

Разработка информационной модели системы контроля поручений

Король Александр Сергеевич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

В данной статье описывается создание системы контроля поручений, которая позволяет назначать определенные задачи, следить за их исполнением, переназначать поручения другим сотрудникам, декомпозировать задачи на более мелкие и вести соответствующую отчетность.

Ключевые слова: информационная система, контроль поручений, архитектура баз данных, модели разработки, web-приложение, система аутентификации.

Development of an information model for an order control system

Korol Alexandr Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the creation of a task control system that allows you to assign specific tasks, monitor their execution, reassign tasks to other employees, decompose tasks into smaller ones, and keep relevant reports.

Keywords: information system, task control, database architecture, development models, web-application, authentication system.

Научный руководитель:

Баженов Руслан Иванович

Приамурский государственный университет им. Шолом-Алейхема

к.п.н., доцент, зав. кафедрой информационных систем, математики и правовой информатики

Информационные технологии уже давно оказывают огромное влияние на все сферы нашей жизни, будь то социальное взаимодействие или новые научные открытия. Практически все, что можно автоматизировать, уже автоматизировано или идет к этому. Информации на бумажных носителях все меньше и меньше, а большинство людей хранят свою информацию у себя на компьютерах, смартфонах или на цифровом облаке.

Технический прогресс не обошел стороной и бизнес сферу. Большинство крупных компаний имеют свой отдел разработки программного обеспечения или тратят большие деньги, покупая информационные системы

у соответствующих компаний. Множество существующих информационных систем автоматизируют тот или иной процесс, делая его более удобным, быстрым и с минимальным влиянием человеческого фактора.

Одной из таких систем можно смело назвать систему контроля поручений, позволяющая назначать определенные задачи, следить за их исполнением, переназначать поручения другим сотрудникам, декомпозировать задачи на более мелкие и вести соответствующую отчетность.

Примерами таких информационных систем на рынке являются: «CompanуMedia», «LanDocs», «OptimaWorkflow», «Босс-Референт», «ДЕЛО», «Jira».

Целью исследования является разработка информационной системы, выполняющая следующие задачи:

- разделённый доступ, в соответствии с ролевой моделью пользователя;
- назначение поручений;
- осуществление контроля над поручениями тем или иным сотрудникам;
- ведение отчетности по проделанным работам;
- накапливать знания для решения в будущем проблем, связанных с подобным родом поручений;
- быть гибкой, для дальнейших изменений и улучшений.

Для достижения поставленной цели нами были выполнены следующие задачи:

- проанализирована литература, связанная с темой исследования;
- изучены методики и технологии, необходимые для разработки информационной системы;
- изучена предметная область, проанализированы автоматизируемые процессы;
- проанализированы существующие решения;
- разработана архитектура информационной системы;
- разработана структура баз данных;
- изучены и внедрены современные методики аутентификации и авторизации;
- разработана информационная система контроля поручений;

Во время разработке информационной системы нами был изучен ряд литературных источников, в виде научных статей, учебников и других подобных работ. Данные работы условно можно разбить на группы, где каждая группа посвящена изучению того или иного аспекта, связанного с разработкой информационной системы контроля поручений.

В процессе разработки были изучены работы по следующим темам:

- моделирование информационных процессов;
- основные технологии для web-разработки;

- системы управления базами данных;
- операционные системы;
- классический и электронный документооборот.

Тему моделирования информационных процессов хорошо описывают В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина в своей работе «Проектирование информационных систем» [6]. С. В. Маклаков посвятил свою работу правильному моделированию бизнес-процессов [16].

О web-технологиях много пишут отечественные авторы. О языке программировании PHP пишут: Д. Н. Колисниченко [13, 14].

Зарубежные авторы также делятся опытом в сфере web-разработки. О PHP писали Э. Гутманс, С. Баккен, Д. Ретанс [27] и Д. Шлоссейгл [8]. О технологиях клиентской web-разработки (HTML, CSS, JavaScript и др.) пишут Д. Дакетт [9], Э. Мейер [18] и Р. Никсон [19].

Основы разработки баз данных затронули С. Д. Кузнецов в своей работе «Основы современных баз данных» [15] и М. П. Малыгина в работе «Базы данных: основы, проектирование, использование» [17].

О языке запросов SQL пишут в своих работах как отечественные авторы: Я. Э. Гречка [7], И. Ф. Астахова [2,3], Ф. Андон [1], так и зарубежные авторы: Д. Селко [23], К. Фиайли [24].

Об операционных системах Linux пишут следующие авторы: Н. Н. Иванов [10], Д. Валади [5]. Об операционной системе Windows можно прочитать в книгах следующих авторов: Д. Н. Колисниченко [13].

О делопроизводстве в своей работе «Эффективное делопроизводство» писал И. В. Клоков [12]. Основные моменты, связанные с электронным документооборотом, описали следующие авторы: С. В. Клименко [11], Ю. Е. Поляк [21], В. В. Персианов, Е. З. Киреева, М. Н. Казакова [20].

Рассмотрим несколько современных методик разработки информационных систем:

Модель Waterfall.

В данной модели процесс разработки представлен в виде «потока», последовательно проходящий через этапы разработки ИС.

Технология RAD.

В основе имеет спиральную модель жизненного цикла. Жизненный цикл состоит из многократно, итеративно-повторяющихся четырех стадий разработки:

- анализ требований и планирование;
- проектирование;
- реализация;
- внедрение версии.

Методология MSF.

MSF (Microsoft Solutions Framework) – пакет руководств, разработанный компанией Microsoft. Данные руководства описывают эффективное проектирование, разработку и сопровождение программных решений, которые были построены на базе их технологий.

Agile-Методики.

Представляет собой семейство подходов к разработке ПО, с использованием итеративной разработки, динамического определения требований и их реализации, путем постоянного взаимодействия рабочих групп, в состав которых входят специалисты различных областей и профилей.

Технология XP.

Как и RAD в основе имеет спиралевидную модель жизненного цикла.

Методология RUP.

В основе лежит спиральная модель жизненного цикла. Основным языком моделирования – Unified Modelling Language (UML).

Методология DSDM.

Метод разработки динамических систем (англ. Dynamic Systems Development Method, DSDM) основан на концепции быстрой разработки приложений (RAD). Метод DSDM – это итеративный и инкрементный подход разработки программного обеспечения, который придает особое значение продолжительному участию в процессе заказчика системы.

Методология Scrum.

Методология управления проектами, активно применяющаяся при разработке информационных систем для гибкой разработки программного обеспечения.

После долгого и тщательного анализа методов, нами был выбран технология XP, отвечающий всем необходимым нам критериям при разработке.

В своей основе информационная системы представляет собой клиент-серверное приложение, а точнее web-приложение. Схематично архитектура изображена на рисунке 1:

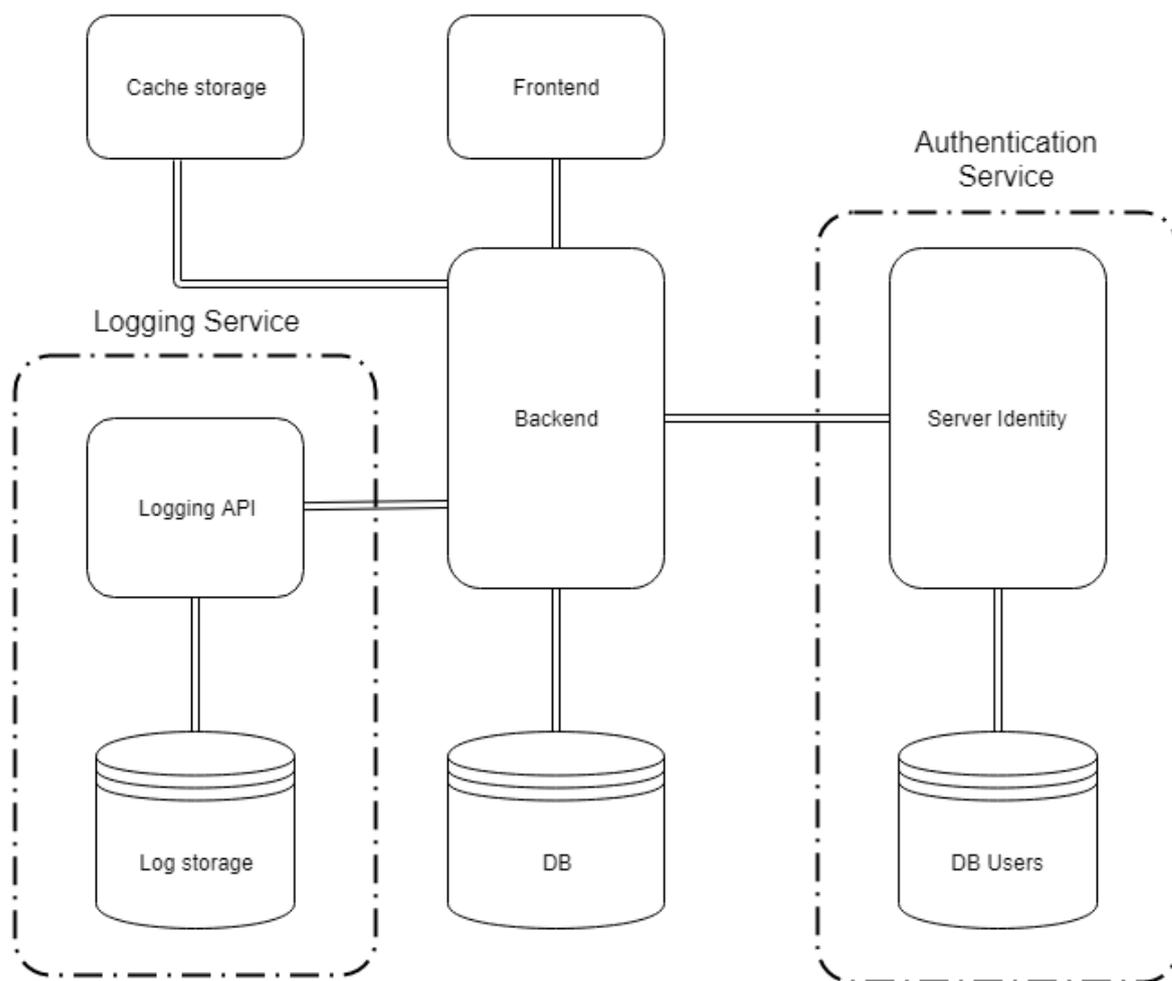


Рис. 1. Архитектура информационной системы контроля поручений

Рассмотрим основные компоненты данной архитектуры:

Backend – в данной схеме является ядром информационной системы. Основная часть бизнес-логики определена именно в этом компоненте. Здесь же осуществляется интеграция с внешними сервисами. Является самым большим компонентом. Архитектура этого компонента будет приведена позднее.

Технологии для реализации: ASP.NET Core, C#.

DB – компонент базы данных (БД) для компонента Backend. В данном компоненте хранится вся основная информация, связанная с основными процессами ИС. Например, данные о сотрудниках, иерархия должностей, данные о поручениях. Не обязательно является одним экземпляром БД, вполне может представлять из себя более сложные структуры, например, кластер БД для отказоустойчивости или решения для горизонтального масштабирования БД с целью балансировки нагрузки.

Технологии для реализации: MS SQL Server 2017

Frontend – представляет собой клиентскую часть приложения. В эту часть входит интерфейс пользователя и логика приложения, которая может быть реализована на клиентской части. Клиентская часть по необходимости

обращается к серверной части (на схеме Backend) для получения или отправки необходимых данных.

Технологии для реализации: HTML, CSS, Js, VueJS.

AuthenticationService – микросервис для аутентификации и авторизации пользователей. Состоит из Web API и БД. Выделено в отдельный микросервис, для возможности замены, с целью реализации другого способа аутентификации и авторизации.

Технологии для реализации: ASP.NET Core, C#, MS SQL Server 2017.

LoggingService – микросервис, для осуществления логгирования (протоколирования) операций в ИС. Logging API принимает на вход сигнал от Backend-а и записывает в свою собственную БД LogStorage.

Технологии для реализации: ASP.NET Core, C#, MS SQL Server 2017.

Cachestorage – хранилище кеша, служит для ускорения выполнения запросов. Особенно актуально для запросов на чтения информации, которая редко меняется, при этом запрашивается слишком часто. Механизм взаимодействия: если информация есть в Cachestorage тогда она сразу отдается клиенту, если нет, Backend выгружает информацию из DB, записывает в Cachestorage и отдает Frontend-у. Если поступил запрос на изменения, то информация записывается последовательно в DB, а затем в Cachestorage.

В качестве решения обычно используют быстрые БД, которые хранят данные в оперативной памяти, за счет чего обработка запросов совершается быстрее.

Технологии для реализации: Redis.

Архитектура компонента Backend

В основе данного компонента лежит Onion архитектура.

Термин «OnionArchitecture» («луковая» архитектура) был предложен Джеффри Палермо (JeffreyPalermo) еще в 2008 году. Спустя годы данная концепция стала довольно популярной и является одной из наиболее применяемых типов архитектуры при построении приложения на ASP.NET.

Onion-архитектура представляет собой разделение приложения на уровни. При чем есть один независимый уровень, который находится в центре архитектуры. От этого уровня зависит второй уровень, от второго - третий и так далее. То есть получается, что вокруг первого независимого уровня наслаивается второй-зависимый. Вокруг второго наслаивается третий, который также может зависеть и от первого. Образно это может быть выражено в виде лука, в котором также есть сердцевина, вокруг которого наслаиваются все остальные слои, вплоть до шелухи.

Количество уровней может отличаться, но в центре всегда находится модель домена (DomainModel), то есть те классы моделей, которые используются в приложении и объекты которых хранятся в базе данных (Рисунок 2).

Первый уровень вокруг модели домена образуют интерфейсы, которые управляют работой с моделью домена. Обычно это интерфейсы репозитория, через которые мы взаимодействуем с базой данных.

Внешний уровень представляет такие компоненты, которые очень часто изменяются.

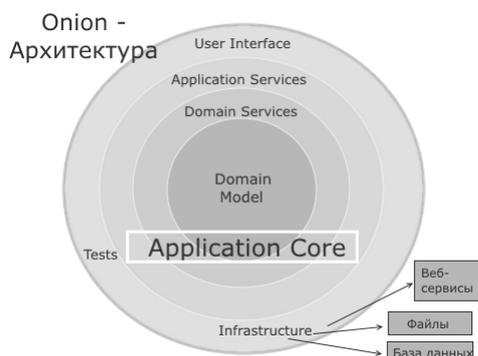


Рис. 2. Onion-архитектура

Также стоит отметить, что все внешние хранилища, как базы данных, файлы, внешние веб-сервисы, от которых мы можем получать данные, - все это является внешним по отношению к архитектуре.

После определения архитектуры системы рассмотрим архитектуру главного компонента ИС, отвечающий непосредственно за контроль поручений (Рисунок 3).

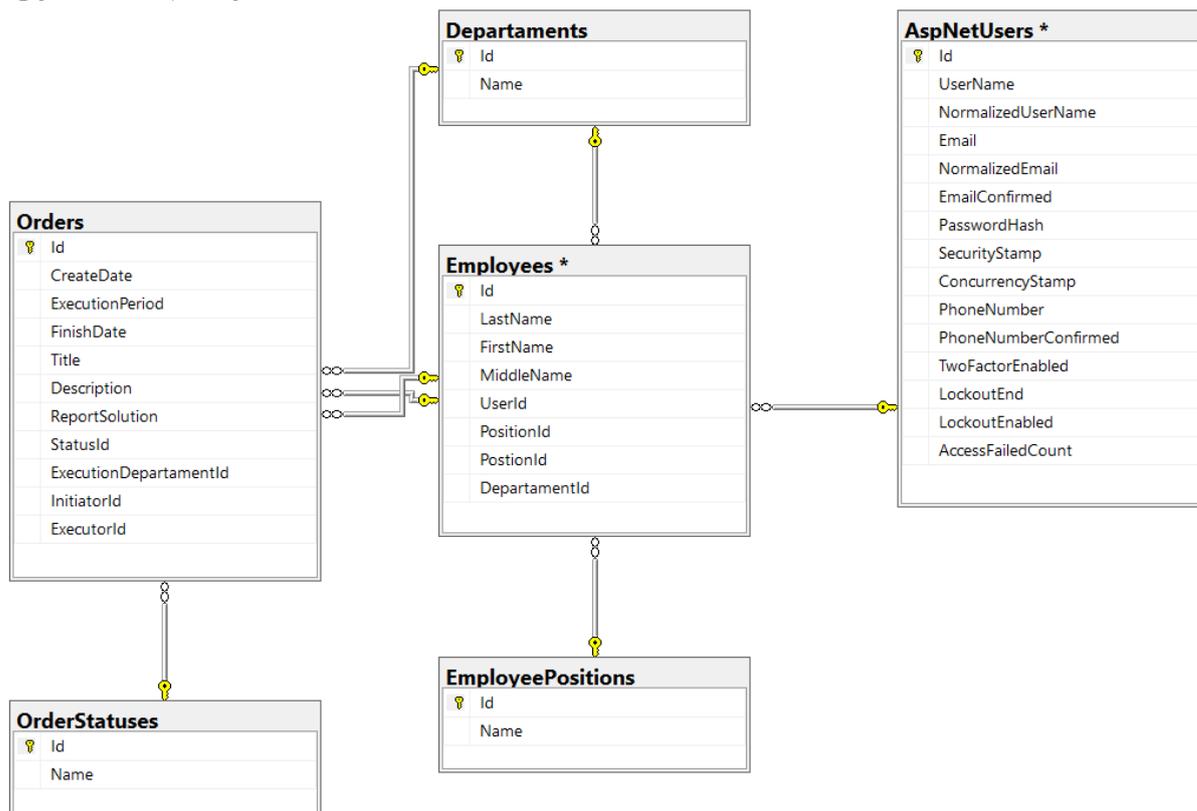


Рис. 3. Архитектура основной базы данных (компонент DB)

Стоит отметить, что таблица AspNetUsers опциональная и в данном является сущностью другой БД. Показана на схеме, каким образом связаны сущности пользователя и сотрудник. Хотя есть вариант с использованием

таблицы Users в основной БД, но такое решение нас жестко привяжет к выбранному способу аутентификации (Таблицы 1–5).

Таблица 1 – Описание таблицы Employees

Название атрибута	Тип данных	Описание
Id	INT	Суррогатный первичный ключ
LastName	NVARCHAR	Фамилия
FirstName	NVARCHAR	Имя
MiddleName	NVARCHAR	Отчество
UserId	NVARCHAR	Идентификатор пользователя в системе авторизации
PositionId	INT	Внешний ключ должности сотрудника
DepartmentId	INT	Внешний ключ отдела сотрудника

Таблица 2 – Описание таблицы EmployeePositions

Название атрибута	Тип данных	Описание
Id	INT	Суррогатный первичный ключ
Name	NVARCHAR	Наименование должности

Таблица 3 – Описание таблицы Departments

Название атрибута	Тип данных	Описание
Id	INT	Суррогатный первичный ключ
Name	NVARCHAR	Наименование подразделения или отдела

Таблица 4 – Описание таблицы OrderStatuses

Название атрибута	Тип данных	Описание
Id	INT	Суррогатный первичный ключ
Name	NVARCHAR	Наименование статуса

Таблица 5 – Описание таблицы Orders

Название атрибута	Тип данных	Описание
Id	INT	Суррогатный первичный ключ
CreateDate	DATETIME2	Дата создания
ExecutionPeriod	DATETIME2	Нормативный срок исполнения
FinishDate	DATETIME2	Дата выполнения
Title	NVARCHAR	Заголовок поручения
Description	NVARCHAR	Подробное описание поручения
ReportSolution	NVARCHAR	Отчет о выполнении

StatusId	INT	Внешний ключ статуса поручения
ExecutionDepartmentId	INT	Внешний ключ исполняющего департамента
InitiatorId	INT	Внешний ключ сотрудника инициатора
ExecutorId	INT	Внешний ключ сотрудника исполнителя

При попытке открыть веб-страницу информационной системе, проводится проверка на наличие JWT-токена аутентификации в хранилище сессии браузера. Если данный токен не обнаружен, тогда пользователь переводится на страницу ввода учетных данных. Пример окна аутентификации приведен на рисунке 4 ниже.

Система контроля поручений [Мои задачи](#) [Мои сотрудники](#)

Вход на сайт

Имя пользователя

Пароль

[Войти](#)

© 2019 - SRC

Рис. 4. Форма аутентификации

Для сотрудника и руководителя подразделения список задач выглядит одинаково, за исключением, что руководитель видит задачи всего подразделения. На рисунке 5 изображен пример отображения задач для руководителя.

Нажав на кнопку «подробнее», открывается карточка поручения, где при наличии определенных прав можно вносить изменения в поручения. Например, написать решение, изменить статус или выбрать другого исполнителя. Пример карточки поручения изображен на рисунке 6.

Окно с созданием заявки, выглядит точно также, за исключением того, что в полях пустые значения, которые пользователь должен ввести сам.

Для руководителей доступна опция просмотра списка сотрудников, который находится у них в подчинении и к чьим поручениям у него имеется доступ. Пример на рисунке 7 ниже.

Для администраторов системы создана панель администратора, которая позволяет регистрировать пользователей, менять составы подразделений,

назначать руководителей подразделения, просматривать и менять поручения, вести справочники возможных статусов поручения и списка должностей. Одна из страниц данной панели администратора представлена на рисунке 8.

Система контроля поручений [Входящие поручения](#) [Исходящие поручения](#) [Мои сотрудники](#) boss ▾

[Создать задачу](#) « 1 2 3 »

#	Название	Описание	Статус	Инициатор	Исполнитель	Решение	Действие
1	Спланировать доставку товара	Необходимо рассчитать наилучшие маршруты для доставки нашего товара. Также стоит рассчитать наиболее выгодное сотрудничество с доступными транспортными компаниями.	В очереди	Иванов Иван Иванович	Петров Петр Петрович		Подробнее
2	Спланировать доставку товара	Необходимо рассчитать наилучшие маршруты для доставки нашего товара. Также стоит рассчитать наиболее выгодное сотрудничество с доступными транспортными компаниями.	В работе	Иванов Иван Иванович	Петров Петр Петрович		Подробнее
3	Спланировать доставку товара	Необходимо рассчитать наилучшие маршруты для доставки нашего товара. Также стоит рассчитать наиболее выгодное сотрудничество с доступными транспортными компаниями.	Приостановлена	Иванов Иван Иванович	Петров Петр Петрович		Подробнее
4	Спланировать доставку товара	Необходимо рассчитать наилучшие маршруты для доставки нашего товара. Также стоит рассчитать наиболее выгодное сотрудничество с доступными транспортными компаниями.	В очереди	Иванов Иван Иванович	Сергеев Сергей Сергеевич		Подробнее
5	Спланировать доставку товара	Необходимо рассчитать наилучшие маршруты для доставки нашего товара. Также стоит рассчитать наиболее выгодное сотрудничество с доступными транспортными компаниями.	В работе	Иванов Иван Иванович	Федоров Федр Федорович		Подробнее
6	Спланировать доставку товара	Необходимо рассчитать наилучшие маршруты для доставки нашего товара. Также стоит рассчитать наиболее выгодное сотрудничество с доступными транспортными компаниями.	В очереди	Иванов Иван Иванович	Федоров Федр Федорович		Подробнее

Рис. 5.Окно со списком задач

Информация о поручении

Дата создания

12.12.2019 22:37:03,532

Дата выполнения

дд.мм.гггг --:--

Заголовок

Спланировать доставку товара

Описание

Необходимо рассчитать наилучшие маршруты для доставки нашего товара. Также стоит рассчитать наиболее выгодное сотрудничество с доступными транспортными компаниями.

Решение

Статус задачи

В очереди

Поручен отделу

Транспорта и логистики

Инициатор

Иванов Иван Иванович

Исполнитель

Петров Петр Петрович

Сохранить

[Вернуться к списку задач](#)

Рис. 6. Окно поручения

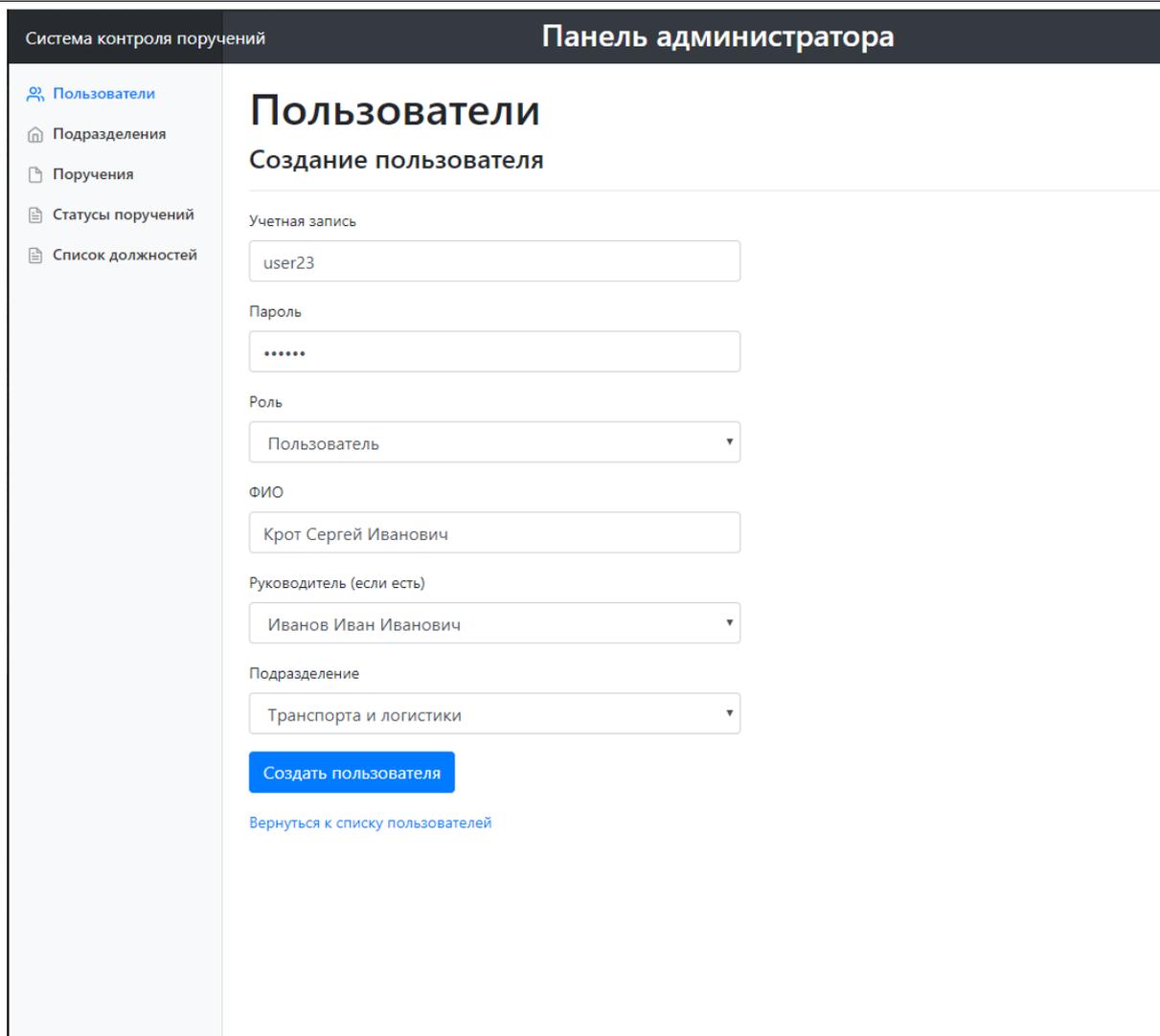
#	Фамилия	Отдел	Менеджер	Назначено задач
4	Петров Петр Петрович	Транспорта и логистики	Иванов Иван Иванович	3
5	Сергеев Сергей Сергеевич	Планирование закупок	Иванов Иван Иванович	1
6	Федоров Федр Федорович	Цифровые инновации	Иванов Иван Иванович	2
7	Иванов Иван Иванович	Транспорта и логистики	Иванов Иван Иванович	0
9	Сергеев Сергей Сергеевич	Транспорта и логистики	Иванов Иван Иванович	0
10	Федоров Федр Федорович	Транспорта и логистики	Иванов Иван Иванович	0

Рис. 7. Список сотрудников в подчинении у руководителя подразделения

Система контроля поручений		Панель администратора					admin
Пользователи		Список пользователей Добавить пользователя					
Логин	ФИО	Роль	Отдел	ФИО руководителя	Действия		
admin		admin			Изменить	Удалить	
admin2		admin2			Изменить	Удалить	
manager	Иванов Иван Иванович	manager			Изменить	Удалить	
employee1	Петров Петр Петрович	user	Транспорта и логистики		Изменить	Удалить	
employee2	Сергеев Сергей Сергеевич	user	Цифровые инновации	Иванов Иван Иванович	Изменить	Удалить	
employee3	Федоров Федр Федорович	user	Цифровые инновации	Иванов Иван Иванович	Изменить	Удалить	
manager2	Сидоров Сидр Сидорович	manager	Транспорта и логистики		Изменить	Удалить	
employee4	Александров Александр Александрович	user	Транспорта и логистики	Сидоров Сидр Сидорович	Изменить	Удалить	
employee5	Алексеев Алексей Алексеевич	user	Транспорта и логистики	Сидоров Сидр Сидорович	Изменить	Удалить	
employee6	Никитенко Никита Никитович	user	Транспорта и логистики	Сидоров Сидр Сидорович	Изменить	Удалить	

Рис. 8. Панель администратора. Список пользователей ИС

Для добавления новой записи в таблицу, что по своей сути является созданием нового пользователя необходимо заполнить форму, которая появляется при нажатии на кнопку добавить пользователя. Пример формы изображен на рисунке 9.



Система контроля поручений **Панель администратора**

[Пользователи](#)
[Подразделения](#)
[Поручения](#)
[Статусы поручений](#)
[Список должностей](#)

Пользователи

Создание пользователя

Учетная запись

Пароль

Роль

ФИО

Руководитель (если есть)

Подразделение

[Создать пользователя](#)

[Вернуться к списку пользователей](#)

Рис. 9. Панель администратора. Форма создания пользователя

Изменение пользователя откроет такую же форму, но уже с заполненными данными. Изменение полей позволит внести изменения и в свойства пользователя.

Красная кнопка удаления в списке пользователей удалит пользователя из системы и лишит его доступа навсегда. Для выполнения этого действия потребуется дополнительное подтверждение, чтобы убедиться в намерениях администратора и защитить список пользователей от случайного нажатия мыши. Подтверждение в виде модального окна можно увидеть на рисунке 10.

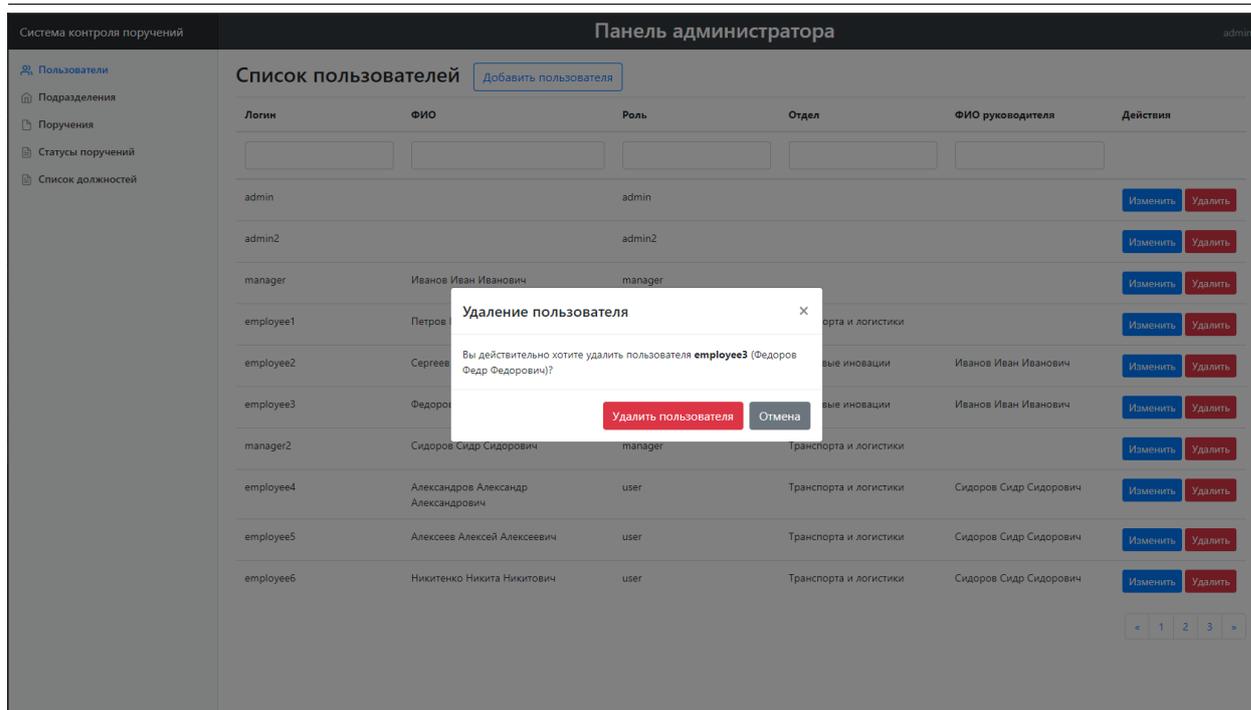


Рис. 10. Панель администратора. Подтверждение удаления пользователя

Для возобновления доступа необходимо заново завести учетную запись пользователя.

Итогом работы стала разработанная система контроля поручений, с возможностью применения для взаимодействия отдела информационных технологий какого-либо учреждения и подведомственных ему учреждений.

Библиографический список

1. Андон Ф., Резниченко В. Язык запросов SQL. Учебный курс. СПб.: Питер, 2006. 416 с.
2. Астахова И. Ф., Толстобров А. П., Чулюков В. А., Потапов А. С. Учебные пособия «СУБД: язык SQL в примерах и задачах», «Искусственный интеллект. Практический курс» // Современные проблемы науки и образования. 2009. № 1. С. 53.
3. Астахова И. Ф., Мельников В. М., Толстобров А. П. СУБД. Язык SQL в примерах и задачах. М.: ФИЗМАТЛИТ, 2009. 166 с.
4. Береснев А. Л. Администрирование GNU/Linux с нуля. СПб.: БХВ-Петербург, 2010. 576 с.
5. Валади Д. Linux. М.: Технолоджи-3000, 2006. 335 с.
6. Грекул В. И., Денищенко Г. Н., Коровкина Н. Л. Проектирование информационных систем: курс лекций. М.: Интернет-Университет информ. технологий, 2005. 304 с.
7. Гречка Я.Э. Создание и управление базой данных посредством SQL // Актуальные вопросы научных исследований: сборник научных трудов по материалам III Международной научно-практической конференции, г. Иваново, 15 июня 2016 г. Иваново, 2016. С. 5–7.

8. Гутманс Э., Баккен С., Ретанс Д. PHP 5. Профессиональное программирование. СПб.: Символ, 2006. 701 с.
9. Дакетт Д. Основы веб-программирования с использованием HTML, XHTML и CSS. М.: Эксмо, 2010. 767 с.
10. Иванов Н. Н. Программирование в Linux. Самоучитель. СПб.: БХВ-Петербург, 2012. 400 с.
11. Клименко С. В., Крохин И. В., Куц В. М. Электронные документы в корпоративных сетях: второе пришествие Гутенберга. М.: Анкей: ЭК ТРЕНДЗ, 1999. 271 с.
12. Клоков И. В., Пташинский В. С. Эффективное делопроизводство. СПб.: Питер, 2008. 218 с.
13. Колисниченко Д. Н. Самоучитель Microsoft Windows 10. СПб.: БХВ-Петербург, 2016. 352 с.
14. Колисниченко Д. Н. Серверное применение Linux. СПб.: БХВ-Петербург, 2011. 528 с.
15. Кузнецов С. Д. Основы современных баз данных. М: Центр Информационных Технологий, 1998. – 263 с.
16. Маклаков С. В. Моделирование бизнес-процессов с BPwin 4.0. М.: Диалог-МИФИ, 2002. 224 с.
17. Малыгина М. П. Базы данных: основы, проектирование, использование. СПб.: БХВ-Петербург, 2006. 517 с.
18. Мейер Э. CSS. Каскадные таблицы стилей: подробное руководство. СПб.; М.: Символ-Плюс, 2008. 573 с.
19. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript и CSS. СПб.: Питер. 2013. 560 с.
20. Персианов В. В., Киреева Е.З., Казакова М.Н. Электронное офисное делопроизводство. М.; Берлин: Директ-Медиа, 2016. 326 с.
21. Поляк Ю. Е. От безбумажной информатики – к электронному государству// Информационные ресурсы России. 2010. № 2 (114). С. 13–25.
22. Резников Ф. А. Ubuntu Server 2012–2015 + настольные ПК с Ubuntu в офисе. Администрирование с помощью Webmin, файловый сервер и сервер печати, 1С: Предприятие 7.7 в Ubuntu, свободные офисные программы, Интернет-шлюз, работа в режиме удаленного терминала. М.: Триумф, 2012. 255 с.
23. Селко Д. Стиль программирования Джо Селко на SQL. М.: Рус. ред.; СПб.: Питер, 2006. 192 с.
24. Файли К. SQL: Руководство по изучению языка: пер. с англ. М.: ДМК Пресс: Питер, 2004. 451 с.
25. Хабаров С. П., Жук Ю. А. Сетевые технологии взаимодействия Ubuntu и Windows платформ. – СПб.: Наука и техника, 2013. 369 с.
26. Шлоснейгл Д. Профессиональное программирование на PHP: практическое руководство по разработке крупномасштабных Web-сайтов и приложений с помощью PHP 5. М.: Вильямс, 2006. 610 с.