* + 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.

Для действующего лица администратора предполагаются следующие варианты использования:

* авторизация, которая предусматривает поддержание сессии текущего пользователя;
* работа с позицией, которая предусматривает добавление, удаление и редактирование 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.

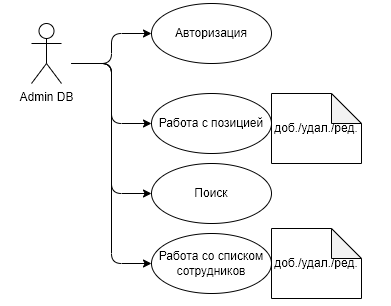


Рисунок 1 - Диаграмма вариантов использования для администратора

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.

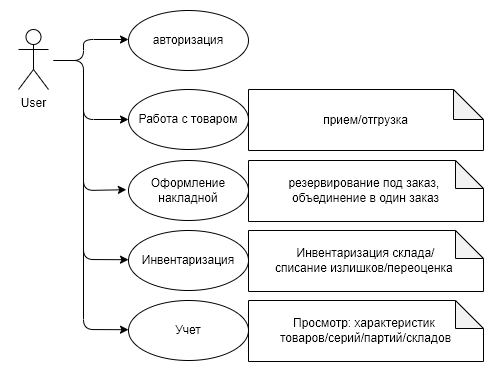


Рисунок 2 - Диаграмма вариантов использования для пользователя

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

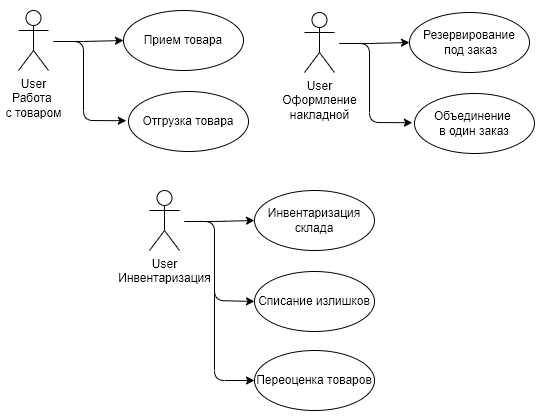


Рисунок 3 - Диаграммы вариантов использования

* 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**
   1. **Обоснование выбора средств реализации**

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;
* имя пользователя;
* фамилия пользователя;
* дата рождения пользователя;
* телефонный номер пользователя;
* роль пользователя;
* наименование товара;
* материал товара;
* длина материала;
* размер сечения;
  1. **Промежуточные данные**
  2. **Выходные данные**