

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математической теории упругости и биомеханики

Автоматизация работы HR-отдела службы доставки

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 442 группы

направления 09.03.03 - Прикладная информатика

механико-математического факультета

Радецкого Александра Романовича

Научный руководитель
доцент, к.ю.н.

Р.В. Амелин

Зав. кафедрой
зав. кафедрой, д.ф.-м.н., профессор

Л.Ю. Коссович

Саратов 2026

Введение. Современные службы доставки представляют собой динамично развивающийся сегмент рынка услуг, характеризующийся высокой интенсивностью операций, постоянным потоком заказов и значительной зависимостью от человеческих ресурсов. В таких условиях эффективность работы HR-отдела напрямую влияет на стабильность функционирования всей организации, поскольку именно он отвечает за подбор, адаптацию и распределение персонала.

Одной из ключевых особенностей данной сферы является высокая текучесть кадров, особенно среди курьеров и линейного персонала. Это приводит к необходимости постоянного найма сотрудников, оперативного рассмотрения заявок кандидатов и быстрого формирования рабочих смен. При этом HR-специалисты часто вынуждены работать в условиях ограниченного времени и большого объёма однотипных операций.

На практике в ряде организаций процессы управления персоналом остаются фрагментированными и недостаточно автоматизированными. Для хранения данных используются электронные таблицы, мессенджеры и отдельные сервисы, не связанные между собой. Это приводит к дублированию информации, повышенному риску ошибок, потере актуальности данных и снижению общей эффективности работы HR-отдела.

Современные HRM-системы частично решают данные проблемы, однако многие из них ориентированы на крупные предприятия и обладают избыточным функционалом, который не всегда востребован в условиях службы доставки. Кроме того, такие решения зачастую требуют значительных затрат на внедрение и обучение персонала, что делает их менее доступными для небольших и средних компаний.

В связи с этим возникает необходимость разработки специализированной информационной системы, ориентированной на автоматизацию ключевых HR-процессов именно в сфере доставки. Такая система должна обеспечивать удобное управление сотрудниками, сменами и кандидатами, а также предоставлять быстрый доступ к актуальной информации через мобильное приложение.

Дополнительным фактором актуальности является рост мобильных технологий и переход многих бизнес-процессов в формат мобильных приложе-

ний. Использование мобильного решения позволяет обеспечить доступ к системе в любое время и из любого места, что особенно важно для оперативного управления персоналом в условиях высокой динамики работы.

Целью бакалаврской работы является разработка системы автоматизации работы HR-отдела службы доставки с использованием мобильного приложения.

Для достижения поставленной цели в работе решаются следующие **задачи**:

- анализ предметной области и существующих решений;
- выявление основных проблем HR-процессов;
- формулирование требований к системе;
- проектирование структуры базы данных;
- разработка архитектуры системы;
- реализация мобильного приложения;
- тестирование разработанного решения.

Структура работы. Бакалаврская работа состоит из введения, трёх разделов, заключения и приложений. Список использованных источников включает 24 наименования.

Раздел 1. Анализ предметной области.

Раздел 2. Выбор технологий разработки и проектирование системы.

Раздел 3. Реализация и тестирование приложения.

В первом разделе рассмотрены особенности работы HR-отдела в службах доставки. Установлено, что данная сфера характеризуется высокой текучестью кадров, необходимостью массового найма сотрудников, а также постоянным управлением графиками работы.

Проведено исследование особенностей работы HR-специалистов, осуществляющих подбор персонала в сфере доставки и логистики.

Проанализированы основные проблемы существующих процессов. Выявлено использование разрозненных инструментов, отсутствие централизованного хранения данных и сложность управления сменами, что приводит к снижению эффективности работы. Сбор информации осуществлялся посредством анкетирования с использованием электронных форм и личного интервьюирования. В исследовании приняли участие 12 HR-специалистов, задей-

ствованных в подборе персонала для логистических и курьерских организаций. Профессиональный опыт участников исследования составлял от одного года до шести лет. В выборку вошли специалисты, осуществляющие подбор курьеров, операторов логистических центров и сотрудников клиентской поддержки

Во втором разделе на основе проведённого в первом разделе анализа сформулированы требования к разрабатываемой системе. Определены функциональные возможности, включая учет сотрудников, управление сменами, обработку заявок кандидатов, а также нефункциональные требования — удобство, мобильность и надёжность.

Описана архитектура разрабатываемой системы. Предложена клиент-серверная модель, включающая мобильное приложение и облачную базу данных для хранения и обработки данных.

Для клиентской части выбран язык Kotlin, который является официально рекомендованным языком для разработки приложений под операционную систему Android. Для серверной части выбран легковесный фреймворк Flask на языке Python, который позволяет быстро создать REST API и не требует сложной настройки. Облачным провайдером выбран Cloud.ru, дата-центры которого расположены на территории Российской Федерации, что соответствует требованиям Федерального закона № 152-ФЗ о локализации персональных данных, а также предоставляющий бесплатный тариф Free Tier с достаточными ресурсами для развёртывания.

Выполнено проектирование структуры **базы данных**, включающей следующие **сущности**:

- Таблица candidates хранит информацию о кандидатах.
- Таблица employees содержит данные о сотрудниках, принятых на работу.
- Таблица history ведёт хронологический журнал всех кадровых действий.

Определены **связи между сущностями**. При одобрении кандидата данные автоматически копируются из таблицы candidates в таблицу employees с добавлением текущей даты в поле hire_date, в таблицу history добавляется запись «НАНЯТ», после чего исходная запись удаляется из таблицы

candidates. При отказе кандидату в таблицу history добавляется запись «ОТКАЗАННО», а запись о кандидате удаляется. При увольнении сотрудника в таблицу history добавляется запись «УВОЛЕН», а запись о сотруднике удаляется из таблицы employees. При назначении собеседования в таблицу history добавляется запись с действием «СОБЕСЕДОВАНИЕ» и указанием выбранной даты.

ER-диаграмма базы данных представлена на рисунке 1.

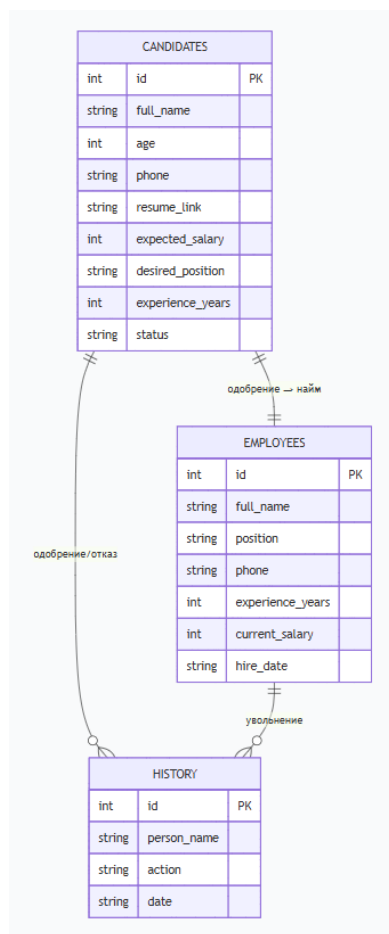


Рисунок 1 — ER-диаграмма базы данных

Выполнено проектирование **пользовательского интерфейса**, который состоит из трех основных экранов: «Кандидаты», «Сотрудники» и «История». Навигация между экранами осуществляется через нижнее таб-меню.

В третьем разделе была реализована серверная часть на языке программирования Python с использованием легковесного веб-фреймворка Flask. Серверное приложение представляет собой единый файл hr-server.py, содержащий весь необходимый код для обработки HTTP-запросов, взаимодействия

с базой данных PostgreSQL и ведения временного хранилища статусов собеседований.

Реализована **клиентская часть**. Приложение реализует три основных экрана, навигацию между ними, сетевое взаимодействие с сервером через HTTP-запросы и асинхронную обработку данных с использованием корутин.

Реализован **экран «Сотрудники»**. При его открытии загружается список сотрудников через `CloudDatabase.getEmployees()` и отображается в виде карточек. Каждая карточка сотрудника содержит фамилию и имя, должность, контактный телефон, опыт работы, текущую заработную плату и дату найма. В нижней части карточки расположена кнопка «Уволить» красного цвета, при нажатии на которую сотрудник удаляется из списка, а в историю добавляется запись «УВОЛЕН».

Экран «Сотрудники» представлен на рисунке 2.

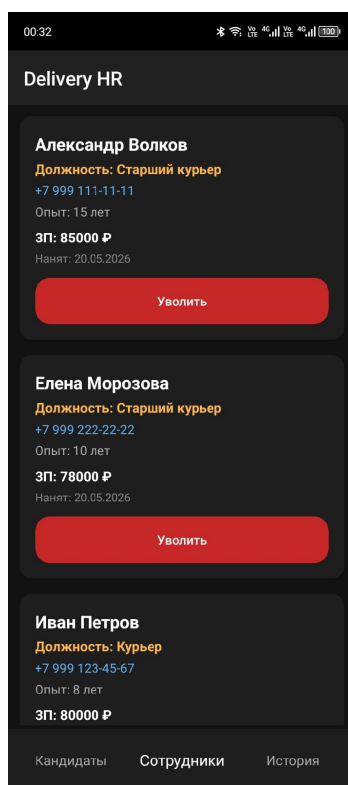


Рисунок 2 — Экран «Сотрудники»

Реализован **экран «Кандидаты»**. При открытии экрана вызывается метод `loadCandidates()`, который отправляет GET-запрос к серверу через `CloudDatabase.getCandidates()`, получает список кандидатов, отсортированный по опыту, возрасту и ожидаемой заработной плате. Для каждого кан-

дидата предусмотрены четыре кнопки. Кнопка «Резюме» открывает отдельный экран с подробной информацией о кандидате. Кнопка «Собеседование» открывает диалог выбора даты, после выбора которой в историю добавляется запись, а сама кнопка меняет вид, сигнализируя о том, что приглашение уже отправлено. При повторном нажатии на кнопку после назначения собеседования система выводит сообщение о том, что кандидат уже приглашён. Кнопка «Одобрить» переносит кандидата в список сотрудников с автоматической фиксацией даты найма и добавлением записи «НАНЯТ» в историю. Кнопка «Отказать» удаляет кандидата и добавляет запись «ОТКАЗАНО» в историю.

Экран «Кандидаты» представлен на рисунке 3.

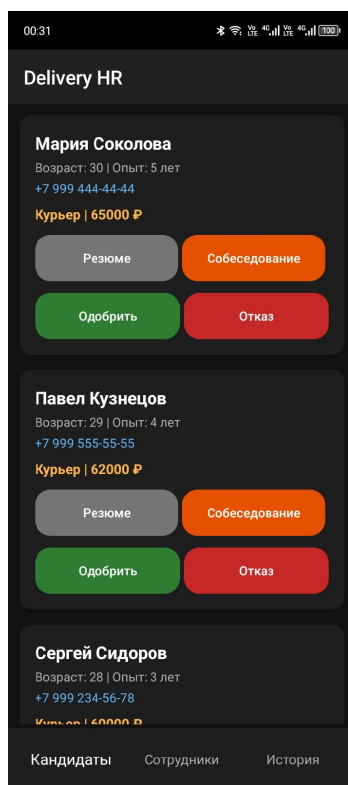


Рисунок 3 — Экран «Кандидаты»

Реализован экран «История». При открытии экрана загружается список записей через `CloudDatabase.getHistory()` и отображается в виде карточек. Каждая запись содержит имя человека, действие и дату с точностью до минуты. Для повышения наглядности записи имеют цветовую и символьную индикацию: записи о найме сопровождаются зелёной галочкой, записи

об увольнении и отказе — красным крестиком, а записи о назначении собеседования — значком часов.

Экран «История» представлен на рисунке 4.

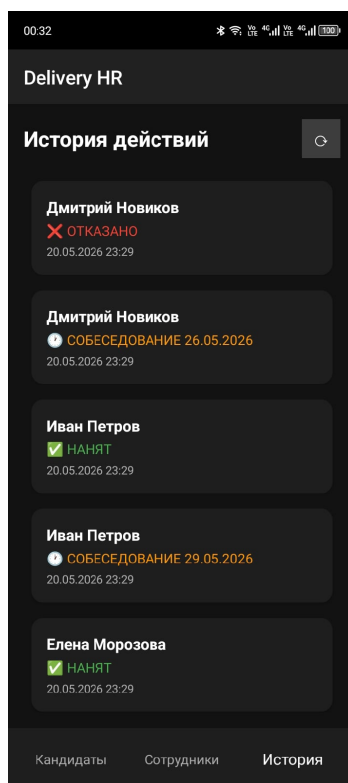


Рисунок 4 — Экран «История»

Реализован **экран просмотра резюме**. Данный экран открывается при нажатии на кнопку «Резюме» в карточке кандидата. В верхней части экрана отображается заголовок с именем кандидата, основная область представляет собой прокручиваемую текстовую зону, где содержится подробная информация о соискателе.

Экран просмотра резюме представлен на рисунке 5.

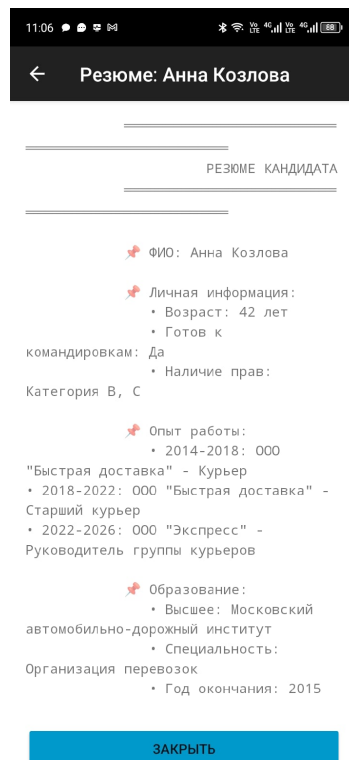


Рисунок 5 — Экран просмотра резюме

После реализации всех функциональных модулей мобильного приложения было проведено **функциональное тестирование** с целью проверки соответствия разработанной системы заявленным требованиям, выявления возможных ошибок и оценки удобства пользовательского взаимодействия.

Тестирование проводилось на физическом устройстве под управлением Android.

Тестирование модуля кандидатов. Загрузка списка кандидатов при открытии экрана выполняется корректно, данные отображаются в виде карточек с полной информацией. Ранжирование кандидатов соответствует заданному порядку: сначала кандидаты с наибольшим опытом работы, при равном опыте — старшие по возрасту, при равном опыте и возрасте — с наименьшей ожидаемой заработной платой. При нажатии на кнопку «Резюме» открывается экран с подробной информацией о кандидате. При нажатии на кнопку «Собеседование» открывается диалог выбора даты. После выбора даты появляется всплывающее сообщение с подтверждением, в историю добавляется запись, а кнопка меняет вид. При повторном нажатии на кнопку «Собеседование» выводится сообщение о том, что кандидат уже приглашен. При

нажатии на кнопку «Одобрить» кандидат исчезает из списка и появляется в списке сотрудников, в историю добавляется запись «НАНЯТ». При нажатии на кнопку «Отказать» кандидат исчезает из списка, в историю добавляется запись «ОТКАЗАНО».

Тестирование модуля сотрудников. Загрузка списка сотрудников при открытии экрана выполняется корректно. После подтверждения сотрудник исчезает из списка, в историю добавляется запись «УВОЛЕН».

Тестирование модуля истории. Загрузка списка записей при открытии экрана выполняется корректно. Записи отображаются в хронологическом порядке (сначала новые). Каждая запись имеет соответствующую иконку и цвет в зависимости от типа действия.

Тестирование обработки ошибок. При отсутствии интернет-соединения при попытке загрузить данные выводится сообщение об ошибке с указанием причины. При попытке выполнить операцию при отсутствии соединения также выводится соответствующее сообщение, данные не изменяются.

Результаты тестирования. Все описанные тестовые сценарии были выполнены успешно. Критических ошибок, приводящих к зависанию или вылету приложения, не выявлено. Система корректно обрабатывает штатные и пограничные ситуации, обеспечивая стабильную работу. Интерфейс приложения интуитивно понятен, все элементы управления доступны и удобны для сенсорного ввода.

Заключение. В результате выполнения выпускной квалификационной работы на тему «Автоматизация работы HR-отдела службы доставки» была достигнута поставленная цель — разработана система автоматизации ключевых HR-процессов, ориентированная на специфику служб доставки.

В первом разделе работы проведен анализ предметной области. Установлено, что службы доставки характеризуются высокой текучестью кадров, необходимостью массового оперативного найма, мобильностью персонала и большим объемом однотипных HR-операций. Выполнен обзор существующих решений для автоматизации HR-процессов. Показано, что каждое из этих решений имеет существенные недостатки для применения в службе доставки — избыточность функционала, высокая стоимость, отсутствие специализи-

рованного мобильного интерфейса для управления сменами и кандидатами, либо низкая надежность и отсутствие целостности данных. Проведенное анкетирование HR-специалистов подтвердило необходимость создания специализированного мобильного решения. Сформулированы функциональные и нефункциональные требования к разрабатываемой системе.

Во втором разделе выбрана клиент-серверная архитектура с централизованным хранением данных в облачной базе данных. Обоснован выбор технологического стека: для клиентской части — язык Kotlin с использованием Retrofit для сетевого взаимодействия, для серверной части — Python с фреймворком Flask. В качестве облачного провайдера выбрана российская платформа Cloud.ru. Спроектирована реляционная база данных PostgreSQL, состоящая из трех таблиц: candidates для хранения информации о кандидатах, employees для сотрудников и history для истории действий. Предусмотрено ранжирование кандидатов по трем критериям: убывание опыта работы, убывание возраста, возрастание ожидаемой заработной платы. Разработан пользовательский интерфейс с карточной структурой, нижней навигацией, цветовым кодированием кнопок и типов действий.

В третьем разделе представлена практическая реализация разработанной системы. Реализована серверная часть на Python с использованием Flask, развернутая на виртуальной машине в облаке Cloud.ru. Сервер предоставляет REST API для получения списка кандидатов, одобрения и отказа, назначения собеседований, управления сотрудниками, получения истории и сброса данных. Настроен systemd-сервис для автоматического запуска сервера при загрузке системы и его перезапуска в случае сбоя. Реализовано мобильное приложение на Kotlin, включающее три основных экрана. Экран кандидатов отображает список соискателей с возможностью просмотра резюме, назначения собеседования, одобрения или отказа. Экран сотрудников показывает список принятых работников с возможностью увольнения. Экран истории содержит хронологическую ленту всех кадровых действий с цветовой и символьной индикацией. Проведено функциональное тестирование, подтвердившее корректную работу всех модулей и соответствие системы заявленным требованиям.

Таким образом, все задачи, поставленные в работе, решены, цель достигнута. Разработанная система автоматизации HR-отдела службы доставки может быть использована в практической деятельности HR-специалистов для повышения эффективности управления персоналом. Приложение имеет интуитивно понятный интерфейс, адаптированный для мобильных устройств, обеспечивает централизованное хранение данных, целостность информации и удобное управление кандидатами и сотрудниками. В перспективе система может быть расширена за счет добавления аутентификации пользователей, модуля формирования отчетов, интеграции с внешними сервисами.