# Тема работы

Работа с классификацией видов тестирования

# Цель работы

Научиться классифицировать виды тестирования на примере выбранного программного продукта.

# Описание программного проекта

Веб-приложение платной клиники.

Само приложение написано мной в качестве курсовой работы (<https://github.com/Redegit/Coursework-PaidClinic> - тут есть более развернутое описание с картинками)

# Классификация тестирования

* 1. Функциональное

Данный вид тестирования необходим. Для веб приложения однозначно нужна проверка основных функций системы, включая запись на прием, просмотр истории посещений, функциональность админ-панели.

Для функционального тестирования можно использовать как ручные тесты, так и автотесты, включающие в себя юнит-тесты, интеграционное тестирование и тестирование интерфейса (через тот же Selenium)

* 1. Навигация

Тоже, проверка навигации в приложении, корректность маршрутизации и переходов между страницами необходима.

* 1. Инсталляция

Тестирование установки и развертывания приложения, проверка конфигурации базы данных и серверной части тоже необходимо

* 1. Нагрузочное тестирование

Однозначно нужно, система подразумевает одновременное использование множеством клиентов.

* 1. Ресурсные тесты

Хотя система не является критически важной, для бизнеса она все-таки важна, поэтому ресурсные тесты надо проводить.

* 1. Юзабилити
  2. UI/GUI
  3. Локализация

Не нужно, так как приложение не предполагает локализацию.

* 1. Безопасность

Тестирование безопасности можно выполнять при помощи интеграционного тестирования.

* 1. Конфигурация
  2. Документация

Выше уже были упомянуты некоторые уровни тестирования, но сейчас более подробно

1. Модульное тестирование (Unit Testing)

◦ Тестирование отдельных функций или методов, особенно если они содержат сложную логику.

2. Интеграционное тестирование (Integration Testing)

◦ Проверка взаимодействия между различными модулями приложения, в данном случае необходимо так как архитектура клиент-серверная.

Главным видом тестирования для данной системы является функциональное, так как для работы системы очень важна правильная работа каждого отдельно взятого компонента и сбой в одном повлечет за собой ошибки и в других. Ущерб от сбоя можно было бы минимизировать, используя микросервисную архитектуру, но она не использовалась

# Примеры дефектов, характерных для каждого вида тестирования

В ходе тестирования можно выявить следующие дефекты:

## Функциональное тестирование

* Запись на прием не сохраняется: Пользователь заполняет форму для записи на прием, но данные не сохраняются в базе данных.
* История посещений не отображается: Пользователь не может просмотреть историю своих посещений, даже если они есть в базе данных.
* Админ-панель не сохраняет изменения: Администратор вносит изменения в таблицу, но они не сохраняются.

## Навигационное тестирование

* Неправильные маршруты: При переходе по определенной ссылке пользователь попадает не на ту страницу.
* Неработающие кнопки: Кнопки навигации не выполняют никаких действий при нажатии.
* Ломается навигация при использовании браузерных кнопок назад/вперед: Переходы не соответствуют ожидаемым действиям.

## Нагрузочное тестирование

* Замедление работы под нагрузкой: При большом количестве одновременных пользователей сервер начинает работать медленно.
* Падение сервера под высокой нагрузкой: Сервер не справляется с пиковыми нагрузками и перестает отвечать.
* Ошибки в базе данных под нагрузкой: База данных выдает ошибки при множественных одновременных запросах.

## Тестирование безопасности

* SQL-инъекции: Возможность выполнения произвольных SQL-запросов через поля ввода.
* Неправильная авторизация: Пользователи могут получить доступ к функциям, на которые у них нет прав.
* Утечка данных: Личные данные пользователей могут быть доступны или изменены без авторизации.

## Тестирование совместимости

* Некорректная работа в различных браузерах: Приложение работает корректно в одном браузере, но с ошибками в другом.
* Неадаптивный дизайн: Интерфейс некорректно отображается на разных устройствах (например, мобильных).
* Проблемы с версиями: Приложение не работает или работает некорректно на старых версиях операционных систем или браузеров

# Выводы по работе

Применение различных видов тестов позволяет провести комплексную оценку качества, а также выявить слабые места в приложении. Каждый из видов тестирования позволяет выявить свойственные именно ему дефекты и уязвимости.

**Список использованных источников**

* Мой проект с GitHub: <https://github.com/Redegit/Coursework-PaidClinic>