МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

**КУРСОВОЙ ПРОЕКТ**

по дисциплине: Технология разработки программного обеспечения

на тему «Web-приложение ведения складского учета»

Выполнил ст. гр. ИВТ-18

Забаровский Р.В

Проверил доцент кафедры ИВТ и ПМ,

к.т.н., доцент Валова О.В

Чита

2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ЗАДАНИЕ

на курсовой проект

По дисциплине: Технология разработки программного обеспечения

Студенту Забаровскому Роману Вадимовичу

специальности 09.03.01 Информатика и вычислительная техника

1 Тема курсового проекта: Web-приложение ведения складского учета

2 Срок подачи студентом законченной работы: 06.04.2022 г.

3 Исходные данные к проекту: описание предметной области

4 Перечень подлежащих разработке в курсовом проекте вопросов:

1. Постановка и анализ задачи;
2. Анализ задачи;
3. Программная реализация;
4. Тестирование;
5. Техническое задание;
6. Руководство пользователя.

5 Перечень графического материала: -

Дата выдачи задания 09.02.2022 г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Валова О.В./

(подпись, расшифровка подписи)

Задание принял к исполнению

«09» февраля 2022 г.

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Забаровский Р.В./

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Забайкальский государственный университет»

(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

по 09.03.01 Информатика и вычислительная техника

на тему: Web-приложение ведения складского учета

Выполнил студент группы ИВТ-18 Забаровский Роман Вадимович

Руководитель работы: доцент кафедры ИВТ и ПМ, к.т.н., доцент Валова Ольга Валерьевна

РЕФЕРАТ

Пояснительная записка - 30 с, 10 рис., 6 таб., 6 источников.

ПОЛЬЗОВАТЕЛЬ, АДМИНИСТРАТОР, ТОВАР, БАЗА ДАННЫХ, PHP, MYSQL, СУБД, WEB-ПРИЛОЖЕНИЕ, ВЕБ-ПРИЛОЖЕНИЕ.

* + 1. **Постановка и анализ задачи**
  1. **Описание предметной области**

За последнее двадцатилетие наука значительно продвинулась в изучении информационных технологий. Во многом производственная и бумажная работа упрощается при их использовании на этапе создания, но и не ограничивается на последующих этапах распределения, поставок, хранения, продажи и обслуживания.

Данная работа выполнена для «Общества с ограниченной ответственностью “Свежий ветер”». Предприятие занимается монтажом систем вентиляции и кондиционирования, установкой холодильного оборудования, технологическим проектированием, монтажом инженерных сетей, к которым относятся: водоснабжение, отопление, электроснабжение, канализация, системы автоматизации и управления.

Цель разработки web-приложения – оперативный учет складских операций, печать складских документов, анализ количества запасов, их стоимости. В данном приложении необходимо предусмотреть разный функционал для двух видов пользователей: администратор базы данных, складской рабочий. Также в процессе разработки web-приложения должны быть приобретены навыки проектирования баз данных, требующиеся для полной работоспособности системы приложения.

Одним из важных шагов при разработке приложения, - это выбор правильного стека технологий. От него не в последнюю очередь зависит успех всего проекта. Он влияет на количество материальных вложений в проект, сроки разработки, безопасность и модифицируемость. В первую очередь перед тем как говорить о выборе стека технологий, нужно пояснить, что такое веб-приложение – это сайт, в котором клиент взаимодействует с веб-сервером при помощи браузера, то есть может нажимать кнопки, вводить данные в формы, оформлять документы и т.п. Примером веб-приложения могут выступать интернет магазины, социальные сети, поисковые службы и т.д.

Архитектура веб-приложения состоит из двух частей. Первая клиентская – она реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы. Также называется фронтенд (front-end). Вторая серверная – она реализует логику. Также называет бэкенд (back-end). К технологиям фронтенда относятся HTML, CSS, JavaScript, а бэкенда относят PHP, Java, Python и др.

Подход к разработке также делиться на несколько видов:

* Single Page Application (SPA) или веб-приложение с одной страницей.
* Multi Page Application (MPA) – многостраничные приложения.
* Progressive Web Application (PWA) – гибрид сайта и мобильного приложения.

В настоящее время множество предприятий уже используют электронные системы, для эффективной работы. Например, такой гигант как «1С: Торговля и склад» удерживает лидерство в предпочтении крупных предприятий, но малый бизнес предпочитает платить меньшие деньги за обслуживание и функциональное насыщение. В результате чего на рынке существует сильный спрос на аналоги данного программного обеспечения, которые проще в функциональной части. На что разработчики выпустили множество простых и понятных программ складского учета для определенных торговых специфик: «ЕКАМ», «Дебет плюс», «МойСклад» и др. Но основной проблемой данных программ является большой разброс цены и возможности настройки предприятием программы для своего направления, если оно имеет специфичные товары и услуги.

* 1. **Диаграмма Вариантов использования**

В веб-приложении предусмотрено два уровня доступа к данным, поэтому пользователи системы поделены на две группы. На диаграмме вариантов использования они представлены в виде двух действующих лиц:

* администратор базы данных;
* пользователь.

Для работы веб-приложения необходимо реализовать функции, показанные на диаграмме использования для администратора и пользователя.

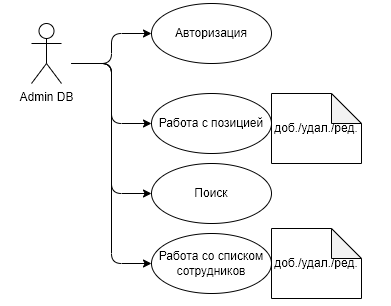


Рисунок 1 - Диаграмма вариантов использования для администратора

Для действующего лица администратора предполагаются следующие варианты использования:

* авторизация, которая предусматривает поддержание сессии текущего пользователя;
* работа с позицией, которая предусматривает добавление, удаление и редактирование типов товаров в базе данных;
* пользование поиском;
* работа со списком сотрудников, которая предусматривает добавление, удаление и редактирования сотрудников в базе данных.

Администратор веб-приложения перед началом работы должен авторизоваться, после этого ему будут доступны функции для работы со списком товаров и списком сотрудников. Диаграмма вариантов использования администратора представлена на рисунке 1.

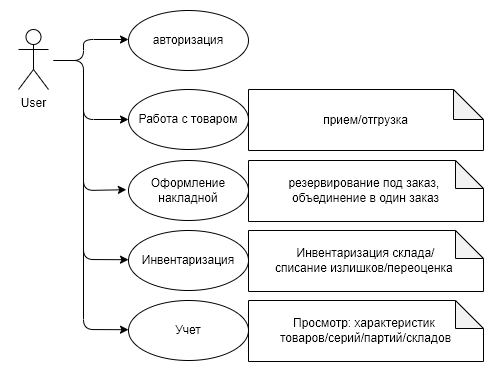


Рисунок 2 - Диаграмма вариантов использования для пользователя

Для пользователя веб-приложения предполагаются следующие варианты использования:

* авторизация, которая предусматривает поддержание сессии текущего пользователя;
* работа с товаром, которая предусматривает прием, отгрузку;
* оформление накладной, которая предусматривает работу с документом на товар: резервирование под заказ, объединение в один заказ;
* инвентаризация, которая предусматривает инвентаризацию склада, списание излишков и переоценку товаров;
* учет, которая предусматривает просмотр атрибутов товаров, характерных для них.

Также, как и администратор пользователь перед началом должен авторизоваться, после этого ему станут доступны функции работы с товаром, оформления накладных, инвентаризации и учета. Диаграмма использования для пользователя представлена на рисунке 2.

В свою очередь работу с товаром, оформление накладной и инвентаризацию можно представить в виде отдельных диаграмм. Диаграммы вариантов использования представлены на рисунке 3.

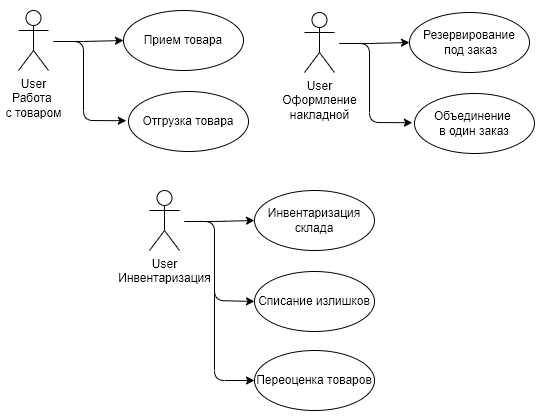


Рисунок 3 - Диаграммы вариантов использования

* 1. **Обоснование выбора средств реализации**

Реализация веб-приложения произведена на следующем стеке технологий:

* язык гипертекстовой разметки HTML;
* формальный язык внешнего вида документа CSS;
* скриптовый язык общего назначения PHP;
* СУБД PosgreSQL. На данной СУБД разработана база данных для хранения данных о товарах и сотрудниках;
* язык клиентской части веб-приложения JavaScript;
* все страницы в проекте динамические и построены на шаблонах.

Системой управления базами данных (СУБД) была выбрана PostgreSQL. Эта СУБД является бесплатной и её достаточно просто настроить для работы с базами данных. Сервером базы данных была выбрана вычислительная машина под управлением операционной системы семейства Windows, но также сервером может выступать машина под управлением других операционных систем таких как Linux и macOS.

Шаблонизация заключается в следующем: разделение представления данных от исполняемого кода. Для этого создаются отдельные файлы (шаблоны) HTML для каждого элемента, содержимое которого может отличаться. Описав шаблон единожды, его можно использовать множество раз, а изменения в шаблоне будут видны везде, где он используется.

Языком программирования для реализации веб-приложения были выбраны: PHP, JavaScript. PHP – это скриптовый язык общего назначения, с его помощью серверная часть принимает запросы, работает с базой данных, производит необходимую проверку данных и позволяет идентифицировать пользователя с помощью сессий. Его преимущества: большой набор встроенных средств для разработки веб-приложений, бесплатен, большое количество открытой документации, простой синтаксис. JavaScript – динамический язык, используемый для создания интерактивных веб-приложений. Он используется в клиентской части приложений: клиент-серверных программ, где клиентом выступает браузер, а сервером – веб-сервер, имеющих распределенную логику. Выбор представлен эффективностью и возможностями языков программирования.

* + 1. **Анализ данных**

Данные, с которыми работает система делятся на три группы. Первая группа входных данных – данные, которые поступают от пользователя в систему. Вторая группа промежуточных данных – данных, которые система использует во время своей работы. И третья группа выходных данных – данные, которые система выводит пользователю.

* 1. **Входные данные**

Входной информацией в разрабатываемом приложении будут выступать такие данные:

* логин;
* пароль;
* информация о пользователе: имя, фамилия, дата рождения, телефонный номер, роль;
* информация о товаре: наименование, материал, длина, размер сечения, количество.

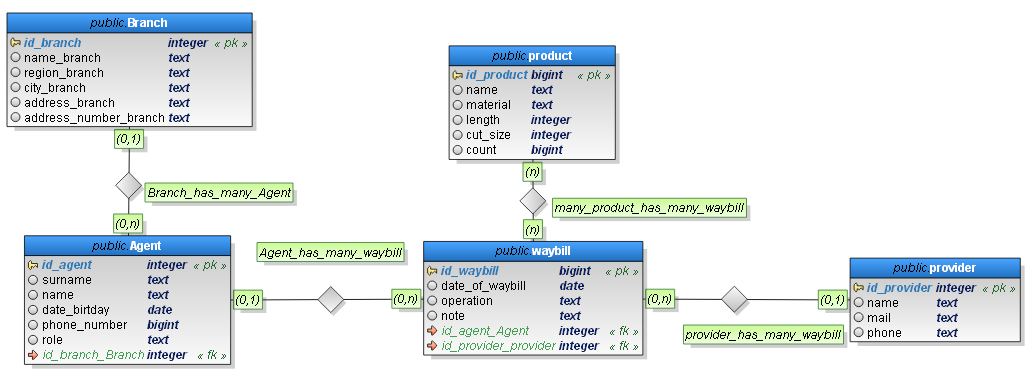


Рисунок 4 – Модель базы данных

Каждая таблица базы данных, которая была разработана состоит из атрибутов для хранения информации о том или ином объекте предметной области. Связи между таблицами показаны на рисунке 4.

* 1. **Промежуточные данные**

Во время работы приложения промежуточными данными являются таблицы базы данных. Вся информация о данных, содержащихся в таблицах, находится в приложении А.

Таблица Admin содержит данные о логине и пароле администратора базы данных для входа в приложение и личную информацию о пользователе, такую как имя, фамилия, дата рождения, телефонный номер.

Таблица Agent содержит информацию о паролях и логинах пользователей, а также личную информацию: имя, фамилию, дату рождения, телефонный номер.

Таблица Product хранит информацию о всех товарах, такую как наименование, материал, размер сечения, длина, количество, цена.

Таблица Waybill хранит данные о всех операциях, совершенных пользователем.

Таблица Provider содержит информацию о поставщиках, их имя, электронную почту и телефонный номер.

Таблица Branch хранит информацию о всех складских помещениях организации.

* 1. **Выходные данные**

К выходным данным можно отнести:

* информационные веб-страницы веб-приложения;
* списки товаров;
* списки операций;
* списки поставщиков;
* списки пользователей;
* отчеты.

1. **Программная реализация**

В ходе программной реализации разработаны алгоритмы для авторизации, внесения данных в базу данных, изменения данных в базе данных.

При авторизации для каждого пользователя в зависимости от уровня доступа загружается свой интерфейс. На странице авторизации загружается модуль layout.php, в котором для не авторизованного пользователя также подгружается свой интерфейс из модуля navigation\_template.php.

Если авторизовался пользователь с правами администратора, то модуль layout.php подгрузит модуль navigation\_admin\_template.php, который отвечает за отображение навигационного меню администратора, а модуль log-in.php подгрузит модуль главной страницы администратора main\_admin\_page.php, где содержатся таблицы пользователей, товаров, поставщиков и операций.

Если авторизовался пользователь с правами пользователя, то модуль layout.php подгрузит модуль navigation\_user\_template.php, который отобразит навигационное меню для роли пользователь. Модуль log-in.php же подгрузит модуль main\_user\_page.php, на которой содержатся элементы пользовательской страницы.

Для внесения данных в базу данных и изменения их в ней были разработаны алгоритмы для связи с базой данных. Модуль database.php открывает подключение к базе данных по заранее известным данным, таким как: логин, пароль, название базы данных, адрес хоста, порт подключения. Открытие соединения для работы с базой данных происходит на этапе авторизации, а закрытие при выходе из системы, либо автоматически через определенный интервал времени. Внесение и изменения данных происходит через модуль query.php, который используется в интерфейсе пользователя, вызовом методов insert и update, либо delete.

* + 1. **Тестирование**

По мере разработки программного продукта проводились многочисленные тесты для выявления ошибок и уязвимостей системы.

Первый тест был направлен на то, чтобы при неверной комбинации логина и пароля система должна оповестить об этом пользователя.

Если пользователь ввел неверную комбинацию логина и пароля, то он увидит следующее сообщение, как на рисунке 4.

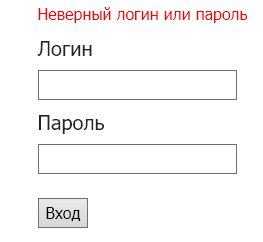


Рисунок 5 – Ошибка введенного логина или пароля

Второй тест направлен на уведомление пользователя о незаполненных полях товаров и услуг. Если пользователь не заполнит ключевые поля, то получит ошибку, как показано на рисунке 5.

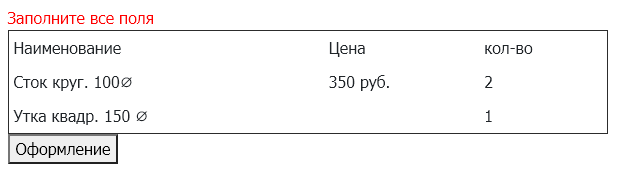


Рисунок 6 – Ошибка о пустых полях

Результаты тестирования подтверждают, что данные методы защиты работают и повышают работоспособность системы.

На текущий момент система находится в стадии разработки, следовательно, при каждом добавление новых функций происходит тестирование и отладка.

1. **Документирование**
   1. Техническое задание
      1. Введение

Проект «Web-приложение для ведения складского учета» предоставляет возможность оформления товаров и услуг складским работником. Программа представлена в виде веб-сервиса.

* + 1. **Назначение разработки**

Данный сервис разрабатывается с целью оптимизации работы компании и повышению продаж.

* + 1. **Требования к функциональным характеристикам**

В данном сервисе, в соответствии с диаграммой использования предусмотрены два действующих лица:

Администратору приложения программный продукт должен предоставить следующие возможности, связанные с полноценным администрированием веб-сервиса:

* авторизация;
* редактирование списка товаров;
* редактирование списка сотрудников.

Пользователю, как сотруднику организации, требуется другой функционал. Веб-сервис должен предоставить следующие функциональные возможности:

* авторизация;
* возможность приема, отгрузки товара;
* возможность оформления накладной;
* возможность списания излишков;
* возможность переоценки товаров;
* возможность учета товаров.
  + 1. **Требования к надежности**

Веб-сервер должен уметь обрабатывать неправильные запросы и отправлять пользователю уведомления об ошибках. Для изучения ошибок и последующего устранения, необходимо заносить сведения обо всех ошибках в журнал ошибок.

При возникновении несерьезных ошибок, которые пользователь может исправить самостоятельно, такие как некорректный ввод данных, сервис должен оповестить об этом пользователя.

* + 1. **Системные требования**

Наличие в системе одного из интернет-обозревателей с версиями не ниже указанных:

* Edge 12;
* Firefox 21;
* Chrome 23.
  + 1. Требования к программной документации

Программная документация должна содержать исчерпывающее руководство пользователя с полным описанием действий пользователя.

* + 1. Требования к составу и параметрам технических средств

Минимальные требования для корректной работы сервиса необходимо: персональный компьютер с любой операционной системой, встроенной оперативной памятью не менее 512 Мб, также свободного места на диске должно быть не менее 550 Мб, процессор с частотой 600 Мгц.

* + 1. **Требования к информационной и программной совместимости**

Сервер не требует определенной операционной системы. На рабочей станции должен быть установлен любой браузер, подходящий по системным требованиям, описанным выше.

* 1. **Руководство пользователя**

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЕ А

(обязательное)

Описание таблиц базы данных

Таблица А.1 – Таблицы Базы данных

|  |  |
| --- | --- |
| Название таблицы | Описание |
| Branch | Информация об отделениях |
| Agent | Информация о пользователях |
| product | Информация о товарах |
| waybill | Накладные |
| provider | Информация о поставщиках |

Таблица А.2 – Описание таблицы Branch

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id\_branch | integer | Идентификатор |
| name\_branch | varchar | Название отделения |
| region\_branch | varchar | Регион отделения |
| city\_branch | varchar | Город отделения |
| address\_branch | varchar | Название улицы отделения |
| address\_number\_branch | varchar | Номер улицы отделения |

Таблица А.3 – Описание таблицы Agent

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id\_agent | integer | Идентификатор |
| surname | varchar | Фамилия |
| name | varchar | Имя |
| date\_birtday | date | Дата рождения |
| phone\_number | varchar | Телефонный номер |
| role\_personal | varchar | Роль пользователя |
| pass | varchar | Пароль пользователя |
| id\_branch\_Brarnch | integer | Идентификатор отделения |

Продолжение приложения А

Таблица А.4 – Описание таблицы product

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id\_product | integer | Идентификатор |
| name\_product | varchar | Наименование |
| material | varchar | Материал |
| length | integer | Длина |
| cut\_size | integer | Размер сечения |
| count | bigint | Количество |
| price | bigint | Цена за единицу |

Таблица А.5 – Описание таблицы waybill

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id\_waybill | integer | Идентификатор |
| operation | varchar | Название операции |
| date\_of\_waybill | varchar | Дата накладной |
| note | integer | Пояснение |
| id\_agent\_FK | integer | Идентификатор пользователя |
| id\_provider\_FK | bigint | Идентификатор поставщика |

Таблица А.6 – Описание таблицы provider

|  |  |  |
| --- | --- | --- |
| Название | Тип | Описание |
| id\_provider | integer | Идентификатор |
| name\_provider | varchar | Имя поставщика |
| mail | varchar | Почта |
| phone | varchar | Телефонный номер |
| address | varchar | Адрес |

ПРИЛОЖЕНИЕ Б

(обязательное)

Алгоритм авторизации

<?php

require\_once 'database.php';

require\_once 'query.php';

if (isset($\_POST['name']) && !empty($\_POST['name']) )

{

$login = $\_POST["name"];

$pass = $\_POST["password"];

$db = new DB();

$query = "

select \"surname\"

from \"Agent\"

where '$login' = \"surname\" and '$pass' = \"pass\" ";

$query\_result = pg\_query($db->get\_link(), $query);

$result = pg\_fetch\_all($query\_result, PGSQL\_ASSOC);

$role = get\_role($login);

var\_dump($result);

if (!empty($result)) {

session\_start();

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

if ($role[0]["role\_personal"] == "admin") {

header('Location:http://localhost/admins/main\_admin\_page.php');

} else {

Продолжение приложения Б

header('Location:http://localhost/users/main\_page.php');

}

}

else{

$error = "reg error";

header("Location:http://localhost/index.php?error=$error");

}

}

?>

ПРИЛОЖЕНИЕ В

(обязательное)

Запрос к серверу

<?php

require\_once 'database.php';

// Создание запроса

function query($query){

$db = new DB();

return pg\_query($db->get\_link(), $query);

}

// Запрос с возвратом массива

function query\_fetch\_all($query){

return pg\_fetch\_all(query($query), PGSQL\_ASSOC);

}

// Запрос с возвратом роли пользователя

function get\_role($surname){

return pg\_fetch\_all(query("select \"role\_personal\" from \"Agent\" where '$surname' = \"surname\""));

}

// Запрос с возвратом пользователя

function get\_user($surname){

return pg\_fetch\_all(query("select \* from \"Agent\" where '$surname' = \"surname\""));

}