

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

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

Кафедра дискретной математики и информационных технологий

**РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ
ПРОЦЕССА ОФОРМЛЕНИЯ УСЛУГ РЕМОНТА И ПРОДАЖИ
ТЕХНИКИ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Губайдулина Дамира Амиржановича

Научный руководитель
доцент, к. ф.-м. н.

В. А. Поздняков

Заведующий кафедрой
доцент, к. ф.-м. н.

Л. Б. Тяпаев

Саратов 2022

ВВЕДЕНИЕ

Сегодня интернет открывает безграничные возможности, связывая людей со всего мира внутри одной системы. Уже долгое время данная технология является наиболее эффективным методом распространения рекламы, ведения продаж, предоставления услуг, став важным элементом современной цивилизации. Процесс покупок стал более комфортным, позволяя, не выходя из дома выбирать товары, решая проблемы очередей и скоплений большого количества людей. «Если вашего бизнеса нет в интернете, то Вас нет в бизнесе» – эти слова Билла Гейтса как нельзя лучше описывают современную экономическую ситуацию. В настоящее время иметь свой собственный сайт для развития своей торговли это уже не мода, а необходимость.

Чтобы пользователю запустить веб-приложение нет необходимости устанавливать какие-либо дополнительные программы или докупать различное оборудование, оно выполняется на любом устройстве с доступом в интернет. Для разработки существует большое множество технологий и связанных с ними фреймворками, поэтому каждый может подобрать удобный стек средств для разработки различных веб-приложений.

Актуальность темы в том, что количество пользователей с доступом в интернет непрерывно растёт и поэтому для небольшой компании наличие в интернете своего веб-приложения способствует автоматизации различных процессов, а также увеличению количества клиентов.

Целью данной выпускной дипломной работы является разработка веб-приложения для автоматизации процесса оформления услуг ремонта и продажи различной техники.

Для достижения цели необходимо решить следующие задачи:

- изучить определение веб-приложения и её структуру;
- исследовать основные технологии разработки современных веб-приложений;
- изучить фреймворк Spring по разработке веб-приложений на языке Java;
- реализовать клиентскую и серверную части веб-приложения.

Бакалаврская работа состоит из введения, 4 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы – 78 страниц, из них 53 страницы – основное содержание, включая 45 рисунков, список использованных источников информации – 29 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе «Веб-приложения» рассматривается понятие веб-приложения, что они из себя представляют, каких бывают видов и какая у неё архитектура.

Веб-приложение – это клиент-серверная программа, в которой клиент взаимодействует с веб-сервером посредством использования браузера. Логика многих веб-приложений распределены между сервером и клиентом, хранение данных осуществляется, чаще всего, на сервере, а обмен информацией происходит по сети [1].

Веб-приложения можно характеризовать пятью типами:

1. Серверные веб-приложения. Различные переходы между страницами вызывают генерацию нового контента, которая отображается у клиента на странице браузера.
2. Клиентские веб-приложения. В чистом виде данный вид приложений не требует серверной части, а с реализацией можно справиться возможностями JavaScript, используя в качестве оболочки браузер пользователя.
3. SPA-приложения – это одностраничные веб-приложения, реализующие сложный функционал в рамках одного окна браузера без перезагрузки страницы.
4. MPA-приложения – это многостраничные веб-приложения, которые применяются для построения сложных систем. Любые изменения данных приводит к полной перезагрузке страницы в браузере.
5. PWA-приложения – это прогрессивные веб-приложения, которые являются способом подачи веб-сервисов, тем самым максимально сближая их с привычными десктопными приложениями, но на качественно более высоком уровне [2].

Архитектура веб-приложения в основном представляет отношения и взаимодействия между такими компонентами, как пользовательские интерфейсы, базы данных, мониторы обработки транзакций и другие. Основная цель – убедиться, что правильно работают совместно все элементы.

Логика работы следующая: в момент, когда пользователь вводит URL-адрес в строке адреса браузера и нажимает клавишу ввода, браузер отправляет запрос на сервер. Далее этот сервер отвечает и показывает требуемую веб-страницу. Эти компоненты создают архитектуру веб-приложения.

Существует четыре общих уровня веб-приложений:

- Уровень представления (PL) – отображает пользовательский интерфейс и упрощает взаимодействие с пользователем.
- Уровень обслуживания данных (DSL) – передает данные, обработанные уровнем бизнес-логики, на уровень представления.
- Уровень бизнес-логики (BLL) – несет ответственность за надлежащий обмен данными.
- Уровень доступа к данным (DAL) – предлагает упрощенный доступ к данным, хранящимся в постоянных хранилищах, таких как двоичные файлы и файлы XML [3].

Веб-приложения состоят из клиентской и серверной частей.

Клиентская часть предназначена для взаимодействия веб-приложения с пользователем и реализуется на компьютере в рамках веб-браузера, смартфона или планшета в виде набора динамических HTML-страниц.

Под серверной частью понимают набор аппаратно-программных средств, с помощью которых реализована логика работы веб-приложения. Она выполняется вне браузера и компьютера пользователя. К данной части относится панель администрирования, работа с данными, логика их передачи посредством запросов на клиентскую часть [1].

Во второй главе «Основные технологии разработки веб-приложений» приведено описание программных средств для создания современных веб-приложений. Данный раздел включает в себя описание основных технологий клиентской и серверной частей, а также способы хранения данных.

Один из наиболее важных шагов при проектировании и разработки веб-приложений – это выбор правильного стека технологий. Этот фактор в значительной степени влияет на успех, сроки выполнения, масштабируемость, безопасность и стоимость проекта.

HTML – это язык гипертекстовой разметки, используемый для создания документов веб-страниц. Он насыщает страницу текстом, изображениями, анимацией, видео и аудио материалами и различными ссылками на элемент документа данного проекта или сервера, или на сторонние ресурсы.

CSS – язык для описания внешнего вида HTML-документа. Отвечает за внешний вид объектов на странице. Их размер, цвет, задний фон, степень прозрачности, радиус формы, расположение относительно других элементов, поведение при наведении курсора и многое другое.

JavaScript – язык программирования, который помогает реализовывать сложное поведение веб-страницы. В большинстве случаев JavaScript используется для создания адаптивных интерактивных элементов для веб-страниц, которые улучшают взаимодействие с пользователем. JavaScript позволяет перехватывать события на странице веб-приложения и выполнять над ними различные действия без перезагрузки страницы.

PHP – один из самых распространенных языков веб-разработки. Язык очень популярен, потому что очень быстро обрабатывает программный код, прост в использовании и функционален, распространяет исходные коды с собственной лицензией и обладает кроссплатформенностью.

Java – язык программирования общего назначения, то есть он может применяться в разработке различных программных продуктов, не имея четкой специализации в конкретной сфере. Это объектно-ориентированный язык, где вся структура строится вокруг объектов, классов, интерфейсов, прочих формальных сущностей и принципов.

Node.js – представляет собой платформу для написания JavaScript-приложений с использованием внешних библиотек. С помощью Node код обращается к жесткому диску, базам данных и сети. Это делает возможным написание абсолютно любых приложений: от утилит командной строки и видеоигр до полноценных веб-серверов.

Ruby on Rails (RoR) – это написанный на языке программирования Ruby фреймворк, реализует архитектурный MVC шаблон для веб-приложений, а также обеспечивает их интеграцию с веб-сервером и сервером баз данных. Он поощряет и облегчает использование веб-стандартов, таких, как JSON или XML для передачи данных, а также HTML, CSS и JavaScript для взаимодействия с пользователями [4].

Веб-приложениям требуется место для хранения информации, и для этого используется база данных. База данных – это место хранения и систематизации всех данных, которые собирает веб-приложение, а система управления базой данных (СУБД) – это программное обеспечение для удобного управления базой данных. Существует два типа баз данных: реляционные и нереляционные. Различия между ними заключаются в способе проектирования, какие поддерживают типы данных, как хранят информацию.

Реляционная база данных – это набор таблиц, между которыми уста-

новлены определенные взаимосвязи. Реляционные БД хранят структурированные данные, которые обычно представляют объекты реального мира.

Нереляционные БД устроены иначе, например, документо-ориентированные базы хранят информацию в виде иерархических структур данных. Речь может идти об объектах с произвольным набором атрибутов. То, что в реляционной БД будет разбито на несколько взаимосвязанных таблиц, в нереляционной может храниться в виде целостной сущности [5].

HTTP (HyperText Transfer Protocol) – это широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (то есть документов, которые могут содержать ссылки, позволяющие организовать переход к другим страницам). В соответствии со спецификацией OSI, HTTP является протоколом прикладного уровня [6].

Веб-API – это интерфейс прикладного программирования для веб-сервера или веб-браузера. ПО использует эти интерфейсы для обмена данными от приложений на смартфонах до глубоко скрытых серверов баз данных [7].

REST (Representational State Transfer) – это архитектурный стиль взаимодействия компонентов распределённого приложения в сети. Иначе говоря, это набор правил того, каким образом программисту необходимо организовать написание кода серверного приложения, чтобы все системы могли легко обмениваться данными и приложение можно было масштабировать [8].

В третьей главе «Spring Framework для разработки веб-приложений» рассмотрен использованный в создании веб-приложения фреймворк Spring, а также средство Gradle для сборки проекта.

Spring – это самым популярным фреймворком разработки веб-приложений на языке программирования Java, созданный Родом Джонсоном и основанный на идее внедрения зависимостей. Инфраструктура внедрения зависимостей позволяет эффективнее управлять большими Java-проектами за счет оформления отношений между объектами в приложении через аннотации вместо создания жестких связей.

Целью Spring является облегчение разработки корпоративных приложений и поощрение слабой связанности. Фреймворк Spring позволяет исполь-

зовать лучшие практики и эффективно проектировать объекты.

Фреймворк Spring использует четыре ключевые стратегии:

- легковесность благодаря применению простых Java-объектов (POJO);
- слабое связывание посредством внедрения зависимостей и ориентированности на интерфейсы;
- программирование через аспекты и общепринятые соглашения;
- уменьшение объема типового кода через аспекты и шаблоны [9].

Spring Boot – это переосмысление фреймворка Spring. С помощью нескольких простых аннотаций разработчик Java может быстро создать службу REST, упаковать и развернуть ее без использования внешнего приложения в роли контейнера. Spring Boot упрощает замену компонентов при изменении проектных требований. Он основывается на экосистеме Spring и библиотеках сторонних разработчиков, включая мнения и формирование соглашений, призванных упростить реализацию приложений, готовых к эксплуатации.

Технология Spring Boot предоставляет следующие ключевые особенности:

- встроенный веб-сервер, помогающий избежать сложностей при развертывании приложения: Tomcat, Jetty или Undertow. Это один из важнейших компонентов Spring Boot. Выбранный веб-сервер включается в развертываемый архив JAR;
- предопределенная конфигурация для быстрого начала работы над проектом;
- автоматическая настройка возможностей Spring;
- широкий спектр готовых к использованию возможностей в промышленном окружении (таких, как метрики, безопасность, проверка статуса, хранение конфигурации вовне и так далее) [10].

Spring Security – чрезвычайно широко настраиваемый и обладающий большими возможностями фреймворк для аутентификации и авторизации (или управления доступом). Это модуль, используемый по умолчанию для обеспечения безопасности приложений Spring [11].

Gradle – это система для автоматизации сборки приложений и сбора сведений об использовании программных библиотек, применяющая языки Groovy, Java, JavaScript, Kotlin и так далее, а также решения из фреймворков Apache Ant и Apache Maven [12].

В четвёртой главе «Процесс разработки веб-приложения» рассмотрена реализация клиентской и серверной части веб-приложения, а также последовательная работа различных функций.

Чаще всего сложные веб-приложения состоят минимум из трёх основных компонентов, это клиентская и серверная части, а также база данных.

Серверная часть определяется выполнением обработки запросов от браузера на веб-сервере. В данном проекте используется фреймворк Spring и код на языке Java для получения запросов, их обработки и отправки сгенерированной HTML-страницы в качестве результата, отображаемой на странице интернет-браузера.

Для того чтобы разработать серверную часть были использованы следующие программы и технологии:

1. IntelliJ IDEA – интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, Python, разработанная компанией JetBrains.
2. Docker – программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений.
3. Java – строго типизированный объектно-ориентированный язык программирования общего назначения.
4. Spring Framework – универсальный фреймворк с открытым исходным кодом для Java-платформы по созданию веб-приложений.
5. Gradle – система автоматической сборки, построенная на принципах Apache Ant и Apache Maven.

Клиентская часть обусловлена графическим отображением страницы в окне браузера. Пользователь непосредственно взаимодействует с данной частью, нажимает на кнопки, переходит по ссылкам и многое другое.

Для клиентской части были использованы следующие языки и методы:

1. Thymeleaf – современный серверный механизм Java-шаблонов для веб и автономных сред, способный обрабатывать HTML, XML, JavaScript, CSS и даже простой текст.
2. HTML – стандартизированный язык разметки документов для просмотра веб-страниц в браузере.
3. CSS – формальный язык описания внешнего вида веб-страницы.

4. JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.
5. JQuery – набор функций JavaScript, фокусирующийся на взаимодействии JavaScript и HTML.
6. Bootstrap – свободный набор инструментов для создания сайтов и веб-приложений.

Исходя из цели работы веб-приложение связано с большим объёмом взаимодействий с базой данных. Созданный проект должен выполнять следующие функции:

1. Сохранение данных.
2. Получение данных из БД и их вывод на страницу веб-приложения.
3. Редактирование данных и обновление их на страницах веб-приложения.
4. Удаление данных.

Данное веб-приложение разработано для магазина по ремонту и продаже техники. Для удобного перемещения по элементам страницы имеется навигационное окно. На ней имеются ссылки на основные разделы, среди которых: главная, магазин, ремонт, контакты, отзывы, предложения, а также вход и регистрация для неавторизованного пользователя.

Страница с товарами обладает основными возможностями популярных веб-приложений, такие как: сортировка данных, фильтрация данных и поиск товаров по названию и ключевому полю. По нажатию на кнопку «Подробнее» происходит переход на новую страницу веб-приложения с подробной информацией о товаре.

Для того чтобы добавить товар в избранное или корзину необходима авторизация. Веб-приложение относится к классу приложений с множественным доступом – разные пользователи отправляют запросы на один веб-сервер. В связи с этим возникает задача идентификации отправителя запроса и определение прав пользователя (отправителя) на выполнение запроса (аутентификация).

Многие приложения хранят информацию о пользователях, включая имена и пароли в реляционной базе данных. Если в приложении предполагается хранить информацию о пользователях именно таким способом, для этой цели можно использовать элемент Spring Security.

Со стороны авторизованного аккаунта возможно изменить свои личные данные, такие как имя пользователя, почтовый адрес, номер телефона, изображение, пароль, а также удаление аккаунта с дополнительным подтверждением пользователя.

После авторизации пользователь может оценить магазин на странице с отзывами, оставив комментарий и выбрав оценку работы доставки и всей деятельности в целом. Также предоставляется возможность оформить все товары или только один из предложенных товаров, а также удалить товар из корзины. После оформления происходит оповещения пользователя о совершённом действии. Также продавец получает сообщение на почтовый адрес компании с оформленной покупкой.

Для владельца веб-приложения важно иметь возможность влиять на отображаемую на странице информацию. Для такой цели подходит панель администратора, главными целями которой является редактирование информации на сайте, получение справок о всех действиях, ответ на заявки по ремонту и многое другое. Панель должна быть защищена от обычных пользователей, потому что неправомерное обладание возможностью менять важную информацию может привести к серьёзным последствиям для компании.

Перейти к написанию заявки можно несколькими способами. Первый из них это воспользоваться навигационными меню. Второй способ это открыть из главной страницы нажав на кнопку «Оформить заявку». После перехода по кнопке появится панель-оформление. Полный процесс оформления содержит пять последовательных шага.

Первый шаг – это выбрать категорию товара, подверженную поломке.

Второй шаг – это выбрать что именно случилось с техникой.

Третий шаг – это добавление описания к существующей проблеме.

Четвёртый шаг – это оставить некоторые данные, необходимые для средства связи с клиентом, а также получить некоторые необходимые данные об услуге.

И последний шаг – это подтверждения заявки.

ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы была достигнута основная цель, а именно было создано функциональное веб-приложение, предоставляющая возможность оформлять заявки на ремонт и продавать технику.

Для осуществления разработки были использованы следующие веб-технологии:

- Spring Framework в качестве контейнера внедрения зависимостей.
- Технология Spring Boot, целью которой является упрощение создания веб-приложений на языке Java.
- Язык программирования Java для разработки бизнес-логики веб-приложения.
- Языки HTML, CSS, JavaScript, JQuery, шаблонизатор Thymeleaf и технология Bootstrap для создания клиентской части веб-сайта.
- Docker для автоматизации развёртывания и управления приложением.
- Gradle для автоматической сборки проекта.
- База данных MySQL для хранения всех сведений.

При разработке архитектуры веб-приложение было разделено на две части:

- клиентская часть, содержащая общедоступную информацию и функции работы с товарами и другими компонентами веб-приложения;
- администраторская часть, имеющая инструменты управления контентом внутри веб-приложения и возможность взаимодействовать с основной базой данных.

Таким образом, все поставленные задачи по реализации работы успешно решены, а разработанная информационная система соответствует всем необходимым требованиям.

Основные источники информации:

- 1 Базовые элементы Web-технологий: [Электронный ресурс] URL: [panda.ispras.ru/ kuliamin/lectures-wtp/Basic%20Web%20Technologies.pdf](http://panda.ispras.ru/kuliamin/lectures-wtp/Basic%20Web%20Technologies.pdf)(дата обращения: 16.01.2022). – Загл. с экрана. – Яз.рус.
- 2 Web-приложение: понятие, компоненты и принципы работы: [Электронный ресурс] URL: smartika.ru/courses/web/lesson-1 (дата обращения: 17.01.2022). – Загл. с экрана. – Яз.рус.
- 3 Архитектура веб-приложения: компоненты, слои и типы: [Электронный

- ресурс] URL: itanddigital.ru/webapplications(дата обращения: 17.01.2022). – Загл. с экрана. – Яз.рус.
- 4 Веб-технологии разработки приложений: Java, PHP, JavaScript: [Электронный ресурс] URL: www.azoft.ru/blog/web-development-stack/(дата обращения: 17.01.2022). – Загл. с экрана. – Яз.рус.
- 5 Ключевые аспекты при выборе базы данных для вашего приложения: [Электронный ресурс] URL: https://habr.com/ru/company/otus/blog/562852/(дата обращения: 19.01.2022). – Загл. с экрана. – Яз.рус.
- 6 Простым языком об HTTP: [Электронный ресурс] URL: habr.com/ru/post/215117/ (дата обращения: 24.01.2022). – Загл. с экрана. – Яз.рус.
- 7 REST API: что это простыми словами, расшифровка, примеры запросов: [Электронный ресурс] URL: blog.skillfactory.ru/glossary/rest-api/(дата обращения: 19.03.2022). – Загл. с экрана. – Яз.рус.
- 8 Что такое REST API и чем оно отличается от другого API?: [Электронный ресурс] URL: appmaster.io/ru/blog/chto-takoe-rest-api-i-chem-ono-otlichaetsya-ot-drugogo-api(дата обращения: 19.03.2022). – Загл. с экрана. – Яз.рус.
- 9 Spring в действии / Уоллс К. – М.: ДМК Пресс, 2013. 752 с.
- 10 Java в облаке. Spring Boot, Spring Cloud, cloud Foundry / Лонг Джош, Бастини Кеннет – СПб.: Питер, 2019. 624 с.
- 11 Spring Boot 2: лучшие практики для профессионалов / Гутьеррес Фелипе – СПб.: Питер, 2020. 464 с.
- 12 Gradle - что это за система и как работает с Java: [Электронный ресурс] URL: blog.skillfactory.ru/glossary/gradle/(дата обращения: 04.04.2022). – Загл. с экрана. – Яз.рус.