Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет**

**при Правительстве Российской Федерации»**

**(Финансовый университет)**

Кафедра информационных технологий

Пояснительная записка к курсовой работе по дисциплине

«Современные технологии программирования»

на тему:

«Информационно-справочная система Кадрового Агентства»

Выполнил:

Студент группы ДПИ22-2

Фром Никита Сергеевич

Научный руководитель:

Авраменко Александр Дмитриевич

Ассистент

Москва – 2025

**Содержание**

[Введение 3](#_Toc191217748)

[1. Определение объекта и предмета исследования 6](#_Toc191217749)

[2. Актуальность автоматизации 8](#_Toc191217750)

[3. Постановка задачи 10](#_Toc191217751)

[4. Описание программы 12](#_Toc191217752)

[4.1. Алгоритмические решения 13](#_Toc191217753)

[4.2. Описание интерфейса 18](#_Toc191217754)

[4.3. Архитектура приложения 26](#_Toc191217755)

[5. Структура реализованных классов и их назначение в проекте 31](#_Toc191217756)

[Заключение 33](#_Toc191217757)

[Список литературы 35](#_Toc191217758)

# **Введение**

В современном мире информационные технологии играют ключевую роль во всех сферах деятельности, включая управление персоналом. Автоматизация кадровых процессов становится необходимым инструментом для повышения эффективности работы организаций. Курсовая работа, посвященная разработке информационно-справочной системы кадрового агентства, направлена на создание программного продукта, способного оптимизировать процедуры подбора кандидатов, управления сотрудниками и координации работы различных отделов компании. Актуальность разработки системы обусловлена необходимостью повышения оперативности принятия решений в условиях роста объема информации и ужесточения конкуренции на рынке труда, что требует внедрения современных технологий для автоматизации рутинных операций [1].

Целью данной курсовой работы является создание информационно-справочной системы, обеспечивающей эффективное управление кадровыми процессами в агентстве. Система предусматривает разграничение доступа для различных категорий пользователей, что позволяет обеспечить безопасность данных и удобство работы. Пользователь входит в систему посредством страницы входа, где вводит логин и пароль, а регистрация новых пользователей не предусмотрена – добавление пользователей осуществляется исключительно администратором, который имеет полномочия для регистрации, назначения ролей и изменения доступа. В системе реализованы различные роли, каждая из которых выполняет свои специфические функции. Администратор имеет полный доступ ко всем функциям системы, что позволяет ему контролировать информационные потоки и осуществлять регистрацию пользователей. Роль отдела кадров ориентирована на подбор кандидатов и работу с резюме, что способствует оперативному ведению учета потенциальных сотрудников. Руководитель отдела отвечает за управление работой соответствующего подразделения, а генеральный директор имеет возможность получать сводные отчеты и контролировать процессы в организации, за исключением доступа к панели администратора. Дополнительно реализованы роли руководителя проекта, менеджера команды и маркетинговой команды, что позволяет организовать эффективное взаимодействие между участниками процесса: руководитель проекта занимается планированием и реализацией проектов, менеджер команды координирует работу внутри групп, а маркетинговая команда отвечает за продвижение услуг агентства и анализ рыночных тенденций [2].

Разрабатываемая система построена с учетом современных требований к безопасности и удобству использования. Четкое разграничение прав доступа минимизирует риск несанкционированного доступа к конфиденциальной информации и обеспечивает высокую скорость обработки данных. Автоматизация рутинных операций позволяет существенно снизить операционные затраты и повысить общую эффективность работы кадрового агентства, предоставляя возможность хранения и обработки информации в режиме реального времени. Такой подход способствует не только оптимизации текущих процессов, но и закладывает основу для дальнейшего развития системы, позволяя интегрировать новые функциональные модули и адаптировать продукт под специфические требования различных организаций.

Для реализации системы были выбраны современные технологии, обеспечивающие высокое качество и надежность программного продукта. Серверная часть реализована с использованием фреймворка Spring, который позволяет создавать масштабируемые веб-приложения с поддержкой архитектуры MVC. Сборка проекта производится с помощью Maven, что упрощает управление зависимостями и автоматизирует процесс сборки. В качестве базы данных используется MySQL, обладающая высокой производительностью и стабильностью при обработке больших объемов информации. В качестве среды разработки применяется Intellij IDEA, предоставляющая широкий набор инструментов для эффективного создания и отладки программного обеспечения. Для формирования динамических веб-страниц используются технологии thymeleaf и JSP, что позволяет создавать гибкие и адаптивные пользовательские интерфейсы, отвечающие современным требованиям веб-разработки [6].

Разработка информационно-справочной системы кадрового агентства представляет собой актуальное и востребованное направление в области информационных технологий и управления персоналом. Внедрение данной системы позволит оптимизировать кадровые процессы, обеспечить прозрачное взаимодействие между различными подразделениями агентства и повысить общую эффективность работы организации. Интеграция современных технологий гарантирует высокую производительность и надежность продукта, что делает его перспективным инструментом для автоматизации управленческих процессов в условиях динамично изменяющейся деловой среды.

Система спроектирована с учетом необходимости разграничения доступа, что является важным аспектом безопасности информации. Каждый пользователь, входящий в систему, получает доступ к функциям, соответствующим его роли. Это позволяет не только оптимизировать рабочие процессы, но и минимизировать риск несанкционированного доступа к конфиденциальной информации. Благодаря четкому разграничению функционала между различными ролями, системой можно эффективно управлять на всех уровнях – от кадрового отдела до руководства компании.

1. **Определение объекта и предмета исследования**

Объектом исследования данной курсовой работы является кадровое агентство как комплексная организационная система, объединяющая в себе множество взаимосвязанных процессов управления персоналом. Это не просто отдельное подразделение, а целостная структура, включающая в себя отдел кадров, управленческие звенья, а также специализированные подразделения, ответственные за подбор, обучение и мотивацию сотрудников. Кадровое агентство функционирует в условиях постоянно меняющейся деловой среды, где оперативность принятия решений, качество подбора кандидатов и эффективное взаимодействие между различными участниками процесса становятся ключевыми факторами успеха организации. Именно поэтому объектом исследования выступает не только сама структура кадрового агентства, но и все информационные и административные процессы, протекающие в рамках его деятельности, включая обработку данных, организацию работы с резюме, планирование вакансий и взаимодействие между различными подразделениями.

Предметом исследования является разработка, проектирование и внедрение информационно-справочной системы, предназначенной для автоматизации и оптимизации процессов кадрового агентства. В фокусе исследования находится создание программного продукта, который позволяет обеспечить централизованное хранение и оперативное обновление информации, разграничение прав доступа для пользователей с различными ролями, а также обеспечение высокого уровня безопасности данных. Разрабатываемая система включает в себя реализацию функционала для таких категорий пользователей, как администратор, отдел кадров, руководитель отдела, генеральный директор, руководитель проекта, менеджер команды и маркетинговая команда. Каждая из этих ролей имеет четко определенные полномочия и задачи, что позволяет не только повысить эффективность управления, но и создать удобный интерфейс для взаимодействия внутри организации.

Особое внимание в предмете исследования уделяется анализу современных технологий, используемых для реализации информационно-справочной системы. В работе рассматриваются возможности платформы Intellij IDEA для разработки, система сборки Maven, фреймворк Spring для создания серверной части приложения, база данных MySQL для надежного хранения информации, а также технологии thymeleaf и JSP для формирования динамичных и адаптивных веб-интерфейсов. Использование этих инструментов позволяет обеспечить масштабируемость, надежность и высокую производительность программного продукта, что в свою очередь способствует автоматизации рутинных кадровых операций и снижению операционных затрат.

Разработка информационно-справочной системы для кадрового агентства становится важным шагом в направлении совершенствования управленческих процессов и повышения конкурентоспособности организации. Применение современных программных решений и архитектурных подходов позволяет не только оптимизировать существующие процессы, но и заложить фундамент для дальнейшего развития системы, предусматривающей интеграцию новых модулей и адаптацию под специфические требования рынка труда. Исследование направлено на выявление наиболее эффективных методов организации работы кадрового агентства с использованием информационных технологий, а также на оценку потенциала автоматизации для улучшения качества кадровых решений.

Изучение кадрового агентства как объекта исследования в совокупности с разработкой информационно-справочной системы, являющейся предметом исследования, позволяет получить целостное представление о современных тенденциях в области управления персоналом и интеграции информационных технологий в административную деятельность.

1. **Актуальность автоматизации**

Автоматизация процессов в современном бизнесе приобретает все большую актуальность ввиду стремительного развития информационных технологий и увеличения объема обрабатываемых данных. В условиях глобализации и постоянного технологического прогресса ручное выполнение рутинных операций зачастую оказывается не только неэффективным, но и рискованным с точки зрения ошибок, потери времени и ресурсов. Особенно остро данный вопрос стоит в кадровых агентствах, где каждая минута и каждая ошибка могут повлиять на качество подбора персонала, своевременность принятия управленческих решений и, как следствие, на конкурентоспособность компании на рынке труда.

В условиях высококонкурентной деловой среды, где каждая организация стремится оптимизировать свои внутренние процессы, автоматизация становится неотъемлемой частью стратегии развития. Применение автоматизированных систем позволяет существенно сократить время обработки информации, устранить человеческий фактор при выполнении стандартных операций и повысить точность выполнения задач. Например, автоматизированная информационно-справочная система кадрового агентства способна обеспечить оперативное обновление базы данных кандидатов, мгновенный доступ к необходимой информации для различных подразделений и эффективное распределение задач между сотрудниками. Это, в свою очередь, приводит к снижению операционных затрат, повышению производительности труда и ускорению принятия решений на всех уровнях управления.

Важным аспектом актуальности автоматизации является повышение уровня безопасности данных. В современном мире информационные утечки и несанкционированный доступ к конфиденциальной информации представляют серьезную угрозу для любой организации. Автоматизированные системы, оснащенные средствами контроля доступа и мониторинга, позволяют не только защитить важные данные, но и обеспечить их корректное обновление и архивирование. Это особенно критично для кадровых агентств, где хранится большое количество персональных данных сотрудников и кандидатов, требующих особой защиты.

Автоматизация способствует унификации и стандартизации бизнес-процессов. Это позволяет не только поддерживать высокий уровень качества оказываемых услуг, но и легко адаптироваться к изменениям внешней среды или внутренним перестройкам в организации. Применение современных технологий, таких как системы управления базами данных, фреймворки для разработки веб-приложений и инструменты для создания динамичных пользовательских интерфейсов, делает возможным создание гибких и масштабируемых решений. Такие системы способны интегрироваться с другими информационными ресурсами предприятия, что открывает новые возможности для анализа и прогнозирования рабочих процессов.

Автоматизация является важнейшей стратегической задачей современного кадрового агентства, позволяющей повысить эффективность управления, сократить временные и финансовые затраты, а также обеспечить надежную защиту данных. Внедрение автоматизированной информационно-справочной системы позволяет не только оптимизировать текущие операции, но и закладывать фундамент для дальнейшего технологического развития, что особенно актуально в условиях динамично изменяющейся деловой среды и стремительного роста информационных потоков.

1. **Постановка задачи**

В рамках данной курсовой работы планируется разработка информационно-справочной системы кадрового агентства, которая позволит автоматизировать ключевые процессы управления персоналом и повысить эффективность работы организации. Будущая система будет интегрировать современные технологии для создания удобного, безопасного и масштабируемого продукта, способного удовлетворить потребности различных подразделений кадрового агентства.

Главной целью проекта является обеспечение оперативного и надежного управления информационными потоками, что позволит снизить операционные затраты, минимизировать человеческий фактор и улучшить качество подбора персонала. Система будет включать функционал для разграничения прав доступа пользователей с разными ролями, интеграцию с базой данных для хранения информации, а также инструменты для быстрого поиска и редактирования данных. Для реализации проекта планируется использование таких технологий, как Spring, Maven, MySQL, thymeleaf и JSP, что позволит создать современное веб-приложение, отвечающее требованиям безопасности и производительности в условиях динамично меняющейся деловой среды [10].

Поставленные задачи в рамках курсовой работы:

* Реализовать функционал регистрации сотрудников с выдачей ролей, осуществляемый администратором.
* Разработать страницу входа, обеспечивающую авторизацию пользователей посредством логина и пароля.
* Подключить базу данных MySQL для хранения и обработки всей информации системы.
* Создать страницу кандидатов с возможностью управления этапами собеседования.
* Обеспечить на всех страницах функциональность поиска и редактирования данных.
* Реализовать страницу управления командами, позволяющую добавлять команды и связывать их с соответствующими отделами.
* Создать страницу управления должностями с возможностью добавления, удаления и обновления информации о должностях.
* Разработать страницу управления отделами, где предусмотрены функции добавления, удаления и редактирования отделов, а также привязка команд к отделам.
* Создать страницу вакансий, позволяющую добавлять вакансии, прикреплять к ним соискателей, управлять вакансиями и формировать отчеты по их статусу.
* Реализовать страницу для управления ресурсами набора персонала, где будет вестись учет информации о вакансиях, компаниях и кадровых потребностях.
* Разработать страницу маркетинга, на которой можно будет просматривать статистику по вакансиям и анализировать эффективность маркетинговых кампаний.
* Создать страницу аккаунта, где пользователи смогут редактировать информацию о себе в соответствии со своими правами доступа.

Реализация данных задач позволит создать комплексную информационно-справочную систему, способную обеспечить оптимизацию кадровых процессов и повышение общей эффективности работы кадрового агентства.

1. **Описание программы**

В главе рассмотрены ключевые алгоритмические решения, лежащие в основе корректного формирования, обновления и обработки информации о кандидатах, этапах собеседования, вакансиях и управлении командной структурой. Описываются основные функциональные модули системы, включающие управление учетными записями сотрудников и регистрацию новых пользователей с выдачей ролей, а также модули для работы с данными по вакансиям, кандидатам и структурированию подразделений. Административный функционал системы позволяет регулировать доступ пользователей, контролировать внесение изменений и вести журнал активности, что обеспечивает высокий уровень безопасности и прозрачности процессов.

Особое внимание уделено архитектуре системы, построенной по принципам модульности и масштабируемости. Такой подход позволяет не только обеспечить высокую производительность и надежность работы приложения, но и создать возможность для дальнейшего развития системы, интеграции с другими информационными ресурсами предприятия и адаптации под специфические требования кадрового агентства. Пользовательский интерфейс был разработан с учетом различных ролей пользователей – от специалистов отдела кадров, непосредственно работающих с данными о кандидатах и вакансиях, до руководителей отделов и администраторов, ответственных за контроль и поддержку системы. Это обеспечивает удобство использования, оперативный доступ к необходимой информации и возможность быстрого реагирования на изменения в организационной структуре.

* 1. **Алгоритмические решения**

Алгоритмическое решение, направленное на обеспечение безопасности хранения паролей в информационно-справочной системе кадрового агентства. Одной из важнейших задач при разработке современных информационных систем является защита учетных данных пользователей, и для этого применяется шифрование паролей с использованием алгоритма BCrypt. BCrypt представляет собой криптографический хэш-функционал, который генерирует хэш-значение исходного пароля с добавлением случайной соли и выполнением множества итераций, что делает процесс обратного восстановления исходного пароля практически невозможным [9].

Алгоритм BCrypt, реализуемый в библиотеке Spring Security, позволяет преобразовывать пароли в зашифрованный формат, который можно безопасно хранить в базе данных. При аутентификации пользователя введенный пароль также проходит через процесс хэширования, и полученный результат сравнивается с ранее сохраненным хэш-значением [7]. Если результаты совпадают, это означает, что введенный пароль корректен, и пользователю предоставляется доступ к системе. Алгоритм шифрования представлен на рисунке 1.

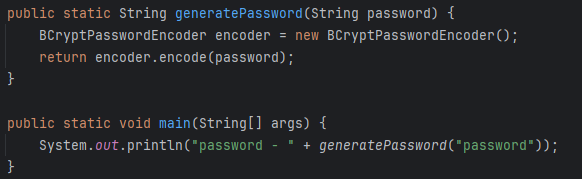


Рисунок 1 – Алгоритм шифрования

В представленном фрагменте класса PasswordGenerator основное внимание уделено методу generatePassword, который принимает строку – исходный пароль, и возвращает его зашифрованное представление. Внутри метода создается объект BCryptPasswordEncoder, предоставляемый библиотекой Spring Security. С помощью метода encode() происходит преобразование исходного пароля в хэш-значение, которое затем возвращается как результат работы метода.

Следующее алгоритмическое решение, представленное на рисунке 2 предназначено для обеспечения конфиденциальности данных посредством реализации шифрования и дешифрования с использованием алгоритма AES. Подобный подход позволяет защитить чувствительную информацию, хранящуюся или передаваемую в системе, от несанкционированного доступа.

Основным классом, реализующим данное решение, является Encryptor. Он включает в себя несколько ключевых методов: setKey, encrypt и decrypt.

Метод setKey предназначен для подготовки секретного ключа, который будет использоваться для шифрования и дешифрования данных. При вызове данного метода переданная строка сначала преобразуется в массив байтов с использованием кодировки UTF-8. Затем с помощью алгоритма SHA-1 вычисляется хэш от этого массива, который обрезается до 16 байт – необходимой длины для использования в алгоритме AES. Итоговый массив байтов используется для создания объекта SecretKeySpec с указанием алгоритма «AES». Такой подход гарантирует, что даже если исходная строка не соответствует требуемой длине, она будет преобразована в корректный ключ для шифрования.

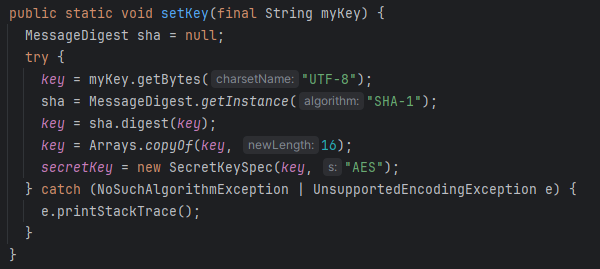


Рисунок 2 – Метод setKey

Алгоритмическое решение, представленное на рисунке 3 и в классе CheckStatus, предназначено для автоматического определения следующего этапа в процессе отбора кандидатов. Основная идея заключается в том, что для каждого текущего статуса кандидата определяется его позиция в заранее определенном списке стадий, после чего возвращается следующий элемент этого списка. Такой подход позволяет автоматизировать переход кандидата от одного этапа к другому, что особенно полезно при управлении большим количеством заявок и необходимости оперативного обновления их статуса.

В реализации используется метод getApplicantStatus, который принимает строковое значение текущего статуса и создает список возможных статусов, включающий этапы от начального состояния «None» до финального «Hired». С помощью метода indexOf определяется индекс текущего статуса, а затем возвращается элемент, находящийся непосредственно после него, что соответствует следующему этапу. В случае, если текущий статус уже последний в списке или отсутствует, алгоритм может потребовать доработки для обработки подобных ситуаций.



Рисунок 3 – Метод CheckStatus

Алгоритмическое решение, представленное на рисунке 4 и в классе GetLocalTime, предназначено для получения текущей локальной даты и времени в удобном для отображения формате. Решение использует API java.time, который предоставляет современные и удобные инструменты для работы с датой и временем в Java. С помощью класса LocalDateTime определяется текущий момент, а класс DateTimeFormatter используется для форматирования даты и времени в заданном виде. В данном случае формат установлен как «dd/MM/yyyy HH:mm:ss», что позволяет выводить дату в виде «день/месяц/год часы:минуты:секунды» [3].

Метод getTime() является статическим и возвращает строковое представление текущего времени в указанном формате.

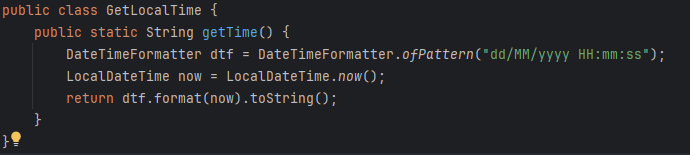


Рисунок 4 – Метод GetLocalTime

В этом фрагменте кода метод getTime() сначала создаёт объект DateTimeFormatter с заданным шаблоном, затем получает текущий момент времени с помощью LocalDateTime.now() и, наконец, форматирует этот момент в строку.

Алгоритмическое решение, реализованное в классе qrCodeGenerator и представленное на рисунке 5, предназначено для генерации QR-кода на основе заданного текстового значения. Для реализации данного функционала используется библиотека ZXing, которая предоставляет инструменты для кодирования информации в формате QR-кодов и последующего преобразования этой информации в графическое изображение.

Основной метод getQRCodeImage принимает три параметра: текст, который необходимо закодировать, а также ширину и высоту генерируемого изображения. Сначала создаётся объект QRCodeWriter, который отвечает за преобразование входного текста в матрицу, представляющую QR-код. Эта матрица содержит бинарное представление QR-кода, где каждый элемент указывает на необходимость закрашивания соответствующего пикселя.

Для формирования итогового изображения используется объект ByteArrayOutputStream, позволяющий записывать данные изображения в поток. Объект MatrixToImageConfig настраивает цветовую палитру: первый параметр задаёт цвет фона, а второй – цвет QR-кода. Затем метод MatrixToImageWriter.writeToStream записывает матрицу в поток в формате PNG, после чего полученные данные преобразуются в массив байт, который и возвращается как результат работы метода.

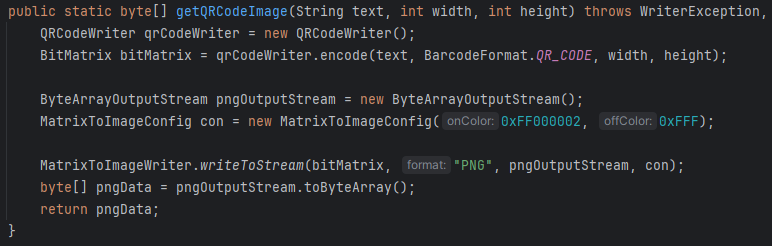


Рисунок 5 – Метод getQRCodeImage

Данное решение позволяет генерировать QR-коды, которые можно использовать для быстрого доступа к информации, проверки подлинности или встраивания в различные интерфейсы системы. Преобразование QR-кода в массив байт обеспечивает удобство хранения и передачи изображения в дальнейших процессах работы информационно-справочной системы кадрового агентства.

* 1. **Описание интерфейса**

Структура интерфейса построена на модульном подходе, что выражается в использовании отдельных компонентов для повторяющихся элементов, таких как навигационные панели, заголовки и подвал страницы. Например, в приведённом файле реализована вставка навигационных блоков с помощью директивы th:insert, что позволяет использовать один и тот же компонент на нескольких страницах без дублирования кода. Помимо этого, в интерфейсе активно используются формы для ввода и редактирования данных, где поля формы связываются с объектами модели через атрибуты th:field. Такой подход не только упрощает процесс разработки, но и повышает надежность взаимодействия между клиентской и серверной частями системы.

Применение Thymeleaf в сочетании с JSP-файлами способствует улучшенной организации кода, позволяя разработчикам легко поддерживать и расширять функциональность интерфейса. Динамическое формирование элементов, таких как списки выбора для отделов, команд, должностей и ресурсов, позволяет пользователям оперативно взаимодействовать с данными. Визуальные элементы, включая динамическую генерацию изображений, делают интерфейс более наглядным и удобным для конечных пользователей.

Интерфейс системы Recruiza представляет собой лаконичную и современную страницу авторизации, выполненную в минималистичном стиле. В центре экрана располагается карточка формы входа, содержащая заголовок с названием системы, два поля для ввода учетных данных – имени пользователя и пароля, а также кнопку "Войти". Дизайн формы выполнен в светлых тонах, фон содержит абстрактные графические элементы в синих оттенках, создавая легкую и профессиональную атмосферу. Ввод данных сопровождается тонкими линиями-указателями, а кнопка входа выделяется насыщенным синим цветом, подчеркивая основное действие. Весь интерфейс интуитивно понятен, не перегружен лишними элементами и ориентирован на удобство пользователя. Страница входа представлена на рисунке 6.

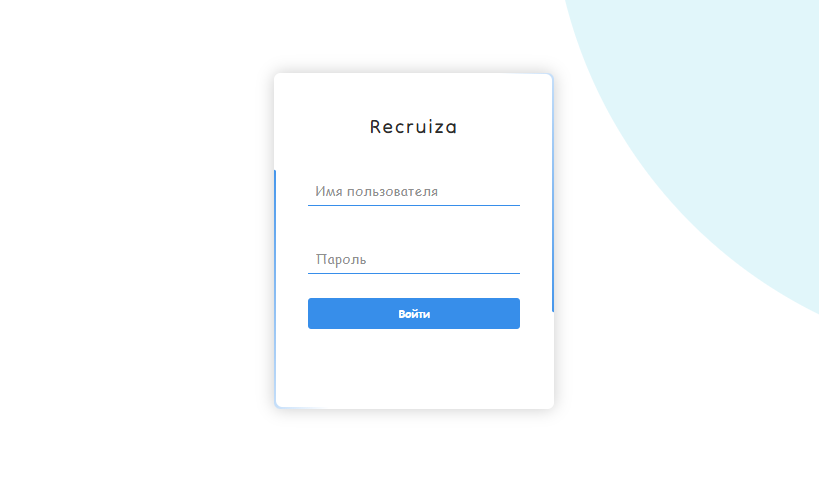


Рисунок 6 – Страница входа

На странице аккаунта, которая представлена на рисунке 7 отображается информация о пользователе, включая UserID, имя пользователя, email, номер телефона и роль в системе. Данные представлены в виде полей ввода, есть кнопка редактирования и QR-код. Ниже расположен раздел «История», где перечислены недавние действия пользователя с временными метками. Слева находится боковое меню с иконками для навигации между разделами.

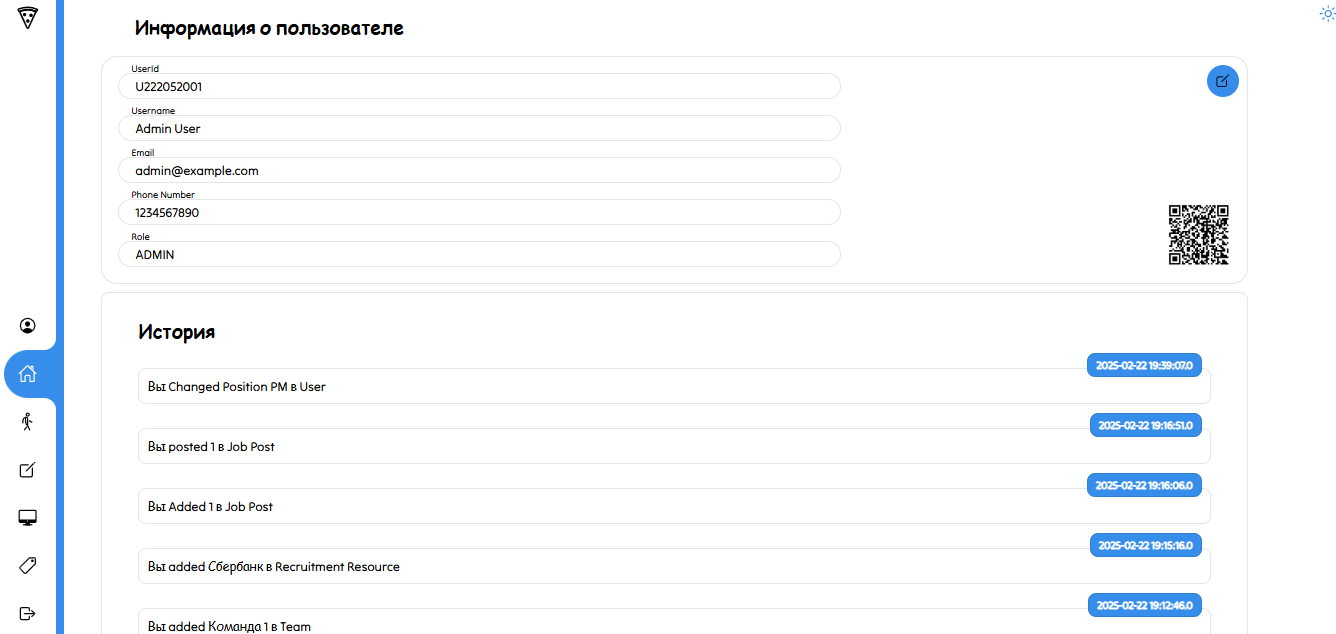


Рисунок 7 – Страница «Аккаунт»

На рисунке 8 представлены вкладки, которые доступны администратору, данные вкладки автоматический раскрываются при наведении.

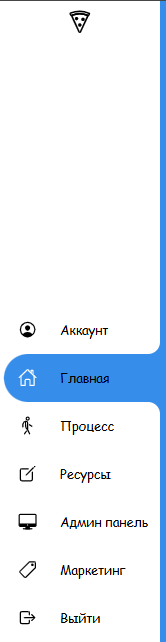


Рисунок 8 – Вкладки роли «Админ»

На главной странице отображается список вакансий. Каждая вакансия содержит статус, код, название, должность, комментарий, информацию о команде и отделе, даты начала и окончания, а также количество необходимых сотрудников. Главная страница представлена на рисунке 9.

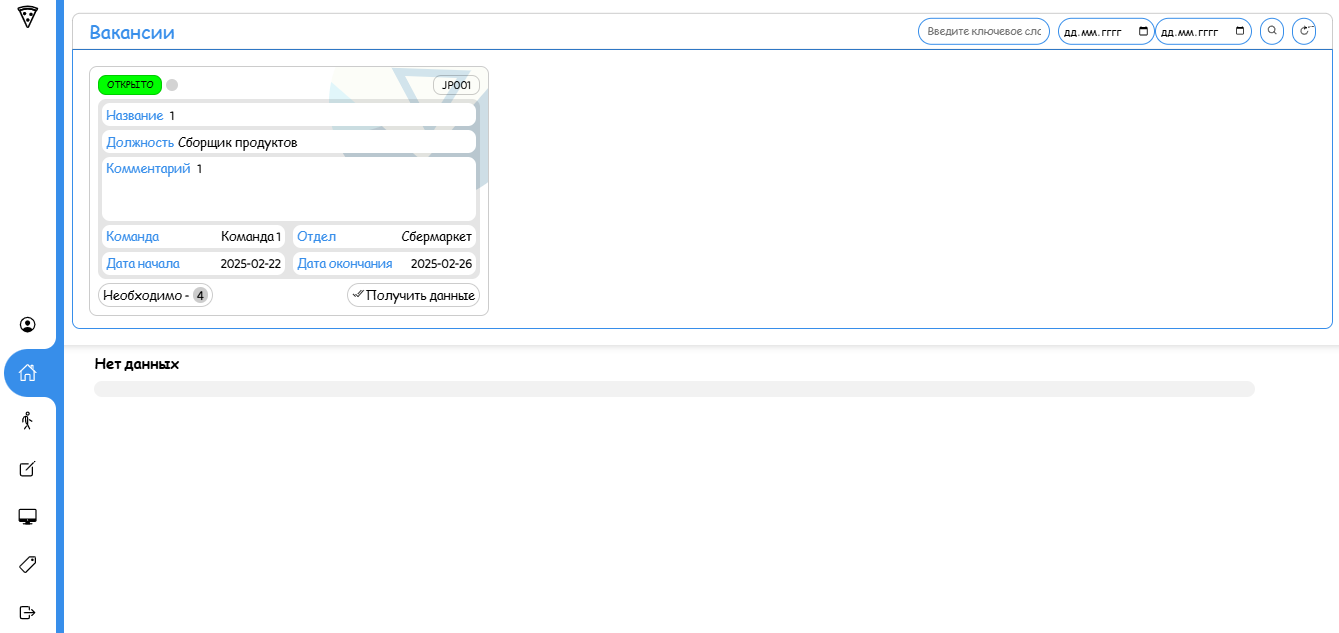


Рисунок 9 – Главная страница

В системе реализована возможность менять цвет информационно-справочной системы, кнопка находится сверху на странице аккаунт, смена цвета представлена на рисунке 10.



Рисунок 10 – Смена цвета

На вкладке процесс отображается процесс подачи заявки кандидатов. В верхней части представлены этапы процесса с количеством кандидатов на каждом этапе. Есть кнопки навигации между страницами списка кандидатов. Данная страница представлена на рисунке 11.

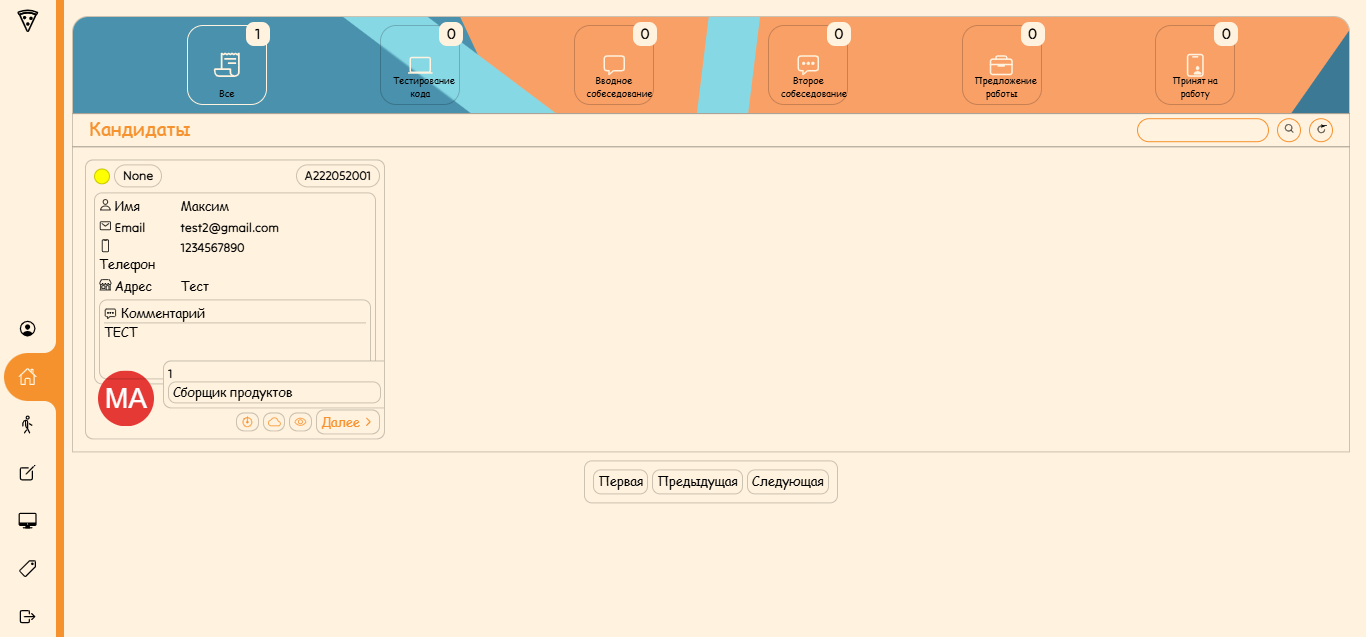


Рисунок 11 – Страница «Процесс»

На вкладке должность отображается список должностей с их кодами и названиями. Для каждой должности указано количество вакансий. Есть кнопки редактирования записей. В верхней части страницы доступны поле поиска, кнопки обновления данных и добавления новой должности. Внизу расположены элементы управления пагинацией. Справа находится переход между страницами в основной странице «Ресурсы». Страница «Должность» представлена на рисунке 12.

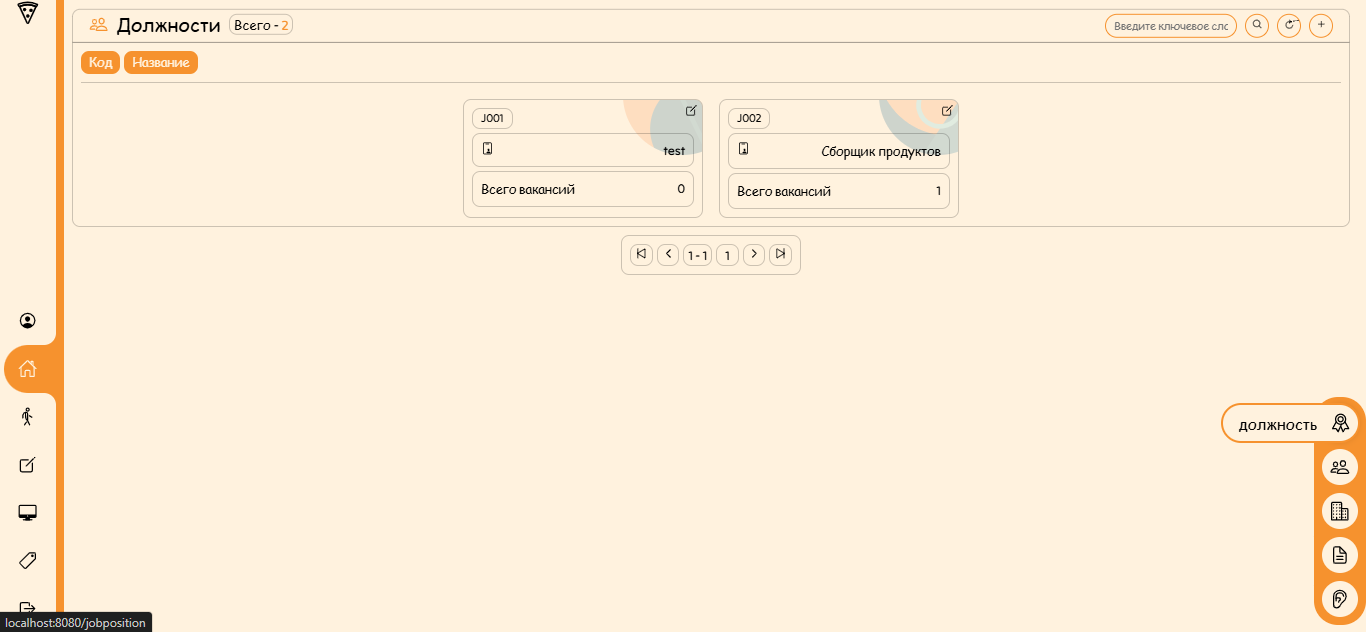


Рисунок 12 – Страница «Должность»

На странице «Команды» отображается список команд с их кодами, названиями и отделами. Для каждой команды есть кнопка редактирования. В верхней части страницы доступны поле поиска, кнопки обновления данных и добавления новой команды. Страница «Команды» представлена на рисунке 13

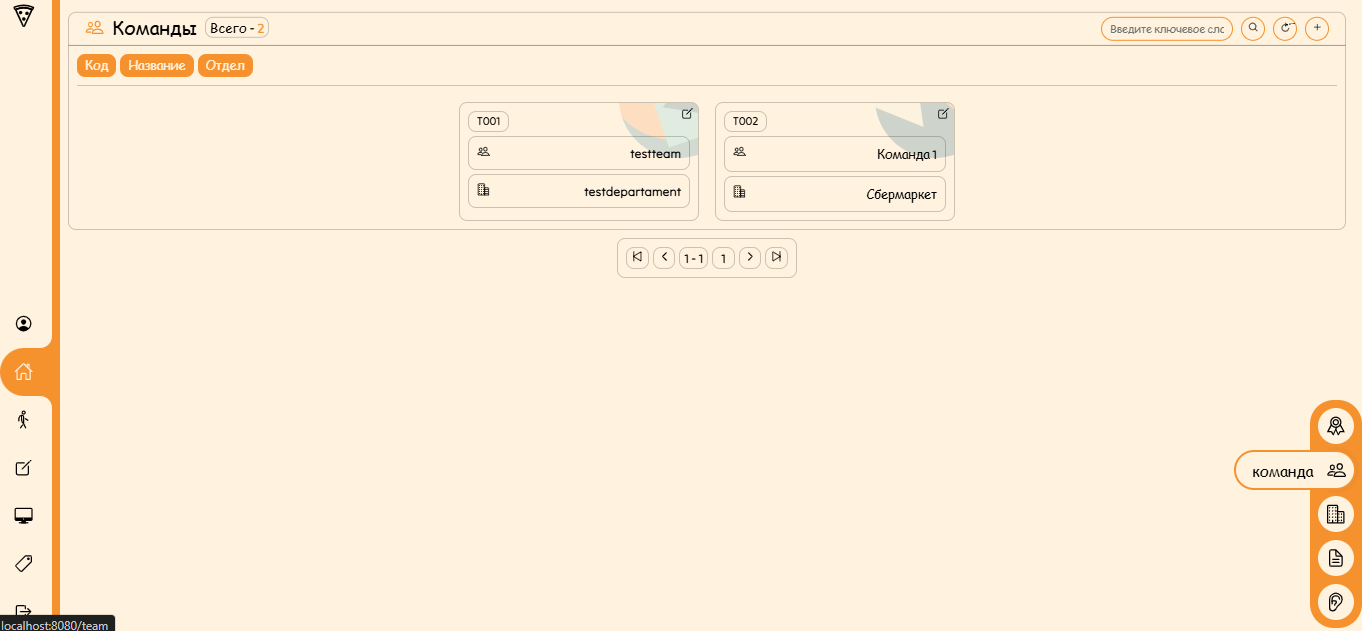


Рисунок 13 – Страница «Команды»

На странице «Отделы» отображается список отделов с их кодами и названиями. Для каждого отдела указано количество команд и есть кнопка добавления команды. Также доступна кнопка редактирования отдела. В верхней части страницы расположены поле поиска, кнопки обновления данных и добавления нового отдела. Страница «Отделы» представлена на рисунке 14.

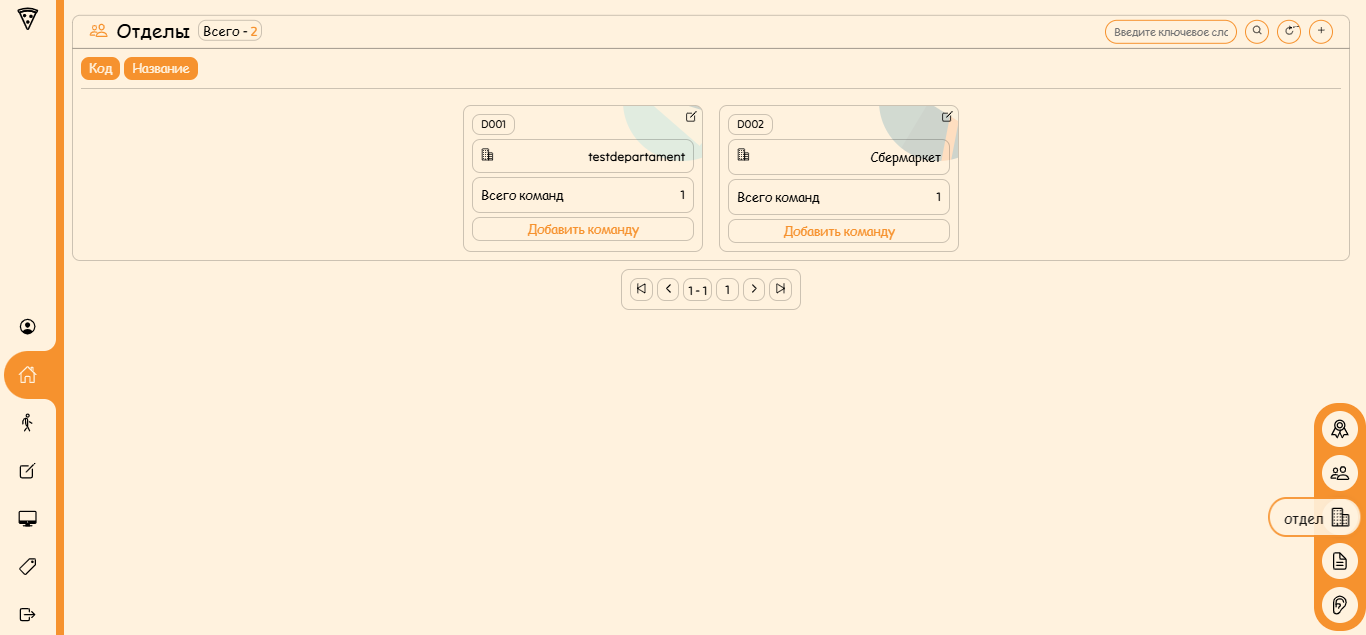


Рисунок 14 – Страница «Отделы»

На странице «Вакансии» отображается список вакансий с их статусами, кодами, названиями, должностями, комментариями, командами, отделами, датами начала и окончания, а также количеством необходимых сотрудников и соискателей. В верхней части представлены сводные данные по статусам вакансий с числовыми показателями и диаграммой. Доступны поле поиска, кнопки фильтрации, обновления данных и генерации отчета.

Страница «Вакансии» представлена на рисунке 15.

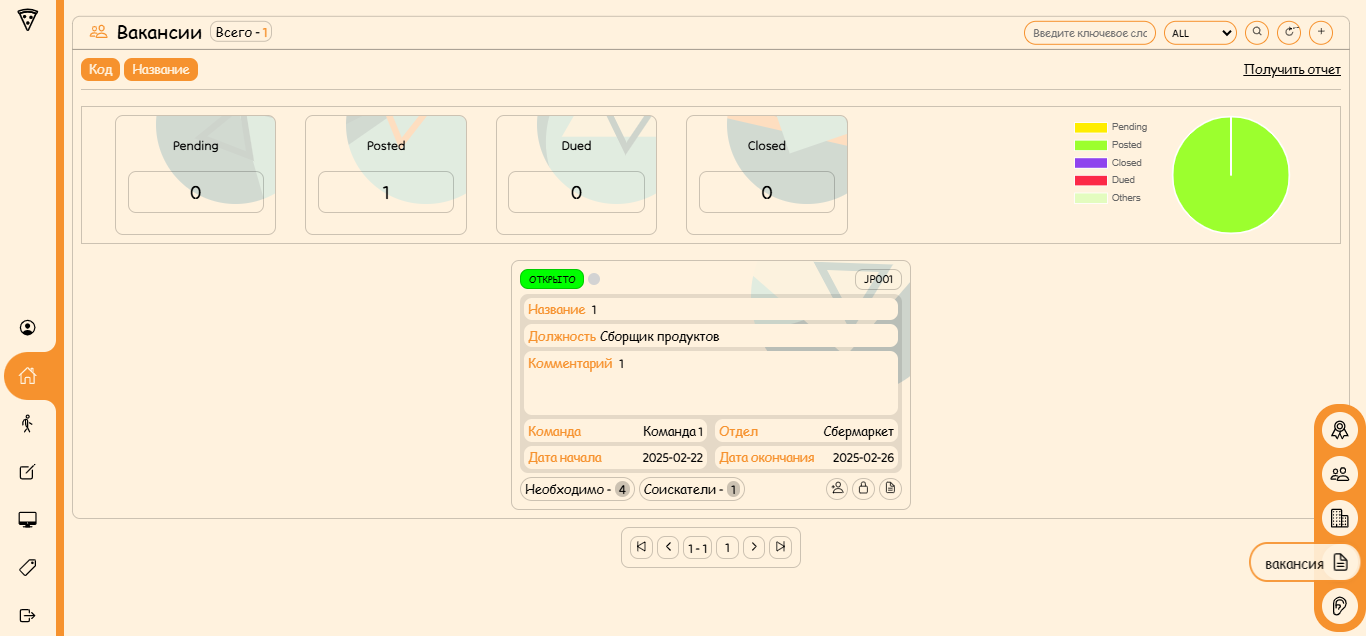


Рисунок 15 – Страница «Вакансии»

На странице «Ресурсы для набора персонала» отображается список ресурсов с их кодами, названиями, ссылками, контактными данными и категориями. В верхней части представлены сводные данные по типам ресурсов с числовыми показателями и диаграммой. Доступны поле поиска, кнопки обновления данных и добавления нового ресурса. Страница «Ресурсы для набора персонала» представлена на рисунке 16.

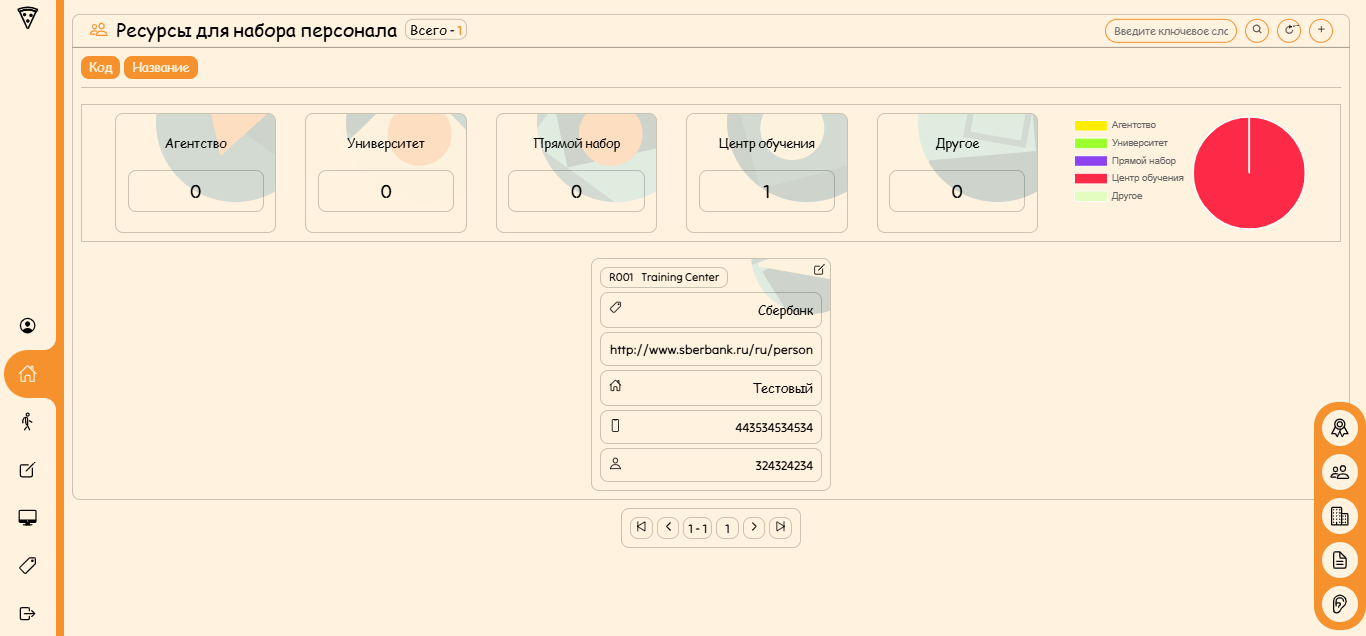


Рисунок 16 – Страница «Ресурсы для набора персонала»

На странице «Маркетинг» отображаются ожидающие и опубликованные вакансии. В разделе «Опубликованные вакансии» представлена сводная информация с числовыми показателями по статусам и диаграмма. Страница «Маркетинг» представлена на рисунке 17.

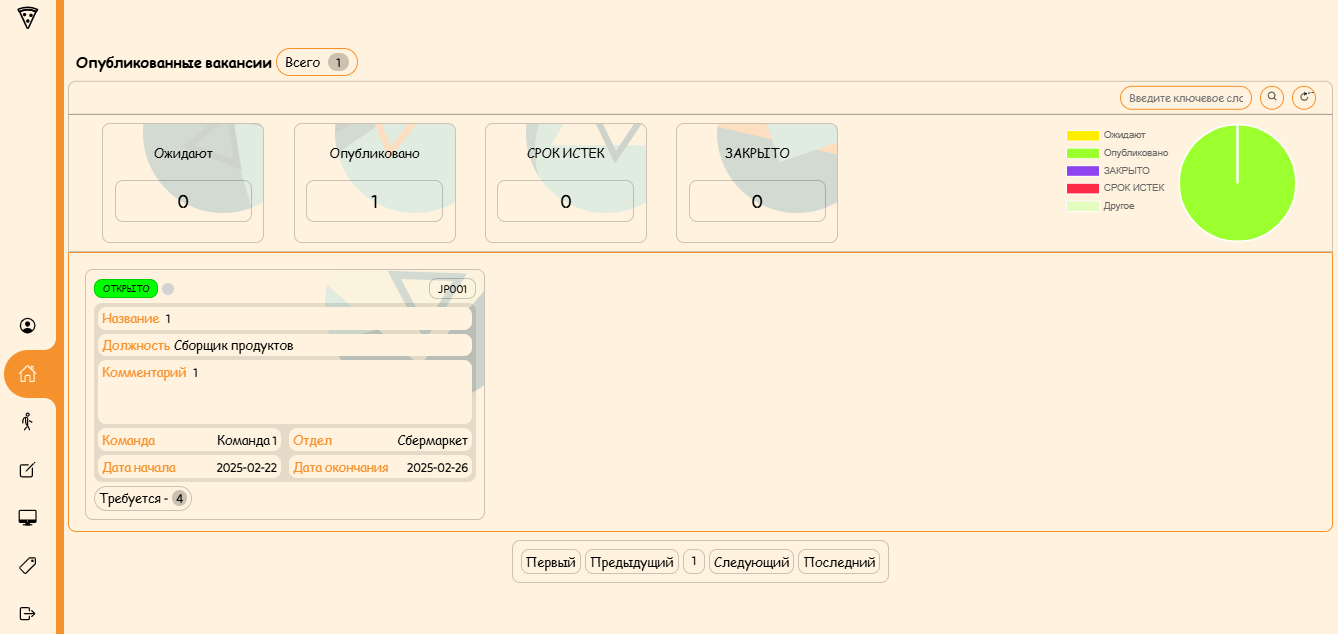


Рисунок 17 – Страница «Маркетинг»

На странице «Админ панель» отображается управление пользователями. В верхней части представлены сводные данные о количестве активных и неактивных пользователей. Для каждого пользователя доступны кнопки «Заблокировать» и «Изменить должность». В верхней части страницы расположены поле поиска, кнопки обновления данных и добавления нового пользователя. Страница «Админ панель» представлена на рисунке 18.

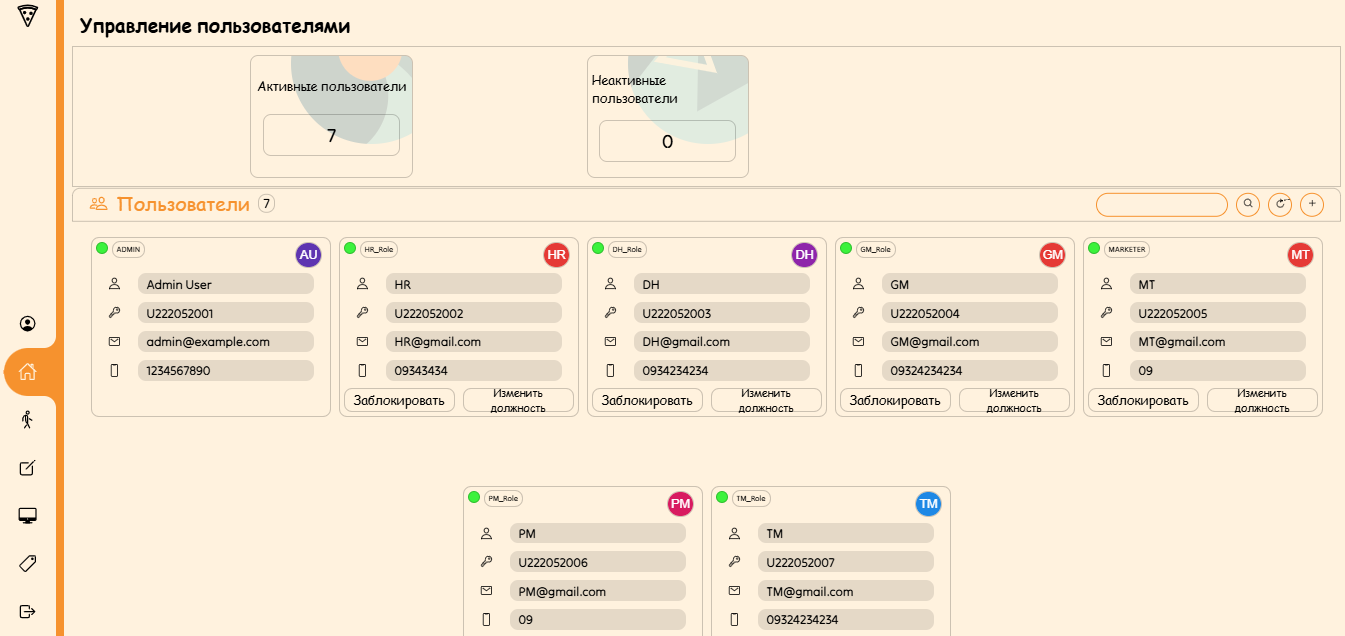


Рисунок 18 – Страница «Админ панель»

* 1. **Архитектура приложения**

Для реализации хранения и обработки данных в приложении используется система управления базами данных MySQL. Это решение позволяет обеспечить надежное, масштабируемое и быстрое хранение информации, необходимой для функционирования информационно-справочной системы кадрового агентства. MySQL выступает в качестве основного хранилища данных, где структурированы все ключевые сущности приложения, такие как кандидаты, вакансии, отделы, должности, команды, пользователи и история операций [5].

В базе данных реализована модель, основанная на взаимосвязанных таблицах. Каждая таблица отражает определенную сущность и содержит набор полей, характеризующих ее атрибуты. Связи между таблицами определяются внешними ключами, что позволяет обеспечить целостность данных и корректное взаимодействие различных модулей системы. Далее приводится описание основных таблиц базы данных с перечнем полей, их типов данных и связей. Схема базы данных представлена на рисунке 19.

Таблица «applicant» содержит информацию о кандидатах, подающих заявки. Помимо личных данных, здесь фиксируются сведения о статусе кандидата, дате подачи, комментариях и файлах, связанных с кандидатом, а также ссылки на вакансии и должности, к которым кандидат относится.

Таблица 1 – Таблица «applicant»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Значение поля |
| id | bigint | Primary Key |
| address | varchar(255) |  |
| code | varchar(255) | UNIQUE KEY |
| email | varchar(255) |  |
| mobile | varchar(255) |  |
| name | varchar(255) |  |
| status | varchar(255) | По умолчанию: Pending |
| apply\_time | timestamp | По умолчанию: CURRENT\_TIMESTAMP |
| comment | varchar(255) |  |
| current\_state | varchar(255) | По умолчанию: Code Test |
| file | longblob |  |
| link | varchar(255) |  |
| source | varchar(255) |  |
| job\_position\_id | bigint | Foreign Key |
| job\_post\_id | bigint | Foreign Key |

Таблица «department» хранит сведения об отделах кадрового агентства, позволяющие структурировать организацию по функциональным подразделениям.

Таблица 2 – Таблица «department»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Значение поля |
| id | bigint | Primary Key |
| code | varchar(255) | UNIQUE KEY |
| name | varchar(255) |  |

Таблица «history» отражает журнал действий, совершенных пользователями системы. Содержит информацию о том, какие операции выполнялись, над какими данными и когда, а также идентификатор пользователя, выполнившего действие.

Таблица 3 – Таблица «history»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Значение поля |
| id | bigint | Primary Key |
| action | varchar(255) |  |
| data | varchar(255) |  |
| data\_name | varchar(255) |  |
| created\_time | timestamp | По умолчанию: CURRENT\_TIMESTAMP |
| table\_name | varchar(255) |  |
| user\_id | bigint | Foreign Key |

Таблица «job\_position» содержит информацию о должностях, на которые претендуют кандидаты. Таблица позволяет централизованно управлять списком вакансий по позициям.

Таблица 4 – Таблица «job\_position»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Значение поля |
| id | bigint | Primary Key |
| code | varchar(255) | UNIQUE KEY |
| name | varchar(255) |  |

Таблица «job\_post» содержит сведения о вакансиях, опубликованных агентством. Здесь фиксируется информация о вакансии, включая комментарии, количество позиций, сроки, статус и связи с другими сущностями.

Таблица 5 – Таблица «job\_post»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Значение поля |
| id | bigint | Primary Key |
| comment | longtext |  |
| count | int |  |
| due\_date | date |  |
| foc | bit(1) |  |
| code | varchar(255) | UNIQUE KEY |
| created\_time | timestamp | По умолчанию: CURRENT\_TIMESTAMP |
| post\_date | date |  |
| name | varchar(255) |  |
| post\_status | varchar(255) |  |
| sheet\_id | varchar(255) |  |
| department\_id | bigint | Foreign Key |
| jobposition\_id | bigint | Foreign Key |
| resource\_id | bigint | Foreign Key |
| team\_id | bigint | Foreign Key |
| user\_id | bigint | Foreign Key |

Таблица «recruitement\_resource» содержит данные о ресурсах для набора персонала, включая контактную информацию, адрес, тип ресурса и дополнительные параметры. Эта таблица помогает управлять внешними источниками кандидатов.

Таблица 6 – Таблица «recruitement\_resource»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Значение поля |
| id | bigint | Primary Key |
| address | varchar(255) |  |
| contact\_person | varchar(255) |  |
| link | varchar(255) |  |
| recruitement\_type | varchar(255) |  |
| code | varchar(255) | UNIQUE KEY |
| created\_time | timestamp | По умолчанию: CURRENT\_TIMESTAMP |
| mobile | varchar(255) |  |
| name | varchar(255) |  |

Таблица «schedule» отвечает за хранение информации о расписании, заметках и запланированных событиях, связанных с кандидатами. Позволяет фиксировать временные метки проведения собеседований или других мероприятий.

Таблица 7 – Таблица «schedule»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Значение поля |
| id | bigint | Primary Key |
| notes | varchar(255) | По умолчанию: no notes |
| code | varchar(255) | UNIQUE KEY |
| name | varchar(255) |  |
| time | datetime |  |
| applicant\_id | bigint | Foreign Key |

Таблица «team» содержит сведения о командах, работающих в рамках отделов кадрового агентства. Позволяет структурировать сотрудников по группам для эффективного управления проектами и задачами.

Таблица 8 – Таблица «team»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Значение поля |
| id | bigint | Primary Key |
| code | varchar(255) |  |
| name | varchar(255) |  |
| department\_id | bigint | Foreign Key |

Таблица «user» хранит данные пользователей системы, включая информацию об учетных записях, ролях, статусе активности, контактные данные и пароли.

Таблица 9 – Таблица «user»

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Значение поля |
| id | bigint | Primary Key |
| enabled | bit(1) |  |
| last\_action | varchar(255) |  |
| password | varchar(255) |  |
| role | varchar(255) |  |
| token | varchar(255) |  |
| code | varchar(255) | UNIQUE KEY |
| created\_time | timestamp | По умолчанию: CURRENT\_TIMESTAMP |
| email | varchar(255) | UNIQUE KEY |
| mobile | varchar(255) |  |
| name | varchar(255) |  |
| status | varchar(255) | Значение по умолчанию: ACTIVE |

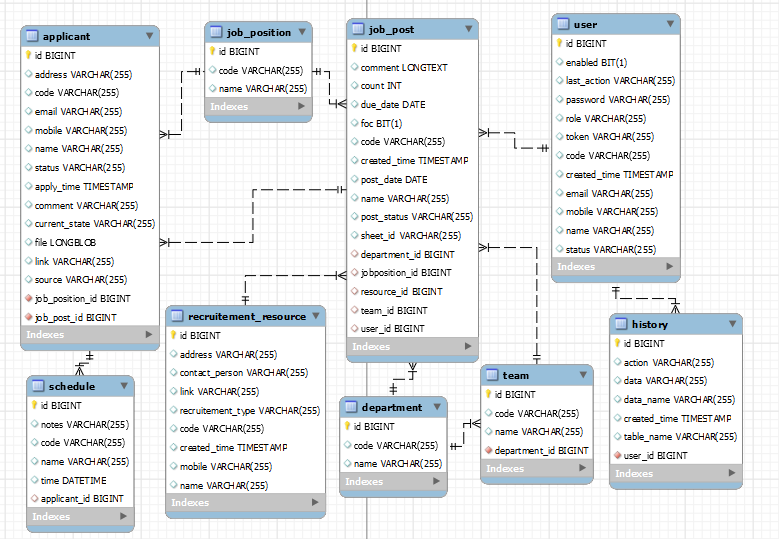


Рисунок 19 – Схема базы данных

1. **Структура реализованных классов и их назначение в проекте**

В проекте используется многослойная архитектура, в рамках которой каждый пакет выполняет определенную функциональную задачу. Пакет controller содержит классы, которые принимают входящие HTTP-запросы от пользователей и формируют соответствующие ответы. Эти классы вызывают методы сервисов, подготавливают данные для отображения и определяют, на какую страницу или ресурс необходимо перенаправить пользователя. В них сосредоточена логика маршрутизации и взаимодействия с клиентской частью приложения [8].

Пакет entity содержит классы-сущности, отражающие структуру таблиц в базе данных MySQL. Каждая сущность соответствует одной таблице и включает поля, связи и аннотации JPA для управления сохранением и загрузкой данных. Эти классы находятся на самом нижнем уровне логической структуры, поскольку являются фундаментом, на котором строится вся работа с данными.

В пакете model располагаются объекты, которые используются для передачи данных между различными слоями приложения, а также для связывания форм на страницах с серверной логикой. Здесь находятся так называемые DTO или Bean-классы, которые упрощают валидацию входных данных и изолируют доменные сущности от деталей представления. Такой подход повышает гибкость и удобство при работе с пользовательскими формами и внешними интерфейсами.

Пакет repository содержит интерфейсы, расширяющие JpaRepository и CrudRepository. Они предоставляют набор стандартных методов для чтения, записи, обновления и удаления записей в базе данных. Spring Data JPA автоматически генерирует реализации этих интерфейсов, что позволяет избежать написания большого количества шаблонного кода и упростить доступ к данным. Репозитории являются связующим звеном между сущностями и сервисами, обеспечивая абстракцию над слоями хранения данных [4].

В пакете securityConfig находятся классы, которые отвечают за безопасность приложения. Здесь определяются правила аутентификации и авторизации пользователей, настраивается шифрование паролей, а также указываются права доступа к ресурсам. В этом же месте инициализируются учетные записи и обрабатываются ситуации, связанные с неудачными попытками входа.

Слой бизнес-логики представлен в пакете service, где располагаются сервисы и их реализации. Они взаимодействуют с репозиториями, обрабатывают данные, выполняют необходимые проверки и формируют ответы, которые контроллеры передают пользователям. В сервисах сконцентрированы основные бизнес-правила, поэтому именно здесь осуществляется координация и управление процессами в приложении.

Пакет utils включает набор вспомогательных классов, которые решают универсальные задачи и не относятся непосредственно к бизнес-логике. Здесь находятся решения для шифрования, генерации QR-кодов, форматирования дат и времени, проверки статусов и другие вспомогательные функции.

# **Заключение**

В ходе выполнения курсовой работы была разработана информационно-справочная система кадрового агентства, которая позволяет автоматизировать процессы подбора персонала, управления вакансиями, структурирования данных о кандидатах и взаимодействия различных подразделений организации. Данная система реализована с использованием современных технологий, таких как Java, Spring Boot, Spring Data JPA, Thymeleaf, а также базы данных MySQL. Применение этих инструментов позволяет не только обеспечить высокую производительность и масштабируемость, но и создать интуитивно понятный пользовательский интерфейс, отвечающий современным требованиям по безопасности и удобству эксплуатации.

В ходе проектирования и реализации системы был проведен глубокий анализ предметной области, выявлены основные проблемы и потребности кадрового агентства, что позволило выработать эффективные решения для оптимизации работы с данными. Система обеспечивает централизованное хранение информации о кандидатах, вакансиях, должностях, отделах, ресурсах набора персонала, а также пользователях и их ролях. В результате интеграции этих данных создается единая информационная среда, способствующая улучшению взаимодействия между подразделениями и оперативному принятию управленческих решений. Реализация функционала по регистрации пользователей, разграничению прав доступа, управлению статусами кандидатов и другими задачами позволяет значительно сократить время обработки информации и снизить риск возникновения ошибок, связанных с ручным вводом данных.

Важным этапом работы стало обеспечение безопасности данных. Для защиты паролей пользователей применяется алгоритм BCrypt, который позволяет надежно зашифровать учетные данные, что минимизирует риск несанкционированного доступа. Дополнительно реализована функция шифрования с использованием алгоритма AES для защиты конфиденциальной информации, что подтверждает высокую степень безопасности системы. Применение этих криптографических методов является существенным преимуществом, поскольку позволяет соответствовать современным стандартам защиты данных и требованиям законодательства в области информационной безопасности.

Архитектурное решение проекта основывается на принципах модульности и многослойности, что позволяет не только упростить разработку и поддержку приложения, но и обеспечивает возможность его дальнейшего расширения. Каждый компонент системы – будь то контроллеры, сервисы, репозитории или утилитные классы – выполняет строго определенную роль, что способствует логическому разделению обязанностей и повышает надежность работы системы.

Система демонстрирует высокую гибкость в настройке и управлении информационными потоками. Использование Thymeleaf и JSP для формирования пользовательского интерфейса позволяет динамически обновлять данные, обеспечивая удобный и понятный интерфейс для различных категорий пользователей. Разработанные страницы для работы с вакансиями, кандидатами, отделами, командами и ресурсами для набора персонала позволяют оперативно вносить изменения, проводить поиск по базе данных и редактировать информацию в режиме реального времени.

В заключение можно отметить, что разработанная информационно-справочная система кадрового агентства является актуальным и востребованным инструментом для оптимизации управленческих процессов в современных организациях. Проект демонстрирует успешную интеграцию современных технологий и методологий разработки, что позволяет создать надежное, безопасное и масштабируемое решение. Полученные результаты подтверждают высокую практическую значимость проекта и открывают возможности для дальнейшего развития системы, интеграции новых функциональных модулей и адаптации под специфические требования различных организаций.

# **Список литературы**

1. Симачкова Людмила Александровна Автоматизация бизнес-процессов кадрового агентства // Вестник УлГТУ. 2009. №2 (46).
2. Иванов И.И. Информационные технологии в управлении персоналом: современные подходы // Журнал "Информационные системы". 2010. №3.
3. Петров П.П. Основы объектно-ориентированного программирования на Java. – М.: Издательство "ДМК", 2015.
4. Сидоров С.С. Spring Boot: разработка корпоративных приложений. – СПб.: Питер, 2018.
5. Кузнецова О.Е. MySQL для профессионалов. – М.: ДМК Пресс, 2017.
6. Васильева Н.А. Веб-технологии: Thymeleaf и JSP в современных веб-приложениях // Журнал "Веб-разработка". 2019. №4.
7. Федоров А.А. Spring Security: основы безопасности в веб-приложениях // Информационные технологии и безопасность. 2018. №2.
8. Михайлов А.А. Модульная архитектура в программировании: принципы и практика // Журнал "Современное программирование". 2020. №5.
9. Новиков В.В. Алгоритмические методы в разработке информационных систем. – М.: Инфра-М, 2016.
10. Егоров Д.Д. Разработка веб-приложений с использованием Spring Framework // Вестник информационных технологий. 2017. №3.