

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра _____ «Прикладная математика и информатика» _____
(наименование)

_____ 09.03.03 Прикладная информатика _____
(код и наименование направления подготовки / специальности)

_____ Разработка программного обеспечения _____
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка программного обеспечения системы учёта кадров»

Обучающийся

_____ А.В. Горлов _____

(Инициалы Фамилия)

_____ (личная подпись)

Руководитель

_____ д.т.н., доцент, С.В. Мкртычев _____

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема бакалаврской работы: «Разработка программного обеспечения системы учёта кадров».

Цель данной выпускной квалификационной работы (ВКР) заключается в разработке программного обеспечения системы учета кадров.

Главной задачей работы является создание функционально-ориентированного программного продукта, который будет эффективно и надежно управлять информацией о сотрудниках организации, их трудовой деятельности и другими аспектами кадрового учета.

Введение формулирует цель работы и задачи ее реализации.

В первой главе рассматриваются функциональные и архитектурные особенности систем учета кадров предприятий социально-экономической сферы, разрабатываются требования к программному обеспечению системы учета кадров предприятия и проводится анализ аналогов систем по учёту кадров с обоснованием о необходимости разработки собственного программного обеспечения.

Во второй главе описывается структура компонентов системы. Представлены логические модели системы и базы данных.

Третья глава описывает процесс реализации и тестирования разработанного программного обеспечения.

В заключении, основные выводы, сделанные в процессе проведенного исследования, и описаны результаты практической реализации окончательной квалификационной работы.

Бакалаврская работа состоит из 49 страниц текста, 22 рисунков, 3 таблиц, 20 источников и 1 приложения.

Оглавление

Введение	4
Глава 1 Постановка задачи на разработку программного обеспечения информационной системы учета кадров.....	6
1.1 Функциональные и архитектурные особенности систем учета кадров предприятий социально-экономической сферы	6
1.2 Разработка требований к ПО системы учета кадров	10
1.3 Анализ аналогов ПО информационной системы учета кадров предприятия.....	14
Глава 2 Проектирование программного обеспечения информационной системы учета кадров.....	18
2.1 Объектно-ориентированный подход к проектированию программного обеспечения системы учёта кадров	18
2.2 Разработка логической модели данных системы учёта кадров	28
Глава 3 Реализация и тестирование программного обеспечения информационной системы учета кадров.....	31
3.1 Выбор инструментов и технологии	31
3.2 Контрольный пример реализации проекта системы учёта кадров ..	33
Заключение	42
Список используемой литературы	44
Приложение А. Фрагмент кода подключения Python к базе данных MS Access	48

Введение

В современных условиях управления персоналом важным инструментом является эффективная система учета кадров. Она позволяет автоматизировать процессы управления персоналом, облегчая контроль за работниками, их рабочим временем, оценкой результативности и многими другими аспектами управления. Разработка программного обеспечения для учета системы кадров является значимым направлением в сфере информационных технологий и управления персоналом.

В этой связи представляет научно-практический интерес разработка программного обеспечения системы учёта кадров в организации по учёту личного состава внутри этой же организации.

Объектом исследования выпускной квалификационной работы является процесс учёта кадров в подчинённом подразделении организации.

Предметом исследования является информационная система учёта кадров подчинённого подразделения организации.

Цель выпускной квалификационной работы (ВКР) заключается в разработке программного обеспечения системы учета кадров для небольшого подразделения. Главной задачей работы является создание функционально-ориентированного программного продукта, который будет эффективно и надежно управлять информацией о сотрудниках организации, их трудовой деятельности и другими аспектами кадрового учета.

Для достижения поставленной цели необходимо провести анализ существующих методов и средств учета кадров, выявить основные требования к программному обеспечению, разработать структуру базы данных, функциональную модель и интерфейс пользователя. Важным этапом работы будет тестирование и апробация разработанного программного продукта на реальных данных организации, дополнительно не рассмотренных авторами программы.

Исследование в области разработки программного обеспечения для учета системы кадров имеет большое практическое значение, поскольку позволяет повысить эффективность управления персоналом, сократить временные и финансовые затраты на учет кадровых данных и повысить общую производительность организации.

Результаты данной работы могут быть использованы в организациях любого уровня и направлены на оптимизацию процессов управления персоналом, повышение эффективности работы кадровых служб и обеспечение более качественного управления трудовыми ресурсами.

Данная работа состоит из введения, трёх глав, заключения и списка используемой литературы и источников.

Первая глава посвящена задачам на разработку программного обеспечения информационной системы учёта кадров.

Вторая глава описывает структуру компонентов системы. Дается реализация проекта выбранными средствами.

Третья глава описывает тестирование исходя из тестовых данных, используемых для визуализации.

В заключении, основные выводы, сделанные в процессе проведенного исследования, и описаны результаты практической реализации окончательной квалификационной работы.

Бакалаврская работа состоит из 49 страниц текста, 22 рисунков, 3 таблиц, 20 источников и 1 приложения.

Глава 1 Постановка задачи на разработку программного обеспечения информационной системы учета кадров

1.1 Функциональные и архитектурные особенности систем учета кадров предприятий социально-экономической сферы

Системы учета кадров являются важным инструментом для управления персоналом в предприятиях социально-экономической сферы. Автоматизация процессов управления персоналом сокращает финансовые и трудовые затраты, так как данные, введенные в систему, единожды используются всеми заинтересованными сотрудниками, и самое важное – управление поднимается на новый качественный уровень [1]. Рассмотрим основные функциональные и архитектурные особенности таких систем.

Рассмотрим функциональные особенности систем учета кадров.

Обработка персональных данных – любая операция или набор операций, выполняемых над персональными данными как автоматическими средствами, так и без таковых, такие как: сбор, запись, организация, хранение, актуализация или изменение, извлечение, консультирование, использование, раскрытие посредством передачи, распространения или предоставления иного доступа, группировка, блокирование, стирание или разрушение [2].

Системы учета кадров позволяют хранить и обрабатывать персональные данные сотрудников, такие как:

- имя, фамилия, контактные данные;
- должность, отдел, руководитель;
- дата найма, зарплата, бонусы;
- данные о трудовом стаже, навыках и образовании.

Управление трудовыми договорами и документами.

Системы предоставляют возможности для создания, хранения и управления трудовыми договорами, а также различными документами, связанными с трудовой деятельностью [4]:

- трудовые контракты и соглашения;
- заявления о приеме на работу, увольнении, переводе;
- графики работы, отпуска, больничные.
- управление процессами найма и адаптации.

Автоматизация предприятия на основе процессного подхода также позволяет оперативно перестраивать бизнес-процессы организации и административные регламенты ведомств [9].

Системы учета кадров поддерживают процессы найма и адаптации новых сотрудников:

- обработка заявок на трудоустройство;
- управление процессом собеседования и принятия решений о найме;
- подготовка документов для новых сотрудников;
- организация процесса адаптации (обучение, знакомство с коллективом и т.д.).

Входной информацией служит сотрудник, устраивающийся на работу или переводящийся на иную должность, согласно его квалификационным данным (повышение, либо понижение в должности), при этом имеет место быть и увольнение с проставлением соответствующих отметок в трудовой книжке и иных документах установленным порядком.

Управление производительностью и развитием персонала.

Персонал находится в постоянном движении вследствие приема на работу одних и увольнения других [6].

Функциональность систем учета кадров позволяет оценивать производительность сотрудников и управлять их развитием:

- оценка эффективности работы сотрудников;
- управление карьерным ростом и продвижением по службе;

– планирование и учёт обучения, тренингов, повышения квалификации.

Отчетность и аналитика.

Анализ способствует принятию верных решений и некоторого набора возможных [11]. Системы учёта кадров включают функции отчетности и аналитики для поддержки принятия управленческих решений:

- создание отчетов о численности, текучести кадров, эффективности;
- анализ данных для прогнозирования потребностей в персонале;
- интеграция с другими системами (финансовыми, бухгалтерскими) для получения сводных отчетов.

Архитектурные особенности систем учета кадров.

Клиент-серверная архитектура.

Системы учёта кадров обычно имеют клиент-серверную архитектуру, где серверная часть отвечает за обработку данных и хранение информации, а клиентская часть предоставляет интерфейс для пользователей.

Серверы могут осуществлять хранение данных, управление базами данных, удаленную обработку заданий, печать заданий и ряд других функций, потребность в которых может возникнуть у пользователей сети [5].

Целью разработки любой базы данных является хранение и использование информации о какой-либо предметной области [7].

Системы учета кадров используют базы данных для хранения информации о персонале. Обычно это реляционные базы данных, которые обеспечивают структурированное хранение данных и позволяют выполнять разные запросы и отчеты.

Интеграция с другими системами.

Учитывая имеющую место тенденцию к укрупнению (порою перманентного) организаций, а также потребность объединённой организации в быстром получении доступа к интегрированной совокупности базы данных всех организаций, вошедших в её состав, подход к быстрому объединению

этих баз данных путём применения системы интеграции данных представляется весьма актуальным [3].

Для эффективного управления персоналом системы учета кадров интегрируются с другими системами, такими как:

- системы управления финансами и бухгалтерией (для расчета зарплаты и налогов);
- системы управления обучением и развитием персонала;
- системы контроля доступа и безопасности.

Безопасность и контроль доступа.

Безопасность обработки заключается в том, что контролер обязан реализовать надлежащие технические и организационные меры для защиты персональных данных от случайного или незаконного уничтожения или случайной утраты, изменения, неправомерного раскрытия или доступа, в частности, когда обработка влечет передачу данных по сети, а также от всех иных незаконных форм обработки [2].

Поскольку системы учета кадров содержат чувствительные персональные данные, они должны обеспечивать высокий уровень безопасности:

- ограничение доступа к данным на основе ролей и привилегий;
- шифрование данных и безопасное соединение;
- журналы аудита для отслеживания доступа и изменений в данных.

Кроме того, доступ к рабочему месту, то есть вход в учётную запись операционной системы, для последующего использования в своей работе, строго ограничен, поэтому рассматривается вариант для пользователей (сотрудников) кадрового обеспечения в небольшой, но специфической организации, не прибегая к дополнительным ограничениям доступа в разрабатываемом программном обеспечении, хоть это и не совсем правильно, ввиду человеческого фактора, однако в будущем рассматривается

модернизация ПО и обновления, включая внедрение дополнительной безопасности.

Гибкость и расширяемость.

При выполнении метрологии программного обеспечения рекомендуется измерить характеристики качества программного обеспечения и определить их соответствие установленным требованиям [12].

Системы учета кадров должны быть гибкими и легко расширяемыми для удовлетворения растущих потребностей организации:

- возможность добавления новых функций и модулей;
- поддержка изменений в законодательстве и нормативных актах;
- интеграция с новыми технологиями и платформами.

Таким образом, системы учета кадров для предприятий социально-экономической сферы предоставляют широкий спектр функциональных возможностей, которые могут быть настроены в соответствии с потребностями организации. Архитектурные особенности обеспечивают надежность, безопасность и гибкость, что позволяет эффективно управлять кадровыми процессами и поддерживать рост организации.

1.2 Разработка требований к ПО системы учета кадров

В процессе выделения требований должен быть создан список уникальных требований к продукту, которые не должны повторяться, но могут дополнять друг друга [10]. Требования делятся на функциональные и нефункциональные. Они описывают поведение системы, ожидаемые результаты, а также стандарты, которые система должна соблюдать.

Требования к программному обеспечению, имеющему прототипы, обычно определяют по аналогии, учитывая структуру и характеристики уже существующего программного обеспечения [10].

Функциональные требования определяют работу, которую должно выполнять программное приложение [10].

Работа с персональными данными.

Система поддерживает ввод, хранение и обновление персональных данных сотрудников, включая:

- личные данные (имя, фамилия, контактная информация);
- служебные данные (должность, отдел, руководитель);
- историю трудоустройства, даты найма, повышения и увольнения;
- финансовую информацию (зарплата, бонусы, премии).

Управление документами.

ПО умеет создавать, хранить и управлять документами, связанными с кадровыми процессами:

- трудовые контракты, заявки о приеме на работу, заявления об увольнении, документы о переводах;
- документы, связанные с отпуском и больничными.

Управление процессами найма и адаптации.

Система поддерживает процессы найма и адаптации сотрудников:

- поддержка различных этапов найма (подача заявки, собеседования, принятие на работу);
- отслеживание прогресса адаптации, обучение, назначение наставников

Управление производительностью и развитием персонала.

Система имеет функциональность для управления производительностью и карьерным ростом:

- оценка эффективности сотрудников (например, на основе KPI);
- управление карьерным ростом, планирование обучения и развития;
- организация процесса аттестации, оценочные собеседования.

Отчетность и аналитика.

ПО поддерживает создание отчетов и анализ данных:

- стандартные отчеты по кадровым процессам (текучесть кадров, статистика по найму);
- возможность генерации пользовательских отчетов;
- интеграция с другими системами для кросс-отчетности (например, с бухгалтерией или финансовым отделом).

Нефункциональные требования. Требования к производительности определяют временные ограничения, которые должны быть выполнены в программе [10].

Безопасность и контроль доступа.

ПО обеспечивает безопасность данных и контроль доступа:

- авторизация и аутентификация пользователей;
- разграничение прав доступа на основе ролей и привилегий;
- шифрование данных и обеспечение безопасности при передаче данных.

Производительность и масштабируемость:

- быстрый отклик на запросы пользователя;
- поддержка одновременной работы нескольких пользователей;
- возможность расширения в будущем без существенных изменений в архитектуре.

Надежность и резервное копирование:

- защита от сбоев и потери данных;
- автоматическое резервное копирование данных;
- возможность восстановления данных в случае сбоев.

Интеграция и совместимость.

ПО совместимо с другими системами и легко интегрируется:

- поддержка стандартных форматов данных для интеграции (например, CSV, XML, JSON);
- возможность подключения к другим системам (например, финансовым, обучающим);

- совместимость с различными операционными системами и аппаратными платформами.

Удобство использования и эргономика [10].

ПО должно быть удобным и интуитивно понятным для пользователей:

- интуитивный и понятный интерфейс;
- минимальное время обучения пользователей;
- поддержка многоязычности и адаптивного дизайна.

Рассмотрим основные требования по модели FURPS+ [18], которая является одной из часто используемых моделей для классификации атрибутов качества программного обеспечения.

Основные требования по модели FURPS+ представлены в таблице 1.

Таблица 1 – Требования по модели FURPS+

Требование	Статус	Полезность	Риск	Стабильность
Функциональность (Functionality)				
Отслеживание данных сотрудников	допустимый	критическая	низкий	высокая
Формирование документов	допустимый	критическая	низкий	высокая
Управление доступом	допустимый	критическая	средний	низкая
Интеграция с базой данных	допустимый	критическая	средний	высокая
Удобство использования (Usability)				
Интуитивный интерфейс	допустимый	критическая	низкий	высокая
Пользовательская документация	допустимый	важная	низкий	высокая
Надежность (Reliability)				
Резервное копирование данных	допустимый	важная	средний	высокая
Обработка ошибок	допустимый	важная	низкий	высокая
Производительность (Performance)				
Быстродействие	допустимый	критическая	средний	высокая
Масштабируемость	допустимый	важная	средний	высокая
Поддержка (Supportability)				
Устранение критических проблем	допустимый	важная	средний	средняя
Проектные ограничения				
Соответствие федеральной нормативно-правовой базе	допустимый	критическая	средний	низкая
Низкая совокупная стоимость владения	допустимый	критическая	средний	низкая

Разработка требований к ПО системы учета кадров предприятия – это ключевой этап, который закладывает основу для успешного проектирования, реализации и внедрения системы.

1.3 Анализ аналогов ПО информационной системы учета кадров предприятия

Аналоги систем учёта кадров представлены в таблице 2.

Таблица 2 – Аналоги коммерческого ПО систем учета кадров

Параметр	1С: «Зарплата и управление персоналом»	Босс-Кадровик	Кадровик
Тип продукта	Программа, система и облачный	Система и облачный	Система и облачный
Целевая аудитория	Малые, средние и крупные предприятия	Средние и крупные предприятия	Малые и средние предприятия
Модель данных	Реляционная	Реляционная	Реляционная
Интеграция с другими продуктами	Интеграция с продуктами 1С и внешними системами	Интеграция с ERP-системами и внешними базами данных	API для интеграции с другими корпоративными системами

Данные продукты предлагают широкий спектр функций, включая:

- управление персональными данными сотрудников;
- создание и управление документами, связанными с трудоустройством;
- планирование и управление отпусками, больничными и рабочим графиком;
- отчетность и аналитика.

Несмотря на богатую функциональность, эти решения часто слишком сложны для небольших предприятий или узкоспециализированных организаций.

Они требуют больших затрат на внедрение, обучение и поддержку, а также могут не соответствовать специфическим требованиям безопасности.

Бесплатные и открытые решения представлены в таблице 3.

Таблица 3 – Аналоги бесплатного ПО систем учёта кадров

Параметр	OrangeHRM [20]	Sentrifugo HRMS [19]
Тип продукта	Система управления человеческими ресурсами	Система управления человеческими ресурсами
Целевая аудитория	Средние и крупные компании	Малые и средние компании
Модель данных	Реляционная	Реляционная
Интеграция с другими продуктами	Сервисы, платформы для обмена сообщениями, системы управления контентом и платежные шлюзы, плагины для интеграции с электронной почтой и календарями.	Также как у OrangeHRM плюс к этому API для интеграции

Эти решения обеспечивают базовую функциональность по управлению кадрами, включая:

- управление данными сотрудников и документами;
- управление отпусками и больничными;
- базовую отчетность.

Однако они могут иметь ограничения по функциональности, гибкости и возможностям кастомизации.

Кроме того, открытые решения часто требуют больше технических знаний для их адаптации к специфическим потребностям организации.

Исходя из особенностей предприятия, необходимость разработки собственного ПО для учета кадров можно обосновать следующим, а именно:

- узкая направленность. Организация имеет специфические требования к системе учета кадров, которые не всегда могут быть реализованы в коммерческих или открытых решениях;
- разработка собственного ПО позволяет сконцентрироваться на наиболее важных функциях, избегая излишней сложности;
- безопасность и конфиденциальность. В связи с особенностями работы предприятия, требования к безопасности данных и контролю доступа чрезвычайно высоки. Использование собственной системы на базе Microsoft Access и Python позволяет обеспечить строгий контроль над доступом к данным и избежать потенциальных уязвимостей, характерных для коммерческих решений, учитывая при этом, что работа с готовым продуктом рассматривается с учётом того, что выход в глобальную информационную сеть будет отсутствовать;
- интеграция с существующей инфраструктурой. Microsoft Access и Python – это хорошо знакомые инструменты, которые легко интегрируются с существующей инфраструктурой организации. Это облегчает взаимодействие с другими системами и сокращает затраты на внедрение и поддержку;
- гибкость и адаптивность. Разработка собственного ПО позволяет создавать гибкую и адаптивную систему, которая может легко модифицироваться и расширяться по мере изменения потребностей организации. Это особенно важно в условиях, когда бизнес-процессы могут быстро меняться;
- экономическая эффективность. Создание собственного ПО может быть экономически эффективным по сравнению с использованием коммерческих продуктов, которые часто требуют высоких затрат на лицензии, поддержку и обучение. Использование собственных ресурсов и знаний Python позволяет снизить затраты на разработку и обслуживание системы.

Анализ аналогов ПО для систем учета кадров показывает, что, хотя существуют коммерческие и открытые решения, разработка собственного ПО может быть более подходящей для специфического и небольшого предприятия.

Собственное ПО обеспечивает гибкость, безопасность и возможность адаптации к специфическим потребностям организации, интегрироваться с существующими системами и базами данных, при этом позволит сэкономить на лицензионных и обслуживающих платежах, связанных с коммерческими решениями.

Выводы по первой главе

Проектирование и создание системы учёта кадров в контексте небольшой организации, тщательно проанализирован и раскрыты её функциональные и нефункциональные требования.

При этом, рассмотрены имеющиеся на рынке готовые решения, со значительно раскрытым функционалом по стандартам кадрового документооборота.

Определена необходимость разработки собственного программного обеспечения, исходя из потребностей и особенностей организации, для которой и будет предназначаться создаваемый продукт.

Глава 2 Проектирование программного обеспечения информационной системы учета кадров

2.1 Объектно-ориентированный подход к проектированию программного обеспечения системы учёта кадров

Для проектирования и описания всех процессов, протекающих в ходе разработки, используем унифицированный язык моделирования UML. Для создания графических нотаций выбран инструмент моделирования StarUML [16].

Рассмотрим варианты использования и взаимодействие актёров и прецедентов.

Диаграмма вариантов использования представлена на рисунке 1.

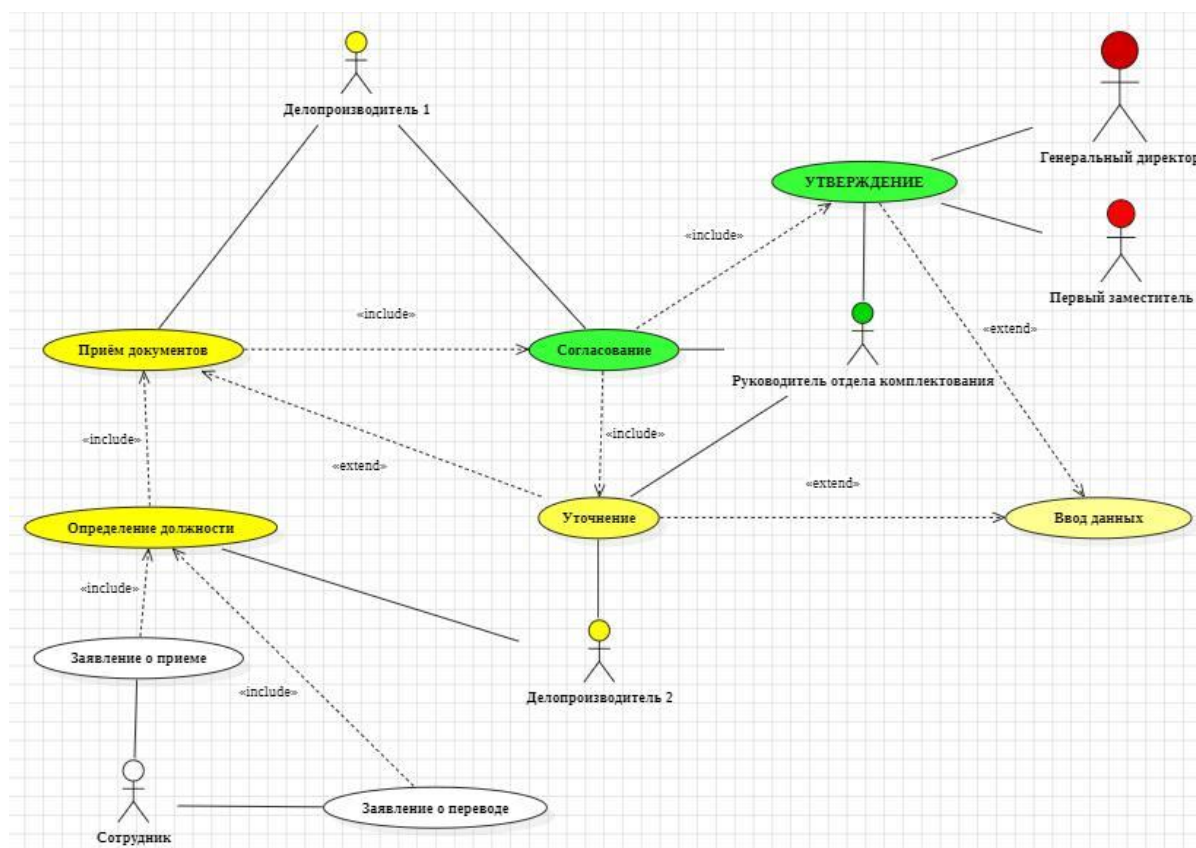


Рисунок 1 – Диаграмма вариантов использования системы учета кадров

Согласно диаграмме вариантов использования:

- сотрудник пишет заявление о приеме на работу или переводе на иную должность;
- делопроизводитель в соответствии с действующим штатом ОСК определяет подходящую по запросу должность;
- делопроизводитель принимает все необходимые документы для трудоустройства и направляет их на согласование руководителю отдела комплектования;
- при успешном согласовании делопроизводитель вносит соответствующие данные (факт запроса о приеме на работу или переводе на другую должность от сотрудника) в систему;
- руководитель отдела комплектования подготавливает комплект документов для утверждения у вышестоящего руководства;
- после утверждения руководитель отдела комплектования направляет комплект документов в делопроизводство для последующего внесения изменений в статус сотрудника;
- делопроизводитель уточняет информацию о штатной численности и иных данных требующих внесения изменений в электронную базу данных и проводит соответствующие мероприятия.

В отношении обобщений между актерами следующее:

- будущий или действующий сотрудник обращается с запросом к делопроизводителю;
- делопроизводитель проверяет и согласовывает данные сотрудника с руководителем отдела комплектования и после согласования вносит соответствующие данные о штатном перемещении сотрудников;
- руководитель отдела комплектования направляет в вышестоящее руководство для утверждения.

Рассмотрим рисунок 2 – диаграмму деятельности для потока событий прецедента «Определение должности».

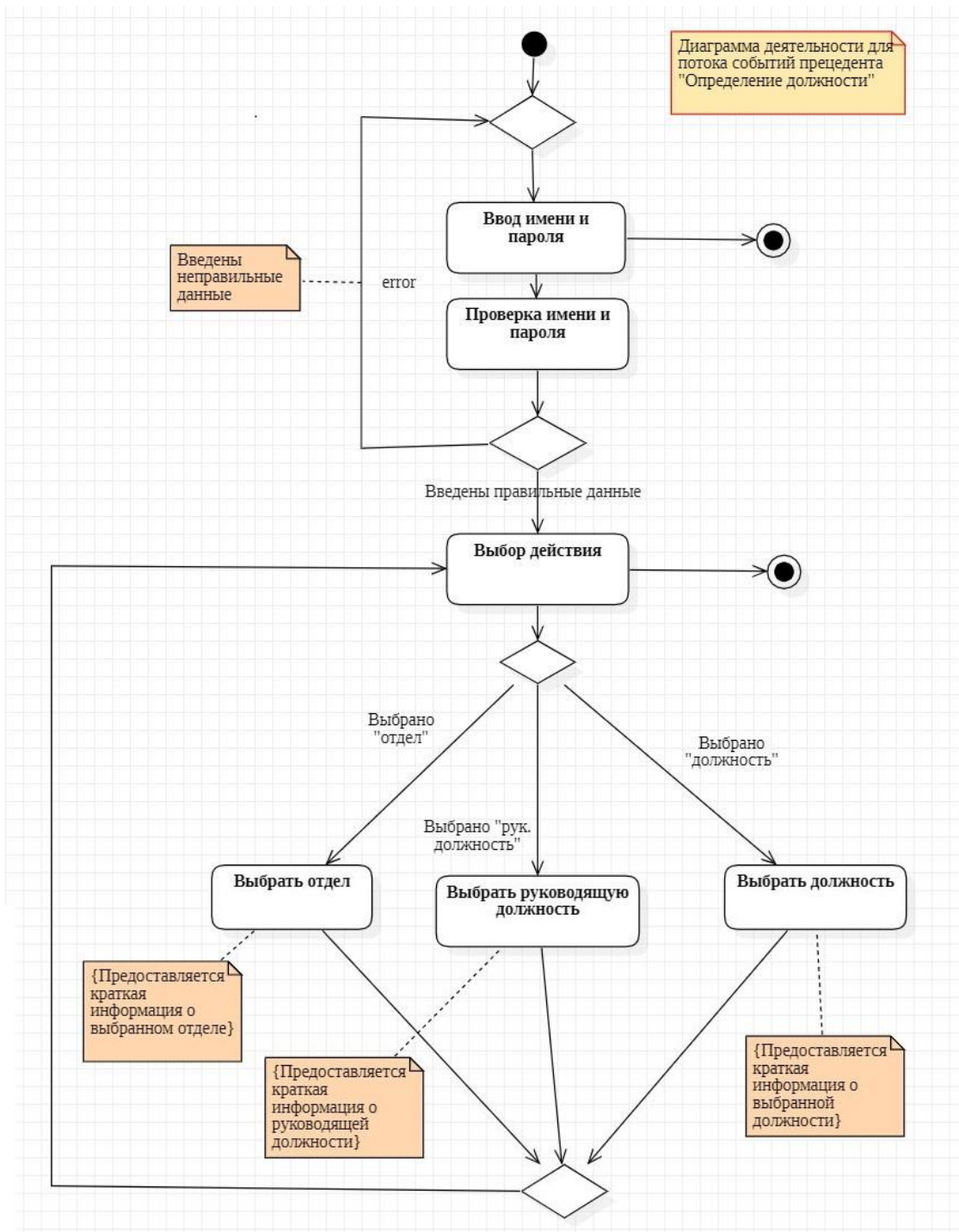


Рисунок 2 – Диаграмма деятельности для потока событий прецедента «Определение должности»

Данная диаграмма представлена для делопроизводителя, так как предоставляемая большая часть информации является служебной:

- делопроизводитель вводит свои данные для доступа в систему, в случае успешного ввода, доступ предоставляется, в ином случае, данные рекомендуется ввести заново;
- после успешного входа в систему, предоставляется выбор по отделам, руководящему составу и личному составу сотрудников, но информация предоставляется, только по вакантным должностям с кратким описанием.

Для ознакомления, будущим сотрудникам, рекомендуется использовать электронные терминалы, а исходя из диаграммы на рисунке 2, для ознакомления открытой информации из терминала, можно сократить схему, начиная с «выбора действия».

Для этого рассмотрим диаграмму на рисунке 3.

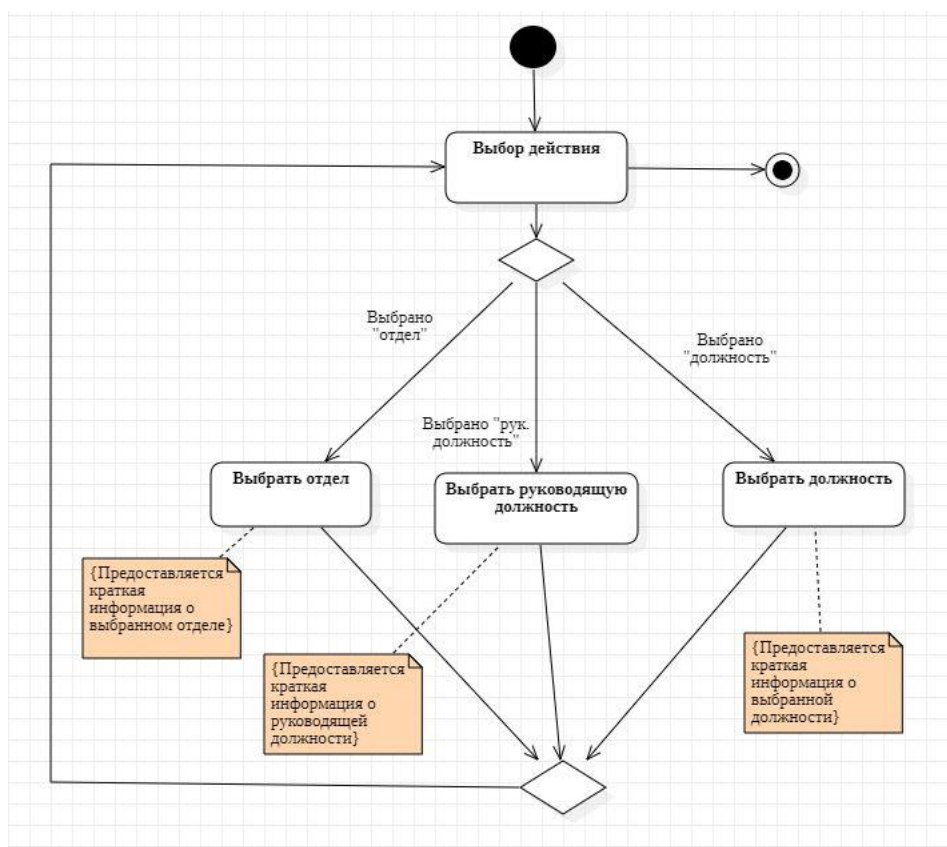


Рисунок 3 – Диаграмма деятельности для потока событий прецедента «Ознакомление»

У действующих сотрудников, те, кто работает внутри системы, есть индивидуальных доступ к ней из-за ограничений служебной деятельности, соответственно каждый раз вход в систему начинается с ввода логина и пароля пользователя или администратора.

Рассмотрим диаграмму деятельности для потока событий прецедента «Уточнение» на рисунке 4.

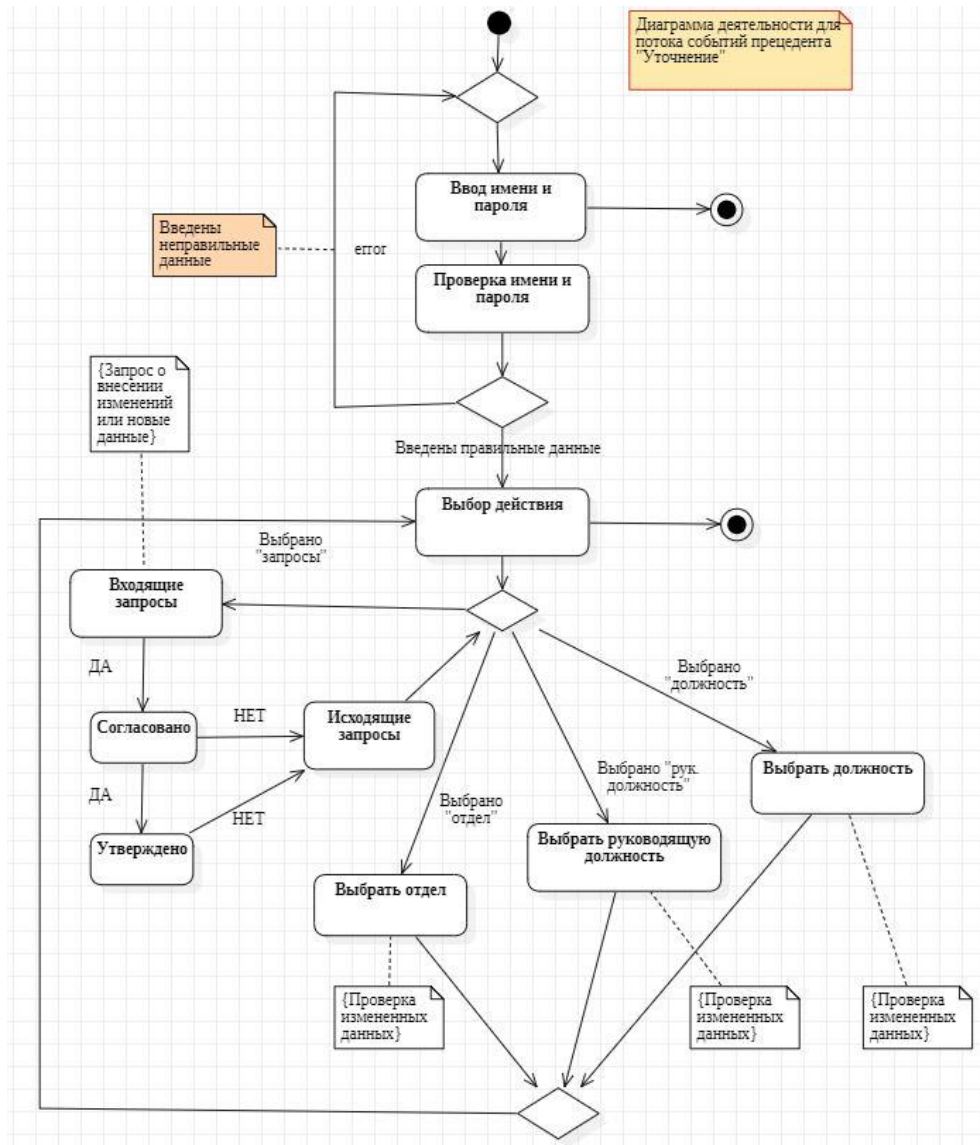


Рисунок 4 – Диаграмма деятельности для потока событий прецедента «Уточнение»

Согласно диаграмме деятельности, для потока событий прецедента «Уточнение»:

- делопроизводитель проверяет входящие данные, поступившие от руководителей для последующей корректировки в базу данных;
- из-за большого потока данных, если запрос поступил без начального согласования, то делопроизводитель направляет поступивший запрос в исходящие запросы, где в последующем, будет корректироваться данные и снова направляться во входящие запросы;
- также и с утверждением от руководства, пока отметки «Утверждено» не поступит, данные запроса направляются в исходящие запросы;
- проверяется и выбранная должность, она должна быть вакантной, так как нельзя исключать человеческого фактора, из-за которого данные не вносились своевременно.

Каждое состояние может являться выполнением операции некоторого класса либо ее части, позволяя использовать диаграммы деятельности для описания реакций на внутренние события системы. На диаграмме деятельности отображается логика или последовательность перехода от одной деятельности к другой, при этом внимание фиксируется на результате деятельности. Сам же результат может привести к изменению состояния системы или возвращению некоторого значения. Состояние действия является специальным случаем состояния с некоторым входным действием и по крайней мере одним выходящим из состояния переходом. Этот переход неявно предполагает, что входное действие уже завершилось [13].

Далее рассмотрим диаграмму деятельности для потока событий прецедента «Ввод данных» на рисунке 5.

утвержденного запросов, появляется возможность входа в штатную книгу;

- как и в предыдущих шагах диаграмм, есть возможность через штатную книгу выбрать отдела, затем должность, которую также возможно изменить;
- делопроизводитель вносит в соответствии с запросом все необходимые корректировки в предстоящую должность будущего, либо существующего сотрудника.

Далее, на рисунке 6, представлена диаграмма классов системы в терминологии классов объектно-ориентированного программирования.

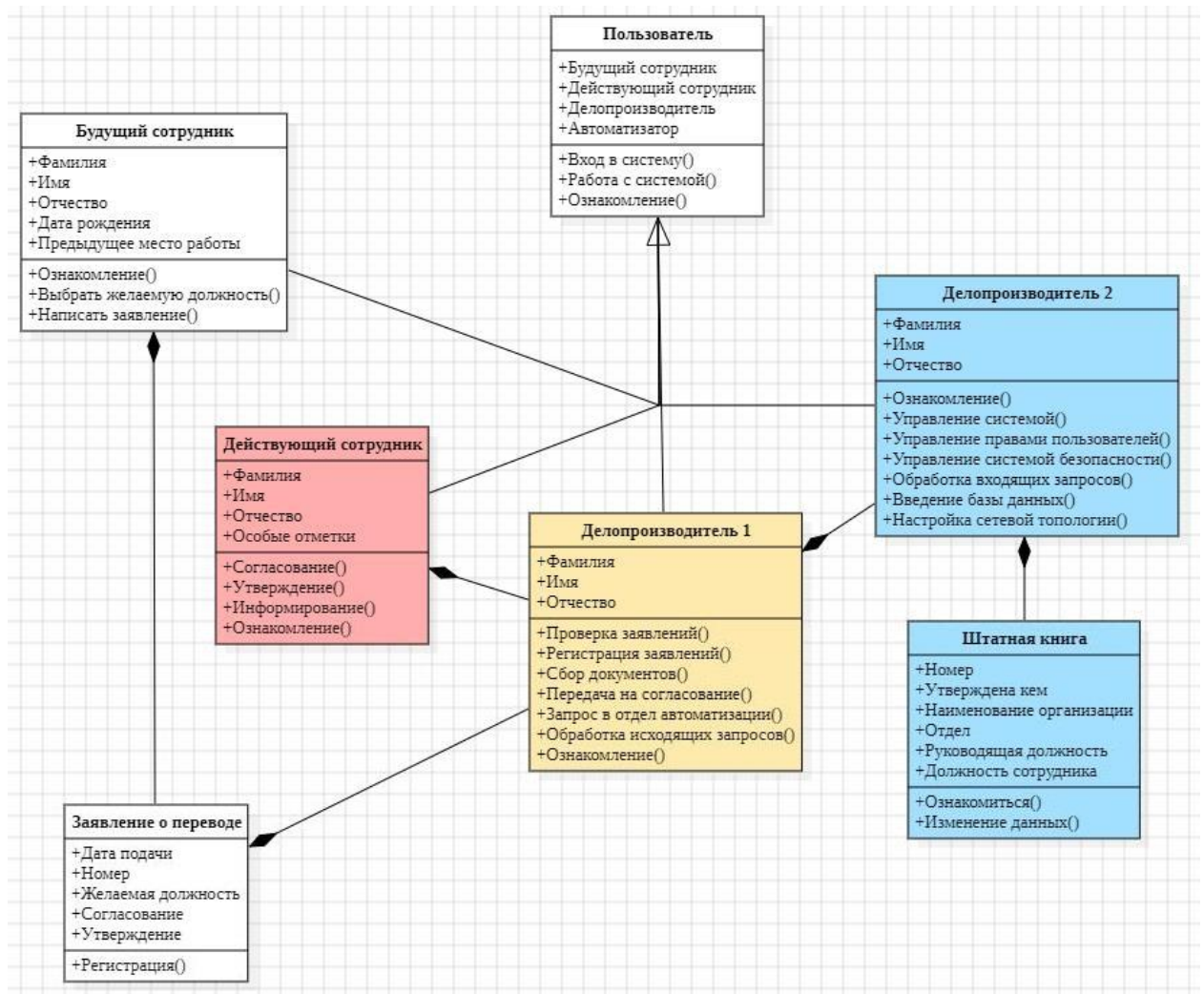


Рисунок 6 – Диаграмма классов системы учета кадров

Для реализации набора интерфейсов разработана диаграмма компонентов, показанная на рисунке 7.

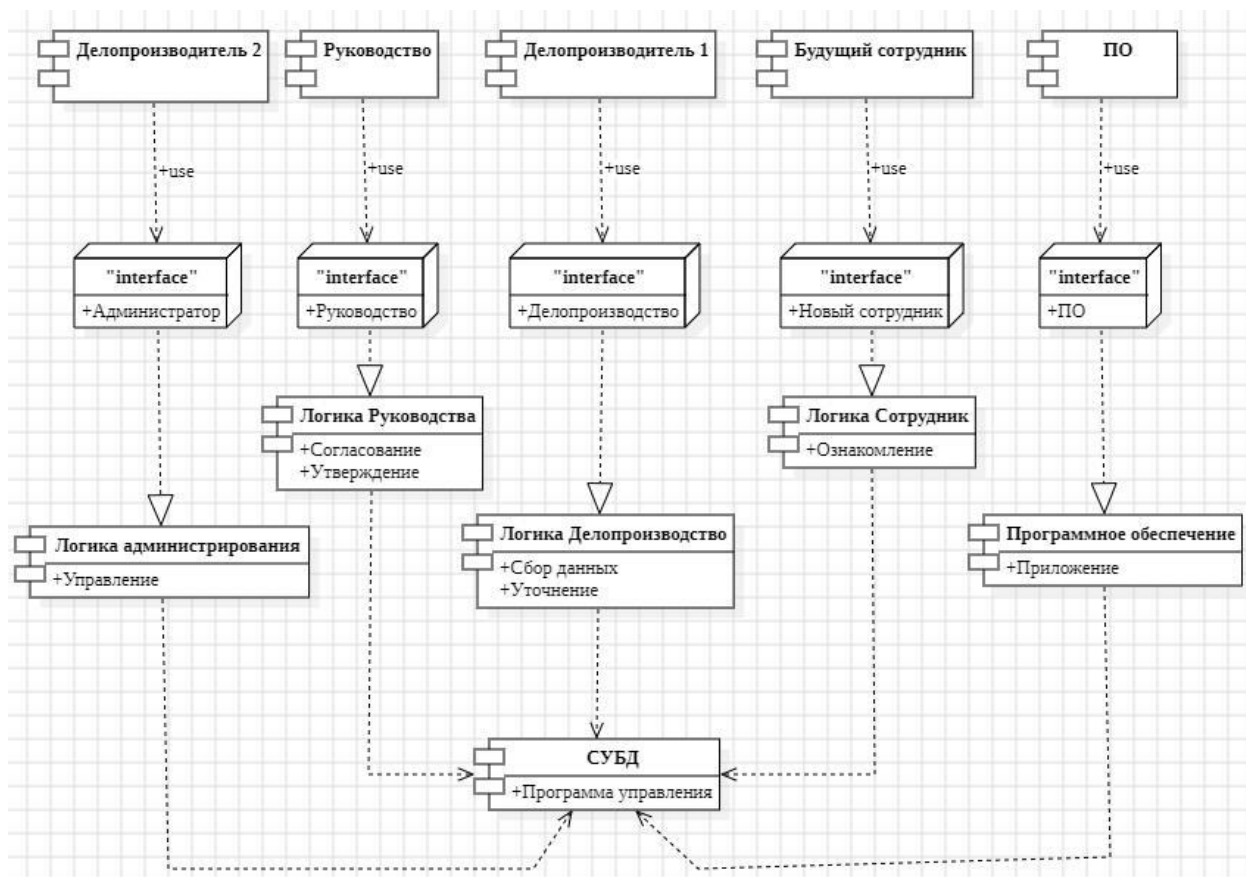


Рисунок 7 – Диаграмма компонентов системы учета кадров

На диаграмме компонентов описаны особенности физического представления системы, что позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код. Пунктирные стрелки, соединяющие модули, показывают отношения взаимозависимости, аналогичные тем, которые имеют место при компиляции.

Цели диаграммы компонентов следующие:

- визуализация общей структуры исходного кода программной системы;
 - спецификация программной системы;
 - обеспечение многократного использования отдельных фрагментов;
 - представление концептуальной и физической системы управления.
- Рассмотрим диаграмму развертывания на рисунке 8.

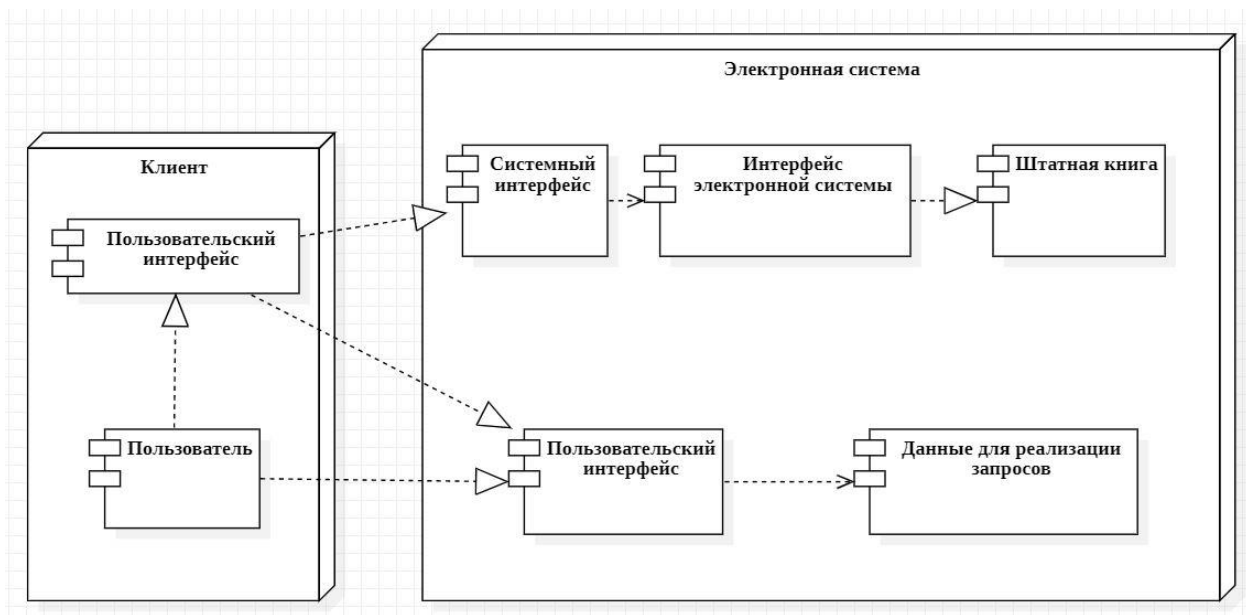


Рисунок 8 – Диаграмма развертывания системы учета кадров

Концепция информационной сети в диаграмме развертывания, основная часть ресурсов которой сосредоточена в электронной системе (сервер), предоставляет возможность пользователю, либо администратору, работать со штатной книгой.

В диаграмме развертывания указано физическое расположение компонентов:

- пользователь – компонент для программного представления пользователя и обеспечивает доступ к данным штатной книги;
- пользовательский и системный интерфейсы – графически визуальное представление системы для пользователя;

- интерфейс электронной системы – графически визуальное представление системы;
- данные для реализации запросов – компонент с основными элементами данных работы со штатной книгой;
- штатная книга – главный системный компонент в электронном виде для кадрового обеспечения.

2.2 Разработка логической модели данных системы учёта кадров

Системная архитектура проекта описывает основные компоненты системы, их взаимодействия и структуру данных. Данная архитектура основывается на следующих элементах:

- структура данных;
- компоненты приложения;
- взаимодействие с базой данных;
- пользовательский интерфейс.

Основой системы является база данных Microsoft Access. База данных включает таблицы, в которых хранятся информация о сотрудниках, отделах, позициях и зарплатах. Основные таблицы:

- Employees (Сотрудники): хранит информацию о сотрудниках, включая идентификатор, имя, фамилию, ID отдела, должность, зарплату;
- Departments (Отделы): содержит данные об отделах, такие как идентификатор отдела и его название;
- Job_Applications (заявка на работу): хранит информацию о сотрудниках, находящиеся на рассмотрении, в зависимости от заявленных ими требований по работе.

Связи в базе данных необходимы для избавления от многократного ввода одних и тех же данных, обеспечения целостности данных, упрощения создания много табличных форм, запросов и отчетов [14].

Рассмотрим эти таблицы на рисунке 9.

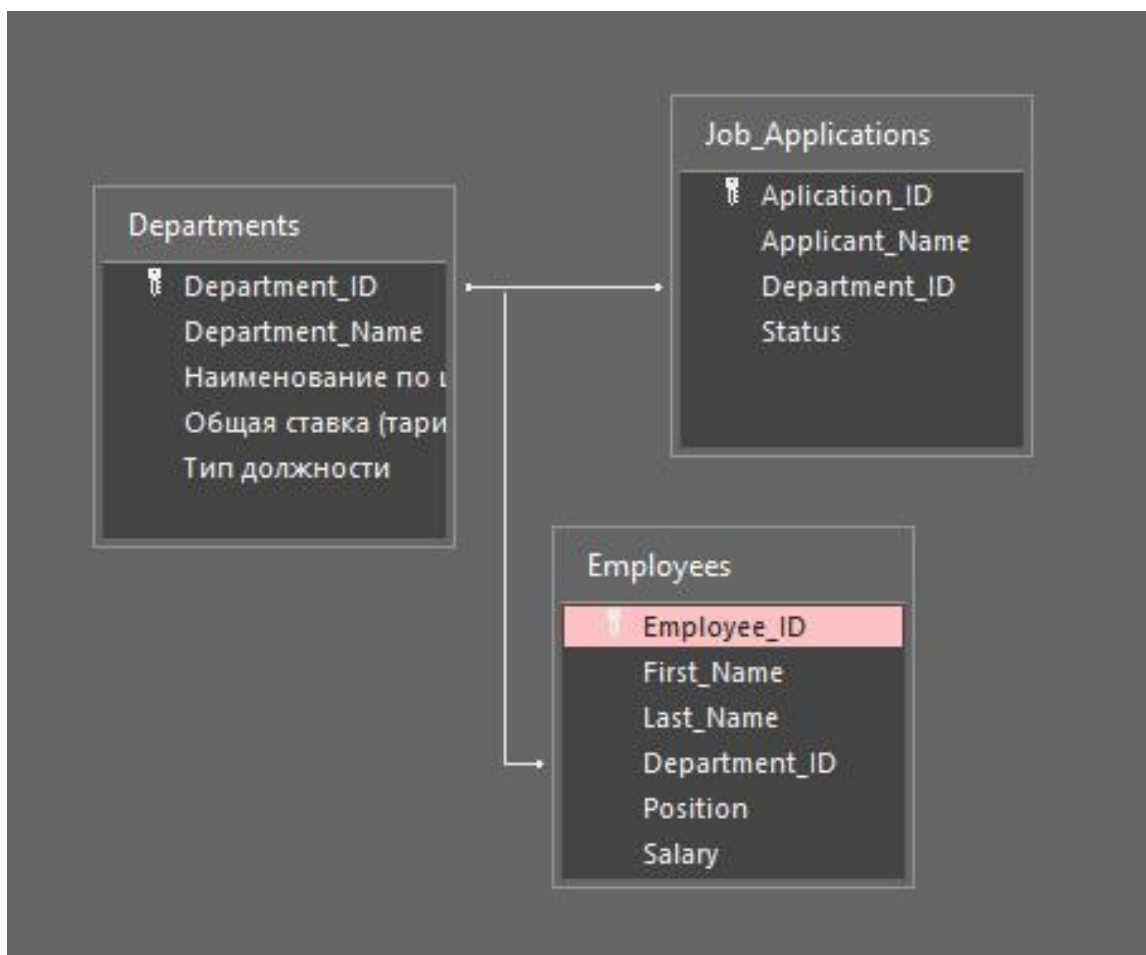


Рисунок 9 – Логическая модель данных системы учета кадров

Эти таблицы связаны через ключи Department_ID в таблице Employees являющиеся данными внешними ключами, указывающие на Department_ID в таблице Departments.

Фрагмент кода организации связи Python и базой данных Microsoft Access представлен в приложении А.

Полный код представлен на электронном ресурсе GitHub [15].

Выводы по главе 2

Логическая модель системы учёта кадров состоит из базовых диаграмм UML, которые являются более простыми для понимания [8].

Основой базы данных выбрана реляционная модель, основанная на отношениях между таблицами. Это решение позволяет эффективно организовать, хранить и управлять данными, обеспечивая целостность и согласованность данных и предлагает следующие преимущества:

- эффективное управление данными – инструменты для создания, чтения, обновления и удаления данных. Это позволяет систематизировать данные, минимизировать избыточность и обеспечить целостность данных с помощью механизмов ссылочной целостности;
- поддержка сложных запросов для анализа данных. Это делает возможным быстрое и эффективное извлечение нужной информации, что особенно важно для систем учета кадров, где требуется частый доступ к данным;
- обеспечение согласованности данных при одновременном выполнении нескольких операций. Это критически важно для систем, где данные постоянно обновляются и изменяются;
- масштабируемость и адаптивность в зависимости от потребностей организации. Это позволяет системе учета кадров расти и адаптироваться к изменениям в структуре и объемах данных без существенных изменений в архитектуре базы данных.

Благодаря реляционной модели, система учета кадров будет обладать высокой надежностью и эффективностью, что делает ее идеальной в использовании для нашей нестандартной организации.

Глава 3 Реализация и тестирование программного обеспечения информационной системы учета кадров

3.1 Выбор инструментов и технологии

Основными компонентами, используемыми в проекте, стали база данных Microsoft Access и язык программирования Python.

Microsoft Access был выбран в качестве основной базы данных для системы учёта кадров по следующим причинам:

- удобство и простота. Access предоставляет удобный графический интерфейс для создания и управления базами данных, что значительно упрощает процесс разработки и администрирования. Это особенно важно для малых организаций, где могут отсутствовать специалисты по базам данных.
- интеграция с другими продуктами Microsoft. Microsoft Access хорошо интегрируется с другими продуктами Microsoft Office, такими как Excel и Word. Это облегчает импорт и экспорт данных, а также создание отчетов и других документов на основе данных из базы.
- реляционная модель данных. Access поддерживает реляционную модель данных, что позволяет организовать данные в виде связанных таблиц. Это обеспечивает целостность и согласованность данных, а также упрощает выполнение сложных запросов.
- экономичность. Для малых и средних предприятий Access является экономически выгодным решением, так как он не требует больших затрат на внедрение и обслуживание.
- локальное хранение. Access позволяет хранить данные локально, что может быть преимуществом в контексте безопасности, особенно для организаций с особыми требованиями к защите данных.

Python был выбран в качестве основного языка программирования для разработки графической оболочки и функциональных модулей системы учёта кадров по следующим причинам:

- простота и читаемость кода. Python известен своей простотой и читаемостью кода, что ускоряет процесс разработки и снижает вероятность ошибок. Это особенно важно для проектов, где возможно участие разработчиков с различным уровнем опыта.
- богатая стандартная библиотека. Python обладает обширной стандартной библиотекой, которая включает в себя модули для работы с базами данных, создания графического интерфейса, обработки данных и других задач. Это позволяет быстро реализовать необходимый функционал без необходимости написания большого объема кода с нуля.
- модульность и расширяемость. Благодаря поддержке модульности, Python позволяет легко расширять функционал системы по мере необходимости. Это делает систему более гибкой и масштабируемой.
- библиотеки для графического интерфейса. Python поддерживает несколько библиотек для создания графического интерфейса, таких как Tkinter [17], PyQt и wxPython. Эти библиотеки позволяют создавать удобные и интуитивно понятные интерфейсы для конечных пользователей.
- поддержка интеграции с базами данных. Python предоставляет множество библиотек для работы с различными базами данных, включая Microsoft Access. Библиотеки, такие как pyodbc, позволяют легко интегрировать Python-приложение с базой данных Access.

Выбор Microsoft Access и Python в качестве основных инструментов разработки обусловлен их усиливающим эффектом и способностью удовлетворить требования проекта.

Также локальное хранение данных в Access и возможность реализации дополнительных уровней безопасности на уровне Python-приложения помогают удовлетворить требования к защите данных.

3.2 Контрольный пример реализации проекта системы учёта кадров

Для проверки работоспособности проекта системы учёта кадров необходимо провести тестирование с использованием контрольных данных, для начала обозначим название программы – «Кадровый Учёт».

Рассмотрим пример тестирования функциональности приложения, обратив внимание на ввод данных, взаимодействие с базой данных и обработку результатов.

Тестовые данные – это ключевой компонент для проверки работоспособности основных функций проекта. В нашем случае, тестовые данные будут использоваться из сотрудников, отделов и документов.

Сотрудники:

- Employee_ID: 1, 2, 3;
- First_Name: «Сергей», «Мария», «Андрей»;
- Last_Name: «Дроздов», «Ветрова», «Никифоров»;
- Department_ID: 6, 7, 6;
- Position: «Помощник начальника ОПО», «Делопроизводитель», «Инженер»;
- Salary: 63000, 41000, 60000.

Отделы:

- Department_ID: 6, 7;
- Department_Name: «Организационно-плановый отдел», «Делопроизводитель».

Типы документов:

- «Hiring» (Заявление о трудоустройстве);

- «Termination» (Заявление об увольнении);
- «Leave» (Заявление о предоставлении отпуска);
- «Other» (Иное заявление).

Рассмотрим процесс обработки тестовых данных на рисунках 10-18.

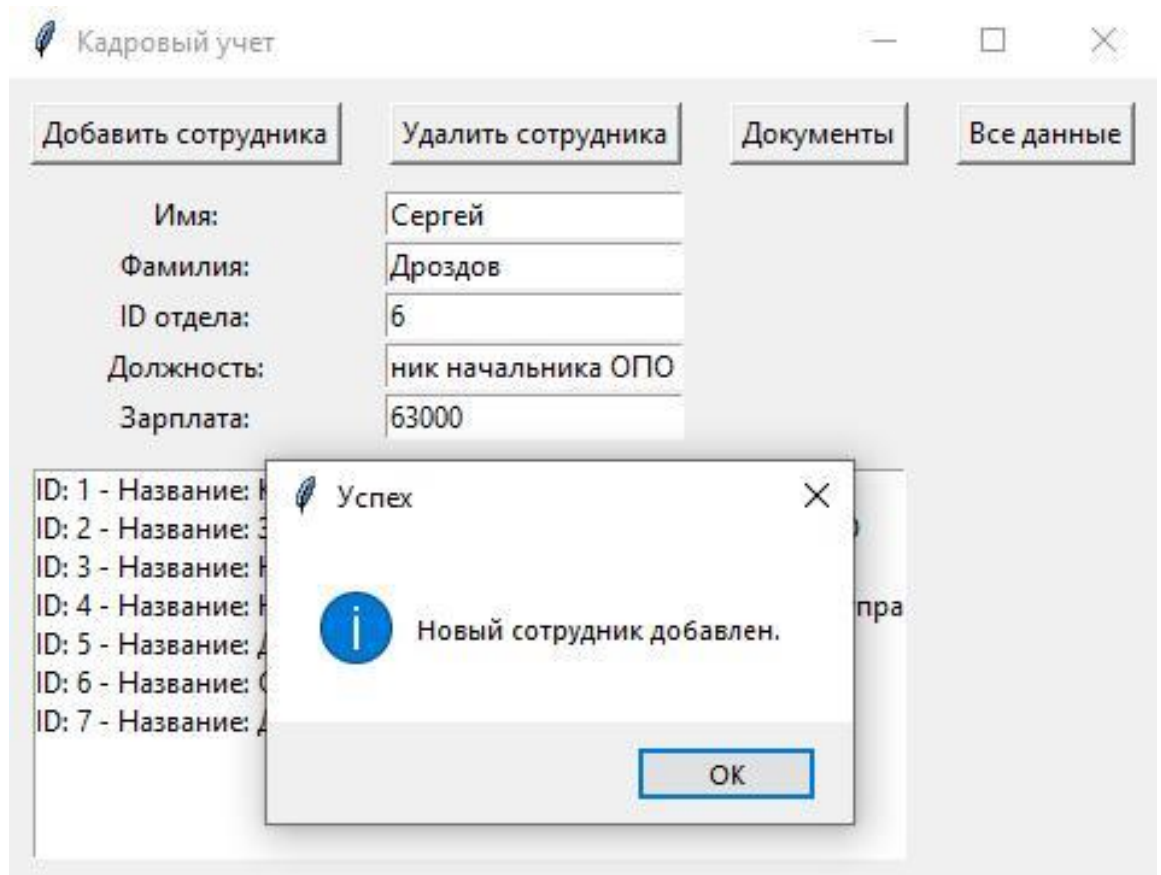


Рисунок 10 – Добавление сотрудника 1

Исходя из введённых корректных данных каждой строки, программа уведомит пользователя об успешном добавлении сотрудника.

В ином случае программа уведомит пользователя об ошибке.

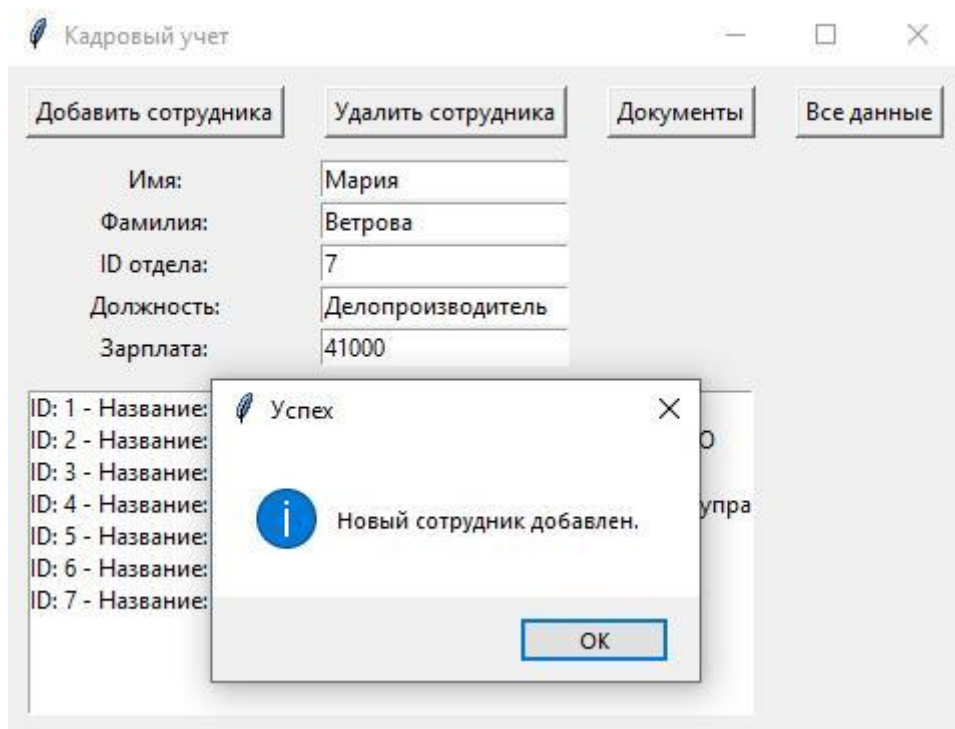


Рисунок 11 – Добавление сотрудника 2

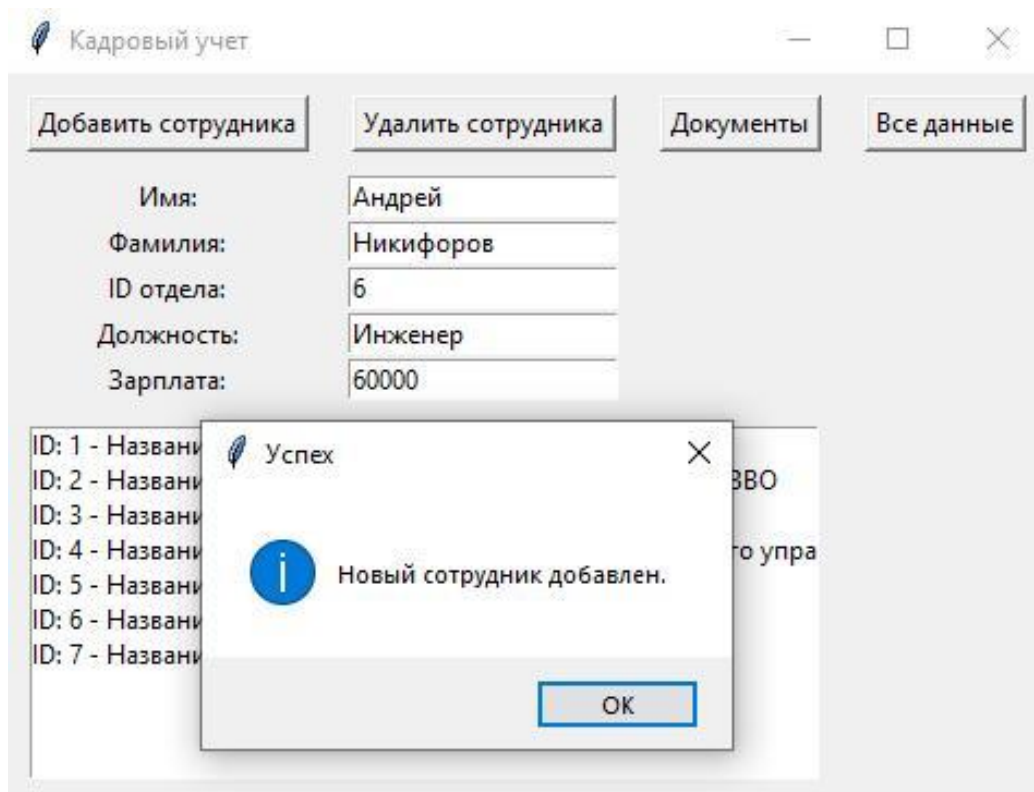


Рисунок 12 – Добавление сотрудника 3

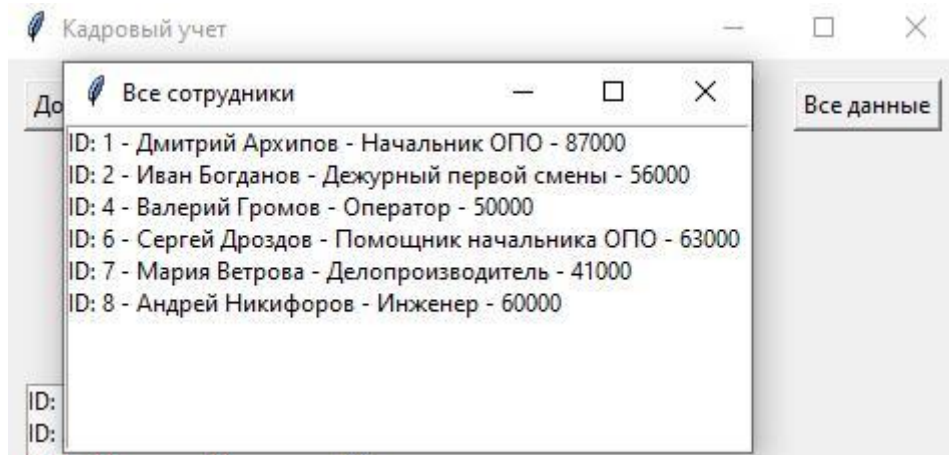


Рисунок 13 – Окно со всеми данными сотрудников в базе

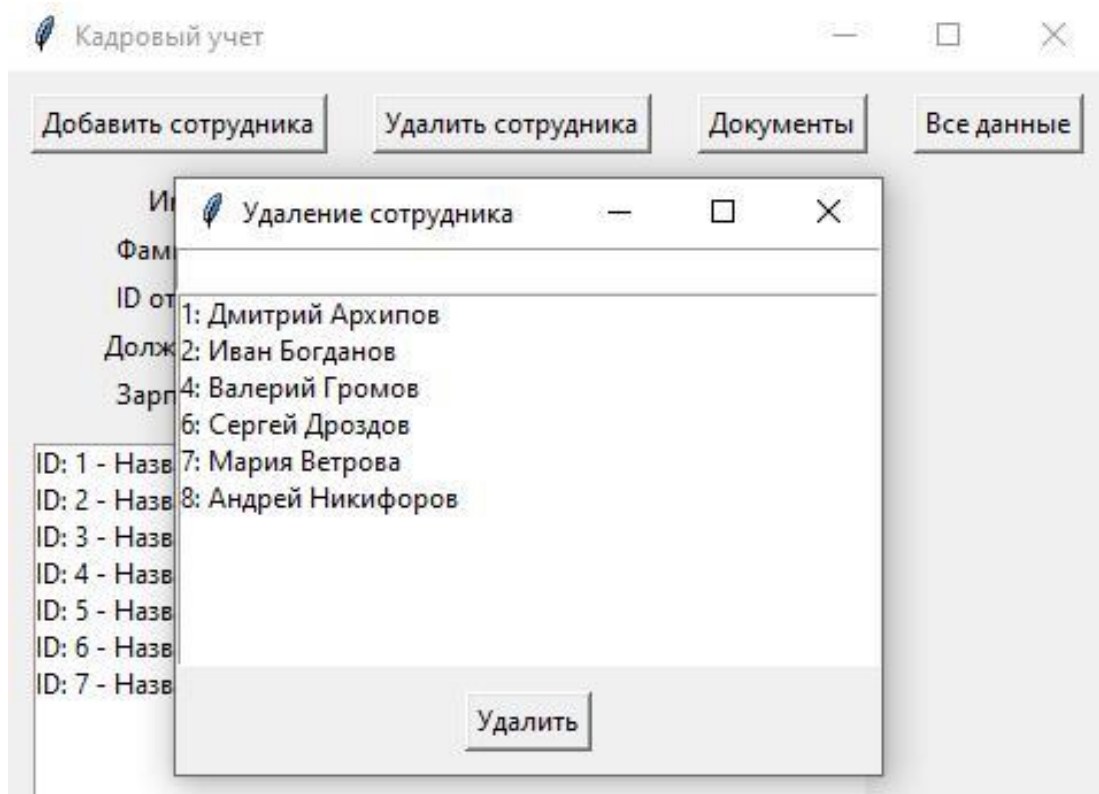


Рисунок 14 – Окно списка со списком сотрудников для удаления

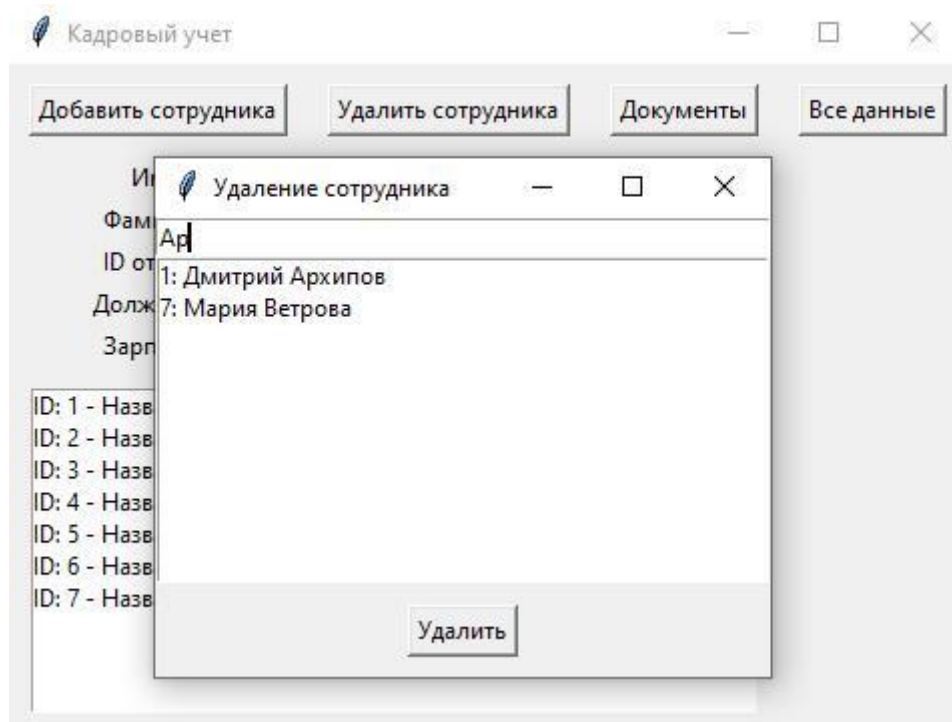


Рисунок 15 – Поиск сотрудника по буквам в имени или фамилии

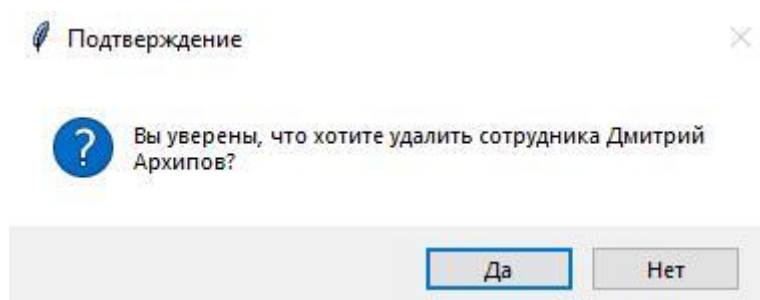


Рисунок 16 – Окно для подтверждения удаления сотрудника

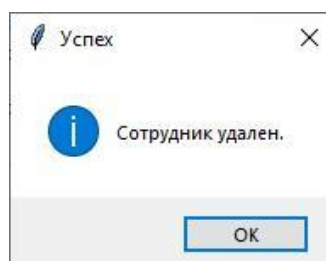


Рисунок 17 – Окно с уведомлением об успешном удалении выбранного сотрудника из базы данных

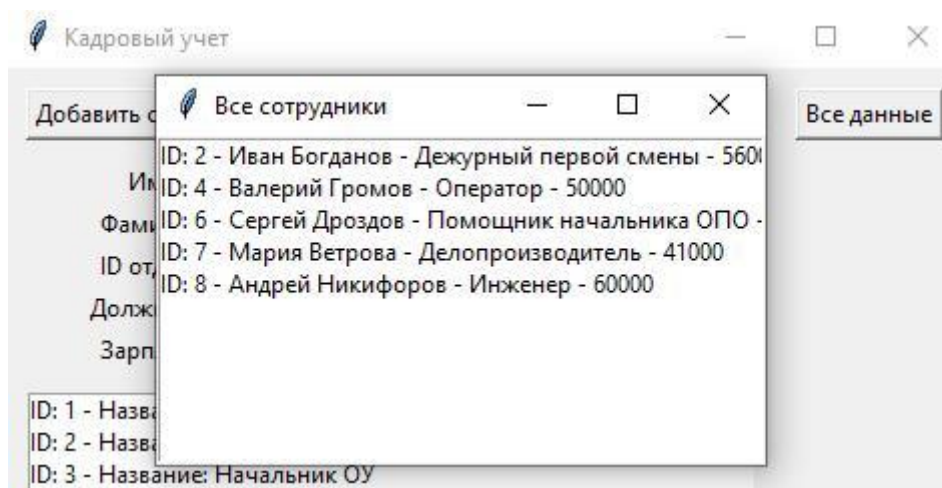


Рисунок 18 – Окно списка всех сотрудников в базе данных после удаления одного из сотрудников

Работа с документами указаны на рисунках 19-21 и включает в себя:

- выбор сотрудника;
- тип заявления;
- создание документа Word с заполненными данными (имя, фамилия, должность).

В нижеуказанном примере документ состоит из двух строк – названия документа и сотрудника, кому данный документ предназначен.

Соответственно шаблон возможно доработать по образцу установленным порядком.

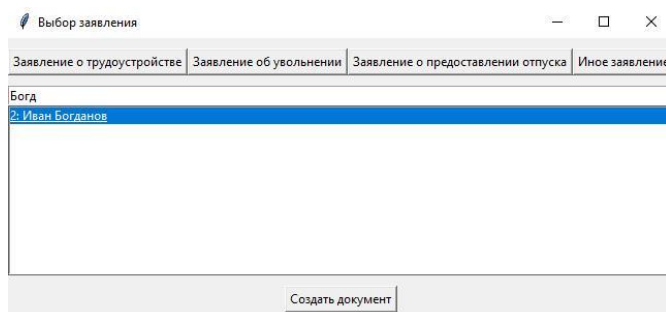


Рисунок 19 – Окно выбора сотрудника с применением поисковой строки

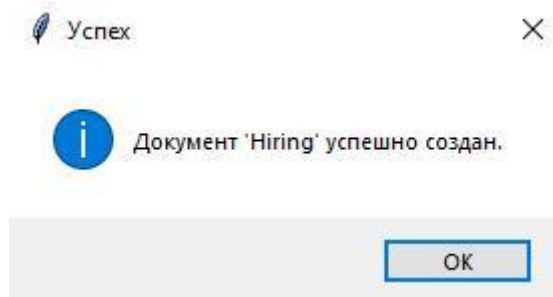


Рисунок 20 – Уведомление об успешном создании документа

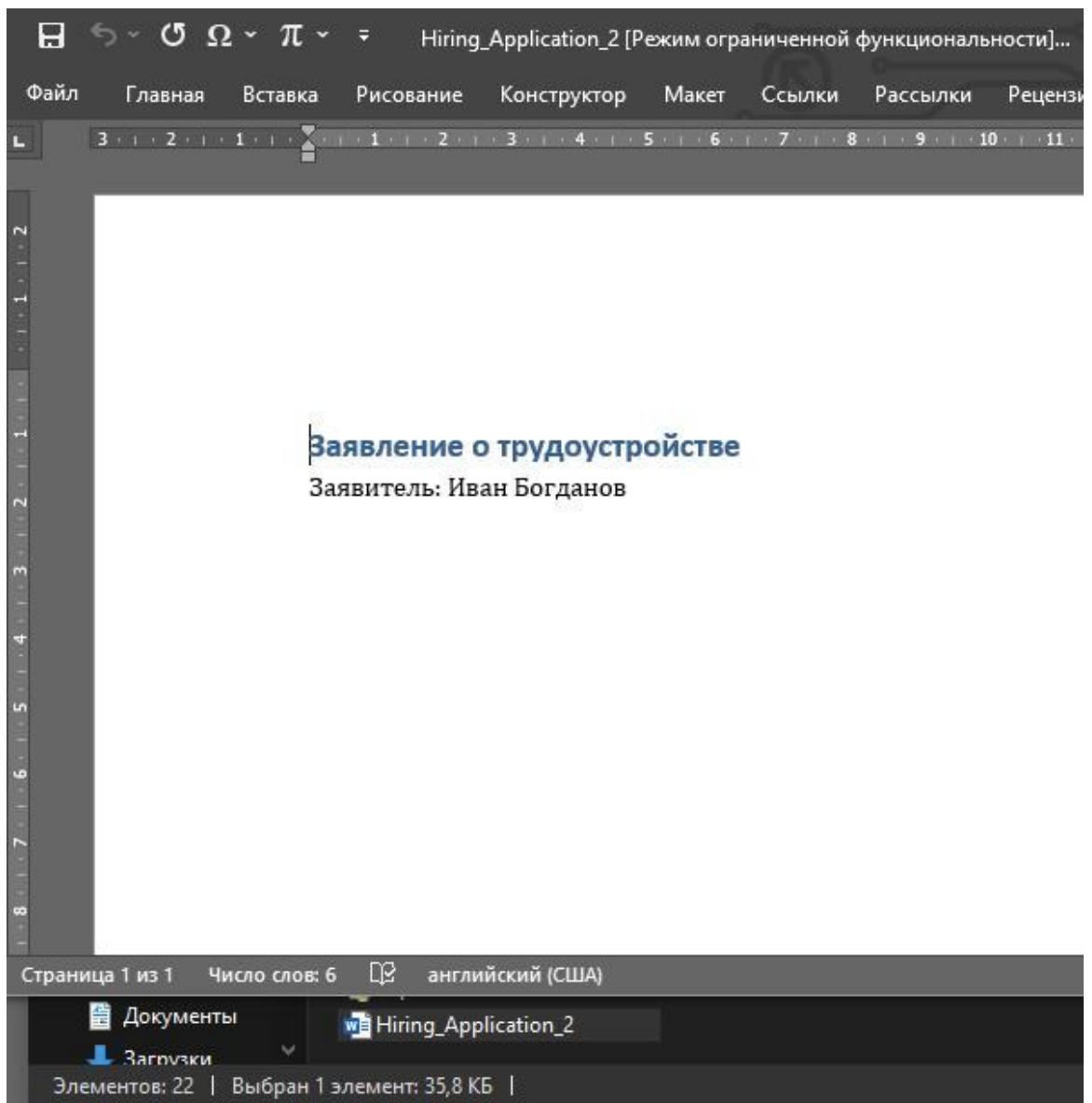


Рисунок 21 – Созданный документ Hiring (заявление о трудоустройстве)

Аналогичным образом создаются другие документы из предлагаемых категорий документов (заявления).

Сообщения и элементы диалога, которые могут возникать в процессе обработки следующие:

- сообщения об ошибке при попытке добавить сотрудника с некорректными данными, например, если зарплата не является числом или не заполнены все необходимые строки/поля;
- подтверждение перед удалением сотрудника;
- сообщение об успешном добавлении/удалении сотрудника;
- сообщение об ошибке, если попытаться удалить несуществующего сотрудника;
- сообщение об успешном создании документа и его сохранении.

Сообщение об ошибке ввода данных сотрудника представлен на рисунке 22.

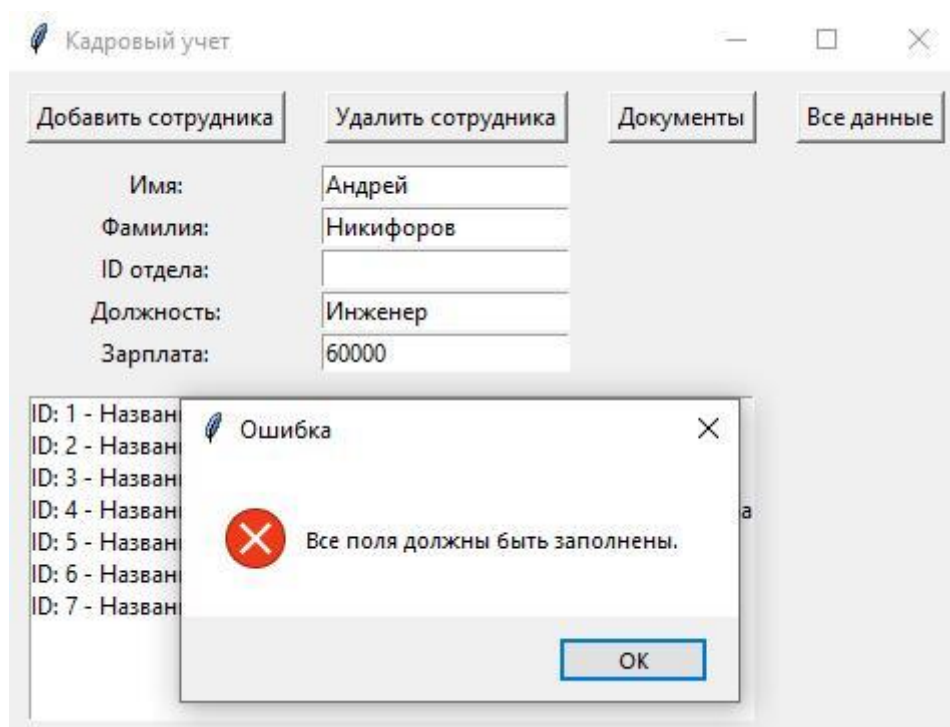


Рисунок 22 – Окно с ошибкой о не всех незаполненных полях

Выводы по третьей главе

Тестирование представляет процесс проверки работоспособности проекта «Кадровый Учёт» с использованием контрольных данных. В данной главе рассмотрен пример тестирования функциональности приложения, включая ввод данных, взаимодействие с базой данных и обработку результатов.

Описана важность тестовых данных для проверки основных функций проекта. Приведены примеры тестовых данных для сотрудников, отделов и документов, а также процесс обработки этих данных, включая добавление сотрудников, удаление сотрудников и работу с документами.

В результате обработки тестовых данных были включены:

- проверка правильности отображения сотрудников и отделов в базе данных;
- корректное удаление сотрудников;
- генерация правильных документов на основе данных сотрудников;
- возможность формирования отчетов о сотрудниках и созданных документах.

Таким образом, подробно рассмотрен процесс тестирования функциональности проекта «Кадровый Учёт» с использованием контрольных данных, что в свою очередь, демонстрирует важность проверки и обработки данных для обеспечения корректной работы приложения.

Заключение

В ходе проекта по разработке программного обеспечения системы учёта кадров была создана автоматизированная информационная система, предназначенная для управления и оптимизации кадровых процессов в организации. Система обеспечивает автоматизированный учёт сотрудников, позволяет создавать документы о приёме на работу, увольнении и переводе, а также взаимодействовать с различными формами отчетности. Достижение цели данного проекта обеспечило внедрение автоматизации ряда бизнес-процессов, связанных с кадровой деятельностью, что привело к сокращению времени на обработку документов и улучшению точности учёта.

Разработанная система предоставляет пользователям гибкость в формировании необходимых данных, а также обеспечивает интуитивный и простой в использовании интерфейс, что упрощает работу кадрового отдела. Система позволяет ускорить выполнение кадровых операций, а также предоставляет удобные механизмы для управления документами, связанными с кадровыми процессами. Это, в свою очередь, облегчает управление персоналом и повышает эффективность работы отдела кадров.

Интерфейс системы оптимизирован для конечного пользователя и обеспечивает интуитивно понятное взаимодействие с элементами управления. Кроме того, автоматизация процессов ведёт к сокращению ошибок и повышению производительности труда сотрудников, что напрямую влияет на общее повышение эффективности организации.

Одним из ключевых моментов в разработке системы является обеспечение безопасности данных. Поскольку в системе хранятся персональные данные сотрудников, была принята стратегия, направленная на предотвращение несанкционированного доступа к чувствительной информации. Система обеспечивает необходимые механизмы контроля доступа и безопасности, предотвращая утечку данных и возможное злоупотребление.

В рамках проекта был проведён анализ предметной области, определены основные функциональные требования, а также спроектированы модели данных и структуры классов системы. Особое внимание уделялось созданию интуитивного интерфейса, который позволил упростить использование системы конечными пользователями.

Также было построено взаимодействие между базой данных Microsoft Access и языком программирования Python, что обеспечило гибкость и расширяемость для будущих разработок.

В конечном итоге, предложенная система выступает как эффективное средство для автоматизации отдела кадров, сокращая трудозатраты и обеспечивая надежное управление кадровыми процессами. Данное решение обладает потенциалом для расширения и интеграции с другими системами, что делает его гибким инструментом для оптимизации бизнес-процессов в организации.

Список используемой литературы

1. Букатов, А. А. Методы и средства интеграции независимых баз данных в распределенных телекоммуникационных сетях: монография / А. А. Букатов, А. В. Пыхалов. – Ростов-на-Дону: Издательство Южного федерального университета, 2013. – 160 с. – ISBN 978-5-9275-1189-1. – Текст электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/47007.html> (дата обращения: 25.05.2024).
2. Головицына, М. В. Информационные технологии в экономике: учебное пособие / М. В. Головицына. – 4-е изд. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. – 589 с. – ISBN 978-5-4497-2401-4. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/133942.html> (дата обращения: 25.05.2024).
3. Дорофиенко, В. В. Управление трудовой деятельностью на малом предприятии : учебное пособие / В. В. Дорофиенко, Н. Е. Муромец, В. В. Лоскутова. – Донецк: Донецкая академия управления и государственной службы, 2020. – 342 с. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/123514.html> (дата обращения: 25.05.2024).
4. Емельянцеv, Н. В. Служба управления персоналом. Ч.1: учебное пособие в 2-х частях / Н. В. Емельянцеv. – Симферополь: Университет экономики и управления, 2018. – 158 с. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/83941.html> (дата обращения: 25.05.2024).
5. Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие / Т. М. Зубкова. – Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2017. – 469 с. – ISBN 978-5-7410-1785-2. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/78846.html> (дата обращения: 25.05.2024).

6. Информационные системы и технологии в экономике и управлении. Техническое и программное обеспечение: учебное пособие / Е. В. Акимова, Д. А. Акимов, Е. В. Катунцов, А. Б. Маховиков. – Саратов: Вузовское образование, 2016. – 190 с. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/47673.html> (дата обращения: 25.05.2024).

7. Колмогорова, С. М. Прикладное программное обеспечение. Microsoft Office Access: практикум / С. М. Колмогорова. – Москва: Ай Пи Ар Медиа, 2024. – 106 с. – ISBN 978-5-4497-2647-6. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/137498.html> (дата обращения: 25.05.2024).

8. Михеев, А. Г. Процессное управление на свободном программном обеспечении: учебное пособие / А. Г. Михеев. – 4-е изд. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. – 230 с. – ISBN 978-5-4497-2479-3. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/133973.html> (дата обращения: 25.05.2024).

9. Мкртычев, С. В. Прикладная информатика. Бакалаврская работа: электрон. учеб.-метод. пособие / С. В. Мкртычев, О. М. Гущина, А. В. Очеповский; Тольяттинский государственный университет. – Тольятти: Изд-во ТГУ, 2019. – ISBN 978-5-8259-1386-5. – Текст: электронный.

10. Петренко, В. И. Защита персональных данных в информационных системах: учебное пособие / В. И. Петренко. – Ставрополь: Северо-Кавказский федеральный университет, 2016. – 201 с. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/66023.html> (дата обращения: 25.05.2024).

11. Самуйлов, С. В. Объектно-ориентированное моделирование на основе UML: учебное пособие / С. В. Самуйлов. – Саратов: Вузовское образование, 2016. – 37 с. – Текст: электронный // Цифровой образовательный

ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/47277.html> (дата обращения: 25.05.2024).

12. Смирнов, А. А. Разработка прикладного программного обеспечения: учебное пособие / А. А. Смирнов. – Москва: Евразийский открытый институт, Московский государственный университет экономики, статистики и информатики, 2003. – 101 с. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/10808.html> (дата обращения: 25.05.2024).

13. Токмаков, Г. П. Базы данных: модели и структуры данных, язык SQL, программирование баз данных: учебное пособие / Г. П. Токмаков. – Ульяновск: Ульяновский государственный технический университет, 2021. – 362 с. – ISBN 978-5-9795-2184-8. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/121263.html> (дата обращения: 25.05.2024).

14. Целых, А. Н. Адаптивные информационные системы для поддержки принятия решений: монография / А. Н. Целых, Л. А. Целых, С. А. Барковский. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2018. – 231 с. – ISBN 978-5-9275-2780-9. – Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. – URL: <https://www.iprbookshop.ru/87696.html> (дата обращения: 25.05.2024).

15. GitHub [Электронный ресурс]. Ссылка на исходный код URL: <https://github.com/FantazeroV/registration/blob/main/persona> (дата обращения: 25.05.2024).

16. OrangeHRM [Электронный ресурс]. URL: <https://www.orangehrm.com/en/company/about-us> (дата обращения: 25.05.2024).

17. Python [Электронный ресурс] URL: <https://docs.python.org/3/library/tkinter.html> (дата обращения: 25.05.2024).

18. Sentrifugo HRMS [Электронный ресурс]. URL: <http://www.sentrifugo.com/features> (дата обращения: 25.05.2024).

19. Software Requirements [Электронный ресурс]. URL: <http://beervolume.com/oop/2020/software-requirements> (дата обращения: 25.05.2024).

20. StarUML [Электронный ресурс]. URL: <https://staruml.io/> (дата обращения: 25.05.2024).

Приложение А

Фрагмент кода подключения Python к базе данных MS Access

```
# Функция для подключения к базе данных Microsoft Access
def connect_to_db():
    try:
        # Параметры подключения: драйвер и путь к файлу базы данных
        conn = pyodbc.connect(
            r'DRIVER={Microsoft Access Driver (*.mdb,
            *.accdb)};DBQ=D:\VKRdb.accdb;'
        )
        return conn # Возврат объекта соединения, если подключение успешно
    except Exception as e:
        # В случае ошибки подключения отображается сообщение об ошибке
        messagebox.showerror("Ошибка подключения к БД", f"Не удалось
        подключиться к базе данных: {str(e)}")
        return None # Возврат None, если не удалось подключиться
#Функция для подтверждения создания документа
def confirm_document():
    selection = employee_listbox.curselection() # Получение выбранного
    сотрудника из списка
    if not selection:
        # Если сотрудник не выбран, отображается сообщение об ошибке
        messagebox.showerror("Ошибка", "Пожалуйста, выберите сотрудника.")
        return
    employee_index = selection[0] # Индекс выбранного сотрудника
    employee_id = employees[employee_index][0] # Идентификатор сотрудника
    doc = docx.Document() # Создание нового документа Word
    # Добавление заголовка и информации о заявителе в документ
    doc.add_heading(document_type_map[document_type], level=1)
```


Продолжение Приложения А

```
doc.add_paragraph("Заявитель: " + f"{employees[employee_index][1]}  
{employees[employee_index][2]}")
```

Сохранение документа с именем, включающим тип документа и идентификатор сотрудника

```
doc.save(f"{document_type}_Application_{employee_id}.docx")
```