ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ПЕРМСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА Д.Н. ПРЯНИШНИКОВА»

Кафедра информационных  
систем и телекоммуникаций

Контрольная работа по дисциплине:  
«Архитектура информационных систем»  
Тема: «Информационная система агентства кадров»

Выполнил: студент группы ИСб-31  
Сединин А.П.  
Проверил: ст. преп. Кафедры ИСТ  
Бочкарёв А.М.

Пермь, 2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc124354330)

[**Теоретическая часть** 4](#_Toc124354331)

[Кадровое агентство 4](#_Toc124354332)

[Рабочий процесс кадрового агентства 5](#_Toc124354333)

[Основные объекты кадрового агентства 6](#_Toc124354334)

[Описание требований 6](#_Toc124354335)

[**Практическая часть** 9](#_Toc124354336)

[Архитектуры информационных систем 9](#_Toc124354337)

[Файл-серверная архитектура 9](#_Toc124354338)

[Двухуровневая архитектура клиент-сервер 10](#_Toc124354339)

[Архитектура клиент-сервер 2.5 слоя 11](#_Toc124354340)

[Трехуровневая клиент-серверная архитектура 12](#_Toc124354341)

[Выбор архитектуры информационной системы 13](#_Toc124354342)

[ИС кадровое агентство по классификациям ИС 14](#_Toc124354343)

[Программный продукт 15](#_Toc124354344)

[Клиент 15](#_Toc124354345)

[Сервер 18](#_Toc124354346)

[База данных 19](#_Toc124354347)

[ЗАКЛЮЧЕНИЕ 21](#_Toc124354348)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 22](#_Toc124354349)

# ВВЕДЕНИЕ

В 80-х и 90-х годах с бурным развитием инфокоммуникационных технологий многие аспекты жизни ушли в цифровую сферу: общение, телевидение, почта, заказ доставки еды на дом, онлайн запись на приём к врачу и так далее. Поиск работы не стал исключением и сейчас можно найти работу, не выходя из дома. Из-за этой стремительной динамики развития многие вещи и вовсе исчезают, и становятся невостребованными.

Несмотря на весь цифровой прогресс, многие элементы не всегда исчезают из истории, а лишь претерпевают незначительные изменения. Например, появился интернет-рекрутинг, но он всё равно не смог сразу убрать со своего пути такие структуры как кадровые агентства или биржи труда, так как не может обхватить всё вакантные места, которые создают работодатели. Работодателям иногда легче обратиться в кадровое агентство, чтобы найти сотрудника, чем размещать вакансию в интернете. Тем более специалист кадрового агентства – *рекрутер* может её более грамотно составить, используя готовый программный комплекс – информационную систему кадрового агентства. Да и, кроме того, кадровые агентства стараются более тщательно проанализировать местный рынок сотрудников и предпринимателей, чем интернет-рекрутинг, который этим не занимается, так как в нем просто размещаются вакансии, и специалист в добавок сам должен написать резюме, чтобы откликнуться на них.

Информационная система может из себя представлять не только какое-либо приложение, устанавливающееся на компьютере, но и может представлять из себя сайт, написанный на таких языках программирования как PHP или JavaScript в связке с HTML и CSS версткой.

# **Теоретическая часть**

## Кадровое агентство

Для начала стоит разобраться, что из себя представляет кадровое агентство, в чем суть существования данной системы, без понимания работы данной системы невозможно будет автоматизировать процессы внутри.

***Кадровое или рекрутинговое агентство*** – посредник на рынке труда, организация, оказывающая услуги работодателям по поиску и подбору персонала и/или оказывающая услуги соискателям по поиску работы и трудоустройству, а также другие сопутствующие услуги.

***Рекрутер*** – сотрудник кадрового агентства, работа, которого представляет собой создание вакансии по запросу *работодателя* для нахождения персонала, и создание резюме для *рабочего* для того, чтобы ему можно было найти с ним работу.

***Профайлинг* –** обычно под этим термином подразумевается чтение психофизических данных человека. В данном случае будет пониматься процесс создания резюме рабочего или создание вакансии для предпринимателя, благодаря которому он найдет себе сотрудника*.*

***Интервью*** *–* разговор рекрутера с работодателем или сотрудником с целью осуществить профайлинг.

***Трудовые данные*** – данные необходимые для создания вакансии или резюме. Например, такими данными являются трудовой стаж, образование, возраст, желаемые профессия и оклад сотрудника и так далее. Рекрутер получает трудовые данные во время проведения интервью.

***Конечным результатом***работы рекрутера является трудоустройство сотрудника и нахождение персонала работодателю.

***Резюме*** – документ, представляющий трудовые данные сотрудника работодателю.

***Вакансия*** – документ, представляющий описание какой профессии, стажа, образования, навыков и других качеств сотрудника, нужных работодателю.

## **Рабочий процесс кадрового агентства**

Сотрудник или работодатель хотят найти работу или персонал. Для осуществления своих потребностей они могут найти вакансию или резюме в интернете на каком-нибудь сайте, специализирующийся в сфере   
интернет-рекрутинга, а также они могут обратиться в агентство кадров. Для того чтобы воспользоваться услугой агентства кадров они также ищут сайт кадрового агентства в интернете, заходят на него и узнают адрес и контактные данные кадрового агентства, записываются на приём, либо пишут рекрутёрам на сайте, если есть такая возможность, либо пишут на почту, либо просто приходят. Рекрутер видит нового клиента и приглашает его на интервью, чтобы провести профайлинг, желательно, чтобы оно прошло очно, но можно и по телефону или видеосвязи. По приходу сотрудника или работодателя рекрутер проводит интервью, где опрашивает его с целью узнать все его трудовые данные. Во время интервью рекрутер выполняет профайлинг сотрудника или работодателя, в результате рекрутер создает резюме сотрудника или вакансию для нахождения сотрудника. После происходит связывание сотрудника и работодателя. Связывание выполняет также рекрутер. Далее связанные работодатель и сотрудник, могут уже наладить контакт – работодатель может прочитать резюме сотрудников, пришедшее ему на почту или на сайте, и по трудовым данным (в которых будут указаны и контакты сотрудника) позвонить любому из них, либо написать на почту, либо написать на сайте (если он может такое предоставить), и если сотрудник соответствует всем требованием, то назначить собеседование. После рекрутер должен удостовериться у работодателя, что рабочий прошел стажировку или испытательный срок и принят на работу.

Если рабочий не сумел устроиться, или работодатель не сумел подобрать персонал, то кадровое агентство продолжает оказывать свои услуги обеим сторонам, если у тех ещё есть желание искать работу или персонал в кадровом агентстве, которое им оказывало услуги.

Рекрутер также может предложить вакансию сотруднику из смежной области его образования, а работодателю сотрудника схожей сферы деятельности, которая ему требуется.

После окончания всех работ необходимо данные удалить из системы, данную работу делает рекрутер.

## **Основные объекты кадрового агентства**

Как видно основным рабочим звеном данной системы является *рекрутер*, именно он осуществляет оказание услуги, которое предоставляет кадровое агентство, именно исходя из его надобностей и будут сформированы требования к системе. Он является пользователем системы.

## **Описание требований**

1. Предварительные замечания к проекту
   1. Цели и рамки проекта

Цель проекта: создать информационную систему для автоматизации работы рекрутера.

* 1. Деловой контекст

Отсутствует.

* 1. Участники проекта

Отсутствуют.

* 1. Идеи в отношении решений

Отсутствуют.

1. Системные сервисы
   1. Рамки и системы

Сайт должен автоматизировать работу рекрутера, но не выполнять её за него. Иначе данная система уже не будет кадровым агентством, это уже будет являться интернет-рекрутингом, где рекрутеры вовсе не нужны.

* 1. Функциональные требования
* ИС должна содержать окно профайлинга для создания резюме сотрудника;
* ИС должна содержать окно профайлинга для создания вакансии подбора сотрудника для нахождения сотрудника работодателем.
  1. Требования к данным

Отсутствуют.

1. Системные ограничения
   1. Требования к интерфейсу

Должен быть интуитивно понятным и не иметь большое количество «красивостей», предоставлять минимально необходимый набор контролов(элементов, необходимых для выполнения функций системы – кнопки, кнопки выбора, кнопки исключающего выбора, выпадающие списки и так далее).

* 1. Требования к производительности;

Устройства-клиенты, реализующее систему, должны запускать браузер, минимальные требования для запуска, к примеру, Яндекс-браузера:

* ОС: Linux, Mac, Windows, BSD, Solaris, Android и другие системы, которые могут запускать этот браузер;
* Процессор: Intel Pentium 4 и выше;
* Оперативная память: минимум 512 МБ;
* Свободное место на диске: минимум 600 МБ.

Также имеется сервер Apache, работающий в виде демона, обрабатывающий http запросы клиентов.

* 1. Требования к безопасности;

У пользователей системы должны быть окна авторизации. Сайт не должен собирать персональные данные пользователей, кроме номера телефона и почты.

* 1. Эксплуатационные требования;

Отстутствуют.

* 1. Политические и юридические требования;

Отстутствуют.

* 1. Другие ограничения;

Отстутствуют.

1. Проектные вопросы;
   1. Открытые вопросы;

Отсутствуют.

* 1. Предварительный план график;

Отсутствуют.

* 1. Предварительный бюджет;

Отсутствуют.

# **Практическая часть**

## **Архитектуры информационных систем**

### **Файл-серверная архитектура**

В архитектуре «файл-сервер», схема которой представлена на   
(рисунок 1), средства организации и управления БД (в том числе и СУБД) целиком располагаются на машине клиента, а БД, представляющая собой обычно набор специализированных структурированных файлов, – на машине-сервере. В этом случае серверная часть компонента представлена даже не средствами СУБД, а сетевыми составляющими операционной системы, обеспечивающими удалённый разделяемый доступ к файлам. Таким образом, «файл-сервер» представляет вырожденный случай клиент-серверной архитектуры.

Рисунок 1 – Архитектура файл-сервер

Взаимодействие между клиентом и сервером происходит на уровне команд ввода-вывода файловой системы, которая возвращает запись или блок данных. Запрос к базе, сформулированный на языке манипулирования данными, преобразуется самой СУБД в последовательность команд ввода-вывода, которые обрабатываются операционной системой машины-сервера.

Достоинства:

* возможность обслуживания запросов нескольких клиентов, но не параллельно;
* Легкость развёртывания и администрирования;
* Низкая стоимость разработки;
* Высокая скорость разработки.

Недостатки:

* Низкая производительность из-за высокой загрузки сети и машин-клиентов, так как обмен идёт на уровне единиц информации файловой системы – физических записей, блоков или даже файлов, из которых на машине клиента будут выбраны и представлены необходимые для приложения элементы данных.   
  Не исключены случаи, когда клиенту необходима вся база, передача которой может задействовать всю пропускную способность сети;
* низкий уровень защиты данных, так как доступ к файлам БД управляется общими средствами ОС сервера;
* низкий уровень управления целостностью и непротиворечивостью информации, так как бизнес-правила функциональной обработки, сосредоточенные на клиентской части, могут быть противоречивыми и несихронизированными;
* Низкие возможности расширения;
* При изменении клиентской программы, придется её менять на всех клиентских станциях.

### **Двухуровневая архитектура клиент-сервер**

Если в файл-серверной архитектуре серверу, где хранится база, внедрить СУБД, которая до этого находилась на каждой клиентской станции, и решить проблему целостности, использовав уже вместо однопользовательского СУБД многопользовательское СУБД, то в результате получится   
клиент-серверная архитектура (рисунок 2). В результате клиентам остаётся только передать запрос серверу, который обработает СУБД, изменяющая базу.

Рисунок 2 – Архитектура клиент-сервер

Достоинства:

* возможность распределить функции вычислительной системы между несколькими независимыми компьютерами;
* все данные хранятся на защищенном сервере;
* поддержка многопользовательской работы гарантия целостности данных.

Недостатки:

* неработоспособность сервера может сделать неработоспособной всю вычислительную сеть;
* сложное администрирование;
* высокая стоимость оборудования;
* бизнес-логика приложений осталась в клиентском ПО;
* При изменении клиентской программы, придется её менять на всех клиентских станциях.

**Архитектура клиент-сервер 2.5 слоя**

В отличие от 2-х слойной архитектуры 2.5-слойная архитектура обычно не требует наличия высокоскоростных каналов связи между клиентской и серверной частями системы, так как по сети передаются уже готовые результаты вычислений - почти все вычисления производятся на серверной стороне. Существенно улучшается также и защита информации - пользователям даются права на доступ к функциям системы, а не на доступ к ее данным и т.д. Однако вместе с преимуществами унитарного подхода архитектура 2.5 перенимает и все его недостатки, как-то: ограниченную масштабируемость, зависимость от программной платформы, ограниченное использование сетевых вычислительных ресурсов. Кроме того программы для серверной части системы пишутся на встроенных в СУБД языках описания хранимых процедур, предназначенных для валидации данных и построения несложных отчетов, а вовсе не для написания ИС масштаба предприятия. Все это снижает быстродействие системы, повышает трудоемкость создания с модификации ИС и самым негативным образом сказывается на стоимости аппаратных средств, необходимых для ее функционирования.

**Трехуровневая клиент-серверная архитектура**

Компоненты трехзвенной архитектуры (рисунок 3), с точки зрения программного обеспечения реализуют определенные сервера БД, web-сервера и браузеры. Место любого из этих компонентов может занять программное обеспечение любого производителя. Сервер БД может быть представлен MySQL-сервером и другими, сервер приложений технологиями: ADO.NET, ASP.NET и web-сервером IIS, Apache, Nginx и другие, а роль клиента выполняет любой web-браузер.

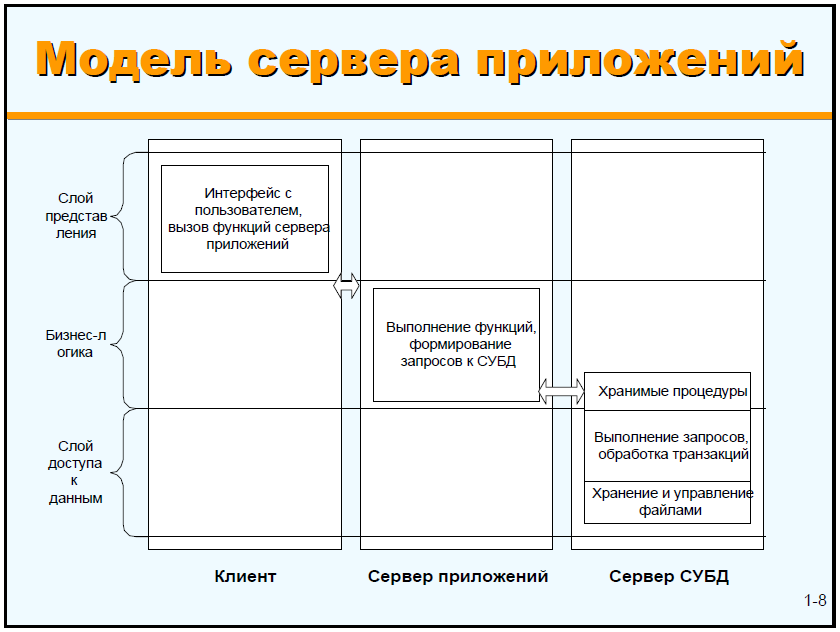


Рисунок 3 – Трехуровневая архитектура клиент-сервер

Достоинства:

* Тонкий клиент;
* Между клиентской программой и сервером приложения передается лишь минимально необходимый поток данных - аргументы вызываемых функций и возвращаемые от них значения. Это теоретический предел эффективности;
* Использования линий связи, даже работа с ANSI-терминалами (не говоря уже об использование протокола http) требует большей нагрузки на сеть;
* Сервер приложения ИС может быть запущен в одном или нескольких экземплярах на одном или нескольких компьютерах, что позволяет использовать вычислительные мощности организации столь эффективно и безопасно как этого пожелает администратор ИС;
* Дешевый трафик между сервером приложений и СУБД. Трафик между сервером приложений и СУБД может быть большим, однако это всегда трафик локальной сети, а их пропускная способность достаточно велика и дешева. В крайнем случае, всегда можно запустить СП и СУБД на одной машине, что автоматически сведет сетевой трафик к нулю;
* Снижение нагрузки на сервер данных по сравнению с 2.5-слойной схемой, а значит и повышение скорости работы системы в целом;
* Дешевле наращивать функциональность и обновлять ПО.

Минусы:

* Выше расходы на администрирование и обслуживание серверной части.

Следующие архитектуры разбирать не стоит, так как выбор пал на трехуровневую клиент-серверную архитектуру. Следующие уже будут иметь излишества для данной системы.

**Выбор архитектуры информационной системы**

В данном случае, информационной системе «кадровое агентство» подходит *трехуровневая архитектура клиент-сервер*. Для ИС кадрового агентства необходимы:

* Возможность высокого масштабирования. Количество пользователей кадрового агентства может быть как большим, так и маленьким. Много кому нужна работа и много кому нужен сотрудник прямо в данный момент времени, из-за этого количество данных, с которыми работают рекрутеры, постоянно меняется, поэтому необходимо чтобы на систему никаким образом не отразилось регистрация или удаление пользователя, как это, к примеру, происходит в файл-серверной;
* Так как ИС постоянно должна добавлять и удалять данные, авторизировать пользователей, выполнять определенные операции, алгоритм которых включает в себя обращения к базе данных, например, получение данных таблиц, значения которых удовлетворяют определенным условиям, то должна быть такая часть, которая будет ответственна за обработку данных запросов. Должен быть сервер приложений, который выполнит часть логики сам, а логику, по изменении данных в базе направит на выполнение СУБД. В данном случае сервером приложений является – web-сервер Apache, принимающий и отдающий http - запросы и интерпретатор языка PHP, выполняющий скрипты клиентских процедур, пришедших из запроса, интегрированный в него в качестве мода;
* Данная система должна иметь клиент, который выполняет минимум логики. При использовании системы с устройства, пользователь только вызывает нужный функционал, который отправляет инструкции на вызов функций и их аргументы, а сама логика обработки выполняется на сервере приложений. При изменении клиентской части не нужно будет менять её на всех устройствах пользователей, изменения применяются динамически, к примеру, при обновлении страницы в браузере. Должен быть тонкий клиент;
* Возможность в качестве клиента использовать браузер, а в качестве сервера - веб-сервер, что опять же свойственно трехуровневой клиент-сервер архитектуре, кроме этого, пользователю не нужно качать что-то лишнее.

**ИС кадровое агентство по классификациям ИС**

* По типу данных: *Документальная*. Так как она работает с информацией (резюме, вакансии), смысл которой может извлечь только человек и работу произвести с ней может тоже только он;
* По типу автоматизации: *Автоматизированная*,для того чтобы система пришла в действие, с ней должен поработать рекрутер;
* По сфере применения: *Интегрированная*. По этой классификации, относится к этому типу, так как предназначена для автоматизации работы рекрутера;
* По характеру обработки: *Информационно-поисковая система*. Относится к этому типу, так как просто выдаёт информацию рекрутеру, которую нужно заполнить, изменить и связать с другой информацией, сложных алгоритмов над информацией не производиться;
* По уровню управления: *Функциональная*. Относится к функциональным, потому что создана для выполнения функции узконаправленного специалиста.

**Программный продукт**

Для разработки информационной системы использовался язык программирования PHP и операционная система Linux ALT Workstation 10.1.

### **Клиент**

Вся система из себя представляет множество процедурных, клиентских окон, с определенным функционалом.

Чтобы войти в рекрутерский аккаунт нужно воспользоваться окном входа (рисунок 4).

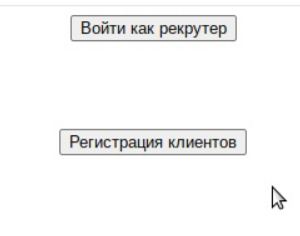


Рисунок 4 – Общее окна входа пользователей системы

Чтобы войти как рекрутер, нужно кликнуть на соответствующею кнопку. После выйдет окно авторизации рекрутера (рисунок 5).

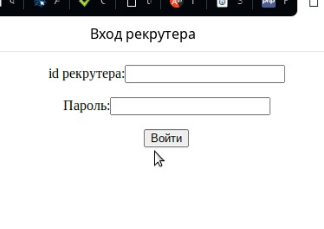


Рисунок 5 – Окно входа рекрутера

После успешной аутентификации появится пять кнопок, перенаправляющие на соответствующие окна профайлинга клиентов кадрового агентства (рисунок 6).

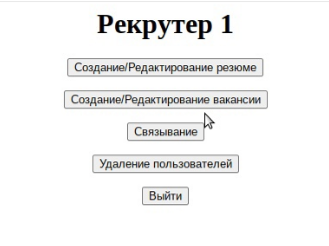


Рисунок 6 – Страница выбора окон

При нажатии на кнопку с редактированием вакансий, выйдет окно выбора вакансий (рисунок 7).

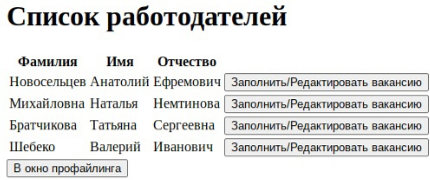


Рисунок 7 – Окно выбора редактируемой вакансии

Выбираем вакансию для заполнения и происходит перемещение на страницу заполнения вакансии (рисунок 8). После заполнения формируется файл вакансии.

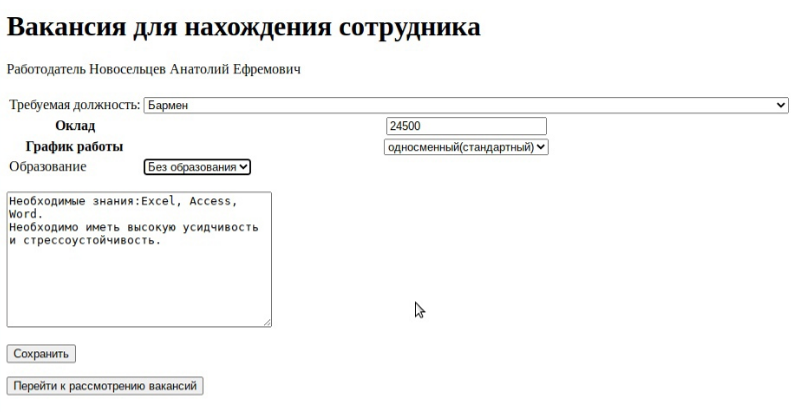


Рисунок 8 - Страница «Заполнение/Создание вакансий»

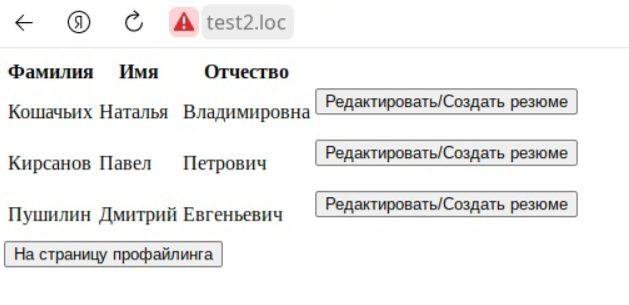


Рисунок 9 - Окно выбора редактируемого резюме

При выборе того, кому нужно создать резюме, происходит переход на страницу с созданием резюме (рисунок 10). После заполнения создается файл с резюме, который рекрутер может отформатировать нужным ему образом.

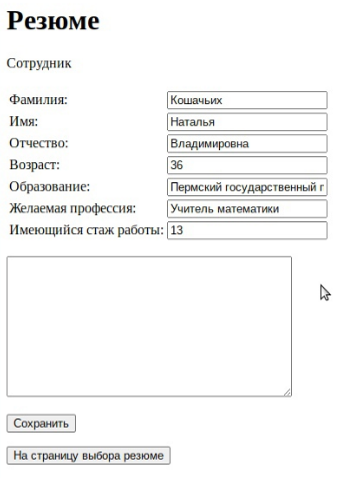


Рисунок 10 – Окно редактируемого резюме

При нажатии на кнопку связывания сотрудников и работодателей выходит список работодателей, относительно которых происходит связывание (рисунок 11).

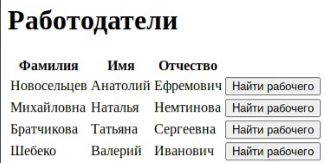


Рисунок 11 – Окно выбора вакансии для связывания

Рекрутор сам должен выбрать из списка внизу кого можно направить на данную специальность (рисунок 12). В данном окне формируется файл-связки, который рекрутер отправляет работодателю.

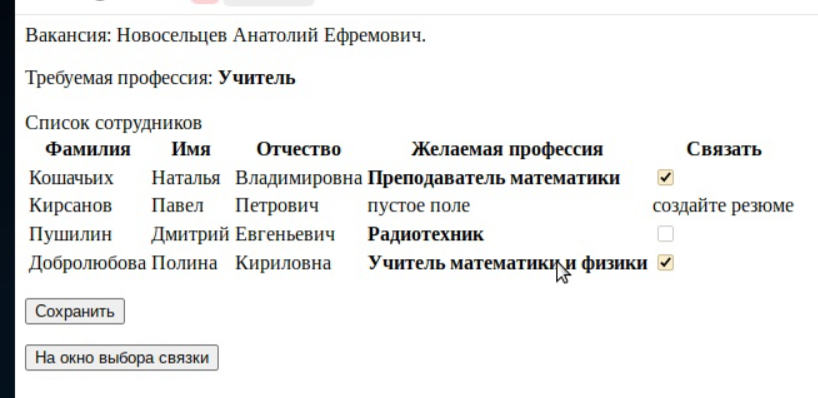


Рисунок 12 - Окно связки

### **Сервер**

Сервер приложений из себя представляет LAMP – набор программ – Linux, Apache, MySql/MariaDB, PHP, работающий в виде демона.   
Apache – WWW – сервер, обрабатывающий http запросы, является основным компонентом сервера, предварительно в нем настроены папка, в которой размещается весь сайт целиком, которую уже может видеть браузер, подключены библиотеки для связки с экземпляром MariaDB в php8.ini, в качестве мода в Apache интегрирован интерпретатор PHP 8.1. Все команды и инструкции для сборки сервера, указаны в источнике [2].

### База данных

База данных состоит из 7 таблиц (рисунок 13). Определения, которых представлены в таблицах 1-7.

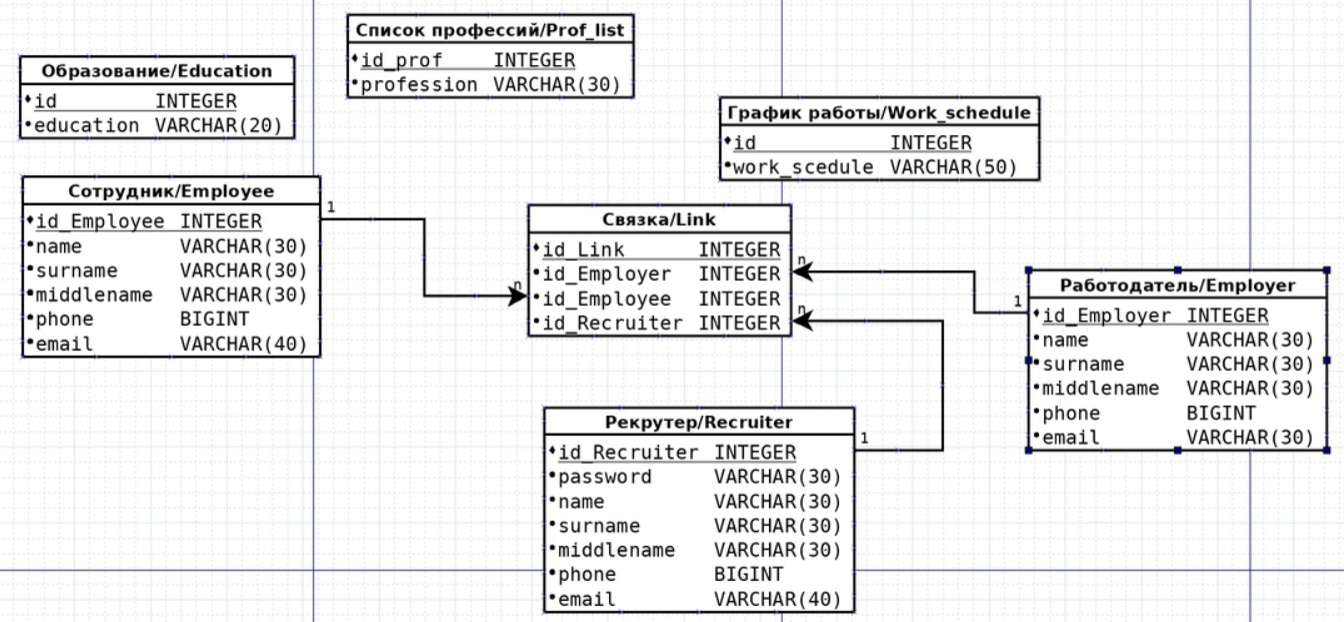


Рисунок 13 – Физическая модель базы данных кадрового агентства

Таблица 1 – Описание таблицы Education

|  |  |
| --- | --- |
| Имя атрибута | Тип данных |
| id | INTEGER |
| education | VARCHAR(20) |

Таблица 2 – Описание таблицы Recruiter

|  |  |
| --- | --- |
| Имя атрибута | Тип данных |
| Id\_Recruiter | INTEGER |
| password | VARCHAR(30) |
| name | VARCHAR(30) |
| surname | VARCHAR(30) |
| middlename | VARCHAR(30) |
| phone | BIGINT |
| email | VARCHAR(40) |

Таблица 3 – Описание таблицы Link

|  |  |
| --- | --- |
| Имя атрибута | Тип данных |
| Id\_link | INTEGER |
| Id\_Employer | INTEGER |
| Id\_Employee | INTEGER |
| Id\_Recruiter | INTEGER |

Таблица 4 – Описание таблицы Employee

|  |  |
| --- | --- |
| Имя атрибута | Тип данных |
| Id\_ Employee | INTEGER |
| name | VARCHAR(30) |
| surname | VARCHAR(30) |
| middlename | VARCHAR(30) |
| phone | BIGINT |
| email | VARCHAR(40) |

Таблица 5 – Описание таблицы Employer

|  |  |
| --- | --- |
| Имя атрибута | Тип данных |
| Id\_ Employer | INTEGER |
| name | VARCHAR(30) |
| surname | VARCHAR(30) |
| middlename | VARCHAR(30) |
| phone | BIGINT |
| email | VARCHAR(40) |

Таблица 6 – Описание таблицы Prof\_list

|  |  |
| --- | --- |
| Имя атрибута | Тип данных |
| id | INTEGER |
| education | VARCHAR(20) |

Таблица 7 – Описание таблицы Work\_schedule

|  |  |
| --- | --- |
| Имя атрибута | Тип данных |
| id | INTEGER |
| education | VARCHAR(20) |

# ЗАКЛЮЧЕНИЕ

Несмотря на весь прогресс в информационной сфере, где все большее количество процессов не требуют вмешательства человека, всё равно остаются некоторые вещи, которые, как кажется, должны уйти в историю, так как они уже изжили себя.

В случае с агентством кадров так и произошло, пока они еще живы, и им требуется автоматизация процессов, но интернет-рекрутинг рано или поздно возьмет своё, после останутся лишь рекрутеры, которые уже не будут представлять из себя отдельную и самостоятельную структуру, а будут частью одного большого какого-либо предприятия, для которого и будут подбирать персонал, что и сейчас уже давно не редкость, и чтобы выполнить свою работу, они будут пользоваться интернет-рекрутингом или тем, во что он эволюционирует.

В результате контрольной работы была создана ИС для кадрового агентства, автоматизирующая работу рекрутера.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Руководство по PHP 8// https://metanit.com: сайт. – URL: https://metanit.com/php/tutorial/ (дата обращения: 25.12.2022).
2. ApacheMySQLPHP // <https://www.altlinux.org>: сайт. – URL: <https://www.altlinux.org/ApacheMySQLPHP> (дата обращения: 02.11.2022).
3. Архитектура информационных систем // <https://portal.psaa.ru>: сайт. – URL: <https://portal.psaa.ru/get/@system/coursefiles/1670560088869/AIS.pptx> (дата обращения 25.12.2022)
4. Типы архитектур информационных систем // <https://portal.psaa.ru>: сайт. – URL:<https://portal.psaa.ru>/get/@system/coursefiles/1670560090090/Арх\_ИС.ppt