**Отчет по лабораторной работе 2.2.**

**Калинин Павел ПИ21-2**

**Тема: Классификация видов тестирования**

**Цель работы**

Цель данной лабораторной работы заключается в изучении и применении различных видов тестирования для контроля качества программного обеспечения. Понимание классификации видов тестирования позволяет выбрать подходящие процедуры и инструменты, оценить трудоемкость и спланировать тестирование.

**Описание программного проекта**

Программный проект, используемый для данной лабораторной работы, представляет собой веб-приложение системы управления отелем. Основные функции включают регистрацию пользователей, бронирование номеров, управление персоналом и обработку платежей.

**Классификация видов тестирования**

Уровни тестирования

1. Компонентное (модульное) тестирование:

* Тестирование отдельных модулей, таких как форма регистрации, модуль бронирования и модуль обработки платежей.
* Пример дефекта: Ошибка валидации email в форме регистрации.

1. Интеграционное тестирование:

* Проверка взаимодействия между модулями, такими как регистрация и бронирование.
* Пример дефекта: Неверная передача данных между формой регистрации и системой бронирования.

1. Системное тестирование:

* Проверка всей системы на соответствие функциональным и нефункциональным требованиям.
* Пример дефекта: Непредусмотренные комбинации данных, приводящие к сбоям в системе.

Типы тестирования

1. Функциональное тестирование:

* Проверка реализации функциональности, такой как регистрация пользователей и бронирование номеров.
* Пример дефекта: Неудачная попытка регистрации при корректных данных.

1. Тестирование безопасности:

* Проверка защиты данных пользователей и платежных операций.
* Пример дефекта: Уязвимость SQL-инъекции в форме входа.

1. Нагрузочное тестирование:

* Проверка производительности системы при высоких нагрузках.
* Пример дефекта: Замедление отклика системы при большом количестве одновременных пользователей.

1. Стресс-тестирование:

* Проверка работы системы в экстремальных условиях.
* Пример дефекта: Сбой системы при переполнении базы данных.

1. Тестирование удобства пользования (юзабилити):

* Оценка удобства интерфейса для пользователей.
* Пример дефекта: Сложность навигации по сайту для новых пользователей.

Методы тестирования

1. Методы белого ящика:

* Тестирование с учетом внутренней структуры кода.
* Пример дефекта: Недостаточная проверка условий в коде, приводящая к ошибкам.

1. Методы черного ящика:

* Тестирование функциональности без знания внутренней структуры системы.
* Пример дефекта: Некорректный ответ системы на ввод неверных данных.

1. Методы серого ящика:

* Комбинация методов белого и черного ящика.
* Пример дефекта: Частичный доступ к коду позволяет выявить недостатки интеграции.

Виды тестирования

1. Функциональное тестирование:

* Проверка соответствия системы требованиям.
* Пример дефекта: Некорректная работа функции бронирования номеров.

1. Навигационное тестирование:

* Проверка корректности работы навигации на сайте.
* Пример дефекта: Кнопка "назад" не возвращает пользователя на предыдущую страницу.

1. Инсталляционное тестирование:

* Проверка процесса установки и первоначальной настройки системы.
* Пример дефекта: Ошибка установки программы на определенной операционной системе.

1. Юзабилити тестирование:

* Проверка удобства использования интерфейса.
* Пример дефекта: Непонятные инструкции на странице регистрации.

1. Тестирование безопасности:

* Проверка устойчивости системы к различным угрозам.
* Пример дефекта: Возможность доступа к личным данным без авторизации.

1. Совместимость:

* Проверка работы на различных операционных системах и браузерах.
* Пример дефекта: Неправильное отображение страницы в браузере Internet Explorer.

**Выводы по работе**

В ходе лабораторной работы был проведен анализ различных видов, типов, методов и уровней тестирования, применимых для веб-приложения системы управления отелем. Примеры дефектов для каждого вида тестирования помогли понять возможные проблемы и их локализацию. Классификация видов тестирования позволяет более эффективно планировать и проводить тестирование, что в итоге улучшает качество программного обеспечения.