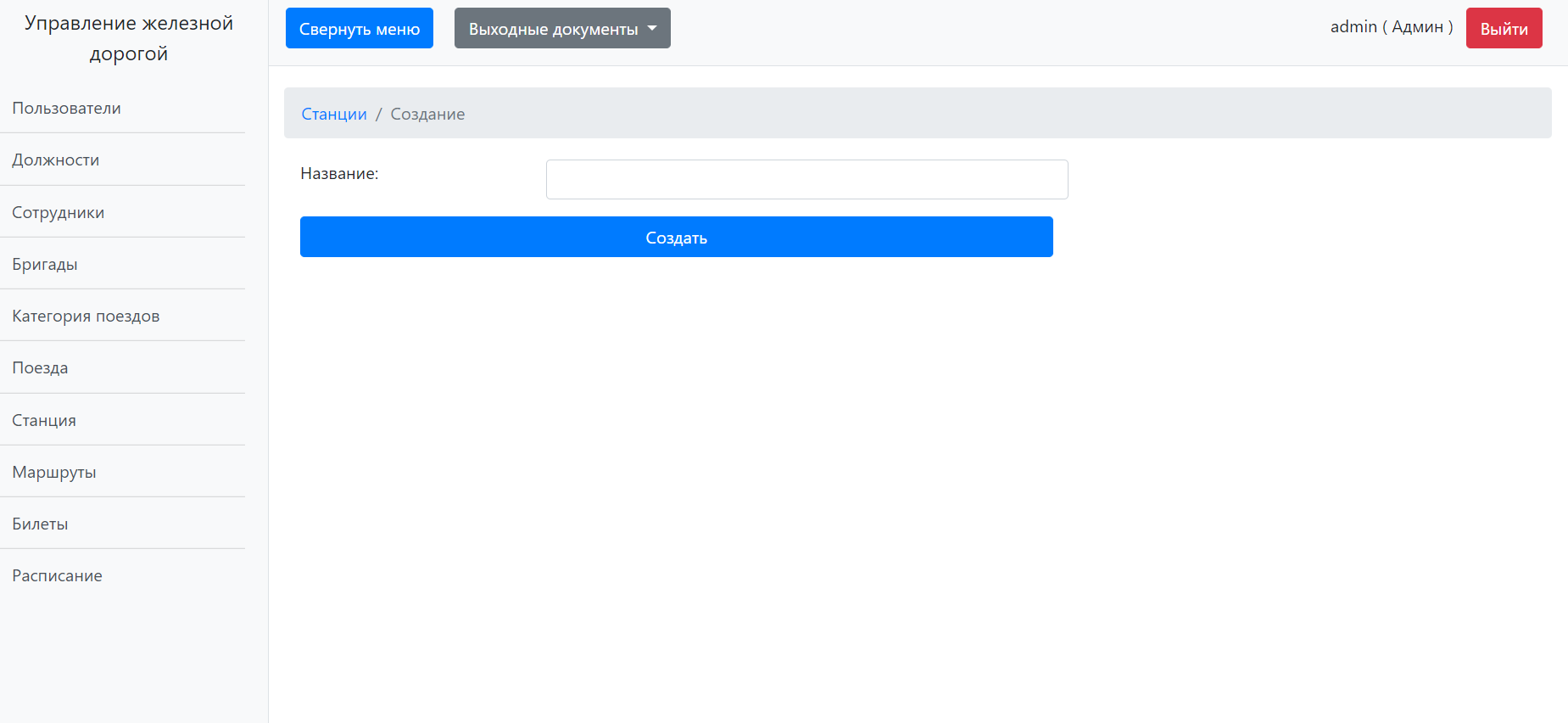


Рисунок 21. Страница для создания станций

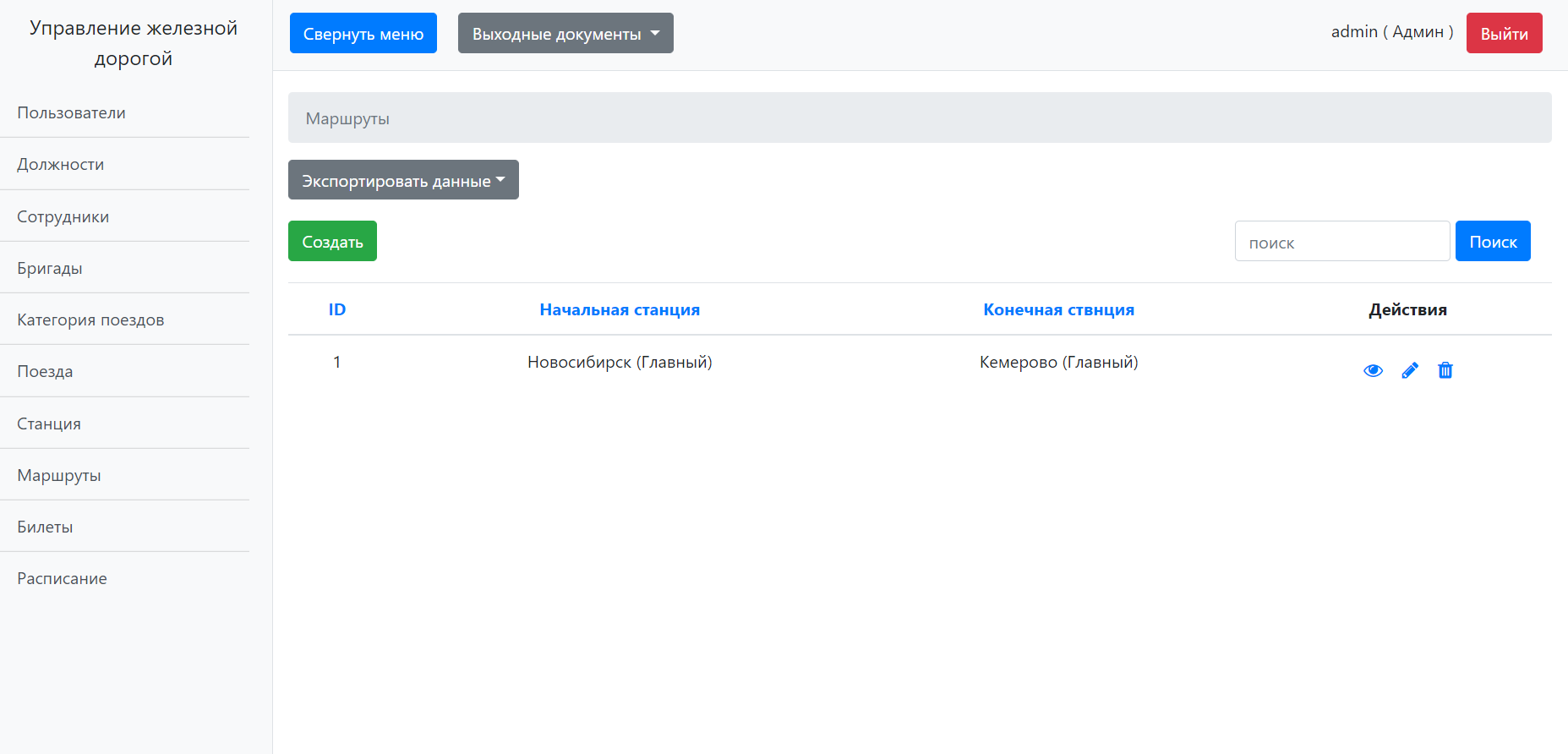


Рисунок 22. Страница для просмотра маршрутов

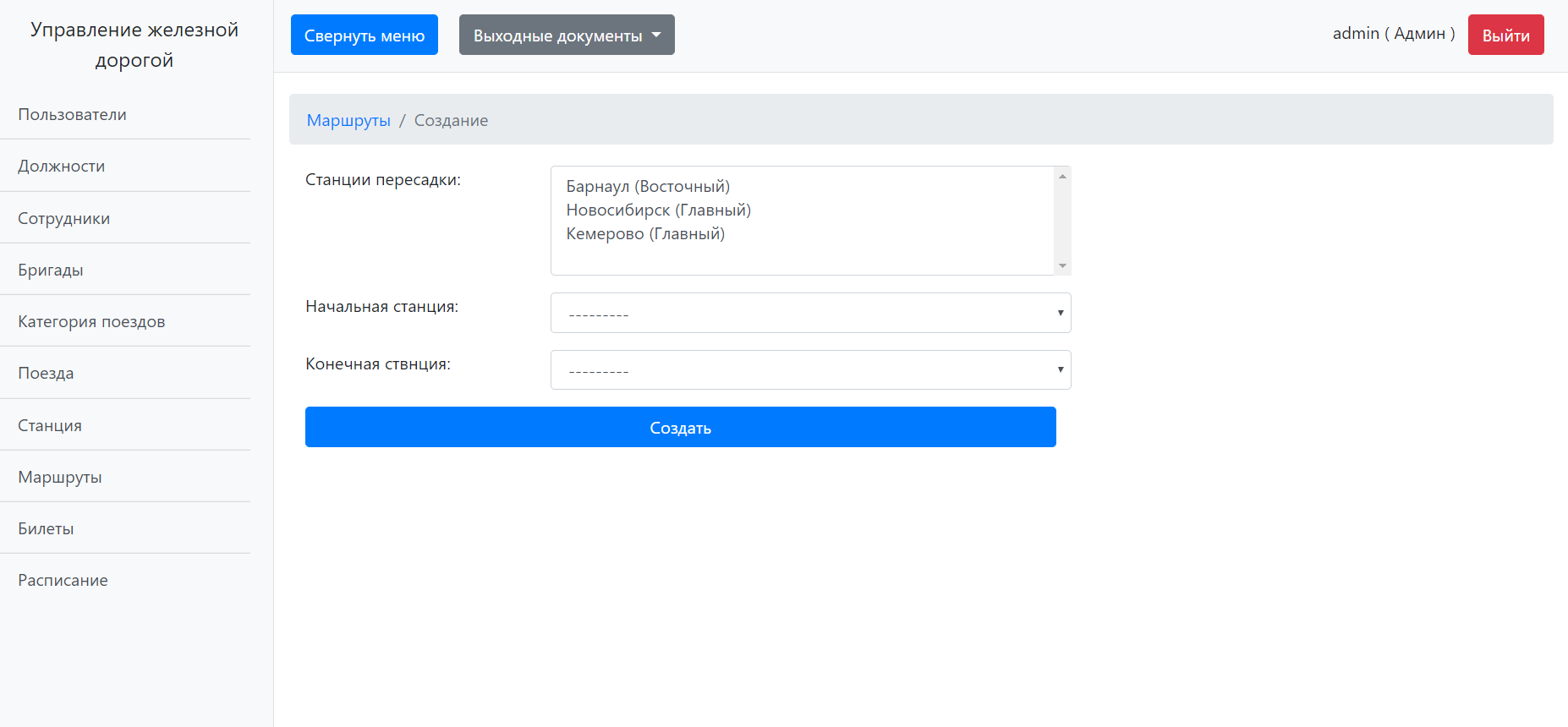


Рисунок 23. Страница для создания маршрутов

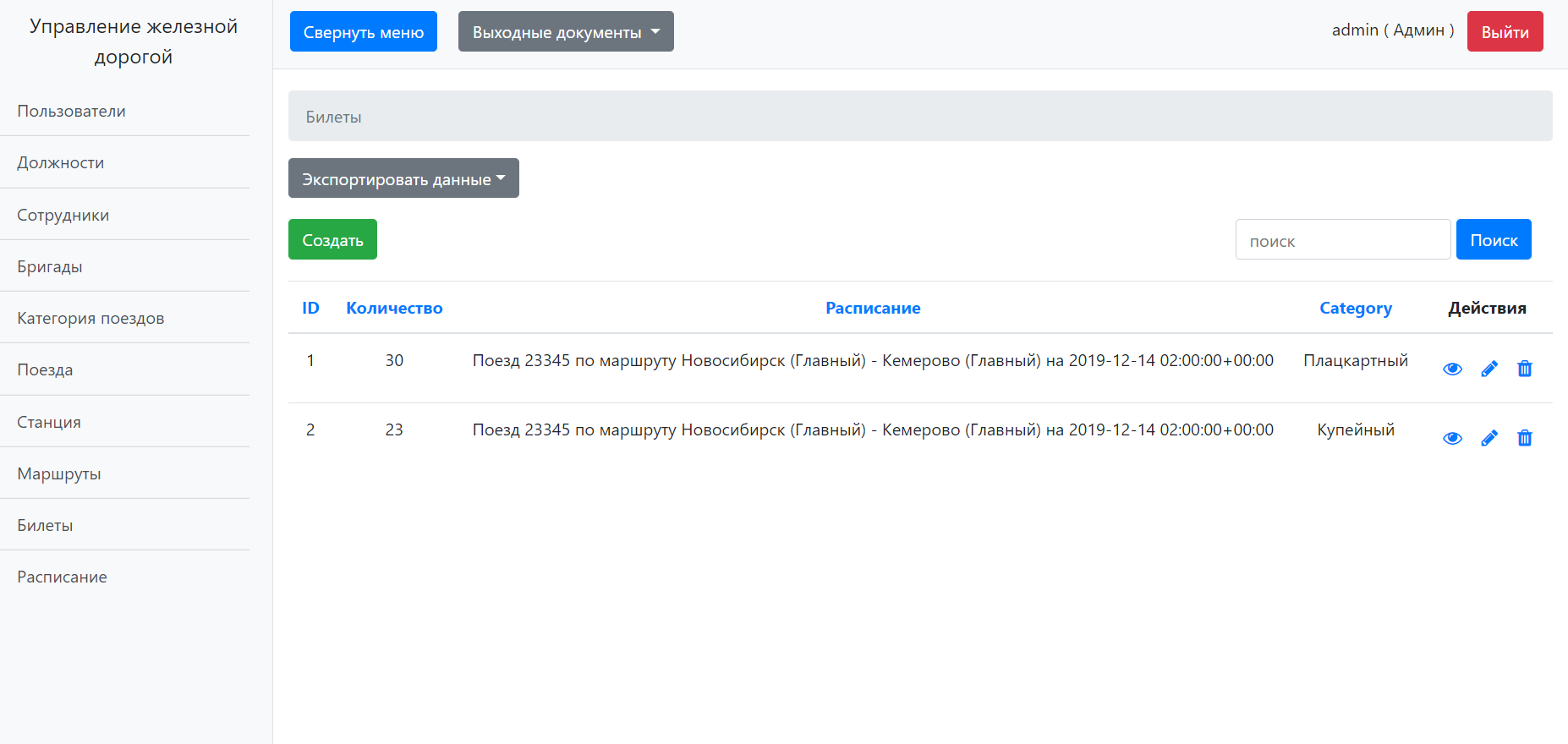


Рисунок 24. Страница для просмотра билетов

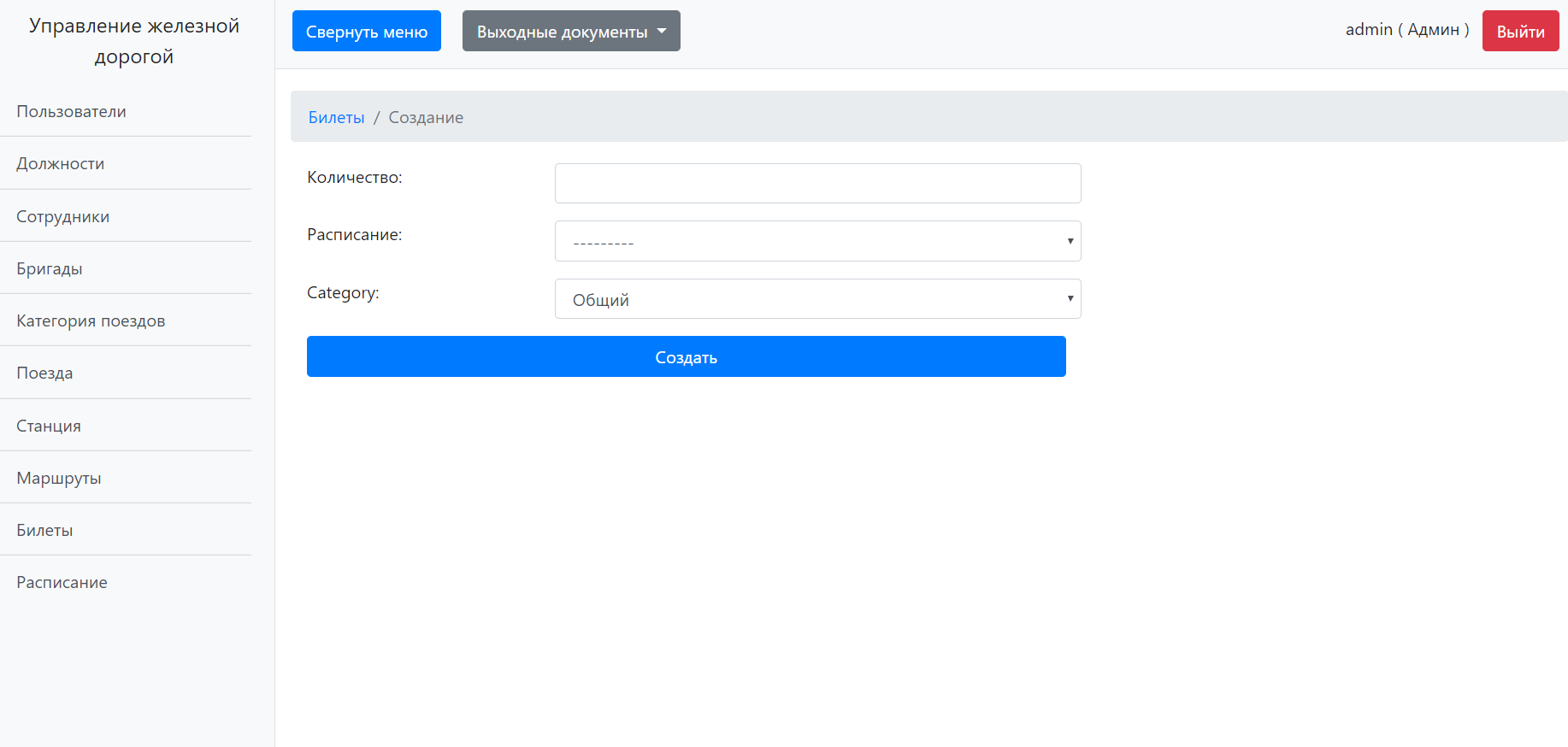


Рисунок 25. Страница для создания билетов

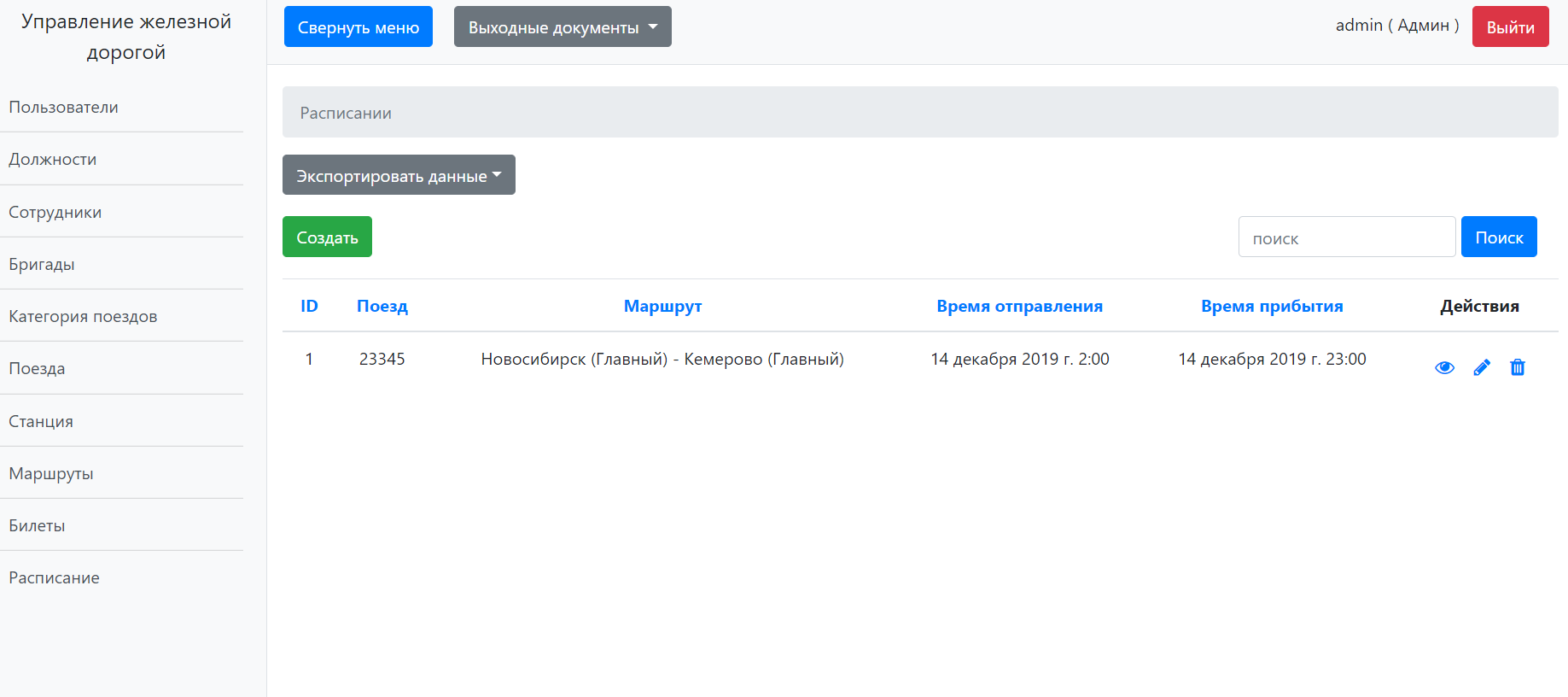


Рисунок 26. Страница для просмотра расписания

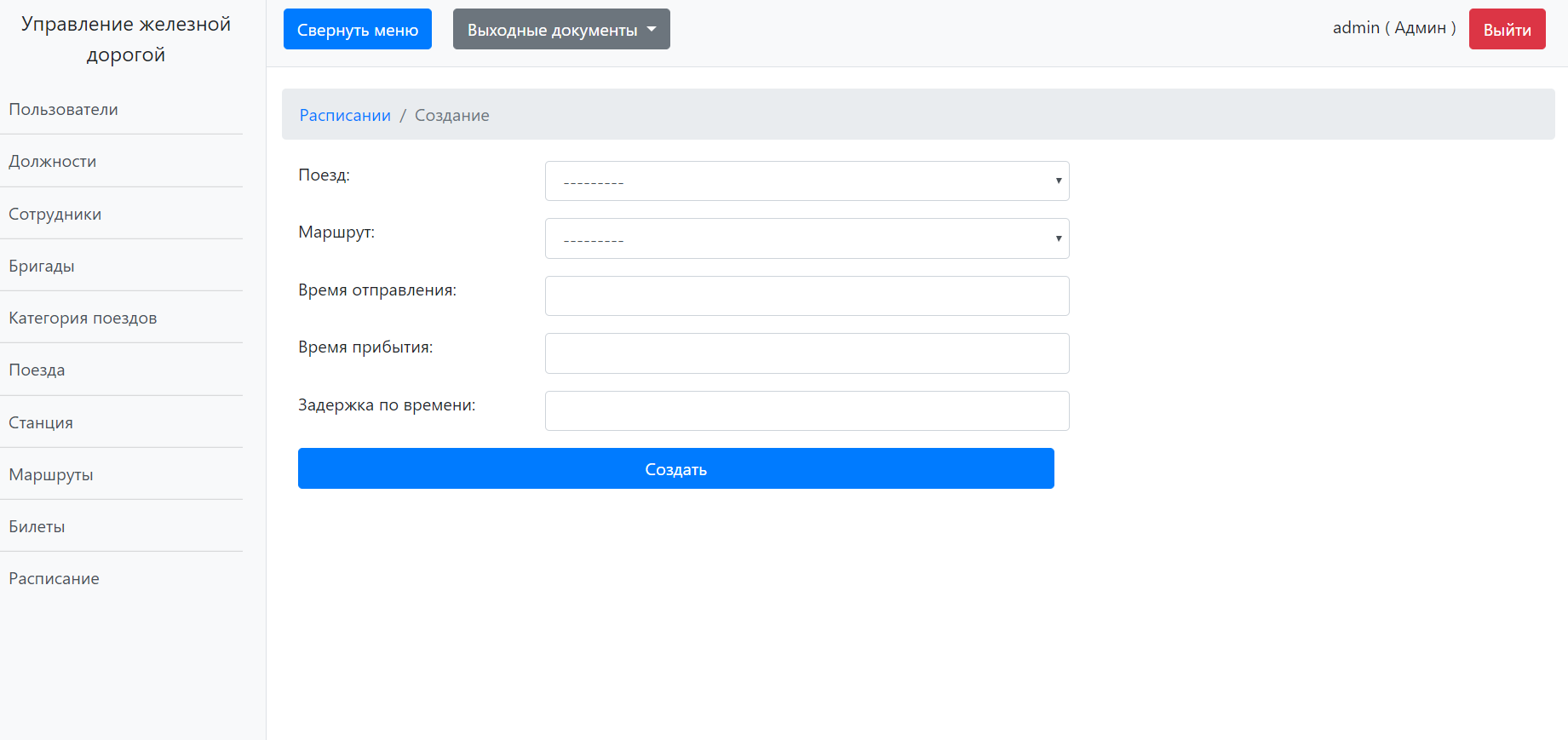


Рисунок 27. Страница для создания расписания



Рисунок 28. Кнопки для манипуляции с пользователем

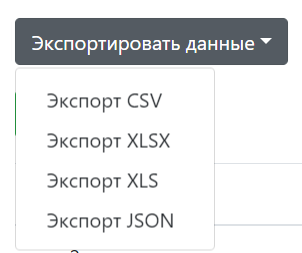


Рисунок 29. Кнопка экспортирования данных

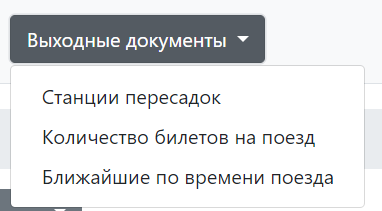


Рисунок 30. Кнопка вывода выходных документов

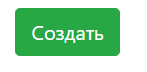


Рисунок 31. Кнопка создания элемента

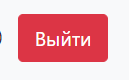


Рисунок 32. Кнопка выхода из системы

# **7 Формы выходных документов**

# Для примера приведем запрос для вывода справки о станции пересадки:

SELECT name FROM app\_station AS s

INNER JOIN app\_route\_transfer AS tr ON tr.station\_id = s.id

INNER JOIN app\_route AS r ON r.id = tr.route\_id

WHERE r.start\_station\_id = 1 AND r.finish\_station\_id = 2

Так как для создания данного приложения мы используем подход ORM, ORM (Object-Relational Mapping) – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных», то нам не нужно вставлять вручную, за нас это делает фреймворк:

def post(self, request):

form = self.form(request.POST) // принимаем данные с формы POST запросы

if form.is\_valid(): // если все данные корректны

result = form.clean()// получаем данные

timetable = Timetable.objects.filter(train=result['train'])// получаем все расписание, которое относится к этому поезду

if not timetable.count():// если не существуют такие расписания

form.add\_error('train', 'Расписаний с таким поездом нету')

return render(request, 'instance/info\_transfer.html', {'form': form})

document = Document() // формирование документа

document.add\_heading('Справка о наличии билетов на указаный поезд', 0)

document.add\_paragraph('Поезд {} имеет следующее поличество билетов: '.format(result['train']))

count\_all = 0

for timetable in timetable.all():// во всех расписаниях происходит перебор данных

count = 0

tickets = Ticket.objects.filter(timetable=timetable)//находим все билеты, у которых есть это расписание

tickets\_list = []

for tickets in tickets.all():// проходим по всем билетам и формируются данные вывода

count += tickets.count

tickets\_list.append("Билеты катергии \"{}\" в количестве {}".format(tickets.get\_category\_display(), tickets.count))

count\_all += count

document.add\_paragraph("Расписание: '{}' имеет {} билетов: {}".format(str(timetable), count, '. '.join(tickets\_list)))// добавляются данные для вывода в документ

document.add\_paragraph("Общее количестов билетов: {}" . format(str(count\_all)))

response = HttpResponse(

content\_type='application/vnd.openxmlformats-officedocument.wordprocessingml.document')

response['Content-Disposition'] = 'attachment; filename=download.docx'

document.save(response)

return response// возвращается документ пользователю

return render(request, 'instance/info\_transfer.html', {'form': form})

Результат выполнения в Word:

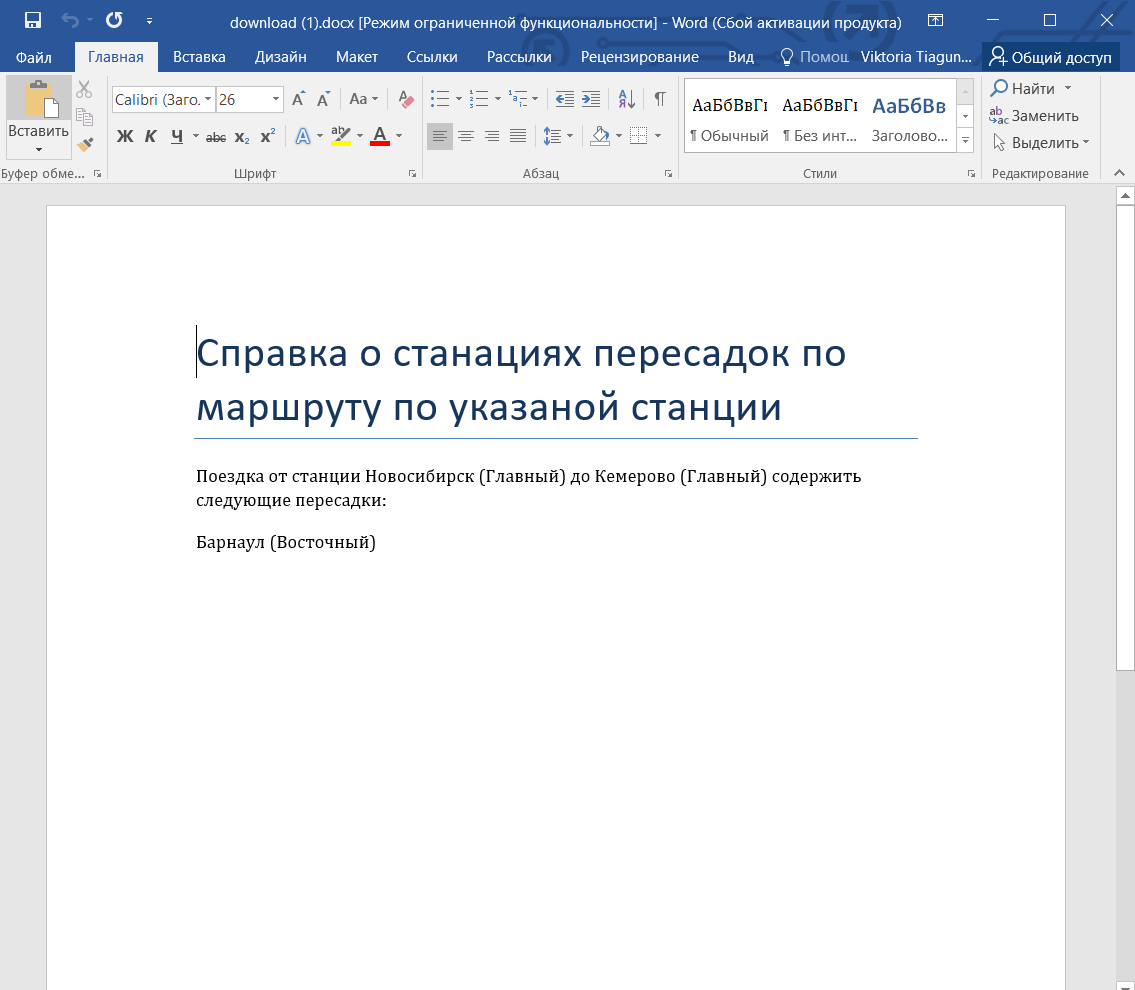


Рисунок 33. Результат запроса.

# **8 Пример запроса**

Рассмотрим работу с запросами на примере добавления нового пользователя. Для перехода к списку пользователей необходимо выбрать соответствующую кнопку в меню. Откроется полный список пользователей. Для создания пользователя нужно нажать на кнопку «Создать», затем откроется страница регистрации:

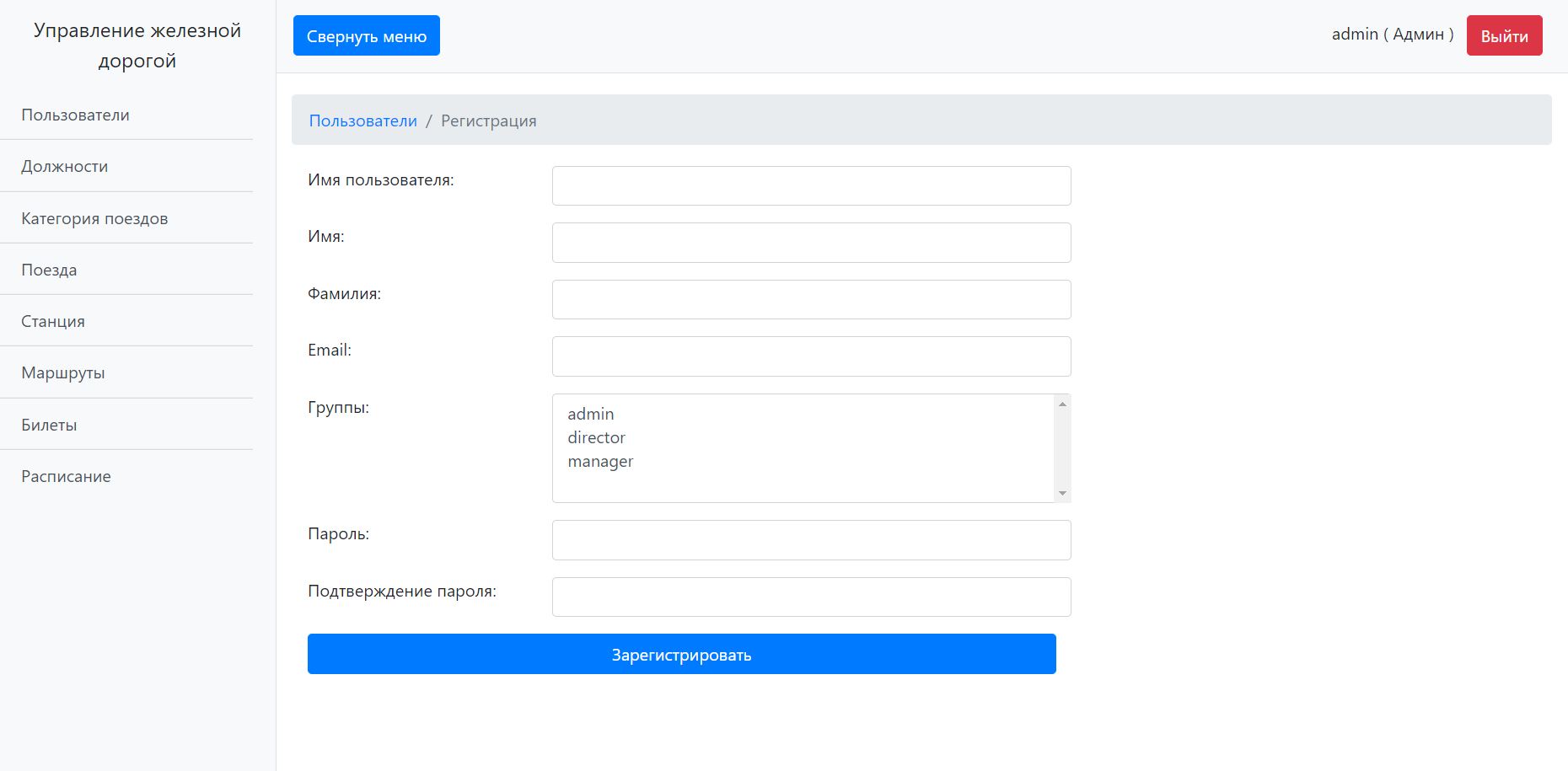


Рисунок 34. Страница регистрации нового пользователя

После заполнения всех полей необходимо нажать клавишу «Зарегистрировать пользователя». В данном примере все поля обязательны для заполнения, также при введении email пользователь обязательно должен использовать корректный email.

Так как для создания данного приложения использовалась технология ORM, ORM (Object-Relational Mapping) – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных», то для сохранения пользователя нужно только вызвать метод объекта save.

При нажатии на кнопку «Зарегистрировать», затем появится введенный нами пользователь.

# **Заключение**

В ходе выполнения данной курсовой работы было создано web-приложение с графическим интерфейсом, использующее базу данных для хранения информации и отвечающее требованиям и поставленным задачам, которые были определены для данной программы.

Также был получен опыт проектирования баз данных на концептуальном, логическом и физическом уровнях, получены навыки работы с СУБД MS SQL Server. Получены практически навыки составления сложных запросов.

# **Приложение А**

# **Руководство пользователя**

Программа «Управление железнодорожным вокзалом» обладает доступным, интуитивно понятным интерфейсом, что позволяет даже неопытным пользователям быстро разобраться в принципах его работы.

Для создания первого пользователя (администратора) необходимо произвести следующую команду на сервере:

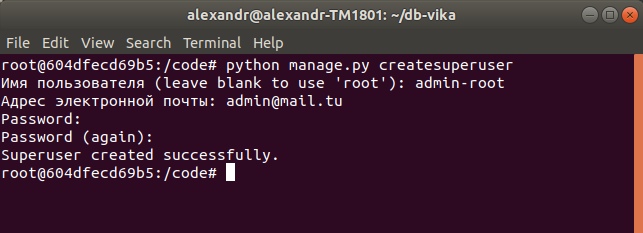


Рисунок 35. Страница регистрации нового пользователя

Пользователю требуется зайти на страницу http://ec2-3-19-32-226.us-east-2.compute.amazonaws.com/accounts/login/ ввести свой логин и пароль, на данном примере пароль не отображается символами, так как используется Unix система.

Для начала работы необходимо авторизоваться. После этого открывается окно с доступными пользователю функциями (в зависимости от роли). Во вкладках «Маршрут», «Расписание», «Поездная бригада», «Билеты», «Сотрудники», «Станции» можно получить содержание соответствующих вкладок, а также, при наличии необходимых прав доступа, можно их редактировать.

При нажатии на кнопку «Экспортировать данные» на всех вкладках, можно вывести данные в нужном формате (CSV, XLSX, XLS, JSON), при нажатии на кнопку «Выходные документы» формируются справки: станции пересадок, количество билетов на поезд, ближайшие по времени поезда.

# **Список литературы**

1. Астапчук В.А. Базы данных: методические указания / В.А. Астапчук, В.Б. Панова, - Новосибирск: НГТУ, 2012 – 60 с.

2. Django. Подробное руководство, 2-е издание. – Пер. с англ. – СПб.: СимволПлюс, 2010. – 560 с.