НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»**

Колледж «Синергия»

|  |  |  |  |
| --- | --- | --- | --- |
| **Специальность** | 09.02.07 | **Кафедра** | ЦЭ |
|  | *(код)* |  | *(аббревиатура)* |

**ДИПЛОМНый проект**

|  |  |
| --- | --- |
| На тему | автоматизация процесса подбора |
| персонала на примере ооо «Кранк» | |
|  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Обучающийся | | Плевако Михаил Павлович |  | **\_\_\_\_\_\_\_\_\_\_\_** |
|  | (Ф.И.О. полностью) | |  | (подпись) |
| Руководитель | Ребус Наталья Анатольевна | |  | **\_\_\_\_\_\_\_\_\_\_\_** |
|  | (Ф.И.О. полностью) | |  | (подпись) |

**МОСКВА 2024 г.**

**ЗАДАНИЕ**

на дипломный проект

обучающемуся

|  |
| --- |
| Плевако Михаил Павлович |

**1. Тема дипломного проекта:**

|  |
| --- |
| Автоматизация процесса подбора персонала на примере ООО «Кранк» |

**2. Структура дипломного проекта:**

**Введение**

**Глава 1.** **Аналитическая часть**

* 1. Технико-экономическая характеристика предметной области и предприятия. Анализ деятельности «КАК ЕСТЬ»
     1. Характеристика предприятия и его деятельности
     2. Организационная структура управления предприятием
     3. Программная и техническая архитектура ИС предприятия
  2. Характеристика комплекса задач, задачи и обоснование необходимости автоматизации
     1. Выбор комплекса задач автоматизации и характеристика существующих бизнес-процессов
     2. Определение места проектируемой задачи в комплексе задач и ее описание
     3. Анализ информационных потоков проектируемой задачи
     4. Анализ системы обеспечения информационной безопасности и защиты информации
  3. Анализ существующих разработок и выбор стратегии автоматизации «КАК ДОЛЖНО БЫТЬ»
     1. Анализ существующих разработок для автоматизации задачи
     2. Выбор и обоснование стратегии автоматизации задачи
     3. Выбор и обоснование способа приобретения ИС для автоматизации задачи

**Глава 2. Проектная часть**

* 1. Разработка проекта автоматизации
     1. Этапы жизненного цикла проекта автоматизации
     2. Ожидаемые риски на этапах жизненного цикла и их описание
  2. Информационное обеспечение задачи
     1. Характеристика нормативно-справочной, входной и оперативной информации
     2. Характеристика результатной информации
  3. Программное обеспечение задачи
     1. Сценарий диалога
     2. Характеристика базы данных
     3. Структурная схема пакета (дерево вызова программных модулей)
     4. Описание программных модулей
  4. Испытания разработанного решения

2.4.1 Перечень объектов и функций, подлежащих испытаниям

2.4.2 Методы проведения испытаний

2.4.3 Проведение проверочных испытаний и их результаты

**Глава 3.** **Обоснование экономической эффективности проекта**

* 1. Выбор и обоснование методики расчета экономической эффективности
  2. Расчет показателей экономической эффективности проекта

**Заключение.**

В «Заключении» необходимо подвести итоги дипломного проектирования. Раскрыть содержание основных выводов, сделанных обучающимся, представить краткую характеристику результатов, полученных в ходе решения поставленных во «Введении» задач и, тем самым, ответить на основной вопрос дипломного проекта: о степени достижимости поставленной цели.

**Список использованной литературы.**

В «Список использованной литературы» приводятся только те информационные источники, которые автор лично использовал при написании данного дипломного проекта. Причем ссылки на данную литературу и информационные источники обязательны по всему тексту работы. Заимствованные чужие тексты в обязательном порядке заключаются в кавычки, как принадлежащие другому автору. Сноски приводятся постранично нарастающим итогом от № 1 до № N. Сноски, используемые обучающимся, должны быть отражены в списке использованной литературы в конце работы.

**Приложение** обязательно должно содержать фрагмент листинга программного кода(на исходном языке программирования отлаженных основных расчетных модулей – около 400 операторов языка высокого уровня или адаптированных программных средств, использованных в работе), также могут быть приведены:

* схемы или таблицы из основной части дипломного проекта;
* результаты выполнения контрольного примера;
* диаграммы потоков данных, демонстрирующих существующую технологию решения задач;
* диаграммы потоков данных, демонстрирующих предлагаемую технологию решения задач;
* схемы документооборота;
* примеры классификаторов;
* формы первичных и результатных документов;
* распечатки меню, экранных форм ввода, получаемых отчетов в разработанной системе;
* а также другие материалы дипломного проекта, кроме текстов договоров с клиентами и иных «шаблонных документов» (в тех случаях, когда для их существенных реквизитов проектируется форма, а по результатам ввода и сохранения в информационную базу имеется возможность распечатки документа «по шаблону»).

В одном приложении нельзя размещать различные по смыслу таблицы или рисунки. Не допускается дублирование в приложении материала, размещенного в основной части дипломного проекта.

С детальным рассмотрением содержания каждого пункта, а также примерами схем и таблиц необходимо ознакомиться в «Методических рекомендациях по выполнению дипломного проекта».

**4. Исходные данные по дипломному проекту:**

***Основная литература:***

1. Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации» // <https://docs.cntd.ru/document/901990051/titles/64U0IK?ysclid=ltfq8y68gb365029912>
2. Грекул, В. И.  Проектирование информационных систем : учебник и практикум для среднего профессионального образования / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. — 2-е изд. — Москва : Издательство Юрайт, 2024. — 423 с. — (Профессиональное образование). — ISBN 978-5-534-17836-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/543034>.
3. Зараменских, Е. П.  Информационные системы: управление жизненным циклом : учебник и практикум для среднего профессионального образования / Е. П. Зараменских. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 497 с. — (Профессиональное образование). — ISBN 978-5-534-16179-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/542807>.
4. Стружкин, Н. П.  Базы данных: проектирование : учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2024. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/542792>.
5. Системы управления технологическими процессами и информационные технологии : учебное пособие для среднего профессионального образования / В. В. Троценко, В. К. Федоров, А. И. Забудский, В. В. Комендантов. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2024. — 136 с. — (Профессиональное образование). — ISBN 978-5-534-09939-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/539749>.

***Дополнительная литература:***

1. Богатырев, В. А.  Надежность информационных систем : учебное пособие для среднего профессионального образования / В. А. Богатырев. — Москва : Издательство Юрайт, 2023. — 318 с. — (Профессиональное образование). — ISBN 978-5-534-15205-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/520442>.
2. Иванов, В. М.  Интеллектуальные системы : учебное пособие для среднего профессионального образования / В. М. Иванов ; под научной редакцией А. Н. Сесекина. — Москва : Издательство Юрайт, 2024. — 93 с. — (Профессиональное образование). — ISBN 978-5-534-07819-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/541299>.
3. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 293 с. — (Профессиональное образование). — ISBN 978-5-534-16217-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/538370>.

***Интернет-ресурсы:***

| **№** | **Наименование портала**  **(издания, курса, документа)** | **Ссылка** |
| --- | --- | --- |
| 1. | Особенности проектирования информационных систем (cyberleninka.ru) | <https://cyberleninka.ru/article/n/osobennosti-proektirovaniya-informatsionnyh-sistem/viewer> |
| 2. | Проектирование информационных систем | <https://moodle.kstu.ru/course/view.php?id=4638> |
| 3. | Техническое задание на разработку информационной системы | <https://dynamicsun.ru/blog/tz-na-inform-systemu.html> |
| 4. | Информационная система (ИС) | <https://processmi.com/terms/informaczionnaya-sistema-is/> |
| 5. | Информационные системы, понятие состав и структура | <https://helpiks.org/8-10945.html> |



Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_Ребус Н.А.\_\_\_\_

подпись расшифровка

Обучающийся задание получил: «17» февраля 2024 г.

Обучающийся: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Плевако М.П.\_\_\_\_\_

подпись расшифровка

[**Введение** 8](#_Toc168844380)

[**Глава 1. Аналитическая часть** 10](#_Toc168844381)

[**1.1.** **Технико-экономическая характеристика предметной области и предприятия. Анализ деятельности «КАК ЕСТЬ»** 10](#_Toc168844382)

[**1.1.1.** **Характеристика предприятия и его деятельности** 10](#_Toc168844383)

[**1.1.2.** **Организационная структура управления предприятием** 13](#_Toc168844384)

[**1.1.3.** **Программная и техническая архитектура ИС предприятия** 16](#_Toc168844385)

[**1.2.** **Характеристика комплекса задач, задачи и обоснование необходимости автоматизации** 28](#_Toc168844386)

[**1.2.1.** **Выбор комплекса задач автоматизации и характеристика существующих бизнес-процессов** 28](#_Toc168844387)

[**1.2.2.** **Определение места проектируемой задачи в комплексе задач и ее описание** 30](#_Toc168844388)

[**1.2.3.** **Анализ информационных потоков проектируемой задачи** 34](#_Toc168844389)

[**1.2.4.** **Анализ системы обеспечения информационной безопасности и защиты информации** 38](#_Toc168844390)

[**1.3.** **Анализ существующих разработок и выбор стратегии автоматизации «КАК ДОЛЖНО БЫТЬ»** 41](#_Toc168844391)

[**1.3.1.** **Анализ существующих разработок для автоматизации задачи** 41](#_Toc168844392)

[**1.3.2.** **Выбор и обоснование стратегии автоматизации задачи** 44](#_Toc168844393)

[**1.3.3.** **Выбор и обоснование способа приобретения ИС для автоматизации задачи** 46](#_Toc168844394)

[**Глава 2. Проектная часть** 48](#_Toc168844395)

[**2.1** **Разработка проекта автоматизации** 48](#_Toc168844396)

[**2.1.1** **Этапы жизненного цикла проекта автоматизации** 48](#_Toc168844397)

[**2.1.2** **Ожидаемые риски на этапах жизненного цикла и их описание** 54](#_Toc168844398)

[**2.2** **Информационное обеспечение задачи** 59](#_Toc168844399)

[**2.2.1** **Характеристика нормативно-справочной, входной и оперативной информации** 59](#_Toc168844400)

[**2.2.2** **Характеристика результатной информации** 63](#_Toc168844401)

[**2.3** **Программное обеспечение задачи** 67](#_Toc168844402)

[**2.3.1** **Сценарий диалога** 67](#_Toc168844403)

[**2.3.2** **Характеристика базы данных** 68](#_Toc168844404)

[**2.3.3** **Структурная схема пакета (дерево вызова программных модулей)** 72](#_Toc168844405)

[**2.3.4** **Описание программных модулей** 73](#_Toc168844406)

[**2.4** **Испытания разработанного решения** 75](#_Toc168844407)

[**2.4.1** **Перечень объектов и функций, подлежащих испытаниям** 75](#_Toc168844408)

[**2.4.2** **Методы проведения испытаний** 76](#_Toc168844409)

[**2.4.3** **Проведение проверочных испытаний и их результаты** 83](#_Toc168844410)

[**Глава 3. Обоснование экономической эффективности проекта** 88](#_Toc168844411)

[**3.1.** **Выбор и обоснование методики расчёта экономической эффективности** 88](#_Toc168844412)

[**3.2.** **Расчёт показателей экономической эффективности проекта** 90](#_Toc168844413)

[**Заключение** 95](#_Toc168844414)

[**Список использованной литературы** 98](#_Toc168844415)

# **Введение**

Обоснование актуальности выбранной темы «Автоматизация подбора персонала на примере ООО «Кранк»:

* Растущая конкуренция за таланты: на конкурентном рынке труда компаниям необходимо найти и привлечь лучших кандидатов, чтобы оставаться конкурентоспособными.
* Нехватка квалифицированных кадров: во многих отраслях наблюдается нехватка квалифицированных кадров, что затрудняет поиск подходящих кандидатов. Автоматизация может расширить поиск и привлечь кандидатов, которые могли быть упущены.
* Повышение эффективности: автоматизация может значительно повысить эффективность процесса подбора персонала. Это позволит рекрутёрам сосредоточиться на более стратегических аспектах подбора персонала, таких как построение отношений между кандидатами и оценка их соответствия культуре компании.
* Снижение предвзятости: Автоматизация может помочь снизить предвзятость в процессе подбора персонала, обеспечивая объективную оценку кандидатов.

Цель исследования выбранной темы: разработка и оценка системы автоматизации процесса подбора персонала, который позволит компании ООО «Кранк» повысить эффективность и точность подбора кандидатов.

Задачи исследования выбранной темы:

1. Проанализировать существующие методы и технологии автоматизации подбора персонала;
2. Разработать архитектуру и алгоритмы для системы автоматизации подбора персонала;
3. Реализовать систему автоматизации подбора персонала с использованием современных технологий;
4. Оценить эффективность и точность системы автоматизации подбора персонала.

Объект исследования. Объектом исследования является процесс подбора персонала в компании ООО «Кранк».

Предмет исследования. Предметом исследования является разработка и оценка системы автоматизации процесса подбора персонала.

Научная и информационная база исследования:

* Теории и методы автоматизации процесса подбора персонала;
* Лучшие практики в области автоматизации подбора персонала;
* Научные статьи, книги и отчёты по теме автоматизации подбора персонала.

# **Глава 1. Аналитическая часть**

* 1. **Технико-экономическая характеристика предметной области и предприятия. Анализ деятельности «КАК ЕСТЬ»**
     1. **Характеристика предприятия и его деятельности**

Объект исследования – ООО «Кранк». «Кранк» – компания, которая разрабатывает и издаёт 2D и 3D игры для персональных компьютеров. С момента своего основания в 2018 году компания создала множество игр, которые завоевали признание критиков и преданных поклонников по всему миру.

Компания «Кранк» гордится своей командой опытных и талантливых разработчиков, которые создают захватывающие и инновационные игры. Команда состоит из художников, дизайнеров, программистов и менеджеров проектов, которые работают вместе в творческой и динамичной среде.

Компания «Кранк» использует одни из самых современных игровых движков и технологий, такие как Unreal Engine 4 и Unity. Это позволяет ей создавать игры с качественной графикой, плавным игровым процессом и захватывающим звуковым сопровождением.

Основной деятельностью компании является создание и издательство 2D и 3D игр.

Дополнительные виды деятельности:

* Консалтинг и разработка игр по контракту;
* Услуги по локализации;
* Участие в выставках и конференциях;
* Поддержка сообщества;

Динамику и общие цифры можно посмотреть в таблице1.

**Таблица 1**

**Количественно-стоимостные оценки и ограничения**

| **№ п/п** | **Наименование характеристики (показателя)** | **2019** | **2020** | **2021** | **2022** | **Ограничения/Примечания** |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | Выручка (млн. руб.) | 15,825 | 18,378 | 22,567 | 25,934 | Выручка включает в себя все доходы от деятельности компании. |
| 2. | Прибыль (млн. руб.) | 3,273 | 4,392 | 5,860 | 6,749 | Прибыль является общей прибылью до вычета налогов и процентов. |
| 3. | Кредиторский долго (млн. руб.) | 2,305 | 3,250 | 4,102 | 5,740 | Кредиторская задолженность представляет собой общую сумму задолженности компании перед её кредиторами. |
| 4. | Дебиторский долг (млн. руб.) | 1,453 | 1,567 | 2,793 | 2,057 | Дебиторская задолженность представляет собой общую сумму задолженности должников перед компанией. |
| 5. | Чистые активы (млн. руб.) | 10,023 | 12,532 | 15,920 | 18,720 | Чистые активы представляют собой разницу между активами и пассивами компании. |
| 6. | Активы (млн. руб.) | 20,456 | 25,824 | 30,193 | 35,384 | Активы включают в себя все имущество и ресурсы, принадлежащие компании. |
| 7. | Кол-во разрабатываемых игр | 6 | 4 | 8 | 10 | Количество разрабатываемых игр включает ка новые, так и текущие проекты. |
| 8. | Средняя зарплата отдела кадров | 40,000 | 40,500 | 70,000 | 80,000 | Средняя зарплата отдела кадров включает в себя зарплаты всех сотрудников отдела. |
| 9. | Средняя зарплата отдела разработки | 50,000 | 65,000 | 70,000 | 77,600 | Средняя зарплата отдела разработки включает в себя зарплаты всех сотрудников отдела, включая программистов, дизайнеров и других специалистов по разработке. |
| 10. | Средний рейтинг игр на Metacritic | 77 | 80 | 82 | 85 | Средний рейтинг игр на Metacritic представляет собой среднее значение оценок всех игр, выпущенных компанией в соответствующем году. Оценки варьируются от 0 до 100, при этом более высокие оценки указывают на более высокие рейтинги. |
| 11. | Затраты на разработку одной игры (млн. руб.) | 2,300 | 2,500 | 4,000 | 5,500 | Затраты на разработку одной игры представляют собой общую сумму затрат, вложенную на разработку одной игры. |
| 12. | Доля рынка в России | 10% | 12% | 13% | 20% | Доля рынка в России представляет собой процентную долю рынка видеоигр в России, занимаемую компанией. |

Благодаря таблице 1 можно увидеть, что финансовые показатели компании ООО «Кранк» указывают на стабильный и прибыльный бизнес. Компания получает значительную выручку и прибыль, и её активы растут. Хотя кредиторский и дебиторский долг также увеличиваются, темпы роста этих показателей ниже, чем темпы роста выручки, что указывает на эффективное управление оборотным капиталом.

Увеличение затрат на разработку одной игры связано с растущей сложностью и масштабом разрабатываемых игр. Тем не менее, рост доли рынка в России свидетельствует о том, что компания укрепляет свои позиции на внутреннем рынке.

Чистые активы компании растут, что является признаком её финансовой стабильности. Это означает, что у компании достаточно активов для покрытия своих обязательств, и она находится в хорошем положении для дальнейшего роста и инвестиций.

* + 1. **Организационная структура управления предприятием**

В компании ООО «Кранк» все вопросы деятельности компании решаются генеральным директором.

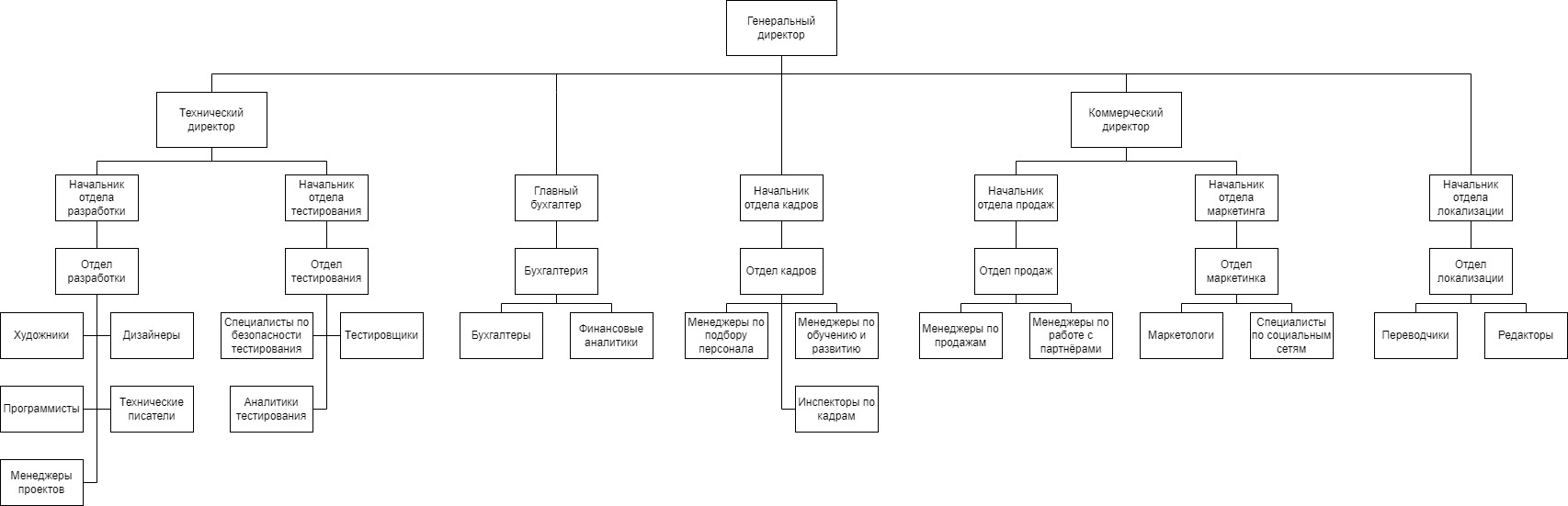
Генеральный директор компании принимает решения о стратегии развития, распределяет обязанности между отделами, контролирует общую деятельность компании, определяет внешнеэкономическую политику. Также он разрешает конфликты между руководителем и сотрудниками.

Определение и понятие организационной структуры управления предприятием. Организационная структура управления предприятием представляет собой систему, которая определяет и организует взаимосвязи между различными подразделениями и должностными лицами внутри компании. Она определяет, как власть, ответственность и информация распределяется по всей организации. Основные элементы организационной структуры включаю в себя:

* Иерархическая структура. Это определение власти и ответственности от верхнего уровня управления до нижних уровней. Обычно включает в себя руководителей, менеджеров и исполнителей.
* Функциональная структура. Предприятие делится на функциональные области в зависимости от основных видов деятельности (например, производство, маркетинг, финансы, человеческие ресурсы).
* Дивизиональная структура. Предприятие разделяется на отдельные дивизии или подразделения в зависимости от географического расположения продукции, рынков или клиентских групп.
* Матричная структура. Предприятие использует комбинацию функциональной и дивизиональной структур, чтобы достичь определённых целей или проектов.
* Сетевая структура. Организация внешне независимых компаний или подразделений, работающих над общими целями.
* Линейно-штабная структура. Одна из самых простых форм, где решения принимаются на верхнем уровне и передаются по вертикали вниз по иерархии.

Организационная структура должна быть гибкой и адаптивной, чтобы эффективно реагировать на изменения внешней среды и достигать стратегических целей компании. Кроме того, она должна обеспечивать чёткое определение ролей и ответственности, чтобы минимизировать конфликты и повысить производительность.

Организационную структуру предприятия можно увидеть на рисунке 1.



**Рисунок 1. Организационная структура предприятия**

Цели функционирования и основные задачи подразделений:

* Название подразделения: отдел разработки;

Основные задачи: создание и разработка новых игровых концепций, включая игровой процесс, повествование и игровой мир.

* Название подразделения: отдел тестирования;

Основные задачи: обеспечение высокого качества продукта и его соответствия требованиям.

* Название подразделения: Бухгалтерия;

Основные задачи: ведение бухгалтерского и налогового учёта, составление первичных документов.

* Название подразделения: отдел кадров;

Основные задачи: определение потребности организации в кадрах и подбор персонала, внедрение системы мотивации труда, ведение трудовой книжки работника.

* Название подразделения: отдел продаж;

Основные задачи: продажи и продвижение продуктов на рынке, привлечение новых клиентов, подготовка отчётов по продажам и показателям эффективности.

* Название подразделения: отдел маркетинга;

Основные задачи: продвижение бренда, изучение целевой аудитории, разработка инструментов для увеличения продаж;

* Название подразделения: отдел локализации;

Основные задачи: перевод игрового текста, диалогов, меню и другой внутриигровой документации на целевые языки. Создание и поддержка согласованного глоссария игровых терминов и фраз для обеспечения единообразия во всех переводах.

* + 1. **Программная и техническая архитектура ИС предприятия**

Определение и описание темы.

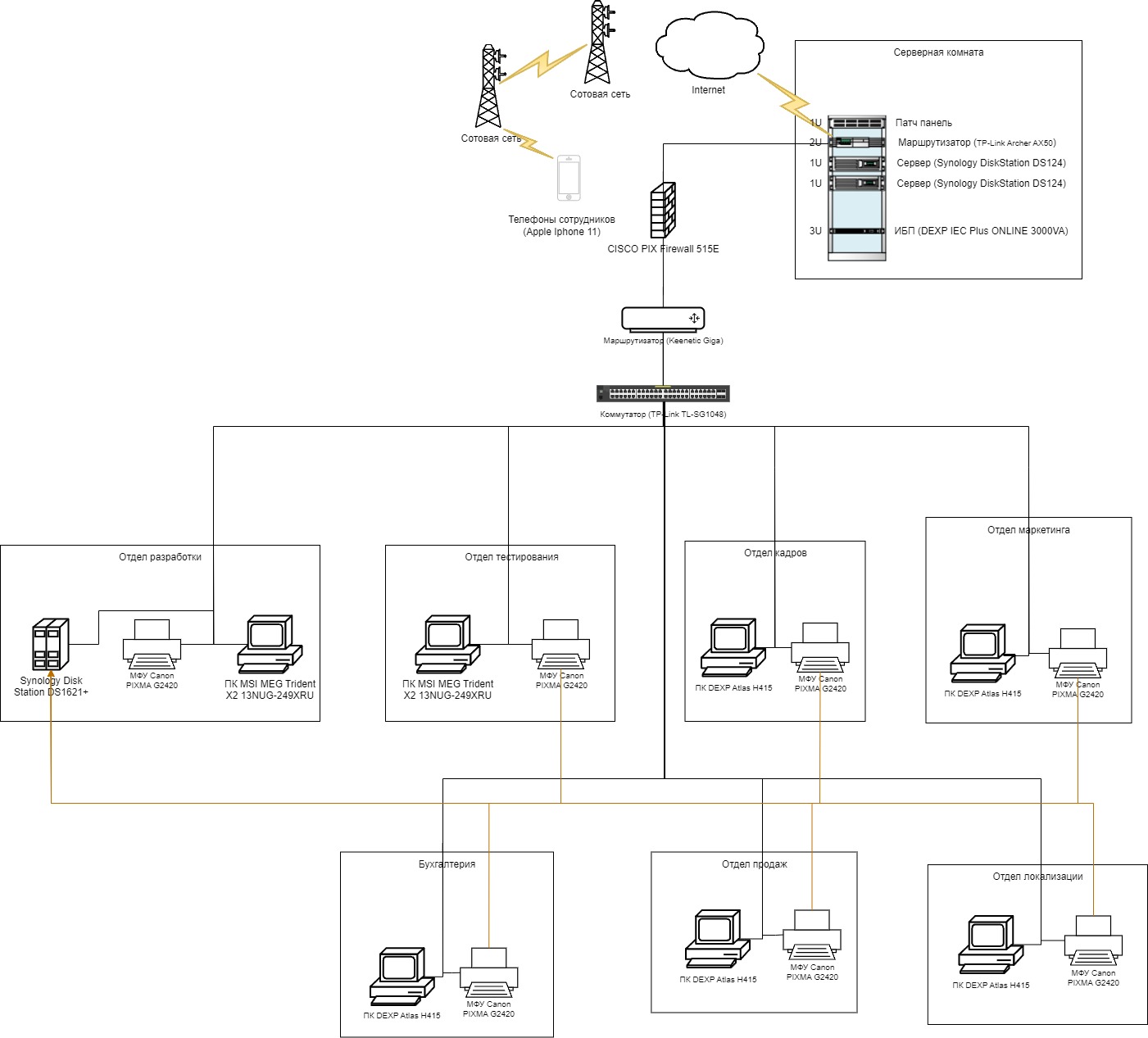
Программная и техническая архитектуры информационной системы предприятия являются ключевыми компонентами её разработки и функционирования. Определения архитектур ИС:

* Программная архитектура: высокоуровневое описание структуры и организации программных компонентов информационной системы. Она включает в себя:
  + Компоненты системы. Это программные модули, приложения и сервисы, которые составляют информационную систему.
  + Интерфейсы. Это способы взаимодействия между компонентами, например, API (интерфейсы программирования приложений), пользовательские интерфейсы и т.д.
  + Логика приложений. Это описание бизнес-логики и алгоритмов, которые определяют функциональность системы.
  + Хранилища данных. Это места для хранения данных, такие как базы данных, файловые системы и т.д.
  + Шаблоны проектирования. Это общие подходы к организации кода и решению типичных проблем проектирования.
* Техническая архитектура: нижестоящий уровень, который определяет аппаратное обеспечение, сетевую инфраструктуру и другие технические аспекты, необходимые для работы информационной системы. Она включает в себя:
  + Серверное оборудование. Это физические серверы, на которых работает система, включая вычислительные ресурсы и хранилища данных.
  + Сетевая инфраструктура. Это коммуникационные средства, необходимые для передачи данных между компонентами системы, такие как сетевое оборудование, протоколы и т.д.
  + Безопасность. Это меры безопасности, такие как аутентификация, авторизация, шифрование данных, и механизмы защиты от вторжений.
  + Отказоустойчивость и масштабируемость. Это механизмы, обеспечивающие непрерывную работу системы даже при сбоях и возможность её масштабирования для обработки роста нагрузки.
  + Интеграция с другими системами. Это взаимодействие с внешними системами, такими как внешние базы данных, облачные сервисы и т.д.

Обе архитектуры тесно взаимосвязаны и влияют друг на друга: программная архитектура определяет, как программные компоненты взаимодействуют, а техническая архитектура обеспечивает необходимую инфраструктуру для функционирования этих компонентов.

В главном помещении компании «Кранк» расположены 8 офисных помещений, которые связаны при помощи 1 коммутатора и 2 маршрутизаторов, подключённых к сети Internet, защищённое соединение обеспечивает межсетевой экран (брандмауэр). Техническую архитектуру предприятия можно увидеть на рисунке 2, на нём присутствуют следующие объекты:

* 2 вида рабочих станций;
* 3 сервера;
* 2 маршрутизатора;
* 1 коммутатор;
* 1 вид МФУ;
* 1 ИБП;
* 1 межсетевой экран.



**Рисунок 2. Техническая архитектура предприятия**

Описание технической архитектуры ИС предприятия описаны в таблице 2.

**Таблица 2**

**Описание технической архитектуры**

| **№ п/п** | **Аппаратное средство** | **Конфигурация** | **Установленное ПО** |
| --- | --- | --- | --- |
| 1. | ПК DEXP Atlas H415 (12 шт.) | Модель CPU: AMD Ryzen 7 5700G  Кол-во ядер: 8  Модель видеокарты: AMD Radeon Graphics  ОЗУ: 16 Гб  Тип памяти: DDR4  Объём SSD: 500 Гб  Интерфейсы: USB 2.0 Type-A x2, USB 3.2 Gen1 Type-A x4, HDMI x1, VGA (D-Sub) x1 | Kaspersky  Jabber  Microsoft Office 2016  OpenSSH  OC Windows 8.1 Home  Windows Defender  Coogle Chrome  Double Commander  Google News  Wireshark  DataFan |
| 2. | ПК MSI MEG Trident X2 13NUG-249XRU (8 шт.) | Модель CPU: Intel Core i7-13700KF  Кол-во ядер: 16  Модель видеокарты: GeForce RTX 4080  ОЗУ: 32 Гб  Тип памяти: DDR5  Объём SSD: 1 Тб SSD M.2  Интерфейсы: USB 2.0 Type-A x4, USB 3.2 Gen1 Type-A x2, DVI-D x1, DisplayPort x1, HDMI x1 | Kaspersky  Wireshark  Microsoft Office 2016  OpenSSH  OC Windows 8.1 Home  Windows Defender  Coogle Chrome  Double Commander  Unity  Unreal Engine 4  Blender  Autodesk 3Ds Max  Autodesk MotionBuilder |
| 3. | МФУ Canon PIXMA G2420 (6 шт.) | Технология печати: Термоструйная  Цветность печати: Цветная, чёрно-белая  Скорость ч-б печати: 9,1 стр./мин  Скорость цветной печати: 5 стр./мин  Кол-во цветов: 4  Интерфейсы: Wi-Fi, 802.11n, USB 2.0 | - |
| 4. | Сервер Synology DiskStation DS124 (2 шт.) | Модель процессора: Realtek RTD1619B  Кол-во ядер: 4  Частота процессора: 1700 МГц  ОЗУ: 1 Гб  Тип памяти: DDR4  Максимальный объём памяти: 16 Тб  Интерфейсы: USB 3.2 Gen1 x2 | Active Directory  WSUS  NetWare |
| 5. | Сервер Synology DiskStation DS1621+ (1 шт.) | Модель процессора: AMD Ryzen V1500B  Кол-во ядер: 4  Частота процессора: 2200 МГц  ОЗУ: 4 Гб  Тип памяти: DDR4  Максимальный объём памяти: 18 Тб  Интерфейсы: USB 3.2 Gen1 x2, PCIe x1 | Microsoft SQL Server  Active Directory  WSUS  Apache  1C: Бухгалтерия 8 |
| 6. | Маршрутизатор TP-Link Archer AX50 (1 шт.) | Тип подключения: Ethernet  Частота процессора: 1.5 ГГц  Кол-во ядер: 2  Стандарт Wi-Fi: 802.11n, 802.11ac, 802.11ax  Интерфейсы: USB 3.0 x1 | - |
| 7. | Маршрутизатор Keenetic Giga (1 шт.) | Тип подключения: Ethernet  Мобильный вход в интернет: 3G, 4G/LTE  Частота процессора: 0.88 ГГц  Кол-во ядер: 2  ОЗУ: 512 Мб  Flash-память: 128 Мб  Стандарт Wi-Fi: 802.11n, 802.11ac, 802.11ax  Интерфейсы: USB 2.0 x1, USB 3.0 x1 | - |
| 8. | ИБП DEXP IEC Plus ONLINE 3000VA (1 шт.) | Вид: с двойным преобразованием  Форм-фактор: Tower  Полная выходная мощность: 3000ВА  Тип формы напряжения: Чистая синусоида  Время работы: 5 мин(2700Вт)  Интерфейсы: USB | - |
| 9. | Коммутатор TP-Link TL-SG1048 (1 шт.) | Вид: Неуправляемый  Уровень коммутатора: L2  Базовая скорость передачи данных: 100 Мбит/сек, 1000 Мбит/сек  Поддержка стандартов: IEEE 802.3ab, IEEE 802.3i, IEEE 802.3u, IEEE 802.3x  Потребляемая мощность: 32.29 Вт | - |
| 10. | Межсетевой экран Cisco PIX Firewall 515E | Процессор: Intel Celeron  Частота процессора: 433 МГц  Flash-память: 16 Мб  Кэш: 128 Кб  Системная шина: 32 бит | - |

Рассмотрим подробнее роли и назначения серверов:

Веб-сервер (Synology DiskStation DS124) имеет много задач, которым требуется постоянное выполнение:

* Хранение и обмен файлами. Хранение и предоставление общего доступа к файлам, таким как игровые активы, документы и маркетинговые материалы для сотрудников компании;
* Управление электронной почтой. Хранение и управление электронной почты сотрудников, что позволяет им получать и отправлять электронные письма, а также управлять своими почтовыми ящиками;
* Виртуализация. Виртуализация серверов и приложений, что позволяет компании запускать несколько операционных систем и приложений на одном физическом сервере, повышая эффективность и гибкость IT-инфраструктуры.

На сервере установлена система управления Apache, которая обеспечивает поддержку модулей для работы с серверными языками программирования, например, Perl, Python, PHP.

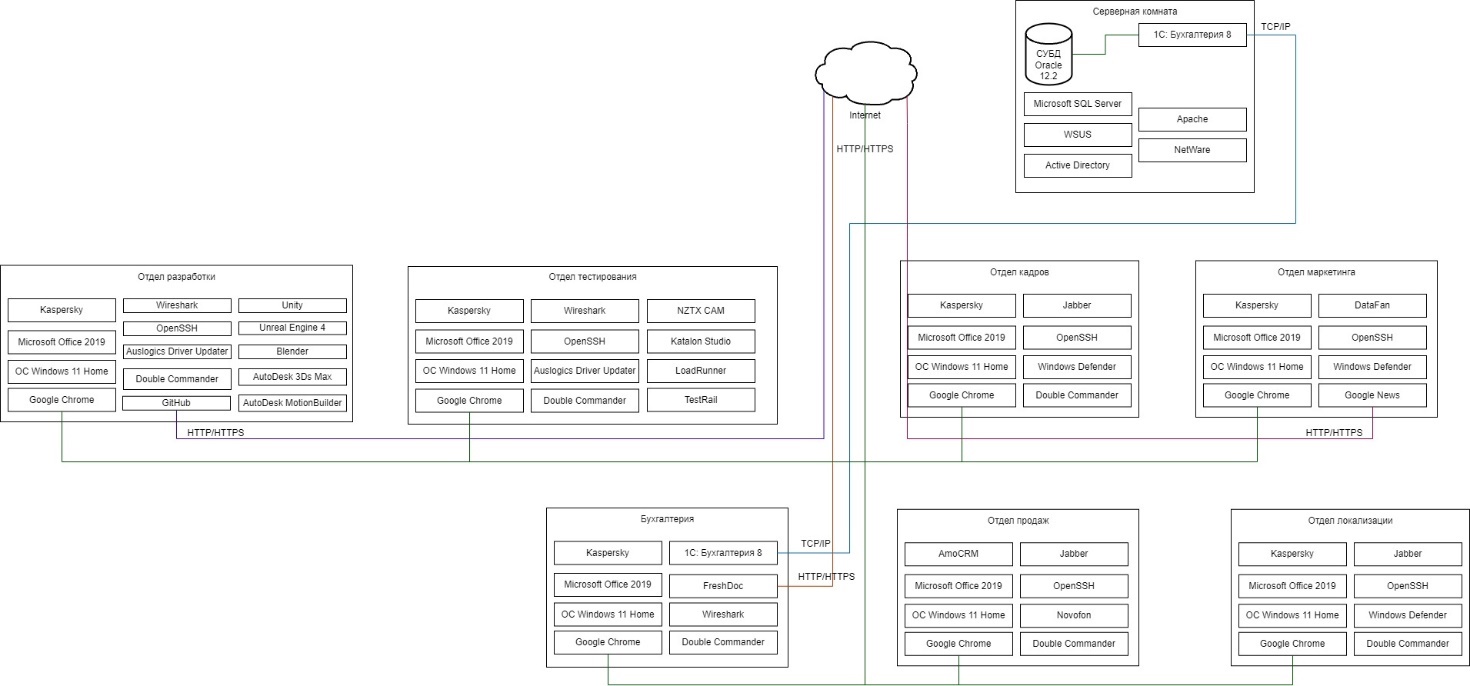
Сервер приложений (Synology DiskStation DS124) обеспечивает интеграцию с базой данных и выполняет следующие задания:

* Мониторинг и оповещения приложений. Мониторинг производительности и работоспособность приложений, и отправка оповещений в случае возникновения проблем, что позволяет IT-персоналу быстро реагировать и устранять неполадки;
* Разработка и тестирование приложений. Разработка и тестирование новых приложений и функций в изолированной и контролируемой среде перед развёртыванием в производственной среде;
* Обеспечение безопасности приложений. Обеспечение безопасности приложений путём внедрения мер безопасности, таких как межсетевые экраны, системы обнаружения вторжений и управление уязвимостями.

Сервер печати (Synology DiskStation DS1621+) выполняет следующие задачи:

* Управление печатью. Сервер позволяет централизованно управлять некоторыми принтерами;
* Совместный доступ к принтерам. Различные сотрудники могут печатать документы на общих принтерах через сервер;
* Управление доступом. Сервер позволяет настраивать права доступа к принтерам для различных пользователей или групп пользователей.

Программную архитектуру информационной системы предприятия можно увидеть на рисунке 3.



**Рисунок 3. Программная архитектура ИС предприятия**

Описание программной архитектуры ИС предприятия:

ПО для серверной комнаты:

* СУБД Oracle 12.2 – объектно-реляционная система управления базами данных от компании Oracle;
* 1C: Бухгалтерия 8 – профессиональный инструмент бухгалтера, с помощью которого можно вести бухгалтерский и налоговый учёт, готовить и сдавать обязательную отчётность;
* Microsoft SQL Server 19.3.4.0 – система управления базами данных, используемая для хранения и извлечения данных из других программных приложений;
* WSUS 12.6.1 – инструмент, предоставляющий системным администраторам возможность управлять жизненным циклом обновлений по всей сети организации;
* Active Directory 1.1 – служба каталогов для операционных систем семейства Windows Server;
* Apache 2.4.59 – открытое кросс-платформенное программное обеспечение для размещения и поддержки веб-сервера;
* NetWare 6.5 – сетевая операционная система и набор сетевых протоколов, которые используются в этой системе для взаимодействия с компьютерами-клиентами, подключёнными к сети.

ПО для отдела разработки:

* Kaspersky 21.17.7.539 – антивирусное программное обеспечение;
* Wireshark 4.2.4 – программа для анализа сетевого трафика;
* Microsoft Office 2019 – офисный пакет американской корпорации Microsoft для разных операционных систем;
* OpenSSH 9.7 – набор программ, предоставляющих шифрование сеансов связи по компьютерным сетям с использованием протокола SSH;
* OC Windows 11 23H2 – семейство проприетарных операционных систем корпорации Microsoft, ориентированных на применение графического интерфейса при управлении;
* Auslogics Driver Updater 1.26.0.1 – инструмент, который позволяет обновлять драйверы на компьютере за один клик;
* Google Chrome 124.0.6367.156 – браузер, разработанный компанией Google;
* Double Commander 1.0.11 – двухпанельный файловый менеджер с открытым исходным кодом;
* GitHub – веб-сервер для хостинга IT-проектов и их совместной разработки;
* Unity 2023.2.17f1 – кроссплатформенный игровой движок, созданный компанией Unity Technologies в 2005 году;
* Unreal Engine 4.27 – игровой движок, разрабатываемый и поддерживаемый компанией Epic Games;
* Blender 4.1.1 – бесплатный и открытый набор программ для 3D-создания.
* AutoDesk 3Ds Max 2023 – профессиональное программное обеспечение для 3D-моделирования, анимации и визуализации при создании игр и проектирования;
* AutoDesk MotionBuilder 2024 – программное обеспечение для 3D-анимации персонажей, созданное компанией Autodesk.

ПО для отдела тестирования:

* Kaspersky 21.17.7.539 – антивирусное программное обеспечение;
* Wireshark 4.2.4 – программа для анализа сетевого трафика;
* Microsoft Office 2019 – офисный пакет американской корпорации Microsoft для разных операционных систем;
* OC Windows 11 23H2 – семейство проприетарных операционных систем корпорации Microsoft, ориентированных на применение графического интерфейса при управлении;
* OpenSSH 9.7 – набор программ, предоставляющих шифрование сеансов связи по компьютерным сетям с использованием протокола SSH;
* Auslogics Driver Updater 1.26.0.1 – инструмент, который позволяет обновлять драйверы на компьютере за один клик;
* Google Chrome 124.0.6367.156 – браузер, разработанный компанией Google;
* Double Commander 1.0.11 – двухпанельный файловый менеджер с открытым исходным кодом;
* NZTX CAM 4.45.3 – небольшая бесплатная программа для мониторинга системы и управления компьютером;
* Katalon Studio 9.4.0 – современная комплексная платформа для управления качеством программного обеспечения;
* LoadRunner 12.55 – инструмент тестирования программного обеспечения от OpenText;
* TestRail 5.5 – инструмент для оптимизации тестирования программного обеспечения.

ПО для отдела кадров:

* Kaspersky 21.17.7.539 – антивирусное программное обеспечение;
* Jabber 14.3.1 – протокол для мгновенного обмена сообщениями и информацией;
* Microsoft Office 2019 – офисный пакет американской корпорации Microsoft для разных операционных систем;
* OpenSSH 9.7 – набор программ, предоставляющих шифрование сеансов связи по компьютерным сетям с использованием протокола SSH;
* OC Windows 11 23H2 – семейство проприетарных операционных систем корпорации Microsoft, ориентированных на применение графического интерфейса при управлении;
* Windows Defender 1000.25992.0.9000 – антивирус компании Microsoft, встроенный по умолчанию в операционные системы Windows;
* Google Chrome 124.0.6367.156 – браузер, разработанный компанией Google;
* Double Commander 1.0.11 – двухпанельный файловый менеджер с открытым исходным кодом.

ПО для отдела маркетинга:

* Kaspersky 21.17.7.539 – антивирусное программное обеспечение;
* DataFan – сервис для постоянного мониторинга данных из социальных сетей с помощью динамических отчётов;
* Microsoft Office 2019 – офисный пакет американской корпорации Microsoft для разных операционных систем;
* OpenSSH 9.7 – набор программ, предоставляющих шифрование сеансов связи по компьютерным сетям с использованием протокола SSH;
* OC Windows 11 23H2 – семейство проприетарных операционных систем корпорации Microsoft, ориентированных на применение графического интерфейса при управлении;
* Windows Defender 1000.25992.0.9000 – антивирус компании Microsoft, встроенный по умолчанию в операционные системы Windows;
* Google Chrome 124.0.6367.156 – браузер, разработанный компанией Google;
* Google News 3.1.1 – агрегатор новостей от компании Google Inc.

ПО для бухгалтерии:

* Kaspersky 21.17.7.539 – антивирусное программное обеспечение;
* 1C: Бухгалтерия 8 – профессиональный инструмент бухгалтера, с помощью которого можно вести бухгалтерский и налоговый учёт, готовить и сдавать обязательную отчётность;
* Microsoft Office 2019 – офисный пакет американской корпорации Microsoft для разных операционных систем;
* FreshDoc – онлайн-конструктор документов;
* OC Windows 11 23H2 – семейство проприетарных операционных систем корпорации Microsoft, ориентированных на применение графического интерфейса при управлении;
* Wireshark 4.2.4 – программа для анализа сетевого трафика;
* Google Chrome 124.0.6367.156 – браузер, разработанный компанией Google;
* Double Commander 1.0.11 – двухпанельный файловый менеджер с открытым исходным кодом.

ПО для отдела продаж:

* AmoCRM 2022 – комплекс инструментов для автоматизации бизнес-процессов;
* Jabber 14.3.1 – протокол для мгновенного обмена сообщениями и информацией;
* Microsoft Office 2019 – офисный пакет американской корпорации Microsoft для разных операционных систем;
* OpenSSH 9.7 – набор программ, предоставляющих шифрование сеансов связи по компьютерным сетям с использованием протокола SSH;
* OC Windows 11 23H2 – семейство проприетарных операционных систем корпорации Microsoft, ориентированных на применение графического интерфейса при управлении;
* Novofon – сервис, предоставляющий услуги IP-телефонии;
* Google Chrome 124.0.6367.156 – браузер, разработанный компанией Google;
* Double Commander 1.0.11 – двухпанельный файловый менеджер с открытым исходным кодом.

ПО для отдела локализации:

* Kaspersky 21.17.7.539 – антивирусное программное обеспечение;
* Jabber 14.3.1 – протокол для мгновенного обмена сообщениями и информацией;
* Microsoft Office 2019 – офисный пакет американской корпорации Microsoft для разных операционных систем;
* OpenSSH 9.7 – набор программ, предоставляющих шифрование сеансов связи по компьютерным сетям с использованием протокола SSH;
* OC Windows 11 23H2 – семейство проприетарных операционных систем корпорации Microsoft, ориентированных на применение графического интерфейса при управлении;
* Windows Defender 1000.25992.0.9000 – антивирус компании Microsoft, встроенный по умолчанию в операционные системы Windows;
* Google Chrome 124.0.6367.156 – браузер, разработанный компанией Google;
* Double Commander 1.0.11 – двухпанельный файловый менеджер с открытым исходным кодом.
  1. **Характеристика комплекса задач, задачи и обоснование необходимости автоматизации**
     1. **Выбор комплекса задач автоматизации и характеристика существующих бизнес-процессов**

Основные автоматизированные бизнес-процессы организации:

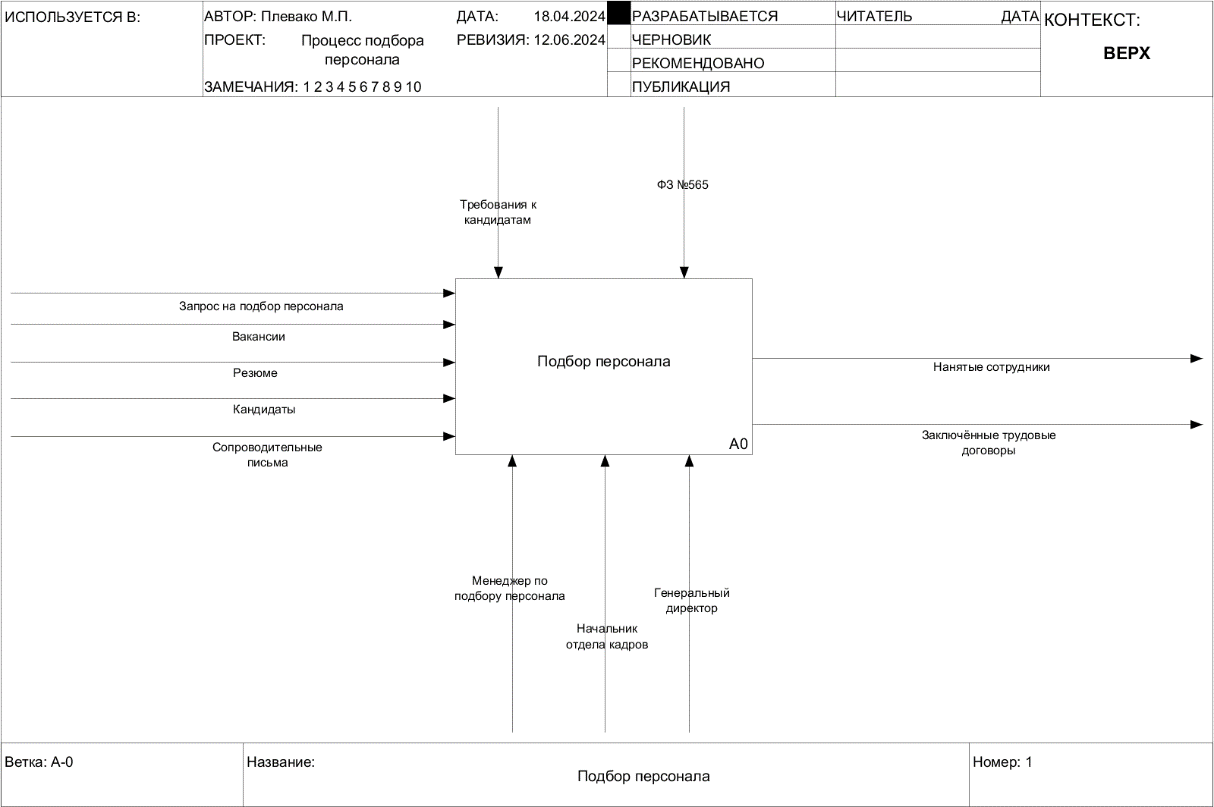
* Финансовое планирование;
* Управление конфигурацией;
* Автоматизация маркетинга;
* Учёт кадров.

Основные неавтоматизированные бизнес-процессы организации:

* Исследование рынка;
* Документооборот;
* Процесс подбора персонала.

Для автоматизации я выбрал бизнес-процесс «Процесс подбора персонала».

Модель деятельности по подбору персонала представлена на рисунке 4.



**Рисунок 4. Контекстная диаграмма подбора персонала AS-IS**

В бизнес-процессе «Подбор персонала» входной информацией являются вакансии, поступающие от кандидатов, откликнувшихся на них. Также на вход подаются требования к кандидатам, которые заранее продумываются и указываются в вакансиях, запрос на подбор персонала, который отправляет генеральный директор, резюме, которое присылает кандидат, сопроводительные письма, которые присылает кандидат вместе с резюме.

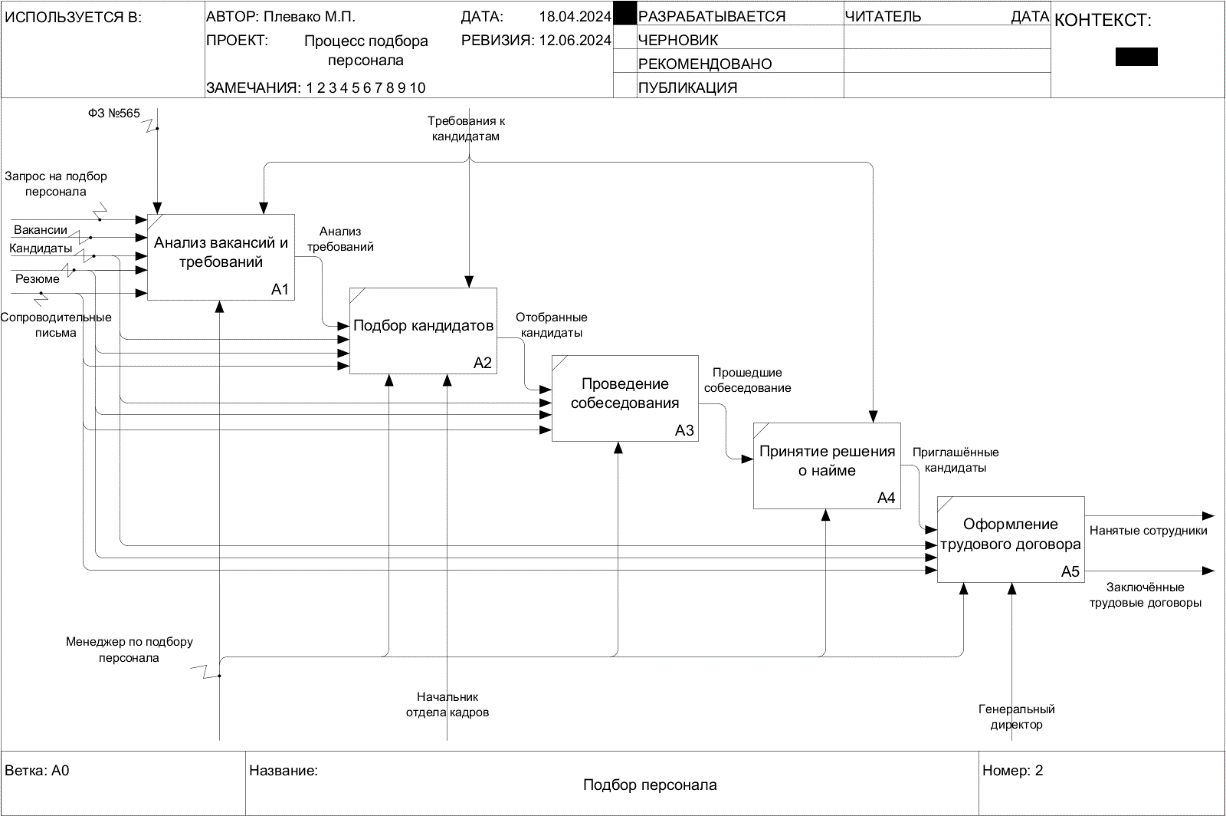
Выходом процесса являются нанятые сотрудники и заключённые трудовые договоры. Нанятым сотрудникам выдаётся трудовой договор и после подписания они приступают к работе. Кандидаты, которые не смогли пройти собеседование вычёркиваются из списка кандидатов и нигде не учитываются.

Следует отметить, что менеджер по подбору персонала присутствует всегда от начала и до конца и проводит собеседования с кандидатами. Начальник отдела кадров, которые участвует в подборе персонала. Генеральный директор, с которым заключается трудовой договор.

Регламентируется бизнес-процесс подбора федеральным законом №565, требованиями к кандидатам. В качестве критерия выделения подпроцессов второго уровня можно использовать пошаговый план действия подбора персонала:

* Анализ вакансий и требований;
* Поиск кандидатов;
* Проведение собеседования;
* Принятие решения о найме;
* Оформление трудового договора.

Декомпозиция IDEF0 модели представлена на рисунке 5.



**Рисунок 5. Декомпозиция процесса подбора персонала AS-IS**

* + 1. **Определение места проектируемой задачи в комплексе задач и ее описание**

При рассмотрении вопроса о выборе бизнес-процесса для автоматизации в этой компании, был сделан вывод о том, что автоматизация процесса подбора персонала является наиболее приоритетным шагом. Вот некоторые причины, аргументирующие мой выбор:

* Эффективность времени и ресурсов. Подбор персонала – это процесс, который требует больших временных и финансовых затрат. Автоматизация этого процесса позволит сократить время, затрачиваемое на рутинные задачи, такие как сортировка резюме и организация собеседований, освобождая ресурсы для более стратегически важных задач.
* Улучшение качества найма. Автоматизированные системы смогут использовать алгоритмы анализа данных для более точной оценки кандидатов и их соответствия требованиям вакансии. Это позволит нам найти наиболее подходящих кандидатов для каждой позиции, что в конечном итоге повысит эффективность работы персонала.
* Улучшение опыта кандидатов. Автоматизированные системы могут предоставлять кандидатам информацию о статусе их заявок и обновлениях в процессе найма. Это создаст позитивное впечатление о компании и поможет привлечь больше квалифицированных кандидатов.
* Безопасность данных. Автоматизированные системы позволят обеспечить высокий уровень безопасности и конфиденциальности персональных данных кандидатов, что соответствует законодательству о защите данных и защищает компанию от рисков, связанных с утечкой информации.

В целом, автоматизация процесса подбора персонала поможет сделать работу «Кранк» более эффективной, оптимизировать использование ресурсов и повысить качество найма. Это сделает компанию более конкурентноспособной и успешной в долгосрочной перспективе.

Модель деятельности по подбору персонала представлена на рисунке 6.



**Рисунок 6. Контекстная диаграмма автоматизированного процесса TO-BE**

В автоматизированном бизнес-процессе «Подбор персонала» входной информацией являются вакансии, поступающие от кандидатов, откликнувшихся на них. Также на вход подаются кандидаты, резюме, сопроводительные письма, шаблон трудового договора и запрос на подбор персонала.

Выходом процесса являются нанятые сотрудники и заключённые трудовые договоры. Нанятым сотрудникам выдаётся трудовой договор и после подписания они приступают к работе. Кандидаты, которые не смогли пройти собеседование вычёркиваются из списка кандидатов и нигде не учитываются.

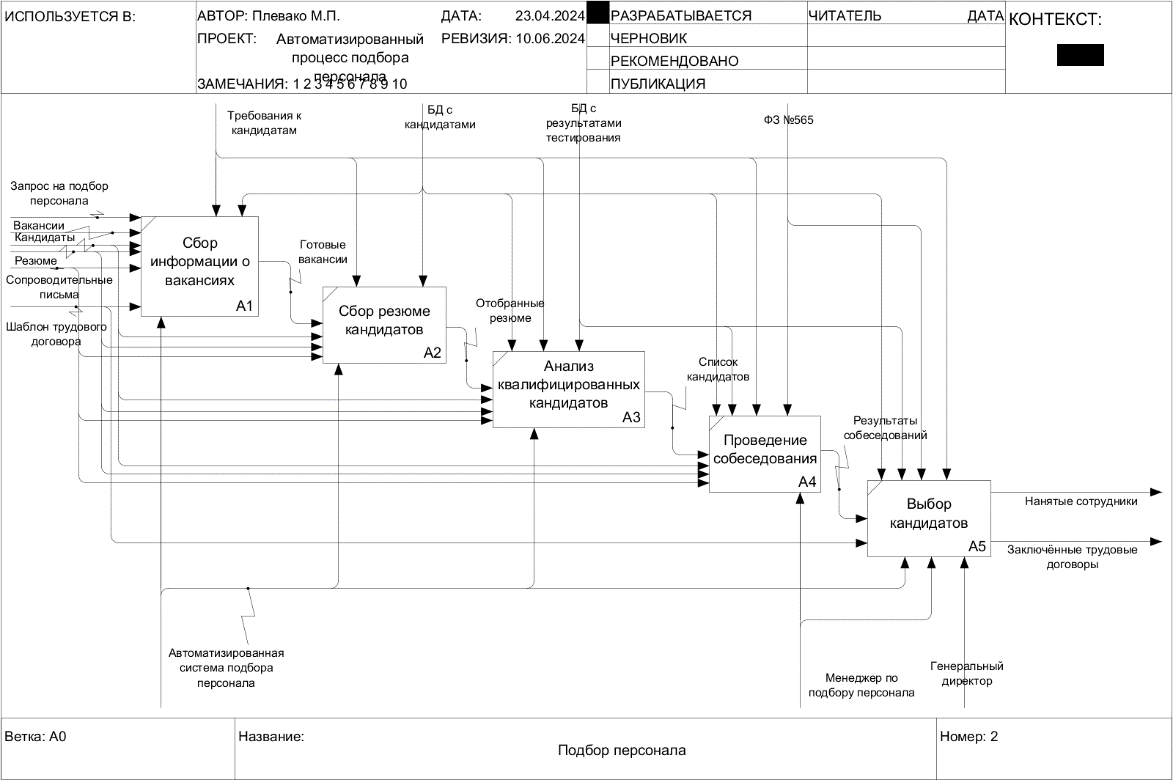
В ресурсах указаны кандидаты, которые подали свои заявки и смогли пройти до собеседования, автоматизированная система подбора персонала, которая необходима для сбора поданных резюме и отбора квалифицированных кандидатов, менеджер по подбору персонала, который проводит собеседование с отобранным кандидатом и генеральный директор.

Регламентируется бизнес-процесс учёта федеральным законом №565, базой данных с информацией о кандидатах, а также базой данных с результатами тестирования кандидатов. В качестве критерия выделения подпроцессов второго уровня можно использовать пошаговый план действия подбора персонала:

* Сбор информации о вакансиях;
* Сбор резюме кандидатов;
* Анализ квалифицированных кандидатов;
* Проведение собеседования;
* Выбор кандидатов.

Оформление трудового договора.

Декомпозиция IDEF0 модели представлена на рисунке 7.



**Рисунок 7. Декомпозиция автоматизированного процесса TO-BE**

Цель разработки информационной системы и автоматизации процесса подбора персонала будет сводиться к устранению существующих недостатков: затраты большого количества времени, плохое качество найма, плохой опыт кандидатов, недостаток безопасности данных кандидатов.

Проектируемая информационная система должна выполнять следующие операции:

* Ввод информации в таблицы базы данных для дальнейшего использования и хранения;
* Просмотр полной информации о кандидатах;
* Поиск информации о кандидах в базе данных по различным параметрам;
* Оформление трудового договора;
* Редактирование и удаление информации в базе данных;

С проектируемой информационной системой должны работать начальник ИТ-отдела, тестировщики, менеджер по подбору персонала. Проектируемая информационная система должна помочь достичь улучшения следующих показателей:

* Качество найма;
* Время, затрачиваемое на подбор персонала;
* Опыт кандидатов;
* Безопасность данных.

Проектируемая информационная система должна решать следующие задачи:

* Анализ вакансий и требований;
* Поиск кандидатов;
* Проведение собеседования;
* Формирование трудового договора;
* Обеспечение оперативного поиска данных по информационной базе.
  + 1. **Анализ информационных потоков проектируемой задачи**

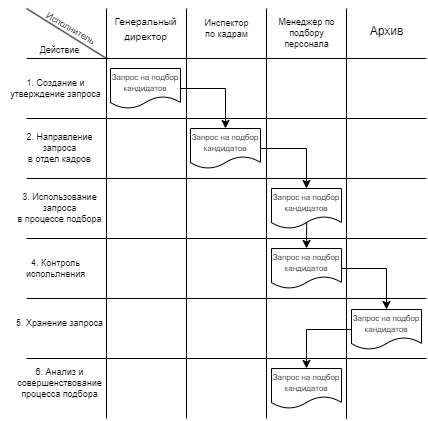
Основные недостатки, присущие существующей практике управления и обработки экономической информации:

* Высокие трудовые затраты на документы: в среднем на каждый документ тратится существенное количество времени. Например, документ о контроле качества требует 50 минут на обработку, что высоко по сравнению с другими документами.
* Неравномерное распределение трудовых затрат: разные типы документов требуют разного количества времени на обработку. Это может создавать неэффективность в процессе управления информацией и использования ресурсов.
* Отсутствие автоматизации: из-за отсутствия автоматизации процессов обработки документов и подбора персонала могут повышаться трудовые затраты и риски человеческих ошибок.
* Недостаточная оптимизация процессов: существует потенциал для оптимизации процессов с целью снижения общего количества времени и увеличения производительности.
* Неоднородность объёмов документов: объёмы различных типов документов значительно отличаются. Это может затруднить планирование и управление ресурсами компании.

Недостатки, устранение которых предполагается осуществить в проекте:

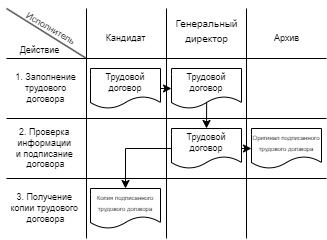
* Ручная обработка кандидатов: в настоящее время подбор персонала осуществляется вручную, что является трудоёмким и неэффективным процессом.
* Долгий срок подбора: текущий процесс подбора требует много времени из-за ручной обработки данных и отсутствия автоматизированных инструментов для фильтрации и оценки кандидатов.
* Недостаточная оценка кандидатов: в ручном режиме упускаются важные параметры оценки кандидатов из-за ограниченных возможностей анализа и сравнения данных.
* Неоднородность критериев оценки: у разных менеджеров по подбору персонала могут быть разные критерии оценки кандидатов, что может привести к несогласованности в процессе подбора.

Схема документооборота документа о запросе на подбор персонала изображена на рисунке 8.



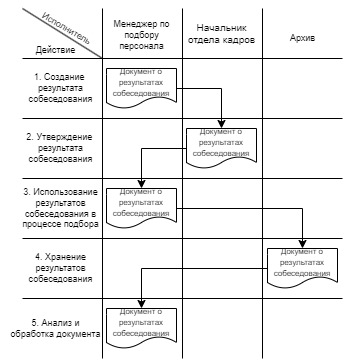
**Рисунок 8. Схема документооборота документа о запросе на подбор кандидатов**

Схема документооборота трудового договора изображена на рисунке 9.



**Рисунок 9. Схема документооборота трудового договора**

Схема документооборота документа о результатах интервью изображена на рисунке 10.



**Рисунок 10. Схема документооборота документа о результатах интервью**

Прагматические оценки потоков информации представлены в таблице 3.

**Таблица 3**

**Оценка потока информации**

| **Документ** | **Количество документов за год** | **Трудовые затраты на 1 документ, мин** | **Трудовые затраты за год, час** |
| --- | --- | --- | --- |
|
|  |
| Документ о запросе на подбор кандидатов | 40 | 30 | 20 |
| Трудовой договор | 40 | 30 | 20 |
| Документ о результатах интервью | 40 | 20 | 13,3 |
| **Итого:** | 120 | 80 | 53,3 |

Исходя из описанных недостатков, осуществлён расчёт планируемого эффекта, который будет достигаться при устранении выявленных недостатков:

* Сокращение времени подбора кандидатов: автоматизация позволит сократить время подбора примерно на 30%. Например, если текущее время подбора составляет 30 дней, то после внедрения автоматизированной системы время сократится до 21 дня.
* Увеличение качества подбора: благодаря автоматизации в компании уменьшится вероятность потери важных параметров кандидатов на 20%.
* Снижение трудовых затрат на подбор персонала: автоматизация позволит сократить трудовые затраты на подбор вплоть до 40%.

В целом, устранение выявленных недостатков позволит компании значительно повысить эффективность и результативность процесса подбора персонала, что приведёт к улучшению работы всей организации и её конкурентоспособности на рынке труда.

* + 1. **Анализ системы обеспечения информационной безопасности и защиты информации**

В компании ООО «Кранк» действует строгая политика безопасности, которая регулирует доступ к информации и обеспечивает защиту данных компании. Нормативно-правовые и организационно-распорядительные документы определяют основные положения политики безопасности.

Регламенты использования сети Internet включают в себя запрет на посещение недостоверных или опасных сайтов, а также на загрузку и использование нелегальных программ или контента. Сотрудникам рекомендуется использовать официальные и безопасные ресурсы в сети и следить за конфиденциальностью информации.

Электронная почта является важным инструментом для общения в компании. Сотрудники обязаны использовать корпоративную почту для официальных целей и следить за безопасность своих аккаунтов. Регламенты доступа к служебной информации определяют правила хранения и передачи конфиденциальных данных, а также требования к защите доступа к ним.

Для обеспечения безопасности коммерческой тайны сотрудникам запрещено раскрывать внутренние процессы и информацию о клиентах без соответствующего разрешения. Регламенты установки и использования программного обеспечения определяют перечень доверенных приложений и инструментов, которые могут быть использованы сотрудниками компании.

Анализ существующих программных средств ИБ и ЗИ представлен в таблице 4.

**Таблица 4**

**Программные средства для защиты ИБ и ЗИ**

| **№ п/п** | **Название ПО** | **Назначение** | **Параметры и возможности** |
| --- | --- | --- | --- |
| 1. | Антивирусное ПО Kaspersky | Предназначен для защиты от вредоносного ПО | Защита в реальном времени |
| 2. | Встроенное антивирусное ПО Microsoft Defender | Предназначен для защиты операционной системы от вредоносного ПО | Облачная защита и антивирусное сканирование |
| 3. | Межсетевой экран Cisco PIX Firewall 515E | Защита от несанкционированного доступа | Фильтрация входящего и исходящего трафика |

Анализ существующих аппаратных средств ИБ и ЗИ представлен в таблице 5.

**Таблица 5**

**Анализ аппаратных средств ИБ и ЗИ**

| **№ п/п** | **Название** | **Назначение** | **Параметры и возможности** |
| --- | --- | --- | --- |
| 1. | Маршрутизатор TP-Link Archer AX50 | Обеспечение доступа в интернет | Поддержка NAT, SPI, DMZ |
| 2. | Маршрутизатор Keenetic Giga | Обеспечение доступа в интернет | Поддержка DHCP |
| 3. | Коммутатор TP-Link TL-SG1048 | Объединение разных устройств в одну локальную сеть | Поддержка стандартов: IEEE 802.3ab, IEEE 802.3i, IEEE 802.3u, IEEE 802.3x |

Порядок реализации системы обеспечения ИБ и ЗИ:

1. Назначение ответственного лица за ИБ и ЗИ. Ответственный – начальник отдела разработки.
2. Разработка политики ИБ и ЗИ. Ответственный – начальник IT-отдела.
3. Аудит безопасности информационной системы. Ответственный – начальник отдела разработки.
4. Разработка и внедрение мер по ИБ и ЗИ. Ответственный – начальник отдела разработки.
5. Проведение обучения сотрудников по вопросам ИБ и ЗИ. Ответственный – начальник отдела разработки.
6. Мониторинг и анализ событий в информационной системе и выявление угроз.
7. Реагирование на инциденты ИБ. Ответственный – начальник отдела разработки.
8. Проведение регулярных тестирований и анализа системы безопасности. Ответственный – начальник отдела разработки.
9. Постоянное совершенствование системы обеспечения ИБ и ЗИ. Ответственный – начальник отдела разработки.

Обеспечение ИБ и ЗИ на разны уровнях:

На программном уровне ИБ и ЗИ обеспечивается с помощью установки антивирусного ПО на все компьютеры сотрудников, использованием средств шифрования данных при передаче и хранении конфиденциальной информации, а также регулярным обновлением всех программных продуктов для предотвращения появления уязвимостей.

На аппаратном уровне ИБ и ЗИ обеспечивается с помощью установки физических барьеров доступа к серверам и другим хранилищам данных, а также использованием системы видеонаблюдения для контроля за доступом к помещениям с информацией.

На организационном уровне ИБ и ЗИ обеспечивается установкой прав доступа к информационным ресурсам в зависимости от роли сотрудника, регулярным обучением сотрудников правилам безопасности информации, использованием сложных паролей для доступа к информационной системе, а также ведением журналов логов для отслеживания действий сотрудников в системе. Кроме того, используются программные средства защиты, такие как брандмауэр и антивирусное ПО, а также встроенные средства защиты в операционных системах и приложениях для обнаружения и предотвращения угроз.

Для защиты от внешних угроз используются следующие средства:

1. Межсетевые экраны – для мониторинга и контроля входящего и исходящего трафика на серверах, чтобы предотвратить несанкционированный доступ к данным;
2. SSL-сертификаты – для обеспечения безопасного соединения между сервером и клиентом;
3. Средства мониторинга угроз – для раннего обнаружения попыток взлома или других внешних атак на систему компании;
4. Регулярные аудиты безопасности – для выявления уязвимостей в сети и их своевременного устранения;
5. Регулярное обновление ПО – для предотвращения использования уязвимостей в более ранних версиях программ;
6. Обучение сотрудников – для повышения осведомлённости о безопасности информации и снижения риска возникновения угроз из-за человеческого фактора;
7. Резервное копирование данных – для предотвращения потери информации в случае взлома или других инцидентов.

Для защиты от инсайдерских угроз используются следующие средства:

1. Мониторинг активности сотрудников при работе с конфиденциальными данными;
2. Регулярное аудирование безопасности информационной системы;
3. Использование двухфакторной аутентификации для доступа к корпоративным ресурсам;
4. Проведение регулярных проверок на предмет наличия вредоносных программ на рабочих устройствах сотрудников;
5. Обеспечение конфиденциальности данных на всех этапах их хранения, передачи и обработки.
   1. **Анализ существующих разработок и выбор стратегии автоматизации «КАК ДОЛЖНО БЫТЬ»**
      1. **Анализ существующих разработок для автоматизации задачи**

На рынке существует много программных решений для автоматизации процесса подбора персонала. Они предлагают разнообразные функциональные возможности, включая анализ резюме, управление кандидатами, а также интеграцию с другими системами управления персоналом.

Одним их популярных программных продуктов в этой области является Applicant Tracking System (ATP), или система управления кандидатами. ATS позволяет автоматизировать весь процесс подбора персонала, начиная с размещения вакансий и заканчивая выбором наилучших кандидатов для конкретной позиции. Программное обеспечение данного типа обычно включает в себя следующие основные характеристики и функциональные возможности:

* Управление вакансиями: создание и публикация вакансий на различных платформах, в том числе на сайте компании, на специализированных порталах и в социальных сетях.
* Анализ резюме: автоматическая обработка и анализ поступающих резюме с помощью алгоритмов машинного обучения и нейронных сетей для выявления наиболее подходящих кандидатов.
* Управление кандидатами: отслеживание статуса каждого кандидата на различных этапах процесса подбора, сохранение истории взаимодействия с кандидатами.
* Коммуникация: возможность взаимодействия с кандидатами через встроенные инструменты электронной почты, чаты или SMS-сообщения.
* Аналитика и отчётность: предоставление статистики и аналитических данных о процессе подбора персонала для оптимизации стратегии найма.

Примером ATS является программное обеспечение Workday Recruiting. Оно предоставляет широкий набор функциональных возможностей, включая управление вакансиями, анализ резюме с использованием искусственного интеллекта, управление кандидатами и сотрудниками, интеграцию с другими HR-системами и инструменты для аналитики и отчётности.

Эти программные средства значительно упрощают и ускоряют процесс подбора персонала, позволяя компаниям эффективно находить наиболее подходящих кандидатов и сокращать время, затрачиваемое на административные процессы.

Проектируемая технология решения задачи подбора персонала должна:

* Технология должна быть способна адаптировать процесс подбора персонала под конкретные потребности кампании;
* Технология должна легко интегрироваться с другими системами управления персоналом;
* Технология должна быть простой и удобной в использовании как для менеджеров по подбору персонала, так и для кандидатов;
* Технология должна предоставлять возможности для анализа данных и составления отчётов, чтобы помочь компании отслеживать эффективность своих процессов подбора персонала.

Существующие программные средства для подбора персонала зачастую не отвечают всем требованиям, написанным выше. Некоторое из них зачастую могут использовать устаревшие технологии и недостаточно интегрированы с другими системами;

Разработка нового программного средства позволит создать решение, которое будет соответствовать конкретным потребностям компании и кандидатов, и предоставит следующие преимущества:

* Повышение эффективности процесса подбора персонала;
* Снижение затрат на подбор персонала;
* Улучшение качества подбора персонала;
* Улучшение взаимодействия с кандидатами.

Новое программное средство для подбора персонала должно отличаться от существующих средств следующими характеристиками:

* Высокий уровень персонализации;
* Глубокая интеграция с другими системами управления персоналом;
* Удобный и интуитивно понятный пользовательский интерфейс;
* Расширенные возможности аналитики и отчётности;

Внедрение этих отличительных особенностей позволит создать информационную систему для подбора персонала, которая будет более эффективной, удобной и аналитически обоснованной, чем существующие решения.

Сравнение найденных разработок с планируемым решением представлено в таблице 6.

**Таблица 6**

**Сравнение разработок с планируемым решением**

| **Критерий/Решение** | **Готовое решение (Workday Recruiting)** | **Разрабатываемая система** |
| --- | --- | --- |
| Стоимость внедрения и поддержки | Возможно высокая, зависит от лицензирования и масштаба | Более низкая стоимость в долгосрочной перспективе |
| Интеграция | Имеет встроенные интеграционные возможности с другими системами | Требует дополнительной работы по интеграции с другими системами компании |
| Производительность алгоритмов анализа данных | Использует стандартные алгоритмы анализа данных, ограниченная точность | Возможность разработки и использования более продвинутых алгоритмов, повышение точности отбора кандидатов |
| Пользовательский опыт | Стандартный пользовательский интерфейс | Может быть настроен на основе потребностей компании, что повышает удобство использования |

Вывод:

Разрабатываемая система предоставляет более гибкие возможности настройки, улучшенные алгоритмы анализа данных и лучшую интеграцию с другими системами, так же имеет более низкую стоимость в долгосрочной перспективе. В любом случае выбор между готовым решением и разрабатываемой системой зависит от конкретных потребностей компании, её бюджета и стратегии развития.

* + 1. **Выбор и обоснование стратегии автоматизации задачи**

Этапы разработки стратегии реализации проекта по автоматизации процесса управления персоналом:

1. Анализ потребностей компании:

Цель: понять требования и ожидания компании от системы автоматизации подбора персонала. То включает в себя определение объёма работы, функциональных требований, бюджета и временных рамок проекта.

2. Оценка рынка решений:

Необходимо изучить доступные на рынке готовые решения, такие как Oracle Talent Management System и Workday Recruiting. Цель – выявить их функциональные возможности, стоимость внедрения и поддержки, а также преимущества и недостатки по сравнению с разрабатываемым решением.

3. Анализ возможностей разработки:

Изучение инструментов разработки, таких как Visual Studio, для создания решения. Цель – оценить гибкость, возможность индивидуальной настройки и сложность разработки подобного решения.

4. Сравнительный анализ готовых решений:

Составление сводной таблицы с учётом специфики проекта и требований компании.

5. Принятие решений:

На основе анализа выбрано разрабатываемое решение для автоматизации процесса подбора персонала. Решение было выбрано, учитывая потребности компании, бюджет проекта и стратегию развития.

6. Разработка и внедрение выбранного решения:

Начало разработки и внедрения выбранного решения для автоматизации процесса подбора персонала. Этот этап включает в себя создание, тестирование и настройку системы под конкретные требования компании.

7. Обучение персонала и поддержка:

Проведение обучения персонала компании работе с новым инструментом после внедрения системы. Обеспечение технической поддержки и обновления системы.

Для данного проекта наиболее подходящей стратегией автоматизации является стратегия по участкам.

Эта стратегия обеспечивает:

* Целенаправленную автоматизацию конкретных участков процесса подбора персонала, что приводит к существенному улучшению эффективности и производительности;
* Управляемое внедрение, снижающее риски и обеспечивающее более плавный переход;
* Гибкость и адаптивность, позволяя масштабировать приложение и автоматизировать дополнительные участки по мере изменения требований;
* Возможность проверки концепции и получения отзывов на ранних этапах, что позволяет внести коррективы и улучшить общее решение;
* Избежание перегрузки, позволяя автоматизировать наиболее критические или трудоёмкие области, не пытаясь автоматизировать весь процесс сразу.

Таким образом, стратегия по участкам наилучшим образом соответствует требованиям и целям проекта по разработке приложений по автоматизации процесса подбора персонала.

* + 1. **Выбор и обоснование способа приобретения ИС для автоматизации задачи**

При выборе информационной системы для автоматизации процесса подбора персонала необходимо учитывать ряд факторов, включая специфику бизнеса, потребности персонала, бюджет проекта сроки внедрения и готовность к изменениям. Далее указаны возможные варианты приобретения информационных систем:

1. Покупка готового решения.

Этот подход подразумевает приобретение готового программного обеспечения от ведущих поставщиков. Такие решения уже прошли проверку на рынке и обычно предлагают широкий спектр функциональности, включая инструменты для рекрутинга, аналитики, управления кандидатами и т.д. Преимущества включают быстрое внедрение и техническую поддержку от поставщика. Однако, такие решения могут быть дорогими, и иногда не удовлетворять уникальным требованиям предприятия.

1. Покупка и доработка.

Этот вариант предполагает приобретение базового готового решения, которое затем настраивается или дорабатывается под специфические потребности предприятия. Это может включать в себя добавление нового функционала, изменение интерфейса или интеграцию с другими системами. Преимущества включают возможность адаптации под уникальные требования и относительно низкие затраты по сравнению с разработкой с нуля. Однако, этот процесс может большое количество времени и ресурсов на доработку и тестирование.

1. Собственная разработка.

В этом случае предприятие принимает на себя полную ответственность за разработку программного обеспечения с нуля. Это предоставляет максимальную гибкость и контроль над функциональностью и архитектурой системы. Однако, разработка собственного решения может быть крайне затратной и требовать значительных временных и финансовых ресурсов. Кроме того, есть риск, что результаты могут не соответствовать ожиданиям или требованиям предприятия.

Был выбран третий вариант и вот некоторые факторы, которые повлияли на выбор:

* Полная настраиваемость: разработка собственного ПО даёт полный контроль над функционалом и архитектурой системы.
* Интеграция с текущими системами: есть возможность интеграции с другими системами без ограничений.
* Долгосрочная перспектива: вложение в собственную разработку в долгосрочной перспективе может быть более выгодным вариантом.
* Уникальность продукта: разработка собственного решения позволит создать уникальный продукт, который может конкурентным преимуществом на рынке.

**Глава 2. Проектная часть**

* 1. **Разработка проекта автоматизации**
     1. **Этапы жизненного цикла проекта автоматизации**

ISO 12207 является идеальным стандартом жизненного цикла информационной системы для автоматизации процесса подбора персонала в ООО «Кранк» по следующим причинам:

* Упор на процессы: ISO 12207 фокусируется на определении и управлении жизненным циклом ИС, включая процессы подбора персонала;
* Гибкость: данный стандарт предоставляет гибкую структуру, которая позволяет адаптировать процесс подбора персонала к конкретным потребностям ООО «Кранк», таким как требования к кандидатам, этапы собеседования и критерии отбора;
* Аналитические методы: ISO 12207 подчёркивает важность аналитических методов, таких как анализ требований и моделирование процесса, которые могут помочь в оптимизации процесса подбора персонала;
* Управление рисками: ISO 12207 включает в себя управление рисками в течение всего жизненного цикла ИС, помогая компании выявлять и устранять риски, связанные с автоматизируемым процессом, такие как предвзятость или неэффективность;
* Непрерывное улучшение: ISO 12207 делает упор на непрерывном улучшении, обеспечивая основу для регулярного пересмотра и обновления автоматизируемого процесса, чтобы он оставался эффективным и соответствовал меняющимся потребностям бизнеса.

Выбор стандарта ISO 12207 в процессе подбора персонала для ООО «Кранк» позволит компании:

1. Стандартизировать и систематизировать процесс подбора персонала;
2. Повысить согласованность и качество принимаемых на работу кандидатов;
3. Сократить время и затраты на подбор персонала;
4. Улучшить процессы принятия решений и повысить удовлетворённость сотрудников.

Стандарт ISO 12207 был выбран потому, что он предоставляет удобный инструмент для управления качеством и рисками, а также помогает создать систему, которая отвечает всем современным стандартам и требованиям.

Описание всех возможных моделей жизненного цикла:

1. Водопадная модель:

* Описание: линейная и последовательная модель, где этапы следуют друг за другом и возращение на предыдущие этапы не предусмотрено;
* Этапы: анализ требований, проектирование, реализация, тестирование, развёртывание, сопровождение;
* Преимущества: простота, понятность, чёткое разделение обязанностей;
* Недостатки: негибкая, сложность внесения изменений, высокие риски при обнаружении ошибок на поздних этапах.

1. Итеративная модель:

* Описание: итеративное и инкрементальное развитие, где функциональность добавляется и тестируется поэтапно, что позволяет получать обратную связь и вносить корректировки на ранних стадиях;
* Этапы: планирование, анализ, проектирование, разработка, тестирование (повторяется для каждого итерационного цикла);
* Преимущества: гибкость, возможность адаптации к меняющимся требованиям, снижение рисков;
* Недостатки: может привести к увеличению времени разработки, требует тщательного планирования итераций.

1. Спиральная модель:

* Описание: комбинация водопадной и итеративной моделей, где каждый цикл разработки включает планирование, анализ рисков, разработку и оценку;
* Этапы: повторяющиеся циклы, включающие определение целей, оценку альтернатив, выполнение разработки и планирование следующего цикла;
* Преимущества: гибкость, управление рисками, вовлечение заинтересованных сторон на каждом этапе;
* Недостатки: может привести к увеличению времени и затрат на разработку, требует опытной команды.

Для данного проекта была выбрана итеративная модель жизненного цикла, так как она обеспечивает гибкость, раннюю обратную связь, управление рисками и хорошее внедрение.

Краткое рассмотрение ключевых положений по каждому из этапов:

1. Планирование:

* Цель: определить область действия проекта, установить цели и задачи, создать план проекта;
* Ключевые участники: менеджеры проектов;
* Требования к входной информации: бизнес-требования, ограничения, риски;
* Получаемые результаты: план проекта, требования к ПО, архитектура высокого уровня.

1. Анализ:

* Цель: проанализировать требования к ПО, рынок с похожими ПО и выяснить экономическую эффективность;
* Ключевые участники: маркетологи, менеджеры проектов, финансовые аналитики;
* Требования к входной информации: требования к ПО, план проекта;
* Получаемые результаты: спецификации требований, модели проектирования, расчёт экономической эффективности проекта.

1. Проектирование:

* Цель: Создание детального плана и спецификаций системы;
* Ключевые участники: менеджеры проектов, технические писатели, менеджеры по подбору персонала;
* Требования к входной информации: спецификации требований, модели проектирования;
* Получаемые результаты: план системы, спецификации системы.

1. Реализация:

* Цель: разработать и закодировать ПО в соответствии с проектными документами;
* Ключевые участники: программисты, тестировщики;
* Требования к входной информации: План и спецификации системы;
* Получаемые результаты: исходный код, модули ПО.

1. Тестирование:

* Цель: проверить, соответствует ли разработанное ПО требованиям и корректно ли оно работает;
* Ключевые участники: программисты, тестировщики;
* Требования к входной информации: исходный код, планы тестирования;
* Получаемые результаты: отчёты о тестировании, исправленный исходный код.

1. Внедрение:

* Цель: установить и настроить ПО в производственной среде;

Ключевые задачи на этапе эксплуатации и сопровождения:

1. Внедрение разрабатываемого решения в производительную среду. Перенос законченных компонентов системы или всего решения из среды разработки и промежуточной среды в операционную систему для фактического использования конечными пользователями;
2. Проверка работоспособности решения. Обеспечение корректной работы системы в производственной среде, включая функциональность, производительность, совместимость и безопасность;
3. Минимизация рисков. Управление и устранение потенциальных рисков, связанных с запуском нового решения, чтобы обеспечить её бесперебойную работу;
4. Обеспечение непрерывности бизнеса. Планирование и реализация процесса внедрения, который сводит к минимуму сбои в рабочем процессе и обеспечивает непрерывное обслуживание клиентов;
5. Обучение конечных пользователей. Предоставление инструкций и обучения пользователям, чтобы они смогли эффективно использовать новое решение и достигать желаемых результатов;
6. Мониторинг и поддержка. Установление механизмов для непрерывного мониторинга и поддержки решения после его внедрения, чтобы обеспечить его работу в соответствии с ожиданиями.

* Ключевые участники: начальник отдела разработки – руководит процессом разработки и внедрения ПО, пользователи – участвует в тестировании и приёмке системы;
* Требования к входной информации: прошедшее тестирование ПО, планы развёртывания;
* Получаемые результаты: развёрнутое ПО, рабочая система;
* Стратегия внедрения: для внедрения была выбрана стратегия «пилотный проект». Пункты, которые повлияли на выбор:
  + Выявление проблем. Возможность выявления потенциальных проблем на ранних стадиях разработки;
  + Настройка конфигурации системы. Возможность настройки параметров и системы на основе реальных данных и отзывов;
  + Проверка работоспособности системы. Возможность проверки правильной работы функций системы.

1. Эксплуатация и сопровождение:

* Цель: поддерживать развёрнутую систему и обеспечивать непрерывность её работы с минимальными перебоями, чтобы удовлетворять текущим и будущим потребностям бизнеса;

Ключевые задачи на этапе эксплуатации и сопровождения:

1. Мониторинг и устранение неполадок. Регулярно отслеживать производительность системы, выявлять и устранять проблемы, прежде чем они повлияют на пользователей и их опыт;
2. Управление изменениями. Контролировать и управлять изменениями в системе для обеспечения стабильности и соответствия бизнес-требованиям;
3. Обновления и исправления. Применять обновления программного обеспечения и исправления безопасности для поддержания актуальности системы и устранения известных уязвимостей;
4. Оптимизация производительности. Анализировать и настраивать систему для улучшения производительности, пропускной способности и времени безотказной работы;
5. Управление безопасностью. Соблюдать лучшие практики безопасности, включая контроль доступа, шифрование данных и регулярные проверки безопасности;
6. Резервное копирование и восстановление. Регулярно выполнять резервное копирование системы и данных для обеспечения возможности восстановления в случае непредвиденных обстоятельств;
7. Поддержка пользователей. Предоставлять техническую поддержку и документацию для пользователей системы, чтобы обеспечить бесперебойную работу;
8. Управление жизненным циклом. Планировать и управлять этапами жизненного цикла системы, включая модернизацию, переход и вывод из эксплуатации;
9. Сбор и анализ данных. Собирать и анализировать данные о производительности и использовании системы для выявления областей для улучшения и принятия обоснованных решений.

* Ключевые участники: начальник отдела разработки – координирует работу всех участников, пользователи – сообщают об ошибках и запросах на новые функции, программисты – вносят исправления в систему и реализуют новые функции, тестировщики – проводят функциональное и регрессионное тестирование системы, художники – разрабатывают интерфейс и графические элементы системы, дизайнеры – обеспечивают согласованность и эстетическую привлекательность системы;
* Требования к входной информации: развёрнутое ПО, отчёты об ошибках;
* Получаемые результаты: обновлённое ПО, улучшения, исправления ошибок.
  + 1. **Ожидаемые риски на этапах жизненного цикла и их описание**

Возможные риски на каждом этапе жизненного цикла:

1. Планирование:
   1. Риски, которые связаны с требованиями: неясные, неполные или изменчивые требования;

Шаги для уменьшения риска:

* Необходимо тщательно определять требования с участием заинтересованных сторон;
* Проводить регулярные проверки требований с пользователями.
  1. Риски, которые связаны с ресурсами: нехватка персонала, недостаточная квалификация или высокая текучка кадров;

Шаги для уменьшения риска:

* Тщательно планировать ресурсы, в том числе персонал, оборудование и бюджет;
* Привлекать квалифицированный персонал и создавать благоприятную рабочую среду.
  1. Риски, которые связаны со сроками: нереалистичные сроки проекта;

Шаги для уменьшения риска:

* Устанавливать реалистичные сроки проекта и учитывать возможные задержки;
* Регулярно отслеживать процесс и при необходимости принимать корректирующие меры.
  1. Риски, которые связаны с бюджетом: недостаточное финансирование или перерасход средств.

Шаги для уменьшения риска:

* Тщательно планировать бюджет и контролировать расходы;
* Искать возможности для оптимизации затрат.

1. Анализ:
   1. Риски, которые связаны с требованиями: неправильное понимание или интерпретация требований;

Шаги для уменьшения риска:

* Убедиться, что требования полностью понятны и правильно интерпретированы;
* Использовать техники прототипирования и моделирования для проверки и уточнения требований.
  1. Риски, которые связаны с технологиями: выбор неподходящих или несовместимых технологий.

Шаги для уменьшения риска:

* Тщательно оценивать технологические риски и выбирать проверенные технологии;
* При необходимости проводить пилотные проекты или использовать сторонних консультантов.

1. Проектирование:
   1. Риски, которые связаны с требованиями: ошибки или упущения в проектировании, вызванные неполными или неточными требованиями;

Шаги для уменьшения риска:

* Проводить тщательный анализ и проектирование на основе чётких требований;
* Проводить проверки и инспекции на промежуточных этапах проектирования.
  1. Риски, которые связаны с технологиями: сложность или недоработка используемых технологий.

Шаги для уменьшения риска:

* Выбирать технологии, соответствующие функциональным и не функциональным требованиям;
* Необходимо тщательно оценивать риски, связанные с использованием новых или сложных технологий.

1. Реализация:
   1. Риски, которые связаны с ресурсами: нехватка квалифицированных разработчиков или перегрузка разработчиков;

Шаги для уменьшения риска:

* Обеспечивать достаточное количество квалифицированных разработчиков и рабочих станций;
* Управлять загруженностью разработчиков и избегать перегрузок.
  1. Риски, которые связаны со сроками: задержки в разработке, вызванные техническими проблемами или нехваткой ресурсов.

Шаги для уменьшения риска:

* Устанавливать реалистичные сроки для задач разработки;
* Следовать чёткому плану разработки с точками контроля.

1. Тестирование:
   1. Риски, которые связаны с тестированием: недостаточное или неэффективное тестирование, что может привести к тому, что ошибки останутся незамеченными;

Шаги для уменьшения риска:

* Разрабатывать и выполнять всесторонний план тестирования, охватывающий все аспекты системы;
* Использовать автоматизированное тестирование для повышения эффективности и покрытия.
  1. Риски, которые связаны со сроками: задержки в тестировании, вызванные сложностью системы или нехваткой ресурсов.

Шаги для уменьшения риска:

* Планировать достаточное время для тестирования и учитывать возможные задержки;
* При необходимости привлекать дополнительных тестировщиков.

1. Развёртывание:
   1. Риски, которые связаны с эксплуатацией: недостаточная подготовка пользователей или ошибки при установке и настройке системы.

Шаги для уменьшения риска:

* Обучать пользователей и администраторов системы;
* Создавать чёткие инструкции по установке и настройке.

1. Эксплуатация и сопровождение:
   1. Риски, которые связаны с эксплуатацией: проблемы с производительностью, доступностью или безопасностью системы;

Шаги для уменьшения риска:

* Регулярно отслеживать производительность, доступность и безопасность системы;
* Быстро реагировать на проблемы и принимать меры по их устранению.
  1. Риски, которые связаны с сопровождением: сложность или затратность внесения изменений в систему по мере изменения требований и технологий.

Шаги для уменьшения риска:

* Управлять изменениями в системе с помощью формального процесса управления конфигурацией;
* Использовать модульное проектирование и документирование для упрощения внесения изменений.

Общие актуальные риски для разрабатываемого проекта:

1. Риски, которые связаны с интеграцией с существующими системами: неправильное взаимодействие с существующими системами;

Шаги для уменьшения риска:

* Проводить тщательный анализ и планирование интеграции;
* Создавать подробную документацию по интерфейсам и протоколам интегрирования.

1. Риски, которые связаны с безопасностью и защитой данных: ненадёжная обработка конфиденциальных и критически важных данных;

Шаги для уменьшения риска:

* Проводить анализ рисков безопасности и разрабатывать соответствующие меры контроля;
* Использовать надёжные методы аутентификации и шифрования.

1. Риски, которые связаны с масштабируемостью и производительность: неспособность обрабатывать большой объём данных и плохая поддержка большого количества пользователей;

Шаги для уменьшения риска:

* Проводить тестирование производительность и нагрузочное тестирование для выявления и устранения уязвимых мест;
* Выбирать масштабируемую архитектуру и технологии.

1. Риски, которые связаны с пользовательским опытом: система сложна в использовании, неудобна или не соответствует ожиданием пользователей;

Шаги для уменьшения риска:

* Проводить исследования пользователей и собирать обратную связь;
* Проводить юзабилити-тестирование для оценки и улучшения пользовательского опыта.

1. Риски, которые связаны с юридическими и нормативными требованиями: несоответствие законам, нормам и стандартам.

Шаги для уменьшения риска:

* Проводить юридический анализ и консультации для обеспечения соответствия всем применимым законам, нормам и стандартам;
* Документировать принятые меры по обеспечению соответствия и регулярно их пересматривать.
  1. **Информационное обеспечение задачи**
     1. **Характеристика нормативно-справочной, входной и оперативной информации**

В проектируемой системе документом является резюме кандидата, которое он заполняет сам и предоставляет свои данные (ФИО, фото, телефон, серия и номер паспорта, возраст, оценка за тестирование, желаемая должность). Так же в системе присутствует документ о запросе на подбор персонала, который поступает от генерального директора. Более подробная информация о резюме кандидата и запросе на подбор персонала представлена в таблице 7.

**Таблица 7**

**Входные документы в системе**

| **Название документа** | **Первичные показатели** | **Источник получения документа** | **Структура документа** | **Число строк** | **Объёмные данные** | **Частота возникновения** |
| --- | --- | --- | --- | --- | --- | --- |
| Резюме кандидата | ФИО, фото, почта, Телефон, Серия и номер паспорта, возраст, оценка за тестирование, желаемая должность, опыт работы, образование, пройденные курсы, навыки, о себе | Сайты, на которых выложены вакансии | Документ состоит из информации о кандидате | 10 строки | Данные в виде 10 строк | Каждый раз, когда человек откликается на вакансию и заполняет резюме |
| Запрос на подбор персонала | Дата составления запроса, срок закрытия запроса, наименование вакансии, наименование подразделения, обязанности, причина открытия вакансии | Генеральный директор | Документ состоит из информации о вакансиях и сроках запроса | 5 строк | Данные в виде 5 строк | Каждый раз при нехватке сотрудников в отделах |

В проектируемую систему входит файл с данными, который выступает в роле трудового договора, который заключается между кандидатом и работодателем. Более подробная информация о файле шаблона трудового договора представлена в таблице 8.

**Таблица 8**

**Входные файлы в системе**

| **Название файла** | **Первичные показатели** | **Источник получения файла** | **Структура файла** | **Объёмные данные** | **Частота поступления файла** |
| --- | --- | --- | --- | --- | --- |
| Шаблон трудового договора | Номер договора, дата заключения договора, город заключения договора, кандидат, серия и номер паспорта, сотрудник, отдел, должность, дата и время начала работы | Печать трудового договора с выбранными данными в системе | Подготовленный текст с полями для добавления данных | Строки с текстовыми данными и полями, в которые подставляются выбранные данные | Каждый раз при заключении трудового договора |

В проектируемой системе для хранения информации на разных формах будут использоваться разные справочники. Справочники, которые имеются на определённых формах:

* Форма «Решение о собеседовании»:
  + Кандидат – справочник, в котором указан тот кандидат, который был выбран на предыдущей форме;
  + Тип собеседования – справочник с доступными типами собеседования;
  + Сотрудник – справочник с сотрудниками, которые могут принимать собеседования;
  + Статус собеседования – справочник, в котором можно выбрать необходимый статус собеседования.
* Форма «Заключение трудового договора»:
  + Кандидат – справочник, в котором указаны кандидаты, которые прошли собеседование;
  + Сотрудник – справочник с работодателями;
  + Отдел – справочник с отделами, которые есть в компании;
  + Должность – справочник с должностями, которые есть в компании.

Более подробная информация о справочниках находится в таблице 9.

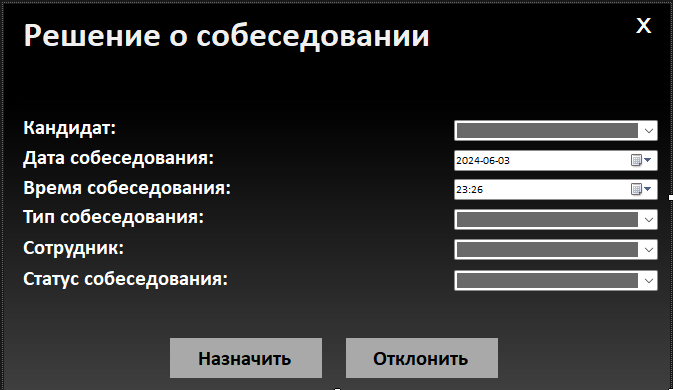
**Таблица 9**

**Подробная информация о справочниках**

| **Название справочника** | **Ведение осуществляет** | **Средний объём справочника в записях** | **Средняя частота обновления** | **Средний объём обновления** |
| --- | --- | --- | --- | --- |
| Кандидат | Менеджеры по подбору персонала | 1 | 3 раза в неделю | 3 записи |
| Тип собеседования | Менеджеры по подбору персонала | 3 | - | - |
| Сотрудник | Менеджеры по подбору персонала | 2 | 1 раз в месяц | 2 записи |
| Статус собеседования | Менеджеры по подбору персонала | 2 | - | - |

Реквизитные составы справочников представлены в таблицах 19, 15, 11, 16.

Заполнение формы «Решение о собеседовании» представлено на рисунке 11.



**Рисунок 11. Заполнение формы «Решение о собеседовании»**

Более подробная информация о справочниках находится в таблице 10.

**Таблица 10**

**Подробная информация о справочниках**

| **Название справочника** | **Ведение осуществляет** | **Средний объём справочника в записях** | **Средняя частота обновления** | **Средний объём обновления** |
| --- | --- | --- | --- | --- |
| Кандидат | Менеджеры по подбору персонала | 10 | 2 раза в неделю | 2 записи |
| Сотрудник | Менеджеры по подбору персонала | 1 | 1 раз в несколько лет | 1 запись |
| Отдел | Менеджеры по подбору персонала | 7 | 1 раз в несколько лет | 1 запись |
| Должность | Менеджеры по подбору персонала | 20 | 1 раз в несколько лет | 1 запись |

Реквизитные составы справочников представлены в таблицах 19, 11, 18, 13.

* + 1. **Характеристика результатной информации**

Описание таблицы «Собеседования». В этой таблице имеются 4 внешних ключа:

* Код\_Кандидата, который берётся из таблицы «Кандидаты» и вместо кода подставляется ФИО кандидата;
* Код\_Типа-собеседования, который берётся из таблицы «Тип\_Собеседования» и вместо кода подставляется название типа собеседования;
* Код\_Сотрудника, который берётся из таблицы «Сотрудники» и вместо кода подставляется ФИО сотрудника;
* Код\_Статуса, который берётся из таблицы «Статус\_Собеседования» и вместо кода подставляется название статуса собеседования.

Так же в этой таблице присутствуют поля «Код\_Собеседования», который подставляется автоматически, «Дата\_Собеседования» и

«Время\_Собеседования».

В приложении эта таблица разделена на 3 формы:

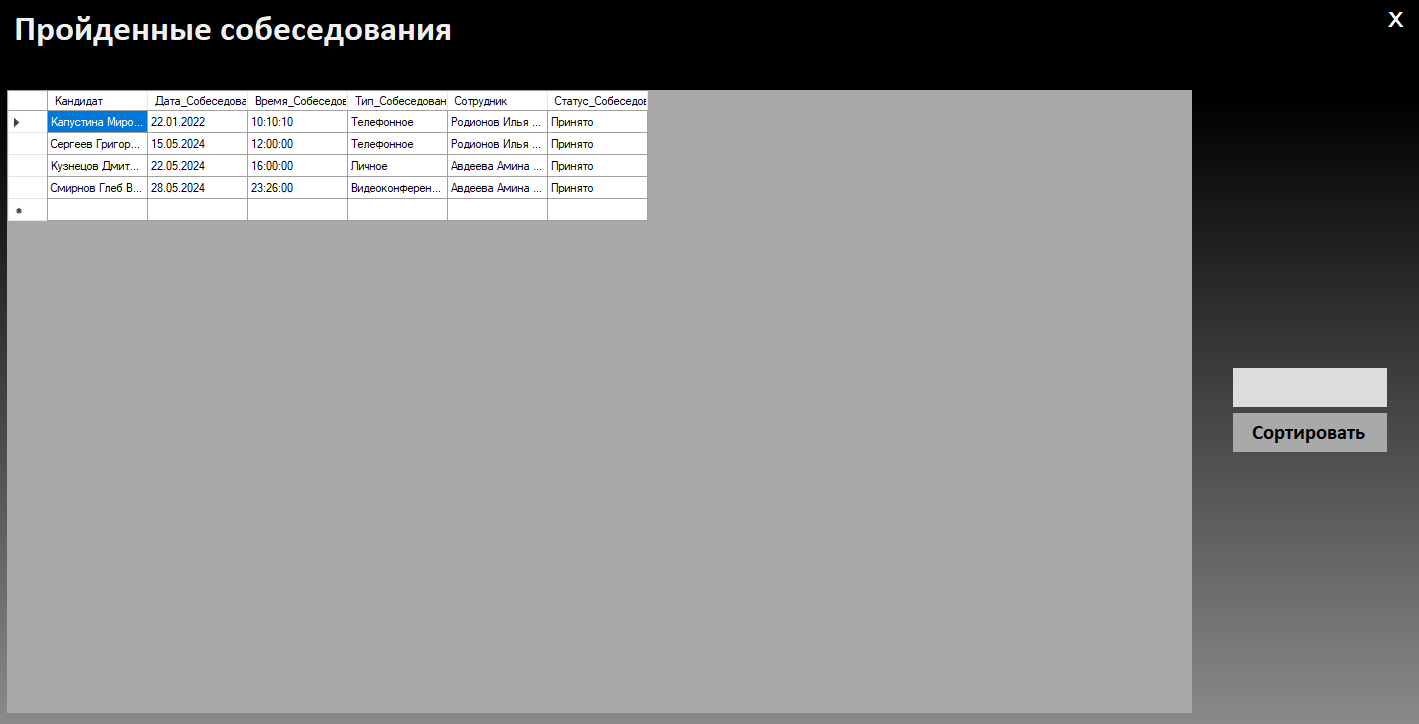
* Пройденные собеседования, в которой показываются только собеседования с кодом статуса 1;
* Назначенные собеседования, в который показываются только собеседования с кодом статуса 3;
* Отклонённые собеседования, в которой показываются только собеседования с кодом статуса 2.

Во всех этих формах присутствует разный функционал:

* С кандидатами, которые прошли собеседования можно заключить трудовой договор с последующей его печатью;
* С кандидатами, с которыми назначено собеседование можно либо принять собеседование, либо отклонить;
* Отклонённые собеседования можно удалить, чтобы освободить место в базе данных.

Код для подключения к базе данных и выборки данных для формы «Пройденные собеседования» представлен в приложении 1.

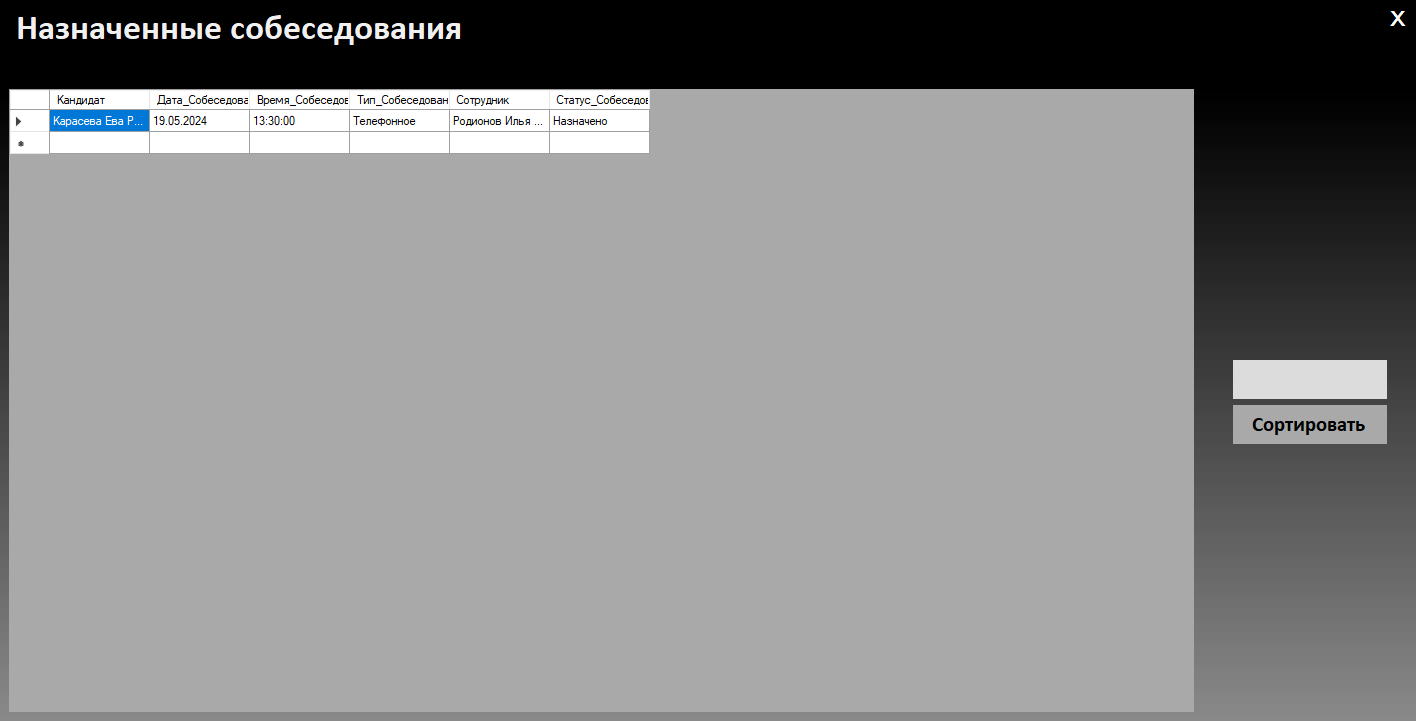
Внешний вид формы «Пройденные собеседования» представлен на рисунке 12.



**Рисунок 12. Форма "Пройденные собеседования"**

Код для подключения к базе данных и выборки данных для формы «Назначенные собеседования» представлен в приложении 2.

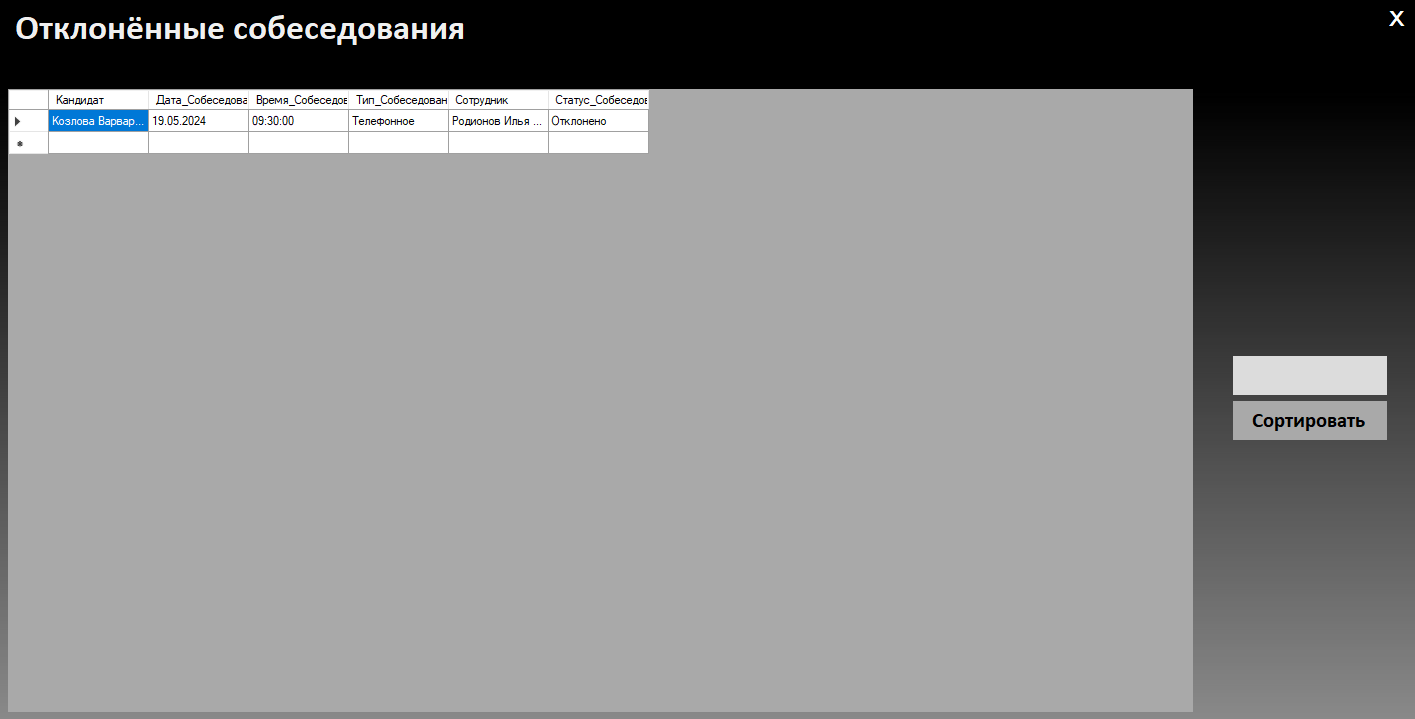
Внешний вид формы «Назначенные собеседования» представлен на рисунке 13.



**Рисунок 13. Форма "Назначенные собеседования"**

Код для подключения к базе данных и выборки данных для формы «Отклонённые собеседования» представлен в приложении 3.

Внешний вид формы «Отклонённые собеседования» представлен на рисунке 14.



**Рисунок 14. Форма "Отклонённые собеседования"**

Описание таблицы «Трудовой\_договор». В этой таблице есть 4 внешних ключа:

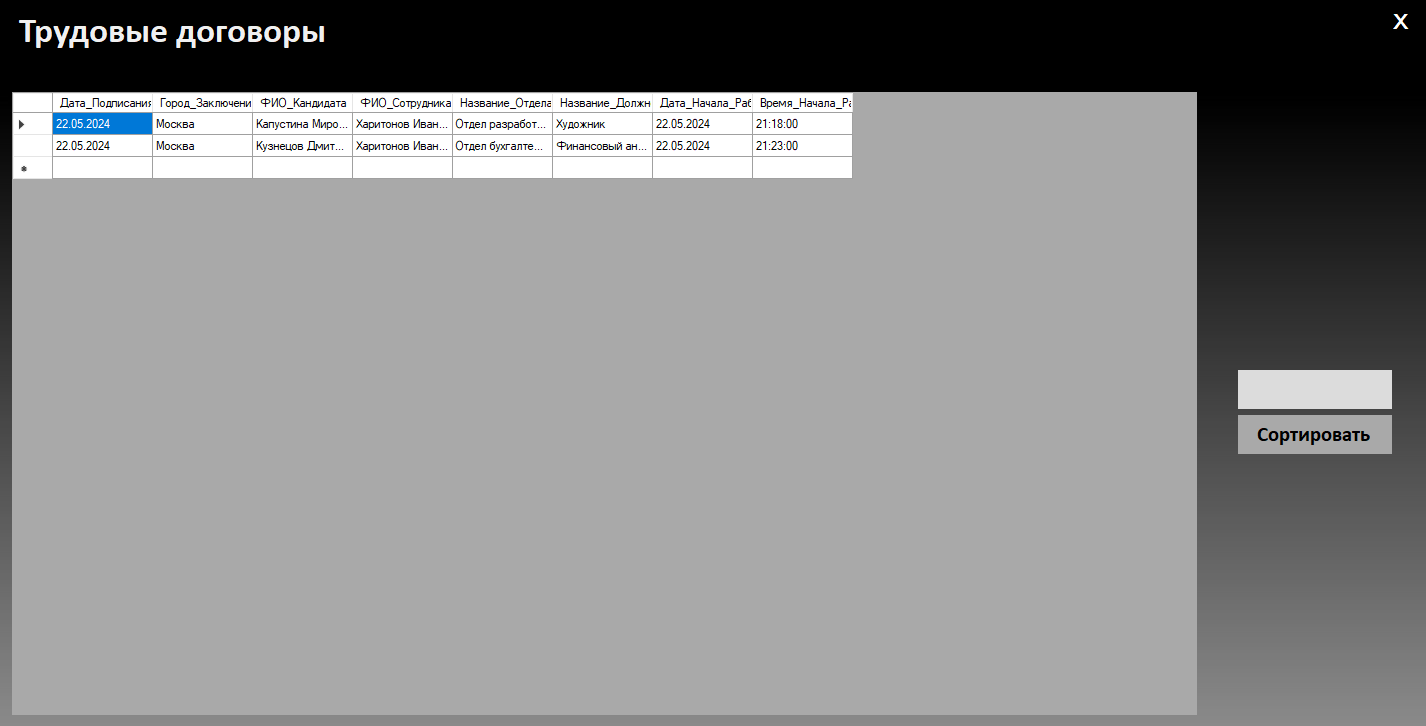
* Код\_Кандидата, который берётся из таблицы «Кандидаты» и вместо кода подставляется ФИО кандидата;
* Код\_Сотрудника, который берётся из таблицы «Сотрудники» и вместо кода подставляет ФИО сотрудника;
* Код\_Отдела, который берётся из таблицы «Отдел» и вместо кода подставляет название отдела;
* Код\_Должности, который берётся из таблицы «Должность» и вместо кода подставляет название должности;

Так же в этой таблице присутствуют поля «Код\_Трудового\_Договора», который подставляется автоматически, «Дата\_Подписания\_Договора», «Город\_Заключения\_Договора», «Дата\_Начала\_Работы», «Время\_Начала\_Работы».

В приложении эта таблица расположена на отдельной форме, в которой можно сортировать данные из таблицы и печатать трудовой договор с нужной информацией. Трудовой договор имеет формат «.docx» и имеет поля для подстановки нужных данных из приложения.

Код для подключения к базе данных и вывода информации в элемент на форме представлен в приложении 4.

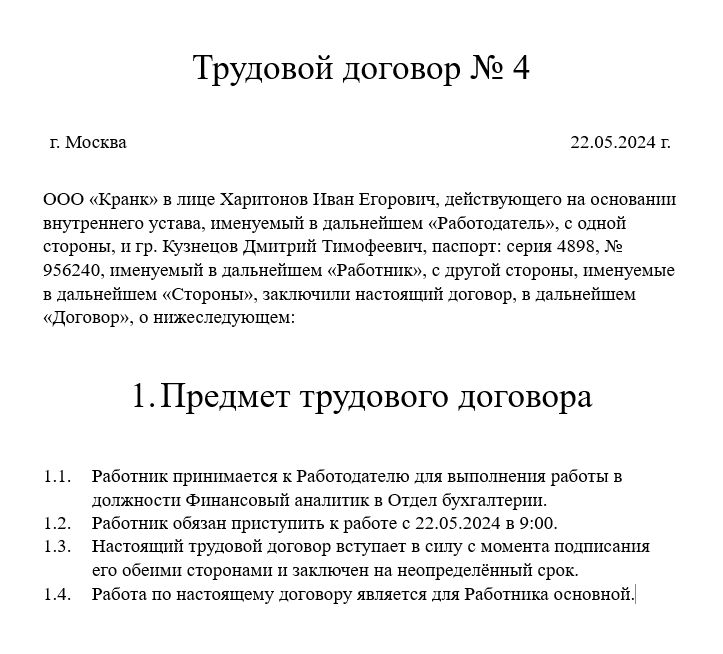
Внешний вид формы «Трудовые договоры» представлен на рисунке 15.



**Рисунок 15. Форма "Трудовые договоры"**

Код для печати трудового договора представлен в приложении 5.

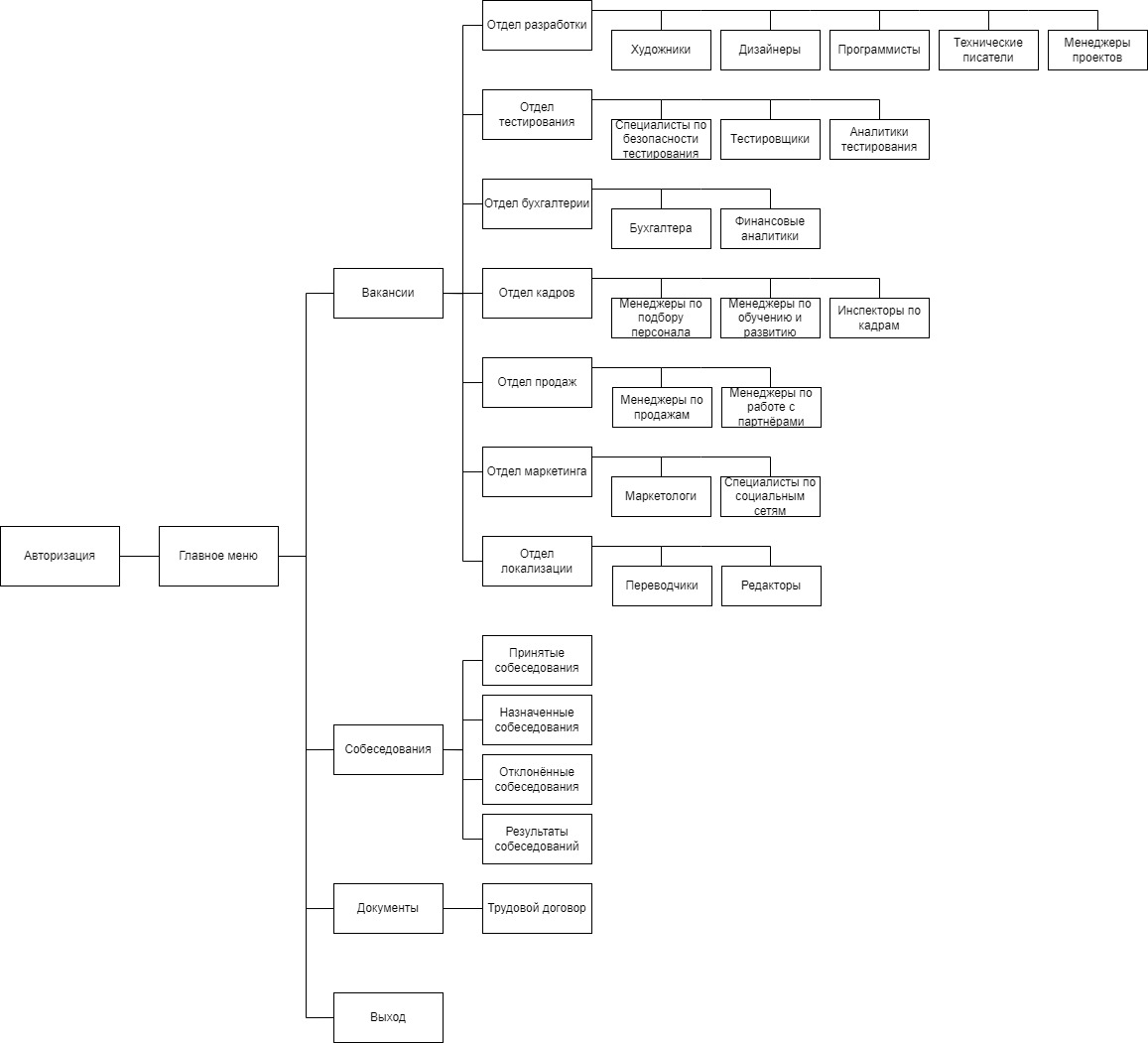
Файл трудового договора с заполненной информацией представлен на рисунке 16.



**Рисунок 16. Заполненный трудовой договор**

* 1. **Программное обеспечение задачи**
     1. **Сценарий диалога**

Сценарий диалога информационной системы представлен на рисунке 17.



**Рисунок 17. Сценарий диалога**

Авторизация – это процесс проверки личности пользователя в информационной системе. Она осуществляется введения пользователем логина и пароля. Если логин и пароль оказались верными, то пользователь попадает в главное меню, в противном случае высветится ошибка авторизации.

Главное меню. В главном меню есть несколько вкладок, при нажатии на которые открываются соответствующие списки.

Вакансии. При наведении на вкладку «Вакансии», открывается список всех отделов в организации, а при наведении на отделы, открываются соответствующие должности.

Должности. При нажатии на любую должность откроется форма с кандидатами, которые подали заявку на эту должность.

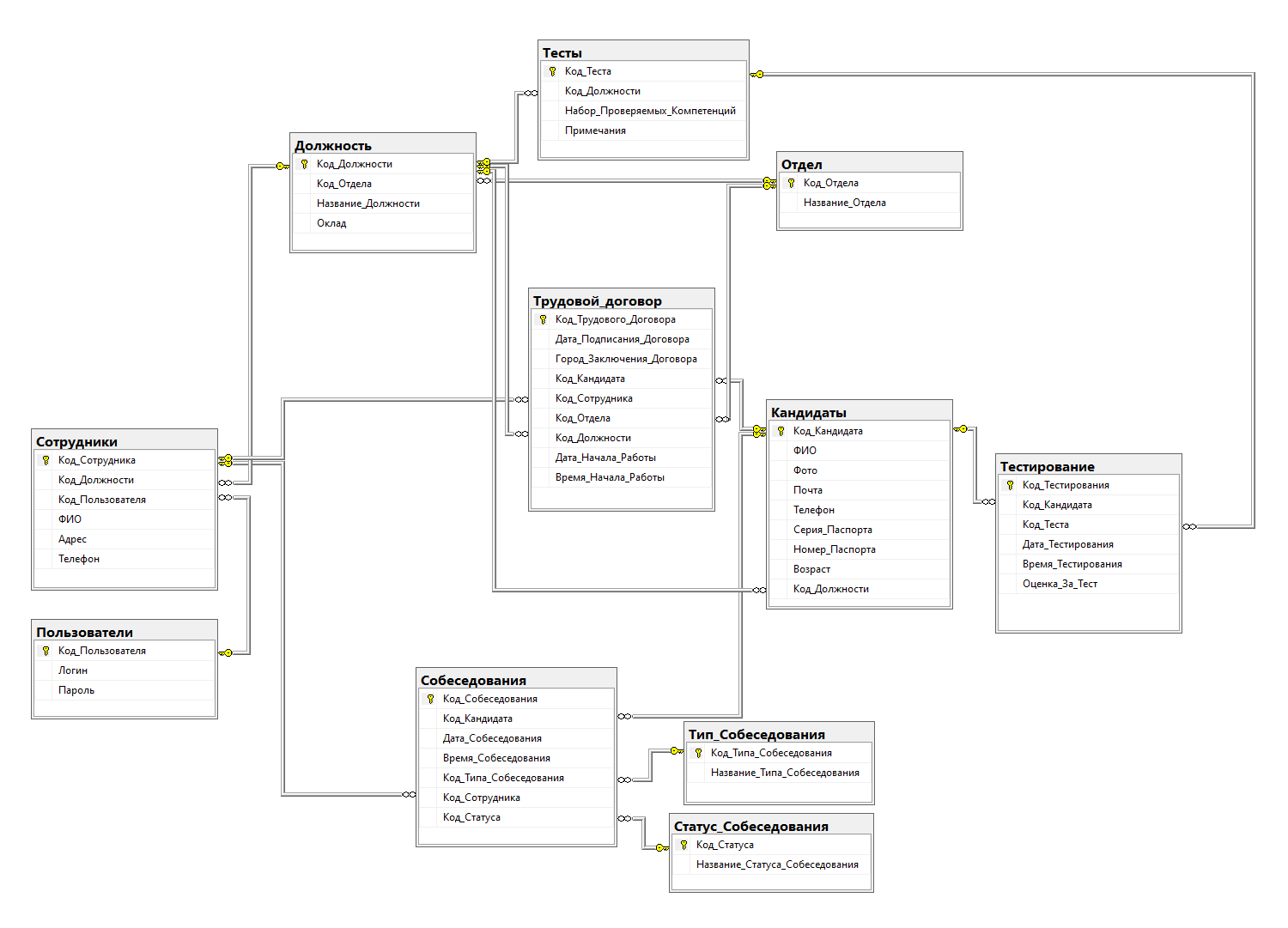
Собеседования. При наведении в главном меню на вкладку «Собеседования», откроется список с принятыми, назначенными и отклонёнными собеседованиями, а также результаты собеседований, при нажатии на которые будут высвечиваться нужные собеседования и информация о них.

Документы. При наведении в главном меню на вкладку «Документы», откроется список с документами, в которых есть трудовой договор.

Трудовой договор. При нажатии на эту вкладку, откроется форма, в которой можно посмотреть заключённые трудовые договоры и распечатать их.

* + 1. **Характеристика базы данных**

ER-диаграмма разработанной базы данных представлена на рисунке 18.



**Рисунок 18. ER-диаграмма базы данных**

Описание таблицы «Сотрудники» из ER-диаграммы представлено в приложении 6.

Описание таблицы «Пользователи» из ER-диаграммы представлено в приложении 7.

Описание таблицы «Должность» из ER-диаграммы представлено в приложении 8.

Описание таблицы «Собеседования» из ER-диаграммы представлено в таблице 11.

**Таблица 11**

**Описание структуры записей таблицы "Собеседования"**

| **Наименование поля** | **Идентификатор поля** | **Тип поля** | **Длина поля** | **Прочее** |
| --- | --- | --- | --- | --- |
| Код\_Собеседования | Id\_Interview | Int | 5 | Ключевое поле |
| Код\_Кандидата | Id\_Candidate | Int | 5 | Внешний ключ |
| Дата\_Собеседования | Date\_Interview | Date | 10 |  |
| Время\_Собеседования | Time\_Interview | Time | 6 |  |
| Код\_Типа\_Собеседования | Id\_Type\_Interview | Int | 5 | Внешний ключ |
| Код\_Сотрудника | Id\_Employee | Int | 5 | Внешний ключ |
| Код\_Статуса | Id\_Status | Int | 5 | Внешний ключ |

Описание таблицы «Тип\_Собеседования» из ER-диаграммы представлено в приложении 9.

Описание таблицы «Статус\_Собеседования» из ER-диаграммы представлено в приложении 10.

Пример описания таблицы «Трудовой\_договор» из ER-диаграммы представлено в таблице 12.

**Таблица 12**

**Описание структуры записей таблицы "Трудовой\_договор"**

| **Наименование поля** | **Идентификатор поля** | **Тип поля** | **Длина поля** | **Прочее** |
| --- | --- | --- | --- | --- |
| Код\_Трудового\_Договора | Id\_Work\_Contract | Int | 5 | Ключевое поле |
| Дата\_Подписания\_Договора | Date\_Signing\_Contract | Date |  |  |
| Город\_Заключения\_Договора | City\_Conclusion\_Contract | Varchar | 20 |  |
| Код\_Кандидата | Id\_Candidate | Int | 5 | Внешний ключ |
| Код\_Сотрудника | Id\_Employee | Int | 5 | Внешний ключ |
| Код\_Отдела | Id\_Department | Int | 5 | Внешний ключ |
| Код\_Должности | Id\_Post | Int | 5 | Внешний ключ |
| Дата\_Начала\_Работы | Date\_Start\_Work | Date |  |  |
| Время\_Начала\_Работы | Time\_Start\_Work | Time | 6 |  |

Описание таблицы «Отдел» из ER-диаграммы представлено в приложении 11.

Описание таблицы «Кандидаты» из ER-диаграммы представлено в таблице 13.

**Таблица 13**

**Описание структуры записей таблицы "Кандидаты"**

| **Наименование поля** | **Идентификатор поля** | **Тип поля** | **Длина поля** | **Прочее** |
| --- | --- | --- | --- | --- |
| Код\_Кандидата | Id\_Candidate | Int | 5 | Ключевое поле |
| ФИО | Full\_Name | Varchar | 255 |  |
| Фото | Photo | Varbinary | MAX |  |
| Почта | Email | Varchar | 50 |  |
| Телефон | Telephone | Varchar | 20 |  |
| Серия\_Паспорта | Passport\_Series | Varchar | 10 |  |
| Номер\_Паспорта | Number\_Series | Varchar | 10 |  |
| Возраст | Age | Int | 5 |  |
| Код\_Должности | Id\_Post | Int | 5 | Внешний ключ |

Описание таблицы «Тестирование» из ER-диаграммы представлено в приложении 12.

Описание таблицы «Тесты» из ER-диаграммы представлено в приложении 13.

Используемые классы с их кратким описанием представлены в таблице 14.

**Таблица 14**

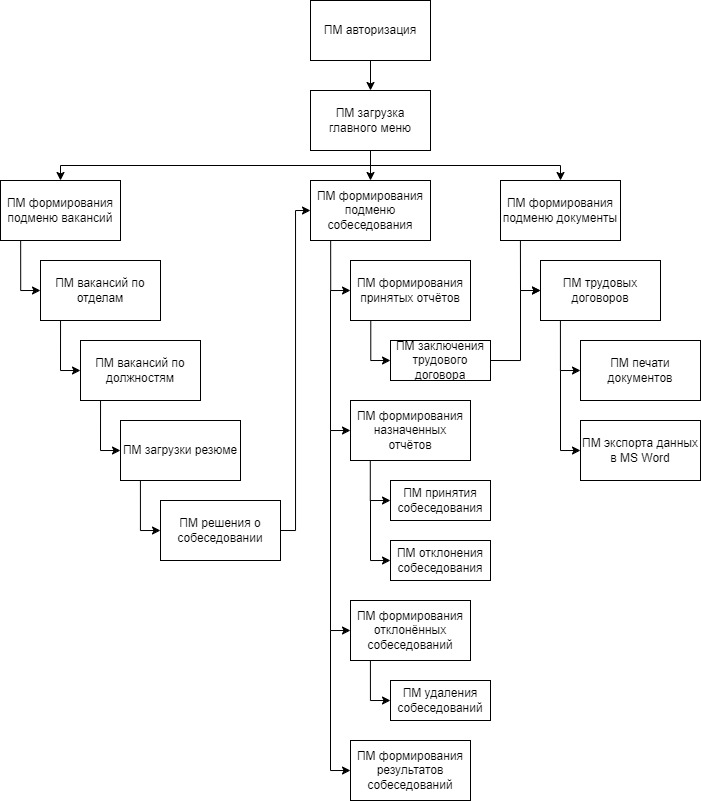
**Описание используемых классов**

| **Имя класса** | **Краткое описание** |
| --- | --- |
| LoginForm | Класс, содержащий строки для проверки и подключения к базе данных через приложение. |
| Menu | Класс, содержащий строки для перехода на многие формы в приложении. |
| SelectedCandidates | Класс, содержащий строки для подключения к базе данных и выборки кандидатов с определённой желаемой должностью. |
| Resume | Класс, содержащий строки для подключения к базе данных и вывода резюме необходимого кандидата. |
| Decision | Класс, содержащий строки для подключения к базе данных и добавления в неё новой информации. |
| AcceptedInterviews | Класс, содержащий строки для подключения к базе данных и выборки пройденных собеседований. |
| ExpectedInterviews | Класс, содержащий строки для подключения к базе данных и выборки ожидаемых собеседований. |
| RejectedInterviews | Класс, содержащий строки для подключения к базе данных и выборки отклонённых собеседований. |
| InterviewResults | Класс, содержащий строки для подключения к базе данных и выборки результатов по всем собеседованиям. |
| EmploymentContract | Класс, содержащий строки для подключения к базе данных и вывода всех заключённых трудовых договоров. |
| ConclusionEmployeeContract | Класс, содержащий строки для подключения к базе данных и заполнения базы данных новой информацией. |
| PrintEmploymentContract | Класс, содержащий строки для подключения к базе данных и печати трудовых договоров. |
| DeleteInterviews | Класс, содержащий строки для подключения к базе данных и удаления отклонённых интервью. |
| PassingTheInterview | Класс, содержащий строки для подключения к базе данных и изменения информации в ней. |

* + 1. **Структурная схема пакета (дерево вызова программных модулей)**

Программный модуль – это автономная единица кода, которая выполняет конкретную задачу или набор задач в рамках более крупной программной системы. Каждый модуль вносит свой вклад в общую функциональность программного обеспечения.

Дерево вызова программных модулей в разработанной системе представлено на рисунке 19.



**Рисунок 19. Дерево вызова программных модулей**

Подробное описание функций, которые выполняются модулями представлено в таблице 15.

**Таблица 15**

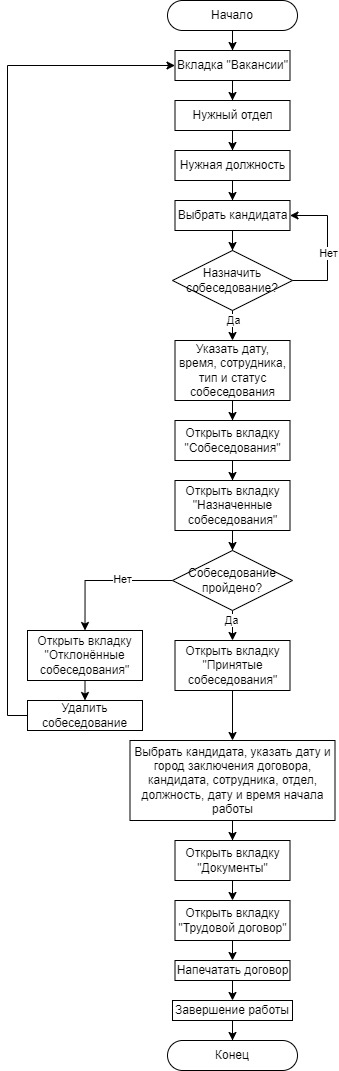
**Описание функций модулей в системе**

| **Наименование модуля** | **Выполняемые функции** |
| --- | --- |
| ПМ авторизация | Вызов меню авторизации для пользователя. |
| ПМ загрузка главного меню | Модуль, который инициирует вызов интерфейса системы. |
| ПМ формирования подменю вакансий | Загрузки и открытие подменю «Вакансии». |
| ПМ вакансий по отделам | Открытие и просмотр вакансий по отделам. |
| ПМ вакансий по должностям | Открытие и просмотр вакансий по должностям. |
| ПМ загрузки резюме | Загрузка и открытие резюме кандидата. |
| ПМ решения о собеседовании | Открытие формы для принятия решения о собеседовании. |
| ПМ формирования подменю собеседования | Загрузка и открытие подменю «Собеседования». |
| ПМ формирования принятых собеседований | Загрузка и просмотр принятых собеседований. |
| ПМ заключения трудового договора | Открытие формы для заключения договора и добавления данных в базу данных. |
| ПМ формирования назначенных собеседований | Загрузка и просмотр назначенных собеседований. |
| ПМ пройденные собеседования | Изменение данных в таблице базы данных. |
| ПМ отклонения собеседования | Изменение данных в таблице базы данных. |
| ПМ формирования отклонённых собеседований | Загрузка и просмотр отклонённых собеседований. |
| ПМ удаления собеседований | Удаление данных их таблицы базы данных. |
| ПМ формирования результатов собеседований | Просмотр результатов всех собеседоний. |
| ПМ формирования подменю документы | Загрузки и открытие подменю «Документы». |
| ПМ трудовых договоров | Загрузка и просмотр заключённых трудовых договоров. |
| ПМ печати документов | Модуль, обеспечивающий функциональность для печати трудового договора. |
| ПМ экспорта данных в MS Word | Отправка данных из базы данных в файл MS Word с помощью системы. |

* + 1. **Описание программных модулей**

Блок-схема – это визуальное представление процесса или алгоритма с использованием стандартных символов и соединительных линий. Это изображение с набором инструкций, которые помогаю выполнить шаги и принять решения, необходимые для достижения определённого результата.

Блок-схема основного модуля представлена на рисунке 20.



**Рисунок 20. Блок-схема основного модуля**

На блок-схеме присутствуют следующие процессы:

* Начало (вход в систему);
* Вкладка «Вакансии»;
* Выбор отдела;
* Выбор должности;
* Выбор кандидата;
* Назначение собеседования;
* Данные о собеседовании;
* Вкладка «Собеседования»;
* Вкладка «Назначенные собеседования»;
* Решение о собеседовании;
* Вкладка «Отклонённые собеседования»;
* Удаление собеседования;
* Вкладка «Пройденные собеседования»;
* Заполнение данных трудового договора;
* Вкладка «Документы»;
* Вкладка «Трудовой договор»;
* Печать договора;
* Завершение работы;
* Конец.

Код модуля представлен в приложении 14.

* 1. **Испытания разработанного решения**
     1. **Перечень объектов и функций, подлежащих испытаниям**

Перечень проверяемых функций и их примечания представлены в таблице 16.

**Таблица 16**

**Перечень проверяемых функций в системе**

| **№ п/п** | **Проверяемая функция** | **Примечание** |
| --- | --- | --- |
| 1 | Авторизация | Проверка на соответствие логину и паролю из базы данных. |
| 2 | Отображение списка вакансий | Проверка на правильность заполнения и корректное отображение списка. |
| 3 | Просмотр отобранных кандидатов по должностям | Проверка на правильность отображения информации о кандидатах. |
| 4 | Открытие резюме кандидата | Проверка на правильность отображения информации о кандидатах на форме. |
| 5 | Решение о собеседовании | Проверка на работоспособность кнопки «Решение» и переходе на следующую форму. |
| 6 | Назначение собеседования | Проверка на работоспособность справочников и кнопки «Назначить». |
| 7 | Отклонить собеседование | Проверка на работоспособность справочников и кнопки «Отклонить». |
| 8 | Отображение списка «Собеседования» | Проверка на правильность заполнения и корректное отображение списка. |
| 9 | Открытие формы «Назначенные собеседования» | Проверка на открытие формы и правильное отображение данных. |
| 10 | Открытие нужного собеседования | Проверка на открытие формы и корректность отображённых данных. |
| 11 | Собеседование пройдено | Проверка на работоспособность кнопки «Собеседование пройдено», обновления данных на форме и в базе данных. |
| 12 | Собеседование не пройдено | Проверка на работоспособность кнопки «Собеседование не пройдено», обновления данных на форме и в базе данных. |
| 13 | Открытие формы «Отклонённые собеседования» | Проверка на открытие формы и корректность отображённых данных. |
| 14 | Удаление отклонённого собеседования | Проверка на удаление нужного собеседования из формы и базы данных. |
| 15 | Открытие формы «Пройденные собеседования» | Проверка на открытие формы и корректность отображённых данных. |
| 16 | Заключение трудового договора | Проверка на открытие формы «Заключение трудового договора», нажатие на кнопку «Заключить» и добавление данных в другую таблицу базы данных. |
| 17 | Отображение списка документов | Проверка на правильность заполнения и корректное отображение списка. |
| 18 | Открытие формы «Трудовой договор» | Проверка на открытие формы и правильное отображение данных. |
| 19 | Открытие нужного трудового договора | Проверка на открытие формы и корректность отображённых данных. |
| 20 | Печать трудового договора | Проверка на работоспособность кнопки «Печать» и заполнение данными файл MS Word. |

* + 1. **Методы проведения испытаний**

Перечень проверок всех функций системы представлен в таблицах 17 – 36.

**Таблица 17**

**Описание проверки функции "Авторизация"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Авторизация |
| Предмет проверки | Время авторизации  Соответствие данным из базы данных |
| Используемые средства | Секундомер  Форма авторизации |
| Исходные данные | Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Ошибки в логине или пароле |
| Ожидаемая реакция | Время авторизации не более 1 секунды  Вывод сообщения об успешной авторизации |
| Способ определения результатов | Открытие формы главного меню |

**Таблица 18**

**Описание проверки функции "Отображение списка вакансий"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Отображение списка вакансий |
| Предмет проверки | Правильность заполнения и отображение списка |
| Используемые средства | Форма главного меню |
| Исходные данные | Данные из списка в главном меню |
| Искажения тестовой информации для имитации нештатных ситуаций | Обновление данных в базе данных |
| Ожидаемая реакция | При нажатии откроется список вакансий |
| Способ определения результатов | Открытие списка |

**Таблица 19**

**Описание проверки функции "Просмотр отобранных кандидатов по должностям"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Просмотр отобранных кандидатов по должностям |
| Предмет проверки | Правильность отображения информации  Отсутствие ошибок в заполнении |
| Используемые средства | Форма главного меню |
| Исходные данные | Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Корректный вывод данных из вакансий |
| Способ определения результатов | Просмотр дынных о каждом кандидате |

**Таблица 20**

**Описание проверки функции "Открытие резюме кандидата"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Открытие резюме кандидата |
| Предмет проверки | Правильность заполнения текстовых полей |
| Используемые средства | Форма резюме кандидатов |
| Исходные данные | Данные из формы вакансий  Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Правильный вывод данных о кандидатах |
| Способ определения результатов | Проверка данных из формы вакансий и базы данных |

**Таблица 21**

**Описание проверки функции "Решение о собеседовании"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Решение о собеседовании |
| Предмет проверки | Работоспособность кнопки «Решение» |
| Используемые средства | Кнопка на форме «Резюме» |
| Исходные данные | Данные из текстовых полей  Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | При нажатии на кнопку откроется форма |
| Способ определения результатов | Успешный переход на форму «Решение о собеседовании» |

**Таблица 22**

**Описание проверки функции "Назначение собеседования"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Назначение собеседования |
| Предмет проверки | Работоспособность кнопки «Назначить»  Успешное добавление данных в базу дынных |
| Используемые средства | База данных  Данные из текстовых полей |
| Исходные данные | Данные из формы «Решение о собеседовании» |
| Искажения тестовой информации для имитации нештатных ситуаций | Установить статус собеседования «Отклонено» |
| Ожидаемая реакция | Успешное добавление данных в базу данных |
| Способ определения результатов | Просмотр данных в форме «Назначенные собеседования» |

**Таблица 23**

**Описание проверки функции "Отклонить собеседование"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Отклонить собеседование |
| Предмет проверки | Работоспособность кнопки «Отклонить»  Успешное добавление данных в базу дынных |
| Используемые средства | База данных  Данные из текстовых полей |
| Исходные данные | Данные из формы «Решение о собеседовании» |
| Искажения тестовой информации для имитации нештатных ситуаций | Установить статус собеседования «Назначено» |
| Ожидаемая реакция | Успешное добавление данных в базу данных |
| Способ определения результатов | Просмотр данных в форме «Отклонённые собеседования» |

**Таблица 24**

**Описание проверки функции "Отображение списка Собеседования"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Отображение списка «Собеседования» |
| Предмет проверки | Правильность заполнения и отображение списка |
| Используемые средства | Форма главного меню |
| Исходные данные | Данные из списка в главном меню |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | При нажатии откроется список собеседований |
| Способ определения результатов | Открытие списка |

**Таблица 25**

**Описание проверки функции "Открытие формы Назначенные собеседования"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Открытие формы «Назначенные собеседования» |
| Предмет проверки | Правильность отображения информации  Отсутствие ошибок в заполнении |
| Используемые средства | Форма главного меню |
| Исходные данные | Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Правильный вывод данных о назначенных собеседованиях |
| Способ определения результатов | Проверка данных из формы назначенных собеседованиях и базы данных |

**Таблица 26**

**Описание проверки функции "Открытие нужного собеседования"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Открытие нужного собеседования |
| Предмет проверки | Проверка на открытие формы  Правильность заполненных данных |
| Используемые средства | Данные из формы «Назначенные собеседования» |
| Исходные данные | Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Успешное открытие формы  Верные данные в полях |
| Способ определения результатов | Открытие формы о прохождении собеседования |

**Таблица 27**

**Описание проверки функции "Собеседование пройдено"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Собеседование пройдено |
| Предмет проверки | Работоспособность кнопки «Собеседование пройдено»  Обновление данных в базе данных |
| Используемые средства | База данных  Данные из текстовых полей |
| Исходные данные | Данные из формы «Прохождение собеседования» |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Успешная работа кнопки  Обновлённые данные в базе данных |
| Способ определения результатов | Просмотр данных в форме «Принятые собеседования» и базе данных |

**Таблица 28**

**Описание проверки функции "Собеседование не пройдено"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Собеседование не пройдено |
| Предмет проверки | Работоспособность кнопки «Собеседование не пройдено»  Обновление данных в базе данных |
| Используемые средства | База данных  Данные из текстовых полей |
| Исходные данные | Данные из формы «Прохождение собеседования» |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Успешная работа кнопки  Обновлённые данные в базе данных |
| Способ определения результатов | Просмотр данных в форме «Отклонённые собеседования» и базе данных |

**Таблица 29**

**Описание проверки функции "Открытие формы Отклонённые собеседования"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Открытие формы «Отклонённые собеседования» |
| Предмет проверки | Правильность отображения информации  Отсутствие ошибок в заполнении |
| Используемые средства | Форма главного меню |
| Исходные данные | Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Правильное отображение информации  Отсутствие ошибок |
| Способ определения результатов | Проверка данных из формы отклонённых собеседований и базы данных |

**Таблица 30**

**Описание проверки функции "Удаление отклонённого собеседования"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Удаление отклонённого собеседования |
| Предмет проверки | Удаление необходимой информации |
| Используемые средства | Форма «Отклонённые собеседования»  SQL-запрос на удаление данных |
| Исходные данные | Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Успешное удаление информации |
| Способ определения результатов | Проверка удалённые данных |

**Таблица 31**

**Описание проверки функции "Открытие формы Пройденные собеседования"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Открытие формы «Пройденные собеседования» |
| Предмет проверки | Проверка на открытие формы  Корректное отображение данных |
| Используемые средства | Форма главного меню |
| Исходные данные | Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Успешное открытие формы  Корректное отображение данных |
| Способ определения результатов | Открытие формы |

**Таблица 32**

**Описание проверки функции "Заключение трудового договора"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Заключение трудового договора |
| Предмет проверки | Открытие формы «Заключение трудового договора»  Работоспособность кнопки «Заключить договор»  Добавление данных в базу данных |
| Используемые средства | Форма «Заключение трудового договора»  SQL-запрос на удаление данных |
| Исходные данные | Данные из таблицы «Принятые собеседования» |
| Искажения тестовой информации для имитации нештатных ситуаций | Ошибки при заполнении данных в форме «Заключение трудового договора» |
| Ожидаемая реакция | Открытие новой формы  Успешная работа кнопки  Добавление данных в базу данных |
| Способ определения результатов | Просмотр данных на форме «Трудовые договоры» и в базе данных |

**Таблица 33**

**Описание проверки функции "Отображение списка документов"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Отображение списка документов |
| Предмет проверки | Правильность заполнения и отображение списка |
| Используемые средства | Форма главного меню |
| Исходные данные | Данные из списка в главном меню |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Правильное заполнение и отображение списка |
| Способ определения результатов | Открытие списка |

**Таблица 34**

**Описание проверки функции "Открытие формы Трудовой договор"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Открытие формы «Трудовой договор» |
| Предмет проверки | Проверка на открытие формы  Правильное отображение данных |
| Используемые средства | Форма главного меню |
| Исходные данные | Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Успешное открытие формы  Правильное отображение данных |
| Способ определения результатов | Открытие формы |

**Таблица 35**

**Описание проверки функции "Открытие нужного трудового договора"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Открытие нужного трудового договора |
| Предмет проверки | Проверка на открытие формы  Корректность заполненных данных |
| Используемые средства | Данные из формы «Трудовые договоры» |
| Исходные данные | Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Успешное открытие формы  Верные данные в полях |
| Способ определения результатов | Открытие формы нужного договора |

**Таблица 36**

**Описание проверки функции "Печать трудового договора"**

| **Параметр** | **Значение** |
| --- | --- |
| Функция | Печать трудового договора |
| Предмет проверки | Работоспособность кнопки «Печать»  Время печати  Правильность вывода полей |
| Используемые средства | Секундомер  Шаблон договора |
| Исходные данные | Данные из базы данных |
| Искажения тестовой информации для имитации нештатных ситуаций | Изменение данных в базе данных |
| Ожидаемая реакция | Успешная работа кнопки «Печать»  Время печати не более 3 секунд  Данные подставлены в нужны поля |
| Способ определения результатов | Просмотр трудового договора |

* + 1. **Проведение проверочных испытаний и их результаты**

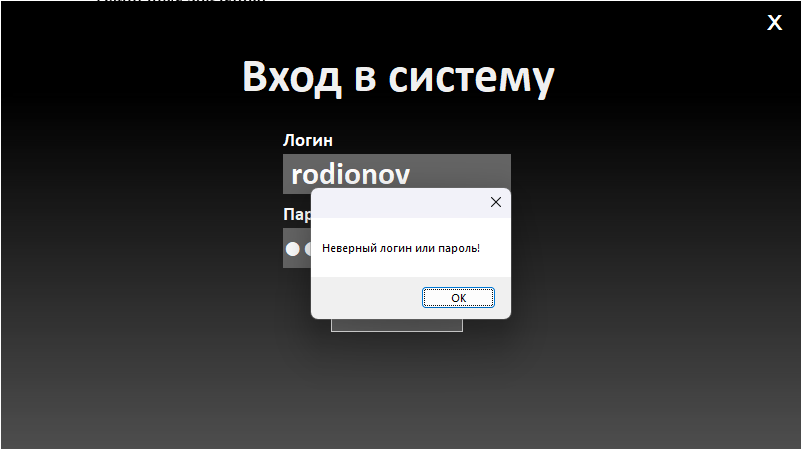
Результаты испытаний информационной системы представлены в таблице 37.

**Таблица 37**

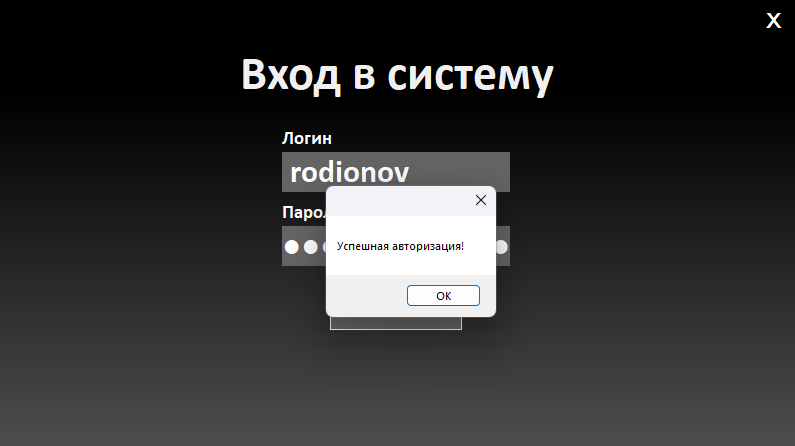
**Результаты испытаний**

| **№ проверки** | **Вид проверки** | **Критериальный параметр** | **Допустимые значения** | **Результат проверки** |
| --- | --- | --- | --- | --- |
| 1 | Ввод данных в форму авторизации | Максимальная длина логина и пароля | Не более 50 | Соответствует |
| Реакция на неверный логин или пароль | Сообщение об ошибке в отдельном окне. | Соответствует |
| Реакция на верные данные | Сообщение об успешной авторизации | Соответствует |
| 2 | Изменение данных в резюме кандидата | Данные не должны изменяться | Данные нельзя изменить | Соответствует |
| 3 | Изменение данных в форме «Решение о собеседовании» | Максимальное количество выбора данных из справочников | Не более одного | Соответствует |
| Допустимый диапазон дат | Настоящая дата-31.12.2100 | Соответствует |
| 4 | Нажатия на кнопки на форме «Решение о собеседовании» | Нажатие на кнопку «Назначить» | Статус - назначено | Соответствует |
| Нажатие на кнопку «Отклонить» | Статус - отклонено | Соответствует |
| 5 | Изменение данных в форме «Прохождение собеседования» | Данные не должны изменяться | Данные нельзя изменить | Соответствует |
| 6 | Изменение данных в форме «Удаление собеседования» | Данные не должны изменяться | Данные нельзя изменить | Соответствует |
| 7 | Изменение данных в форме «Заключение трудового договора» | Максимальное количество выбора данных из справочников | Не более одного | Соответствует |
| Допустимый диапазон дат | Настоящая дата-31.12.2100 | Соответствует |
| 8 | Изменение данных в форме «Печать трудового договора» | Данные не должны изменяться | Данные нельзя изменить | Соответствует |
| 9 | Печать трудового договора | Время печати договора | Не более 3 с | 2,4 с |

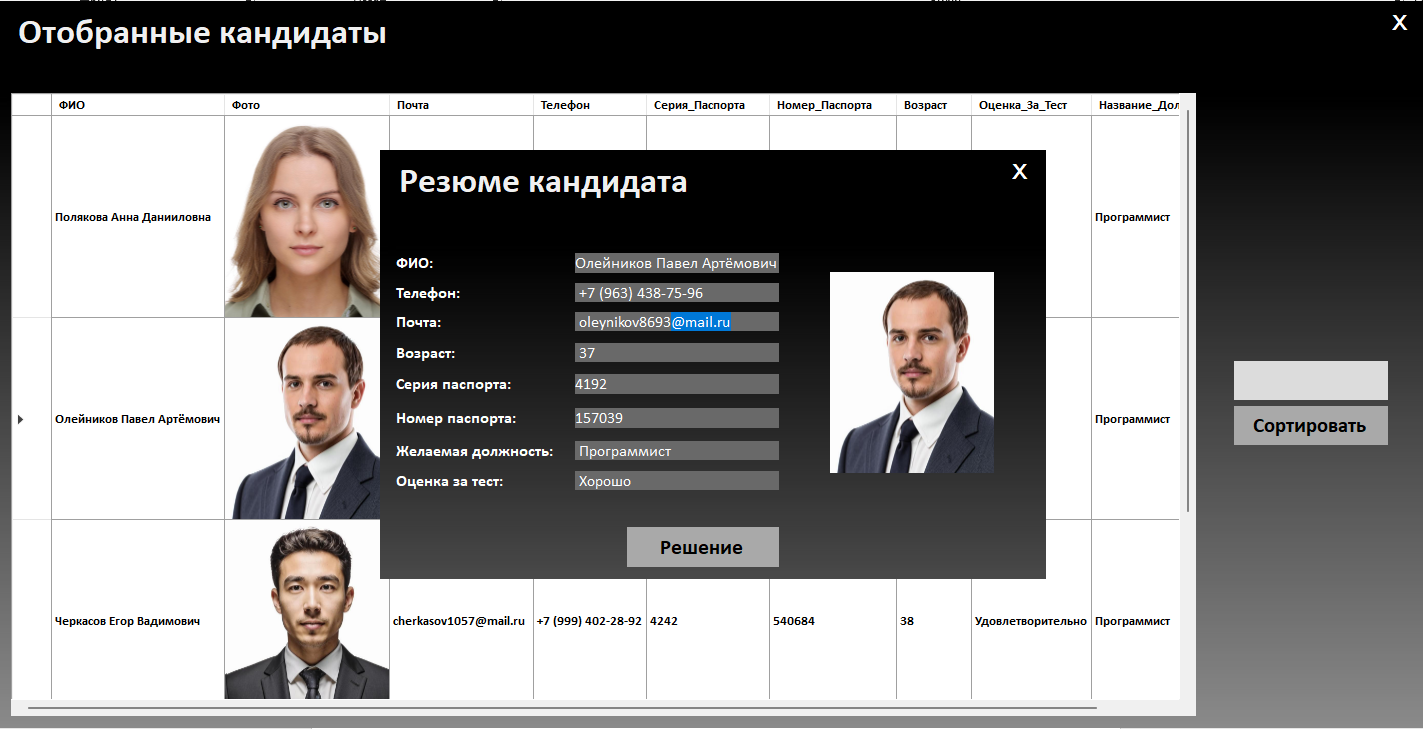
Экранные формы при проведении тестов представлены на рисунках 21 – 30.



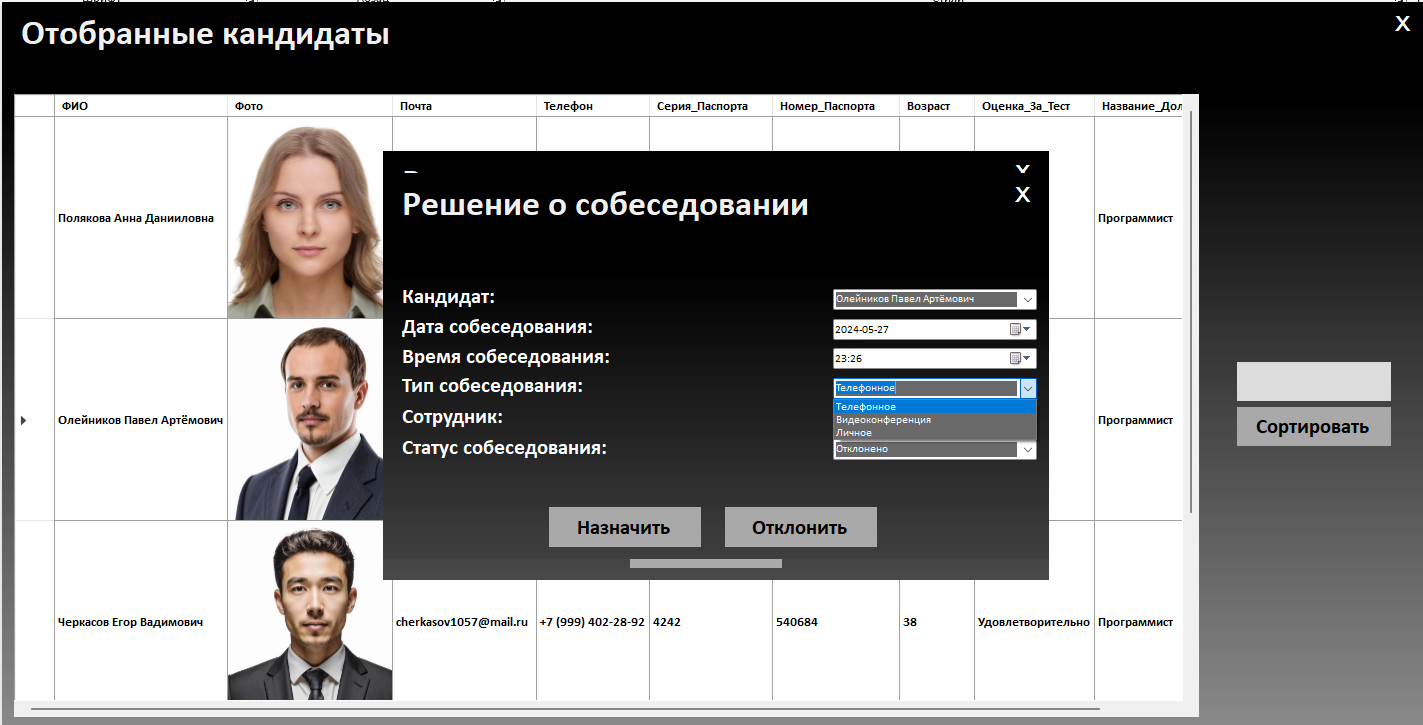
**Рисунок 21. Реакция на неверные данные**



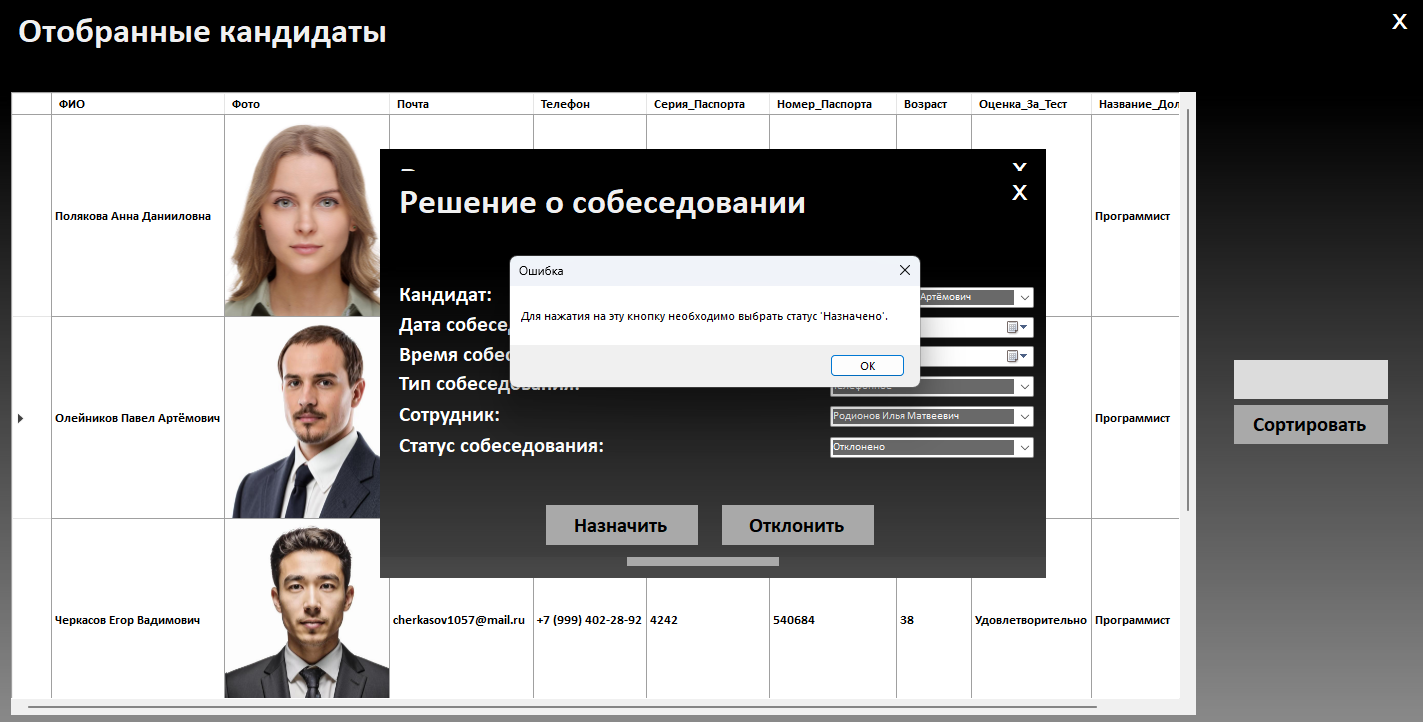
**Рисунок 22. Реакция на верные данные**



**Рисунок 23. Возможность изменения данных на форме "Резюме кандидата" отсутствует**



**Рисунок 24. Выбор данных из справочников на форме "Решение о собеседовании"**

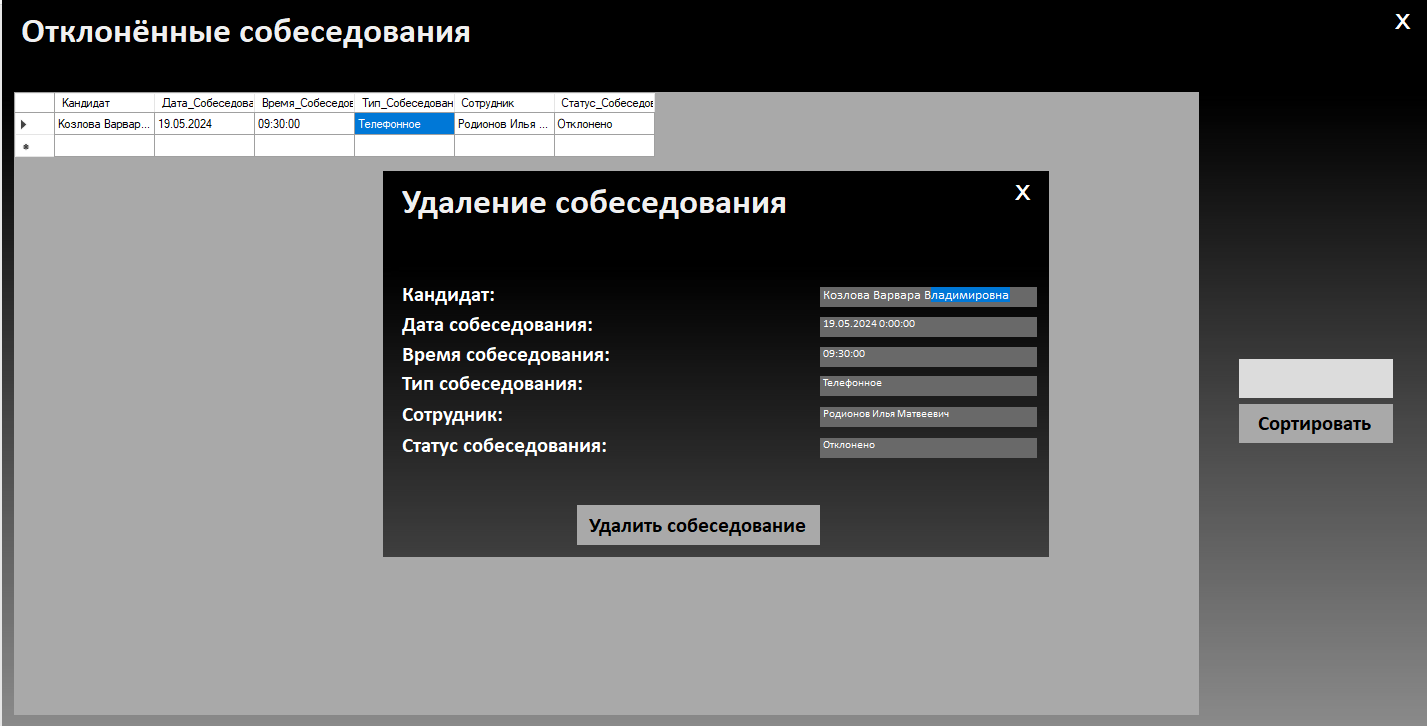


**Рисунок 25. Нажатие на кнопку "Назначить" при статусе "Отклонено"**

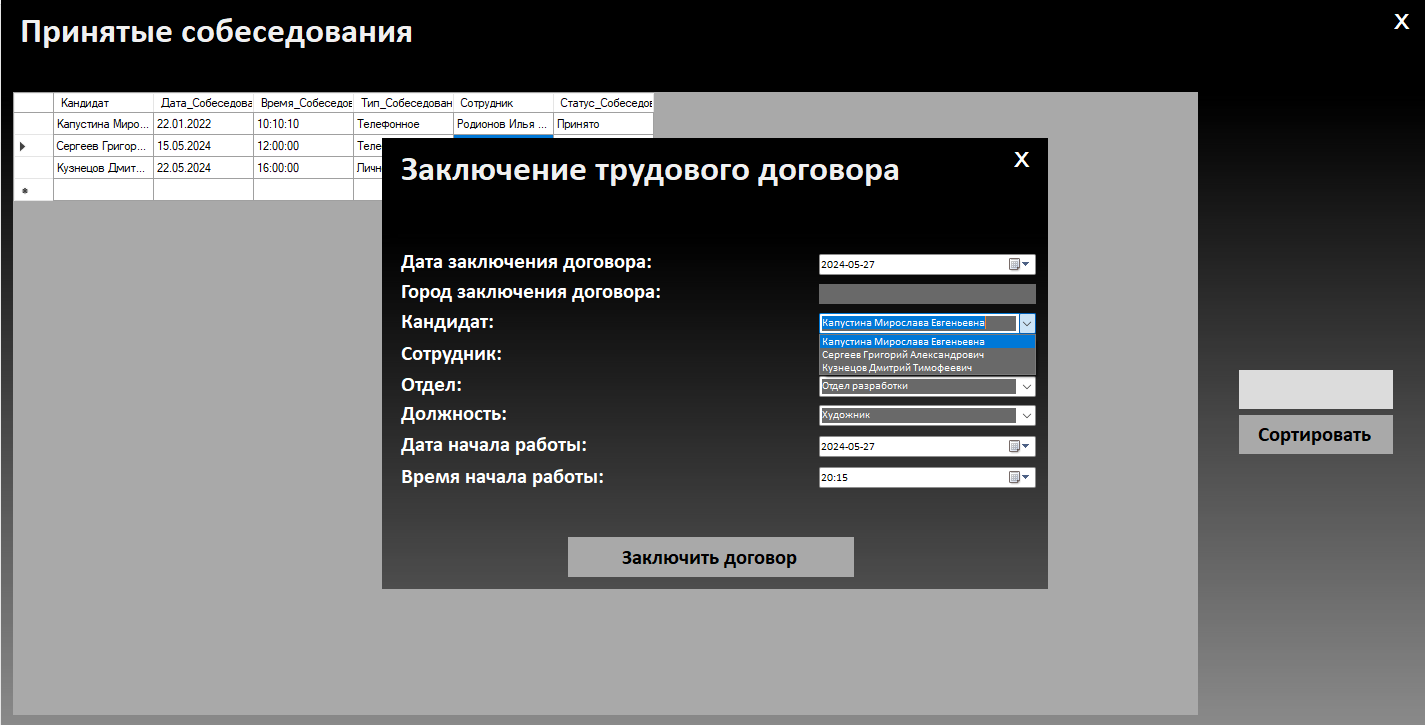
Результат после нажатия на кнопку "Отклонить" при статусе "Назначено" представлен в приложении 15.



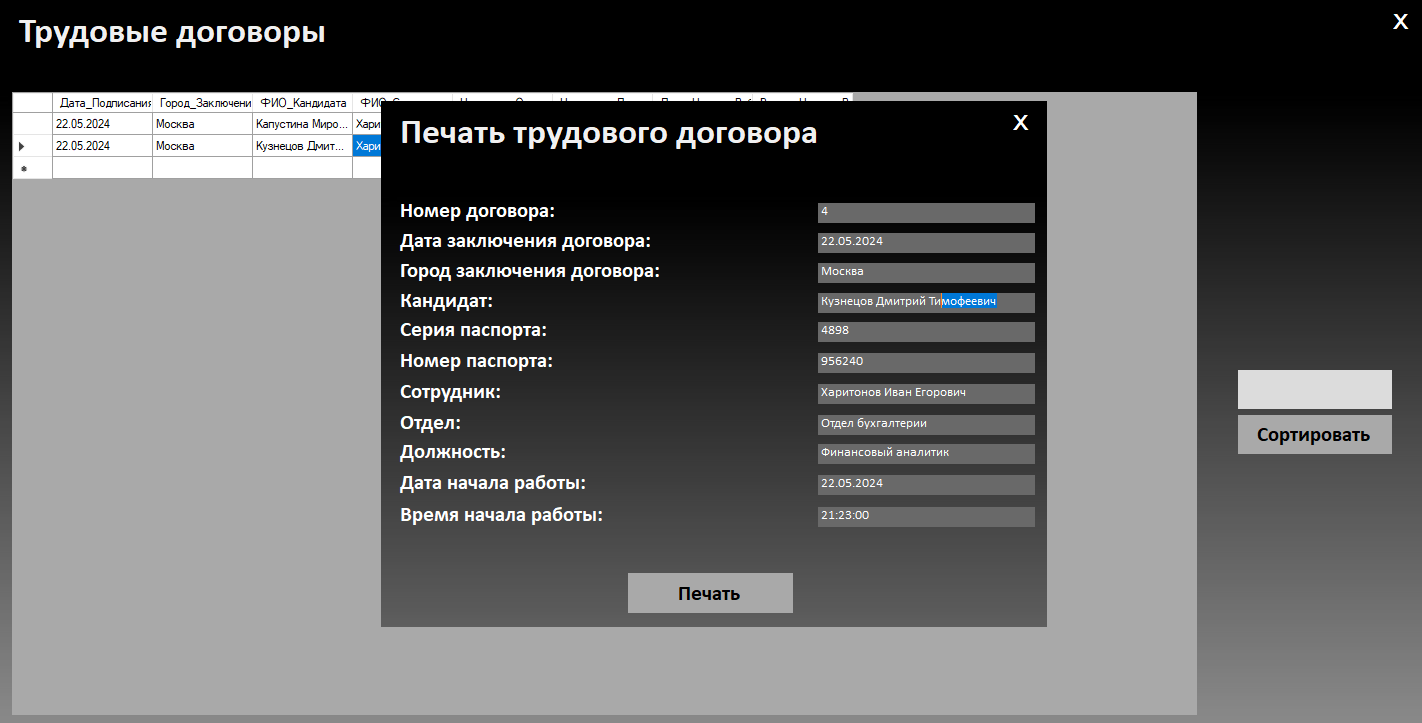
**Рисунок 26. Возможность изменения данных на форме "Прохождение собеседования" отсутствует**



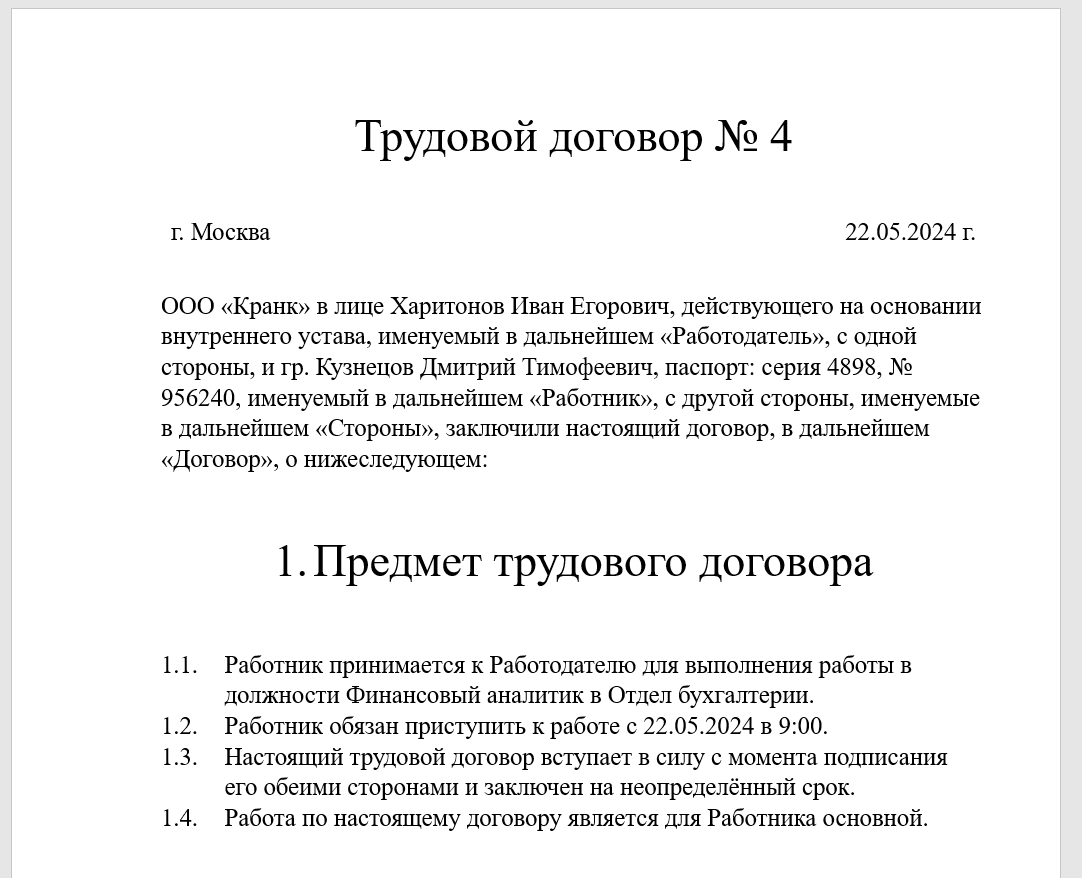
**Рисунок 27. Возможность изменения данных на форме "Удаление собеседования" отсутствует**



**Рисунок 28. Выбор данных из справочников на форме "Заключение трудового договора"**



**Рисунок 29. Возможность изменения данных на форме "Печать трудового договора" отсутствует**



**Рисунок 30. Напечатанный трудовой договор**

**Глава 3. Обоснование экономической эффективности проекта**

* 1. **Выбор и обоснование методики расчёта экономической эффективности**

Оценка экономической эффективности ИС – это комплексный анализ, который количественно определяет ценность и влияние системы на финансовые показатели бизнеса. Её цель – определить, как ИС способствует прибыльности, производительности и общему успеху компании.

Оценка экономической эффективности информационной системы имеет решающее значение для обеспечения её успешного внедрения и полной реализации её ценности для бизнеса. Она требует тщательного анализа и сосредоточения внимания нп количественной оценке как материальных, так и нематериальных выгод. Уделяя время и ресурсы такой оценке, предприятия могут принимать обоснованные решения, оптимизировать свои инвестиции и максимизировать отдачу от своих технологических активов.

Экономическая эффективность обычно оценивается путём сравнения затраченных финансовых ресурсов с полученными результатами. При этом ключевым фактором является достижение установленной нормы прибыли. Данная оценка проводится как для проекта в целом, так и для каждого участника с учётом его вклада в общий результат.

Экономическая эффективность проекта складывается из двух составляющих:

* Косвенного эффекта;
* Прямого эффекта;

Ожидаемый косвенный эффект характеризуется:

* Увеличением прибыли;
* Привлечением большого числа кандидатов;
* Снижением уровня ошибок при подборе персонала.

Ожидаемы прямой эффект характеризуется:

* Снижением трудовых и стоимостных затрат.

Оценка экономический эффективности будет проводиться путём сравнения эффективности обработки информации в рамках текущей практики ведения бизнеса с эффективностью после внедрения разработанной системы. Ключевым показателем этой эффективности будет снижение экономических затрат, связанных с обработкой информации, в связи с переходом к машинной обработке. При оценке прямой эффективности в стоимостных единицах измерения рассчитываются две группы показателей – показатель затрат и показатель снижения стоимостных затрат.

При расчёте изменения трудовых затрат на обработку информации используется следующая система показателей:

1. Абсолютный показатель снижения трудовых затрат на обработку информации.

**ΔT = Т0 - Т1** (3.1.1)

Т0 – годовая трудоёмкость обработки информации при базовом варианте;

Т1 – годовая стоимость обработки информации при проектируемом варианте.

1. Коэффициент снижения трудовых затрат.

**Kт=(ΔТ/Т0)\*100%** (3.1.2)

1. Индекс снижения трудовых затрат, который показывает рост производительности труда при обработке информации.

**Yт=Т0/Т1**(3.1.3)

К стоимостным показателям относятся: показатель снижения стоимостных затрат, коэффициент эффективности по затратам, индекс изменения стоимостных затрат:

1. Показатель снижения стоимостных затрат.

**ΔС=С0-С1** (3.1.4)

С0 – годовая стоимость обработки информации при базовом варианте;

С1 – годовая стоимость обработки информации при проектируемом варианте.

1. Коэффициент эффективности по затратам.

**Kc=(ΔС/С0)\*100%** (3.1.5)

1. Индекс изменения стоимостных затрат.

**Yc=С0/С1** (3.1.6)

Помимо рассматриваемых показателей будет проведён расчёт срока окупаемости затрат на внедрение проекта машинной обработки информации.

**Ток = КП /ΔC** (3.1.7)

КП - затраты на создание проекта (проектирование и внедрение).

* 1. **Расчёт показателей экономической эффективности проекта**

Для того, чтобы рассчитать сумму затрат на проект автоматизации необходимо произвести следующие расчёты:

* Расчёт стоимости трудовых затрат (представлен в таблице 38);
* Расчёт стоимости материальных ресурсов (представлен в таблице 39);
* Расчёт косвенных затрат на проект (представлен в таблице 40).

**Таблица 38**

**Расчёт трудовых затрат на разработку ИС**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Этап жизненного цикла** | **Виды работ** | **Количество специалистов** | **Время выполнения** | **Затраты**  **(руб.)** |
| Планирование | Определение целей | 1 | 5 дней | 16.000 |
| Определение требований к системе | 2 | 5 дней | 25.000 |
| Составление ТЗ | 1 | 2 дня | 6.000 |
| Анализ | Анализ рынка ПО | 1 | 1 неделя | 10.000 |
| Анализ бизнес-процессов | 2 | 1 неделя | 45.000 |
| Анализ данных | 1 | 5 дней | 15.000 |
| Проектирование | Выбор технологий и платформ | 1 | 2 дня | 8.000 |
| Функциональное описание | 1 | 5 дней | 10.000 |
| Дизайн системы | 2 | 1 неделя | 40.000 |
| Реализация | Разработка системы | 2 | 2 недели | 50.000 |
| Интеграция | 1 | 4 дня | 20.000 |
| Документирование | 1 | 4 дня | 6.000 |
| Тестирование | Функциональное тестирование | 1 | 2 дня | 8.000 |
| Тестирование производительности | 1 | 2 дня | 8.000 |
| Тестирование безопасности | 1 | 2 дня | 10.000 |
| Внедрение | Подготовка среды | 2 | 4 дня | 20.000 |
| Установка системы | 1 | 1 день | 10.000 |
| Обучение пользователей | 1 | 4 дня | 20.000 |

**Таблица 39**

**Стоимость материальных затрат на проект**

| **Статья затрат** | **Стоимость** |
| --- | --- |
| Покупка дополнительных аппаратных средств (сервер) | 50.000 |
| Покупка специализированного ПО для разработки (Visual Studio Enterprise 2022) | 25.000 |
| Итого: | 75.000 |

**Таблица 40**

**Косвенные затраты на проект**

|  |  |
| --- | --- |
| **Статья затрат** | **Стоимость** |
| Затраты на электроэнергию | 15.000 |
| Рисковый запас | 80.000 |
| Накладные расходы | 3.000 |
| Итого: | 98.000 |

Таким образом полные затраты на проект (КП) составляют 327,000 + 75,000 + 98,000 = 500,000 рублей.

Расчёт трудовых и стоимостных затрат для базового варианта представлен в таблице 41.

**Таблица 41**

**Расчёт трудовых и стоимостных затрат для базового варианта**

| **№п/п** | **Наименование операции** | **Количество документов в мес., шт.** | **Трудоемкость на 1 документ (чел.-час)** | **Трудоемкость за год (чел.-час)** | **З/п специалиста за час (руб.)** | **Накладные расходы на 1 документ (руб.)** | **Накладные расходы за год (руб.)** | **Общие стоимостные затраты за год (руб.)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. | Анализ вакансий и требований | 20 | 0,5 | 120 | 900 | 0 | 0 | 108000 |
| 2. | Подбор кандидатов | 40 | 0,3 | 144 | 900 | 0 | 0 | 129600 |
| 3. | Проведение собеседования | 40 | 1 | 480 | 900 | 4 | 1920 | 433920 |
| 4. | Принятие решения о найме | 40 | 0,3 | 144 | 900 | 4 | 1920 | 131520 |
| 5. | Оформление трудового договора | 40 | 0,3 | 144 | 900 | 10 | 4800 | 134400 |
|  | Итого |  |  | 1032 |  |  |  | 937440 |

Расчёт трудовых и стоимостных затрат для проектируемого проекта представлен в таблице 42.

**Таблица 42**

**Расчёт трудовых и стоимостных затрат для проектируемого варианта**

| **№п/п** | **Наименование операции** | **Количество документов в мес., шт.** | **Трудоемкость на 1 документ (чел.-час)** | **Трудоемкость за год (чел.-час)** | **З/п специалиста за час (руб.)** | **Накладные расходы на 1 документ (руб.)** | **Накладные расходы за год (руб.)** | **Общие стоимостные затраты за год (руб.)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. | Сбор информации о вакансиях | 20 | 0,4 | 96 | 900 | 0 | 0 | 86400 |
| 2. | Сбор резюме кандидатов | 40 | 0,2 | 96 | 900 | 0 | 0 | 86400 |
| 3. | Анализ квалифицированных специалистов | 40 | 0,3 | 144 | 900 | 0 | 0 | 129600 |
| 4. | Проведение собеседования | 40 | 0,4 | 192 | 900 | 4 | 1920 | 174720 |
| 5. | Выбор кандидатов | 40 | 0,2 | 96 | 900 | 10 | 4800 | 91200 |
|  | Итого |  |  | 624 |  |  |  | 568320 |

Расчёт показателей эффективности внедрения проекта представлен в таблице 43.

**Таблица 43**

**Расчёт показателей эффективности внедрения проекта**

|  | **Абсолютное изменение затрат** | **Коэффициент изменения затрат Kт** | **Индекс изменения затрат Yт** |
| --- | --- | --- | --- |
| Трудовые затраты | ΔТ =Т0-Т1 | =ΔТ/T0 \*100% | =T0/T1 |
| 408 | 0,4 | 1,6 |
| Стоимостные затраты | ΔС =C0-C1 (руб.) | КC=ΔС/C0 \*100% | YC=C0/C1 |
| 369120 | 0,4 | 1,6 |

Рассчитанные показатели экономической эффективности и их сопоставление продемонстрированно на диаграммах 1 и 2.

**Диаграмма 1. Сопоставление трудовых затрат**

**Диаграмма 2. Сопоставление стоимостных затрат**

Расчёт срока окупаемости разработанного проекта:

Ток = Кп/ΔC = 500000 / 369120 = 1,3 (года) = 15 месяцев.

Улучшение качественных характеристик процесса подбора персонала:

* Скорость и эффективность. Разработанная система значительно сокращает время, затрачиваемое на такие задачи, как анализ вакансий, подбор кандидатов, проведение собеседования, принятие решения о найме и оформление трудового договора;
* Улучшенный опыт с кандидатами. Разработанная система обеспечивает более плавный и упорядоченный процесс работы с кандидатами, что приводит к положительному восприятию компании и потенциально привлекает кандидатов более высокого уровня.

Влияние автоматизированного процесса подбора персонала на эффективность деятельности органов управления:

* Снижение нагрузки. Отдел кадров тратит меньше времени на выполнение ручных задач, что способствует более быстрой и эффективной работе;
* Улучшенная отчётность и аналитика. Разработанная система предоставляет данные о показателях подбора персонала, что позволяет отделу кадров принимать более обоснованные решения о найме.

Влияние на конечные результаты:

* Повышение качества найма. Разработанная система способствует найму более квалифицированных и подходящих кандидатов, что приводит к повышению производительности и успешности персонала;
* Снижение затрат на найм. Разработанная система снижает затраты, связанные с временем и финансами.

Вывод исходя из показателей в таблице. Проектируемая система демонстрирует высокий потенциал для повышения эффективности подбора персонала за счёт снижения трудозатрат и материальных затрат. Внедрение системы приведёт к значительному повышению эффективности автоматизируемого бизнес-процесса и увеличению прибыли компании.

**Заключение**

В ходе этой дипломной работы была решена задача автоматизации бизнес-процесса подбора персонала на примере компании ООО «Кранк», которая занимается разработкой успешный и популярных игр в разных жанрах. Была составлена организационная структура компании, а также техническая и программная архитектуры с подробным описанием аппаратных средств и установленным программным обеспечением. Определены основные автоматизированные и неавтоматизированные бизнес-процессы в компании. Составлена контекстная диаграмма и декомпозиция выбранного бизнес-процесса до внедрения и после, подробно описаны входные и выходные документы, а также механизмы и ресурсы.

Были созданы схемы документооборотов документов, таких как: документ о запросе на подбор персонала, трудовой договор, документ о результатах интервью, документ о подписании соглашения о конфиденциальности. Также были проведены оценки потоков документов в компании.

Проведён анализ политики безопасности, которая действует в компании, рассмотрены регламенты использования сети Internet, электронной почты и коммерческой тайны. Перечислены программные средства для защиты ИБ и ЗИ, проведён анализ аппаратных средств ИБ и ЗИ, показан порядок реализации системы обеспечения ИБ и ЗИ. Анализировано обеспечение защиты на разных уровнях, таких как: программный, аппаратный и организационный. Составлен список средств, которые используются для защиты от внешних и инсайдерских угроз.

Проведён анализ популярного программного продукта в области автоматизации процесса подбора персонала, разобраны его плюсы и минусы. Также написан список задач, которые должна решать проектируемая технология и чем она должна отличаться от существующих средств. Составлена таблица со сравнением разрабатываемого от готового решения и написан вывод.

В следующем пункте были показаны этапы разработки стратегии реализации проекта, с кратким описанием этапов. Выбрана стратегия реализации и расписано, что эта стратегия обеспечивает.

Были указаны и расписаны возможные варианты приобретения информационных систем. Выбран подходящий вариант и факторы, на которые было обращено внимание при выборе.

Далее рассматривались все возможные модели жизненного цикла и выбрана та модель, которая подходит для проекта побольше всего. Приведён список ключевых положений по каждому этапу жизненного цикла и написаны их цели, ключевые участники, требования к входной информации и получаемые результаты. Приведён список ожидаемых рисков и шаги для их предотвращения на каждом этапе жизненного цикла.

Подробно описаны входные документы и файлы в системе, справочники, которые присутствуют в системе их реквизитный состав и формы приложения.

Далее были описаны внешние ключи, которые присутствуют в таблицах базы данных, код, с помощью которого происходит подключение к базе данных и выборка нужных данных из таблицы. Так же было показано, как данные из базы данных отображаются на формах и как происходит печать трудового договора.

Представлен сценарий диалога информационной системы с описанием функций.

Показана ER-диаграммы разработанной базы данных с подробным описанием каждого атрибута сущностей. Описание представлено в таблице, состоящей из таких колонок, как наименование поля, идентификатор поля, тип поля, длина поля и указание ключевого поля или внешнего ключа. Также были перечислены используемые классы с их кратким описанием.

Далее было представлено дерево вызова программных модулей в разработанной системе с описанием их функций. Показана блок-схема основного модуля и перечислены процессы, которые входят в этот модуль.

Приведён перечень проверяемых функций с их описанием. Проведены испытания на работоспособность каждой функции с подтверждением проверки на рисунках. Составлена таблица с результатами испытаний.

Выбрана и обоснована методика расчёта экономической эффективности проекта. Приведены формулы для расчёта каждого пункта трудовых и стоимостных затрат. Представлена таблица с расчётом стоимости создания и внедрения проекта, таблица с трудовыми затратами и показателями экономической эффективности проекта.

В будущем планируется интеграция с другими системами, например системы планирования ресурсных предприятий для синхронизации данных о вакансиях сотрудниках и финансовых показателях. Добавить в систему больше отчётов и панелей мониторинга для отслеживания ключевых показателей эффективности и выявления областей для улучшения.

**Список использованной литературы**

**Нормативно-правовые акты**

1. Трудовой кодекс Российской Федерации от 30.12.2001 – URL: <https://www.consultant.ru/document/cons_doc_LAW_34683/>
2. Федеральный закон «О персональных данных» от 27.07.2006 N 152-ФЗ – URL: <https://www.consultant.ru/document/cons_doc_LAW_61801/>

**Научная и учебная литература**

1. Управление персоналом организации: Учебник/ Под ред. А.Я. Кибанова. – М.: ИНФРА-М, 2024 – URL: <https://book.ru/book/954629>
2. Базаров Т.Ю., Еремин Б.Л., Управление персоналом. Учебник 2002 – URL: <https://studylib.ru/doc/6288026/bazarov-t.yu.--eremin-b.l.--upravlenie-personalom.-uchebnik>
3. ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средст – URL: <https://www.rbsnt-new.ru/wp-content/uploads/2020/11/ГОСТ-12207-2010.pdf>

**Электронные ресурсы**

| **№** | **Наименование портала (издания, курса, документа)** | **Ссылка** |
| --- | --- | --- |
| 1. | Проверка и анализ российских юридических лиц и предпринимателей | <https://www.rusprofile.ru/> |
| 2. | Виды и функции организационной структуры предприятия | <https://journal.sovcombank.ru/biznesu/organizatsionnaya-struktura-predpriyatiya-kak-glavnii-borets-s-haosom-v-biznes-protsessah> |
| 4. | Архитектура программного обеспечения | <https://practicum.yandex.ru/blog/arhitektura-programmnogo-obespecheniya/> |
| 5. | Описание технической архитектуры | <https://www.calscenter.ru/uploads/ОКУЛЯР_Описание-технической-архитектуры.pdf> |
| 6. | Методология функционального моделирования IDEF0 | <https://advanced-quality-tools.ru/assets/idef0-rus.pdf> |
| 7. | Ведение и организация документооборота на предприятии | <https://kontur.ru/diadoc/spravka/44407-vedenie_i_organizaciya_dokumentooborota> |
| 8. | Программные средства для защиты информации | <https://studfile.net/preview/5866814/page:5/> |
| 9. | Аппаратные средства ИС и варианты их применения | <https://konfo.ru/company-konfo/kolonka-eksperta-apparatnye-sredstva-zashchity-informatsionnykh-sistem/> |
| 10. | Модели жизненного цикла программного обеспечения | <https://habr.com/ru/articles/111674/> |
| 11. | Создание приложения Windows Forms с помощью C# | <https://learn.microsoft.com/ru-ru/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022> |
| 12. | Что такое SQL Server Management Studio (SSMS) | <https://learn.microsoft.com/ru-ru/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16> |
| 13. | Подключение и запрос SQL Server с помощью SSMS | <https://learn.microsoft.com/ru-ru/sql/ssms/quickstarts/ssms-connect-query-sql-server?view=sql-server-ver16> |
| 14. | Структурная схема пакета (дерево вызова программных модулей) | <https://studfile.net/preview/8998414/page:6/> |
| 15. | Блок-схема программного модуля | <https://studfile.net/preview/9258522/page:9/> |

**Приложения**

**Приложение 1. Код для подключения к базе данных и выборки данных для формы «Пройденные собеседования»**

string connectionString18 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection18 = new SqlConnection(connectionString18);

connection18.Open();

SqlDataAdapter dataAdapter18 = new SqlDataAdapter

("SELECT Собеседования.Код\_Собеседования, Кандидаты.ФИО AS Кандидат, Собеседования.Дата\_Собеседования, Собеседования.Время\_Собеседования, Тип\_Собеседования.Название\_Типа\_Собеседования AS Тип\_Собеседования, " +

"Сотрудники.ФИО AS Сотрудник, Статус\_Собеседования.Название\_Статуса\_Собеседования AS Статус\_Собеседования " +

"FROM Собеседования, Кандидаты, Тип\_Собеседования, Сотрудники, Статус\_Собеседования " +

"WHERE Собеседования.Код\_Кандидата = Кандидаты.Код\_Кандидата AND Собеседования.Код\_Типа\_Собеседования = Тип\_Собеседования.Код\_Типа\_Собеседования " +

"AND Собеседования.Код\_Сотрудника = Сотрудники.Код\_Сотрудника AND Собеседования.Код\_Статуса = Статус\_Собеседования.Код\_Статуса AND Статус\_Собеседования.Код\_Статуса LIKE 1", connection18);

DataTable table18 = new DataTable();

dataAdapter18.Fill(table18);

dataGridView1.DataSource = table18;

dataGridView1.Columns[0].Visible = false;

connection18.Close();

break;

**Приложение 2. Код для подключения к базе данных и выборки данных для формы «Назначенные собеседования»**

string connectionString18 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection18 = new SqlConnection(connectionString18);

connection18.Open();

SqlDataAdapter dataAdapter18 = new SqlDataAdapter

("SELECT Собеседования.Код\_Собеседования, Кандидаты.ФИО AS Кандидат, Собеседования.Дата\_Собеседования, Собеседования.Время\_Собеседования, Тип\_Собеседования.Название\_Типа\_Собеседования AS Тип\_Собеседования, " +

"Сотрудники.ФИО AS Сотрудник, Статус\_Собеседования.Название\_Статуса\_Собеседования AS Статус\_Собеседования " +

"FROM Собеседования, Кандидаты, Тип\_Собеседования, Сотрудники, Статус\_Собеседования " +

"WHERE Собеседования.Код\_Кандидата = Кандидаты.Код\_Кандидата AND Собеседования.Код\_Типа\_Собеседования = Тип\_Собеседования.Код\_Типа\_Собеседования " +

"AND Собеседования.Код\_Сотрудника = Сотрудники.Код\_Сотрудника AND Собеседования.Код\_Статуса = Статус\_Собеседования.Код\_Статуса AND Статус\_Собеседования.Код\_Статуса LIKE 3", connection18);

DataTable table18 = new DataTable();

dataAdapter18.Fill(table18);

dataGridView1.DataSource = table18;

dataGridView1.Columns[0].Visible = false;

connection18.Close();

break;

**Приложение 3. Код для подключения к базе данных и выборки данных для формы «Отклонённые собеседования»**

string connectionString18 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection18 = new SqlConnection(connectionString18);

connection18.Open();

SqlDataAdapter dataAdapter18 = new SqlDataAdapter

("SELECT Собеседования.Код\_Собеседования, Кандидаты.ФИО AS Кандидат, Собеседования.Дата\_Собеседования, Собеседования.Время\_Собеседования, Тип\_Собеседования.Название\_Типа\_Собеседования AS Тип\_Собеседования, " +

"Сотрудники.ФИО AS Сотрудник, Статус\_Собеседования.Название\_Статуса\_Собеседования AS Статус\_Собеседования " +

"FROM Собеседования, Кандидаты, Тип\_Собеседования, Сотрудники, Статус\_Собеседования " +

"WHERE Собеседования.Код\_Кандидата = Кандидаты.Код\_Кандидата AND Собеседования.Код\_Типа\_Собеседования = Тип\_Собеседования.Код\_Типа\_Собеседования " +

"AND Собеседования.Код\_Сотрудника = Сотрудники.Код\_Сотрудника AND Собеседования.Код\_Статуса = Статус\_Собеседования.Код\_Статуса AND Статус\_Собеседования.Код\_Статуса LIKE 2", connection18);

DataTable table18 = new DataTable();

dataAdapter18.Fill(table18);

dataGridView1.DataSource = table18;

dataGridView1.Columns[0].Visible = false;

connection18.Close();

break;

**Приложение 4. Код для подключения к базе данных и вывода информации в элемент на форме**

string connectionString18 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection18 = new SqlConnection(connectionString18);

connection18.Open();

SqlDataAdapter dataAdapter18 = new SqlDataAdapter

("SELECT Трудовой\_Договор.Код\_Трудового\_Договора, Трудовой\_Договор.Дата\_Подписания\_Договора, Трудовой\_Договор.Город\_Заключения\_Договора, " +

"Кандидаты.ФИО AS ФИО\_Кандидата, Сотрудники.ФИО AS ФИО\_Сотрудника, Отдел.Название\_Отдела, " +

"Должность.Название\_Должности, Трудовой\_Договор.Дата\_Начала\_Работы, Трудовой\_Договор.Время\_Начала\_Работы " +

"FROM Трудовой\_Договор, Кандидаты, Сотрудники, Отдел, Должность " +

"WHERE Трудовой\_Договор.Код\_Кандидата = Кандидаты.Код\_Кандидата AND Трудовой\_Договор.Код\_Сотрудника = Сотрудники.Код\_Сотрудника " +

"AND Трудовой\_Договор.Код\_Отдела = Отдел.Код\_Отдела AND Трудовой\_Договор.Код\_Должности = Должность.Код\_Должности", connection18);

DataTable table18 = new DataTable();

dataAdapter18.Fill(table18);

dataGridView1.DataSource = table18;

dataGridView1.Columns[0].Visible = false;

connection18.Close();

break;

**Приложение 5. Код для печати трудового договора**

private void PrintButton\_Click(object sender, EventArgs e)

{

var nomertextBox = NomertextBox.Text;

var cityDetentiontextBox = CityDetentiontextBox.Text;

var dateСonclusiontextBox = DateСonclusiontextBox.Text;

var candidatestextBox = CandidatestextBox.Text;

var passportSeriestextBox = PassportSeriestextBox.Text;

var passportNumbertextBox = PassportNumbertextBox.Text;

var employeetextBox = EmployeetextBox.Text;

var departmenttextBox = DepartmenttextBox.Text;

var posttextBox = PosttextBox.Text;

var startDatetextBox = StartDatetextBox.Text;

var startTimetextBox = StartTimetextBox.Text;

var wordap = new Word.Application();

wordap.Visible = false;

var worddoc = wordap.Documents.Open(fail);

ReplaceWord("{nomertextBox}", nomertextBox, worddoc);

ReplaceWord("{cityDetentiontextBox}", cityDetentiontextBox, worddoc);

ReplaceWord("{dateСonclusiontextBox}", dateСonclusiontextBox, worddoc);

ReplaceWord("{candidatestextBox}", candidatestextBox, worddoc);

ReplaceWord("{passportSeriestextBox}", passportSeriestextBox, worddoc);

ReplaceWord("{passportNumbertextBox}", passportNumbertextBox, worddoc);

ReplaceWord("{employeetextBox}", employeetextBox, worddoc);

ReplaceWord("{departmenttextBox}", departmenttextBox, worddoc);

ReplaceWord("{posttextBox}", posttextBox, worddoc);

ReplaceWord("{startDatetextBox}", startDatetextBox, worddoc);

ReplaceWord("{startTimetextBox}", startTimetextBox, worddoc);

worddoc.SaveAs("C:/Users/mihal/Desktop/TrudovoyDogovor1.docx");

worddoc.Close();

}

private void ReplaceWord(string stubreplace, string text, Word.Document worddoc)

{

var range = worddoc.Content;

range.Find.ClearFormatting();

range.Find.Execute(FindText: stubreplace, ReplaceWith: text);

}

**Приложение 6. Описание структуры записей таблицы «Сотрудники»**

**Таблица 44**

**Описание структуры записей таблицы "Сотрудники"**

| **Наименование поля** | **Идентификатор поля** | **Тип поля** | **Длина поля** | **Прочее** |
| --- | --- | --- | --- | --- |
| Код\_Сотрудника | Id\_Employee | Int | 5 | Ключевое поле |
| Код\_Должности | Id\_Post | Int | 5 | Внешний ключ |
| Код\_Пользователя | Id\_User | Int | 5 | Внешний ключ |
| ФИО | Full\_Name | Varchar | 255 |  |
| Адрес | Address | Varchar | 255 |  |
| Телефон | Telephone | Varchar | 20 |  |

**Приложение 7. Описание структуры записей таблицы «Пользователи»**

**Таблица 45**

**Описание структуры записей таблицы "Пользователи"**

| **Наименование поля** | **Идентификатор поля** | **Тип поля** | **Длина поля** | **Прочее** |
| --- | --- | --- | --- | --- |
| Код\_Пользователя | Id\_User | Int | 5 | Ключевое поле |
| Логин | Login | Varchar | 50 |  |
| Пароль | Password | Varchar | 50 |  |

**Приложение 8. Описание структуры записей таблицы «Должность»**

**Таблица 46**

**Описание структуры записей таблицы "Должность"**

| **Наименование поля** | **Идентификатор поля** | **Тип поля** | **Длина поля** | **Прочее** |
| --- | --- | --- | --- | --- |
| Код\_Должности | Id\_Post | Int | 5 | Ключевое поле |
| Код\_Отдела | Id\_Department | Int | 5 | Внешний ключ |
| Название\_Должности | Title\_Post | Varchar | 50 |  |
| Оклад | Salary | int | 5 |  |

**Приложение 9. Описание структуры записей таблицы «Тип\_Собеседования»**

**Таблица 47**

**Описание структуры записей таблицы "Тип\_Собеседования"**

| **Наименование поля** | **Идентификатор поля** | **Тип поля** | **Длина поля** | **Прочее** |
| --- | --- | --- | --- | --- |
| Код\_Типа\_Собеседования | Id\_Type\_Interview | Int | 5 | Ключевое поле |
| Название\_Типа\_Собеседования | Title\_Type\_Interview | Varchar | 50 |  |

**Приложение 10. Описание структуры записей таблицы «Статус\_Собеседования»**

**Таблица 48**

**Описание структуры записей таблицы "Статус\_Собеседования"**

| **Наименование поля** | **Идентификатор поля** | **Тип поля** | **Длина поля** | **Прочее** |
| --- | --- | --- | --- | --- |
| Код\_Статуса | Id\_Status | Int | 5 | Ключевое поле |
| Название\_Статуса\_Собеседования | Title\_Status\_Interview | Varchar | 50 |  |

**Приложение 11. Описание структуры записей таблицы «Отдел»**

**Таблица 49**

**Описание структуры записей таблицы "Отдел"**

| **Наименование поля** | **Идентификатор поля** | **Тип поля** | **Длина поля** | **Прочее** |
| --- | --- | --- | --- | --- |
| Код\_Отдела | Id\_Department | Int | 5 | Ключевое поле |
| Название\_Отдела | Title\_Department | Varchar | 50 |  |

**Приложение 12. Описание структуры записей таблицы «Тестирование»**

**Таблица 50**

**Описание структуры записей таблицы "Тестирование"**

| **Наименование поля** | **Идентификатор поля** | **Тип поля** | **Длина поля** | **Прочее** |
| --- | --- | --- | --- | --- |
| Код\_Тестирования | Id\_Testing | Int | 5 | Ключевое поле |
| Код\_Кандидата | Id\_Candidate | Int | 5 | Внешний ключ |
| Код\_Теста | Id\_Test | Int | 5 | Внешний ключ |
| Дата\_Тестирования | Date\_Testing | Date | 10 |  |
| Время\_Тестирования | Time\_Testing | Time | 7 |  |
| Оценка\_За\_Тест | Score\_Test | Varchar | 20 |  |

**Приложение 13. Описание структуры записей таблицы «Тесты»**

**Таблица 51**

**Описание структуры записей таблицы "Тесты"**

| **Наименование поля** | **Идентификатор поля** | **Тип поля** | **Длина поля** | **Прочее** |
| --- | --- | --- | --- | --- |
| Код\_Теста | Id\_Test | Int | 5 | Ключевое поле |
| Код\_Должности | Id\_Post | Int | 5 | Внешний ключ |
| Набор\_Проверяемых\_компетенций | Competencies\_Tested | Varchar | MAX |  |
| Примечания | Notes | Varchar | 255 |  |

**Приложение 14. Код основного модуля**

Код с SQL-запросами для выбора кандидатов определённой должности:

private void loadString()

{

switch (Properties.Settings.Default.l)

{

case 1:

string connectionString = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection = new SqlConnection(connectionString);

connection.Open();

SqlDataAdapter dataAdapter = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 3 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection);

DataTable table = new DataTable();

dataAdapter.Fill(table);

connection.Close();

FIOtextBox.Text = table.Rows[0][1].ToString();

byte[] photoBytes = (byte[])table.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table.Rows[0][3].ToString();

TelephonetextBox.Text = table.Rows[0][4].ToString();

PassportSeriestextBox.Text = table.Rows[0][5].ToString();

PassportNumbertextBox.Text = table.Rows[0][6].ToString();

AgetextBox.Text = table.Rows[0][7].ToString();

ScoretextBox.Text = table.Rows[0][8].ToString();

Id\_Post.Text = table.Rows[0][9].ToString();

break;

case 2:

string connectionString1 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection1 = new SqlConnection(connectionString1);

connection1.Open();

SqlDataAdapter dataAdapter1 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 4 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection1);

DataTable table1 = new DataTable();

dataAdapter1.Fill(table1);

connection1.Close();

FIOtextBox.Text = table1.Rows[0][1].ToString();

byte[] photoBytes1 = (byte[])table1.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes1))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table1.Rows[0][3].ToString();

TelephonetextBox.Text = table1.Rows[0][4].ToString();

PassportSeriestextBox.Text = table1.Rows[0][5].ToString();

PassportNumbertextBox.Text = table1.Rows[0][6].ToString();

AgetextBox.Text = table1.Rows[0][7].ToString();

ScoretextBox.Text = table1.Rows[0][8].ToString();

Id\_Post.Text = table1.Rows[0][9].ToString();

break;

case 3:

string connectionString2 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection2 = new SqlConnection(connectionString2);

connection2.Open();

SqlDataAdapter dataAdapter2 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 5 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection2);

DataTable table2 = new DataTable();

dataAdapter2.Fill(table2);

connection2.Close();

FIOtextBox.Text = table2.Rows[0][1].ToString();

byte[] photoBytes2 = (byte[])table2.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes2))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table2.Rows[0][3].ToString();

TelephonetextBox.Text = table2.Rows[0][4].ToString();

PassportSeriestextBox.Text = table2.Rows[0][5].ToString();

PassportNumbertextBox.Text = table2.Rows[0][6].ToString();

AgetextBox.Text = table2.Rows[0][7].ToString();

ScoretextBox.Text = table2.Rows[0][8].ToString();

Id\_Post.Text = table2.Rows[0][9].ToString();

break;

case 4:

string connectionString3 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection3 = new SqlConnection(connectionString3);

connection3.Open();

SqlDataAdapter dataAdapter3 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 9 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection3);

DataTable table3 = new DataTable();

dataAdapter3.Fill(table3);

connection3.Close();

FIOtextBox.Text = table3.Rows[0][1].ToString();

byte[] photoBytes3 = (byte[])table3.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes3))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table3.Rows[0][3].ToString();

TelephonetextBox.Text = table3.Rows[0][4].ToString();

PassportSeriestextBox.Text = table3.Rows[0][5].ToString();

PassportNumbertextBox.Text = table3.Rows[0][6].ToString();

AgetextBox.Text = table3.Rows[0][7].ToString();

ScoretextBox.Text = table3.Rows[0][8].ToString();

Id\_Post.Text = table3.Rows[0][9].ToString();

break;

case 5:

string connectionString4 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection4 = new SqlConnection(connectionString4);

connection4.Open();

SqlDataAdapter dataAdapter4 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 8 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection4);

DataTable table4 = new DataTable();

dataAdapter4.Fill(table4);

connection4.Close();

FIOtextBox.Text = table4.Rows[0][1].ToString();

byte[] photoBytes4 = (byte[])table4.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes4))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table4.Rows[0][3].ToString();

TelephonetextBox.Text = table4.Rows[0][4].ToString();

PassportSeriestextBox.Text = table4.Rows[0][5].ToString();

PassportNumbertextBox.Text = table4.Rows[0][6].ToString();

AgetextBox.Text = table4.Rows[0][7].ToString();

ScoretextBox.Text = table4.Rows[0][8].ToString();

Id\_Post.Text = table4.Rows[0][9].ToString();

break;

case 6:

string connectionString5 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection5 = new SqlConnection(connectionString5);

connection5.Open();

SqlDataAdapter dataAdapter5 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 23 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection5);

DataTable table5 = new DataTable();

dataAdapter5.Fill(table5);

connection5.Close();

FIOtextBox.Text = table5.Rows[0][1].ToString();

byte[] photoBytes5 = (byte[])table5.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes5))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table5.Rows[0][3].ToString();

TelephonetextBox.Text = table5.Rows[0][4].ToString();

PassportSeriestextBox.Text = table5.Rows[0][5].ToString();

PassportNumbertextBox.Text = table5.Rows[0][6].ToString();

AgetextBox.Text = table5.Rows[0][7].ToString();

ScoretextBox.Text = table5.Rows[0][8].ToString();

Id\_Post.Text = table5.Rows[0][9].ToString();

break;

case 7:

string connectionString6 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection6 = new SqlConnection(connectionString6);

connection6.Open();

SqlDataAdapter dataAdapter6 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 24 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection6);

DataTable table6 = new DataTable();

dataAdapter6.Fill(table6);

connection6.Close();

FIOtextBox.Text = table6.Rows[0][1].ToString();

byte[] photoBytes6 = (byte[])table6.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes6))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table6.Rows[0][3].ToString();

TelephonetextBox.Text = table6.Rows[0][4].ToString();

PassportSeriestextBox.Text = table6.Rows[0][5].ToString();

PassportNumbertextBox.Text = table6.Rows[0][6].ToString();

AgetextBox.Text = table6.Rows[0][7].ToString();

ScoretextBox.Text = table6.Rows[0][8].ToString();

Id\_Post.Text = table6.Rows[0][9].ToString();

break;

case 8:

string connectionString7 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection7 = new SqlConnection(connectionString7);

connection7.Open();

SqlDataAdapter dataAdapter7 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 26 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection7);

DataTable table7 = new DataTable();

dataAdapter7.Fill(table7);

connection7.Close();

FIOtextBox.Text = table7.Rows[0][1].ToString();

byte[] photoBytes7 = (byte[])table7.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes7))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table7.Rows[0][3].ToString();

TelephonetextBox.Text = table7.Rows[0][4].ToString();

PassportSeriestextBox.Text = table7.Rows[0][5].ToString();

PassportNumbertextBox.Text = table7.Rows[0][6].ToString();

AgetextBox.Text = table7.Rows[0][7].ToString();

ScoretextBox.Text = table7.Rows[0][8].ToString();

Id\_Post.Text = table7.Rows[0][9].ToString();

break;

case 9:

string connectionString8 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection8 = new SqlConnection(connectionString8);

connection8.Open();

SqlDataAdapter dataAdapter8 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 10 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection8);

DataTable table8 = new DataTable();

dataAdapter8.Fill(table8);

connection8.Close();

FIOtextBox.Text = table8.Rows[0][1].ToString();

byte[] photoBytes8 = (byte[])table8.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes8))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table8.Rows[0][3].ToString();

TelephonetextBox.Text = table8.Rows[0][4].ToString();

PassportSeriestextBox.Text = table8.Rows[0][5].ToString();

PassportNumbertextBox.Text = table8.Rows[0][6].ToString();

AgetextBox.Text = table8.Rows[0][7].ToString();

ScoretextBox.Text = table8.Rows[0][8].ToString();

Id\_Post.Text = table8.Rows[0][9].ToString();

break;

case 10:

string connectionString9 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection9 = new SqlConnection(connectionString9);

connection9.Open();

SqlDataAdapter dataAdapter9 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 11 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection9);

DataTable table9 = new DataTable();

dataAdapter9.Fill(table9);

connection9.Close();

FIOtextBox.Text = table9.Rows[0][1].ToString();

byte[] photoBytes9 = (byte[])table9.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes9))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table9.Rows[0][3].ToString();

TelephonetextBox.Text = table9.Rows[0][4].ToString();

PassportSeriestextBox.Text = table9.Rows[0][5].ToString();

PassportNumbertextBox.Text = table9.Rows[0][6].ToString();

AgetextBox.Text = table9.Rows[0][7].ToString();

ScoretextBox.Text = table9.Rows[0][8].ToString();

Id\_Post.Text = table9.Rows[0][9].ToString();

break;

case 11:

string connectionString10 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection10 = new SqlConnection(connectionString10);

connection10.Open();

SqlDataAdapter dataAdapter10 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 12 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection10);

DataTable table10 = new DataTable();

dataAdapter10.Fill(table10);

connection10.Close();

FIOtextBox.Text = table10.Rows[0][1].ToString();

byte[] photoBytes10 = (byte[])table10.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes10))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table10.Rows[0][3].ToString();

TelephonetextBox.Text = table10.Rows[0][4].ToString();

PassportSeriestextBox.Text = table10.Rows[0][5].ToString();

PassportNumbertextBox.Text = table10.Rows[0][6].ToString();

AgetextBox.Text = table10.Rows[0][7].ToString();

ScoretextBox.Text = table10.Rows[0][8].ToString();

Id\_Post.Text = table10.Rows[0][9].ToString();

break;

case 12:

string connectionString11 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection11 = new SqlConnection(connectionString11);

connection11.Open();

SqlDataAdapter dataAdapter11 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 13 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection11);

DataTable table11 = new DataTable();

dataAdapter11.Fill(table11);

connection11.Close();

FIOtextBox.Text = table11.Rows[0][1].ToString();

byte[] photoBytes11 = (byte[])table11.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes11))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table11.Rows[0][3].ToString();

TelephonetextBox.Text = table11.Rows[0][4].ToString();

PassportSeriestextBox.Text = table11.Rows[0][5].ToString();

PassportNumbertextBox.Text = table11.Rows[0][6].ToString();

AgetextBox.Text = table11.Rows[0][7].ToString();

ScoretextBox.Text = table11.Rows[0][8].ToString();

Id\_Post.Text = table11.Rows[0][9].ToString();

break;

case 13:

string connectionString12 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection12 = new SqlConnection(connectionString12);

connection12.Open();

SqlDataAdapter dataAdapter12 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 14 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection12);

DataTable table12 = new DataTable();

dataAdapter12.Fill(table12);

connection12.Close();

FIOtextBox.Text = table12.Rows[0][1].ToString();

byte[] photoBytes12 = (byte[])table12.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes12))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table12.Rows[0][3].ToString();

TelephonetextBox.Text = table12.Rows[0][4].ToString();

PassportSeriestextBox.Text = table12.Rows[0][5].ToString();

PassportNumbertextBox.Text = table12.Rows[0][6].ToString();

AgetextBox.Text = table12.Rows[0][7].ToString();

ScoretextBox.Text = table12.Rows[0][8].ToString();

Id\_Post.Text = table12.Rows[0][9].ToString();

break;

case 14:

string connectionString13 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection13 = new SqlConnection(connectionString13);

connection13.Open();

SqlDataAdapter dataAdapter13 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 15 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection13);

DataTable table13 = new DataTable();

dataAdapter13.Fill(table13);

connection13.Close();

FIOtextBox.Text = table13.Rows[0][1].ToString();

byte[] photoBytes13 = (byte[])table13.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes13))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table13.Rows[0][3].ToString();

TelephonetextBox.Text = table13.Rows[0][4].ToString();

PassportSeriestextBox.Text = table13.Rows[0][5].ToString();

PassportNumbertextBox.Text = table13.Rows[0][6].ToString();

AgetextBox.Text = table13.Rows[0][7].ToString();

ScoretextBox.Text = table13.Rows[0][8].ToString();

Id\_Post.Text = table13.Rows[0][9].ToString();

break;

case 15:

string connectionString14 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection14 = new SqlConnection(connectionString14);

connection14.Open();

SqlDataAdapter dataAdapter14 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 16 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection14);

DataTable table14 = new DataTable();

dataAdapter14.Fill(table14);

connection14.Close();

FIOtextBox.Text = table14.Rows[0][1].ToString();

byte[] photoBytes14 = (byte[])table14.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes14))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table14.Rows[0][3].ToString();

TelephonetextBox.Text = table14.Rows[0][4].ToString();

PassportSeriestextBox.Text = table14.Rows[0][5].ToString();

PassportNumbertextBox.Text = table14.Rows[0][6].ToString();

AgetextBox.Text = table14.Rows[0][7].ToString();

ScoretextBox.Text = table14.Rows[0][8].ToString();

Id\_Post.Text = table14.Rows[0][9].ToString();

break;

case 16:

string connectionString15 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection15 = new SqlConnection(connectionString15);

connection15.Open();

SqlDataAdapter dataAdapter15 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 17 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection15);

DataTable table15 = new DataTable();

dataAdapter15.Fill(table15);

connection15.Close();

FIOtextBox.Text = table15.Rows[0][1].ToString();

byte[] photoBytes15 = (byte[])table15.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes15))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table15.Rows[0][3].ToString();

TelephonetextBox.Text = table15.Rows[0][4].ToString();

PassportSeriestextBox.Text = table15.Rows[0][5].ToString();

PassportNumbertextBox.Text = table15.Rows[0][6].ToString();

AgetextBox.Text = table15.Rows[0][7].ToString();

ScoretextBox.Text = table15.Rows[0][8].ToString();

Id\_Post.Text = table15.Rows[0][9].ToString();

break;

case 17:

string connectionString16 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection16 = new SqlConnection(connectionString16);

connection16.Open();

SqlDataAdapter dataAdapter16 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 18 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection16);

DataTable table16 = new DataTable();

dataAdapter16.Fill(table16);

connection16.Close();

FIOtextBox.Text = table16.Rows[0][1].ToString();

byte[] photoBytes16 = (byte[])table16.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes16))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table16.Rows[0][3].ToString();

TelephonetextBox.Text = table16.Rows[0][4].ToString();

PassportSeriestextBox.Text = table16.Rows[0][5].ToString();

PassportNumbertextBox.Text = table16.Rows[0][6].ToString();

AgetextBox.Text = table16.Rows[0][7].ToString();

ScoretextBox.Text = table16.Rows[0][8].ToString();

Id\_Post.Text = table16.Rows[0][9].ToString();

break;

case 18:

string connectionString17 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection17 = new SqlConnection(connectionString17);

connection17.Open();

SqlDataAdapter dataAdapter17 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 19 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection17);

DataTable table17 = new DataTable();

dataAdapter17.Fill(table17);

connection17.Close();

FIOtextBox.Text = table17.Rows[0][1].ToString();

byte[] photoBytes17 = (byte[])table17.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes17))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table17.Rows[0][3].ToString();

TelephonetextBox.Text = table17.Rows[0][4].ToString();

PassportSeriestextBox.Text = table17.Rows[0][5].ToString();

PassportNumbertextBox.Text = table17.Rows[0][6].ToString();

AgetextBox.Text = table17.Rows[0][7].ToString();

ScoretextBox.Text = table17.Rows[0][8].ToString();

Id\_Post.Text = table17.Rows[0][9].ToString();

break;

case 19:

string connectionString18 = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection18 = new SqlConnection(connectionString18);

connection18.Open();

SqlDataAdapter dataAdapter18 = new SqlDataAdapter

("SELECT Кандидаты.Код\_Кандидата, Кандидаты.ФИО, Кандидаты.Фото, Кандидаты.Почта, Кандидаты.Телефон, Кандидаты.Серия\_Паспорта, Кандидаты.Номер\_Паспорта, Кандидаты.Возраст, " +

"Тестирование.Оценка\_За\_Тест, Должность.Название\_Должности " +

"FROM Кандидаты, Должность, Тестирование " +

"WHERE Кандидаты.Код\_Должности = Должность.Код\_Должности AND Должность.Код\_Должности LIKE 21 AND Кандидаты.Код\_Тестирования = Тестирование.Код\_Тестирования AND Кандидаты.Код\_Кандидата =" + id, connection18);

DataTable table18 = new DataTable();

dataAdapter18.Fill(table18);

connection18.Close();

FIOtextBox.Text = table18.Rows[0][1].ToString();

byte[] photoBytes18 = (byte[])table18.Rows[0][2];

using (MemoryStream ms = new MemoryStream(photoBytes18))

{

Image photo = Image.FromStream(ms);

PhotopictureBox.Image = photo;

}

MailtextBox.Text = table18.Rows[0][3].ToString();

TelephonetextBox.Text = table18.Rows[0][4].ToString();

PassportSeriestextBox.Text = table18.Rows[0][5].ToString();

PassportNumbertextBox.Text = table18.Rows[0][6].ToString();

AgetextBox.Text = table18.Rows[0][7].ToString();

ScoretextBox.Text = table18.Rows[0][8].ToString();

Id\_Post.Text = table18.Rows[0][9].ToString();

break;

}

}

Код для кнопки «Назначить», при нажатии на которую назначается собеседование:

private void Button\_Appoint\_Click(object sender, EventArgs e)

{

if (InterviewStatuscomboBox.SelectedValue != null && int.Parse(InterviewStatuscomboBox.SelectedValue.ToString()) == 3)

{

string connectionString = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection = new SqlConnection(connectionString);

connection.Open();

SqlDataAdapter dataAdapter = new SqlDataAdapter

("INSERT INTO Собеседования (Код\_Кандидата, Дата\_Собеседования, Время\_Собеседования, Код\_Типа\_Собеседования, Код\_Сотрудника, Код\_Статуса) " +

" VALUES ('" + CandidatescomboBox.SelectedValue + "'," +

"'" + InterviewdatePicker.Value.ToString("yyyy-MM-dd") + "'," +

"'" + InterviewtimePicker.Value.ToString("HH:mm") + "'," +

"'" + TypeOfInterviewcomboBox.SelectedValue + "'," +

"'" + EmployeecomboBox.SelectedValue + "'," +

"'" + InterviewStatuscomboBox.SelectedValue + "')", connection);

DataTable table = new DataTable();

dataAdapter.Fill(table);

connection.Close();

Close();

}

else

{

MessageBox.Show("Для нажатия на эту кнопку необходимо выбрать статус 'Назначено'.", "Ошибка");

}

}

Код для кнопки «Отклонить», при нажатии на которую собеседование отклоняется:

private void Button\_Reject\_Click(object sender, EventArgs e)

{

if (InterviewStatuscomboBox.SelectedValue != null && int.Parse(InterviewStatuscomboBox.SelectedValue.ToString()) == 2)

{

string connectionString = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection = new SqlConnection(connectionString);

connection.Open();

SqlDataAdapter dataAdapter = new SqlDataAdapter

("INSERT INTO Собеседования (Код\_Кандидата, Дата\_Собеседования, Время\_Собеседования, Код\_Типа\_Собеседования, Код\_Сотрудника, Код\_Статуса) " +

" VALUES ('" + CandidatescomboBox.SelectedValue + "'," +

"'" + InterviewdatePicker.Value.ToString("yyyy-MM-dd") + "'," +

"'" + InterviewtimePicker.Value.ToString("HH:mm") + "'," +

"'" + TypeOfInterviewcomboBox.SelectedValue + "'," +

"'" + EmployeecomboBox.SelectedValue + "'," +

"'" + InterviewStatuscomboBox.SelectedValue + "')", connection);

DataTable table = new DataTable();

dataAdapter.Fill(table);

connection.Close();

Close();

}

else

{

MessageBox.Show("Для нажатия на эту кнопку необходимо выбрать статус 'Отклонено'.", "Ошибка");

}

}

Код для кнопки «Собеседование пройдено»:

private void AcceptButton\_Click(object sender, EventArgs e)

{

string connectionString = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection = new SqlConnection(connectionString);

connection.Open();

SqlDataAdapter dataAdapter = new SqlDataAdapter

("UPDATE Собеседования SET Код\_Статуса = 1 " +

"WHERE Код\_Собеседования = " + id, connection);

DataTable table = new DataTable();

dataAdapter.Fill(table);

connection.Close();

Close();

}

Код для кнопки «Собеседование не пройдено»:

private void RejectButton\_Click(object sender, EventArgs e)

{

string connectionString = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection = new SqlConnection(connectionString);

connection.Open();

SqlDataAdapter dataAdapter = new SqlDataAdapter

("UPDATE Собеседования SET Код\_Статуса = 2 " +

"WHERE Код\_Собеседования = " + id, connection);

DataTable table = new DataTable();

dataAdapter.Fill(table);

connection.Close();

Close();

}

Код для кнопки «Удалить собеседование»:

private void DeleteButton\_Click(object sender, EventArgs e)

{

string connectionString = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection = new SqlConnection(connectionString);

connection.Open();

SqlDataAdapter dataAdapter = new SqlDataAdapter

("DELETE FROM Собеседования WHERE Код\_Собеседования = " + id, connection);

DataTable table = new DataTable();

dataAdapter.Fill(table);

connection.Close();

Close();

}

Код для кнопки «Заключить договор»:

private void ConcludeButton\_Click(object sender, EventArgs e)

{

string connectionString = "Data Source=USER\\SQLEXPRESS;Initial Catalog=PodborPersonala;Integrated Security=True";

SqlConnection connection = new SqlConnection(connectionString);

connection.Open();

SqlDataAdapter dataAdapter = new SqlDataAdapter

("INSERT INTO Трудовой\_Договор (Дата\_Подписания\_Договора, Город\_Заключения\_Договора, Код\_Кандидата, Код\_Сотрудника, Код\_Отдела, Код\_Должности, Дата\_Начала\_Работы, Время\_Начала\_Работы) " +

" VALUES ('" + DateСonclusionPicker.Value.ToString("yyyy-MM-dd") + "'," +

"'" + CityDetentiontextBox.Text + "'," +

"'" + CandidatescomboBox.SelectedValue + "'," +

"'" + EmployeecomboBox.SelectedValue + "'," +

"'" + DepartmentcomboBox.SelectedValue + "'," +

"'" + PostcomboBox.SelectedValue + "'," +

"'" + StartDatePicker.Value.ToString("yyyy-MM-dd") + "'," +

"'" + StartTimePicker.Value.ToString("HH:mm") + "')", connection);

DataTable table = new DataTable();

dataAdapter.Fill(table);

connection.Close();

Close();

}

Код для кнопки «Печать», при нажатии на которую печатается трудовой договор:

private void PrintButton\_Click(object sender, EventArgs e)

{

var nomertextBox = NomertextBox.Text;

var cityDetentiontextBox = CityDetentiontextBox.Text;

var dateСonclusiontextBox = DateСonclusiontextBox.Text;

var candidatestextBox = CandidatestextBox.Text;

var passportSeriestextBox = PassportSeriestextBox.Text;

var passportNumbertextBox = PassportNumbertextBox.Text;

var employeetextBox = EmployeetextBox.Text;

var departmenttextBox = DepartmenttextBox.Text;

var posttextBox = PosttextBox.Text;

var startDatetextBox = StartDatetextBox.Text;

var startTimetextBox = StartTimetextBox.Text;

var wordap = new Word.Application();

wordap.Visible = false;

var worddoc = wordap.Documents.Open(fail);

ReplaceWord("{nomertextBox}", nomertextBox, worddoc);

ReplaceWord("{cityDetentiontextBox}", cityDetentiontextBox, worddoc);

ReplaceWord("{dateСonclusiontextBox}", dateСonclusiontextBox, worddoc);

ReplaceWord("{candidatestextBox}", candidatestextBox, worddoc);

ReplaceWord("{passportSeriestextBox}", passportSeriestextBox, worddoc);

ReplaceWord("{passportNumbertextBox}", passportNumbertextBox, worddoc);

ReplaceWord("{employeetextBox}", employeetextBox, worddoc);

ReplaceWord("{departmenttextBox}", departmenttextBox, worddoc);

ReplaceWord("{posttextBox}", posttextBox, worddoc);

ReplaceWord("{startDatetextBox}", startDatetextBox, worddoc);

ReplaceWord("{startTimetextBox}", startTimetextBox, worddoc);

worddoc.SaveAs("C:/Users/mihal/Desktop/TrudovoyDogovor1.docx");

worddoc.Close();

}

private void ReplaceWord(string stubreplace, string text, Word.Document worddoc)

{

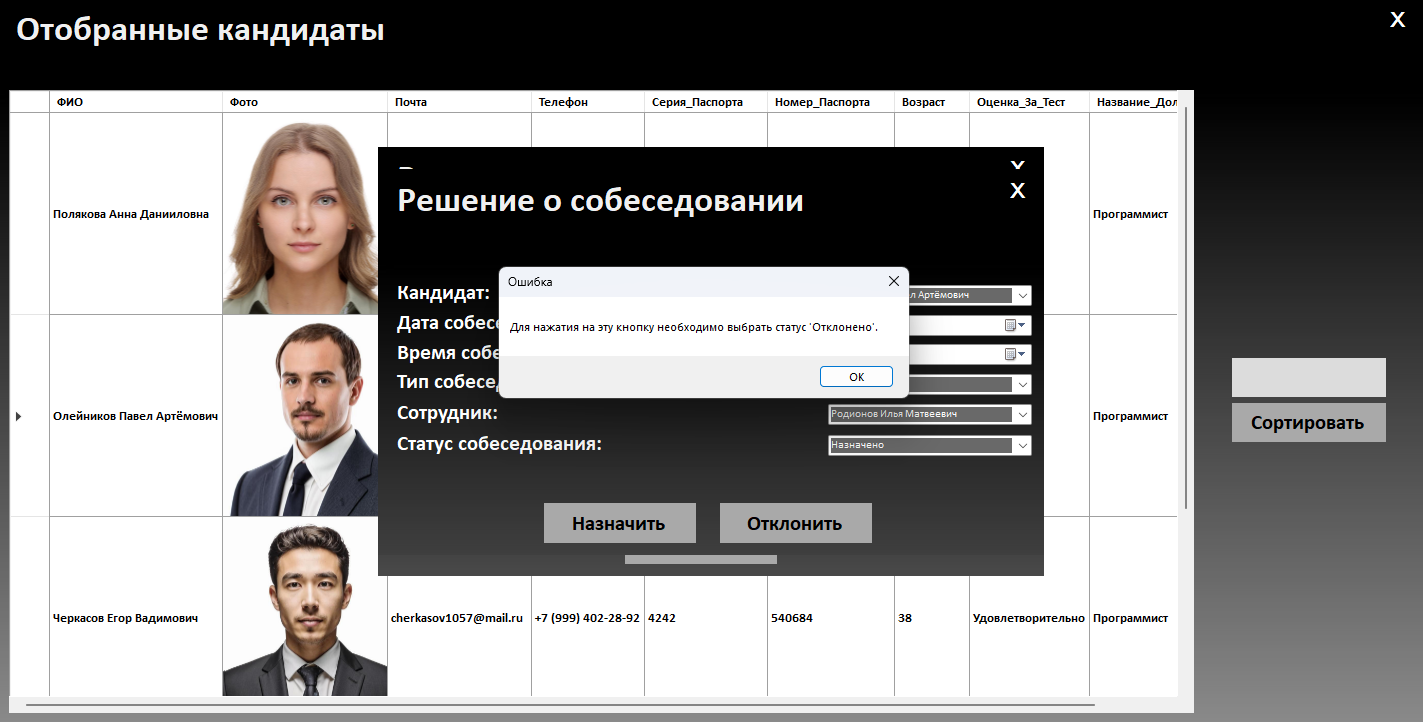
var range = worddoc.Content;

range.Find.ClearFormatting();

range.Find.Execute(FindText: stubreplace, ReplaceWith: text);

}

**Приложение 15. Нажатие на кнопку «Отклонить» при статусе «Назначено»**



Дипломный проект выполнен мной совершенно самостоятельно. Все использованные в проекте материалы и концепции из опубликованной литературы и других источников имеют ссылки на них. Дипломный проект прошел проверку на корректность заимствования в системе «Антиплагиат.ВУЗ».

Настоящим подтверждаю, что даю разрешение Университету «Синергия» на размещение полного текста моего дипломного проекта, отзыва на дипломный проект в электронно-библиотечной системе Университета «Синергия».

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Плевако М.П. |
|  | подпись |  | Фамилия ИО |

«11» \_Июня\_ 2024 г.