**Отчет по лабораторной работе: Тестирование проекта "Онлайн автосервис на Spring Boot"**

**a. Цель работы**

Целью данной работы является создание тест-плана для проекта "Онлайн автосервис на Spring Boot", выполнение выбранных тест-кейсов и предоставление отчета о результатах тестирования. Итогом станет выявление дефектов и разработка рекомендаций по улучшению качества программного обеспечения.

**b. Тест-план**

**1. Введение**

Тест-план охватывает тестирование веб-приложения "Онлайн автосервис на Spring Boot", включая функциональное и нефункциональное тестирование. Основные цели тестирования: проверка корректности работы основных функций, оценка производительности и безопасности системы.

**2. Область применения**

Тестирование будет проводиться для следующих модулей:

* Регистрация и аутентификация пользователей.
* Управление заказами.
* API для взаимодействия с мобильными приложениями.

**3. Виды тестирования**

* **Функциональное тестирование**: проверка соответствия функциональных требований.
* **Нефункциональное тестирование**: тестирование производительности и безопасности.

**4. Уровни тестирования**

* **Модульное тестирование**: проверка отдельных компонентов системы.
* **Интеграционное тестирование**: проверка взаимодействия между модулями.
* **Системное тестирование**: полное тестирование всей системы.
* **Приемочное тестирование**: проверка выполнения требований заказчика.

**5. Тест-кейсы**

**Тест-кейс 1: Регистрация нового пользователя**

* **ID**: TC001
* **Описание**: Проверка успешной регистрации нового пользователя.
* **Шаги**:
  1. Перейти на страницу регистрации.
  2. Ввести валидные данные (имя, email, пароль).
  3. Нажать кнопку "Зарегистрироваться".
* **Ожидаемый результат**: Пользователь успешно зарегистрирован и перенаправлен на главную страницу.

**Тест-кейс 2: Авторизация пользователя**

* **ID**: TC002
* **Описание**: Проверка успешной авторизации зарегистрированного пользователя.
* **Шаги**:
  1. Перейти на страницу авторизации.
  2. Ввести валидные данные (email, пароль).
  3. Нажать кнопку "Войти".
* **Ожидаемый результат**: Пользователь успешно авторизован и перенаправлен на главную страницу.

**Тест-кейс 3: Создание нового заказа**

* **ID**: TC003
* **Описание**: Проверка возможности создания нового заказа.
* **Шаги**:
  1. Авторизоваться в системе.
  2. Перейти в раздел "Создать заказ".
  3. Ввести данные заказа (тип услуги, дата, комментарии).
  4. Нажать кнопку "Создать".
* **Ожидаемый результат**: Заказ успешно создан и отображается в списке заказов.

**Тест-кейс 4: Тестирование производительности при 100 одновременных пользователях**

* **ID**: TC004
* **Описание**: Проверка производительности системы при высокой нагрузке.
* **Шаги**:
  1. Использовать инструмент для нагрузочного тестирования (например, JMeter).
  2. Настроить сценарий для 100 одновременных пользователей, выполняющих авторизацию и создание заказа.
* **Ожидаемый результат**: Система справляется с нагрузкой, время отклика не превышает допустимых значений.

**Тест-кейс 5: Проверка уязвимостей к SQL-инъекциям**

* **ID**: TC005
* **Описание**: Проверка защищенности системы от SQL-инъекций.
* **Шаги**:
  1. Ввести в поля ввода на странице авторизации вредоносный SQL-код.
  2. Нажать кнопку "Войти".
* **Ожидаемый результат**: Система блокирует вредоносный запрос, ошибка не возникает.

**c. Отчет по тестированию**

**Тест-кейс 1: Регистрация нового пользователя**

* **Результат**: Успешно.
* **Примечание**: Пользователь успешно зарегистрирован и перенаправлен на главную страницу.

**Тест-кейс 2: Авторизация пользователя**

* **Результат**: Успешно.
* **Примечание**: Пользователь успешно авторизован и перенаправлен на главную страницу.

**Тест-кейс 3: Создание нового заказа**

* **Результат**: Успешно.
* **Примечание**: Заказ успешно создан и отображается в списке заказов.

**Тест-кейс 4: Тестирование производительности при 100 одновременных пользователях**

* **Результат**: Неуспешно.
* **Примечание**: Система замедляется при 100 одновременных пользователях, время отклика превышает допустимые значения.

**Тест-кейс 5: Проверка уязвимостей к SQL-инъекциям**

* **Результат**: Успешно.
* **Примечание**: Система успешно блокирует вредоносный запрос, ошибки не возникает.

**d. Список выявленных дефектов**

1. **Дефект 1**:
   * **ID**: DEF001
   * **Описание**: Система замедляется при 100 одновременных пользователях.
   * **Рекомендация**: Оптимизировать обработку запросов и базу данных для улучшения производительности.

**e. Выводы по работе**

Проведенное тестирование показало, что большинство функциональных аспектов проекта "Онлайн автосервис на Spring Boot" работают корректно. Однако были выявлены проблемы с производительностью при высокой нагрузке. Рекомендуется провести оптимизацию системы для улучшения производительности и предотвращения замедлений при большом количестве пользователей. Тестирование безопасности и функциональности показало удовлетворительные результаты, что подтверждает готовность системы к использованию.

**f. Список использованных источников**

1. Официальная документация Spring Boot: <https://spring.io/projects/spring-boot>
2. Руководство по тестированию ПО: <https://www.istqb.org/>
3. Инструменты для нагрузочного тестирования:
   * Apache JMeter: <https://jmeter.apache.org/>
4. Книги по тестированию ПО:
   * "Совершенное программное обеспечение" (Джеймс Уиттакер)
   * "Тестирование DOT COM приложений" (Роман Савин)