**Управление качеством программных систем**

**Мерзлова Анастасия, ПИ21-2**

**Лабораторная 5**

*Ручное тестирование*

## Цель работы

Целью данной лабораторной работы является проведение ручного функционального тестирования приложения "Приют для животных" на основе ранее разработанной тестовой документации. Задачи включают формирование тест-плана, выполнение тест-кейсов, документирование процесса тестирования и выявленных дефектов, а также формирование выводов по результатам тестирования.

## Тест-план

Техническое задание: Приложение для управления приютом для животных, включающее функции добавления, редактирования и удаления животных, регистрацию и авторизацию пользователей.

Задачи/Функциональность, которая должна быть протестирована:

- Добавление животного

- Редактирование животного

- Удаление животного

- Регистрация пользователя

- Авторизация пользователя

Виды проводимого тестирования:

- Smoke-тестирование

- Тестирование навигации

- Тестирование ввода данных

- Тестирование бизнес-логики

Список тестовой документации (тест-кейсы):

- TC 1.1: Добавление животного

- TC 1.2: Редактирование животного

- TC 1.3: Удаление животного

- TC 2.1: Регистрация пользователя

- TC 2.2: Авторизация пользователя

Список инструментов тестирования:

- Веб-браузер

- Инструменты для разработчиков (консоль браузера)

- Среда разработки (IDE)

Сервер: http://localhost:8080 (локальный сервер для тестирования)

Оценка риска:

- Возможные ошибки в работе с формами

## Отчет по тестированию

Тест-кейс 1.1: Добавление животного

- Предусловия: Пользователь на странице «Пристроить питомца».

- Шаги:

1. Заполнить форму с данными нового животного.

2. Нажать кнопку "Добавить животное".

- Ожидаемый результат: Животное успешно добавлено.

- Результат выполнения: Успешно.

Тест-кейс 1.2: Редактирование животного

- Предусловия: Пользователь на странице управления животными с существующим животным.

- Шаги:

1. Выбрать животное для редактирования, нажав на кнопку «Подробнее о животном».

2. Нажать кнопку «Изменить информацию о животном».

3. Изменить одно или несколько полей.

4. Нажать кнопку «Обновить информацию».

- Ожидаемый результат: Изменения сохранены успешно.

- Результат выполнения: Успешно.

Тест-кейс 1.3: Удаление животного

- Предусловия: Пользователь на странице управления животными с существующим животным.

- Шаги:

1. Выбрать животное для удаления, нажав на кнопку «Подробнее о животном».

2. Нажать кнопку "Удалить животное".

- Ожидаемый результат: Животное успешно удалено.

- Результат выполнения: Успешно.

Тест-кейс 2.1: Регистрация пользователя

- Предусловия: Пользователь на странице регистрации.

- Шаги:

1. Заполнить форму регистрации. Вводить уникальный номер телефона. Обязательные поля помечены .

2. Нажать кнопку «Зарегистрироваться».

- Ожидаемый результат: Пользователь успешно добавлен, исключения обработаны.

- Результат выполнения: Успешно.

Тест-кейс 2.2: Авторизация пользователя

- Предусловия: Пользователь на странице входа.

- Шаги:

1. Заполнить форму входа, состоящую из номера телефона и пароля, уже хранящихся в базе данных.

2. Нажать кнопку «Войти».

- Ожидаемый результат: Существующий пользователь успешно авторизован, на несуществующего возникает корректная обработка исключения «Такого пользователя не существует».

- Результат выполнения: Успешно.

## Список выявленных дефектов

Дефект 1:

- Краткое описание: Некорректное отображение кнопки "Обновить информацию" на странице редактирования животного.

- Серьезность: Незначительная (minor)

- Приоритет: Средний

- Шаги к воспроизведению:

1. Перейти на страницу редактирования животного.

2. Изменить одно или несколько полей.

3. Наблюдать за отображением кнопки.

- Результат: Кнопка частично скрыта.

- Ожидаемый результат: Кнопка отображается корректно.

Дефект 2:

- Краткое описание: Ошибка "500 Internal Server Error" при удалении пользователя.

- Серьезность: Критическая (critical)

- Приоритет: Высокий

- Шаги к воспроизведению:

1. Перейти на страницу пользователя.

2. Нажать кнопку «Удалить пользователя».

- Результат: Ошибка "500 Internal Server Error".

- Ожидаемый результат: Пользователь успешно удален.

## Выводы по работе

В ходе тестирования приложения "Приют для животных" были выявлены несколько дефектов, которые необходимо исправить для обеспечения стабильной работы системы. Тест-кейсы, выполненные в рамках данного тест-плана, успешно покрыли основные функциональные области приложения, такие как добавление, редактирование и удаление животных, регистрация и авторизация пользователей. Результаты тестирования показали, что большинство функций работают корректно, однако существует несколько проблем, требующих внимания.

## Контрольные вопросы

1. Какие виды тестирования можно выполнять в ручном режиме?

В ручном режиме можно выполнять следующие виды тестирования:

- Функциональное тестирование: Проверка функциональности приложения на соответствие требованиям.

- Регрессионное тестирование: Проверка, что изменения в коде не вызвали новых дефектов в уже проверенной функциональности.

- Smoke-тестирование: Быстрая проверка основных функций приложения, чтобы убедиться, что система работает и готова к более подробному тестированию.

- Тестирование юзабилити: Проверка удобства использования интерфейса и взаимодействия пользователя с системой.

- Системное тестирование: Проверка всей системы в целом, включая интеграцию всех модулей и взаимодействие с внешними системами.

- Приемочное тестирование: Проверка, что система соответствует критериям приемки и готова к передаче заказчику или пользователям.

- Тестирование безопасности: Проверка системы на уязвимости и обеспечение защиты данных.

- Тестирование на совместимость: Проверка работы системы в разных средах (операционные системы, браузеры, устройства).

- Тестирование производительности: Оценка работы системы под нагрузкой и ее способности справляться с большим количеством запросов.

2. Какая документация участвует в ручном тестировании?

В ручном тестировании участвует следующая документация:

- Тест-план: Описание целей, объема, подхода и фокуса тестирования.

- Тестовые сценарии (тест-кейсы): Описания шагов, которые необходимо выполнить для проверки определенной функциональности.

- Тестовые наборы (тест-сьюты): Группы тест-кейсов, объединенные для проверки определенного модуля или функции.

- Тестовая стратегия: Общее руководство по подходам и методологиям, которые будут использованы для тестирования.

- Отчеты о тестировании: Документация результатов тестирования, включая информацию о пройденных и проваленных тестах.

- Отчеты о дефектах (баг-репорты): Документирование найденных дефектов с подробным описанием и шагами для воспроизведения.

- Чек-листы: Списки проверок, которые необходимо выполнить в рамках тестирования.

3. Может ли существовать программная система, свободная от дефектов?

Абсолютно свободная от дефектов программная система практически невозможна по следующим причинам:

- Сложность и масштаб: Современные системы часто сложны и многокомпонентны, что увеличивает вероятность наличия дефектов.

- Человеческий фактор: Программное обеспечение создается людьми, и человеческие ошибки неизбежны.

- Изменения и обновления: Системы постоянно эволюционируют и изменяются, что может вводить новые дефекты.

- Различные окружения: Программное обеспечение может работать по-разному в различных средах, что может привести к дефектам, не выявленным в других условиях.

4. Каковы требования к описанию дефекта?

Описание дефекта должно включать следующие элементы:

- Краткое описание: Текстовое описание дефекта, поясняющее его суть.

- Серьезность: Уровень воздействия дефекта на работу системы (блокирующая, критическая, значительная, незначительная, тривиальная).

- Приоритет: Очередность исправления дефекта (высокий, средний, низкий).

- Шаги к воспроизведению: Последовательность действий, необходимых для воспроизведения дефекта.

- Результат: Полученное состояние системы после выполнения шагов к воспроизведению.

- Ожидаемый результат: Эталонное состояние системы, которое ожидалось после выполнения шагов.

- Дополнительные материалы: Скриншоты, видео, логи и другие материалы, помогающие в воспроизведении и понимании дефекта.

5. Каков жизненный цикл дефекта?

Жизненный цикл дефекта включает следующие этапы:

1. Идентификация: Обнаружение и документирование дефекта.

2. Назначение: Присвоение дефекта ответственному лицу (разработчику).

3. Исправление: Внесение изменений в код для устранения дефекта.

4. Тестирование: Проверка исправленного кода, чтобы убедиться в устранении дефекта и отсутствии новых проблем.

5. Закрытие: Завершение работы над дефектом и его закрытие в системе отслеживания.

6. Повторное открытие: В случае, если дефект не был устранен, он может быть снова открыт для дальнейшего исправления.