# ВВЕДЕНИЕ

Еще несколько десятков лет назад вычислительная техника занимала целые комнаты, на данный момент технология изготовления их модернизировалась и в целом в корне изменила темп и образ жизни общества. Процесс компьютеризации необратим, а развитие информационных систем невозможно остановить ввиду их теоретической и практической значимости во всех отраслях народного хозяйства.

Согласно обзору инвестиционного портала, инвестиции в сферу IT-технологий являются на данный момент наиболее выгодными. И в ближайшем будущем ситуация вероятней всего не изменится, а IT направления будут наиболее популярными, ведь каждый современный человек имеет косвенное или прямое отношение к вычислительным устройствам и сети Интернет. В связи с этим другие сферы жизнедеятельности людей будут вынуждены подстраиваться к новшествам, принимая при этом возможно иные формы, чем сейчас.

На современном этапе развития интернет-технологий все наибольшую популярность приобретает электронный бизнес. Современные интернет-технологии, которые реализуются в области экономики, предоставляют организациям большие возможности в организации, ведении и поддержании электронной торгово-посреднической деятельности. Средой для реализации данной коммерции выступает сеть Internet, в которой и реализуются все проекты.

Большинство людей полагают, что электронная коммерция – это покупка товаров в режиме онлайн. Обычно это называется электронной коммерцией – традиционная розничная торговля "магазинного" типа. К числу современных и крупнейших представителей такого рода электронной коммерции в сети Internet относятся сайты:

* Amazon (<http://amazon.com>)
* Ebay (<http://ebay.com>)
* Aliexpress.com (<http://aliexpress.com>)

Интернет стал доступнее, быстрее и дешевле. Сейчас уже никого не удивишь выделенной линией или безлимитным тарифом. В таких условиях удобнее открыть страницу сайта, чем покупать газету. Однако статистика показывает, что максимум активности пользователей приходится на рабочее время – многие используют доступ в Интернет на работе. Действительно, работу секретаря или бухгалтера сейчас невозможно представить без компьютера, а компьютер – без доступа в сеть.

Исходя из этого объемы продаж через Internet уже приблизились к традиционным розничным продажам. Такие сайты уже сейчас занимаю важную область рынка и, несомненно, будут продолжать расти.

Целью данного проекта является разработка интернет приложения ведения электронной коммерции.

Задачи проекта:

* реализовать возможность оформления и добавление товара\услуг в корзину;
* реализовать возможность регистрации и авторизации пользователя с разделением прав доступа;
* реализовать возможность отображения новостей;
* реализовать возможность отображение самых популярных заказываемых товаров или услуг.

# СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

# Анализ предметной области

Использование всемирной паутины для ведения электронной коммерции является сегодня наиболее актуальным и востребованным способом. Связано это с удобством совершения данных мероприятий. Для достижения цели, гражданам нужно лишь иметь персональный компьютер или смартфон с браузером и выходом в интернет. Просматривать имеющиеся сведения вполне реально в любое время дня и ночи. При этом, совершенно не важно конечное местоположение пользователя.

Тема дипломного проекта «Интернет приложение ведения электронной коммерции».

Основной целью данного проекта является разработать приложение облегчающее покупку любых товаров и услуг. Просмотр популярных товаров или услуг которые автоматически формируются исходя из заказов пользователей.

Дипломный проект будет представлять собой веб-приложение, предназначенное для просмотра товаров или услуг, добавления в корзину, просмотра новостей компании, оформления заказов. Со стороны администратора будет предусмотрена возможность манипулирования данными в базе данных с помощью MYSQL Workbench.

Аналогами данного проекта являются следующие ресурсы:

* Deal.by – орговый центр в интернете. Потребительские, промышленные и оптовые товары. Все для вашего бизнеса, быта и отдыха!
* Belassist.by – организация приема платежей по банковским картам Белкарт, Visa, MasterCard, Visa Electron, Maestro, American Express на сайтах интернет-магазинов.

Обозначим основные части дипломного проекта.

Первоначально необходимо разработать базу данных. После сайт, для чего нужно выполнить следующие этапы создания веб-сайта:

1. Определение целей веб-сайта и его позиционирование.

На этом этапе необходимо определить, для чего нужен сайт, т.е. какие задачи он должен решать: предоставить общее представление о товаре и услуге, или многосторонне осветить какую-либо сторону человеческой деятельности, увеличить продажи по традиционным каналам или организовать веб-торговлю, провести рекламную или маркетинговую кампанию.

Цели веб-сайта, в большинстве случаев, должны ставиться заказчиком, а затем, вместе с исполнителем они уточняются и корректируются.

Это один из самых важных этапов не только создания веб-сайта как такового, но важнейший этап интернет-маркетинга.

После определения целей сайта надо совершенно четко и как можно подробнее представить и описать целевую аудиторию сайта, т.к. это влияет на то, в каком виде будет представлена информация.

Определение целевой аудитории веб-сайта – это не менее важный, чем определение целей веб-сайта, этап. Не всегда целевая аудитория компании из офлайн будет прямо проецироваться на он-лайн. Зная целевую аудиторию и аудиторию части Интернета можно сделать предположения о том, кто будет являться основными посетителями веб-сайта.

Определение и как можно более подробное описание целевой аудитории сайта дает возможность разработать правильный дизайн для проекта, а также выбрать правильное направление для написания текстов.

На заключительной стадии этого этапа примерно определяют, по каким поисковым запросам сайт должен появляться в результатах поискового запроса и посмотреть интернет-проекты конкурентов будущего веб-проекта.

Дополнительную информацию о положении вещей в Интернете на вашем рынке даст обзор сайтов конкуретнов. Необходимо иметь в виду, что совсем необязательно конкуренты из реальной жизни будут являться таковыми в Интернете. Нужно просмотреть некоторое количество сайтов, которые расположены в ТОП 10 ведущих поисковых машин, обратите внимание на графическое решение, на тексты, на поисковые запросы. Это даст достаточно пищи для размышления, каким должен быть сайт вашей компании.

1. Создание Технического Задания (далее ТЗ) на разработку веб-сайта.

В ТЗ необходимо как можно более подробно описать:

* цели создания сайта и его целевую аудиторию;
* структуру веб-сайта и количество страниц в каждом разделе;
* работу динамических модулей;
* пожелания по дизайну (цвета, использоание фирменного стиля, соотношение графика/текст и т.д.;
* используемые технологии (Java 8, Spring Boot, HTML5, Spring JPA, Bootstrap, Thymeleaf, hibernate);
* порядок предоставления, обработки или создания графической и текстовой информации;
* технические требования к сайту.

ТЗ является основным документом, на основе которого осуществляются все последующие этапы разработки веб-сайта.

1. Создание дизайн-макета веб-сайта.

На этом этапе дизайнер в специальной графической программе создает дизайн страниц будущего веб-сайта с прорисовкой всех графических (баннеров, кнопок, фотографий) и текстовых элементов. Дизайнер создает дизайн веб-страниц с учетом пожеланий заказчика и задания, прописанного в ТЗ.

Современные мониторы имеют различные разрешения. В разговоре о веб-сайтах нас особенно интересует ширина. Размер монитора по ширине в пикселях может быть 800, 1024, 1280, 1600 и даже больше. Более того, пользователь может просматривать сайт в полускрученном окне или на смартфоне.

Фиксированный дизайн сайта предполагает одну единую ширину сайта для всех разрешений экрана и для всех ширин окон, т.е. если ширина сайта 1000 пикселейй, то на экране 800 пикселей будет появляться горизонтальная прокрутка, а на экране 1280 пикселей – пустые поля слева и справа (или только с одной стороны)

Такое построение сайта позволяет четко управлять композицией сайта и однозначно знать, где будет каждый элемент навигации на каждом экране. Этот тип сайта позволяет использовать более насыщенные графические элементы.

Резиновый дизайн, в отличие от фиксированного «подстраивается» под ширину экрана.

Это приводит к тому, что сайт без горизонтальной прокрутки увидит большинство пользователей, но на разных экранах композиция сайта будет непредсказуемой.

Данный сайт адаптивный, поэтому будет удобен на всех видах устройств.

1. Верстка сайта.

После того, как был утвержден дизайн – за работу принимается верстальщик – это тот человек, который переводит дизайн-макет на язык, понятный компьютеру с использованием языка HTML5 и фреймворка Bootstrap.

1. Программирование сайта.

Очень часто этап программирования и верстки объединяют в один. На мелких и средних проектах оба действия в состоянии выполнить один человек. На крупных проектах в силу специфики работ эти этапы разделяют.

На этапе программирования (как правило, с использовавнием HTML, Java, Thymeleas, Spring framework и баз данных) происходит создание всех страниц сайта, определяется порядок работы меню, расставляются гиперссылки, создается динамика на сайте, программируются такие составляющие, как гостевая книга, форум, новостная лента и проч.

Если сайт должен иметь администраторский интерфейс, то он создается именно на этапе программирования.

Очень важно на этапе программирования определить, на какой системе администрирования (Content Management Site) будет работать ваш сайт. На сегодня на рынке существует великое множество систем администрирования.

1. Наполнение сайта информацией.

На этом этапе информация размещается на сайте, т.е. путем перевода в специальный формат текст и графика располагаются на сайте на определенных страницах, и эта информация становится доступной для просмотра.

1. Тестирование сайта.

Этот этап можно осуществить как до, так и после размещения сайта по его «родному» адресу. На этом этапе выявляются все ошибки и недочеты в программировании и написании текстов.

После разработки сайта его нужно разместить в сети интернет.

Для этого он должен иметь домен и хостинг.

Доменное имя или домен – символическое имя. под которым сайт будет известен в сети. Получить его можно либо, купив у регистратора, бесплатно, но в виде поддомена от одного из бесплатных хостингов.

Хостинг – услуга размещения сайта на серверах, имеющих выход в Интернет В настоящее время компаний, предоставляющих эту услугу – огромное количество. Остается только выбрать подходящую. К критериям выбора относятся:

1. Цена есть как платные, так и бесплатные хостинги.
2. Объем предоставляемого графика и дискового пространства.

Поддержка тех или иных средств (Java, Tomcat, MySQL, возможность развертывания CMS. наличие дополнительных скриптов и т.д.).

Для данного проекта был выбран хостинг ActiveCloud был основан в 2003 году. Уже 14-ый год компания предлагает все современные облачные решения и хостинг, осуществляя поддержку более 45000 [клиентов](http://www.active.by/ru-by/company/clients/) в Беларуси и 6 странах СНГ. ActiveCloud входит в группу [Softline](http://softlinegroup.com/) – ведущую международную компанию в сфере лицензирования программного обеспечения, оказывающую полный спектр IT-услуг.

Далее необходимо разместить все файлы сайта на сервере у хостера. Как правило, предоставляется несколько вариантов для этого, а именно:

1. Обычно это загрузка содержимого по НТТР-лрогоколу, с использованием панели управления хостера или пo FTP с помощью FTP- клиента.
2. Второй вариант намного быстрее и проще. Одним из лучших FTP- клиентов является встроенный в Total Commander и бесплатный FileZilla.

Как только устанавливается связь с сервером FTP (хостер обязательно сообщит адрес, логин и пароль входа), пространство сервера станет доступным в виде логического устройства (как и обычные диски компьютера) на одной из двух панелей. После чего необходимо запустить процесс копирования и ждать его окончания. По окончании процесса копирования сайт будет опубликован в Интернете.

Для привлечения клиента на свой сайт существует большое количество способов.

Поисковые системы – по статистике большой популярностью пользуются те сайты, которые находятся в верхних строках поисковой выдачи. На первые 3-5 сайтов приходится 80% посетителей – именно поэтому все сайтовладельцы так жаждут быть в поисковых системах как можно выше.

Основные методы для привлечения посетителей:

1. Контекстная реклама – один из самых эффективных платных методов привлечения целевой аудитории – контекстная реклама. Данный вид рекламы работает через крупные поисковые системы (Яндекс, Google). Ее суть – появление рекламного объявления только тогда, когда пользователь вводит строго определенные поисковые запросы. Определенность в ключевом запросе даст гарантию показов только целевой аудитории.
2. Социальные сети в современном мире – это место, где общаются миллионы. Поскольку они собраны компактно, то велика вероятность быстро выйти на целевую аудиторию. Создание и превлечение в него людей со сходными интересами, поддерживайте жизнь. Ссылка на сайт может стоять в статусе, а можно просто давать анонсы на новый материал сайта.
3. Реклама в социальных сетях. В соцсетях есть возможность подачи рекламы. По эффективности ничто не сравнится с контекстной рекламой, но в соцсетях вы можете выбрать ту группу пользователей, которым будет показываться ваше объявление.
4. Бесплатные доски объявлений. В интернете немало бесплатных досок объявлений, но реально ценных не так уж много, а потому не стоит тратить время на доски с маленькой посещаемостью. Лучше потратить время на качественное объявление, которое обращает на себя внимание, и разместить его в трех-четырех местах, но серьезных, с вероятностью высокой конверсии.

# Структура программы

Бизнес-процессы программного продукта можно сформировать и описать BPWin-диаграммой (см. рисунок 1.1, 1.2).

IDEF0 – нотация графического моделирования, используемая для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции. Нотация IDEF0 является одной из самых популярных нотаций моделирования бизнес-процессов. К ее особенностям можно отнести:

* использование контекстной диаграммы;
* поддержка декомпозиции;
* доминирование;
* выделение 4 типов стрелок.

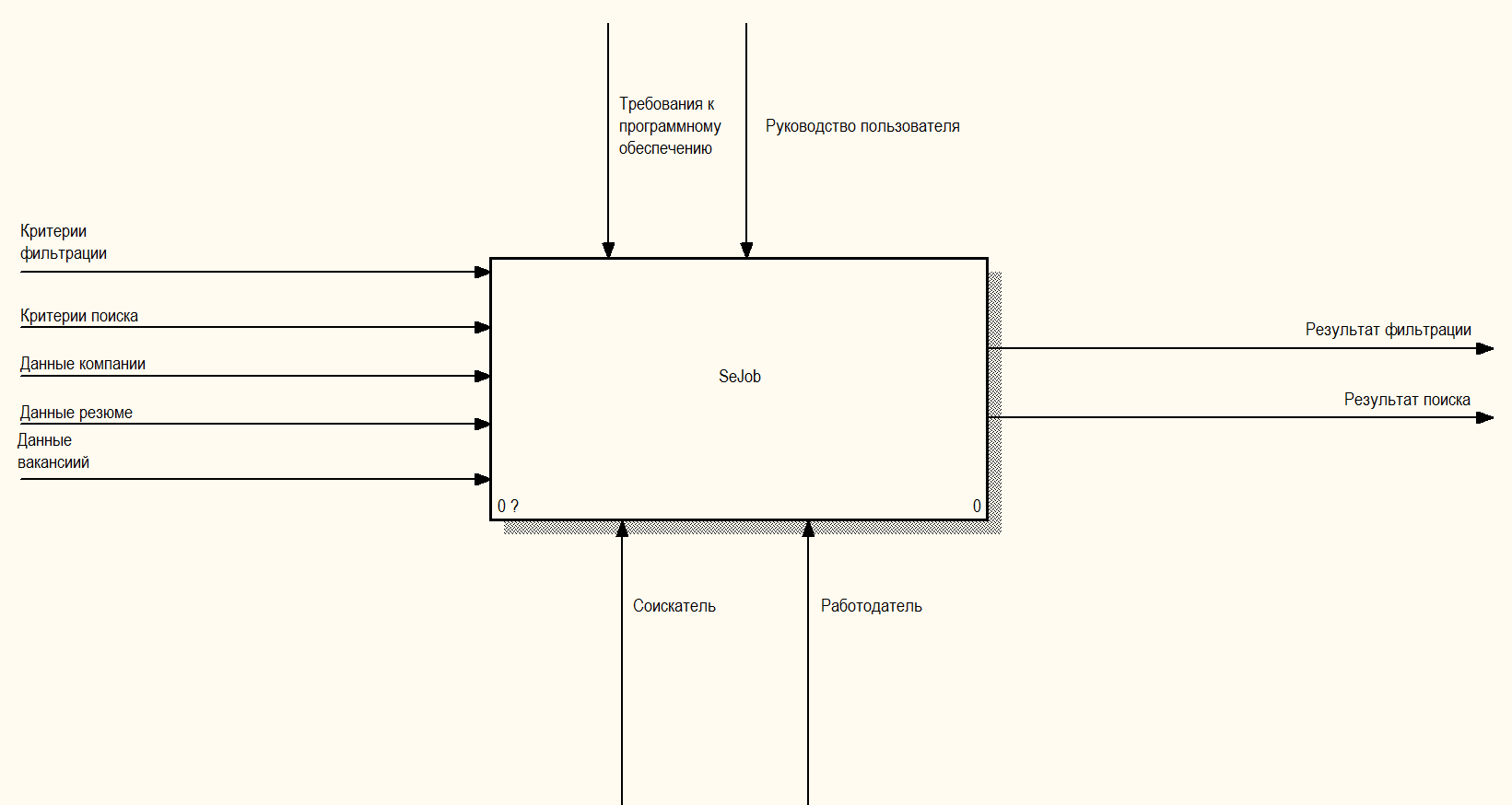


Рисунок 1.1 – Главный процесс веб-приложения «SeJob»

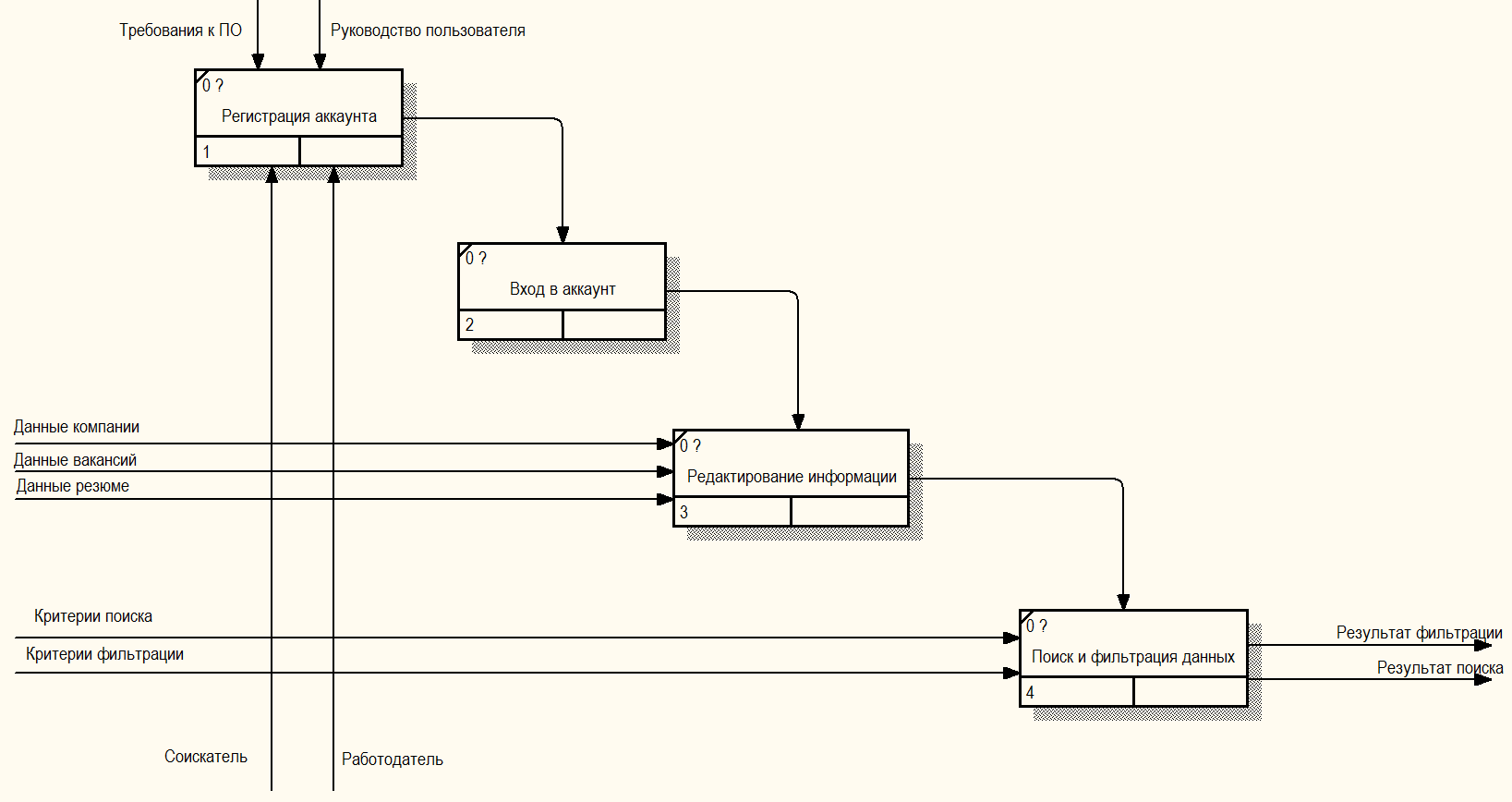


Рисунок 1.2 – Декомпозиция главного процесса веб-приложения «SeJob»

# Выбор и обоснование среды разработки

Основной средой разработкой является Brackets – бесплатный редактор с открытым кодом. Brackets ориентирован на работу с HTML, CSS и JavaScript. Эти же технологии лежат в основе самого редактора, что обеспечивает его кроссплатформенность т.е. совместимость с операционными системами Mac, Windows и Linux. Редактор удобен в первую очередь функцией Live Preview, которая позволяет при изменении код редактора – в окне браузера автоматически отображаются изменения. Так же имеется подсветка синтаксиса, подсказки при редактировании CSS, JS и HTML-файлов и огромное количество скачиваемых плагинов, позволяющих сделать редактор еще удобнее.

Для разработки базы данных использовалась приложение MySQL WorkBench.

MySQL Workbench – приложение для визуального проектирования баз данных, в составе которого инструменты проектирования, моделирования, создание и эксплуатация БД.

MySQL Workbench – преемник систем DBDesigner 4 из FabForce и предлагается в двух вариантах:

Community Edition – распространяется под свободной лицензией GNU GPL Standard Edition – доступна по ежегодной оплачиваемой подписке. Эта версия включает в себя дополнительные функции, которые повышают производительность разработчиков и администраторов БД.

Возможности программы MySQL Workbench:

* Позволяет наглядно представить модель базы данных в графическом виде.
* Наглядный, функциональный механизм установки связей между таблицами, в том числе «многие ко многим» с созданием таблицы связей.
* Reverse Engineering – восстановление структуры таблиц из уже существующей на сервере БД (связи восстанавливаются в InnoDB, при использовании MyISAM – связи необходимо устанавливать вручную).
* Удобный редактор SQL запросов, позволяющий сразу же отправлять их серверу и получать ответ в виде таблицы.
* Возможность редактирования данных в таблице в визуальном режиме.

Для создания сайта на локальном компьютере используется программный комплекс Open Server.

Open Server – это портативная серверная платформа и программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий.

Программный комплекс имеет богатый набор серверного программного обеспечения, удобный, многофункциональный продуманный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов. Платформа широко используется с целью разработки, отладки и тестирования веб-проектов, а так же для предоставления веб-сервисов в локальных сетях.

Состав программного комплекса, нужны для данного дипломного проекта:

* OSPanel 5.2.6;
* Apache 2.2.31 / 2.4.25;
* MySQL 5.1.73 / 5.5.53 / 5.6.34 / 5.7.16;
* PHP 5.6.29 (Xdebug 2.5.0, Memcache 3.0.8, Mongo 1.6.14, MongoDB 1.2.2, Redis 2.2.7, Imagick 3.2.0);
* PHP 7.0.14  (Xdebug 2.5.0, PDFlib 9.1.0, MongoDB 1.2.7, Redis 3.1.0);
* PHPMyAdmin 4.6.5.2.

Apache – это HTTP сервер, обладающий высокой надежностью и гибкостью, под HTTP сервером следует понимать программное обеспечение для обработки HTTP запросов. Основная работа Apache это обработка и ответ на HTTP запросы и генерирование динамическое содержание страниц. Гибкость достигается путем использования файла .htaccess, благодаря которому можно переопределять глобальные настройки сервера Apache. Вебмастерами этот файл используется для настройки редиректов, обработки ошибок, безопасности, доступа, кодировки и т.д.  Apache имеет механизм для определения виртуальных хостов, благодаря чему на одном IP адресе может находится неограниченное количество сайтов, именно таким образом работает виртуальный хостинг. Также существует большое количество модулей, позволяющих ему работать с большинством популярных языков программирования и расширять функциональность. На сегодняшний день Apache является одним из самых популярных ПО для веб-сервера. Его используют  хостинг-провайдеры по всему миру.

MySQL – это одна из самых популярных и самых распространенных СУБД (система управления базами данных) в интернете. Она не предназначена для работы с большими объемами информации, но ее применение идеально для интернет сайтов, как небольших, так и достаточно крупных. MySQL отличатся хорошей скоростью работы, надежностью, гибкостью. Работа с ней, как правило, не вызывает больших трудностей. Поддержка сервера MySQL автоматически включается в поставку PHP.

Немаловажным фактором является ее бесплатность. MySQL распространяется на условиях общей лицензии GNU (GPL, GNU Public License).

PHP – это широко используемый язык сценариев общего назначения с открытым исходным кодом. PHP это язык программирования, специально разработанный для написания web-приложений (сценариев), исполняющихся на Web-сервере.

# ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

# Входная информация

Входной информацией является, информация которую пользователь вносит в базу данных, заполняя необходимые поля выбранной таблицы, а также вводя информацию в базу данных с помощью запросов SQL.

Для разрабатываемого программного продукта входной будет служить следующая информация:

* Резюме.
* Информация о компании.
* Информация о вакансиях компании.
* Информация, передаваемая при нажатии пользователем на кнопку «В избранное».
* Критерии фильтрации и поиска.

# Организация данных

Для хранения данных часто используются таблицы реляционных баз данных. Обычно данные могут быть представлены как абстрактные пары «ключ-значение». Данные могут быть организованы в различные виды структур данных, таких как массивы, связанные списки или объекты. Структуры данных могут хранить данные различных типов, включая числа, строки и другие структуры данных. Ввод и вывод данных в компьютеры производится через периферийные устройства.

Модель данных (datamodel) отображает состав и связи таблиц БД, использующихся в программном проекте. Модель данных для данного приложения смотреть на рисунке 2.1.

Модель можно условно разделить на 3 блока.

* Вакансии (CompanyVacancy).
* Компании (Company).
* Резюме (Summary).

Все 3 блока дочерние к блоку «Accounts» (данные о пользователе).

«Summary» и «Company» связаны с блоком «Accounts» один к одному.

Все дочерние блоки связанные с блокам «CompanyVacancy», «Company» и «Summary» подразумеваются как нефиксированная информация, то есть тех же телефонов в резюме может быть указано и 2, и 3, и 5. Для этого такие данные были выведены в отдельные таблицы.

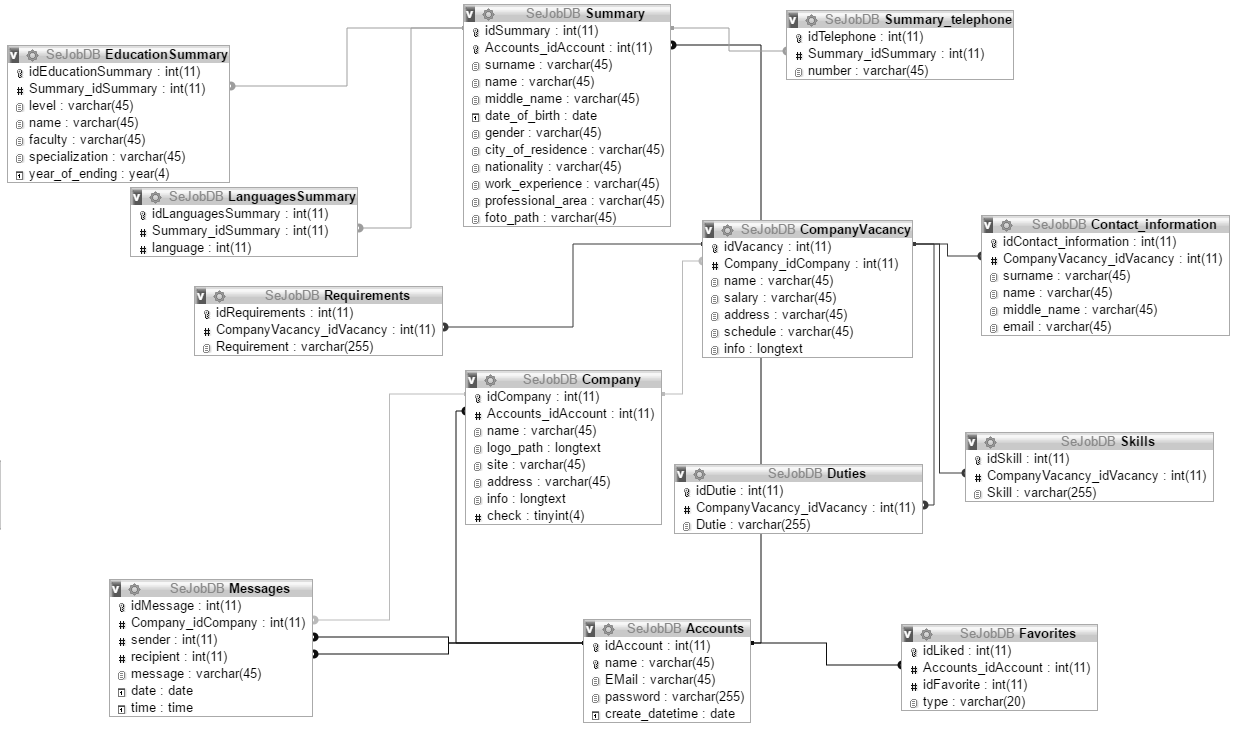


Рисунок 2.1 – Модель данных веб-приложения «SeJob»

Данные организованы при помощи следующих таблиц:

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица 2.1 – Структура таблицы «Accounts» | | | |
| Поле | Тип | Null | По умолчанию |
| idAccount | int(11) | Нет |  |
| Name | varchar(45) | Нет |  |
| EMail | varchar(45) | Нет |  |
| Password | varchar(255) | Нет |  |
| create\_datetime | date | Да | NULL |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| В данной таблице хранится информация об аккаунтах пользователей.  Таблица 2.2 – Структура таблицы «emailchangepasid»   |  |  |  |  | | --- | --- | --- | --- | | Поле | Тип | Null | По умолчанию | | Id | varchar(128) | Нет |  | | Email | text | Нет |  |   В данной таблице хранятся запросы на изменение пароля по почте. | | | |
| Таблица 2.3 – Структура таблицы «Company» | | | |
| Поле | Тип | Null | По умолчанию |
| idCompany | int(11) | Нет |  |
| Accounts\_idAccount | int(11) | Нет |  |
| Name | varchar(45) | Да | NULL |
| logo\_path | varchar(45) | Да | NULL |
| Site | varchar(45) | Да | NULL |
| City | varchar(45) | Да | NULL |
| Info | longtext | Да | NULL |
| Check | tinyint(4) | Да | NULL |
| В данной таблице хранится информация об компании пользователя.  Таблица 2.4 – Структура таблицы «CompanyVacancy» | | | |
| Поле | Тип | Null | По умолчанию |
| idVacancy | int(11) | Нет |  |
| Company\_idCompany | int(11) | Нет |  |
| Name | varchar(45) | Да | NULL |
| Salary | varchar(45) | Да | NULL |
| City | varchar(45) | Да | NULL |
| Address | varchar(45) | Да | NULL |
| experience\_required | varchar(45) | Да | NULL |
| type\_of\_employment | varchar(45) | Да | NULL |
| date\_of\_creation | date | Да | NULL |
| В данной таблице хранится информация об вакансиях компании пользователя.  Таблица 2.5 – Структура таблицы «Contact\_information» | | | |
| Поле | Тип | Null | По умолчанию |
| idContact\_information | int(11) | Нет |  |
| CompanyVacancy\_idVacancy | int(11) | Нет |  |
| Surname | varchar(45) | Да | NULL |
| Name | varchar(45) | Да | NULL |
| Email | varchar(45) | Да | NULL |
| Telephone | varchar(45) | Да | NULL |

В данной таблице хранится контактная информация вакансии пользователя.

Таблица 2.6 – Структура таблицы «Duties»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Null | По умолчанию |
| Dutie | varchar(64) | Нет |  |
| CompanyVacancy\_idVacancy | int(11) | Нет |  |

|  |  |  |  |
| --- | --- | --- | --- |
| В данной таблице хранится обязанности вакансий.  Таблица 2.7 – Структура таблицы «EducationSummary» | | | |
| Поле | Тип | Null | По умолчанию |
| idEducationSummary | int(11) | Нет |  |
| Summary\_idSummary | int(11) | Нет |  |
| Level | varchar(45) | Да | NULL |
| Name | varchar(45) | Да | NULL |
| Faculty | varchar(45) | Да | NULL |
| Specialization | varchar(45) | Да | NULL |
| year\_of\_ending | year(4) | Да | NULL |
| В данной таблице хранится информация об образовании соискателя.  Таблица 2.8 – Структура таблицы «LanguagesSummary» | | | |
| Поле | Тип | Null | По умолчанию |
| idLanguagesSummary | int(11) | Нет |  |
| Summary\_idSummary | int(11) | Нет |  |
| Language | int(11) | Да | NULL |

В данной таблице хранятся языки которыми владеет соискатель.

Таблица 2.9 – Структура таблицы «Favotites»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Null | По умолчанию |
| idLiked | int(11) | Нет |  |
| Accounts\_idAccount | int(11) | Нет |  |
| idCompany | varchar(45) | Да | NULL |
| В данной таблице хранятся избранное пользователей.  Таблица 2.10 – Структура таблицы «Messages» | | | |
| Поле | Тип | Null | По умолчанию |
| idMessage | int(11) | Нет |  |
| Company\_idCompany | int(11) | Да | NULL |
| Sender | int(11) | Нет |  |
| Recipient | int(11) | Нет |  |
| Message | varchar(45) | Да | NULL |
| Date | date | Да | NULL |
| Time | time | Да | NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| В данной таблице хранятся сообщения пользователей.  Таблица 2.11 – Структура таблицы «Requirements» | | | |
| Поле | Тип | Null | По умолчанию |
| idRequirements | varchar(64) | Нет |  |
| CompanyVacancy\_idVacancy | int(11) | Нет |  |
| В данной таблице хранится требования вакансий.  Таблица 2.12 – Структура таблицы «Skills» | | |  |
| Поле | Тип | Null | По умолчанию |
| Skill | varchar(64) | Нет |  |
| CompanyVacancy\_idVacancy | int(11) | Нет |  |
| В данной таблице хранится требуемые навыки вакансий.  Таблица 2.13 – Структура таблицы «Summary» | | | |
| Поле | Тип | Null | По умолчанию |
| idSummary | int(11) | Нет |  |
| Accounts\_idAccount | int(11) | Нет |  |
| Surname | varchar(45) | Да | NULL |
| Name | varchar(45) | Да | NULL |
| date\_of\_birth | date | Да | NULL |
| Gender | varchar(45) | Да | NULL |
| city\_of\_residence | varchar(45) | Да | NULL |
| Nationality | varchar(45) | Да | NULL |
| work\_experience | varchar(45) | Да | NULL |
| foto\_path | varchar(45) | Да | NULL |
| В данной таблице хранится резюме соискателей.  Таблица 2.14 – Структура таблицы «Summary\_telephone» | | | |
| Поле | Тип | Null | По умолчанию |
| idSummary\_telephone | int(11) | Нет |  |
| Summary\_idSummary | int(11) | Нет |  |
| В данной таблице хранятся номера телефонов резюме. | | | |

# Выходная информация

Выходная информация – результат выполнения запросов, фильтрации данных, вывод необходимой информации в отчет, печать информации.

Информация, получаемая в результате выполнения функций информационной системы и выдаваемая на объект ее деятельности, пользователю или в другие системы.

Выходной информацией для данного приложения является:

* Списки вакансий, резюме и компаний.
* Резюме определенного пользователя.
* Кампания и ее вакансии пользователя.
* Избранное пользователя.
* Местоположения на карте.
* Результаты фильтрации и поиска.

# Возможности системы

Возможности системы можно представить в виде диаграммы деятельности (см. прил. A).

Диаграмма деятельности – это диаграмма, на которой показано расположение некоторой деятельности на её составные части. Под деятельностью понимается спецификация исполняемого поведения в виде последовательного и параллельно выполнения подчинённых элементов, вложенных видов деятельности и отдельных действий, связанных между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграмма показывает последовательность действий, необходимых для ее достижения.  Действия (action) это элементарные шаги, которые не предполагают дальнейшую декомпозицию.

Деятельность может содержать входящие и/или исходящие дуги деятельности, показывающие потоки управления и потоки данных. Если поток соединяет две деятельности, он является потоком управления. Если поток заканчивается объектом, он является потоком данных.

Деятельность выполняется, только тогда, когда готовы все его «входы», после выполнения, деятельность передает управление и(или) данные на свои «выходы».

В данной диаграмме показано, как пользователь использует веб-приложение. Первоначально проходит процесс регистрации, где пользователь заполняет поля. При вводе логина или почты, уже существующей в базе, предусмотрен вывод сообщения об ошибке.

Далее представлено, как пользователь заполняет данные резюме, компании и вакансии. Затем используется функция фильтрации и поиска, есть возможность добавления в избранное.

# Описание взаимодействия объектов системы

Взаимодействие объектов в системе можно представить диаграммой последовательностей (см. прил. Б).

Главный акцент в диаграмме последовательностей – порядок и динамика поведения, то есть, как и в каком порядке происходят события. Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники с названиями объектов), вертикальные «линии жизни», отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение им определенной функции (прямоугольники на пунктирной «линии жизни»), и стрелки, показывающие обмен сигналами или сообщениями между объектами.

В данной диаграмме показано, как пользователь взаимодействует с веб-приложением, а оно в свою очередь с базой данных. Показано что пользователь должен зарегистрироваться, после войти, а дальше заполнить данные о компании, резюме и вакансии. Далее он вводит критерии поиска и фильтрации и выбирает избранное.

# 

# Варианты использования системы

Взаимодействие между актерами и вариантами использования представлено диаграммой вариантов использования (см. прил. В).

Диаграмма вариантов использования – это исходное концептуальное представление или концептуальная модель системы в процессе ее проектирования и разработки. Создание диаграммы вариантов использования имеет следующие цели:

Определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы.

Сформулировать общие требования к функциональному поведению проектируемой системы.

Разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей.

Подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Назначение данной диаграммы состоит в следующем: проектируемая программная система представляется в форме так называемых вариантов использования, с которыми взаимодействуют внешние сущности или актеры. При этом актером или действующим лицом называется любой объект, субъект или система, взаимодействующая с моделируемой бизнес системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая служит источником воздействия на моделируемую систему так, как определит разработчик. Вариант использования служит для описания сервисов, которые система предоставляет актеру.

В данной диаграмме показано, что работодатель может создать компанию и вакансии, а соискатель только резюме. Варианты использования фильтрации, поиска может использовать любой. Также показано что работодателю и соискателю дана возможность редактирования своих данных (удаление, изменение, добавление).

# РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

# Разработка класса User

Класс User описывает сущность пользователя. Основной задачей данного класса является вход и регистрация на сайте. Данный функционал реализован следующим кодом:

class User

{

public function SignIn($login, $password)

{

try {

require ("/db\_conn.php");

$sql = "SELECT `password`, `name` FROM `Accounts` WHERE `name` LIKE '".$login."'";

foreach ($dbh->query($sql) as $result);

if ($result["name"] == $login)

{

if (password\_verify($password, $result["password"])==true) {

$\_SESSION['SLOGIN']=$login;

return "Success";

}

else {

return "Bad password";

}

}

else

{

return "Bad login";

}

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function Register($login, $password, $email)

{

if ($email==NULL)

{

return "Ошибка";

}

if ($password==NULL)

{

return "Ошибка";

}

if ($email==NULL)

{

return "Ошибка";

}

try {

require ("/db\_conn.php");

$sql = "SELECT \* FROM `Accounts` WHERE `name` LIKE '".$login."'";

foreach ($dbh->query($sql) as $result);

if ($result[0]!=NULL)

{

return "Bad login";

}

$sql = "SELECT \* FROM `Accounts` WHERE `EMail` LIKE '".$email."'";

foreach ($dbh->query($sql) as $result);

if ($result[0]!=NULL)

{

return "Bad email";

}

$hashpas = password\_hash($password, PASSWORD\_DEFAULT);

$sql = "INSERT INTO `Accounts` (`name`, `email`, `password`) values('".$login."','".$email."','".$hashpas."')";

$dbh->query($sql);

return "Success";

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function MessageSend($login, $message, $recipient)

{

try {

require ("/db\_conn.php");

$time = date("H:i:s");

$today = date("Y-m-d");

$sql = "INSERT INTO `Messages` (sender, recipient, message, date, time) values('".$login."','".$recipient."','".$message."','".$today."','".$time."')";

$dbh->query($sql);

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function СalculateAge($birthday) {

$birthday\_timestamp = strtotime($birthday);

$age = date('Y') – date('Y', $birthday\_timestamp);

if (date('md', $birthday\_timestamp) > date('md')) {

$age--;

}

return $age;

}

}

# Разработка класса Content

Основной задачей класса Content является вывод, добавление, изменение информации. Его код представлен ниже:

class Content

{

public function ImageAdd($uploaddir)

{

$uploadfile = $uploaddir . basename($\_FILES['img']['name']);

// return $uploadfile;

if (move\_uploaded\_file($\_FILES['img']['tmp\_name'], $uploadfile)) {

return $uploadfile;

} else

{

return "Возможная атака с помощью файловой загрузки!";

}

}

public function IUD($sql){ // Insert Update Delete

require("/db\_conn.php");

try {

$dbh->query($sql);

return $sql;

} catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function ViewAll($sql) {

require ("/db\_conn.php");

try {

$sth = $dbh->prepare($sql);

$sth->execute();

$allarr=$sth->fetchAll();

return $allarr;

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function CheckVal($login, $colname, $tablename){

require("/db\_conn.php");

try {

$id = "(SELECT `idAccount` FROM `Accounts` WHERE `name` LIKE '".$login."')";

$sql = "SELECT `".$colname."` FROM `".$tablename."` WHERE `".$colname."` LIKE ".$id."";

foreach ($dbh->query($sql) as $row);

if ($row[0] != NULL) {

return "true";

} else {

return "false";

}

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function CheckFavorite($idf,$login,$type)

{

require("/db\_conn.php");

try {

$id = "(SELECT `idAccount` FROM `Accounts` WHERE `name` LIKE '".$login."')";

$sql = "SELECT \* FROM `Favorites` WHERE `Accounts\_idAccount` LIKE ".$id." AND `idFavorite` LIKE '".$idf."' AND `type` LIKE '".$type."'";

foreach ($dbh->query($sql) as $row);

if ($row[0] == NULL) {

return "true";

} else {

return "false";

}

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function filter($array,$tablename)

{

$str = "SELECT \* FROM `".$tablename."` WHERE";

foreach($array as $row)

{

if ($row['value']!=NULL)

{

$str = $str."`".$row['name']."` LIKE '".$row['value']."' AND";

}

}

strstr($str, ' AND', true)

# ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

# Типы ошибок

Тестирование программных продуктов – процесс исследования, испытания программного обеспечения, имеющий две цели:

* продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
* выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

Существующие на сегодня методы тестирования программного обеспечения не позволяют однозначно и полностью выявить все дефекты и установить корректность функционирования анализируемой программы, поэтому все существующие методы тестирования действуют в рамках формального процесса проверки исследуемого или разрабатываемого продукта.

Такой процесс формальной проверки, или верификации, может доказать, что дефекты отсутствуют с точки зрения используемого метода. То есть нет никакой возможности точно установить или гарантировать отсутствие дефектов в программном продукте с учётом человеческого фактора, присутствующего на всех этапах жизненного цикла ПО.

Качество программного обеспечения можно определить, как совокупную характеристику исследуемого ПО с учётом следующих составляющих:

* надёжность;
* сопровождаемость;
* практичность;
* эффективность;
* мобильность;
* функциональность.

Так же существуют различные уровни тестирования:

* Модульное тестирование – тестируется минимально возможный для тестирования компонент, например, отдельный класс или функция. Часто модульное тестирование осуществляется разработчиками программного обеспечения.
* Интеграционное тестирование – тестируются интерфейсы между компонентами, подсистемами или системами. При наличии резерва времени на данной стадии тестирование ведётся итерационно, с постепенным подключением последующих подсистем.
* Системное тестирование – тестируется интегрированная система на её соответствие требованиям.

Существует множество подходов к решению задачи тестирования и верификации ПО, но эффективное тестирование сложных программных продуктов – это процесс в высшей степени творческий, не сводящийся к следованию строгим и чётким процедурам или созданию таковых.

# Функциональное тестирование

Функциональное тестирование – это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает ПО, какие задачи оно решает.

Во-первых, тестирование дает уверенность в том, что в ходе разработки были реализованы все функциональные требования заказчика к продукту.

Во-вторых, тестирование обеспечивает надежность и безопасность продукта на предмет постороннего вмешательства или неправомерного доступа к конфиденциальной информации. Данные задачи стоят перед таким тестированием безопасности.

Функциональное тестирование является одним из ключевых видов тестирования, задача которого – установить соответствие разработанного программного обеспечения (ПО) исходным функциональным требованиям заказчика. То есть проведение функционального тестирования позволяет проверить способность информационной системы в определенных условиях решать задачи, нужные пользователям.

Его ключевые преимущества:

* ПО полностью имитирует фактическое использование системы.
* Позволяет своевременно выявить системные ошибки ПО и, тем самым, избежать множества проблем при работе с ним в дальнейшем.
* Экономия за счет исправления ошибок на более раннем этапе жизненного цикла ПО.

Функциональные требования включают в себя:

* Функциональная пригодность (англ. suitability).
* Точность (англ. accuracy).
* Способность к взаимодействию (англ. interoperability).
* Соответствие стандартам и правилам (англ. compliance).
* Защищённость (англ. security).

При проведении тестирования были испытаны все функции, которые были задуманы при разработке программного продукта.

Вот некоторые из них:

* добавление в избранное;
* добавление в базу данных;
* удаления из базы данных;
* изменение в базе данных;
* вывод из базы данных;
* отправка сообщений;
* вход в аккаунт;
* регистрация.

# Полное тестирование

Тестирование программы включает в себя проверку её работы в различной среде и разных условиях с целью выявления ошибок и проверки корректности её работы.

Тест – это последовательность входных данных и соответствующих им набор эталонных результатов, которые должны быть получены при верном функционировании программы.

Так как разрабатываемый программный продукт главным образом предназначен для автоматизиции создания вакансии, а так же резюме, и компаний. Исходя из этого, особое внимание уделялось функциям создания, их редактирования и удаления. Также особое внимание было уделено интерфейсу программного продукта. За время пользования программой у пользователя не должны уставать глаза, а так же он должен быть интуитивно понятным. Полное тестирование можно проанализировать в тест кейсе (см. таблицу 4.1).

Таблица 4.1 – Тест кейсы

|  |  |  |
| --- | --- | --- |
| Действие | Ожидаемый результат | Результат теста |
| Адаптация страницы под различное разрешение | При работе с различными устройствами, веб-приложения должно изменять свои размеры | Тестирование прошло успешно. Веб-приложение изменяет размеры в соответствии с разрешением |
| Тестирование формы авторизации пользователя | После заполнения формы и нажатие кнопки, должна быть создана сессия, и изменена шапка | Тестирование прошло успешно. Сессия создана, шапка изменена |
| Тестирование формы регистрации пользователей | После заполнения формы с определёнными условиями и отправки данных, должна произойти регистрация пользователя | Тестирование прошло успешно. Данные добавлены в базу |
| Тестирование ввода данных резюме | При вводе данных пользователя данные должны добавиться в базу данных | Тестирование прошло успешно. Данные добавлены в базу |
| Тестирование добавления в избранное | После нажатия на кнопку избранного, данные должны добавиться в базу данных | Тестирование прошло успешно. Данные добавлены в базу |

Продолжение таблицы 4.1

|  |  |  |
| --- | --- | --- |
| Тестирование изменения динамических данных в окне резюме | После изменения в полях, данные должны обновиться | Тестирование выявило ошибку. Данные были добавлены, но не изменены |
| Тестирование поиска | После окончания ввод в поле поиска, должны отобразится отфильтрованные данные | Тестирование прошло успешно. Данные выведены |
| Тестирование фильтра | После окончания изменения блока с фильтром. Должен сгенерироваться правильный SQL запрос. После чего должны отобразится отфильтрованные данные | Тестирование прошло успешно. Данные выведены |
| Тестирование отправки электронной почты | При вводе почты в поле с восстановлением пароля на указанную почту должно отправится сообщение с ссылкой на изменение пароля | Тестирование прошло успешно |
| Тестирование открытия страницы с изменением пароля с несуществующим в базе данных запроса на изменение пароля | После нажатия на ссылку отправленную по электронной почте должна открыться страница с сообщением «Запроса на изменение не существует» | Тестирование прошло успешно |
| Тестирование открытия страницы с изменением пароля с существующим в базе данных запроса на изменение пароля | После нажатия на ссылку отправленную по электронной почте должна открыться страница с окном для ввода нового пароля | Тестирование прошло успешно |
| Тестирование изменения пароля | После ввода пароля и его повтора, по нажатию на кнопку «Изменить » пароль должен обновится. И запрос на изменение пароля должен быть удален | Тестирование прошло успешно. Пароль обновлен. Запрос удален |

Продолжение таблицы 4.1

|  |  |  |
| --- | --- | --- |
| Тестирование диалогового модального окна «Да или Нет» | После нажатия на кнопку (например удаления резюме) должно вывестись диалоговое окно, в котором при нажатии на кнопку «Да» должна выполнится функция, а при нажатии на кнопку «Нет» окно должно закрыться | Тестирование прошло успешно |
| Тестирование кнопки «Откликнуться» | После нажатия на кнопку должно открыться окно сообщений и в поле «Получатель» должно добавится имя пользователя | Тестирование прошло успешно. Имя пользователя добавлено |
| Тестирование отправки сообщения | После заполнения всех полеq в модальном окне «Сообщения» и на нажатия на кнопку «Добавить», в базу данных должно добавится сообщение | Тестирование прошло успешно. Данные добавлены |
| Тестирование выхода из сессии | После нажатия на кнопку «Выход» сессия должна завершится и страница перезагрузится | Тестирование прошло успешно |
| Тестирование функции добавления изображения на сервер | После выбора изображения на локальном ПК, оно должно загрузится во временное хранилище на сервере, а после переместится в указанную в функции папку на, и изменить название на номер аккаунта | Тестирование прошло успешно. Файл загружен |

Окончание таблицы 4.1

|  |  |  |
| --- | --- | --- |
| Тестирование перемещения всех локальных избранных в базу данных | После нажатия на кнопку «Переместить все в веб-хранилище» все локальное избранное должно добавится в базу данных | Тестирование прошло успешно. Данные добавлены |
| Тестирование изменения динамических данных в окне вакансии | После изменения в полях, данные должны обновиться | Тестирование прошло успешно. Данные обнавлены |
| Тестирование изменения динамических данных в окне компания | После изменения в полях, данные должны обновиться | Тестирование прошло успешно. Данные обновлены |
| Тестирование разбиения на страницы | В зависимости от страницы указанной в url должен выполняться запрос на вывод данных с лимитов вывода указанным в функции | Тестирование прошло успешно |

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

# Назначение

Тема дипломного проекта «Программное средство размещения вакансий на Интернет-ресурсе» (далее по тексту SeJob).

Основной целью данного проекта является помощь людям для поиска подходящих вакансий и поиском подходящих резюме для компаний. Описание каждой компании строится из представления о наличии вакансий.

Можно отфильтровать нужные вам параметры и выбрать из них подходящий вариант, не же ли чем в бумаге искать вручную, есть возможность сохранить список выбранных вакансий. И это экономит ваше время и время работодателя.

Дипломный проект представляет собой веб-приложение предназначенное для просмотра, сортировки, поиска и сохранения нужных вакансии и резюме. Так же для администратора есть возможность манипулировать данными в базе данных с помощью PhpMyAdmin.

# Требования к компьютерному и аппаратному обеспечению

Минимальные требования, которым должен соответствовать компьютер для того, чтобы на нем работала данная программа, следующие:

* процессор: Intel Pentium III или AMD Sempron 2600+ и выше;
* объем оперативной памяти: 2 ГБ и выше;
* операционная система: ®Windows™XP, Android 4.4;
* веб-браузер Google Chrome и другие браузеры актуальных версий;
* наличие интернет-соединения.

Оптимальные требования, которым должен соответствовать компьютер для того, чтобы на нем работала данная программа, а пользователь чувствовал себя комфортно, следующие:

* процессор: IntelPentiumE5300 или AMD Sempron 3600+;
* 4 ГБ оперативной памяти;
* операционная система: ®Windows™XP/Vista/7/8.1/10, Android 4.4;
* веб-браузер Google Chrome и другие браузеры актуальных версий;
* наличие интернет-соединения.

Функционирование программы на ПЭВМ, характеристики которой удовлетворяют этим требованиям, приведет к ускорению обработки данных приложением и повышению удобства и эффективности его использования.

# Работа с приложением

Для запуска проекта на обычном локальном компьютере рекомендуется использовать пакет Open Server. Для того, чтобы запустить проект на Open Server, требуется переместить папку с проектом в директорию Open Server в папку domains.

После того, как вы переместили папку, нужно запустить Open Server, появится значок в трее, на который нужно нажать правой кнопкой мыши, далее нажать на кнопку «Запустить».

Чтобы открыть сайт сайт требуется в адресную строку браузера вести название домена (стандартный «http://sejob.atservers.net») или открыть файл «index.html» в папке с проектом.

Для манипулирования данными в базе данных рекомендуется использовать PhpMyAdmin, который имеется в Open Server или использовать MySQL Workbench скачиваемый отдельно.

Для входа на главную страницу приложения введите в url-строку адрес указанный на хостинге.

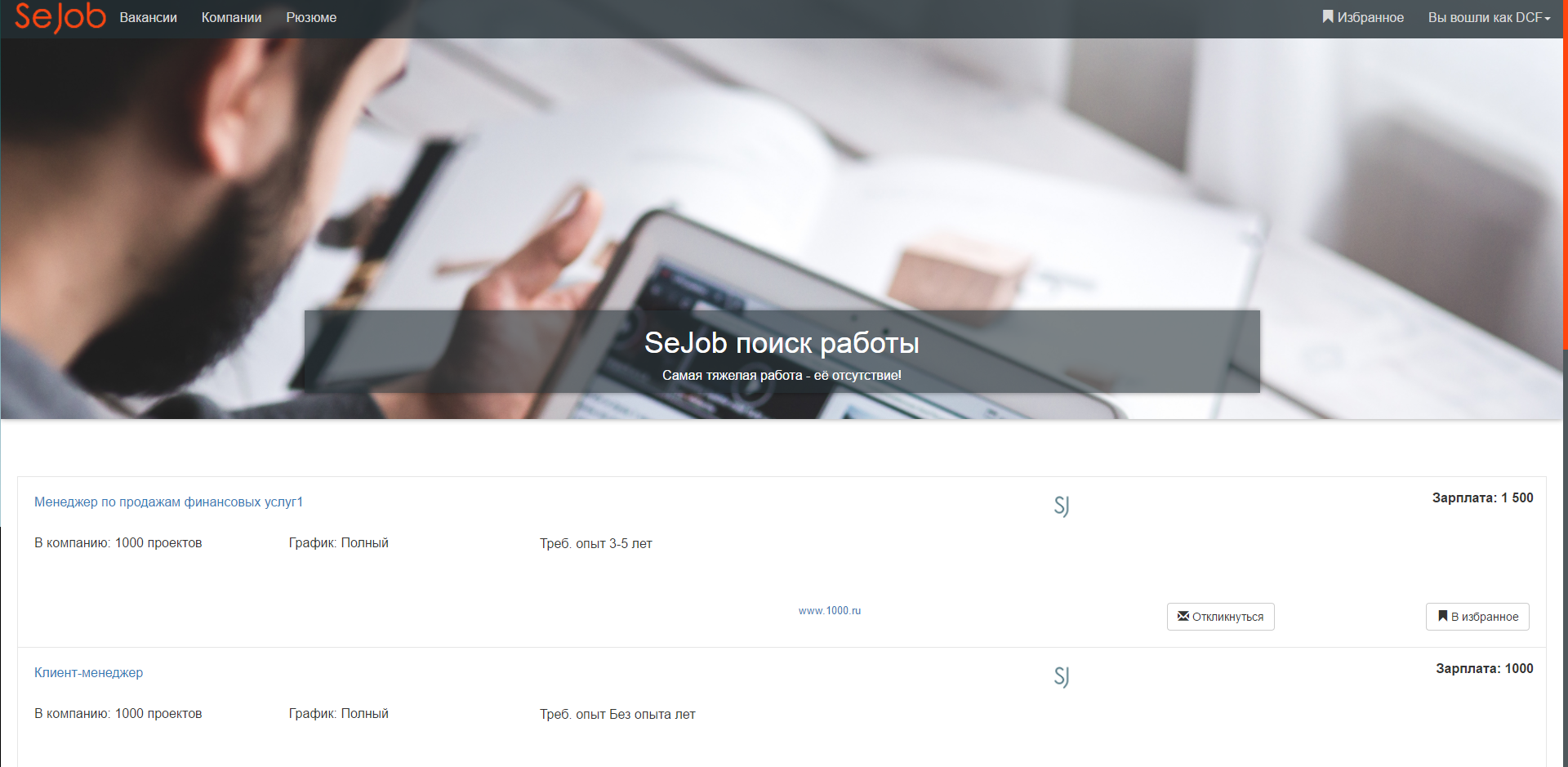


Рисунок 5.1 – Главная страница веб-приложения «SeJob»

Для регистрации аккаунта нужно нажать на кнопку «Регистрация», откроется модальное окно, в котором нужно ввести данные пользователя (см. рисунок 5.2).

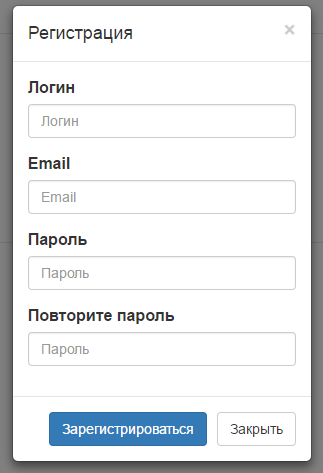


Рисунок 5.2  – Окно регистрации веб-приложения «SeJob»

Для входа в аккаунт нужно нажать на кнопку «Войти», после чего откроется окно входа в аккаунт, где нужно вписать данные указанные при регистрации (см. рисунок 5.3).

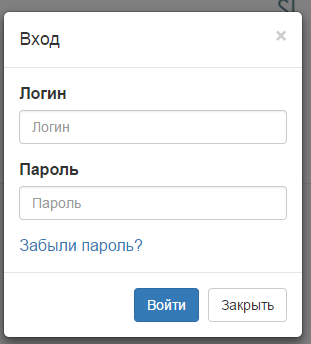


Рисунок 5.3 – Окно входа веб-приложения «SeJob»

После входа откроется возможность редактирования данных.

Так как редактирование отличается только полями ввода, оно будет показано на примере заполнения резюме.

Для того что открыть окно редактирования нужно нажать на кнопку «Вы вошли как» (см. рисунок 5.4).

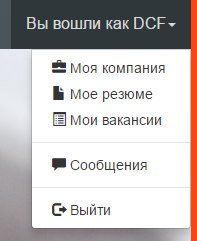


Рисунок 5.4 – Кнопка «Вы вошли как» и ее выпадающий список веб-приложения «SeJob»

И нажать на кнопку «Мое резюме», после откроется окно, в котором нужно ввести данные (см. рисунок 5.5).

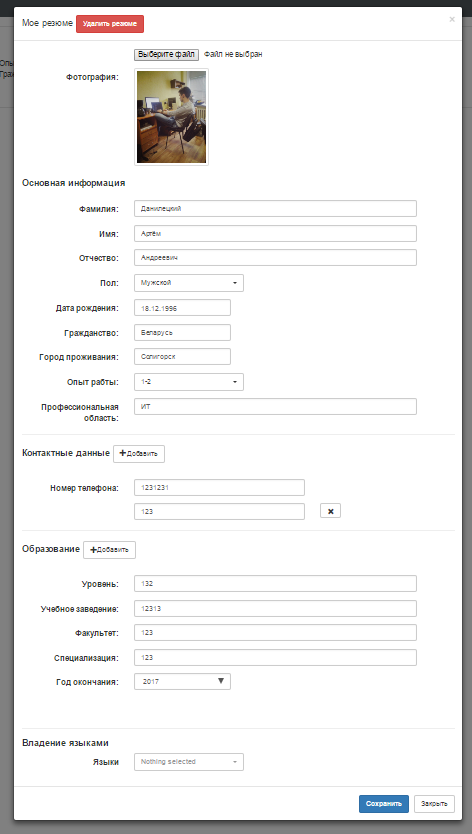


Рисунок 5.5 – Окно редактирования резюме веб-приложения «SeJob»

Далее нужно ввести данные, нажать на кнопку «Сохранить» и после «Закрыть».

Для открытия списка вакансий, резюме или компаний, нужно нажать на соответствующую кнопку (см. рисунок 5.6).

Пример списков показана на рисунках 5.7, 5.8, 5.9.

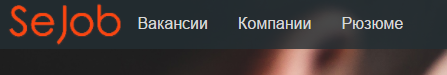


Рисунок 5.6 – Кнопки открытия списков веб-приложения «SeJob».



Рисунок 5.7 – Список вакансий веб-приложения «SeJob».

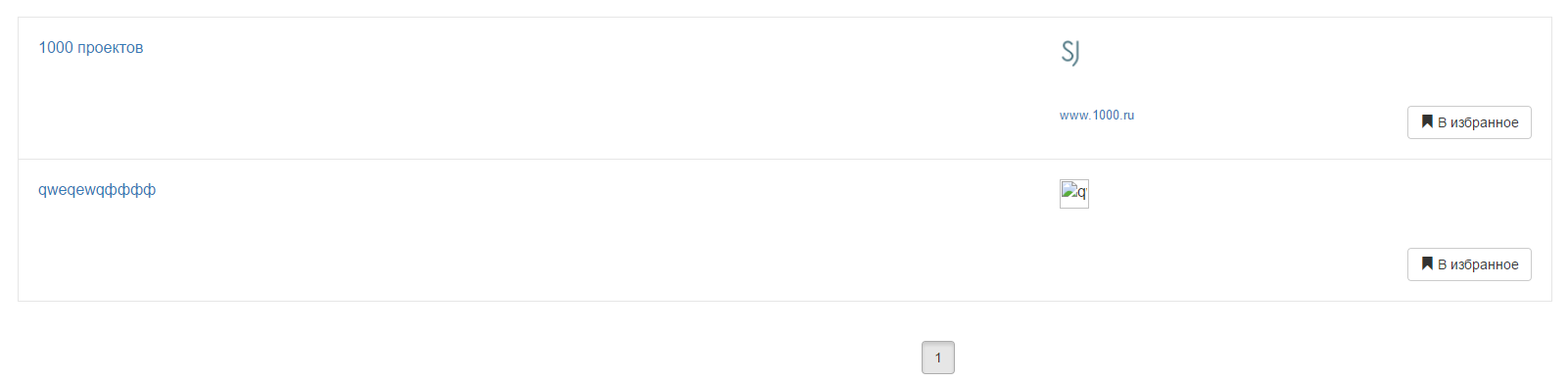


Рисунок 5.8 – Список компаний веб-приложения «SeJob».

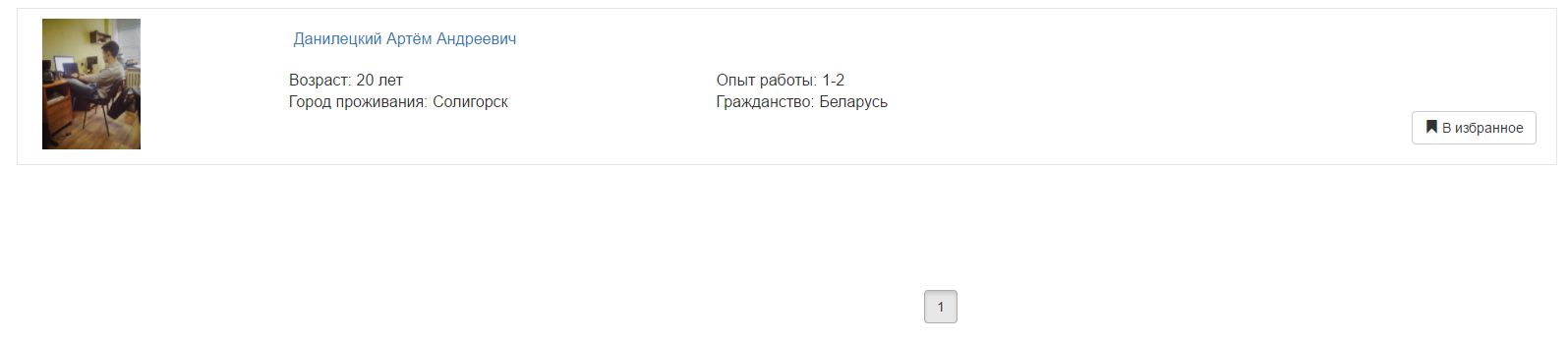


Рисунок 5.9 – Список резюме веб-приложения «SeJob».

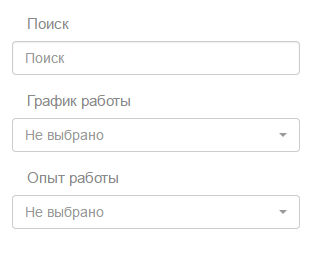


Рисунок 5.10 – Пример фильтра веб-приложения «SeJob».

Для фильтрации и поиска нужно воспользоваться блоком расположенным справа от списка (см. рисунок 5.10). В мобильной версии нужно сделать свайп влево и фильтр откроется.

Для открытия полной информации о вакансии, компании или резюме нужно нажать на ссылку с её названием или двойным щелчком по блоку.

Для добавления или удаления избранного нужно нажать на кнопку «В избранное» (см. рисунок 5.11). Эта кнопка становится зелёной, если объект находится в избранном. Если вы не зашли в сессию, то избранное сохранится на вашем компьютере.



Рисунок 5.11 – Кнопка «В избранное» веб-приложения «SeJob».

Для открытия списка избранного нужно нажать на кнопку «Избранное» (см. рисунок 5.12).



Рисунок 5.12 – Кнопка «Избранное» веб-приложения «SeJob».

Для отправки сообщения нужно нажать на кнопку «Сообщения» в списке (см. рисунок 5.4). Откроется окно, в котором нужно ввести получателя и сообщения и нажать кнопку «Отправить» (см. рисунок 5.13).

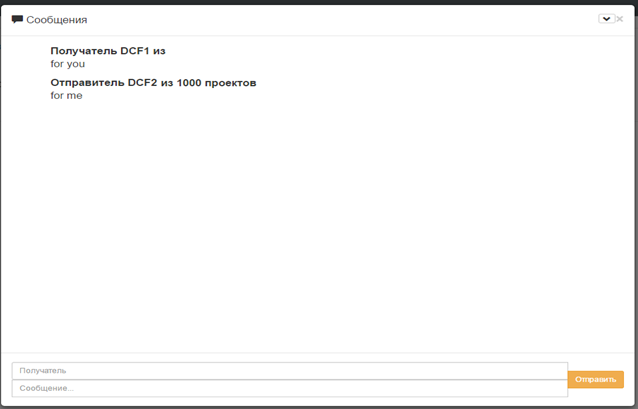


Рисунок 5.13 – Окно «Сообщения» веб-приложения «SeJob».

При нажатии на кнопку «Откликнуться» (см. рисунок 5.14)  в поле «Получатель» автоматически добавится имя пользователя которому принадлежит вакансия.



Рисунок 5.14 – Кнопка «Откликнуться» веб-приложения «SeJob».

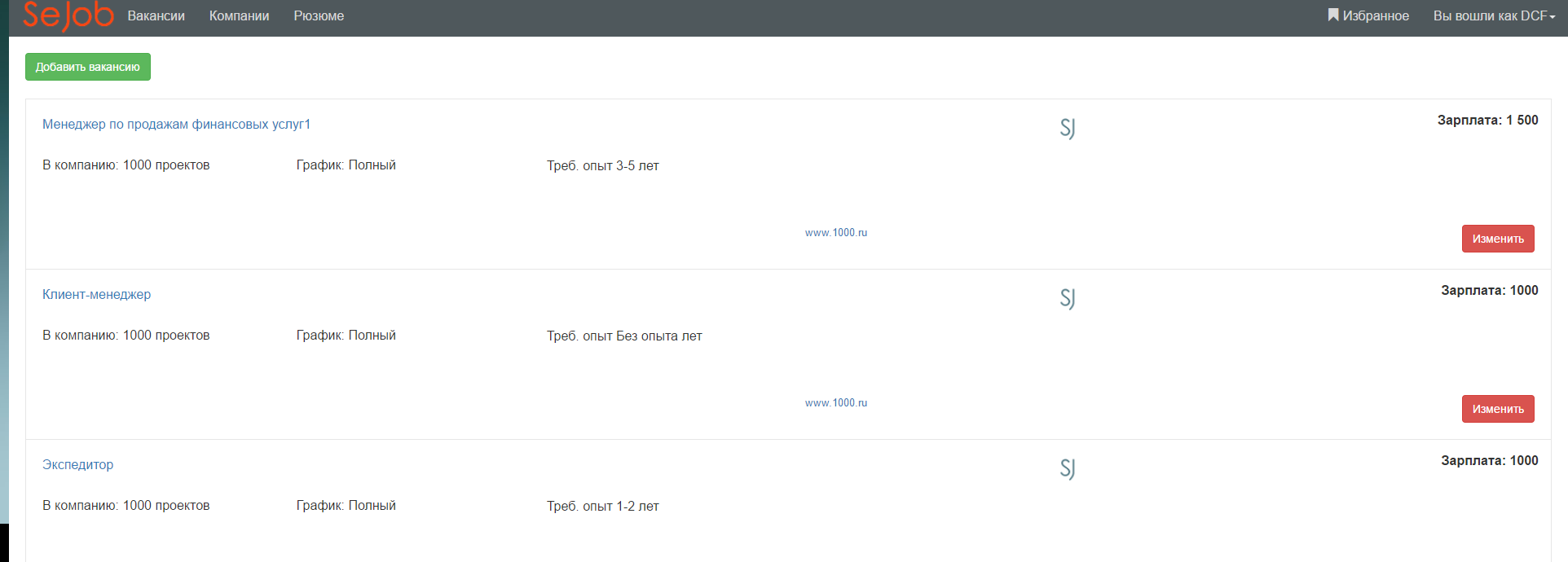


Рисунок 5.15 – Список «Мои вакансии» веб-приложения «SeJob».

Для открытия списка «Мои вакансии» (см. рисунок 5.15) нужно нажать на кнопку «Мои вакансии» (см. рисунок 5.4). Для изменения вакансии нужно нажать кнопку «Изменить», а для добавления «Добавить вакансию». Откроется окно для изменения или добавления вакансии.

Для удаления вакансии, компании или резюме нужно нажать на кнопку «Удалить» (см. рисунок 5.16).



Рисунок 5.16 – Кнопка «Удалить» веб-приложения «SeJob» на примере вакансии.

Для выхода из сессии можно нажать на кнопку «выйти» в списке (см. рисунок 5.4) или закрыть браузер.

Если вы забыли пароль для входа в аккаунт, нужно нажать на ссылку «Забыли пароль?» (см. рисунок 5.3). После чего откроется окно с вводом электронной почты аккаунта» (см. рисунок 5.17), ввести электронную почту и нажать на кнопку «Сбросить». На указанную придет сообщение с ссылкой, по которой вы должны перейти, ввести новый пароль и нажать на кнопку «Изменить».

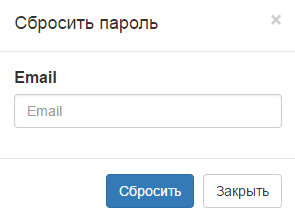


Рисунок 5.17 –Окно «Сбросить пароль» веб-приложения «SeJob».

# ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ

# Затраты на используемые материалы

В данном разделе пояснительной записки рассчитаем затраты на разработанное программное обеспечение, тема дипломного проекта «Программное средство размещения вакансий на Интернет-ресурсе».

Для начала рассчитаем затраты на используемые материалы, к ним относятся оптический диск (CD-R), бумага, канцтовары (папка-скоросшиватель), картридж с краской.

Стоимость бумаги рассчитаем по формуле:

 (6.1)

где *N*пачк – количество листов в пачке (*Nпачк* =500), лист;

*С*пачк – стоимость одной пачки бумаги (*Спачк*=6), руб.



Затраты по остальным материалам определяются по экспертным оценкам. Результаты расчёта представлены в виде таблицы (см. таблицу 6.1).

Таблица 6.1 – Расчёт стоимости материалов

|  |  |  |
| --- | --- | --- |
| Материалы | Количество, шт. | Стоимость, руб. |
| Магнитный носитель | 1 | 1,5 |
| Бумага | 100 | 1,2 |
| Папка-скоросшиватель | 1 | 3,06 |
| Краска для принтера | 1 | 36 |
| Итого | | 41,76 |

# Расчёт трудоёмкости

После того как рассчитали затраты на использованные материалы перейдем к расчету трудоёмкости.

Трудоёмкость – это затраты рабочего времени на производство единицы продукции. Так через трудоемкость в конечном итоге, оценивается один из основных затратных показателей на оплату труда исполнителей.

Общие затраты труда на разработку и внедрение изделия (проекта) ** определяют по следующей формуле:

 (6.2)

где *t****i*** – затраты труда на выполнение *i*-го этапа проекта.

Для заполнения столбцов 3, 4, 5 в таблице 6.2 воспользуемся методом экспертной оценки. При экспертной оценке ожидаемая продолжительность работ вычисляется по формуле:

 (6.3)

где *T*min – минимальная продолжительность работы;

*T*max – максимальная продолжительность работы.

Таблица 6.2 – Перечень выполняемых работ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  раработы | Стадии разработки | Трудоемкость, чел.дни | | |
| *T*min | *T*max | *T*i |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Постановка задачи | 1 | 1 | 1 |
| 2 | Сбор материалов и анализ существующих разработок | 1 | 3 | 2 |
| 3 | Подбор литературы | 1 | 1 | 1 |
| 4 | Определение требований к системе | 1 | 1 | 1 |
| 5 | Определение стадий, этапов и сроков разработки | 1 | 2 | 1 |
| 6 | Анализ программных средств схожей тематики | 1 | 2 | 1 |
| 7 | Разработка функциональной схемы программы | 2 | 3 | 2 |
| 8 | Разработка структуры программы управления по подсистемам | 3 | 5 | 4 |
| 9 | Документирование | 3 | 4 | 3 |
| 10 | Определение требований к программе управления | 1 | 1 | 1 |
| 11 | Выбор инструментальных средств | 1 | 2 | 1 |
| 12 | Определение свойств и требований к аппаратному обеспечению | 2 | 2 | 2 |
| 13 | Программирование | 15 | 20 | 20 |
| 14 | Тестирование и отладка программы управления | 3 | 5 | 3 |
| 15 | Разработка программной документации | 2 | 3 | 2 |
| 16 | Согласование и утверждение работоспособности системы | 2 | 3 | 2 |
| 17 | Опытная эксплуатация | 3 | 5 | 4 |
| 18 | Анализ данных, полученных в результате эксплуатации | 3 | 5 | 4 |

Окончание таблицы 6.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 19 | Корректировка технической документации по результатам испытаний | 3 | 5 | 4 |
| Итог | | | | 59 |

Воспользуемся формулой общие затраты труда на разработку и внедрение проекта (см. формулу 6.2) и рассчитаем общие затраты на разработку продукта, исходя из таблицы 6.2.

*Qp*=1+2+1+1+1+1+2+4+3+1+1+2+21+3+2+2+4+4+4=59 дней.

# Построение графика разработки программного продукта

В качестве инструмента планирования работ используем ленточный график. Ленточный график позволяет наглядно представить логическую последовательность и взаимосвязь отдельных работ, срок начала и срок окончания работ.

Он представляет собой таблицу, где перечислены наименования стадий разработки и видов работ, длительность выполнения каждого вида работ. Продолжением таблицы является график, отражающий продолжительность каждого вида работ в виде отрезков времени, которые располагаются в соответствии с последовательностью выполнения работ (см. таблицу 6.3).

Таблица 6.3 – Ленточный график

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Наименование работ | Календарные месяцы, дни | | | | | | | | |
| Апрель | | | Май | | | Июнь | | |  |  |
| 0 10 20 30 | | | 10 20 30 | | | 10 20 30 | | |
| ТЗ |  |  |  |  |  |  |  |  |  |
| ЭП |  |  |  |  |  |  |  |  |  |
| ТП |  |  |  |  |  |  |  |  |  |
| РП |  |  |  |  |  |  |  |  |  |
| ВП |  |  |  |  |  |  |  |  |  |

# Расчёт основной заработной платы

Основная заработная плата – заработная плата, начисленная рабочим и служащим за фактически проработанное время и выполнение работы на предприятии, рассчитывается по формуле:

 (6.4)

где *Т*ЗАН – число дней, отработанных исполнителем проекта; *О*ДН – дневной оклад исполнителя.

Величина месячного оклада определяется ставкой тарифного разряда так называемой Единой Тарифной Сетки (ЕТС). Тарификация специалистов по ЕТС в соответствии с Инструкцией о порядке применения Единой тарифной сетки работников Республики Беларусь относит техника-программиста со средне специальным образованием без опыта работы в диапазоне с 7 по 10 разряд. Отнесем техника-программиста без опыта работы к 7-му разряду ЕТС.

На предприятии тарифная ставка первого разряда равна 31 рубль. Тарифные коэффициенты представлены в единой тарифной сетке работников РБ (см. таблицу 6.4).

*О*мес = 31 \* 2,03 = 62,93 руб.

Исходя из этого, дневной оклад исполнителя будет равен:

*О*дн = 62,93/ 21= 3 руб.

Таблица 6.4 – Единая тарифная сетка работников РБ

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Разряды | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Коэфф. | 1 | 1,16 | 1,35 | 1,57 | 1,73 | 1,9 | 2,03 | 2,17 | 2,32 | 2,48 | 2,65 |

Теперь можно рассчитать основную ЗП, по формуле:

*С* З.ОЗН = 3 \* 59 = 177 руб.

Умножим на повышающий коэффициент, который в данном случае будет равен 2.

*С*З.ОЗН = 177 \* 2 = 354 руб.

# Отчисления в фонд социальной защиты населения

Отчисления в фонд социальной защиты населения (*З*фсзн) определяются в соответствии с действующими законодательными актами по нормативу, в процентном отношении к фонду основной и дополнительной зарплаты исполнителей, установленному в целом по организации и рассчитывается по формуле:

 (6.5)

где *Н*оз – норматив отчислений в фонд социальной защиты населения (%).



# Дополнительная заработная плата

Зная основную заработную плату можно рассчитать дополнительную заработную плату.

Дополнительная заработная плата – заработная плата, начисленная рабочим и служащим не за фактически выполненные работы или проработанное время, а в соответствии с действующим законодательством включает выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью исполнителей). Дополнительная ЗП определяется по нормативу в процентах к основной заработной плате по формуле:

 (6.6)

где *З*Дi– дополнительная заработная плата исполнителей на конкретное ПС (руб.); *Н*Д – норматив дополнительной заработной платы рассчитывается по формуле:



Налоги, рассчитываемые от фонда оплаты труда, определяются в соответствии с действующими законодательными актами по нормативам, в процентном отношении к сумме основной и дополнительной заработной платы, относимой на ПС (Не) и рассчитывается по формуле:

 (6.7)

где *Н*не – норматив налога, уплачиваемого единым платежом (%).



Затраты, связанные с обеспечением работ оборудованием, рассчитаем как амортизационные отчисления, т.к. оборудование специально не закупалось.

При создании программного продукта используется стационарный компьютер (Intel Core i5-4690k , 8GB DDR3, DVD-Multi Recorder).

Закупочная цена ПЭВМ составила – 1500 руб. Примем норму амортизации на технические средства На = 20%, на основании трёх лет.

Для создания пакета программ, являющегося конечным результатом исследований, применялось следующее программное обеспечение:

* платформа Windows 10.

Общая стоимость программного обеспечения *Ц*по составляет 280 рублей.

Общую стоимость технических средств и программного обеспечения *Цо*, руб., определим по формуле:

*Ц*о *= Ц*тс*+Ц*по (6.8)

*Цо* = 1500+280=1780 руб.

Годовые амортизационные отчисления на полное восстановление технических средств и программного обеспечения *Ао*, руб., рассчитываются по формуле:

*А*о *= Ц*о*\*Н*а (6.9)

*А*о = 1780\*0,2=356 руб.

Амортизационные отчисления *А*п, руб. за период создания программного продукта определим по формуле:

*А*п *= А*о*/К*рг*\*К*дн (6.10)

где *К*дн = 59 дн. – количество отработанных дней;

*К*рг = 280 дн. – количество рабочих дней в году.

*А*п = 356\*60/280 = 75,01 руб.

# Расчёт расходов на энергопотребление

Так же необходимо рассчитать расходы электроэнергии. ПЭВМ, на которой была разработана программа, является потребителем электрической энергии сети переменного тока, напряжением 220 В.

Расход денежных средств *Р*э, руб., связанный с энергопотреблением технических средств можно вычислить по формуле:

*Р*э*=К*дн*\*В*раб*\* М*с *\* Ц*эн (6.11)

где *К*дн– период написания программы, дн.; *В*раб – длительность рабочей смены, ч.;

*М*с – мощность, потребляемая техническими средствами, кВт·ч;

*Ц*эн – стоимость электроэнергии по действующим тарифам (0,1013).

*Р*э = 59\*6\*0,2\*0,1013=7,17 руб.

# Общая сумма расходов по всем статьям сметы

Общая сумма расходов по всем статьям сметы (*СРi*) на ПС рассчитывается по формуле:

 (6.12)



# Расходы по статье «Прочие затраты»

Расходы по статье «Прочие затраты» определяются по нормативу, разрабатываемому в целом по научной организации, в процентах к основной заработной плате (см. формулу 6.13)

 (6.13)

где *Н*ПЗ – норматив прочих затрат в целом по научной организации.



# Рентабельность и прибыль

Рентабельность и прибыль **-** определяются исходя из результатов анализа рыночных условий, переговоров с заказчиком (потребителем) и согласования с ним отпускной цены, включающей дополнительно налог на добавленную стоимость. Рассчитывается по формуле:

 (6.14)

где *П*р – прибыль от реализации ПС заказчику (руб.);

*У*пр – уровень рентабельности ПС (%);

*С*п – себестоимость ПС (руб.).



# Прогнозируемая цена ПС без налогов

Прогнозируемая цена ПС без налогов (*Ц*п) рассчитывается, как сумма расходов по всем статьям смены и прибыли.

Рассчитывается по формуле:

. (6.15)



Необходимо рассчитать налог на добавленную стоимость. Поскольку мы уже рассчитали прогнозируемую стоимость и нам известно, что % НДС = 20% то расчет проведем по формуле:

 (6.16)

где *С*тндс – норматив НДС(%).



# Прогнозируемая отпускная цена

Прогнозируемая отпускная цена рассчитывается, как сумма прогнозируемой цены ПС без налогов и налог на добавленную стоимость(НДС).

Рассчитывается по формуле:

 (6.17)

*Ц*о*=* 871,29 +174,39 = 1045,68 *руб.*

# Расчет чистой прибыли

Имея все необходимые данные, рассчитаем чистую прибыль.

Рассчитывается по формуле:

 (6.18)

где *%Н*п – 18%.



*Ч*п *= П*р *– Н*п (6.19)

*Ч*п = 145,32 – 26,16 = 119,16 руб.

# Выводы по организационно-экономической части дипломного проекта

После произведенных расчетов, было выявлено, что разработка программного продукта составила 59 дней. Основная и дополнительная заработная плата работника составляет – 407,1 руб. Обязательные отчисления в ФСЗН составляет – 138,41 руб. Амортизационные отчисления за период создания программного обеспечения составляет – 75,01 руб. Расходы на энергопотребление, а также расходы на прочие затраты составляют соответственно – 7,17 руб. и 53,1 руб. Чистая прибыль за разработку программного обеспечения составляет – 119,16 руб. Отпускная цена программного продукта – 1045,68. Данный реализованный проект выгодный.

# ОХРАНА ТРУДА И ОКРУЖАЮЩЕЙ СРЕДЫ

# Правовые, нормативные, социально-экономические и организационные вопросы охраны труда

Охрана труда – это система обеспечения безопасности жизни и здоровья работников в процессе трудовой деятельности, включающая в себя правовые, социально-экономические, организационно-технические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия и средства.

Здоровые и безопасные условия труда способствуют повышению производительности, удовлетворенности  работников своим  трудом, созданию хорошего психологического климата в трудовых коллективах, что ведет к снижению текучести кадров, созданию стабильных трудовых коллективов [8].

Недостатки в работе по охране труда обуславливают значительные экономические потери. Заболеваемость и травматизм работников, затраты на компенсации за работу в неблагоприятных условиях труда приводят к ухудшению экономических результатов работы предприятия.

Основные законодательные акты регулирующие охрану труда: Конституция, закон Республики Беларусь «Об охране труда», Трудовой кодекс Республики Беларусь, законы Республики Беларусь, декреты президента, постановления правительства и другие.

Согласно законодательству, каждый работник имеет право на благоприятные условия труда, а работодатель обязан создать эти условия.

При работе с видеодисплейным терминалом (далее ВДТ) или персональной электро-вычислительной машиной (далее ПЭВМ) происходит воздействие вредных факторов (излучение, блики, шум, зрительное напряжение, нервно-эмоциональные перегрузки и др.) на организм работника, для предотвращения и минимизации их воздействия необходимо организовать рабочее место и труд работника в соответствии с Санитарными нормами и правилами, а также с правилами безопасности. Правильная организация рабочего места увеличивает производительность труда.

# Требования к помещениям для эксплуатации ВДТ и ПЭВМ

Помещения для эксплуатации ВДТ и ПЭВМ должны иметь естественное и искусственное освещение.

Запрещается выполнение основной работы с использованием ВДТ и ПЭВМ на постоянных рабочих местах без естественного освещения, если это не обусловлено технологическим процессом.

Естественное освещение на рабочих местах с ВДТ и ПЭВМ должно осуществляться через световые проемы, ориентированные преимущественно на север, северо-восток, восток, запад или северо-запад и обеспечивать коэффициент естественной освещенности не ниже 1,5 %.

Оконные проемы должны быть оборудованы регулируемыми устройствами типа жалюзи, занавесей, внешних козырьков и другое.

Площадь одного рабочего места для пользователей ВДТ и ПЭВМ на базе плоских дискретных экранов (жидкокристаллические, плазменные и другое) должна составлять не менее 4,5 м2.

Площадь одного рабочего места для пользователей ВДТ и ПЭВМ на базе электронно-лучевой трубки (далее – ЭЛТ) должна составлять не менее 6 м2.

Запрещается размещать рабочие места с ВДТ и ПЭВМ на расстоянии менее 10 м от силовых кабелей, вводов и высоковольтных трансформаторов.

Помещения, в которых для работы используются преимущественно ВДТ и ПЭВМ, не должны граничить с помещениями, в которых уровни шума и вибрации превышают нормируемые значения для данной категории проводимых в них работ и их типа. А так же поверхность пола должна быть ровной, нескользкой, без выбоин, удобной для очистки и влажной уборки, обладать антистатическими свойствами. Внутреняя отделка должна осуществлятся при помощи диффузно отражающих материалов с коэффициентом отражения для потолка – 0,7-0,8; для стен – 0,5-0,6; для пола – 0,3-0,5.

Помещения с ВДТ и ПЭВМ должны оборудоваться системами отопления, кондиционирования воздуха или эффективной приточно-вытяжной вентиляцией.

Помещения, где размещаются рабочие места с ВДТ и ПЭВМ, должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации[11].

# Требования к производственному оборудованию и рабочим местам

Оборудование должно удовлетворять следующим требованиям:

* конструкция ВДТ и ПЭВМ должна обеспечивать возможность поворота корпуса в горизонтальной и вертикальной плоскости с фиксацией в заданном положении для обеспечения фронтального наблюдения экрана, должна предусматривать регулирование яркости и контрастности;
* дизайн ВДТ и ПЭВМ и периферийных устройств должен предусматривать окраску в мягких тонах, иметь матовую поверхность, и не иметь блестящих деталей, создающих блики;
* ВДТ и ПЭВМ и периферийные устройства подлежат государственной санитарно-гигиенической экспертизе на соответствие требованиям;
* допустимые визуальные параметры устройств отображения ВДТ и ПЭВМ, а также уровни физических факторов (уровни вибрации, ультрафиолетового, инфракрасного, мягкого рентгеновского излучения и др.) должны быть в пределах допустимых уровней.

Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах 680-800 мм. При отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

При размещении нескольких рабочих мест с ВДТ и ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора) должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов – не менее 1,2 м.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. Экран видеомонитора должен находиться на расстоянии 600-700 мм от глаз пользователя, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов. Клавиатуру следует располагать на поверхности стола на расстоянии 100-300 мм от края, обращенного к взрослому пользователю, или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

Поверхность сиденья, спинки и других элементов рабочего стула должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений. Не допускается использование табуреток, скамеек без опоры вместо стульев.

# Обеспечение электробезопасности при работе с ВДТ и ПЭВМ

Рабочее помещение с ВДТ и ПЭВМ относится к классу помещений без повышенной опасности с точки зрения поражения электрическим током. Безопасность для работников в помещениях с ВДТ и ПЭВМ обеспечивается следующим образом:

* применение надлежащей изоляции;
* соблюдение соответствующих расстояний до токоведущих частей, ограничение доступа к ним;
* заземления или зануления корпусов электрооборудования, которые могут оказаться под напряжением вследствие повреждения изоляции.

Зоной повышенной электроопасности являются места подключения электроприборов. Нередко подключающие розетки располагают на полу, что недопустимо. Часто совершается другая ошибка – перегрузка розеток по мощности, и, как следствие, происходит нарушение изоляции, приводящее к короткому замыканию.

Для предотвращения поражения электрическим током не допускается:

* включение сильно охлаждённого (принесённого в зимнее время с улицы) оборудования;
* производить влажную уборку при включённом оборудовании;
* при включённом питании прикасаться к панелям с разъёмами оборудования, разъёмам питающих и соединительных кабелей, экрану монитора.

# Требования производственной санитарии

Производственная санитария – это система организационных мероприятий и технических средств, предотвращающих или уменьшающих воздействие на работающих вредных производственных факторов.

Основными вредными факторами при работе с ВДТ, ЭВМ и ПЭВМ являются:

* напряжение зрения, памяти и внимания;
* длительное статическое напряжение;
* монотонность труда;
* эмоциональные перегрузки;
* запылённости воздуха рабочей зоны;
* повышенные уровни электромагнитного, инфракрасного и ультрафиолетового излучения, шума и вибрации;
* повышенный или пониженный уровень освещения рабочей зоны.

Для снижения уровня вредных факторов, перечисленных выше, следует обеспечить в холодный период года температуру 22-24 °С, влажность воздуха 40-60 %, скорость движения воздуха 0.1 м/с – 0.3 м/с, а для теплого периода года температуру 23-25 °С, влажность воздуха 40-60 %, скорость движения воздуха 0.1 м/с – 0.3 м/с.

В связи с тем, что состояние производственного воздуха в значительной мере зависит от эффективности его обмена, необходимо уделять вентиляции помещения.

В соответствии с требованиями действующего законодательства работники должны обеспечиваться гардеробными, умывальниками, душевыми, помещениями для личной гигиены женщин, туалетами.

В помещениях, оборудованных ВДТ, ЭВМ и ПЭВМ, должна проводиться ежедневная влажная уборка и систематическое проветривание после каждого часа работы с ВДТ, ЭВМ и ПЭВМ.

Нормирование параметров микроклимата в рабочей зоне производится в зависимости от времени года, категории работ по энергозатратам и избытку явного тепла.

Содержание вредных химических веществ в воздухе помещений, предназначенных для использования ВДТ, ЭВМ и ПЭВМ не должно превышать предельно допустимых среднесуточных концентраций для атмосферного воздуха, в соответствии с нормативами предельно допустимых концентраций, загрязняющих веществ в атмосферном воздухе и ориентировочно безопасных уровней воздействия загрязняющих веществ в атмосферном воздухе населенных пунктов.

Для оптимизации микроклимата в помещении с ВДТ, ЭВМ и ПЭВМ используется:

* кондиционер – устройство для поддержания оптимальных климатических условий;
* увлажнитель воздуха – устройство для повышения влажности воздуха.
* вентиляционная система – совокупностью устройств для обработки, транспортирования, подачи и удаления воздуха;
* система отопления – комплекс устройств, выполняющих функцию отопления;

По принципу организации производственное освещение подразделяется на естественное, искусственное и совмещенное.

Рабочие столы следует размещать таким образом, чтобы экраны ВДТ, ЭВМ или ПЭВМ были ориентированы боковой стороной к световым проемам (исключение составляет периметральная расстановка рабочих мест), чтобы естественный свет падал преимущественно слева.

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 люкс. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 люкс[11].

Необходимо ограничивать отраженную блесткость на рабочих поверхностях (экран, стол, клавиатура и другое) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ВДТ и ПЭВМ не должна превышать 40 кд/м2.

Искусственное освещение в помещениях для эксплуатации ВДТ, ЭВМ и ПЭВМ должно осуществляться системой общего равномерного освещения. В административных помещениях в случаях преимущественной работы с документами следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

Также для обеспечения нормируемых значений освещенности в помещениях для использования ВДТ, ЭВМ и ПЭВМ следует проводить чистку стекол оконных рам и светильников со своевременной заменой перегоревших ламп.

Звукоизоляция ограждающих конструкций помещений с ВДТ, ЭВМ и ПЭВМ должна обеспечивать нормируемые параметры шума в них.

При выполнении основной работы на ВДТ, ЭВМ и ПЭВМ (дис­петчерские, операторские, расчетные кабины и посты управления, залы вычислительной техники и др.), уровень шума на рабочем месте не должен превышать 50 дБ [11].

Для снижения шума применяют различные методы коллективной защиты: уменьшение уровня шума в источнике его возникновения; рациональное размещение оборудования; борьба с шумом на путях его распространения, в том числе изменение направленности излучения шума, использование средств звукоизоляции, звукопоглощение и установка глушителей шума, в том числе акустическая обработка поверхностей помещения.

Для ослабления передачи вибрации от источников ее возникновения полу, рабочему месту, сиденью, рукоятке и так далее, широко применяют методы виброизоляции в виде виброизоляторов из резины, пробки, войлока, асбеста, стальных пружин.

Важным фактором для снижения опасного воздействия вибрации на организм человека является правильная организация режима труда и отдыха, постоянное медицинское наблюдение за состоянием здоровья, лечебно-профилактические мероприятия.

Допустимые уровни напряженности (плотности потока мощности) электромагнитных полей, излучаемых клавиатурой, системным блоком, компьютерной мышью, беспроводными системами передачи информации на расстояния и иными вновь разработанными устройствами в зависимости от основной рабочей частоты изделия, не должны превышать значений.

Для того, чтобы избежать вредного влияния электромагнитного излучения, следует соблюдать следующие правила:

* использовать жидкокристаллический монитор, поскольку его излучение значительно меньше, чем у распространённых ЭЛТ мониторов (монитор с электроннолучевой трубкой);
* в связи с тем, что электромагнитное излучение от стенок монитора намного больше, по возможности монитор поставить в угол, так что бы излучение поглощалось стенами;
* выполнять ежедневную влажную уборку помещения;
* проветривание помещений, а в теплое время года работа с открытыми фрамугами;
* экран монитора должен быть удален от глаз минимум на 50-60 сантиметров.
* не игнорировать регламентированные перерывы.

# Организация режима труда и отдыха при работе с ВДТ и ПЭВМ

Для обеспечения оптимальной работоспособности и сохранения здоровья профессиональных пользователей, на протяжении рабочего дня (смены) должны устанавливаться регламентированные перерывы.

К работе допускаются только те, кто прошёл медицинскую комиссию и не имеет никаких противопоказаний.

Режимы труда и отдыха при работе с ВДТ и ПЭВМ должны определяться видом и категорией трудовой деятельности.

Виды трудовой деятельности разделяются на 3 группы:

* группа А – работа по считыванию информации с экрана ВДТ, ЭВМ или ПЭВМ с предварительным запросом;
* группа Б – работа по вводу информации;
* группа В – творческая работа в режиме диалога с ЭВМ.

Вид трудовой деятельности относится к группе В – творческая работа в режиме диалога с ЭВМ.

Продолжительность непрерывной работы с ВДТ и ПЭВМ без регламентированного перерыва не должна превышать двух часов.

При восьмичасовом рабочем дне (смене) и работе на ВДТ, ЭВМ и ПЭВМ регламентированные перерывы следует устанавливать:

* для I категории работ – через 2 часа от начала рабочего дня (смены) и через 2 часа после обеденного перерыва продолжительностью 15 минут каждый;
* для II категории работ – через 2 часа от начала рабочего дня (смены) и через 1,5-2 часа после обеденного перерыва продолжительностью 15 минут каждый или продолжительностью 10 минут через каждый час работы;
* для III категории – через 1,5-2 часа от начала рабочего дня (смены) и через 1,5-2 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

При двенадцатичасовом рабочем дне (смене) регламентированные перерывы должны устанавливаться в первые 8 часов работы аналогично перерывам при восьмичасовом рабочем дне (смене), а в течении последних 4 часов работы, независимо от категории и вида работ, каждый час продолжительностью 15 минут.

# Обеспечение пожарной безопасности

Пожарная безопасность – состояние объекта, при котором с регламентирующей вероятностью исключается возможность возникновения и развития пожара, а также обеспечивается защита людей  и материальных ценностей от воздействия опасных факторов.

Воспламенившаяся электронная техника, находится под напряжением и следовательно имеет электронные компаненты, поэтому для устранения пламени используются огнетушители, предназначенные для устранения пламени на объектах, которые находятся под напряжением. Для устранения пламени используются углекислотные огнетушители типов ОУ-2, ОУ-5, ОУ-8.

Достоинством углекислотного огнетушителя, является высокая эффективность тушения пожара и сохранность электронного оборудования.

Обеспечение пожарной безопасности является обязанностью руководителей, соответствующих должностных лиц и работников, а также граждан. Эти обязанности должны быть отражены в соответствующих инструкциях [6].

Для обеспечения пожарной безопасности на рабочем месте необходимо соблюдать следующие требования:

* необходимо наличие сигнализации для звукового оповещения о пожаре, средств связи для вызова пожарной службы;
* у телефонных аппаратов на видных местах должны быть вывешены таблички с указанием номера вызова пожарных аварийно-спасательных подразделений, а также подробный план этажа здания на случай экстренной эвакуации;
* проходы, подходы к средствам пожаротушения, места включения и выключения электрооборудования не должны быть загромождены прочими вещами;
* все работники обязаны знать правила и порядок применения первичных средств пожаротушения и меры безопасности при пользовании ими;
* необходимо наличие первичных средств пожаротушения (углекислотный огнетушитель).

Каждый работник независимо от занимаемой должности обязан знать и выполнять правила пожарной безопасности, не допускать действий, которые могут привести к возникновению пожара. В связи с этим со всеми работниками должен проводиться вводный, первичные и повторные на рабочем месте инструктажи по пожарной безопасности (повторный не реже 1 раза в 6 месяцев) [10].

# Обеспечение охраны окружающей среды

Основные принципы охраны окружающей среды допускают воздействие предприятий на природную среду, исходя из требований в области охраны окружающей среды. При этом снижение негативного воздействия на окружающую среду должно достигаться на основе использования наилучших существующих технологий с учетом экономических и социальных факторов.

Технологический процесс, осуществляемый при работе с ВДТ и ПЭВМ, не должен наносить вреда окружающей среде. С этой целью при производстве работ необходимо обязательно предусматривать комплекс соответствующих мероприятий применительно к данной работе [7].

Переработка устаревшей электроники требует значительных усилий и представляет собой немалую проблему. Следует бережно относиться к оборудованию, и в случае поломки отправить на ремонт, а не заменять новым.

В число конкретных мер по улучшению охраны окружающей среды входят:

* меры по экономии энергии – замена энергоемких устройств, использование энергоэффективных компьютеров и мониторов или оптимизированного освещения в административном помещении, как, собственно, и в производственном;
* правильная сортировка отходов с соблюдением строгих норм, превышающих требования законодательства и утилизация устаревшей сложной техники;
* создание производственно-материальной базы для такой работы;
* повышение уровня охвата населения системой экологического просвещения, образования, подготовки кадров.

При выполнениие всех мероприятий окружающая среда станет более благопрятна для работника. Воздействие вредний факторов на окружающую среду уменьшиться, следовательно качество работы улучшится.

# ЗАКЛЮЧЕНИЕ

Представленная версия приложения является законченным программным продуктом.

Разработанное программное средство автоматизирует поиск работы, с его помощью можно осуществить поиск, фильтрацию, добавить вакансии в избранное, также можно связаться с лицом компании, создавшим вакансию, посредством текстовых сообщений. Позволяет облегчить поиск работы для работодателя и соискателя. На основании данных можно выбрать оптимальную вакансию, компанию или резюме.

Теперь работодатель или соискатель может искать резюме удовлетворяющее его критериям, выбрав нужные параметары в фильтре. Также ему не придется записывать нужную информацию, он может добавить данные в избранное.

В результате выполнения проекта были реализованные поставленные задачи, среди которых интуитивный и приятный интерфейс, а также набор функций для удобного и быстрого поиска и фильтрации.

Проект «Программное средство размещения вакансий на интернет-ресурсе» отлично подходит для решения поставленных задач.

Программный продукт работает под управлением любой операционной системы, главное требование – это наличие актуальной версии веб-браузера, а также интернет соединения.

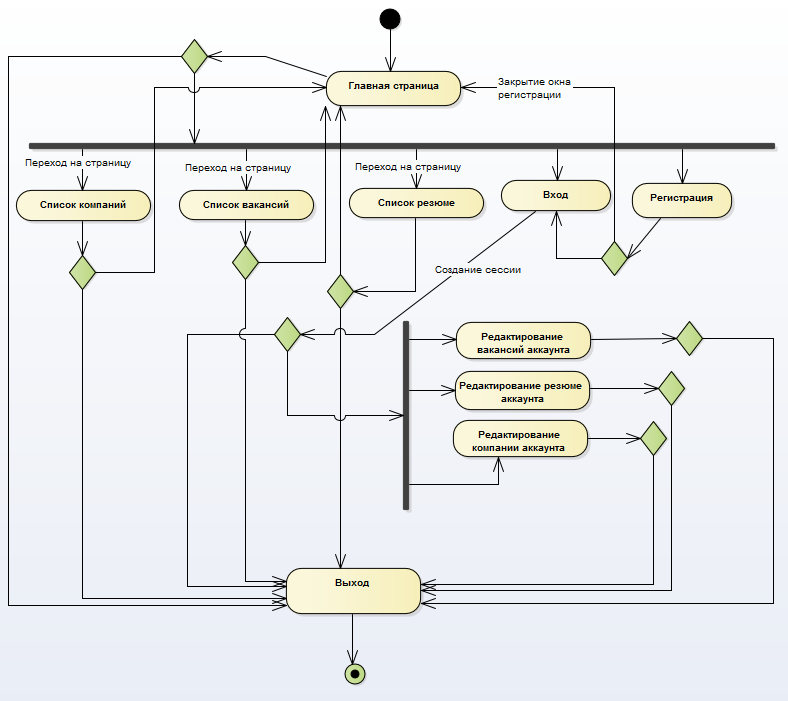
# ЛИТЕРАТУРА

* + - 1. HTML5.BY [Электронный ресурс]. Руководство по HTML. Режим доступа: http://html5.by.
      2. jquery-docs.ru [Электронный ресурс]. Руководство по JQUERY. Режим доступа: https://jquery-docs.ru.
      3. PHP.NET [Электронный ресурс]. Руководство по PHP. Режим доступа: http://php.net.
      4. Вейнер П. – Языки программирования JAVA и JavaScript. – М: ЛОРИ, 2000.
      5. Гарнаев А. – Web-программирование на Java и JavaScript. – СПб.: БХВ Санкт-Перебург, 2002.
      6. Закон РБ «О пожарной безопасности».
      7. Закон РБ «Об охране окружающей среды».
      8. Закон РБ «Об охране труда».
      9. Лешек А. – Анализ и проектирование информационных систем с помощью UML 2.0. 3-е издание – Москва, Санкт-Петербург, Киев, 2008.
      10. Нормы пожарной безопасности РБ. Категорирование помещений, зданий и наружных установок по взрывопожарной и пожарной опасности.
      11. Постановление Министерства здравоохранения РБ №59 от 28 июня 2013 Санитарные нормы и правила «Требования при работе с видеодисплейными терминалами и электронно-вычислительными машинами».
      12. Робин Никсон. – Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 4-е издание – БХВ-Петербург, 2016.
      13. Хабрахабр [Электронный ресурс]. Электронные данные. Режим доступа: https://habrahabr.ru.

# Приложение А

**Диаграмма деятельности**

*(обязательное)*

****

# Приложение Б

**Диаграмма последовательности**

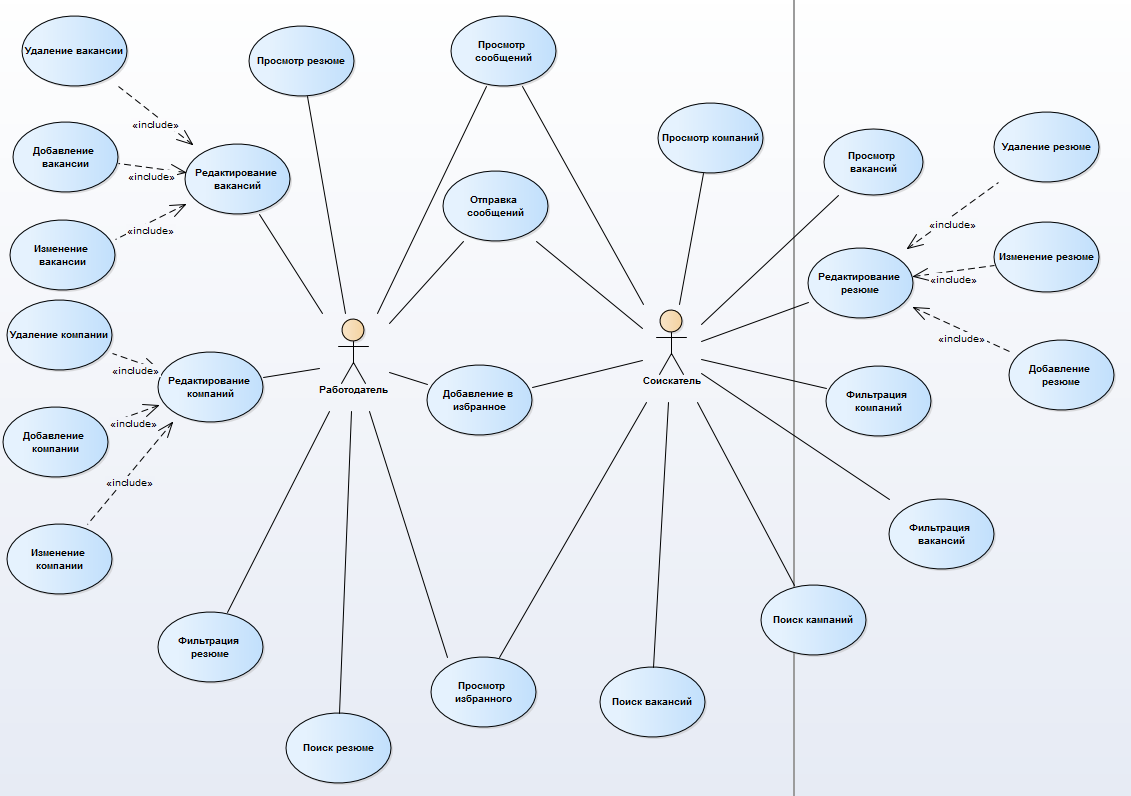
*(обязательное)*

# unnamed

# Приложение В

**Диаграмма вариантов использования**

*(обязательное)*

****

# Приложение Г

**Листинг программы**

*(обязательное)*

function session\_check() {

$.ajax({

url: "./includes/session.php",

method: "POST",

success: function (data) {

$(".session\_content").html(data);

}

});

}

//function f\_update() {

// $.ajax({

// url: "./includes/favoritecontent.php",

// method: "POST",

// success: function (data) {

// $(".favorite\_content").html(data);

// }

// });

//}

if (device.mobile() == true || device.tablet() == true) {

$(function () {

$(".container-fluid").swipe({

swipeLeft: function (event, direction, distance, duration, fingerCount) {

$("#filter").hide("fast");

$(".filter").removeClass("hidden-xs hidden-sm col-md-2 col-lg-2");

$(".filter").addClass("col-xs-12 col-sm-12 col-md-12 col-lg-12");

$(".live\_data").removeClass("col-xs-12 col-sm-12 col-md-10 col-lg-10");

$(".live\_data").addClass("hidden-xs hidden-sm hidden-md hidden-lg");

$("#filter").show("fast");

},

swipeRight: function (event, direction, distance, duration, fingerCount) {

$(".filter").removeClass("col-xs-12 col-sm-12 col-md-12 col-lg-12");

$(".filter").addClass("hidden-xs hidden-sm col-md-2 col-lg-2");

$(".live\_data").removeClass("hidden-xs hidden-sm hidden-md hidden-lg");

$(".live\_data").addClass("col-xs-12 col-sm-12 col-md-10 col-lg-10");

},

threshold: 1

});

});

}

function fetch\_data(sql, url) {

var favls = localStorage.getItem("favorite");

$.ajax({

url: url,

method: "POST",

data: {

ASQL: sql,

AFavoritLS: favls

},

success: function (data) {

$(".live\_data").html(data);

}

});

}

var arr = new Array();

function fetch\_data\_map(sql, url, center) {

var favls = localStorage.getItem("favorite");

$.ajax({

url: url,

method: "POST",

data: {

ASQL: sql,

AFavoritLS: favls

},

dataType: "json",

success: function (data) {

arr = data;

console.log(data);

}

});

}

function init2() {

var map = new ymaps.Map("map", {

center: [55.76, 37.64],

zoom: 16

})

arr.forEach(function (row, i) {

console.log(row);

var geocoder = ymaps.geocode(row["address"]);

geocoder.then(

function (res) {

var firstGeoObject = res.geoObjects.get(0),

coords = firstGeoObject.geometry.getCoordinates()

console.log(coords);

// firstGeoObject.options.set('preset', 'islands#darkBlueDotIconWithCaption');

// // Получаем строку с адресом и выводим в иконке геообъекта.

// firstGeoObject.properties.set('hintContent', row["name"]);

// map.geoObjects.add(firstGeoObject);

var placemark = new ymaps.Placemark([coords[0], coords[1]], {

hintContent: row["name"]

}, {

preset: "islands#dotCircleIcon",

iconColor: '#ff0000',

});

placemark.events.add('click', function () {

document.location.href = "./vacancy?id=" + row["idVacancy"];

});

map.geoObjects.add(placemark);

map.panTo(coords);

map.setCenter(coords);

},

function (err) {

console.log("Адрес не найден");

}

);

})

circle = new ymaps.Circle([[55.43, 37.7], 10000], null, {

draggable: false

});

circle.events.add('drag', function () {

// Объекты, попадающие в круг, будут становиться красными.

var objectsInsideCircle = objects.searchInside(circle);

objectsInsideCircle.setOptions('preset', 'twirl#redIcon');

// Оставшиеся объекты – синими.

objects.remove(objectsInsideCircle).setOptions('preset', 'twirl#blueIcon');

});

map.geoObjects.add(circle);

}

function getLS(ls) {

var favls = localStorage.getItem(ls);

return favls;

}

$(window).on('load', function () {

$('.spinner').delay(250).fadeOut('slow');

$('.load-div').delay(250).fadeOut('slow');

});

function AddItem(index, maxItem, NbutAdd, htmls, AddDiv) {

if (maxItem != 0 && index >= maxItem) {

$(NbutAdd).prop('disabled', true);

}

$(AddDiv).append(htmls);

}

function deleteItem(element, NbutAdd) {

$(element).parent().parent().remove();

if ($(NbutAdd + ':hidden')) {

$(NbutAdd).prop('disabled', false);

}

}

function deleteItem2(element, NbutAdd) {

$(element).parent().parent().parent().remove();

if ($(NbutAdd + ':hidden')) {

$(NbutAdd).prop('disabled', false);

}

}

//yandexmaps

var address;

function init() {

map = new ymaps.Map("map", {

center: [55.76, 37.64],

zoom: 16

});

var geocoder = ymaps.geocode(address);

geocoder.then(

function (res) {

map.geoObjects.add(res.geoObjects);

var array = res.geoObjects.getBounds();

map.panTo([array[0][0], array[0][1]], {

flying: true

})

},

function (err) {

console.log("Адрес не найден");

}

);

}

function yearsinit(name) {

var date = new Date();

for (i = date.getFullYear(); i > 1900; i--) {

$(name).append('<option value="' + i + '">' + i + '</option> ');

}

}

var date = new Date();

for (i = date.getFullYear(); i > 1900; i--) {

var years = years + '<option value="' + i + '">' + i + '</option> ';

}

var accdenname;

function accden(name) {

accdenname = name;

$("#accdenModal").modal("show");

return;

}

$(document).ready(function () {

$(document).on("click", "#accept\_", function () {

eval(accdenname);

accdenname = "";

$("#accdenModal").modal("hide");

})

$(document).on("click", "#denied\_", function () {

$("#accdenModal").modal("hide");

accdenname = "";

})

$.ajax({

url: "./includes/sessionCheck.php",

method: "POST",

success: function (data) {

if (data == "true") {

$("#webs").button('toggle')

} else {

$("#locs").button('toggle')

}

}

})

session\_check();

// favorite\_update();

$(document).on("click", "#add\_edu", function () {

index = $(".formedu").length;

var selectValues = [];

var htmls = '\

<div class="formedu">\

<div class="form-group">\

<label for="inputEducationLevel" class="col-sm-3 control-label">Уровень:</label>\

<div class="col-sm-8">\

<input type="text" class="form-control" id="inputEducationLevel" name="inputEducationLevel" placeholder="" data-id="">\

</div>\

</div>\

<div class="form-group">\

<label for="inputEducation" class="col-sm-3 control-label">Учебное заведение:</label>\

<div class="col-sm-7">\

<input type="text" class="form-control" id="inputEducation" name="inputEducation" placeholder="">\

</div>\

<div class="col-sm-1">\

<button type="button" class="btn btn-default" id="del\_edu"><span class="glyphicon glyphicon-remove"></span></button>\

</div>\

</div>\

<div class="form-group">\

<label for="inputEducationFak" class="col-sm-3 control-label">Факультет:</label>\

<div class="col-sm-8">\

<input type="text" class="form-control" id="inputEducationFak" name="inputEducationFak" placeholder="">\

</div>\

</div>\

<div class="form-group">\

<label for="inputEducationSpec" class="col-sm-3 control-label">Специализация:</label>\

<div class="col-sm-8">\

<input type="text" class="form-control" id="inputEducationSpec" name="inputEducationSpec" placeholder="">\

</div>\

</div>\

<div class="form-group">\

<label for="inputEducationYear" class="col-sm-3 control-label">Год окончания:</label>\

<div class="col-sm-3">\

<select class="form-control" id="inputEducationYear" name="inputEducationYear" title="Не выбрано">' + years + '\

</select>\

</div>\

</div>\

</div>\

';

AddItem(index, 2, this, htmls, '.edu');

});

$(document).on("click", "#add\_tel", function () {

index = $(".formnumber").length;

var htmls = '\

<div class="form-group formnumber" >\

<label for="inputphone" class="col-sm-3 control-label"></label>\

<div class="col-xs-9 col-sm-5 col-md-5">\

<input type="text" class="form-control inputphone" name="inputphone" id="inputphone" data-id="">\

</div>\

<div class="col-xs-1 col-sm-1 col-md-1">\

<button type="button" class="btn btn-default" id="del\_tel"><span class="glyphicon glyphicon-remove"></span></button>\

</div>\

</div>\

'

AddItem(index, 5, this, htmls, '.tel');

});

$(document).on("click", "#add\_req", function () {

index = $(".formnumber1").length;

var htmls = '\

<div class="form-group formnumber1">\

<label for="inputReq" class="col-sm-3 control-label"></label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputReq" id="inputReq" data-id=""> \

</div>\

<div class="col-xs-2 col-sm-2 col-md-2 col-lg-2">\

<button type="button" class="btn btn-default" id="del\_req"><span class="glyphicon glyphicon-remove"></span></button>\

</div>\

</div>\

';

AddItem(index, 10, this, htmls, '.req');

});

$(document).on("click", "#del\_req", function () {

var id = $(this).data('id');

if (id != null) {

$.ajax({

url: "./includes/vacancyReqDelete.php",

method: "POST",

data: {

AID: id

},

success: function (data) {

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

}

deleteItem(this, '#add\_req');

});

$(document).on("click", "#add\_ski", function () {

index = $(".formnumber2").length;

var htmls = '\

<div class="form-group formnumber2">\

<label for="inputSki" class="col-sm-3 control-label"></label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputSki" id="inputSki" data-id="">\

</div>\

<div class="col-xs-2 col-sm-2 col-md-2 col-lg-2">\

<button type="button" class="btn btn-default" id="del\_ski"><span class="glyphicon glyphicon-remove"></span></button>\

</div>\

</div>\

';

AddItem(index, 10, this, htmls, '.ski');

});

$(document).on("click", "#del\_ski", function () {

var id = $(this).data('id');

if (id != null) {

$.ajax({

url: "./includes/vacancySkiDelete.php",

method: "POST",

data: {

AID: id

},

success: function (data) {

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

}

deleteItem(this, '#add\_ski');

});

$(document).on("click", "#add\_dut", function () {

index = $(".formnumber3").length;

var htmls = '\

<div class="form-group formnumber3">\

<label for="inputDut" class="col-sm-3 control-label"></label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputDut" id="inputDut" data-id="">\

</div>\

<div class="col-xs-2 col-sm-2 col-md-2 col-lg-2">\

<button type="button" class="btn btn-default" id="del\_dut"><span class="glyphicon glyphicon-remove"></span></button>\

</div>\

</div>\

';

AddItem(index, 10, this, htmls, '.dut');

});

$(document).on("click", "#del\_dut", function () {

var id = $(this).data('id');

if (id != null) {

$.ajax({

url: "./includes/vacancyDutDelete.php",

method: "POST",

data: {

AID: id

},

success: function (data) {

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

}

deleteItem(this, '#add\_dut');

});

$(document).on("click", "#add\_telVac", function () {

index = $(".formnumber4").length;

var htmls = '<div class="formnumber4">\

<div class="form-group">\

<label for="inputFamVac" class="col-sm-3 control-label">Фамилия:</label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputFamVac" id="inputFamVac" data-id="">\

</div>\

<div class="col-xs-2 col-sm-2 col-md-2 col-lg-2">\

<button type="button" class="btn btn-default" id="deltelVac"><span class="glyphicon glyphicon-remove"></span></button>\

</div>\

</div>\

<div class="form-group ">\

<label for="inputNameTelVac" class="col-sm-3 control-label">Имя:</label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputNameTelVac" id="inputNameTelVac">\

</div>\

</div>\

<div class="form-group ">\

<label for="inputOtVac" class="col-sm-3 control-label">Отчество:</label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputOtVac" id="inputOtVac">\

</div>\

</div>\

<div class="form-group ">\

<label for="inputEmailVac" class="col-sm-3 control-label">Электронная почта:</label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputEmailVac" id="inputEmailVac">\

</div>\

</div>\

<div class="form-group ">\

<label for="inputTelVac" class="col-sm-3 control-label">Телефон:</label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputTelVac" id="inputTelVac">\

</div>\

</div>\

</div>\

<br>\

';

AddItem(index, 10, this, htmls, '.vacTel');

});

$(document).on("click", "#deltelVac", function () {

var id = $(this).data('id');

if (id != null) {

$.ajax({

url: "./includes/vacancyTelDelete.php",

method: "POST",

data: {

AID: id

},

success: function (data) {

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

}

deleteItem2(this, '#add\_telVac');

});

var vacid;

$(document).on("click", "#changeVacancy", function () {

var id = $(this).data('id');

vacid = id;

$.ajax({

url: "./includes/vacancyviewacc.php",

method: "POST",

dataType: "text",

data: {

AID: id

},

success: function (data) {

var arr = JSON.parse(data);

console.log(arr);

if (arr[0].length != 0) {

$("#inputNameVac").val(arr[0][0]['name']);

$("#inputGrafic").val(arr[0][0]['schedule']);

$("#inputMoney").val(arr[0][0]['salary']);

$('#inputExpVac [value="' + arr[0][0]['experience'] + '"]').attr("selected", "selected");

$("#inputExpVac").selectpicker('val', arr[0][0]['experience']);

$("#inputAddress").val(arr[0][0]['address']);

$("#inputFamVac").val(arr[4][0]['surname']);

$("#inputFamVac").attr("data-id", arr[4][0]['idContact\_information']);

$("#inputNameTelVac").val(arr[4][0]['name']);

$("#inputOtVac").val(arr[4][0]['middle\_name']);

$("#inputEmailVac").val(arr[4][0]['email']);

$("#inputTelVac").val(arr[4][0]['telephone']);

// $("#inputReq").val(arr[1][0]['Requirement']);

// $("#inputSki").val(arr[2][0]['Skill']);

// $("#inputDut").val(arr[3][0]['Dutie']);

arrlen = arr[1].length;

arr[1].forEach(function (item, i, arr) {

index = $(".formnumber1").length;

if (i >= 0 && index < arrlen) {

var htmls = '\

<div class="form-group formnumber1">\

<label for="inputReq" class="col-sm-3 control-label"></label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputReq" id="inputReq" data-id="' + arr[i]['idRequirements'] + '" value="' + arr[i]['Requirement'] + '"> \

</div>\

<div class="col-xs-2 col-sm-2 col-md-2 col-lg-2">\

<button type="button" class="btn btn-default" id="del\_req" data-id="' + arr[i]['idRequirements'] + '"><span class="glyphicon glyphicon-remove" ></span></button>\

</div>\

</div>\

';

AddItem(index, 10, this, htmls, '.reqNN');

}

});

arrlen = arr[2].length;

arr[2].forEach(function (item, i, arr) {

index = $(".formnumber2").length;

if (i >= 0 && index < arrlen) {

var htmls = '\

<div class="form-group formnumber2">\

<label for="inputSki" class="col-sm-3 control-label"></label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputSki" id="inputSki" data-id="' + arr[i]['idSkill'] + '" value="' + arr[i]['Skill'] + '">\

</div>\

<div class="col-xs-2 col-sm-2 col-md-2 col-lg-2">\

<button type="button" class="btn btn-default" id="del\_ski" data-id="' + arr[i]['idSkill'] + '"><span class="glyphicon glyphicon-remove"></span></button>\

</div>\

</div>\

';

AddItem(index, 10, this, htmls, '.skiNN');

}

});

arrlen = arr[3].length;

arr[3].forEach(function (item, i, arr) {

index = $(".formnumber3").length;

if (i >= 0 && index < arrlen) {

var htmls = '\

<div class="form-group formnumber3">\

<label for="inputDut" class="col-sm-3 control-label"></label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputDut" id="inputDut" data-id="' + arr[i]['idDutie'] + '" value="' + arr[i]['Dutie'] + '">\

</div>\

<div class="col-xs-2 col-sm-2 col-md-2 col-lg-2">\

<button type="button" class="btn btn-default" id="del\_dut" data-id="' + arr[i]['idDutie'] + '"><span class="glyphicon glyphicon-remove"></span></button>\

</div>\

</div>\

';

AddItem(index, 10, this, htmls, '.dutNN');

}

});

arrlen = arr[4].length;

arr[4].forEach(function (item, i, arr) {

index = $(".formnumber4").length;

if (i >= 1 && index < arrlen) {

var htmls = '<div class="formnumber4">\

<div class="form-group">\

<label for="inputFamVac" class="col-sm-3 control-label">Фамилия:</label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputFamVac" id="inputFamVac" data-id="' + arr[i]['idContact\_information'] + '" value="' + arr[i]['surname'] + '">\

</div>\

<div class="col-xs-2 col-sm-2 col-md-2 col-lg-2">\

<button type="button" class="btn btn-default" id="deltelVac"><span class="glyphicon glyphicon-remove"></span></button>\

</div>\

</div>\

<div class="form-group ">\

<label for="inputNameTelVac" class="col-sm-3 control-label">Имя:</label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputNameTelVac" id="inputNameTelVac" value="' + arr[i]['name'] + '">\

</div>\

</div>\

<div class="form-group ">\

<label for="inputOtVac" class="col-sm-3 control-label">Отчество:</label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputOtVac" id="inputOtVac" value="' + arr[i]['middle\_name'] + '">\

</div>\

</div>\

<div class="form-group ">\

<label for="inputEmailVac" class="col-sm-3 control-label">Электронная почта:</label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputEmailVac" id="inputEmailVac" value="' + arr[i]['email'] + '">\

</div>\

</div>\

<div class="form-group ">\

<label for="inputTelVac" class="col-sm-3 control-label">Телефон:</label>\

<div class="col-sm-7 col-lg-7">\

<input type="text" class="form-control inputphone" name="inputTelVac" id="inputTelVac" value="' + arr[i]['telephone'] + '">\

</div>\

</div>\

</div>\

<br>\

';

AddItem(index, 10, this, htmls, '.vacTelNN');

}

});

$("#MyVacancyModal").modal('show');

}

},

error: function () {

alert('Ошибка AJAX запроса');

}

})

});

$(document).on("click", "#BtnMyVacEnterNull", function () {

$(this).button('loading');

var jsonarr = $('#VacancyFormNull').serializeArray();

var dataid = getAllDataForm(jsonarr, 'input');

console.log(jsonarr);

$.ajax({

url: "/includes/vacancyinsert.php",

method: "POST",

dataType: "text",

data: {

AJSON: jsonarr,

ADATA: dataid

},

success: function (data) {

location.reload();

console.log(data);

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

$(this).button('reset');

});

$(document).on("click", "#BtnMyVacEnter", function () {

$(this).button('loading');

var jsonarr = $('#VacancyForm').serializeArray();

var dataid = getAllDataForm(jsonarr, '#VacancyForm input');

console.log(jsonarr);

$.ajax({

url: "./includes/vacancyupdate.php",

method: "POST",

dataType: "text",

data: {

AJSON: jsonarr,

ADATA: dataid,

AVACID: vacid

},

success: function (data) {

location.reload();

console.log(data);

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

$(this).button('reset');

});

//--------------------------------------------------------------

// Поиск

$(document).on("input", "#search-input", function () {

fetch\_data("SELECT \* FROM `" + $('#search-input').data('nametable') + "` WHERE `" + $('#search-input').data('rowname') + "` LIKE '%" + $('#search-input').val() + "%'", $('#search-input').data('url'));

});

$(document).on("click", "#del\_tel", function () {

var id = $(this).data('id');

if (id != null) {

$.ajax({

url: "./includes/summaryTelDelete.php",

method: "POST",

data: {

AID: id

},

success: function (data) {

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

}

deleteItem(this, '#add\_tel');

});

$(document).on("click", "#del\_edu", function () {

var id = $(this).data('id');

if (id != null) {

$.ajax({

url: "./includes/summaryEduDelete.php",

method: "POST",

data: {

AID: id

},

success: function (data) {

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

}

deleteItem2(this, '#add\_edu');

});

function favoritealladdls() {

strarray = localStorage.getItem("favorite");

$.ajax({

url: "./includes/sessionCheck.php",

method: "POST",

success: function (data) {

if (data == "true") {

$.ajax({

url: "./includes/addalllsFavorite.php",

method: "POST",

data: {

AARR: strarray

},

success: function (data) {

console.log(data);

localStorage.setItem("favorite", '[]');

}

});

}

}

});

}

function favoriteaddls(id, type) {

var arr = [];

var strarray;

var check = false;

strarray = localStorage.getItem("favorite");

arr = JSON.parse(strarray);

if (arr == null) {

arr = [{

id: id,

type: type

}];

check = false;

console.log(check);

} else {

arr.forEach(function (row, i) {

if (id == row["id"] && type == row["type"]) {

arr.splice(i, 1);

check = true;

strarray = JSON.stringify(arr);

localStorage.setItem("favorite", strarray);

return check;

} else {

check = false;

}

});

if (check == true) return check;

arr.push({

id: id,

type: type

});

}

strarray = JSON.stringify(arr);

localStorage.setItem("favorite", strarray);

console.log(check);

// console.log(arr);

// console.log(strarray);

return check;

}

function favoritedells(id, type) {

var arr = [];

var strarray;

strarray = localStorage.getItem("favorite");

arr = JSON.parse(strarray);

arr.forEach(function (row, i) {

if (id == row["id"] && type == row["type"]) {

arr.splice(i, 1);

}

})

strarray = JSON.stringify(arr);

localStorage.setItem("favorite", strarray);

}

$(document).on("click", ".addallfav", function () {

favoritealladdls();

});

$(document).on("click", "#favorite", function () {

var id = $(this).data('id');

var type = $(this).data('type');

$.ajax({

url: "./includes/sessionCheck.php",

method: "POST",

success: function (data) {

if (data == "true") {

$.ajax({

url: "./includes/adddelFavorite.php",

method: "POST",

data: {

AID: id,

ATYPE: type

},

success: function (data) {

if (data == "true") {

$("#favorite[data-id='" + id + "'][data-type='" + type + "']").addClass("green");

$("#favorite[data-id='" + id + "'][data-type='" + type + "']").html('<span class="glyphicon glyphicon-bookmark"></span> Избранное');

favoritedells(id, type);

} else if (data == "false") {

$("#favorite[data-id='" + id + "'][data-type='" + type + "']").removeClass("green");

$("#favorite[data-id='" + id + "'][data-type='" + type + "']").html('<span class="glyphicon glyphicon-bookmark"></span> В избранное');

favoritedells(id, type);

}

}

});

} else if (data == "false") {

var g = favoriteaddls(id, type);

if (g == false) {

$("#favorite[data-id='" + id + "'][data-type='" + type + "']").addClass("green");

$("#favorite[data-id='" + id + "'][data-type='" + type + "']").html('<span class="glyphicon glyphicon-hdd" ></span> <span class="glyphicon glyphicon-bookmark"></span> Избранное');

} else if (g == true) {

$("#favorite[data-id='" + id + "'][data-type='" + type + "']").removeClass("green");

$("#favorite[data-id='" + id + "'][data-type='" + type + "']").html(' <span class="glyphicon glyphicon-bookmark"></span> В избранное');

}

}

}

});

});

$(document).on("click", "#smfavorite", function () {

var id = $(this).data('id');

var type = $(this).data('type');

$.ajax({

url: "./includes/sessionCheck.php",

method: "POST",

success: function (data) {

if (data == "true") {

$.ajax({

url: "./includes/adddelFavorite.php",

method: "POST",

data: {

AID: id,

ATYPE: type

},

success: function (data) {

if (data == "true") {

$("#smfavorite[data-id='" + id + "'][data-type='" + type + "']").addClass("green");

} else if (data == "false") {

$("#smfavorite[data-id='" + id + "'][data-type='" + type + "']").removeClass("green");

}

}

});

} else if (data == "false") {

var g = favoriteaddls(id, type);

if (g == false) {

$("#smfavorite[data-id='" + id + "'][data-type='" + type + "']").addClass("green");

$("#smfavorite[data-id='" + id + "'][data-type='" + type + "']").html('<span class="glyphicon glyphicon-hdd" ></span> <span class="glyphicon glyphicon-bookmark"></span>');

} else if (g == true) {

$("#smfavorite[data-id='" + id + "'][data-type='" + type + "']").removeClass("green");

$("#smfavorite[data-id='" + id + "'][data-type='" + type + "']").html('</span> <span class="glyphicon glyphicon-bookmark"></span>');

}

}

}

});

});

//маска ввода

$(document).on("change", "#country", function () {

});

$(".ui-loader").hide();

$(document).on("click", "#BtnSignin", function () {

$("#LoginModal").modal("show");

});

$(document).on("click", "#SessionDes", function () {

$.ajax({

url: "./includes/session\_exit.php",

method: "POST",

success: function (data) {

session\_check();

location.reload(); //\*\*\*

}

});

});

$(document).on("click", "#BtnRegister", function () {

$("#RegisterModal").modal("show");

});

$(document).on("click", "#BtnEnter", function () {

var login = $("#InputLogin").val(),

password = $("#InputPassword").val(),

btn = $(this);

if (login == "") {

$("#DivLogin").addClass("has-warning");

$("#ImgLog").addClass("glyphicon-warning-sign");

return;

}

if (password == "") {

$("#DivPassword").addClass("has-warning");

$("#ImgPas").addClass("glyphicon-warning-sign");

return;

}

btn.button('loading');

$.ajax({

url: "./includes/signin.php",

method: "POST",

data: {

ALOGIN: login,

APASSWORD: password

},

dataType: "TEXT",

success: function (data) {

if (data == "Bad password") {

$("#DivPassword").addClass("has-error");

$("#ImgPas").addClass("glyphicon-remove");

$("#alertLogPas").removeClass("hidden");

} else if (data == "Bad login") {

$("#DivLogin").addClass("has-error");

$("#ImgLog").addClass("glyphicon-remove");

$("#alertLogPas").removeClass("hidden");

} else if (data == "Success") {

$("#InputPassword").text = "";

$("#LoginModal").modal("hide");

session\_check();

location.reload();

} else {

alert(data.replace(/<\/?[^>]+>/g, ''));

}

btn.button('reset');

},

error: function () {

alert('Ошибка AJAX запроса')

}

});

});

$(document).on("input", "#InputLogin", function () {

$("#DivLogin").removeClass("has-error");

$("#ImgLog").removeClass("glyphicon-remove");

$("#DivLogin").removeClass("has-warning");

$("#ImgLog").removeClass("glyphicon-warning-sign");

$("#alertLogPas").addClass("hidden");

});

$(document).on("input", "#InputPassword", function () {

$("#DivPassword").removeClass("has-error");

$("#ImgPas").removeClass("glyphicon-remove");

$("#DivPassword").removeClass("has-warning");

$("#ImgPas").removeClass("glyphicon-warning-sign");

$("#alertLogPas").addClass("hidden");

});

$(document).on("click", "#changepasemail", function () {

// $("#LoginModal").modal("hide");

$("#PasChangeEmailModal").modal("show");

});

// $(document).on("click", "#DivChangePasEmailClose", function () {

// $("#LoginModal").modal("show");

// });

$(document).on("click", "#BtnPasEmailChange", function () {

var btn = $(this);

var email = $("#InputChangePasEmail").val();

if (email == "") {

$("#DivChangePasEmail").addClass("has-warning");

$("#ImgChangePasEmail").addClass("glyphicon-warning-sign");

return;

}

btn.button('loading');

$.ajax({

url: "./includes/sendmail.php",

method: "POST",

data: {

AEMAIL: email

},

dataType: "TEXT",

success: function (data) {

if (data == "Bad email") {

$("#DivChangePasEmail").addClass("has-error");

$("#ImgChangePasEmail").addClass("glyphicon-remove");

} else if (data == "Success") {

alert("Сообщение с активацией отправлено на почту.")

$("#PasChangeEmailModal").modal("hide");

}

btn.button('reset');

// console.log(data);

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

});

$(document).on("click", "#BtnRegEnter", function () {

var btn = $(this);

var email = $("#InputRegEmail").val(),

login = $("#InputRegLogin").val(),

pas1 = $("#InputRegPassword").val(),

pas2 = $("#InputRegPassword2").val();

if (login == "") {

$("#DivRegister").addClass("has-warning");

$("#ImgRegLog").addClass("glyphicon-warning-sign");

return;

}

if (email == "") {

$("#DivRegEmail").addClass("has-warning");

$("#ImgRegEmail").addClass("glyphicon-warning-sign");

return;

}

if (pas1 == "") {

$("#DivRegPassword").addClass("has-warning");

$("#ImgRegPas1").addClass("glyphicon-warning-sign");

return;

}

if (pas2 == "") {

$("#DivRegPassword2").addClass("has-warning");

$("#ImgRegPas2").addClass("glyphicon-warning-sign");

return;

}

if (pas1 != pas2) {

$("#alertRegPas").removeClass("hidden");

return;

}

btn.button('loading');

$.ajax({

url: "./includes/register.php",

method: "POST",

data: {

ALOGIN: login,

APASSWORD: pas1,

AEMAIL: email

},

dataType: "TEXT",

success: function (data) {

if (data == "Bad login") {

$("#DivRegister").addClass("has-error");

$("#ImgRegLog").addClass("glyphicon-remove");

} else if (data == "Bad email") {

$("#DivRegEmail").addClass("has-error");

$("#ImgRegEmail").addClass("glyphicon-remove");

} else if (data == "Ошибка") {

alert("Ошибка: Данные не заполнены в БД");

} else if (data == "Success") {

$("#RegisterModal").modal("hide");

}

btn.button('reset');

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

});

$(document).on("input", "#InputRegEmail", function () {

$("#DivRegEmail").removeClass("has-warning");

$("#ImgRegEmail").removeClass("glyphicon-warning-sign");

$("#DivRegEmail").removeClass("has-error");

$("#ImgRegEmail").removeClass("glyphicon-remove");

});

$(document).on("input", "#InputRegLogin", function () {

$("#DivRegister").removeClass("has-warning");

$("#ImgRegLog").removeClass("glyphicon-warning-sign");

$("#DivRegister").removeClass("has-error");

$("#ImgRegLog").removeClass("glyphicon-remove");

});

$(document).on("input", "#InputRegPassword", function () {

$("#DivRegPassword").removeClass("has-warning");

$("#ImgRegPas").removeClass("glyphicon-warning-sign");

$("#alertRegPas").addClass("hidden");

});

$(document).on("input", "#InputRegPassword2", function () {

$("#DivRegPassword2").removeClass("has-warning");

$("#ImgRegPas2").removeClass("glyphicon-warning-sign");

$("#alertRegPas").addClass("hidden");

});

$(document).on("click", "#VacancyAdd", function () {

$("#MyVacancyModalNull").modal("show");

});

$(document).on("click", "#MySummary", function () {

$.ajax({

url: "./includes/summaryviewacc.php",

method: "POST",

dataType: "text",

success: function (data) {

var arr = JSON.parse(data);

console.log(arr);

if (arr[0].length != 0) {

$("#photoSummary").attr("src", './summaryPhoto/'+arr[0][0]['Accounts\_idAccount']+'.png');

$("#inputSurname").val(arr[0][0]['surname']);

$("#inputName").val(arr[0][0]['name']);

$("#inputMidName").val(arr[0][0]['middle\_name']);

$('#inputgender').selectpicker('val', arr[0][0]['gender']);

$("#inputDateB").val(arr[0][0]['date\_of\_birth']);

$("#inputcitizenship").val(arr[0][0]['nationality']);

$("#inputciti").val(arr[0][0]['city\_of\_residence']);

$("#inputprofarea").val(arr[0][0]['professional\_area']);

$("#inputwork\_exp").selectpicker('val', arr[0][0]['work\_experience']);

//$("#inputwork\_exp option[value=" + arr[0][0]['work\_experience'] + "]").attr("selected", "selected");

$("#inputphone").val(arr[1][0]['number']);

$("#inputphone").attr("data-id", arr[1][0]['idTelephone']);

$("#inputEducationLevel").val(arr[2][0]['level']);

$("#inputEducation").val(arr[2][0]['name']);

$("#inputEducationFak").val(arr[2][0]['faculty']);

$("#inputEducationSpec").val(arr[2][0]['specialization']);

$("#inputEducationYear").val(arr[2][0]['year\_of\_ending']);

$("#inputEducationLevel").attr("data-id", arr[2][0]['idEducationSummary']);

arrlen = arr[1].length;

arr[1].forEach(function (item, i, arr) {

index = $(".formnumber").length;

if (i >= 1 && index < arrlen) {

var htmls = '\

<div class="form-group formnumber" >\

<label for="inputphone" class="col-sm-3 control-label"></label>\

<div class="col-xs-9 col-sm-5 col-md-5">\

<input type="text" class="form-control inputphone" name="inputphone" id="inputphone" value="' + arr[i]['number'] + '" data-id="' + arr[i]['idTelephone'] + '">\

</div>\

<div class="hidden-xs hidden-sm">\

\

</div>\

<div class="col-xs-1 col-sm-1 col-md-1">\

<button type="button" class="btn btn-default" id="del\_tel" data-id="' + arr[i]['idTelephone'] + '"><span class="glyphicon glyphicon-remove"></span></button>\

</div>\

</div>';

AddItem(index, 5, '#add\_tel', htmls, '.tel');

}

});

arrlen = arr[2].length;

arr[2].forEach(function (item, i, arr) {

index = $(".formedu").length;

if (i >= 1 && index < arrlen) {

var htmls = '\

<div class="formedu" >\

<div class="form-group">\

<label for="inputEducationLevel" class="col-sm-3 control-label">Уровень:</label>\

<div class="col-sm-8">\

<input type="text" class="form-control" id="inputEducationLevel" name="inputEducationLevel" placeholder="" value="' + arr[i]['level'] + '" data-id="' + arr[i]['idEducationSummary'] + '">\

</div>\

</div>\

<div class="form-group">\

<label for="inputEducation" class="col-sm-3 control-label">Учебное заведение:</label>\

<div class="col-sm-7">\

<input type="text" class="form-control" id="inputEducation" name="inputEducation" placeholder="" value="' + arr[i]['name'] + '" >\

</div>\

<div class="col-sm-1">\

<button type="button" class="btn btn-default" id="del\_edu" data-id="' + arr[i]['idEducationSummary'] + '"><span class="glyphicon glyphicon-remove"></span></button>\

</div>\

</div>\

<div class="form-group">\

<label for="inputEducationFak" class="col-sm-3 control-label">Факультет:</label>\

<div class="col-sm-8">\

<input type="text" class="form-control" id="inputEducationFak" name="inputEducationFak" placeholder="" value="' + arr[i]['faculty'] + '" >\

</div>\

</div>\

<div class="form-group">\

<label for="inputEducationSpec" class="col-sm-3 control-label">Специализация:</label>\

<div class="col-sm-8">\

<input type="text" class="form-control" id="inputEducationSpec" name="inputEducationSpec" placeholder="" value="' + arr[i]['specialization'] + '" >\

</div>\

</div>\

<div class="form-group">\

<label for="inputEducationYear" class="col-sm-3 control-label">Год окончания:</label>\

<div class="col-sm-3">\

<select class="form-control" id="inputEducationYear" name="inputEducationYear" >' + years + '\

</select>\

</div>\

</div>\

</div>\

';

AddItem(index, 2, '#add\_edu', htmls, '.edu');

$("#inputEducationYear option[value=" + arr[i]['year\_of\_ending'] + "]").attr("selected", "selected");

}

});

}

yearsinit('.opYears');

$("#MySummaryModal").modal("show");

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

});

function mes\_up() {

$.ajax({

url: "./includes/messageView.php",

method: "POST",

success: function (data) {

$(".chat-mes").html(data);

$("#Chat").modal("show");

}

});

}

function mes\_send() {

var mes = $("#mes-inp").val();

var rec = $("#rec-inp").val();

$.ajax({

url: "./includes/messageSend.php",

method: "POST",

data: {

AMES: mes,

AREC: rec,

},

success: function (data) {

mes\_up();

console.log(data);

}

});

}

$(document).on("click", "#btn-sendmes", function () {

btn = $(this);

btn.button('loading');

mes\_send();

btn.button('reset');

});

$(document).on("click", "#mes-update", function () {

mes\_up();

});

// $(document).on("click", "#mes-del", function () {

// $.ajax({

// url: "./includes/messageAllDel.php",

// method: "POST",

// success: function (data) {

// mes\_up();

// }

//

// })

// });

$(document).on("click", "#mesbut", function () {

$("#rec-inp").val($(this).data("id"));

mes\_up();

});

$(document).on("click", "#MyMes", function () {

mes\_up();

});

// ----

function getAllDataForm(array, name) {

var map = $(name).map(function () {

return this.dataset.id;

}).get();

console.log(map);

return map;

}

$(document).on("click", "#BtnMySumEnter", function () {

$(this).button('loading');

$.ajax({

url: "./includes/checkvalsummary.php",

method: "POST",

success: function (data) {

var jsonarr = $('#SummaryForm').serializeArray();

var dataid = getAllDataForm(jsonarr, 'input');

console.log(jsonarr);

if (data == "true") {

//Update

$.ajax({

url: "/includes/summaryupdate.php",

method: "POST",

dataType: "text",

data: {

AJSON: jsonarr,

ADATA: dataid

},

success: function (data) {

console.log(data);

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

} else if (data == "false") {

$.ajax({

url: "./includes/summaryinsert.php",

method: "POST",

dataType: "text",

data: {

AJSON: jsonarr,

ADATA: dataid

},

success: function (data) {

console.log(data);

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

} else {

alert(data);

}

}

});

$(this).button('reset');

});

$(document).on("click", "#BtnMyComEnter", function () {

$(this).button('loading');

$.ajax({

url: "./includes/checkvalcompany.php",

method: "POST",

success: function (data) {

var jsonarr = $('#SummaryForm').serializeArray();

console.log(jsonarr);

if (data == "true") {

//Update

$.ajax({

url: "/includes/companyupdate.php",

method: "POST",

dataType: "text",

data: {

AJSON: jsonarr

},

success: function (data) {

console.log(data);

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

} else if (data == "false") {

//Insert

$.ajax({

url: "./includes/companyinsert.php",

method: "POST",

dataType: "text",

data: {

AJSON: jsonarr,

},

success: function (data) {

console.log(data);

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

} else {

alert(data);

}

}

});

$(this).button('reset');

});

// $("#MyCompanyModal").modal("show");

$(document).on("click", "#MyCompany", function () {

$("#MyCompanyModal").modal("show");

});

$('#uploadCompanyLogo').change(function () {

var file = this.files[0];

var fdata = new FormData;

fdata.append("img", file);

console.log(file);

console.log(fdata.getAll("img"));

$.ajax({

url: "./includes/addcompanylogo.php",

method: "POST",

processData: false,

contentType: false,

data: fdata,

success: function (data) {

alert(data);

$("#logoCompany").attr("src", data);

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

});

function deletefunc(url, modalname) {

console.log('del');

// $.ajax({

// url: url,

// method: "POST",

// success: function (data) {

// $(modalname).modal("hide");

// },

// error: function () {

// alert('Ошибка AJAX запроса');

// }

// });

}

$(document).on("click", ".delete-summary", function () {

accden('deletefunc("./includes/summarydelete.php","#MySummaryModal")');

});

setFilter();

function setFilter() {

if (window.location != "http://sejob.atservers.net/") {

$.ajax({

url: "./includes/getFilter.php",

method: "POST",

success: function (data) {

var strarray = data;

var go = false;

var BreakException = {};

// console.log(data);

if (strarray != "") {

var array = JSON.parse(strarray);

try {

array.forEach(function (row) {

if (row['value'] == "") {

go = false;

console.log(go);

} else {

go = true;

console.log(go);

throw BreakException;

}

})

} catch (e) {

if (e !== BreakException) throw e;

}

array.forEach(function (row) {

if (row['value'] != "") {

$("#" + row['name']).val(row['value']);

$('#' + row['name'] + ' [value="' + row['value'] + '"]').attr("selected", "selected");

$("#" + row['name']).selectpicker('val', row['value']);

}

})

if (go == true) {

filterInit();

}

}

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

}

}

function filterInit() {

var url = $("#filter").data('url');

var type = $("#filter").data('typefilter');

var page = window.location.search.replace('?page=', '');

var jsonarr = $("#filter").serializeArray();

strarray = JSON.stringify(jsonarr);

$.ajax({

url: "./includes/filterInit.php",

method: "POST",

data: {

AJSON: jsonarr,

ATYPE: type,

APAGE: page,

AFILTER: strarray

},

success: function (data) {

console.log(data);

fetch\_data(data, url);

},

error: function () {

alert('Ошибка AJAX запроса');

}

});

}

//filter

$(document).on("change", "#filter", function () {

filterInit();

});

$(document).on("dblclick", ".data", function () {

document.location.href = $(this).children(".row").children(".data-name").children("a").attr("href");

})

});

<?php

class User

{

public function SignIn($login, $password)

{

try {

require ("db\_conn.php");

$sql = "SELECT `password`, `name` FROM `Accounts` WHERE `name` LIKE '".$login."'";

foreach ($dbh->query($sql) as $result);

if ($result["name"] == $login)

{

if (password\_verify($password, $result["password"])==true) {

$\_SESSION['SLOGIN']=$login;

return "Success";

}

else {

return "Bad password";

}

}

else

{

return "Bad login";

}

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function Register($login, $password, $email)

{

if ($email==NULL)

{

return "Ошибка";

}

if ($password==NULL)

{

return "Ошибка";

}

if ($email==NULL)

{

return "Ошибка";

}

try {

require ("db\_conn.php");

$sql = "SELECT \* FROM `Accounts` WHERE `name` LIKE '".$login."'";

foreach ($dbh->query($sql) as $result);

if ($result[0]!=NULL)

{

return "Bad login";

}

$sql = "SELECT \* FROM `Accounts` WHERE `EMail` LIKE '".$email."'";

foreach ($dbh->query($sql) as $result);

if ($result[0]!=NULL)

{

return "Bad email";

}

$hashpas = password\_hash($password, PASSWORD\_DEFAULT);

$sql = "INSERT INTO `Accounts` (`name`, `email`, `password`) values('".$login."','".$email."','".$hashpas."')";

$dbh->query($sql);

return "Success";

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function MessageSend($login, $message, $recipient)

{

try {

require ("db\_conn.php");

$time = date("H:i:s");

$today = date("Y-m-d");

$sql = "INSERT INTO `Messages` (sender, recipient, message, date, time) values('".$login."','".$recipient."','".$message."','".$today."','".$time."')";

$dbh->query($sql);

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function СalculateAge($birthday) {

$birthday\_timestamp = strtotime($birthday);

$age = date('Y') – date('Y', $birthday\_timestamp);

if (date('md', $birthday\_timestamp) > date('md')) {

$age--;

}

return $age;

}

}

class Content

{

// public function pageGen($url,$page)

// {

// $start = $\_GET['page']-$limit;

// $end = $\_GET['page']+$limit;

// if ($end==$\_GET['page'])

// {

// $start=$end;

// }

// if ($pagearr[0]>0)

// {

// echo '<a class="btn btn-default" href="'.$url.'?page='.($\_GET['page']-1).'">Назад</a>';

// }

// for($i=1;$i<=$pagearr[2];$i++) {

// if ($i>=$start && $i<=$end) {

// if ($i == $\_GET['page']) {

// echo '<a class="btn btn-default active" href="/'.$url.'"?page='.$i.'" >'.$i.'</a>';

// } else {

// echo '<a class="btn btn-default" href="/'.$url.'page='.$i.'">'.$i.'</a>';

// }

// }

// }

// if ($pagearr[2]<$\_GET['page'])

// {

// echo '<a class="btn btn-default" href="'.$url.'?page='.($\_GET['page']+1).'">Вперед</a>';

// }

// }

//

public function pageSep($page,$name)

{

$obg\_in\_page=2;

$limit=5;

$page=$page-1;

$start=abs($page\*$obg\_in\_page);

$sql = "SELECT count(\*) FROM `".$name."`";

$result = Content::ViewAll($sql);

$num\_pages=ceil($result[0][0]/$obg\_in\_page);

return $arr = [$start,$obg\_in\_page,$num\_pages,$limit-1];

}

public function filter($array,$tablename,$page)

{

if ($array[0]['value']!=NULL)

{

$array[0]['value']="LIKE '%".$array[0]['value']."%'";

}

$str = "SELECT \* FROM `".$tablename."` WHERE ";

foreach($array as $row)

{

if ($row['value']!=NULL)

{

$str = $str."`".$row['name']."` ".$row['value']." AND ";

}

}

$pos = strripos($str," AND ");

$str = substr($str,0, $pos);

$arr=Content::pageSep($page,$tablename);// <----------------

$str= $str." LIMIT ".$arr[0].",".$arr[1];

// $pagearr = Content::pageSep($\_GET['page'],$tablename);

// $str = $str." LIMIT ".$pagearr[0].','.$pagearr[1];

return $str;

}

public function ImageAdd($uploaddir,$name)

{

$uploadfile = $uploaddir . $name;

if (move\_uploaded\_file($\_FILES['img']['tmp\_name'], $uploadfile)) {

return $uploadfile;

} else

{

return "Возможная атака с помощью файловой загрузки!";

}

}

public function IUD($sql){ // Insert Update Delete

require("db\_conn.php");

try {

$dbh->query($sql);

return $sql;

} catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function ViewAll($sql) {

require ("db\_conn.php");

try {

$sth = $dbh->prepare($sql);

$sth->execute();

$allarr=$sth->fetchAll();

return $allarr;

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function CheckVal($login, $colname, $tablename){

require("db\_conn.php");

try {

$id = "(SELECT `idAccount` FROM `Accounts` WHERE `name` LIKE '".$login."')";

$sql = "SELECT `".$colname."` FROM `".$tablename."` WHERE `".$colname."` LIKE ".$id."";

foreach ($dbh->query($sql) as $row);

if ($row[0] != NULL) {

return "true";

} else {

return "false";

}

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

public function CheckFavorite($idf,$login,$type)

{

require("db\_conn.php");

try {

$id = "(SELECT `idAccount` FROM `Accounts` WHERE `name` LIKE '".$login."')";

$sql = "SELECT \* FROM `Favorites` WHERE `Accounts\_idAccount` LIKE ".$id." AND `idFavorite` LIKE '".$idf."' AND `type` LIKE '".$type."'";

foreach ($dbh->query($sql) as $row);

if ($row[0] == NULL) {

return "true";

} else {

return "false";

}

}

catch (PDOException $e) {

return "Ошибка PDO: " . $e->getMessage() . "<br>";

}

}

}

?>