Титульный лист

Задание на бакалаврскую работу

Содержание

[Введение 4](#_Toc165924624)

[ГЛАВА 1 ОБЩАЯ ЧАСТЬ 8](#_Toc165924625)

[1.1 Описание действующих информационных систем экстренных служб ОАО «РЖД» 8](#_Toc165924626)

[1.2 Организационная структура 9](#_Toc165924627)

[1.3 Описание предметной области 11](#_Toc165924628)

[1.4 Архитектура разрабатываемой информационной системы 13](#_Toc165924629)

[1.5 План работы с информационной системой 15](#_Toc165924630)

[1.6 Описание аварийных ситуаций 15](#_Toc165924631)

[1.7 Обзор средств разработки 17](#_Toc165924632)

[ГЛАВА 2 ПРАКТИЧЕСКАЯ ЧАСТЬ 23](#_Toc165924633)

[2.1 Требования к функциональным характеристикам 23](#_Toc165924634)

[2.2 Требования к организации входных данных 23](#_Toc165924635)

[2.3 Требования к организации выходных данных 24](#_Toc165924636)

[2.4 UML диаграммы вариантов использования интерфейса управления пользователями для информационной системы 24](#_Toc165924637)

[2.5 ER диаграмма базы данных 27](#_Toc165924638)

[2.6 Описание сущностей базы данных 28](#_Toc165924639)

# Введение

Информационные системы (ИС) представляют собой комплекс программных и аппаратных средств, предназначенных для сбора, хранения, обработки, передачи и использования информации. Они позволяют автоматизировать процессы обработки информации, улучшить доступ к данным, оптимизировать работу с информацией, повысить эффективность деятельности организаций и обеспечить принятие обоснованных управленческих решений на основе данных и аналитики.

Информационные системы являются важной частью современного информационного общества и используются в самых различных областях человеческой деятельности, начиная от бизнеса и управления, заканчивая медициной и образованием.

Дипломная работа посвящена разработке информационной системы для экстренных служб ОАО «РЖД». Информационная система позволяет получить доступ к информации о происшествиях, автоматизировать процесс добавления обнаруженных происшествий, их редактирования, удаления, а также проведения и учет мероприятий для ликвидации происшествий.

**Актуальность выпускной квалификационной работы заключается** в предоставлении своевременной информации об происшествиях и методах их ликвидации в экстренные службы ОАО «РЖД».

**Объектом исследования является** ЦСС – филиал ОАО «РЖД»

**Предмет исследования:** работники подразделений ОАО «РЖД», отвечающие за безопасность и коммуникацию.

методика исследования

Информационная система для экстренных служб ОАО «РЖД» будет разработана со следующими целями:

- обеспечение единого информационного пространства для осуществления функций контроля в сфере отслеживания происшествий;

- предоставление экстренным службам ОАО «РЖД» актуальных сведений о происшествиях;

- автоматизация деятельности по осуществлению уполномоченными должностными лицами функций контроля за происшествиями.

**Основной целью выпускной квалификационной работы** на тему «Разработка информационной системы для экстренных служб ОАО «РЖД» является создание современной и эффективной информационной системы, которая позволит повысить быстродействие принятия решений по ликвидации происшествий, а также быстрой передаче информации ответственным лицам.

**Задачами выпускной квалификационной работы** являются:

- ведение реестра происшествий;

- возможность изменения отображения страниц (для светлого времени суток – черный, для темного времени суток - светлый);

- работа с картами: добавление, изменение, удаление и просмотр каждого происшествия;

- отображение кроткой информации об происшествиях на карте;

- отслеживание статуса происшествий: возможность отслеживания текущего состояния каждого происшествия;

- отслеживание всех происшествий на общей карте;

- информация об пользователях;

- информация об ответственных лицах;

- разграничение доступа к различным функциям информационной системы;

- оперативный поиск по происшествиям, пользователям и другим сущностям базы данных;

- поэтапное внесение информации об происшествиях;

- разграничение доступа к различным функциям информационной системы.

Задачи по работе с картами нацелены на предоставлении информации о происшествии, его обнаружителе и местоположении с целью оперативной передачи этой информации экстренным службам ОАО «РЖД». В зависимости от типа выявленного происшествия будут применяются различные мероприятия для его ликвидации. Также в информационной системе ведётся реестр об ответственных лицах, разбитый по районам.

Разработка информационной системы для экстренных служб ОАО «РЖД» имеет огромное значение для повышения эффективности деятельности компании, обеспечивания безопасности и оперативности реагирования на чрезвычайные ситуации.

**Практическая значимость** разработки информационной системы представлена в следующих аспектах:

- улучшение оперативности реагирования на происшествия и аварии на объектах железнодорожного транспорта. Посредством внедрения разработанной информационной системы оперативные службы смогут своевременно получать данные о возникших ситуациях, что позволит ускорить принятие решения и минимизировать ущерб;

- повышение эффективности взаимодействия между различными подразделениями и службами ОАО «РЖД». Информационная система предоставит возможность оперативному обмену данными между сотрудниками различных служб (описать), что способствует согласованной работе и эффективному взаимодействию в экстренных ситуациях;

- улучшение контроля руководства над ходом решения проблем. Разработанная информационная система позволит ответственному лицу ОАО «РЖД» оперативно отслеживать текущие события, принимать важные решения на основе актуальных данных и обеспечивать контроль за выполнением мероприятий по ликвидации происшествий;

- снижение операционных рисков и повышение безопасности на транспорте. Благодаря внедрению информационной системы экстренные службы смогут быстрее реагировать на возможные угрозы безопасности, что способствует снижению рисков и обеспечению безопасности пассажиров и сотрудников.

Для разработки необходимо использовать фреймворк Django на языке Python, которая будет соответствовать требованиям безопасности, а также быть легко расширяемой и обновляемой. Для работы с базами данных был выбран SqLite, который является компактной встроенной реляционной базой данных с открытым кодом.

# ГЛАВА 1 ОБЩАЯ ЧАСТЬ

## 1.1 Описание действующих информационных систем экстренных служб ОАО «РЖД»

1. **«Информационная система обработки заявок»**

Функции системы:

- организации единой точки регистрации заявок;

- сокращения бумажного документооборота при создании заявок на услуги связи;

- сокращения трудоёмкости обработки заявок;

- сокращения времени обработки заявок;

- исключения потерь информации, связанных с оформлением заявок;

- исторического хранения информации о поданных заявках.

2. **«Информационная система предоставления данных в части обеспечения безопасности движения, мониторинга и управления чрезвычайными ситуациями»**

«Информационная система представления данных в части обеспечения безопасности движения, мониторинга и управления чрезвычайными ситуациями» предназначено для обеспечения сотрудников ЦЧС (Ситуационный центр мониторинга и управления чрезвычайными ситуациями) и членов оперативного штаба ОАО «РЖД» единым инструментом для мониторинга и управления чрезвычайными ситуациями. Визуализация сведений о нештатной ситуации реализована как для коллективного табло, так и для стационарных рабочих мест в ОАО «РЖД» и мобильных устройств (планшетов).

Главной задачей информационной системы является информационное обеспечение работы оперативного штаба по ликвидации последствий транспортных происшествий и чрезвычайных ситуаций на инфраструктуре ОАО «РЖД».

## 1.2 Организационная структура

Организационная структура ОАО «РЖД» включает в себя: аппарат управления, филиалы, структурные подразделения и дочерние зависимые общества. Железные дороги - филиалы ОАО «РЖД» состоят из железных дорог по направлениям, всего их 17. Функциональные филиалы включают в себя центральные аппараты управления, такие как: центральная станция связи (ЦСС), центральная дирекция по ремонту пути, дирекция тяги и другие.

Дорожные дирекции включают в себя: службы электрификации и электроснабжения (ЭЧ), службы автоматики и телемеханики (СЦБ), службы пути и сооружений (ПЧ), служба гражданских сооружений (НГЧ).

«Информационная система для экстренных служб ОАО «РЖД»» разрабатывается для дорожных дирекций и функциональных филиалов ОАО «РЖД».



Рисунок 1. Организационная структура ОАО «РЖД»

Источник [1]

## 1.3 Описание предметной области

Информационные системы экстренных служб ОАО «РЖД» играют ключевую роль в обеспечении безопасности и оперативности обслуживания на железнодорожном транспорте. В их состав входят различные компоненты, которые обеспечивают связь, контроль и оперативное реагирование в случае чрезвычайных ситуаций.

Для поддержания оперативного реагирования на происшествия информационная система должна отвечать таким требованиям как:

- надежности и стабильности работы системы, чтобы минимизировать риски сбоев и перебоев в работе экстренных служб;

- быстрой передачи информации о чрезвычайных ситуациях и возможность оперативного реагирования на них;

- защите данных от несанкционированного доступа и сохранении их конфиденциальности;

- совместимости с другими информационными системами и возможности обмена данными между различными службами;

- возможности масштабирования системы в случае расширения функционала или увеличения объемов перевозок;

- постоянного обновления и совершенствования системы в соответствии с новыми требованиями и технологиями.

С появлением новых технологий действующие информационные системы ОАО «РЖД» необходимо обновлять, либо создавать новые, так как данные информационные системы имеют устаревшие протоколы безопасности и ограниченный набор функционала, что затрудняет их масштабируемость и поддержку. В ходе прохождения преддипломной практики мне было предложено разработать новую информационную систему с использованием фреймворка Django на языке Python, которая будет соответствовать новым требованиям безопасности, а также быть легко расширяемой и обновляемой.

Новая информационная система для экстренных служб ОАО «РЖД» предназначена для автоматизации выявления, учета и контроля ликвидации происшествий на железных дорогах. Основными функциями являются:

- ведение реестра происшествий;

- возможность изменения отображения страниц (для светлого времени суток – черный, для темного времени суток - светлый);

- работа с картами: добавление, изменение, удаление и просмотр каждого происшествия;

- отображение кроткой информации об происшествиях на карте;

- отслеживание статуса происшествий: возможность отслеживания текущего состояния каждого происшествия;

- отслеживание всех происшествий на общей карте;

- информация об пользователях;

- информация об ответственных лицах;

- разграничение доступа к различным функциям информационной системы;

- оперативный поиск по происшествиям, пользователям и другим сущностям базы данных;

- поэтапное внесение информации об происшествиях;

- разграничение доступа к различным функциям информационной системы.

Главная цель работы с картами - предоставить информацию о происшествии, его обнаружителе и местоположении с целью оперативной передачи этой информации экстренным службам ОАО «РЖД», а также принятия решения о ликвидации чрезвычайной ситуации.

## 1.4 Архитектура разрабатываемой информационной системы

При разработке информационной системы (ИС) важно учитывать ряд различных аспектов, таких как бизнес-требования, технологические возможности, уровень безопасности и масштабируемость системы. Архитектура информационной системы определяет структуру, компоненты и их взаимодействие, а также общую концепцию реализации системы.

Для разработки «информационной системы для экстренных служб ОАО «РЖД»» предлагается использовать клиент-серверную архитектуру.

Клиент-серверная архитектура - это модель распределенной системы, в которой клиентские устройства (клиенты) обращаются к серверу для получения данных, ресурсов или услуг. В этой архитектуре сервер является центральным узлом, который предоставляет услуги или данные клиентам, которые являются удаленными устройствами или приложениями.

Клиенты обычно отправляют запросы серверу, который обрабатывает эти запросы и отправляет обратно клиентам ответ. Сервер может быть физическим устройством или программным обеспечением, которое предоставляет определенные услуги или данные. Клиенты могут быть как обычные пользователи, так и другие программные приложения. Пример клиент-серверной архитектуры изображен на рисунке 2.



Рисунок 2. Клиент-серверная архитектура

Источник: Составлено автором

Бизнес логика в информационной системе - это структурированное описание процедур и процессов, которые используются для выполнения определенной задачи в организации. Функции информационной системы могут быть описаны как набор действий или операций, которые выполняются системой для обработки, хранения и передачи данных.

Бизнес логика функций информационной системы включает в себя слудующие аспекты:

- ведение реестра происшествий позволяет эффективно контролировать и обрабатывать информацию о происшествиях, обеспечивая оперативное принятие решений по их ликвидации;

- возможность изменения отображения страниц в зависимости от времени суток обеспечивает удобство использования системы и повышает комфорт пользователя;

- работа с картами позволяет визуализировать информацию о происшествиях, облегчая их анализ и принятие управленческих решений;

- отслеживание статуса происшествий помогает контролировать текущее состояние каждого происшествия и своевременно принимать меры по их ликвидации;

- информация об пользователях и ответственных лицах способствует учету участников процесса и обеспечивает прозрачность действий каждого из них;

- разграничение доступа к различным функциям системы обеспечивает безопасность данных и предотвращает несанкционированный доступ.

- оперативный поиск по происшествиям, пользователям и другим сущностям базы данных ускоряет процесс поиска необходимой информации и повышает эффективность работы с информационной системой;

- поэтапное внесение информации об происшествиях обеспечивает структурированное и последовательное заполнение данных, что повышает их качество и достоверность;

## 1.5 План работы с информационной системой

1. работник железной дороги, совершающий обход обнаруживает происшествие;

2. с помощью мобильный устройств, пример которых приведен в действующей информационной системе «представления данных в части обеспечения безопасности движения, мониторинга и управления чрезвычайными ситуациями», добавляет обнаруженное происшествие на карту (путем установки метки на интерактивной карте и внесении дополнительной информации);

3. информация поступает начальнику участка, который также на расширенной форме вносит данные о способе ликвидации происшествия;

4. экстренные службы участка получают информацию об происшествии и способе его ликвидирования, выезжают на место происшествия и ликвидируют происшествие, согласно способу ликвидации;

5. после ликвидации происшествия начальник участка закрывает форму происшествия, указав, что происшествие было ликвидировано. Если происшествие носит серьезный характер и ликвидировать его силами экстренных служб участка считается невозможным, начальник участка должен дополнить форму происшествия, внеся всю собранную информацию о нем.

## 1.6 Описание аварийных ситуаций

Аварийные ситуации на железных дорогах могут иметь серьезные последствия для безопасности пассажиров, грузов и персонала, а также привести к материальным убыткам и разрушениям. В зависимости от конкретной ситуации потери могут быть различными. Ниже предтавлены примеры аварийных ситуаций, которые будут отслеживаться информационной системой:

1. обрыв линии связи: приведет к нарушению связи между станциями, оперативному персоналу и диспетчерскому центру, что может замедлить реагирование на другие аварии и аварийные ситуации;

2. дефекты путей: могут привести к возможным авариям поездов, что повлечет за собой задержки в движении, увеличение времени в пути и потенциально крупные материальные убытки;

3. обрушение мостов и тоннелей: может вызвать блокировку железнодорожного движения, чрезвычайные ситуации с пассажирами и грузами, а также значительные затраты на восстановление инфраструктуры;

4. производственные аварии на железнодорожном участке: возможны выбросы опасных веществ, пожары, загрязнение окружающей среды, что потребует оперативных мер по ликвидации угрозы и восстановлению рабочей мощности;

5. незначительный прорыв водопровода на территории железнодорожной станции: может вызвать затопление путей, электрических установок, что повлечет за собой простои и ремонтные работы;

6. обрушение части платформы на станции из-за просадки грунта: Ситуация может привести к травмам пассажиров, закрытию станции, а также длительным процессам восстановления структур;

7. взрыв газопровода вблизи железнодорожной станции: Угроза для пассажиров, персонала, железнодорожных путей и построек, включая возможные пожары, повреждения инфраструктуры и чрезвычайные ситуации;

8. наводнение железнодорожного участка: может вызвать срыв путей, затопление электрических установок, остановку движения поездов, эвакуацию пассажиров, а также значительные финансовые потери.

Статистику по данным конкретным аварийным ситуациям предоставить сложно, так как каждое происшествие уникально и зависит от множества факторов, включая масштаб и скорость реагирования на чрезвычайные ситуации. Однако можно сказать, что аварии на железнодорожном транспорте могут иметь серьезные последствия как для инфраструктуры, так и для жизни и здоровья людей. Для минимизации потерь необходимо строгое соблюдение правил безопасности, оперативная реакция на чрезвычайные ситуации и постоянное обновление технического обслуживания инфраструктуры.

Данные аварийные ситуации будут классифицироваться по цветам (маркеров на карте), в зависимости от серьезности происшествия. Например, незначительные проблемы с линией связи между персоналом на станции, которые не приведут к материальным и другим потерям, будут классифицироваться как зеленые происшествия, аварийные ситуации которые приводят только к материальным потерям классифицируются как оранжевые (незначительный прорыв водопровода на территории железнодорожной станции), а ситуации которые приводят к большим материальным потерям и создают угрозу для жизни и здоровья людей (взрыв газопровода вблизи железнодорожной станции, обрушение части платформы на станции из-за просадки грунта), классифицируются как красные.

## 1.7 Обзор средств разработки

Для разработки информационной системы учета происшествий необходимо использовать современные фреймворки и библиотеки, которые будут реализовывать все актуальные требования к информационной системе.

В рамках данного обзора будут рассмотрены различные средства разработки, которые будут использованы при разработке информационной системы для учета происшествий для экстренных служб ОАО «РЖД».

**SqLite**

SqLite - это компактная встраиваемая однофайловая СУБД на языке С, которая предоставляет небольшой, быстрый и надежный способ хранения и доступа к данным. Она является одной из наиболее популярных баз данных, широко используемых во множестве приложений и операционных систем.

Преимуществами SqLite являются:

- хранение данных в одном файле;

- кроссплатформенность;

- малый размер сконфигурированной СУБД.

SQLite поддерживает стандартный язык запросов SQL для работы с данными. Это включает в себя возможность создания таблиц, вставки, обновления и удаления записей, а также выполнение сложных запросов с использованием операторов JOIN, GROUP BY и других. SQLite также поддерживает транзакции и сохранение точек восстановления, чтобы обеспечить целостность данных.

SqLite является бесплатным программным обеспечением. Данную СУБД можно использовать не только в коммерческих, но и в учебных целях.

**SqLiteStudio**

Программа SqLiteStudio является одной из многочисленных клиентов, предназначенных для управления базами данных SqLite. После установки вам будут доступны все инструменты разработки и работы с базами данных, такие как, создание, удаление и редактирование баз данных. Благодаря интерфейсу разработка баз данных SqLite станет намного удобнее. Установка приложения возможна после скачивания файла с официального сайта SqLiteStudio.

**Python**

Python – это высокоуровневый интерпретируемый язык программирования общего назначения с динамической типизацией. Python отличается простым и понятным синтаксисом, что делает его очень доступным для новичков, но в то же время он мощный и гибкий для опытных разработчиков.

Одним из ключевых принципов языка Python является читаемость кода, что делает его идеальным выбором для разработки чистого и понятного программного обеспечения. Python поддерживает объектно-ориентированное, императивное, функциональное и процедурное программирование.

Python имеет богатую стандартную библиотеку, которая включает множество модулей и инструментов для различных задач, начиная от работы с файлами и сетью, заканчивая разработкой веб-приложений и научными вычислениями. Благодаря этой библиотеке и активному сообществу разработчиков, Python стал популярным выбором для широкого круга приложений, включая веб-разработку, научные исследования, автоматизацию задач, разработку игр и многое другое.

**Django**

Django - это высокоуровневый веб-фреймворк, разработанный на языке Python, предназначенный для быстрой и эффективной разработки веб-приложений.

Особенности и функциональность Django:

1. url-маршрутизация: Django использует гибкую систему маршрутизации URL, которая позволяет разработчикам определять, какие действия должны выполняться при обращении к определенным URL-адресам;

2. шаблонизация: Django предлагает свой собственный язык шаблонов, который упрощает создание динамических веб-страниц. С его помощью вы можете создавать шаблоны HTML с динамическими данными и встроенными фильтрами для обработки данных;

3. обработка форм: Django предоставляет удобный набор инструментов для обработки форм на стороне сервера, что упрощает валидацию и сохранение данных, отправленных пользователем;

4. безопасность: Django имеет встроенные механизмы для защиты от основных угроз, таких как CSRF-атаки (межсайтовая подделка запроса) и инъекции SQL;

5. автоматическое администрирование: Django предлагает готовый административный интерфейс, который автоматически создается на основе определенных моделей вашего приложения;

6. база данных: Django поддерживает различные системы управления базами данных, включая PostgreSQL, MySQL, SQLite и другие. Он предоставляет ORM (Object-Relational Mapping), который позволяет работать с базой данных через объекты и запросы, а не с использованием прямых SQL-запросов.

Кроме того, Django имеет большое сообщество разработчиков, что обеспечивает доступ к множеству расширений, пакетов и ресурсов, которые можно использовать для ускорения и улучшения процесса разработки.

**Jinja2**

Jinja2 — это шаблонизатор Python, который обеспечивает простой и эффективный способ вставки динамических данных в HTML-шаблоны. В связке с Django, Jinja2 может использоваться в качестве альтернативы стандартному шаблонизатору Django.

Основные преимущества использования Jinja2:

1. удобство: Jinja2 предлагает более чистый и удобный синтаксис для вставки переменных, условий и циклов в шаблонах, что делает код более читаемым и легким для поддержки;

2. безопасность: Jinja2 предоставляет механизмы для предотвращения атак на безопасность, таких как защита от внедрения кода (XSS);

3. гибкость: Использование Jinja2 позволяет разработчикам выбирать между шаблонизаторами в зависимости от различных потребностей проекта.

**PyCharm**

PyCharm - это интегрированная среда разработки (IDE) для языка программирования Python. Это один из наиболее популярных инструментов для разработки Python-приложений, который предлагает множество функций и возможностей.

Одной из ключевых особенностей PyCharm является его мощный интеллектуальный анализ кода. Он предлагает подсветку синтаксиса, автодополнение кода, быстрый переход к определению функций и классов, проверку типов и многое другое. Благодаря этому разработчики могут работать более эффективно и быстро находить ошибки в своем коде.

PyCharm также предлагает широкий спектр инструментов для разработки веб-приложений на базе фреймворков, таких как Django, Flask и Pyramid. Он облегчает создание, отладку и развертывание веб-приложений, предоставляя шаблоны, инструменты миграции базы данных и поддержку статических файлов.

**Git**

Git - это распределенная система управления версиями, которая позволяет эффективно отслеживать изменения в исходном коде проекта. Система версий Git основана на нескольких ключевых концепциях, которые позволяют управлять историей изменений, объединять работы разработчиков и управлять конфликтами. Вот некоторые из основных аспектов системы версий Git:

1. репозиторий: Git использует репозиторий для хранения исходного кода проекта, а также всей его истории изменений. Репозиторий может быть локальным (на компьютере разработчика) или удаленным (на сервере);

2. ветвление: одна из ключевых концепций Git - ветвление. Разработчики могут создавать отдельные ветки для работы над определенной функциональностью или исправлением ошибок, не затрагивая основную ветку проекта;

3. слияние: после завершения работы в отдельной ветке разработчики могут слить изменения обратно в основную ветку. Git автоматически управляет слиянием изменений и разрешает конфликты при необходимости;

4. коммиты: для фиксации изменений разработчики используют коммиты. Каждый коммит содержит описание сделанных изменений, а также уникальный идентификатор;

5. откат изменений: Git позволяет откатывать изменения до определенного состояния с помощью команды git revert или git reset;

6. работа с удаленным репозиторием: Git поддерживает работу с удаленными репозиториями, что позволяет разработчикам обмениваться изменениями и совместно работать над проектом.

Система версий Git обладает большой гибкостью и мощными возможностями, делая ее популярным выбором для разработчиков при управлении версиями исходного кода.

**Gunicorn**

Gunicorn - это веб-сервер WSGI для Python, предназначенный для запуска веб-приложений, написанных на Python. Он работает как HTTP-сервер и обрабатывает запросы от клиентов, передавая их веб-приложению для выполнения, а затем возвращая результат клиенту.

Gunicorn поддерживает различные виды рабочих процессов, такие как sync, async и gevent, позволяя выбирать наиболее подходящий способ обработки запросов в зависимости от требований приложения. Он также обладает возможностью масштабирования, поддерживая запуск нескольких рабочих процессов, что позволяет обрабатывать большое количество запросов одновременно.

Gunicorn легко настраивается, имеет хорошую производительность и позволяет эффективно управлять веб-приложениями на Python. Он является популярным инструментом для развертывания и запуска веб-приложений и обеспечивает устойчивую работу веб-сервера.

# ГЛАВА 2 ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1 Требования к функциональным характеристикам

Информационная система должна обеспечивать возможность выполнения перечисленных ниже функций:

- ведение реестра происшествий;

- возможность изменения отображения страниц (для светлого времени суток – черный, для темного времени суток - светлый);

- работа с картами: добавление, изменение, удаление и просмотр каждого происшествия;

- отображение кроткой информации об происшествиях на карте;

- отслеживание статуса происшествий: возможность отслеживания текущего состояния каждого происшествия;

- отслеживание всех происшествий на общей карте;

- информация об пользователях;

- информация об ответственных лицах;

- разграничение доступа к различным функциям информационной системы;

- оперативный поиск по происшествиям, пользователям и другим сущностям базы данных;

- поэтапное внесение информации об происшествиях;

- разграничение доступа к различным функциям информационной системы.

## 2.2 Требования к организации входных данных

Входные данные информационной системы должны быть организованы в виде базы данных подключаемой непосредственно к самой системе. Также входные данные заполняются пользователями на формах, для дальнейшего добавления их в базу данных. Требования к типам входных данных будет отображено в пункте 2.2222 Описание основных сущностей базы данных.

## 2.3 Требования к организации выходных данных

Выходные данные программы должны быть организованны в виде цветных (в зависимости от степени происшествия) маркеров на интерактивной карте информационной системы.

## 2.4 UML диаграммы вариантов использования интерфейса управления пользователями для информационной системы

На данный момент в информационной системе существует 5 типов пользователей - это сотрудник управления персоналом, начальник участка, работник железной дороги, начальник экстренной службы участка, администратор.

На рисунке 3 предоставлена UML - диаграмма вариантов использования для сотрудника управления персоналом.



Рисунок 3. Сотрудник управления персоналом

Источник: Составлено автором

На Рисунке 4 предоставлена UML- диаграмма вариантов использования для начальника участка.



Рисунок 4. Начальник участка

Источник: Составлено автором

На Рисунке 5 предоставлена UML- диаграмма вариантов использования для работника железной дороги.



Рисунок 5. Работник железной дороги

Источник: Составлено автором

На Рисунке 6 предоставлена UML- диаграмма вариантов использования для начальника экстренной службы участка.



Рисунок 6. Начальник экстренной службы участка

Источник: Составлено автором

На Рисунке 7 предоставлена UML- диаграмма вариантов использования для администратора.



Рисунок 7. Администратор

Источник: Составлено автором

## 2.5 ER диаграмма базы данных

На Рисунке 8 предоставлена ER- диаграмма базы данных.

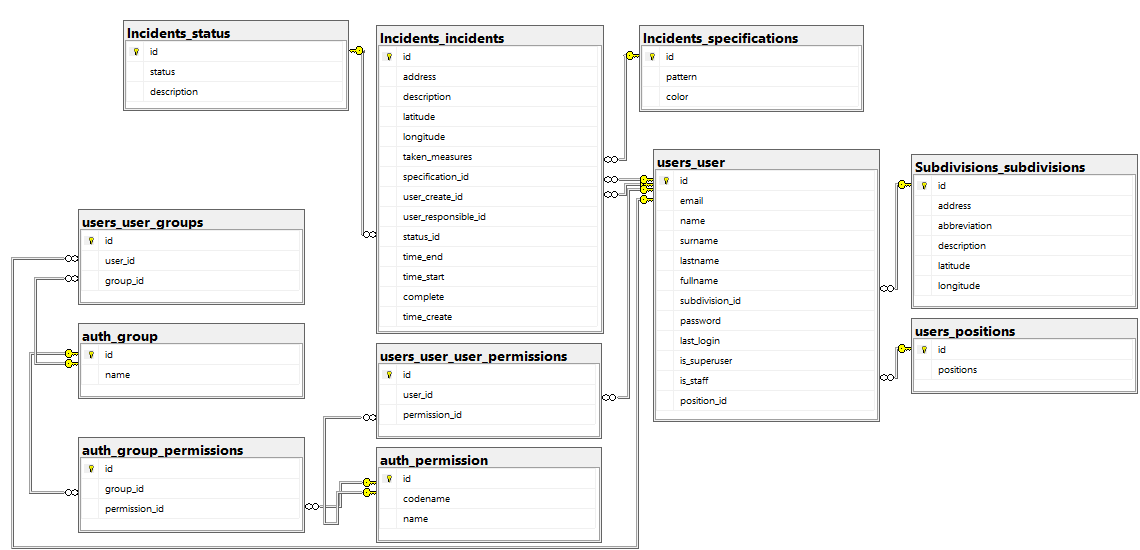


Рисунок 8. ER диаграмма базы данных

Источник: Составлено автором

## 2.6 Описание сущностей базы данных

В таблице 2.1 представлена таблица «Incidents\_incidents» базы данных информационной системы.

Таблица 2.1 – Incidents\_incidents

| **Название поля** | **Тип данных** | **Размер поля** |
| --- | --- | --- |
| id | integer | - |
| address | varchar | 255 |
| description | varchar | 255 |
| latitude | Real | - |
| longitude | Real | - |
| taken\_measures | varchar | 255 |
| specification\_id | bigint | - |
| user\_create\_id | bigint | 5 |
| user\_responsible\_id | bigint | 255 |
| status\_id | bigint | - |
| time\_end | datetime | - |
| time\_start | datetime | - |
| complete | bool | - |
| time\_create | datetime | - |

В таблице 2.2 представлена таблица «Incidents\_specifications» базы данных информационной системы.

Таблица 2.2 – Incidents\_specifications

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Размер поля** |
| id | integer | - |
| Pattern | varchar | 100 |
| color | varchar | 20 |

В таблице 2.3 представлена таблица «Incidents\_status» базы данных информационной системы.

Таблица 2.3 – Incidents\_status

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Размер поля** |
| id | integer | - |
| status | varchar | 100 |
| description | varchar | 100 |

В таблице 2.4 описана таблица «Subdivisions\_subdivisions» базы данных информационной системы.

Таблица 2.4 – Subdivisions\_subdivisions

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Размер поля** |
| id | integer | - |
| address | varchar | 255 |
| abbreviation | varchar | 10 |
| description | varchar | 255 |
| latitude | Real | - |
| longitude | Real | - |

В таблице 2.5 описана таблица «users\_positions» базы данных информационной системы.

Таблица 2.5 – users\_positions

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Размер поля** |
| id | integer | - |
| positions | varchar | 255 |

В таблице 2.6 описана таблица «users\_user\_groups» базы данных информационной системы.

Таблица 2.6 – users\_user\_groups

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Размер поля** |
| id | integer | - |
| user\_id | bigint | - |
| group\_id | integer | - |

В таблице 2.7 представлена таблица «users\_user» базы данных информационной системы.

Таблица 2.7 – users\_user

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Размер поля** |
| id | integer | - |
| email | varchar | 255 |
| name | varchar | 50 |
| surname | varchar | 50 |
| lastname | varchar | 50 |
| fullname | varchar | 255 |
| subdivision\_id | bigint | - |
| password | varchar | 128 |
| last\_login | datetime | - |
| is\_superuser | bool | - |
| is\_staff | bool | - |
| position\_id | bigint | - |

В таблице 2.8 представлена таблица «users\_user\_user\_permissions» базы данных информационной системы.

Таблица 2.8 – users\_user\_user\_permissions

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Размер поля** |
| id | integer | - |
| user\_id | bigint | - |
| permission\_id | integer | - |

В таблице 2.9 представлена таблица «auth\_group» базы данных информационной системы.

Таблица 2.9 – auth\_group

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Размер поля** |
| id | integer | - |
| name | varchar | 150 |

В таблице 2.10 описана таблица «auth\_group\_permissions» базы данных информационной системы.

Таблица 2.10 – auth\_group\_permissions

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Размер поля** |
| id | integer | - |
| group\_id | integer | - |
| permission\_id | integer | - |

В таблице 2.11 описана таблица «auth\_permission» базы данных информационной системы.

Таблица 2.11 – auth\_permission

|  |  |  |
| --- | --- | --- |
| **Название поля** | **Тип данных** | **Размер поля** |
| id | integer | - |
| codename | varchar | 100 |
| name | varchar | 255 |

## 2.7 Описание функционала информационной системы

Для запуска информационной системы необходимо ввести ее адрес в браузере устройства.

**Авторизация**

После перехода по ссылке информационной системы появляется форма авторизации. Для выполнения входа в информационную систему необходимо ввести логин и пароль, после чего нажать кнопку «Войти». Пример формы авторизации представлен на рисунке 2.7.1.

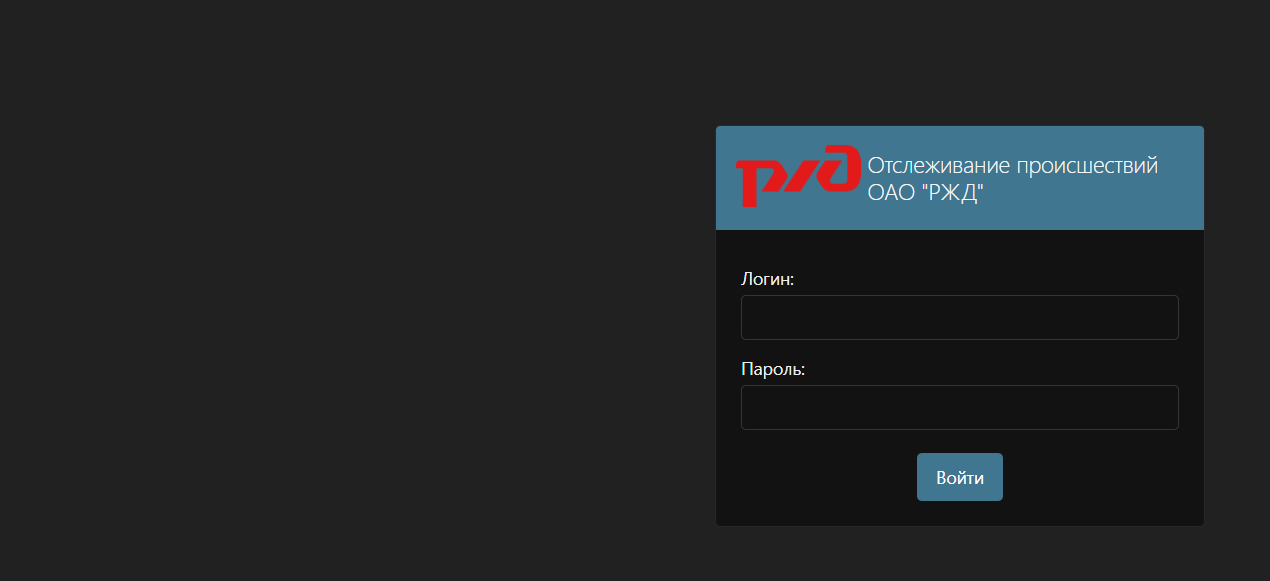


Рис. 2.7.1 – Форма авторизации

**Главная страница информационной системы**

После авторизации нас перенаправит на главную страницу информационной системы. Данная страница содержит информацию о последних действиях активного пользователя, а также ссылки на доступный функционал пользователя. Для каждого пользователя есть свои права доступа, в зависимости от них отображается доступный функционал. Пример отображения главной страницы приведен на рисунке 2.7.2.

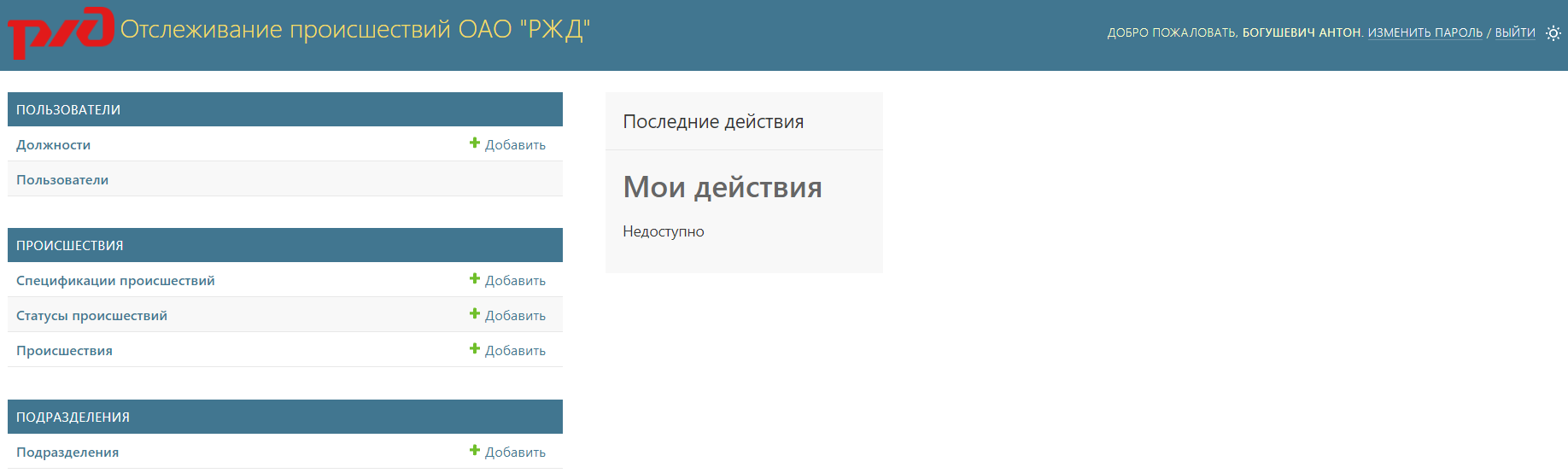


Рис. 2.7.2 – Главная страница информационной системы

**Должности**

Блок «ПОЛЬЗОВАТЕЛИ» содержит ссылки на просмотр информации, связанной с пользователями, а именно: «Должности» и «Пользователи». Для просмотра информации о должностях необходимо перейти по ссылке «Должности» в левой части страницы. Осуществится переход на страницу всех должностей, которые доступны в информационной системе. Пример отображения страницы «Должности» представлен на рисунке 2.7.3.

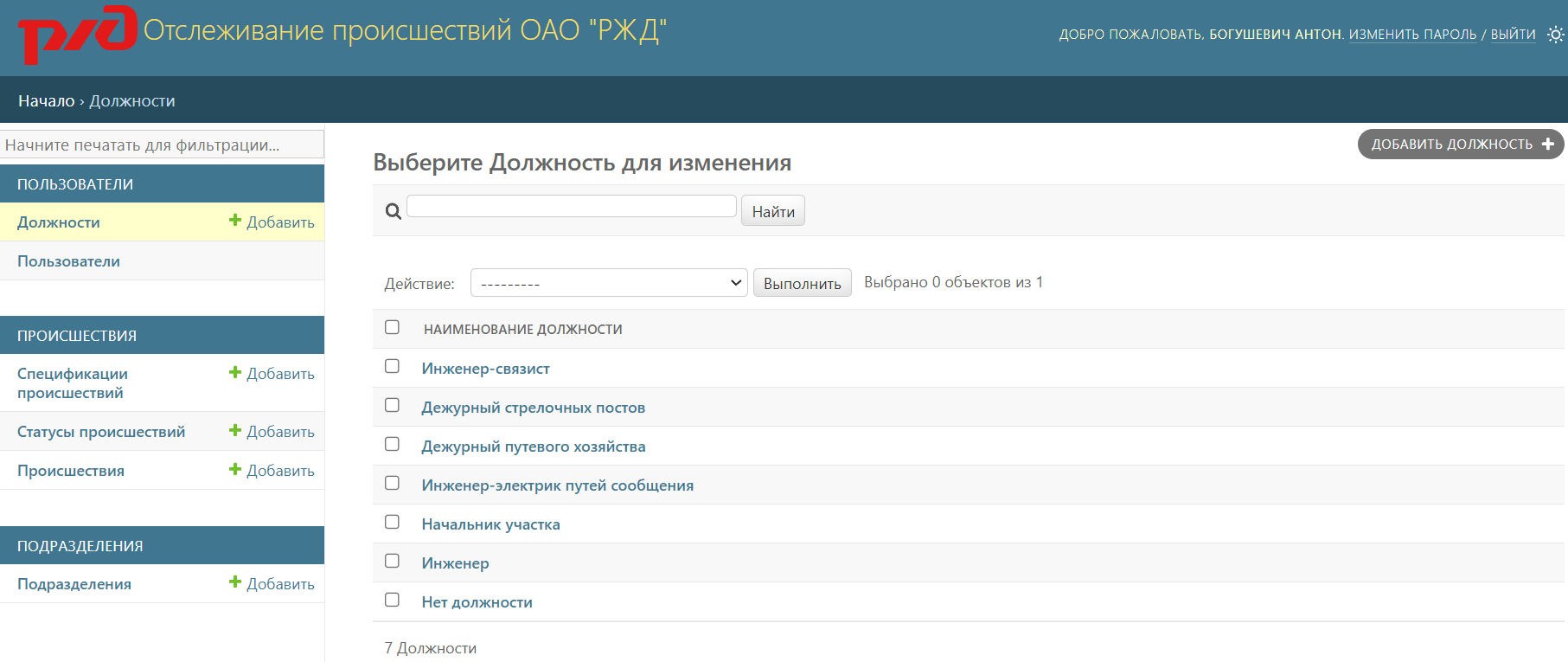


Рис. 2.7.3 – Страница «Должности»

Для добавления новой должности в информационной системе необходимо нажать на ссылку «Добавить» или «Добавить должность», которые находится в левой и правой части страницы. Пример страницы добавления должностей представлен на рисунке 2.7.4.

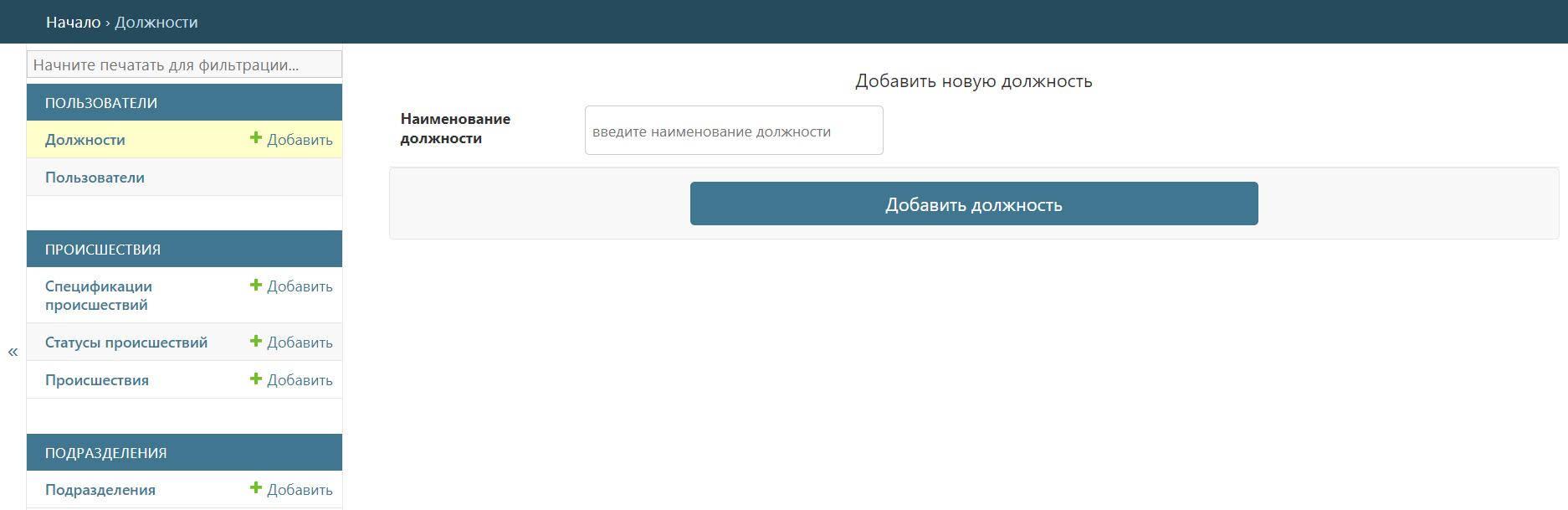


Рис. 2.7.4 – Страница добавление должности

После ввода новой должности необходимо нажать на кнопку «Добавить должность». После сохранения должности в базе данных информационной системы нас перенаправит на страницу просмотра созданной должности, пример которой изображен на рисунке 2.7.6. Также новая должность отобразится на странице «Должности». Пример добавления новой должности приведен на рисунках 2.7.5 и 2.7.7.

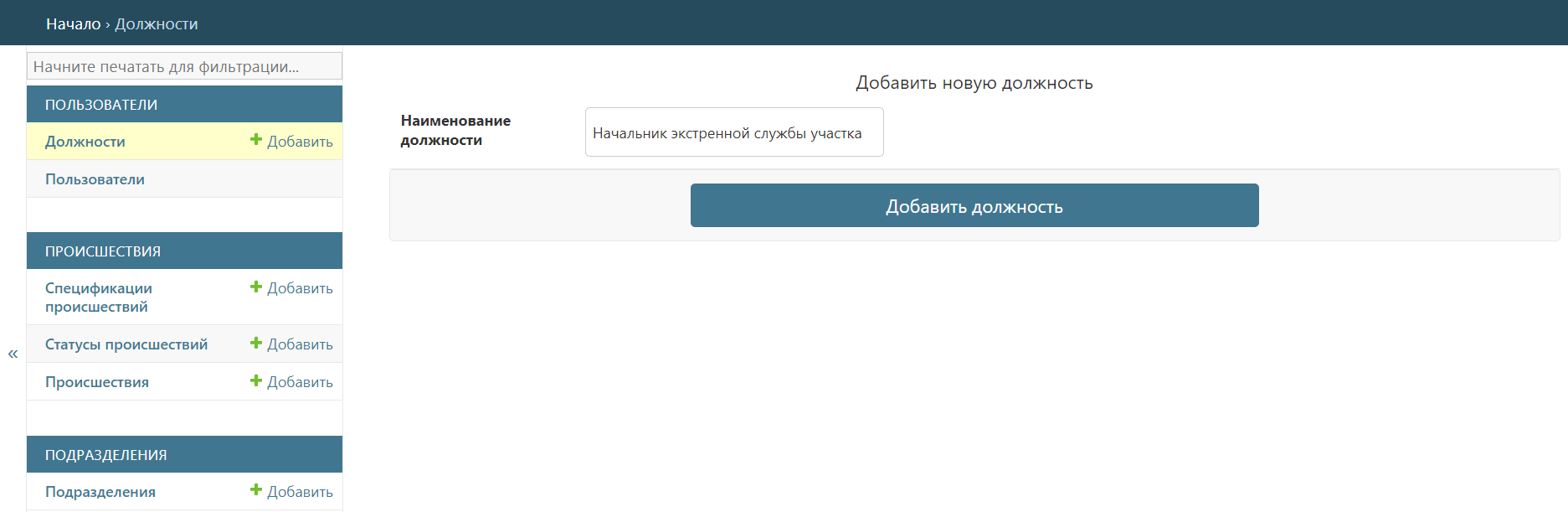


Рис. 2.7.5 – Добавление новой должности «Начальник экстренной службы участка»

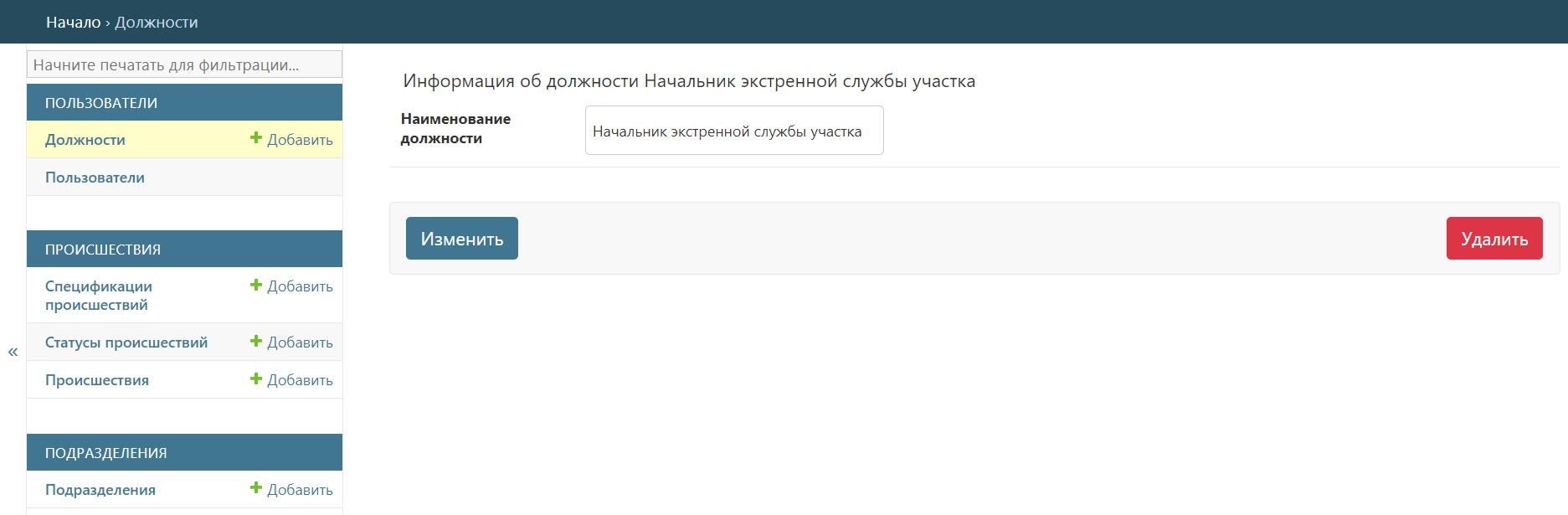


Рис. 2.7.6 – Просмотр созданной должности

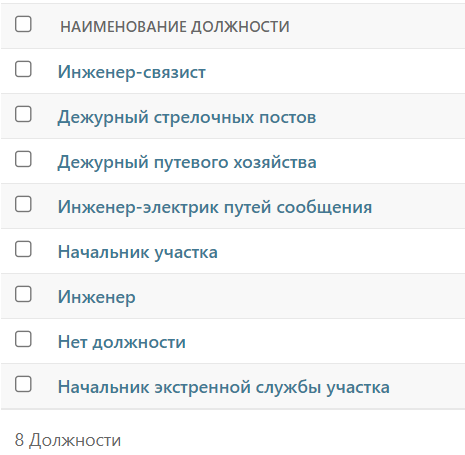


Рис. 2.7.7 – Отображение созданной должности на странице «Должности»

Для изменения информации о выбранной должности, необходимо нажать кнопку «Изменить» на странице «Просмотр должности», пример которой изображен на рисунке 2.7.6. После нас перенаправит на страницу изменения должности, изображенной на рисунке 2.7.8. На данной странице находится поле формы и информация, которую необходимо изменить, в этом поле. Для сохранения изменений необходимо нажать кнопку «Сохранить изменения должности». Наша информация об должности обновится и также отобразится на странице «Должности». Пример изменения информации о должности изображен на рисунках 2.7.9 и 2.7.10.

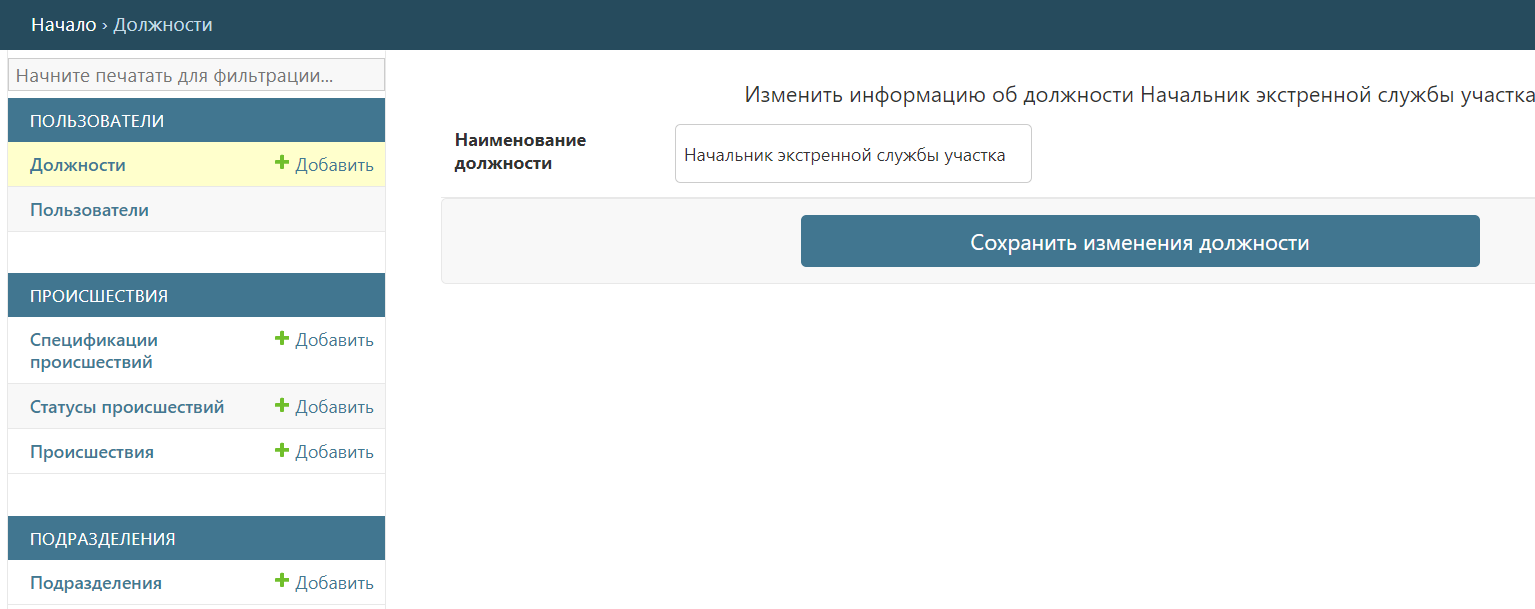


Рис. 2.7.8 – Отображение страницы изменения информации о должности

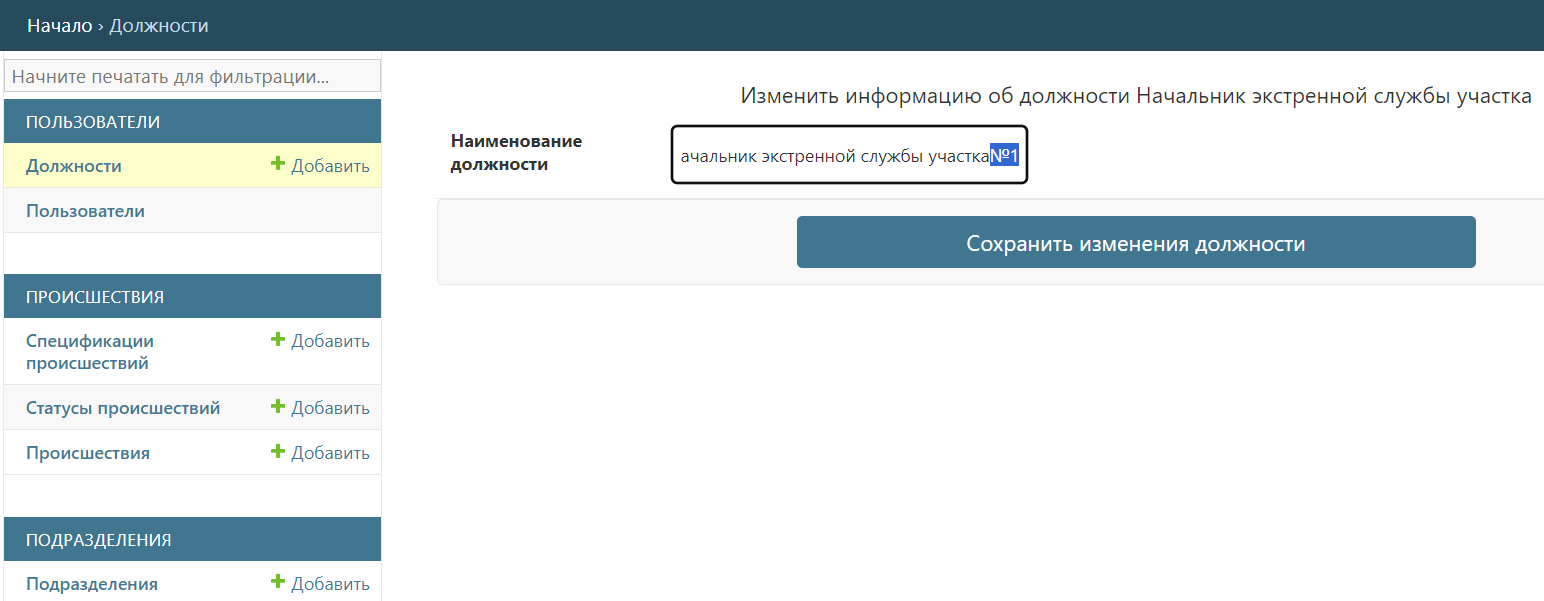


Рис. 2.7.9 – Изменение информации о должности

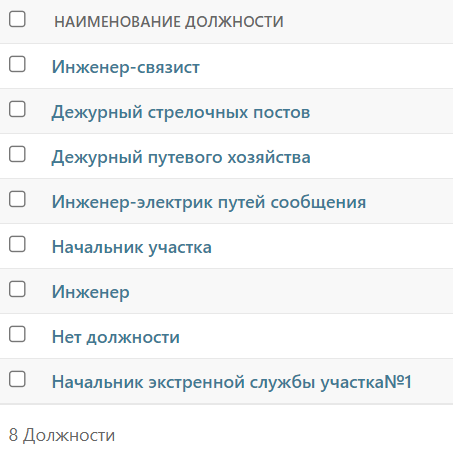


Рис. 2.7.10 – Сохраненные изменения информации о должности

Для удаления должности необходимо нажать кнопку «Удалить» на странице «Просмотр должности», пример которой изображен на рисунке 2.7.6. Наша должность удалится и перестанет отображаться на странице «Должности».

**Пользователи**

Для просмотра информации о пользователях информационной системы необходимо перейти по ссылке «Пользователи» в левой части страницы. Осуществится переход на страницу всех пользователей. Пример отображения страницы «Пользователи» представлен на рисунке 2.7.11.

## 

## 