**Самостоятельная работа по теме: Тестовый сценарий и тестовый пакет**

**1. Теоретическая часть:**

**Определения**

Тестовый сценарий — это документ, который описывает последовательность действий, необходимых для выполнения тестирования определённой функции или компонента системы

Тестовый пакет — это группа тестов, которые объединены по какому-либо признаку, например, по функциональности или области тестируемого приложения.

Позитивный тест — это тест, который проверяет, что система ведет себя правильно при корректных входных данных.

Негативный тест — это тест, который проверяет, как система реагирует на некорректные или невалидные входные данные.

Нефункциональное тестирование — это тип тестирования, который оценивает характеристики системы, такие как производительность, безопасность, удобство использования и надежность.

**Пример тестового сценария:**

Идентификатор: TS-001

Название: Регистрация нового пользователя

Описание:

Тестовый сценарий направлен на проверку процесса регистрации нового пользователя в системе.

Предусловия:

- Пользователь должен иметь доступ к веб-приложению.

- Пользователь должен быть на странице регистрации.

Шаги выполнения:

1. Открыть веб-приложение.

2. Перейти на страницу регистрации.

3. Ввести корректные данные в поля:

* Имя: "Иван"
* Фамилия: "Петров"
* Электронная почта: "petrov@example.com"
* Пароль: "Password123&"
* Подтверждение пароля: "Password123&"

4. Нажать кнопку "Зарегистрироваться".

После выполнения шагов регистрации пользователь успешно зарегистрирован, и система отображает сообщение об успешной регистрации. Пользователь перенаправляется на страницу приветствия или личного кабинета.

**Структура тестового пакета:**

Тестовый пакет для модуля «Корзина покупок»:

Тестовый сценарий 1:

1. Добавление товара в корзину
2. Товар успешно добавлен в корзину, и количество товаров отображается корректно на иконке корзины.

Тестовый сценарий 2:

1. Удаление товара из корзины
2. Товар успешно удален из корзины, и общее количество товаров уменьшилось на 1.

Тестовый сценарий 3:

1. Проверка корректности расчета общей стоимости
2. Общая стоимость корзины корректно рассчитывается исходя из добавленных товаров и их цен, включая возможные скидки или налоги.

**2. Практическая часть:**

Тестовые сценарии для раздела «Профиль пользователя» в социальном приложении:

1. Проверка редактирования профиля:
   * Изменение имени и фамилии: Проверка правильности обновления данных в профиле пользователя.
   * Настройки личных данных: Возможность редактирования других данных (например, адрес, номер телефона, биография) и их корректное отображение после изменений.
   * Проверка валидации полей ввода: Убедитесь, что введенные данные соответствуют требованиям (например, минимальная/максимальная длина, отсутствие специальных символов, если это необходимо).
   * Отображение уведомлений об успешном или неуспешном редактировании профиля.
2. Проверка загрузки аватара:
   * Поддержка форматов файлов: Проверка, что система поддерживает стандартные форматы изображений (например, JPG, PNG, GIF) и что загрузка правильного формата проходит успешно.
   * Ограничения по размеру: Проверка, что загружаемые изображения соответствуют установленным ограничениям по размеру (например, не более 2 Мб) и что система корректно обрабатывает и отклоняет файлы, превышающие этот лимит.
   * Обновление аватара: Убедиться, что новый аватар отображается на странице профиля после загрузки.
3. Проверка обработки ошибок при загрузке изображения неподдерживаемого формата:
   * Проверка, что система корректно реагирует на попытку загрузки неподдерживаемого формата изображения (например, файлы типа BMP или TIFF).
   * Убедиться, что пользователю показывается чёткое и понятное сообщение об ошибке, объясняющее причину (например, "Неподдерживаемый формат файла. Пожалуйста, загрузите JPG или PNG.").
   * Проверить, что неподдерживаемые файлы не загружаются в систему и не влияют на другие функции.

**3. Дополнительное задание**

Отслеживание покрытия тестами позволяет не только контролировать качество кода, но и обеспечивает долгосрочную стабильность и надежность приложения, что критически важно в условиях современного рынка. Это также позволяет сократить время на отладку, минимизировать количество ошибок и улучшить качество кода.

Примеры недостаточного покрытия тестами и его влияние на качество ПО

1. Непокрытые сценарии ошибок: если в коде не протестированы обработка исключений или ошибки, то в продуктивной среде это может привести к сбоям приложения, которые трудно диагностировать. Например, в случае некорректного ввода данных пользователем приложение может упадать, не предоставляя адекватной обратной связи.
2. Неполное тестирование бизнес-логики: если некоторые ветви бизнес-логики не покрыты тестами, это может привести к тому, что критические условия не будут протестированы. Например, если забыть протестировать определенные параметры при расчете скидок, покупатели могут получать неверные цены, что негативно скажется на репутации компании.
3. Отсутствие тестов для интеграции с внешними сервисами: недостаточное покрытие тестами может привести к проблемам при взаимодействии с внешними API, что может быть критично, если эти сервисы не работают как ожидалось. Например, если приложение требует интеграции с платежной системой, а тесты не учитывают различные сценарии, приложения могут сталкиваться с проблемами приема платежей.
4. Проблемы с производительностью: если тестирование производительности не охватывает все возможные случаи, в том числе пиковые нагрузки, то приложение может не справляться с нагрузкой в момент актуальности, что сильно подорвет доверие к продукту.

**4. Аналитическая часть:**

Недостаточное тестирование функций аутентификации и авторизации может привести к различным серьезным рискам и проблемам, к таким как:

1. Риск несанкционированного доступа: если функции аутентификации недостаточно протестированы, злоумышленники могут легко получить доступ к учетным записям пользователей или администраторов.
2. Нарушение конфиденциальности: слабости в аутентификации могут позволить злоумышленникам получить доступ к личной информации пользователей, что может привести к утечкам конфиденциальной информации, таким как адреса, телефонные номера и финансовые данные.
3. Модификация данных: если системы авторизации не защищают действия пользователей, злоумышленники могут изменять или удалять данные, что отрицательно скажется на целостности информации.
4. Атаки с использованием компонентов ПО: некорректно реализованные функции аутентификации могут быть уязвимы к различным видам атак, позволяя злоумышленникам извлекать данные или получать доступ к ресурсам.
5. Репутационные и финансовые потери компании после утечки данных или инцидента безопасности, связанные с регулированием, восстановлением системы и удалением уязвимостей.
6. Пользователи могут потерять доверие к компании, если их данные будут скомпрометированы, что приведет к снижению пользовательской базы и негативным отзывам.

Примеры проблем в случае недостаточного тестирования:

1. Злоумышленник может легко протестировать множество комбинаций, получая доступ к учетным записям пользователей, если функция аутентификации не защищена от перебора паролей,
2. Если реализация проверок прав доступа не протестирована, обычные пользователи могут случайно получить доступ к администраторским функциям, позволяя им выполнять операции, которые могут повредить системе.
3. Если сессионные токены не завершаются корректно, злоумышленник может использовать старые токены для доступа к системе, даже если пользователь вышел из своей учетной записи.
4. Злоумышленник может получить подсказки по логину, если вход пользователя в систему не защищен, что упростит процесс несанкционированного доступа.

**5. Заключение:**

Таким образом, можно сделать вывод, что тестовые сценарии и тестовые пакеты играют жизненно важную роль в процессе тестирования программного обеспечения, обеспечивая структурированный и эффективный подход к выявлению ошибок и проверке функциональности системы. Они помогают не только в выявлении дефектов на ранних стадиях, но и в обеспечении надежности конечного продукта.