



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«МИРЭА – Российский технологический университет»  
РТУ МИРЭА**

Институт информационных технологий

Кафедра математического обеспечения и стандартизации информационных  
технологий

## **КУРСОВАЯ РАБОТА**

по дисциплине «Обоснование и разработка требований к программным  
системам»

**Тема курсовой работы: «Обоснование и разработка требований к программной  
системе поддержки деятельности агентства по подбору персонала»**

**Студент группы ИКБО-63-23**

Оганнисян Нарек Геворгович

(подпись студента)

**Руководитель курсовой работы**

доцент, к.т.н. Бирюкова А.А.

(подпись руководителя)

Работа представлена к защите

«\_\_\_\_» 2025 г.

Допущен к защите

«\_\_\_\_» 2025 г.

Москва 2025



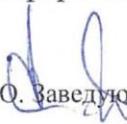
## МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»  
**РТУ МИРЭА**

Институт информационных технологий

Кафедра математического обеспечения и стандартизации информационных технологий

  
Утверждаю  
И.О. Заведующего кафедрой МОСИТ  
Головин С.А.

«8» сентября 2025 г.

### **ЗАДАНИЕ на выполнение курсовой работы по дисциплине**

«Обоснование и разработка требований к программным системам»

Студент Оганисян Нарек Геворгович

Группа ИКБО-63-23

Тема «Обоснование и разработка требований к программной системе поддержки деятельности агентства по подбору персонала»

**Исходные данные:** учебный материал по дисциплине, выбранная студентом тема курсовой работы

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:**

Обоснование необходимости разработки ПС. Определение и документирование бизнес-потребностей (построение диаграмм в состоянии «as is» и «to be», построение дерева целей). Определение и документирование пользовательских требований (построение диаграммы вариантов использования в нотации UML, составление пользовательских историй). Определение и документирование функциональных требований (построение диаграмм в нотации UML: взаимодействия, классов анализа, состояния и деятельности, классов уровня проектирования и реализации, компонентов и развертывания системы). Определение и документирование нефункциональных требований.

**Срок представления к защите курсовой работы:**

до «15» декабря 2025 г.

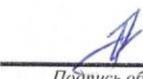
Задание на курсовую работу  
выдал

  
Подпись руководителя

Бирюкова А.А.  
(ФИО руководителя)

«8» сентября 2025 г.

Задание на курсовую работу  
получил

  
Подпись обучающегося

Оганисян Н.Г.  
(ФИО обучающегося)

«8» сентября 2025 г.

## Содержание

Введение.....	6
Глава 1. Теоретические основы разработки требований к программной системе .....	8
1.1. Требования к программной системе и обоснование их разработки .....	8
1.2. Пользовательские требования .....	9
1.3. Функциональные и нефункциональные требования.....	10
Глава 2. Предпроектное исследование и обоснование необходимости разработки ПС .....	12
2.1. Анализ предметной области .....	12
Описание текущего положения.....	12
Организационная структура .....	13
Классы пользователей .....	14
2.2. Сравнительный анализ существующих решений.....	14
Критерии сравнения .....	14
Сравнительная таблица .....	15
Обоснование актуальности разработки .....	15
2.3. Моделирование бизнес-процессов организации .....	16
Модель «Как есть» (As-Is) .....	16
Определение бизнес-потребностей.....	18
Дерево целей.....	19
Морфологическая карта .....	19
Модель «Как должно быть» (To-Be).....	20
2.4. Концепции продукта и границы проекта .....	24
Глава 3. Разработка требований к программной системе .....	25

3.1. Определение и документирование пользовательских требований.....	25
Диаграмма вариантов использования .....	25
Пользовательские истории.....	26
1. Блок: Управление базой кандидатов (UC: Управление базой кандидатов, Парсинг и добавление резюме, Детальный поиск).....	26
2. Блок: Управление вакансиями и отбор (UC: Управление заявками и вакансиями, Регистрация этапов подбора, Просмотр и отклик на вакансию).....	27
3. Блок: Отчетность и мониторинг (UC: Формирование сводных отчетов, Мониторинг ключевых показателей).....	28
4. Блок: Взаимодействие с Работодателем (UC: Управление заявками и вакансиями, Формирование сводных отчетов).....	28
5. Блок: Администрирование (UC: Управление пользователями и правами) .....	29
3.2. Определение и документирование функциональных требований .....	30
3.2.1. Диаграмма классов анализа .....	30
3.2.2. Диаграмма последовательности .....	31
3.2.3. Диаграмма коммуникации .....	32
3.2.4. Диаграммы деятельности .....	32
3.2.5. Диаграмма классов уровня проектирования.....	34
3.2.6. Диаграмма «сущность-связь» (ERD) .....	34
3.2.7. Диаграмма состояний .....	35
3.3. Определение и документирование нефункциональных требований ....	36
3.3.1. Пользовательский интерфейс и карта окон .....	36
3.3.2. Диаграмма компонентов .....	39
3.3.3. Диаграмма развертывания .....	40

3.3.4. Коммуникационные интерфейсы.....	41
Заключение .....	42
Список использованной литературы.....	44
Приложения .....	45
Приложение 1. Документ о концепции и границах.....	45
Приложение 2. Атрибуты качества .....	49

## **Введение**

### **Актуальность темы.**

В современных условиях эффективность работы кадровых агентств напрямую зависит от скорости обработки информации и качества коммуникации с участниками рынка труда. Ежедневно рекрутеры сталкиваются с необходимостью обработки сотен резюме, ведения баз данных кандидатов и согласования встреч с работодателями.

Несмотря на развитие информационных технологий, многие процессы в агентствах по подбору персонала до сих пор выполняются вручную или с использованием разрозненных инструментов (электронные таблицы, мессенджеры, бумажные носители). Недостаточный объем систематизированной информации, дублирование данных, ошибки при ручном вводе и отсутствие единого информационного пространства приводят к потере кандидатов, увеличению времени закрытия вакансий и снижению конкурентоспособности агентства.

В связи с этим возникает острая необходимость в разработке требований к специализированной программной системе, которая позволит автоматизировать рутинные операции, обеспечить централизованное хранение данных и повысить прозрачность бизнес-процессов. Грамотно разработанные требования являются фундаментом успешного ИТ-проекта, позволяя минимизировать риски и создать продукт, полностью соответствующий ожиданиям бизнеса.

**Объект исследования** — деятельность кадрового агентства «TalentBridge Agency».

**Предмет исследования** — процессы подбора персонала (рекрутинга) и требования к автоматизированной системе «TalentBridge», обеспечивающей их поддержку.

**Цель работы** — обоснование необходимости создания и разработка полного комплекса требований (пользовательских, функциональных и

нефункциональных) к программной системе поддержки деятельности агентства по подбору персонала «TalentBridge».

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить теоретические основы разработки и управления требованиями к программному обеспечению.

2. Провести предпроектное исследование предметной области: проанализировать текущие бизнес-процессы агентства («как есть») и выявить их недостатки.

3. Обосновать необходимость разработки новой системы и смоделировать целевые бизнес-процессы («как должно быть»).

4. Определить и задокументировать пользовательские требования, разработав диаграммы вариантов использования и пользовательские истории.

5. Разработать функциональные требования, спроектировав архитектуру системы с помощью диаграмм классов, последовательности, деятельности и модели «сущность-связь».

6. Сформулировать нефункциональные требования к системе, включая требования к интерфейсам, безопасности и надежности.

7. Разработать техническое задание (ТЗ) на создание системы.

**Методы исследования.** В работе использовались методы системного анализа, моделирование бизнес-процессов в нотации IDEF0, объектно-ориентированное проектирование и моделирование требований с использованием языка UML (Unified Modeling Language), а также проектирование баз данных (ER-моделирование).

# **Глава 1. Теоретические основы разработки требований к программной системе**

## **1.1. Требования к программной системе и обоснование их разработки**

Разработка программного обеспечения (ПО) — это сложный инженерный процесс, успех которого во многом зависит от того, насколько точно и полно были определены цели и задачи будущей системы. Согласно статистике IT-индустрии, одной из главных причин провала проектов или превышения бюджета является некачественная работа с требованиями на ранних этапах.

**Требование** — это условие или возможность, которой должна обладать система, чтобы удовлетворять потребностям пользователя, либо условие, которому система должна соответствовать, чтобы удовлетворять контракту, стандарту или спецификации [1].

Разработка требований (Requirements Engineering) включает в себя несколько ключевых этапов:

**1. Выявление (Elicitation):** Сбор информации от заинтересованных лиц (стейкхолдеров) через интервью, анкетирование, анализ нормативной документации и наблюдение за рабочими процессами.

**2. Анализ (Analysis):** Структурирование полученной информации, поиск противоречий, определение приоритетов и моделирование процессов.

**3. Документирование (Specification):** Фиксация требований в виде спецификаций (SRS — Software Requirements Specification), моделей и диаграмм.

**4. Утверждение (Validation):** Проверка требований на полноту, корректность и реализуемость, согласование с заказчиком.

Классификация требований обычно проводится по трем уровням [2]:

**1. Бизнес-требования:** Описывают высокоуровневые цели организации (зачем нужна система? какую выгоду она принесет?).

**2. Пользовательские требования:** Описывают задачи, которые пользователи смогут выполнять с помощью системы (что пользователь хочет сделать?).

**3. Функциональные требования:** Описывают поведение системы (что система должна делать, какие вычисления производить, как реагировать на ввод данных).

Обоснование разработки требований заключается в необходимости создания единого видения продукта у заказчика и разработчиков. Это позволяет избежать дорогостоящих переделок на этапе программирования и тестирования.

## **1.2. Пользовательские требования**

Пользовательские требования (User Requirements) описывают цели и задачи, которые пользователи должны иметь возможность решать с помощью создаваемой системы. Они служат мостом между бизнес-потребностями и технической реализацией.

Для описания и моделирования пользовательских требований в современной практике чаще всего используются следующие инструменты:

**1. Диаграммы вариантов использования (Use Case Diagrams) в нотации UML.** Это графическое представление функциональности системы. Основными элементами являются:

Актеры (Actors): Роли пользователей (например, «Рекрутер», «Администратор») или внешние системы.

Варианты использования (Use Cases): Овалы, обозначающие конкретную функцию или цель (например, «Создать вакансию», «Найти кандидата»).

Связи: Отношения между актерами и вариантами использования (ассоциация, включение, расширение).

**2. Пользовательские истории (User Stories).** Это краткие описания требований, сформулированные на естественном языке, понятном заказчику. Стандартный шаблон выглядит так:

«Как <роль>, я хочу <действие/цель>, чтобы <ценность/результат>».

Пример: «Как рекрутер, я хочу автоматически загружать резюме из файлов, чтобы не тратить время на ручной ввод данных».

Документирование пользовательских требований позволяет четко определить границы проекта и гарантировать, что система будет удобной и полезной для конечных пользователей.

### **1.3. Функциональные и нефункциональные требования**

После определения того, что нужно пользователю, необходимо детально описать, как система будет это реализовывать. Для этого разрабатываются функциональные и нефункциональные требования.

Функциональные требования (Functional Requirements) определяют конкретные действия системы: ввод данных, обработку, вычисления, формирование отчетов, передачу данных.

Для моделирования функциональных требований используются различные виды диаграмм UML [3]:

1. Диаграмма классов (Class Diagram): Показывает статическую структуру системы, классы, их атрибуты, методы и взаимосвязи.

2. Диаграмма последовательности (Sequence Diagram): Демонстрирует взаимодействие объектов во времени (обмен сообщениями).

3. Диаграмма деятельности (Activity Diagram): Описывает алгоритмы и потоки работ (workflow), включая ветвления и параллельные действия.

4. Диаграмма «сущность-связь» (ER-diagram): Используется для проектирования структуры базы данных.

**Нефункциональные требования (Non-functional Requirements)** или атрибуты качества, описывают свойства системы, не связанные напрямую с её функциями, но критически важные для эксплуатации. К ним относятся:

1. Производительность:

Время отклика, пропускная способность.

2. Надежность:

Устойчивость к сбоям, сохранность данных.

3. Безопасность:

Защита от несанкционированного доступа, шифрование.

4. Удобство использования (Usability):

Интуитивность интерфейса.

5. Интерфейсы:

Требования к взаимодействию с внешними системами (API) и аппаратным обеспечением.

Совокупность этих требований формирует техническое задание (ТЗ), которое является основным документом для команды разработчиков.

## **Глава 2. Предпроектное исследование и обоснование необходимости разработки ПС**

### **2.1. Анализ предметной области**

#### **Описание текущего положения**

Объектом исследования является кадровое агентство «TalentBridge Agency», специализирующееся на поиске и подборе специалистов различного профиля.

В настоящий момент процессы подбора персонала в организации выполняются преимущественно вручную и имеют ряд существенных недостатков. Как показало предпроектное обследование (отчет «As-Is»), основные бизнес-процессы строятся следующим образом:

**1. Прием заявок:** Заявки от клиентов поступают по телефону или электронной почте в неструктурированном виде, часто с неполными требованиями.

**2. Обработка резюме:** Рекрутеры вручную скачивают резюме с работных сайтов, сохраняют их в локальные папки на рабочих компьютерах. Данные из резюме перепечатываются или копируются в разрозненные Excel-таблицы.

**3. Коммуникация:** Взаимодействие с кандидатами и заказчиками ведется через личную почту и телефон, история переписки не централизована.

**4. Отчетность:** Сводные отчеты для руководства собираются вручную в конце месяца, что трудоемко и влечет за собой высокий риск ошибок.

#### **Основные проблемы текущего состояния:**

1. Низкая эффективность ручного сбора данных (высокие трудозатраты).
2. Дублирование информации и риск её потери из-за отсутствия единой базы.
3. Сложность в поиске кандидатов по накопленному архиву (нет фильтров).
4. Зависимость от человеческого фактора.

## Организационная структура

Для выявления ключевых пользователей системы была проанализирована организационная структура предприятия. Организация «TalentBridge Agency» имеет линейно-функциональную структуру. Ключевые подразделения, действовавшие в процессе подбора, представлены на организационной модели.

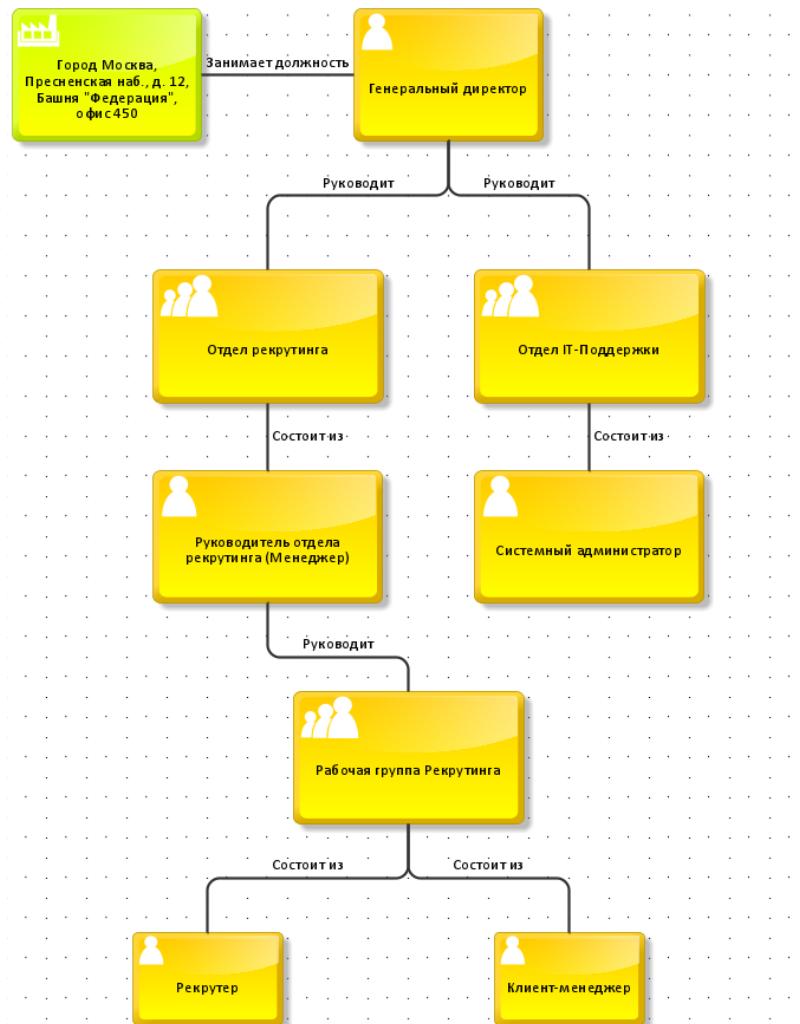


Рисунок 1 — Организационная модель предприятия

## Классы пользователей

На основе анализа структуры и процессов были выделены следующие классы пользователей (актеров), которые будут взаимодействовать с будущей системой:

*Таблица 1 — Классы пользователей системы*

№	Класс пользователя	Описание и роль
1	Рекрутер	Основной пользователь системы. Задачи: создание вакансий, поиск кандидатов, парсинг резюме, ведение кандидатов по воронке подбора, организация собеседований.
2	Менеджмент (Руководитель рекрутинга)	Управленческий персонал. Задачи: распределение вакансий между рекрутерами, мониторинг KPI, просмотр аналитических отчетов, принятие решений.
3	Клиент (Работодатель)	Внешний пользователь. Задачи: создание заявок на подбор, отслеживание статуса работы, просмотр отобранных кандидатов, предоставление обратной связи.

## 2.2. Сравнительный анализ существующих решений

Для обоснования необходимости собственной разработки был проведен анализ рынка существующих программных продуктов для автоматизации рекрутинга (ATS — Applicant Tracking Systems). В качестве аналогов для сравнения были выбраны популярные системы: **E-Staff** и **Huntflow**.

### Критерии сравнения

Для анализа были выбраны следующие критерии, критически важные для «TalentBridge Agency»:

**1. Наличие модуля парсинга:** Возможность автоматического извлечения данных из файлов резюме.

**2. Интеграция с внутренними сервисами:** Возможность доработки под специфические процессы агентства.

**3. Стоимость владения:** Затраты на лицензии и обслуживание.

**4. Безопасность данных (ФЗ-152):** Возможность размещения базы на собственных серверах компании (On-Premise) для полного контроля.

Сравнительная таблица

*Таблица 2 — Сравнительный анализ систем автоматизации*

Критерий	E-Staff Рекрутер	Huntflow (Хантфлоу)	Проектируемая АС «TalentBridge»
Парсинг резюме	Есть (встроенный)	Есть (облачный)	Есть (интеграция с ExternalParser API)
Тип поставки	Desktop / Server	Cloud (SaaS)	Web-приложение (On- Premise)
Гибкость доработки	Низкая (коробочный продукт)	Низкая (закрытый код)	Высокая (собственная разработка под процессы агентства)
Стоимость	Высокая (лицензия на раб. место)	Высокая (ежемесячная подписка)	Средняя (разовые затраты на разработку, низкие на поддержку)
Защита данных	Зависит от настройки локальной сети	Данные хранятся в облаке провайдера	Полный контроль (хранение в закрытом контуре агентства)

#### Обоснование актуальности разработки

Существующие на рынке решения (E-Staff, Huntflow) обладают широким функционалом, но имеют ряд недостатков для конкретного агентства: высокая стоимость лицензий для масштабирования штата и невозможность гибкой адаптации под внутренние уникальные регламенты работы. Кроме того, политика безопасности агентства требует полного

контроля над базой персональных данных кандидатов без передачи их в публичные облака.

Таким образом, разработка собственной системы «TalentBridge» является экономически и технически обоснованной.

### 2.3. Моделирование бизнес-процессов организации

#### Модель «Как есть» (As-Is)

Для детального анализа проблем была построена контекстная диаграмма процессов в текущем состоянии. На верхнем уровне процесс характеризуется ручным управлением и отсутствием автоматизации.

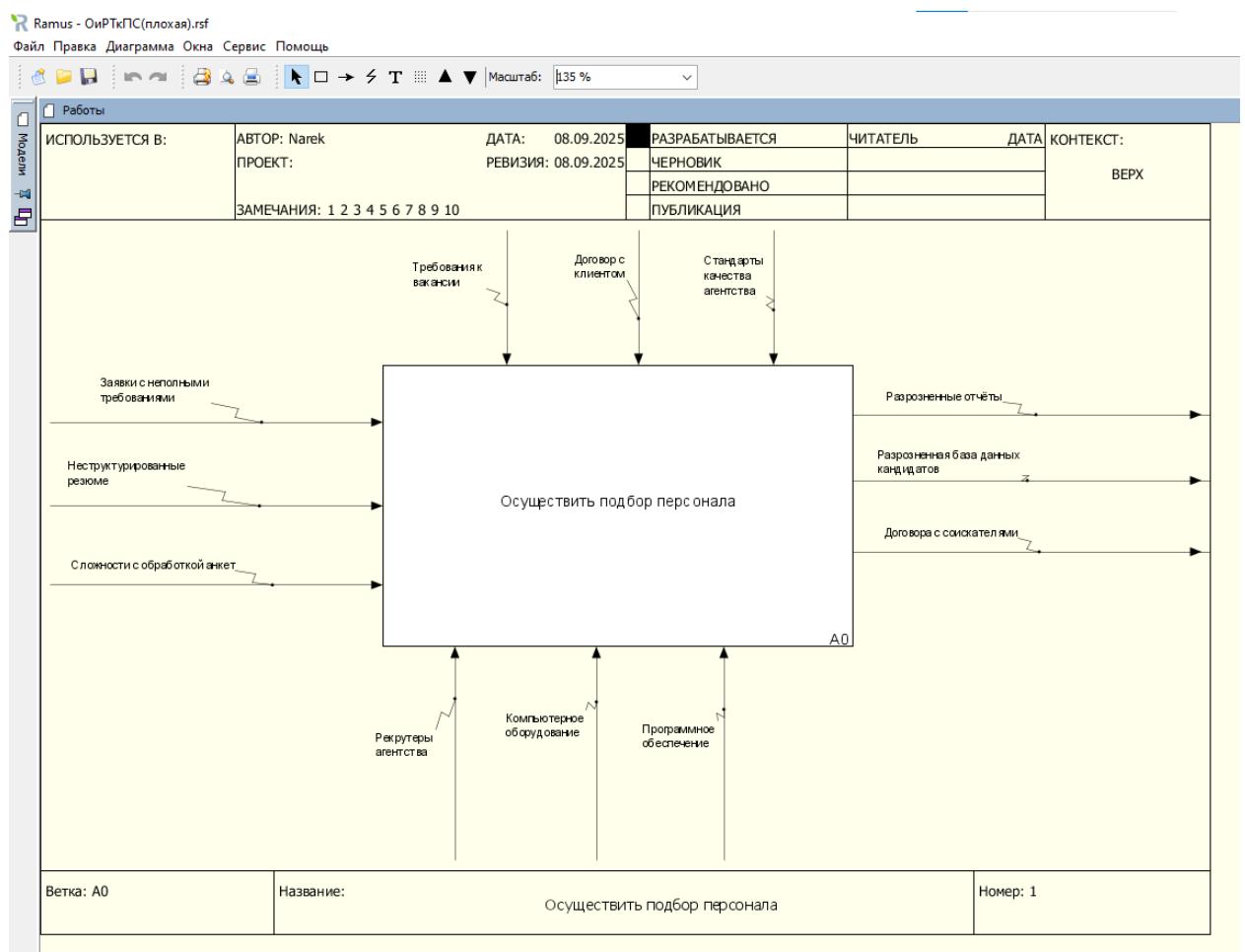


Рисунок 2 — Контекстная диаграмма (A0)

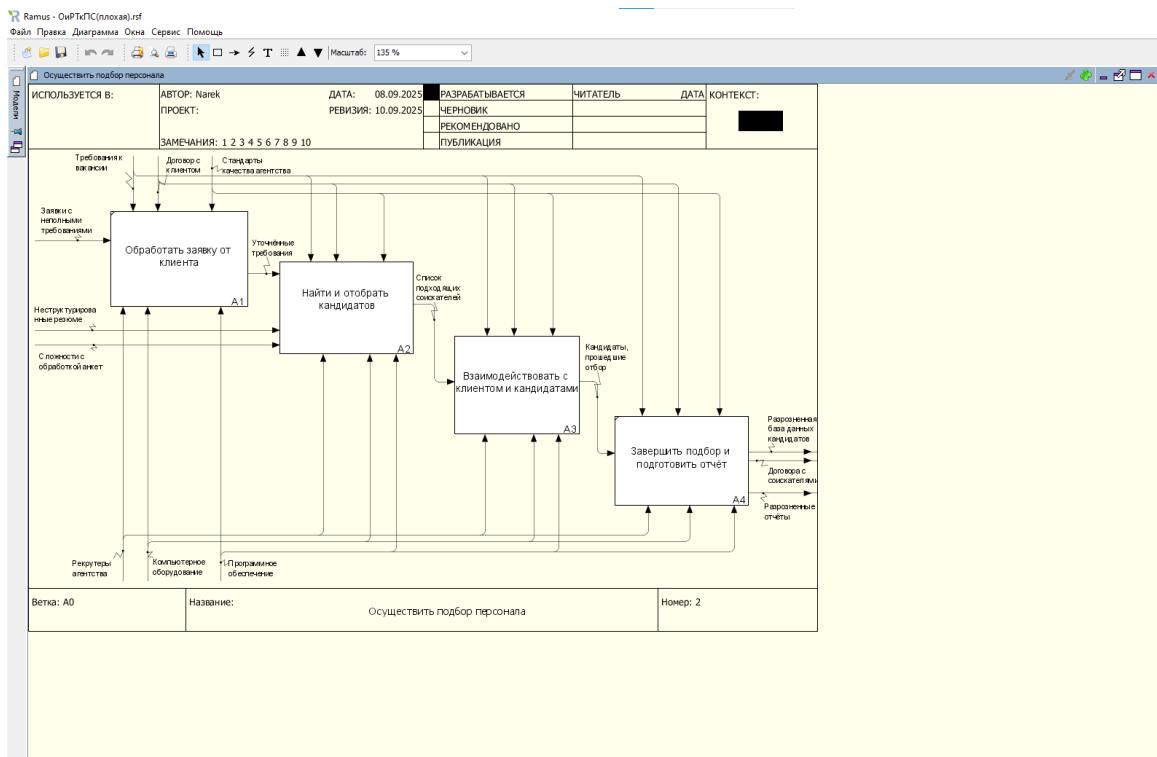


Рисунок 3 — Декомпозиция процесса (A1)

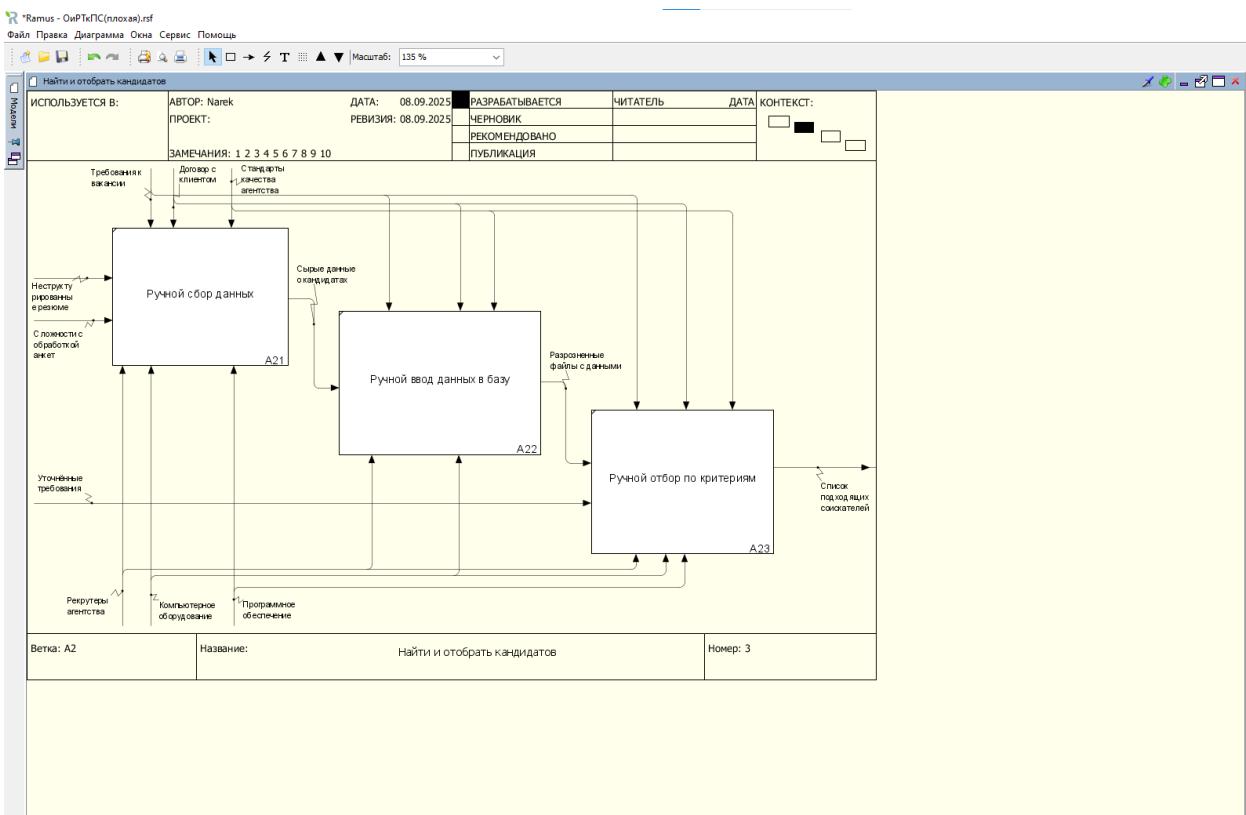


Рисунок 4 — Детализация блока A2 (Найти и отобрать кандидатов)

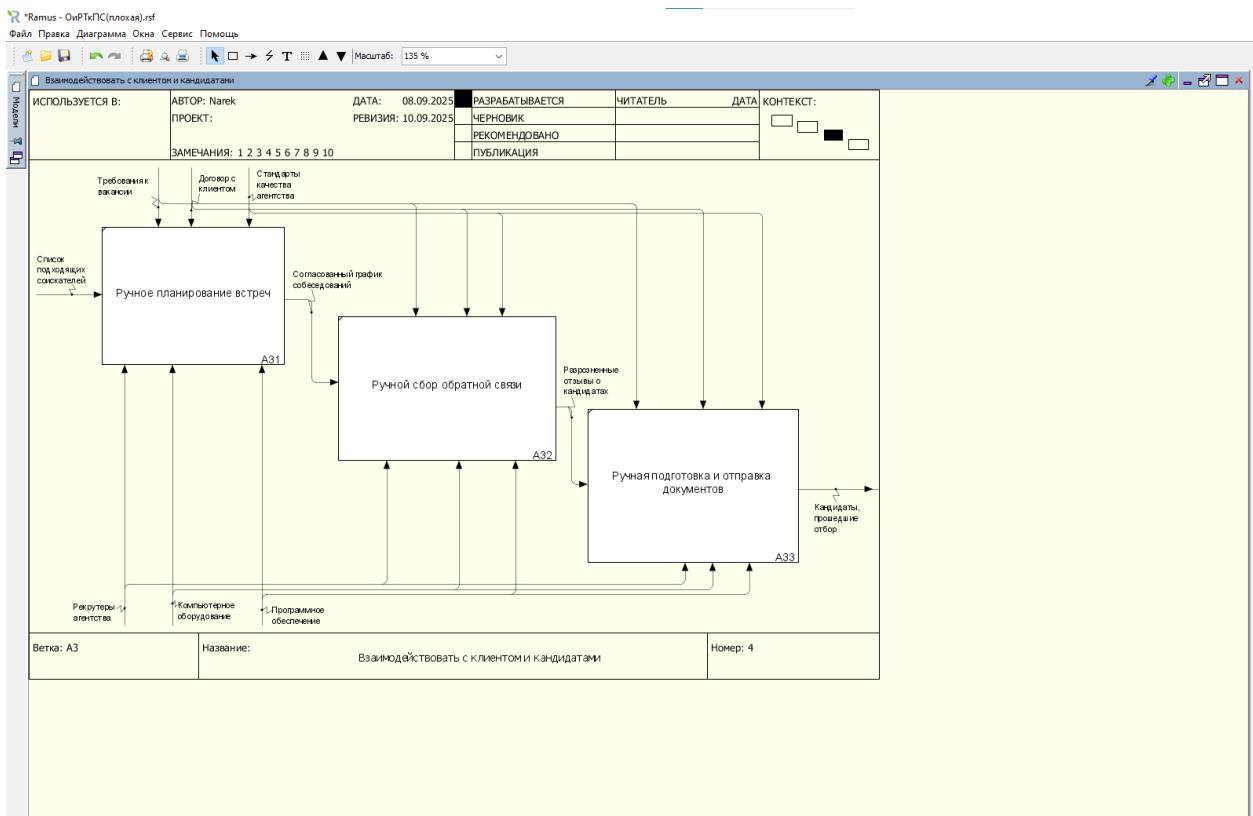


Рисунок 5 — Детализация блока А3 (Взаимодействовать с клиентом и кандидатами)

Анализ диаграммы показал наличие «узких мест»: ручной ввод данных в Excel, потеря информации при передаче между сотрудниками, задержки в формировании отчетности.

#### Определение бизнес-потребностей

На основе выявленных проблем сформированы ключевые бизнес-потребности:

1. Создать единую структурированную базу данных кандидатов и клиентов.
2. Автоматизировать рутинные операции (парсинг резюме, создание карточек).
3. Повысить скорость закрытия вакансий за счет удобного поиска и фильтрации.

## Дерево целей

Для достижения поставленных задач было построено дерево целей, декомпозирующее главную миссию проекта на подцели.

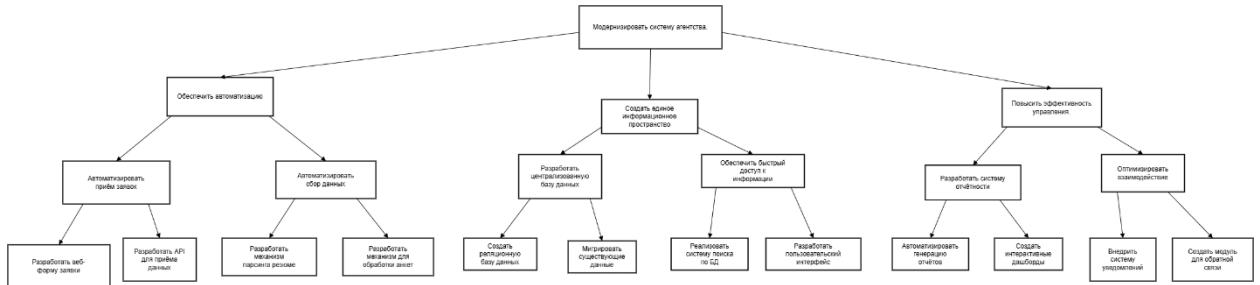


Рисунок 6 - Дерево целей

## Морфологическая карта

Для выбора оптимального технического решения был использован метод морфологического анализа. Рассмотрены альтернативные варианты реализации ключевых функций системы.

Таблица 3 — Морфологическая карта

Характеристика / Функция	Вариант 1 (Ручной способ)	Вариант 2 (Офисное ПО)	Вариант 3 (Проектируемая система TalentBridge)
Способ ввода резюме	Перепечатывание с бумаги	Копирование (Ctrl+C/Ctrl+V) в Excel	Автоматический парсинг файлов (PDF/Docx)
Хранение базы данных	Папки бумагами	Локальные файлы Excel на ПК	Единая база данных PostgreSQL (Client-Server)
Поиск кандидатов	Ручной перебор папок	Поиск по строкам в Excel	Многокритериальный поиск и фильтрация
Формирование отчетов	Ручной расчет на калькуляторе	Формулы и сводные таблицы Excel	Автоматическая генерация PDF/XLS по шаблону

Взаимодействие с клиентом	Телефонные звонки	Электронная почта (Outlook)	Личный кабинет (Веб-портал)
---------------------------	-------------------	-----------------------------	-----------------------------

*Примечание: Зеленым цветом выделена выбранная комбинация решений, обеспечивающая максимальную эффективность.*

### Модель «Как должно быть» (To-Be)

С учетом выбранных решений была построена модель процессов будущего состояния. В новой модели внедрена «Единая база данных кандидатов» и модули автоматизации, что устраняет выявленные проблемы.

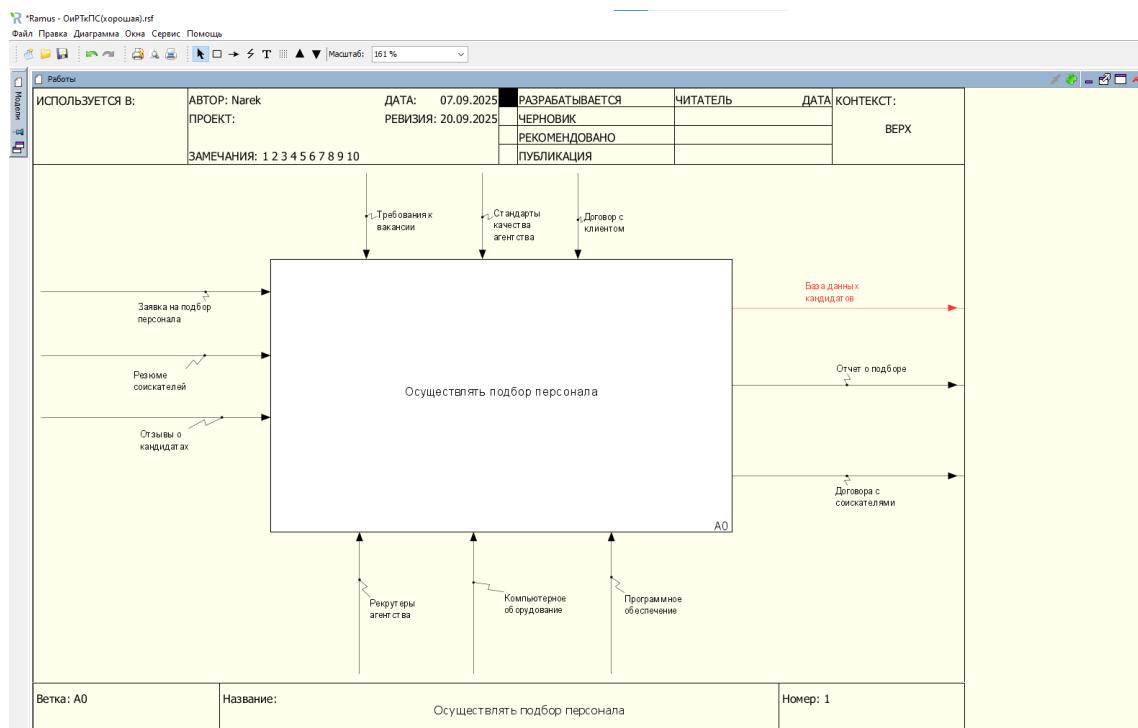


Рисунок 5 — Контекстная диаграмма (A0)

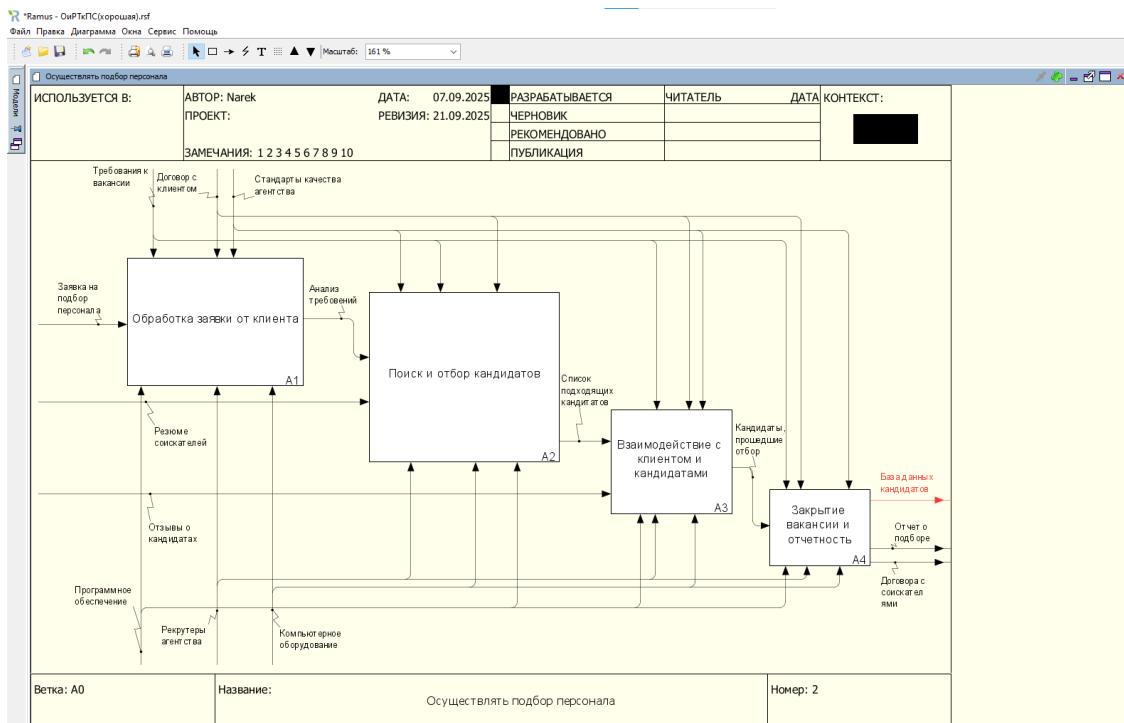


Рисунок 6 — Декомпозиция процесса (A1)

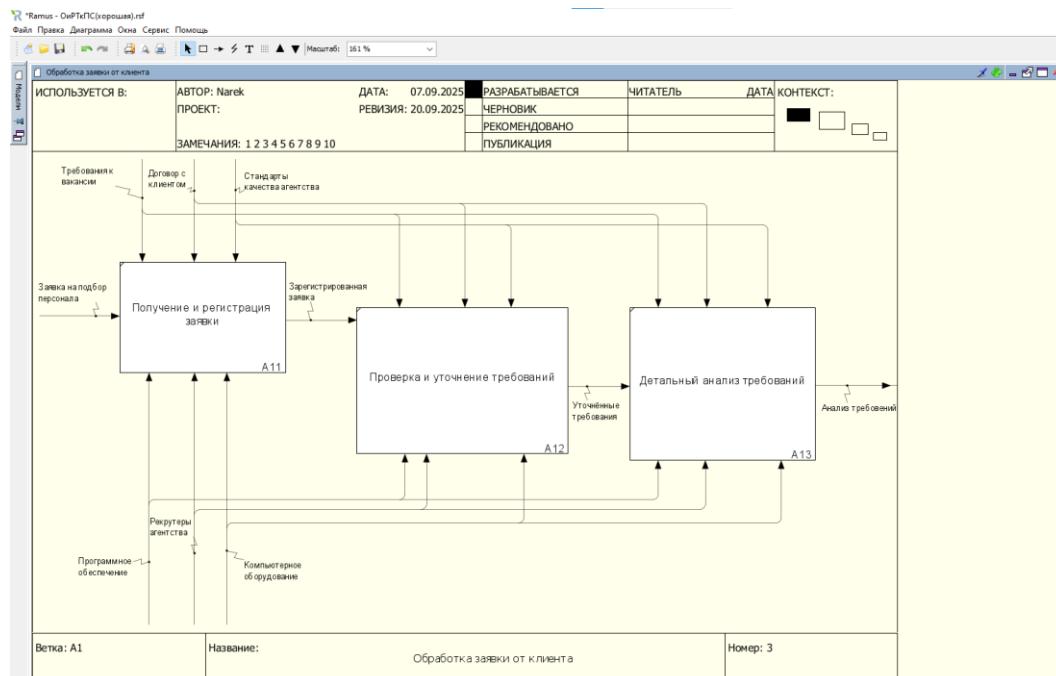


Рисунок 7 — Детализация блока A1 (Обработка заявки от клиента)

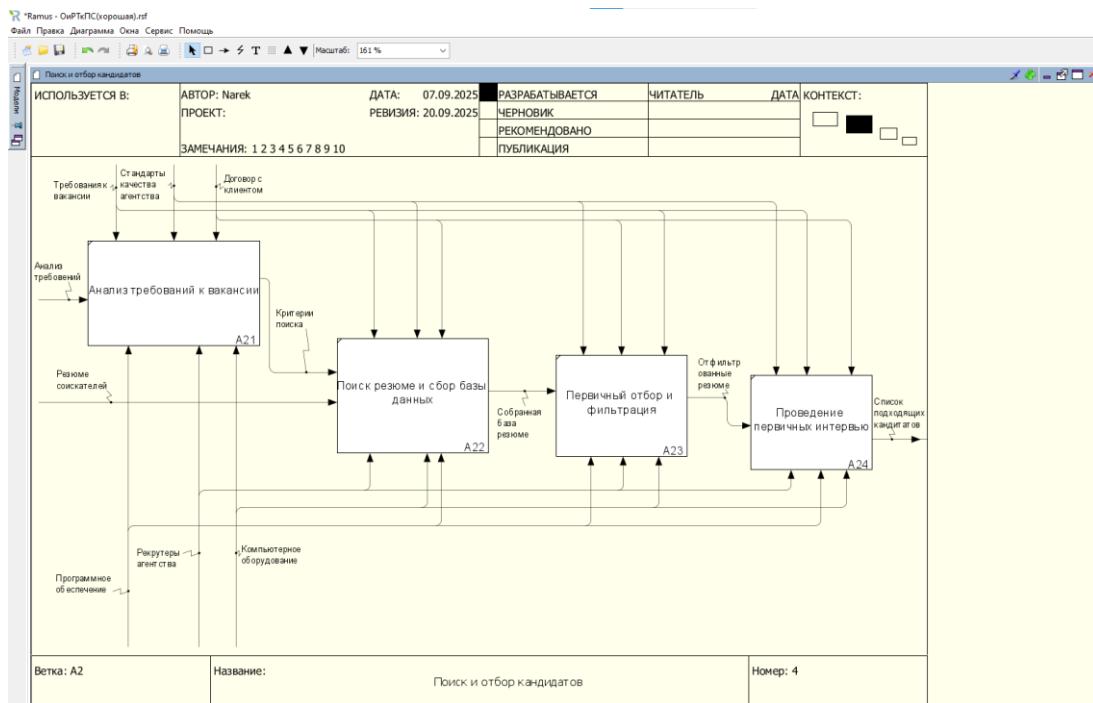


Рисунок 8 — Детализация блока А2 (Поиск и отбор кандидатов)

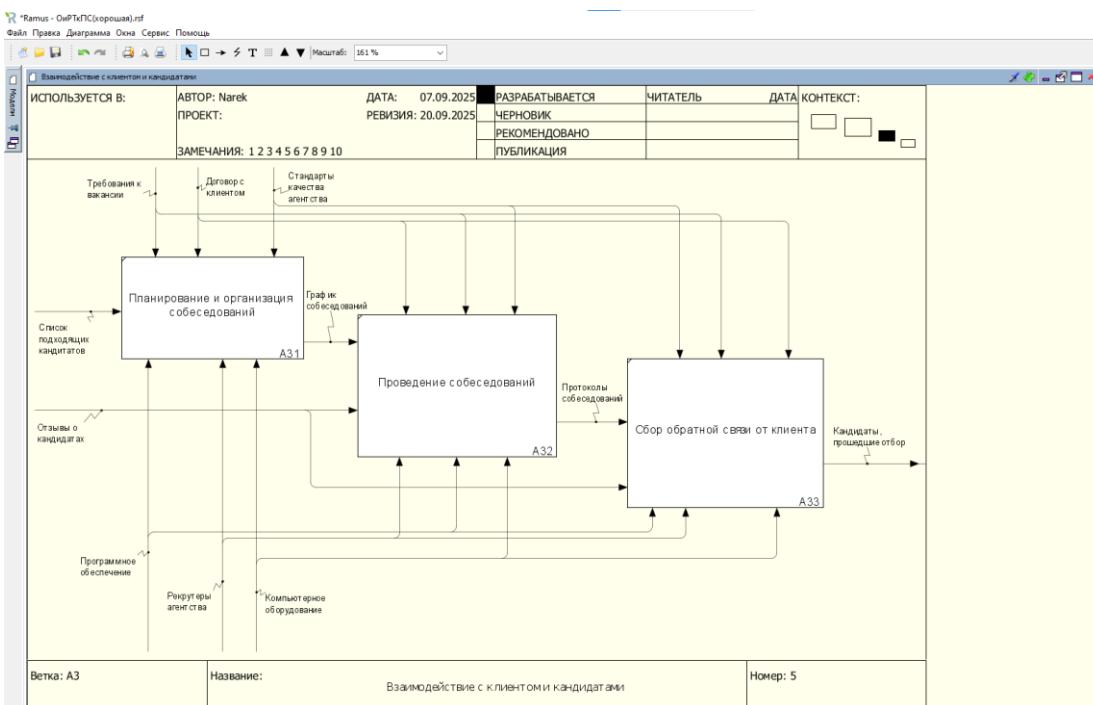


Рисунок 9 — Детализация блока А3 (Взаимодействие с клиентом и кандидатами)

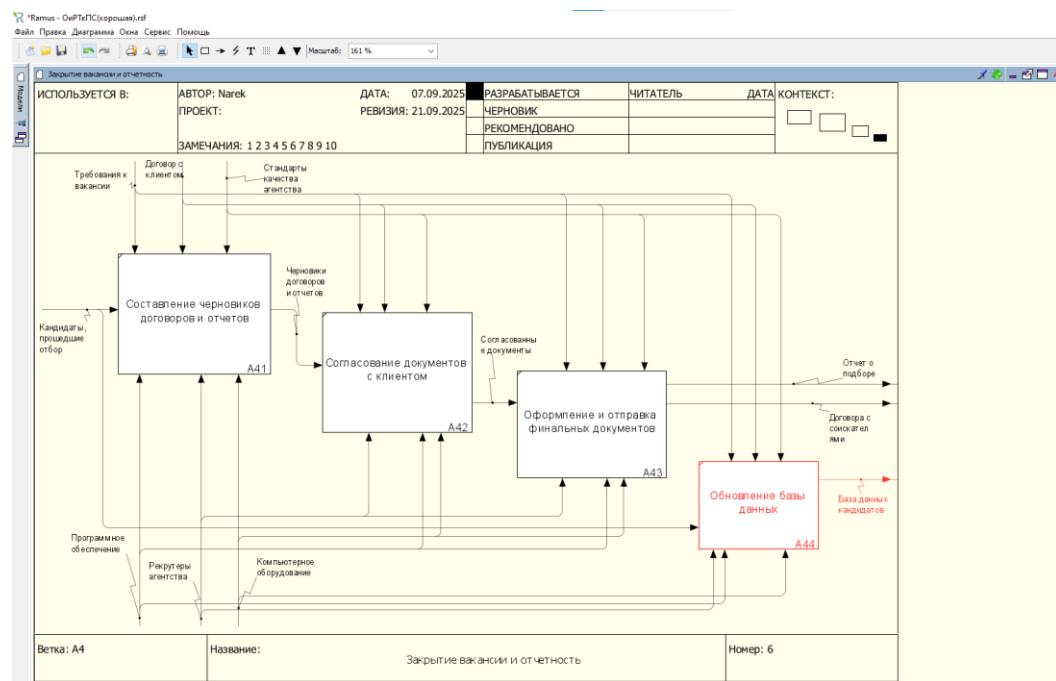


Рисунок 10 — Детализация блока А4 (Закрытие вакансий и отчетность)

## **2.4. Концепции продукта и границы проекта**

На основе проведенного исследования была сформулирована концепция продукта.

АС «**TalentBridge**» — это программная система, предназначенная для автоматизации полного цикла рекрутинга: от получения заявки до выхода кандидата на работу.

### **Границы проекта (MVP - Первая версия):**

В рамках первой версии системы будут реализованы следующие функции:

1. Управление базой вакансий и кандидатов.
2. Парсинг резюме.
3. Ведение истории взаимодействий.
4. Формирование базовой отчетности.

Функции, выходящие за рамки первой версии (интеграция с работными сайтами HeadHunter/SuperJob, мобильное приложение), запланированы к реализации в следующих релизах.

Полный текст документа «О концепции и границах» приведен в **Приложении 1.**

## Глава 3. Разработка требований к программной системе

### 3.1. Определение и документирование пользовательских требований

На этапе анализа требований важно определить функциональные возможности системы с точки зрения конечных пользователей. Для этого используется визуальное моделирование в нотации UML (Unified Modeling Language).

#### Диаграмма вариантов использования

Диаграмма вариантов использования (Use Case Diagram) определяет границы системы и отношения между актерами (пользователями) и прецедентами (функциями).

Для системы «TalentBridge» были выделены основные актеры: Рекрутер, Руководитель рекрутинга, Сотрудник Агентства, Клиент-менеджер, Работодатель, Кандидат и Системный администратор. Основные функции включают управление базой кандидатов, управление вакансиями и формирование отчетности.

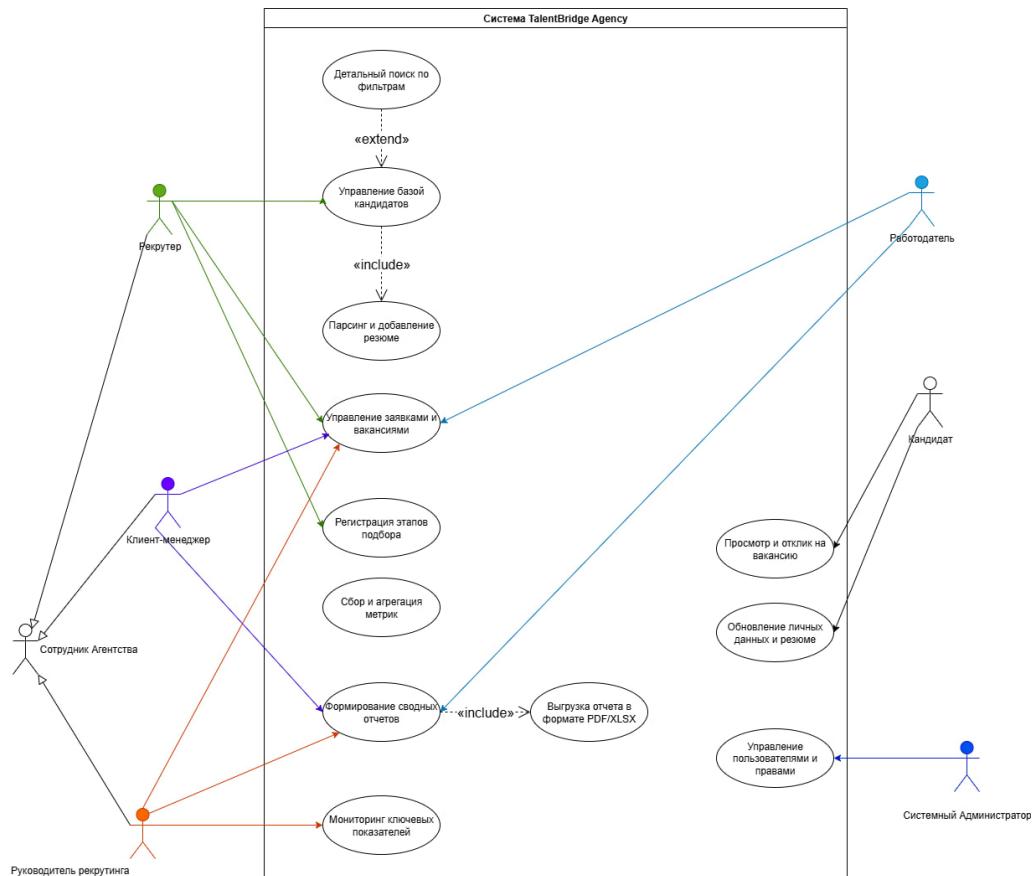


Рисунок 11 — Диаграмма вариантов использования «TalentBridge»

## Пользовательские истории

Для детализации требований диаграмма вариантов использования была декомпозирована в набор пользовательских историй (User Stories), описывающих ценность каждой функции для бизнеса.

1. Блок: Управление базой кандидатов (UC: Управление базой кандидатов, Парсинг и добавление резюме, Детальный поиск)

Роль	Цель (Что я хочу)	Польза (Зачем мне это нужно)
Рекрутер	Я хочу полноценно управлять базой кандидатов, включая автоматический парсинг загруженных PDF/DOCX-резюме и детальный поиск по комбинации сложных фильтров (навыки, зарплатные ожидания, статус).	Чтобы сократить время на ручной ввод данных и гарантированно находить наиболее релевантных кандидатов для сложной или узкопрофильной вакансии.
Рекрутер	Я хочу оперативно регистрировать и изменять статусы кандидата (например, "Активный поиск", "Собеседование назначено"), добавляя краткие комментарии.	Чтобы всегда иметь актуальную информацию о кандидате и понимать, на каком этапе работы он находится.
Кандидат	Я хочу иметь возможность обновить свои личные данные и резюме через личный кабинет в любое удобное время.	Чтобы информация, которую видит Рекрутер, всегда была актуальной, что повышает мои шансы на успешное трудоустройство.

2. Блок: Управление вакансиями и отбор (UC: Управление заявками и вакансиями, Регистрация этапов подбора, Просмотр и отклик на вакансию)

Роль	Цель (Что я хочу)	Польза (Зачем мне это нужно)
Руководитель рекрутинга	Я хочу создать новую заявку на вакансию в специальной форме, указав требования, бюджет и сроки.	Чтобы официально начать процесс подбора и автоматически уведомить команду рекрутеров о новой задаче.
Рекрутер	Я хочу просматривать и управлять списком всех активных вакансий, назначенных мне, с возможностью сортировки по приоритету.	Чтобы эффективно планировать свою рабочую нагрузку и фокусироваться на задачах с высоким приоритетом.
Рекрутер	Я хочу регистрировать все этапы подбора (звонок, интервью, тестовое задание, отправка Клиенту) по каждой вакансии, с возможностью прикрепить документы.	Чтобы обеспечить прозрачность процесса и иметь полную историю взаимодействия с кандидатом и работодателем.
Кандидат	Я хочу просматривать открытые вакансии и иметь возможность быстро откликнуться на заинтересовавшую меня позицию.	Чтобы максимально оперативно участвовать в процессе подбора и не упустить подходящую работу.

3. Блок: Отчетность и мониторинг (UC: Формирование сводных отчетов, Мониторинг ключевых показателей)

Роль	Цель (Что я хочу)	Польза (Зачем мне это нужно)
Руководитель рекрутинга	Я хочу формировать сводный отчет о закрытых вакансиях и среднем времени закрытия (Time to Hire) за выбранный квартал.	Чтобы оценить эффективность работы команды и определить "узкие места" в процессе подбора для дальнейшей оптимизации.
Руководитель рекрутинга	Я хочу видеть интерактивный дашборд с ключевыми показателями (KPI) по всей моей команде в реальном времени.	Чтобы оперативно принимать управленческие решения и контролировать выполнение плана по подбору, не дожидаясь генерации отчетов.
Рекрутер	Я хочу просматривать персональный дашборд со своими ключевыми метриками эффективности (например, среднее время отбора) за текущий период.	Чтобы самостоятельно отслеживать выполнение KPI и понимать, где я могу улучшить свою работу.

4. Блок: Взаимодействие с Работодателем (UC: Управление заявками и вакансиями, Формирование сводных отчетов)

Роль	Цель (Что я хочу)	Польза (Зачем мне это нужно)
Работодатель	Я хочу создавать и отслеживать статус моих заявок, а также	Чтобы получать информацию оперативно

	комментировать и утверждать/отклонять предложенных кандидатов через личный кабинет.	и быть в курсе прогресса, активно участвуя в процессе отбора.
Клиент-менеджер	Я хочу иметь возможность выгружать сводные отчеты по клиентским заявкам, сформированные системой.	Чтобы оперативно предоставлять Работодателю детальную информацию о прогрессе без необходимости ручного сбора данных.
Клиент-менеджер	Я хочу видеть общую историю коммуникации (звонки, встречи, письма) по каждому Работодателю в одном месте.	Чтобы обеспечить высокое качество обслуживания и не потерять важную информацию о предпочтениях клиента.

## 5. Блок: Администрирование (UC: Управление пользователями и правами)

Роль	Цель (Что я хочу)	Польза (Зачем мне это нужно)
Системный Администратор	Я хочу гибко управлять правами доступа для каждого пользователя, назначая и деактивируя роли (Рекрутер, Клиент-менеджер, Работодатель и др.).	Чтобы обеспечить безопасность данных и гарантировать, что сотрудники видят только ту информацию, которая соответствует их должности и внешней роли.

### 3.2. Определение и документирование функциональных требований

Функциональные требования описывают внутреннее поведение системы, необходимое для реализации пользовательских запросов.

#### 3.2.1. Диаграмма классов анализа

Для понимания внутренней структуры системы на концептуальном уровне была разработана диаграмма классов анализа. В ней выделены три типа классов:

1. **Boundary (Граничные):** Интерфейсы пользователя (Интерфейс Сотрудника, Интерфейс Кандидата).
2. **Control (Управляющие):** Логика обработки (Управление Вакансиями, Управление Базой).
3. **Entity (Сущности):** Данные (Вакансия, Кандидат, Этап Подбора).

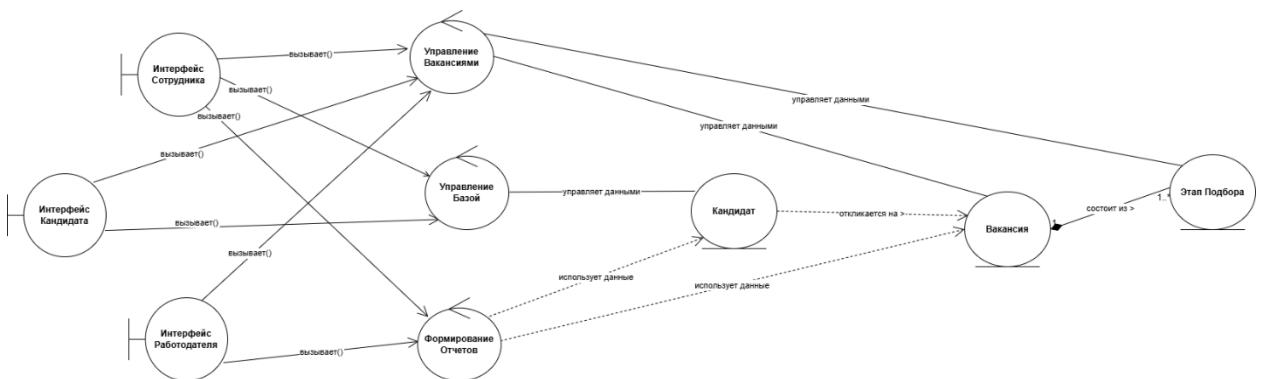


Рисунок 12 – Диаграмма классов анализа системы TalentBridge Agency

### 3.2.2. Диаграмма последовательности

Для моделирования взаимодействия объектов во времени был рассмотрен ключевой процесс — «Регистрация новой вакансии». Диаграмма показывает обмен сообщениями между Рекрутером, Интерфейсом, Контроллером и Базой данных.

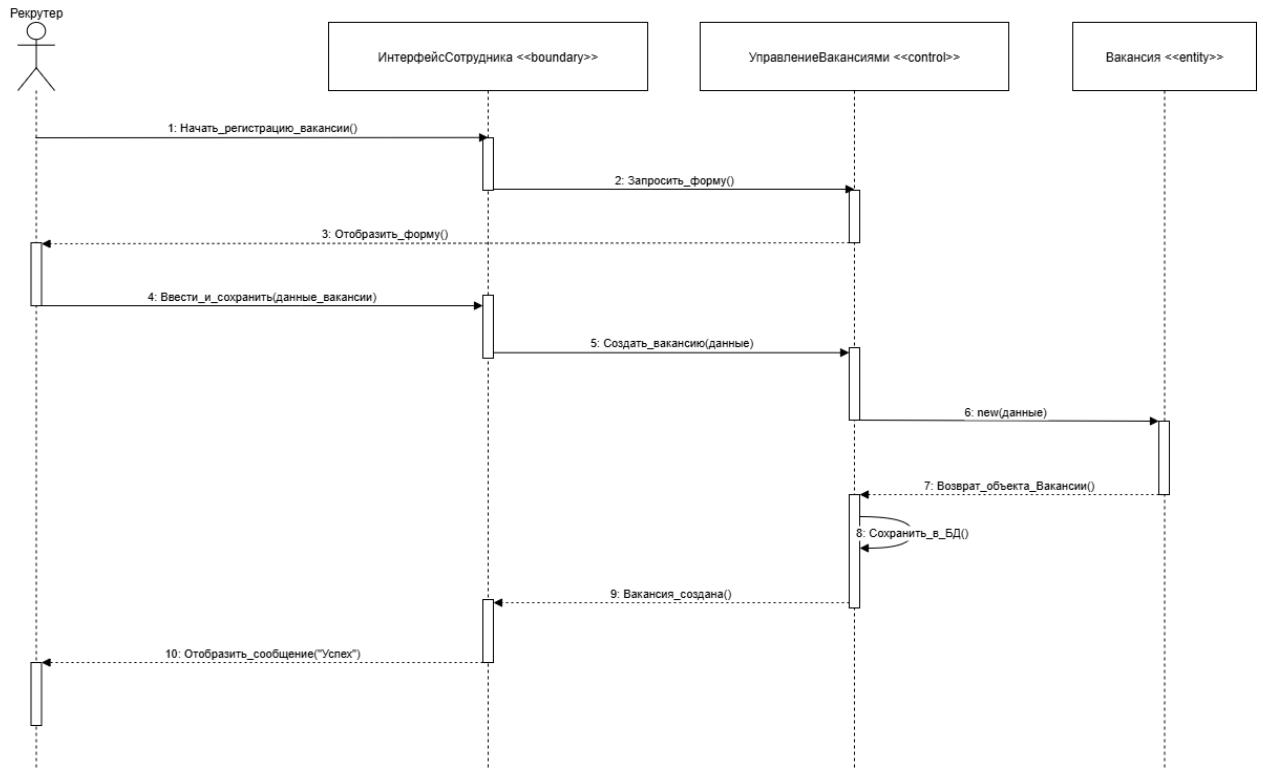


Рисунок 13 – Диаграмма последовательности для процесса «Регистрация новой вакансии»

### 3.2.3. Диаграмма коммуникации

Диаграмма коммуникации дополняет диаграмму последовательности, акцентируя внимание на структурных связях между объектами, участвующими в регистрации вакансии.

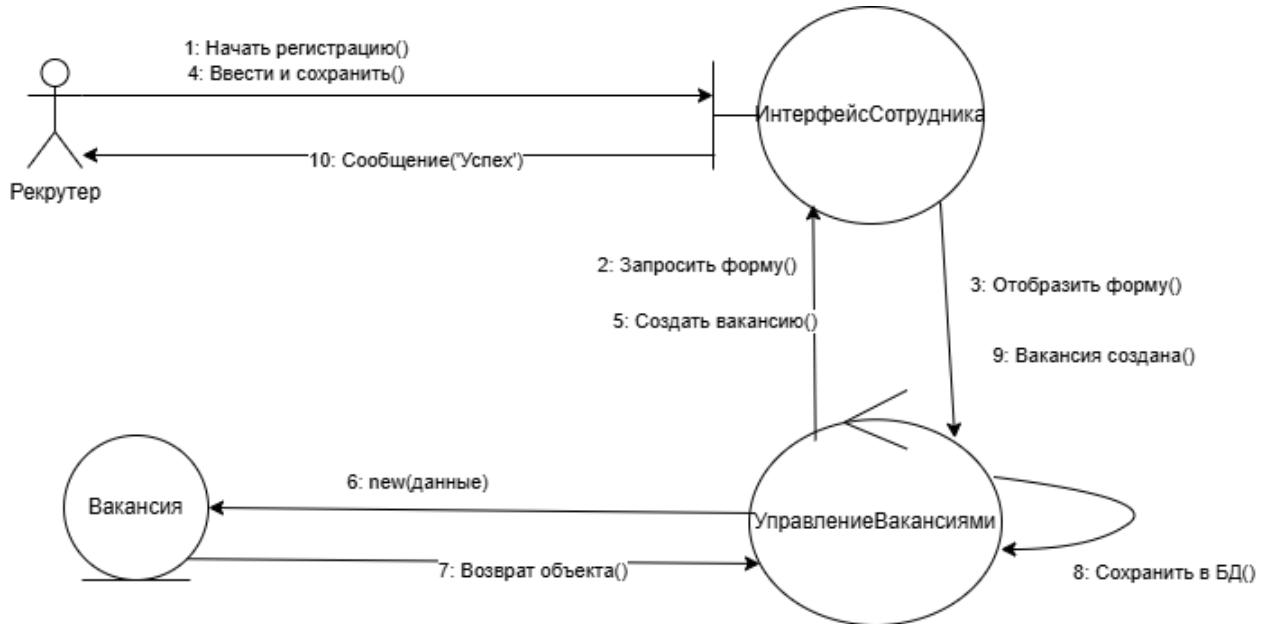


Рисунок 14 — Диаграмма коммуникации процесса регистрации вакансии

### 3.2.4. Диаграммы деятельности

Для описания алгоритмов выполнения сложных бизнес-процессов разработаны диаграммы деятельности.

**1. Процесс управления базой кандидатов:** Описывает логику проверки дубликатов при загрузке резюме. Если кандидат уже есть в базе, система предлагает обновление, иначе — создает нового.

**2. Процесс регистрации вакансии:** Показывает шаги заполнения формы и валидацию обязательных полей.

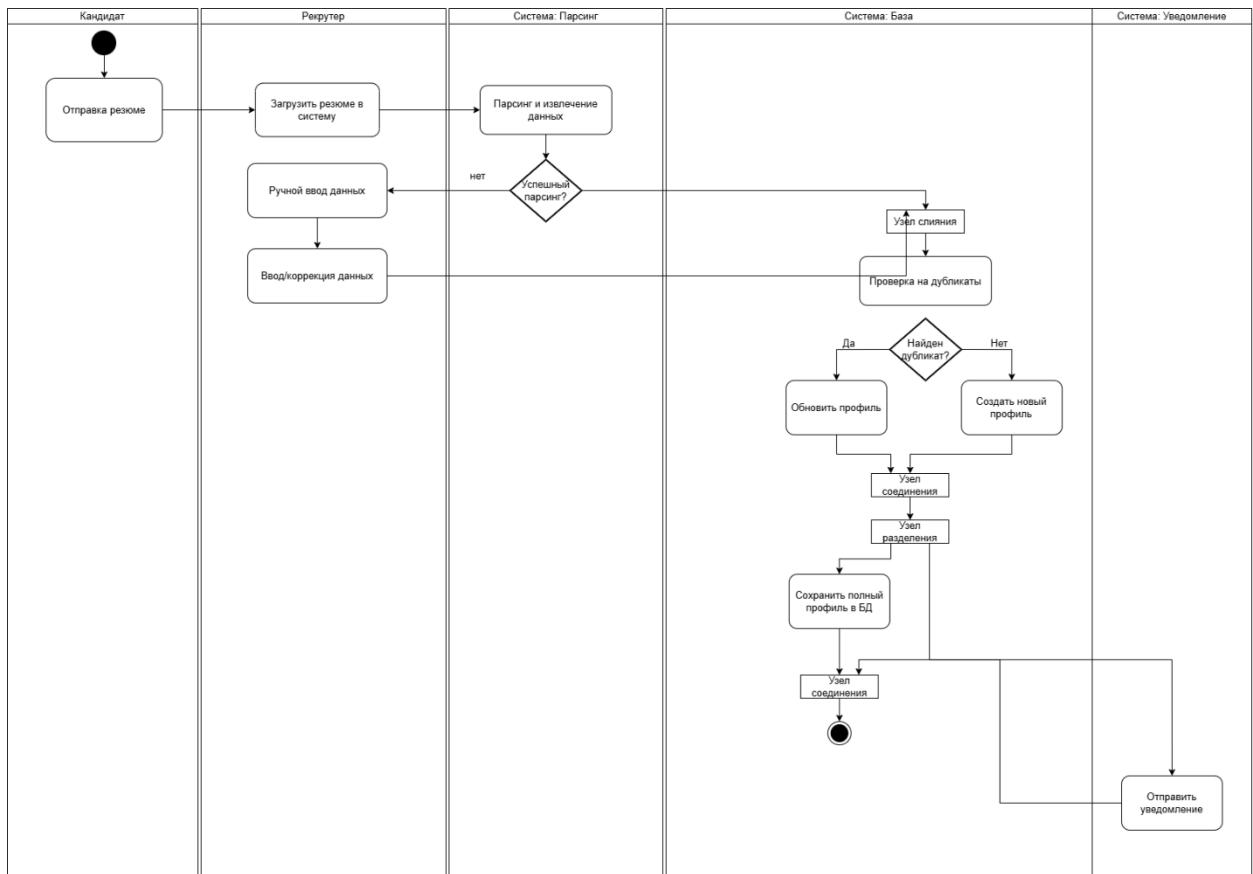


Рисунок 15 — Диаграмма деятельности для управления базой кандидатов

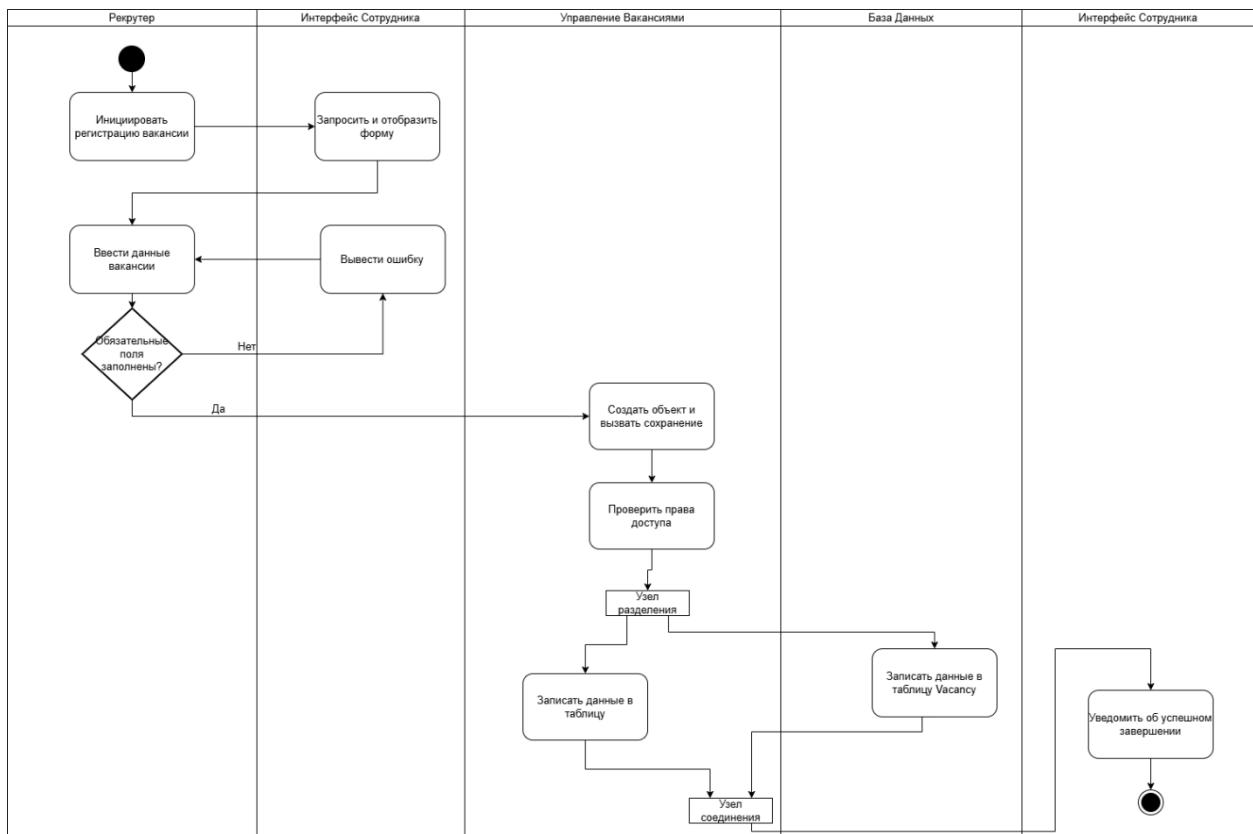


Рисунок 16- Диаграмма деятельности для процесса «Регистрация новой вакансии»

### 3.2.5. Диаграмма классов уровня проектирования

На этапе проектирования диаграмма классов была детализирована: определены атрибуты (типы данных), методы (операции) и отношения (наследование, композиция) для последующей реализации в коде.



Рисунок 17 - Диаграмма классов: только названия классов

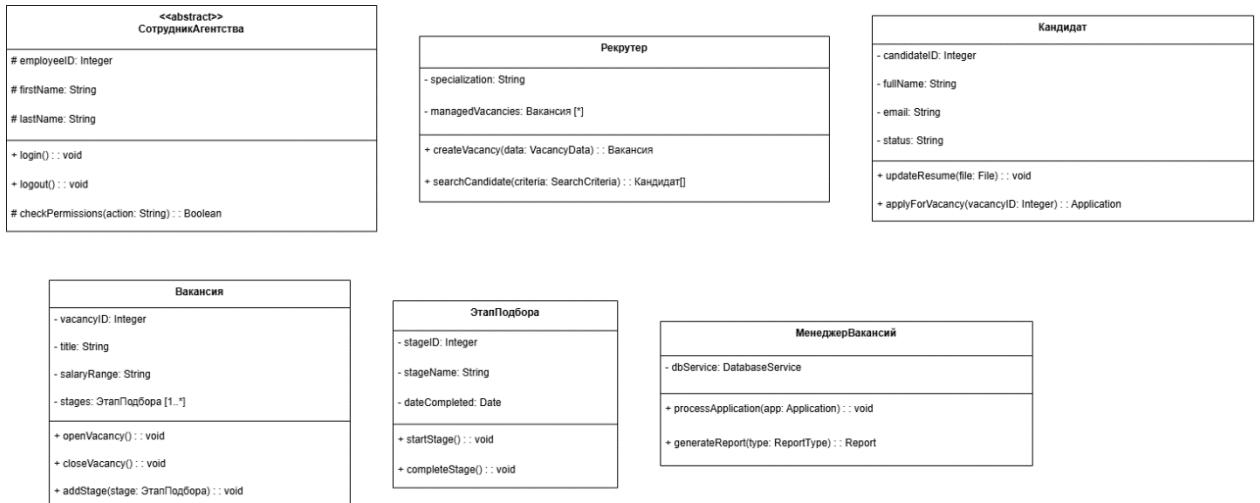


Рисунок 18 - Диаграмма классов с атрибутами и методами (без связей)

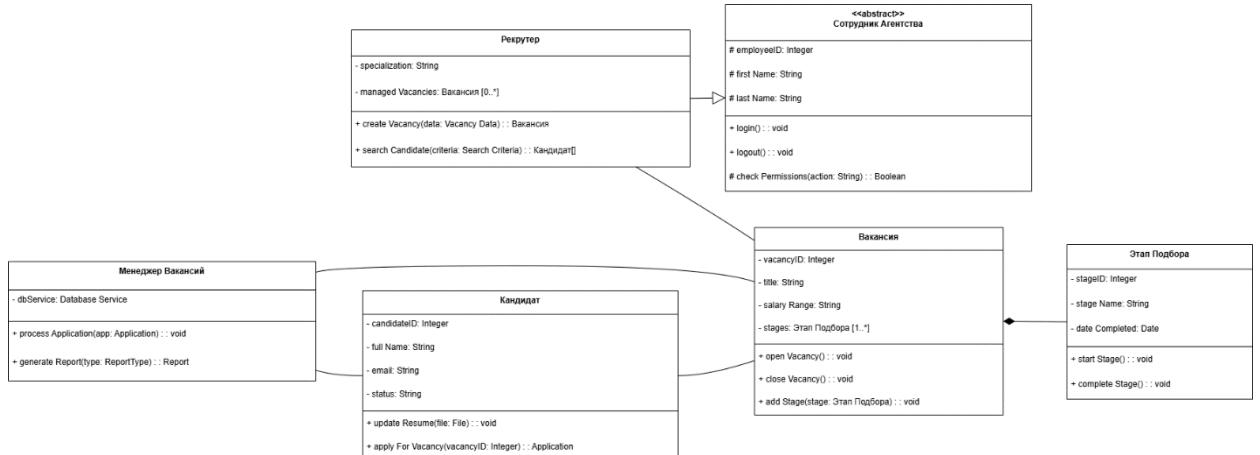


Рисунок 19 - Диаграмма классов уровня проектирования/реализации с

указанием связей

### 3.2.6. Диаграмма «сущность-связь» (ERD)

Для проектирования физической модели базы данных (PostgreSQL) разработана ER-диаграмма. Она описывает таблицы (Candidate, Vacancy,

Application, Recruiter), их поля (Primary Key, Foreign Key) и связи (один-ко-многим).

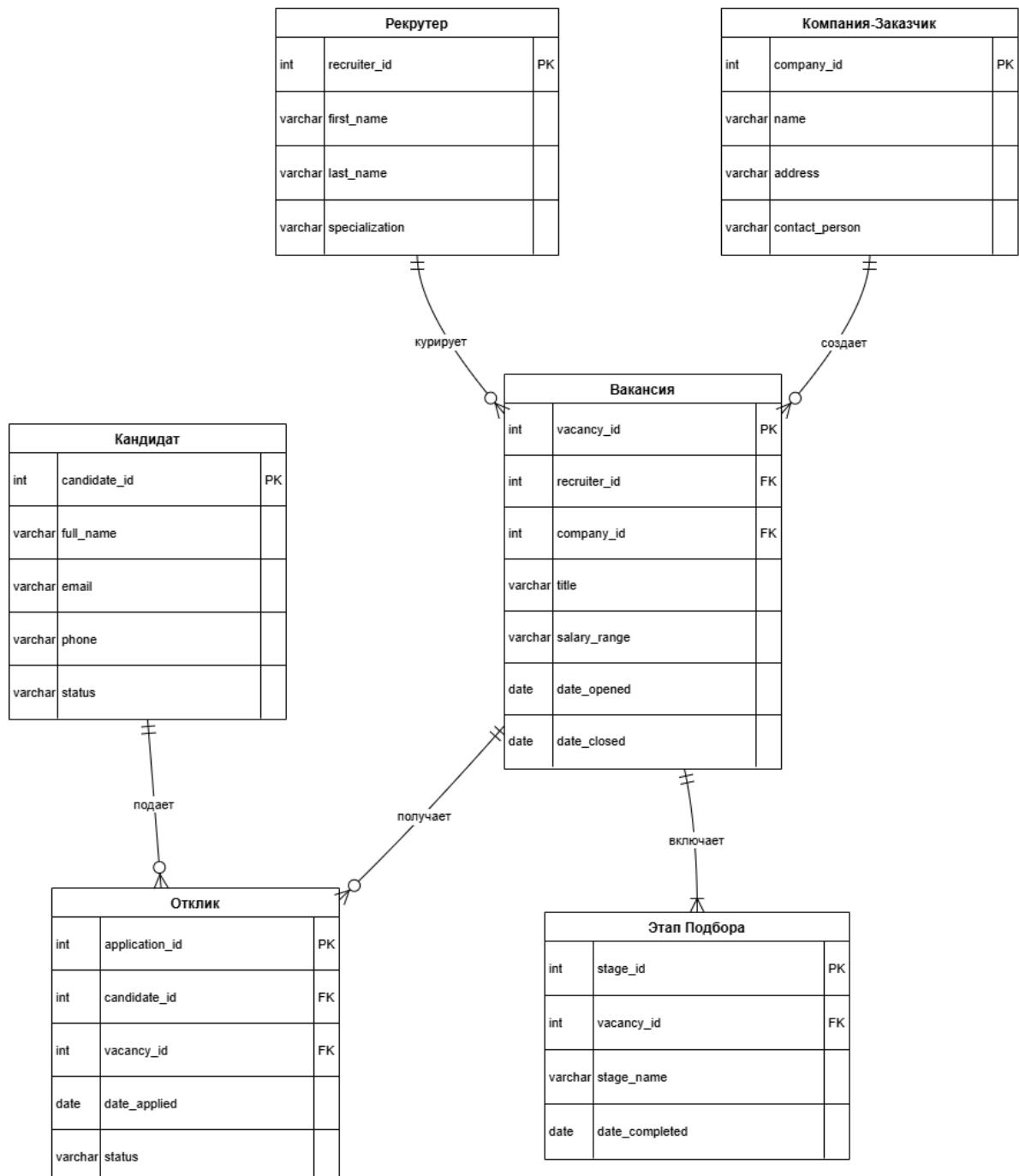


Рисунок 20- Диаграмма сущность-связи

### 3.2.7. Диаграмма состояний

Диаграмма состояний описывает жизненный цикл ключевых объектов системы.



Рисунок 21 – Диаграмма состояния жизненного цикла

### 3.3. Определение и документирование нефункциональных требований

Нефункциональные требования определяют качественные характеристики системы и архитектурные решения.

#### 3.3.1. Пользовательский интерфейс и карта окон

Интерфейс системы спроектирован в виде веб-приложения с разделением на порталы (Сотрудника, Работодателя, Кандидата). Схема навигации (карта диалоговых окон) показывает переходы между экранами Dashboard, списков и карточек сущностей.

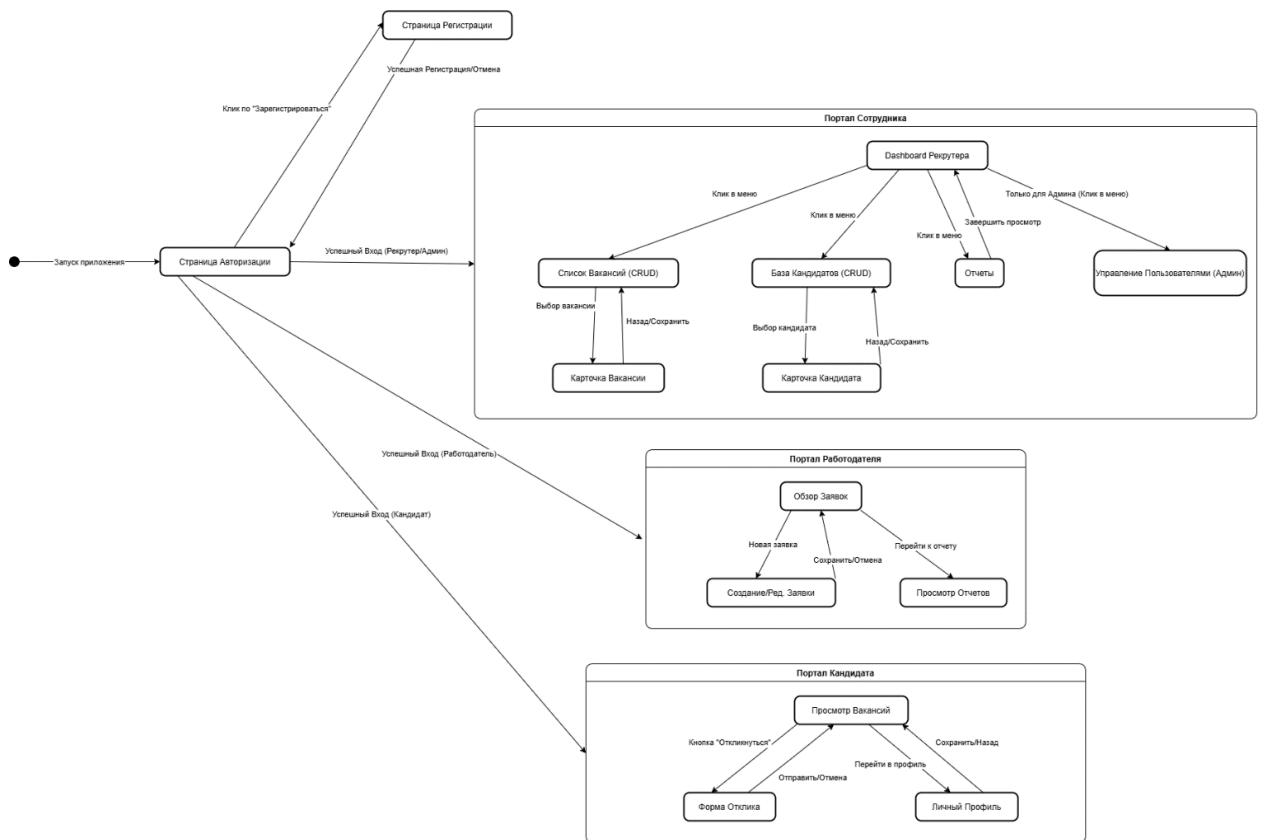


Рисунок 22 — Схема навигации и взаимодействия экранных форм

Также разработаны прототипы основных экранных форм (Вход, Дашборд, Карточка кандидата).

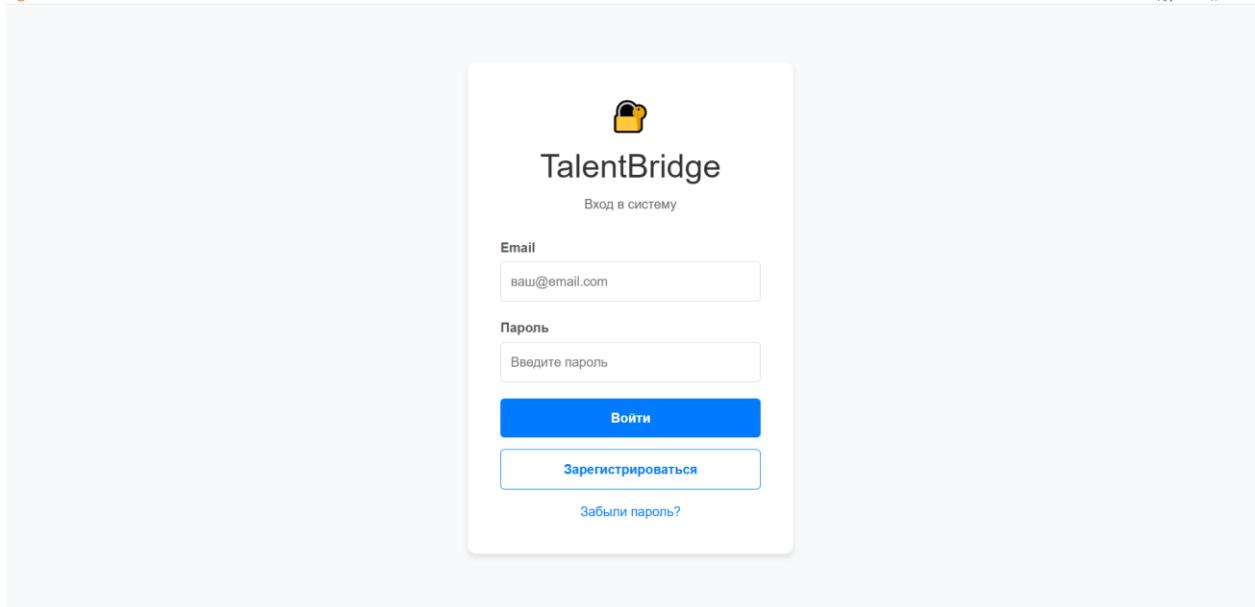


Рисунок 23 - Вход в систему

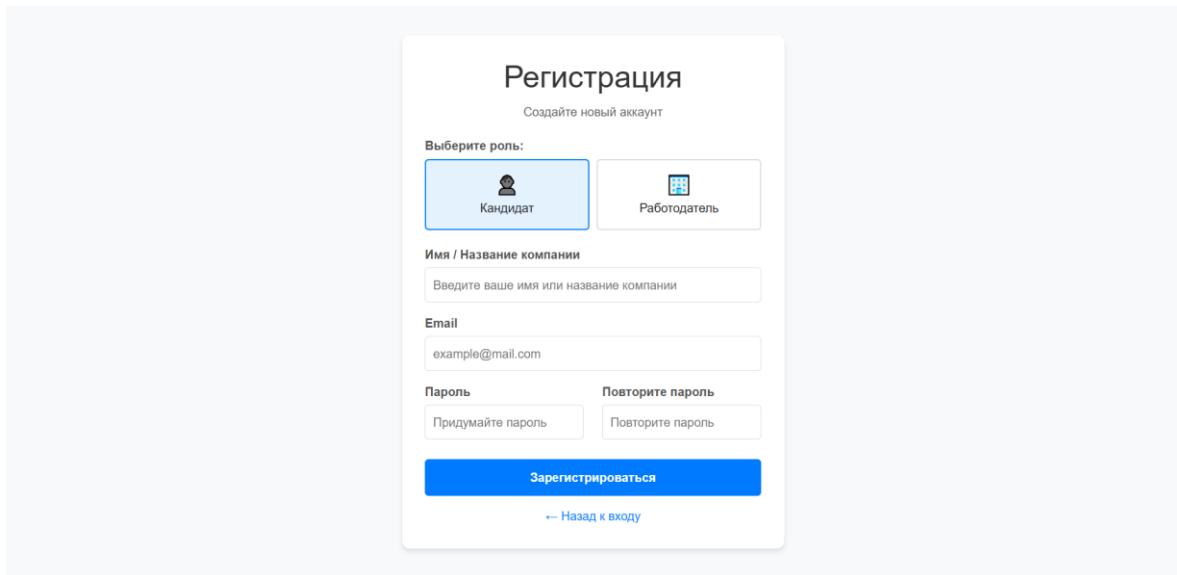


Рисунок 24 - Регистрация

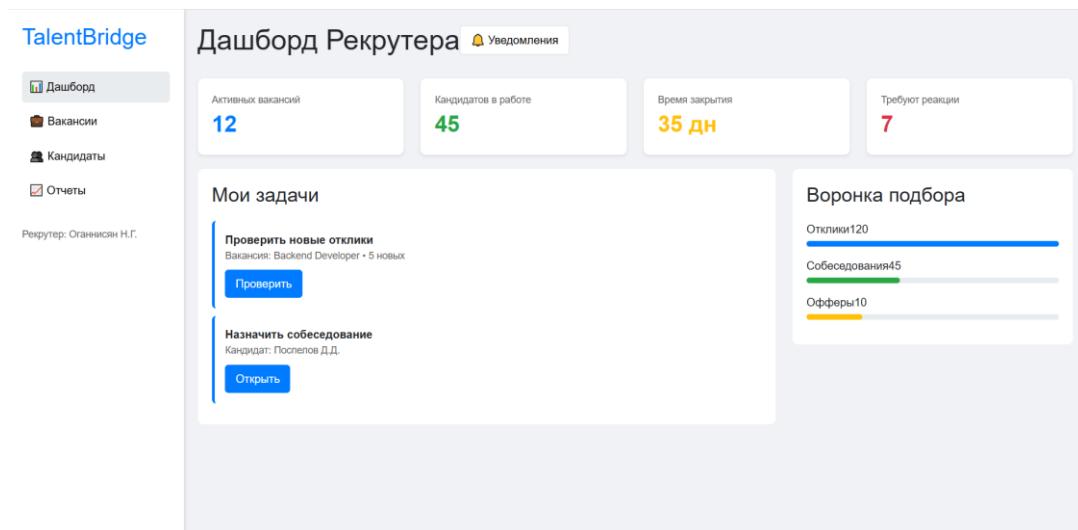


Рисунок 25 - Дашборд рекрутера

**Поспелов Дмитрий Денисович**  
Рассматривается на: Front or Deport (Senior)

**Изменить статус** **Сохранить и выйти**

**Текущий этап: Первичное собеседование**

**Общая информация** Опыт и Резюме Комментарии (3)

**Общие данные**

Email: pospelov.dima@mail.ru Телефон: +7 (228) 228-69-69

Зарплатные ожидания: 180 руб.

**История подбора**

01.10.2025 Отклик получен Система автоматически добавила кандидата

03.10.2025 Первичное собеседование Рекрутер Оганисян Н. назначил звонок

Рисунок 26 - Карточка кандидата

**Мои заявки на подбор** **+ Создать новую заявку**

Поиск по названию... Всё статусы Отчеты

Название должности	Дата создания	Рекрутер	Кандидатов	Статус	Действия
Менеджер по развитию бизнеса	15.09.2025	Трисветов Р.	8	В работе	<b>Изменить</b>
Backend Developer (Senior)	01.08.2025	Оганисян Н.	2	Завершена	<b>Просмотр</b>
Специалист по подбору персонала	20.10.2025	Жулёва Е.	0	Открыта	<b>Изменить</b>

Вы вошли как Работодатель: МММ "MigrationMarathonMode"

Рисунок 27 - Заявки на подбор

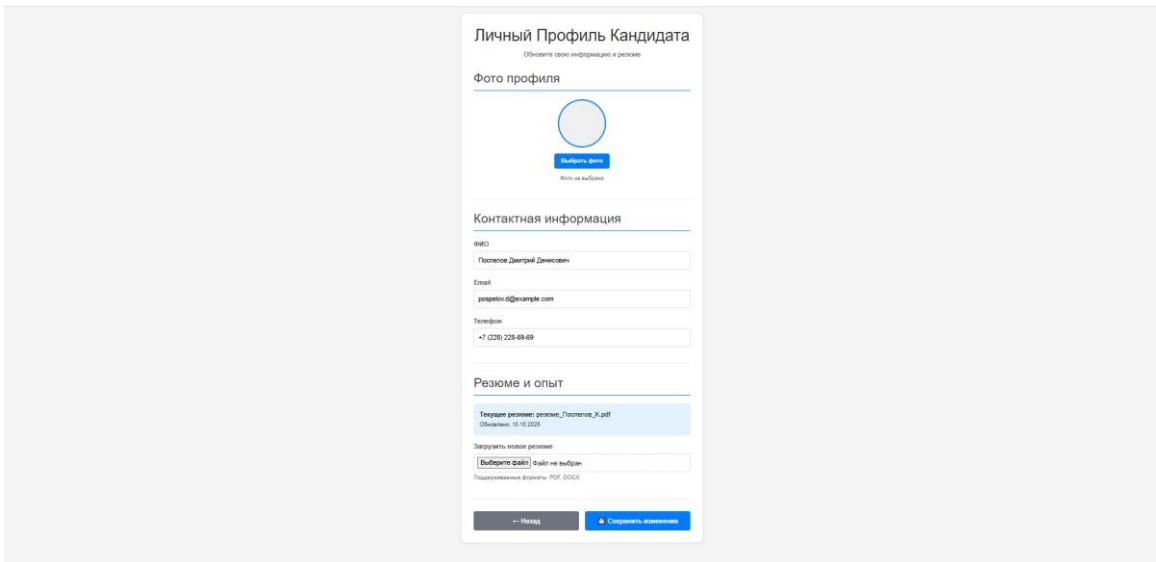


Рисунок 28 - Личный профиль кандидата

### 3.3.2. Диаграмма компонентов

Архитектура системы построена по модульному принципу. Диаграмма компонентов демонстрирует разделение на клиентскую часть (Client UI Web), серверную логику (Application Service, Auth Service) и внешние интеграции (External Parser).

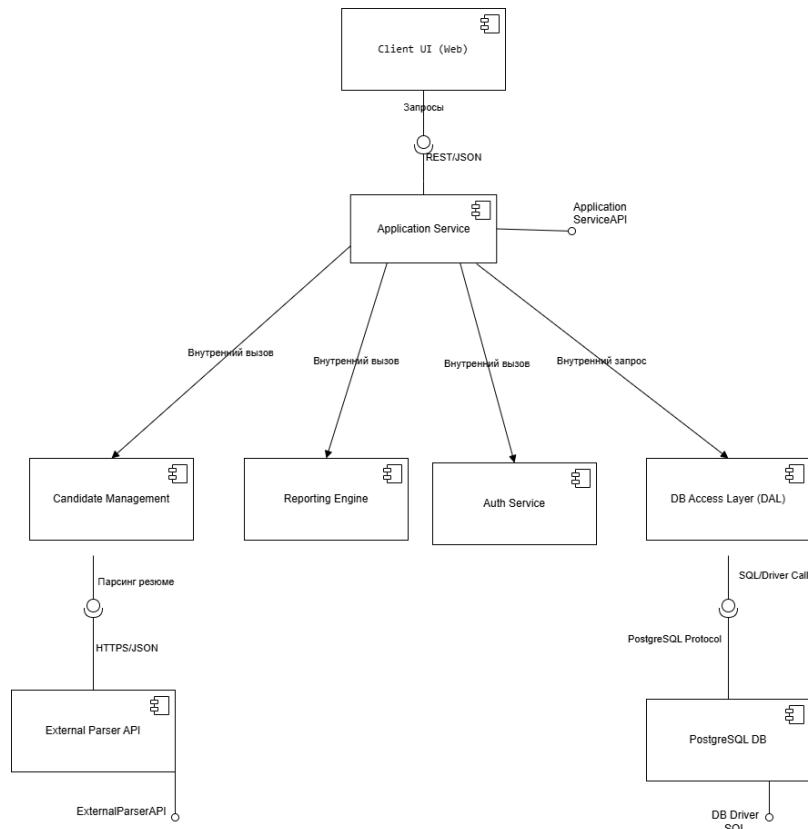


Рисунок 29 — Диаграмма компонентов системы

### 3.3.3. Диаграмма развертывания

Диаграмма развертывания описывает физическое размещение программных узлов на оборудовании. Используется трехуровневая архитектура: Web-сервер, Сервер приложений и Сервер базы данных.

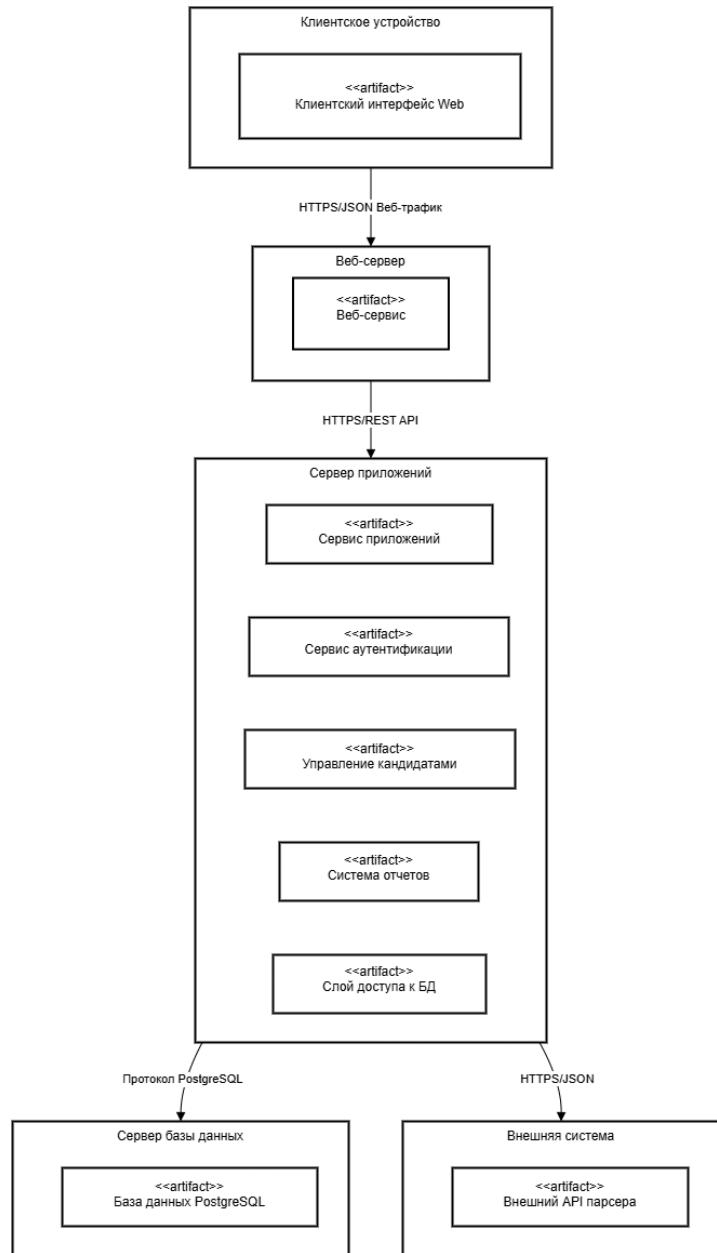


Рисунок 30 — Диаграмма развертывания (Deployment Diagram)

### 3.3.4. Коммуникационные интерфейсы

Для взаимодействия компонентов используются стандартные протоколы:

1. **HTTPS/TLS:** Шифрование трафика между клиентом и сервером.
2. **REST API (JSON):** Обмен данными между фронтендом и бэкендом, а также с внешним сервисом парсинга.
3. **PostgreSQL Protocol:** Связь сервера приложений с базой данных.
4. **SMTP:** Отправка почтовых уведомлений пользователям.

Полный перечень нефункциональных требований (производительность, надежность, безопасность) приведен в **Приложении 2**.

## **Заключение**

В ходе выполнения курсовой работы было проведено комплексное обоснование и разработка требований к программной системе поддержки деятельности агентства по подбору персонала «TalentBridge».

В теоретической части работы были изучены основные подходы к инженерии требований, классификация требований и стандарты их документирования.

В рамках предпроектного исследования был проведен анализ предметной области кадрового агентства. Выявлены ключевые недостатки текущих ручных процессов («As-Is»): высокая трудоемкость обработки резюме, разрозненность данных и отсутствие оперативной отчетности. Обоснована необходимость внедрения автоматизированной системы, которая позволит сократить рутинные операции и создать единое информационное пространство. Сравнительный анализ аналогов (E-Staff, Huntflow) подтвердил целесообразность собственной разработки для обеспечения гибкости и безопасности данных.

Практическим результатом работы стал полный пакет требований к системе «TalentBridge»:

**1. Пользовательские требования** зафиксированы в виде диаграммы вариантов использования и детализированных пользовательских историй.

**2. Функциональные требования** проработаны на уровне анализа и проектирования. Построены диаграммы классов, последовательности, коммуникации и деятельности, описывающие логику работы ключевых модулей (регистрация вакансий, управление базой кандидатов). Разработана ER-модель базы данных.

**3. Нефункциональные требования** определили архитектуру системы (диаграммы компонентов и развертывания), требования к интерфейсам, безопасности и производительности.

Разработанный комплект документации и моделей полностью готов к передаче команде разработчиков для реализации системы, что позволит достичь поставленных бизнес-целей агентства.

## **Список использованной литературы**

1. ГОСТ 34.602–2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. — Москва : Стандартинформ, 2020. — 26 с.
2. Ахмедова, Х. Г. Обоснование и разработка требований к программным системам : учебное пособие / Х. Г. Ахмедова. — Москва : РТУ МИРЭА, 2023. — 104 с. — ISBN 978-5-7339-1934-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/382694> (дата обращения: 22.09.2025).
3. Ахмедова, Х. Г. Обоснование и разработка требований к программным системам : учебно-методическое пособие / Х. Г. Ахмедова, А. В. Овсянникова, А. А. Бирюкова. — Москва : РТУ МИРЭА, 2023. — 68 с. — ISBN 978-5-7339-2010-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/398132> (дата обращения: 20.10.2025).
4. Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс, Д. Битти ; пер. с англ. — 3-е изд., доп. — Москва : БХВ-Петербург, 2014. — 736 с. — ISBN 978-5-9775-3348-5.
5. Брусникин, Г. Н. Разработка UML-моделей при проектировании информационных систем : учебное пособие / Г. Н. Брусникин, Н. Ю. Соколова. — Москва : МИЭТ, 2023. — 52 с. — ISBN 978-5-7256-1016-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/461570> (дата обращения: 20.10.2025).
6. Завьялов, А. В. Диаграммы UML для анализа и проектирования информационных систем : учебно-методическое пособие / А. В. Завьялов. — Москва : РТУ МИРЭА, 2021. — 65 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/218630> (дата обращения: 22.09.2025).

## **Приложения**

### **Приложение 1. Документ о концепции и границах**

#### **1. БИЗНЕС-ТРЕБОВАНИЯ**

**1.1 Исходные данные** В агентстве процессы подбора персонала выполняются вручную: заявки принимаются, данные о кандидатах фиксируются в разрозненных файлах, а взаимодействие осуществляется вручную. **Проблемы:** дублирование данных, ошибки в данных, неэффективное взаимодействие, отсутствие единой базы.

#### **1.2 Возможности бизнеса**

1. Централизованный учёт всех данных о кандидатах и клиентах.
2. Автоматизация рутинных операций (планирование, ввод данных).
3. Быстрое формирование отчётов.
4. Удобная отчётность для менеджмента.

#### **1.3 Бизнес-цели**

1. Повысить скорость и эффективность подбора.
2. Исключить ошибки, связанные с ручным вводом данных.
3. Создать единое информационное пространство для всех рекрутеров.

#### **1.4 Критерии успеха**

1. Сокращение времени на обработку заявки на 30%.
2. Отсутствие разрозненных баз данных.
3. Минимум ошибок при формировании отчётов.

#### **1.5 Положение о концепции проекта**

- 1. Название продукта:** Программная система поддержки деятельности агентства по подбору персонала.
- 2. Назначение продукта:** Система предназначена для автоматизации ключевых процессов агентства.
- 3. Целевая ценность:** Исключение ошибок; сокращение времени на подбор; централизованный учёт; повышение прозрачности процессов.
- 4. Ключевые пользователи:** Рекрутеры, менеджеры.

**5. Основные сценарии использования:** Приём заявки, обработка резюме, ведение базы, планирование встреч, формирование отчётов.

**6. Ограничения:** Первая версия охватывает только базовые процессы; расширенные функции и интеграции не входят в MVP; система ориентирована на одно агентство.

### **1.6 Бизнес-риски**

1. Сопротивление персонала новым процессам.
2. Технические сбои при внедрении (потеря данных).
3. Несоответствие требованиям по защите персональных данных.
4. Недостаток бюджета.

### **1.7 Предположения и зависимости**

1. Большинство заявок поступают по телефону/почте.
2. Система внедряется для одного агентства.
3. Минимальная интеграция: только локальная база данных и отчётность.

## **2. РАМКИ И ОГРАНИЧЕНИЯ ПРОЕКТА**

### **2.1 Основные функции**

1. Учёт и хранение данных о клиентах и кандидатах.
2. Управление вакансиями.
3. Ведение истории взаимодействий.
4. Формирование отчётов.

### **2.2 Объём первоначальной версии (MVP)**

<b>Функция</b>	<b>Входит в первую версию?</b>	<b>Комментарий</b>
Управление данными	Да	Регистрация и ведение баз данных
Управление вакансиями	Да	Учёт и контроль заявок
Учёт взаимодействий	Да	Хранение истории звонков/встреч

Формирование отчётов	Да	«Закрытые вакансии», «Эффективность рекрутеров»
----------------------	----	-------------------------------------------------

### 2.3 Объем последующих версий

Функция	Планируется позже?	Комментарий
Интеграция с сервисами	Да	HeadHunter, SuperJob
Автоматическая рассылка	Да	Рассылка резюме клиентам
Интерактивные дашборды	Да	Расширенная аналитика
Мобильное приложение	Да	Доступ для рекрутеров

## 3. БИЗНЕС-КОНТЕКСТ

### 3.1 Профили заинтересованных лиц

Заинтересованное лицо	Роль	Интересы	Влияние на проект
Рекрутер	Пользователь	Удобная работа, меньше рутинны	Высокое
Менеджер	Руководитель	Контроль загрузки, отчётность	Высокое
Клиент	Заказчик	Качество подбора, скорость	Среднее
Руководство агентства	Заказчик	Снижение издержек, повышение сервиса	Высокое

### **3.2 Приоритеты проекта**

<b>Приоритет</b>	<b>Значимость</b>
Создание единой базы	Высокая
Автоматизация рутинных операций	Высокая
Формирование отчётности	Средняя
Интеграция с внешними сервисами	Низкая (отложена)

### **3.3 Особенности развертывания**

1. Внедрение планируется в одном агентстве.
2. Платформа: веб-приложение или настольное приложение.
3. Пользователи: до 5-10 рекрутеров и 1-2 менеджера.

### **3.4. Пробелы в требованиях или излишние требования**

1. Система не охватывает процессы расчёта и бухгалтерии в первой версии.
2. Интеграция с внешними сервисами отложена на будущее.
3. Мобильное приложение для гостей — за рамками текущего проекта.

## **Приложение 2. Атрибуты качества**

### Внешние атрибуты качества

#### 1. Требования к доступности (НФ-ДОС)

**1. НФ-ДОС-001** Система должна быть доступна 24 часа в сутки, 7 дней в неделю, за исключением планового технического обслуживания, которое проводится в нерабочее время (например, с 02:00 до 04:00 по московскому времени).

**2. НФ-ДОС-002** Плановая доступность должна составлять 99.5% в месяц.

**3. НФ-ДОС-003** Время восстановления после некритического сбоя не должно превышать 30 минут.

#### 2. Требования к установке (НФ-УСТ)

**1. НФ-УСТ-001** Клиентский интерфейс (Client UI Web) не требует установки и доступен через любой современный веб-браузер.

**2. НФ-УСТ-002** Установка серверных компонентов (на WebServer и AppServer) должна быть полностью автоматизирована с помощью скриптов (например, Docker/Ansible) и занимать не более 1 часа на чистой операционной системе.

#### 3. Требования к целостности (НФ-ЦЕЛ)

**1. НФ-ЦЕЛ-001** Система должна обеспечивать целостность транзакций (CRUD-операций) при работе с сущностями "Вакансия", "Кандидат" и "APPLICATION". Все записи в базе данных должны быть атомарными.

**2. НФ-ЦЕЛ-002** При неудачном парсинге резюме (ExternalParser), данные кандидата не должны сохраняться в БД до ручного подтверждения/коррекции рекрутером.

#### 4. Требования к совместимости (НФ-СОВ)

**1. НФ-СОВ-001** Система должна быть совместима с любой почтовой службой, использующей стандартный протокол SMTP для отправки уведомлений.

**2. НФ-СОВ-002** Сформированные отчеты должны быть совместимы и корректно открываться в программах для просмотра PDF и редактирования XLSX (например, Microsoft Excel или LibreOffice Calc).

5. Требования к производительности (НФ-ПРФ)

**1. НФ-ПРФ-001** Время отклика на запросы по поиску и фильтрации в Базе Кандидатов (по 5+ параметрам) не должно превышать 3 секунды при наличии в базе до 50 000 записей.

**2. НФ-ПРФ-002** Время загрузки Карточки Вакансии с полной информацией и списком откликов не должно превышать 2 секунды.

6. Требования к надежности (НФ-НАД)

**1. НФ-НАД-001** Среднее время наработки на отказ (MTBF) критически важных модулей (Application Service, DB Access Layer) должно составлять не менее 1000 часов.

**2. НФ-НАД-002** Система должна автоматически логировать все ошибки и исключения в специальный журнал для последующего анализа и предотвращения повторных сбоев.

7. Требования к устойчивости (НФ-УСТЧ)

**1. НФ-УСТЧ-001** Система должна использовать механизмы кэширования для снижения нагрузки на DBServer и обеспечивать быстрое восстановление после временного сбоя.

**2. НФ-УСТЧ-002** При временной недоступности ExternalParser, система должна предоставлять рекрутеру возможность ручного ввода данных резюме, не блокируя процесс.

8. Требования к защите (НФ-ЗАЩ)

**1. НФ-ЗАЩ-001** Модуль Auth Service должен обеспечивать строгую ролевую модель доступа (RBAC), гарантируя, что, например, Работодатель не может получить доступ к S7: Управление Пользователями.

**2. НФ-ЗАЩ-002** Пароли пользователей должны храниться в базе данных (PostgreSQL DB) в хешированном виде с использованием современных, устойчивых к взлому алгоритмов.

## 9. Требования к безопасности (НФ-БЕЗ)

**1. НФ-БЕЗ-001** Все персональные данные должны передаваться исключительно по шифрованным каналам (HTTPS/TLS).

**2. НФ-БЕЗ-002** Система должна быть устойчива к основным видам веб-атак (SQL Injection, XSS) и проходить ежеквартальный аудит безопасности.

## 10. Требования к удобству использования (НФ-УДОБ)

**1. НФ-УДОБ-001** Интерфейс должен быть интуитивно понятным, с единой стилистикой для всех порталов.

**2. НФ-УДОБ-002** Все основные операции (CRUD) с сущностями "Вакансия" и "Кандидат" должны быть доступны не более чем в трех кликах от главного экрана (Dashboard).

### Внутренние атрибуты качества

#### 1. Требования к эффективности (НФ-ЭФФ)

**1. НФ-ЭФФ-001** Загрузка процессора на AppServer не должна превышать 70% при пиковой нагрузке, чтобы обеспечить запас мощности.

**2. НФ-ЭФФ-002** Объем занимаемой памяти серверными процессами должен контролироваться и не приводить к частым операциям подкачки (swapping).

#### 2. Требования к возможности модификации (НФ-МОД)

**1. НФ-МОД-001** Система должна быть спроектирована с использованием принципа слабой связанности модулей, чтобы изменение логики в Candidate Management не требовало изменений в Reporting Engine.

**2. НФ-МОД-002** Слои архитектуры (Client, Application, DAL) должны быть физически и логически отделены друг от друга.

#### 3. Требования к обслуживанию (НФ-ОБС)

**1. НФ-ОБС-001** Система должна вести подробные логи всех операций с указанием времени, пользователя и результата.

**2. НФ-ОБС-002** Должны быть предусмотрены административные инструменты для просмотра и ручного редактирования статусов сущностей (Вакансия, Кандидат) в случае сбоя.

#### 4. Требования к переносимости (НФ-ПЕР)

1. **НФ-ПЕР-001** Серверные компоненты должны быть контейнеризированы (например, с использованием Docker) для обеспечения быстрой миграции между различными облачными или локальными серверами (Linux, Windows Server).

#### 5. Требования к повторному использованию (НФ-ПОВ)

1. **НФ-ПОВ-001** Модули Auth Service и DB Access Layer DAL должны быть спроектированы как независимые библиотеки для потенциального повторного использования в других внутренних системах агентства.

#### 6. Требования к масштабируемости (НФ-МАС)

1. **НФ-МАС-001** Архитектура должна поддерживать горизонтальное масштабирование AppServer и WebServer (добавление новых инстансов) без изменения кода.

2. **НФ-МАС-002** Должен быть предусмотрен механизм балансировки нагрузки перед WebServer для равномерного распределения запросов.

#### 7. Требования к проверяемости (НФ-ПРВ)

1. **НФ-ПРВ-001** Для всех ключевых модулей (Candidate Management, Auth Service) должны быть разработаны модульные тесты с покрытием кода не менее 80%.

2. **НФ-ПРВ-002** Интеграционные тесты должны автоматически проверять корректность взаимодействия между Application Service и ExternalParser.