Самостоятельная работа по теме: Тестовый сценарий и тестовый пакет:

**Климентов Иван**

Задания:

1.Теорические часть

Определения:

· Тестовый сценарий: Это документ, содержащий последовательность шагов для тестирования определенной функции или компонента программного обеспечения. Он может описывать действия, которые необходимо выполнить, входные данные и ожидаемые результаты.

· Тестовый пакет: Это набор тестовых сценариев, которые сгруппированы для удобства выполнения и управления процессом тестирования. Тестовый пакет может включать как позитивные, так и негативные тесты и используется для проверки определенной функциональности или модуля программного обеспечения.

· Позитивный тест: Это тест, который проверяет правильность и стабильность работы системы при корректных входных данных. Позитивные тесты подтверждают, что функция работает как задумано.

· Негативный тест: Это тест, который направлен на проверку поведения системы при некорректных или неожиданно проблемных входных данных. Негативные тесты помогают выявить ошибки и отсутствие обработчиков исключений.

· Нефункциональное тестирование: Это тип тестирования, который фокусируется на немаловажных аспектах, таких как производительность, безопасность, удобство использования и совместимость. Нефункциональное тестирование проверяет, как софт ведет себя в различных условиях, а не только его функциональные возможности.

Пример тестового сценария:

Тестовый сценарий по регистрации нового пользователя

Идентификатор: TS-001

Название: Регистрация нового пользователя с корректными данными

Описание: Проверка возможности успешной регистрации нового пользователя с корректными данными

Предусловия: Пользователь находится на странице регистрации и не авторизован в системе

Шаги выполнения:

Перейти на страницу регистрации.

Заполнить поле "Имя" допустимым значением.

Заполнить поле "Email" допустимым значением.

Заполнить поле "Пароль" допустимым значением.

Нажать кнопку "Зарегистрироваться".

Ожидаемый результат: Появляется сообщение о успешной регистрации, и пользователь перенаправляется на страницу профиля.

Структура тестового пакета для модуля 'Корзина покупок':

Тестовый пакет: Проверка функциональности Корзины покупок

Тестовый сценарий 1: Добавление товара в корзину

Ожидаемый результат: Товар успешно добавлен в корзину, отображается уведомление о добавлении и обновляется количество товаров в корзине.

Тестовый сценарий 2: Удаление товара из корзины

Ожидаемый результат: Товар успешно удален из корзины, отображается уведомление об успешном удалении и обновляется количество товаров в корзине.

Тестовый сценарий 3: Оформление заказа из корзины

Ожидаемый результат: Пользователь успешно переходит на страницу оформления заказа, все товары из корзины отображаются с корректными ценами и количеством.

2. Практическая часть:

Для раздела 'Профиль пользователя' в социальном приложении необходимо провести различные виды тестов, чтобы убедиться в корректной работе функциональности и удобстве использования. Рассмотрим аспекты, которые будут проверяться в следующих сценариях:

1. Проверка редактирования профиля

Аспекты функциональности для проверки:

Изменение имени и фамилии: Проверка, что пользователь может успешно изменить имя и фамилию. Это включает в себя тестирование ввода данных, а также проверку на отображение обновленной информации после сохранения изменений.

Настройка личных данных: Проверка возможности редактирования других личных данных (дата рождения, пол, биография и т. д.). Это включает тестирование корректной обработки вводимых данных, таких как длина текста, допустимые символы и т. д.

Отображение изменений: Проверка, что изменения в профиле отображаются корректно после обновления страницы или при переходе на страницу профиля.

Сохранение изменений: Проверка, что все данные сохраняются корректно и могут быть восстановлены после их редактирования.

Валидация данных: Проверка, что система корректно обрабатывает некорректные данные (например, специальные символы, слишком короткие или длинные строки).

2. Проверка загрузки аватара

Аспекты функциональности для проверки:

Поддержка форматов файлов: Проверка, что система правильно обрабатывает загрузку различных форматов файлов (JPEG, PNG, GIF и т.д.). Это может включать позитивные тесты для корректных форматов и негативные тесты для неподдерживаемых форматов.

Ограничения по размеру: Проверка, что система обрабатывает изображения с различными размерами. Это включает тестирование загрузки изображений, превышающих допустимый размер (например, 5 МБ), чтобы убедиться, что система выдает соответствующее сообщение об ошибке.

Отображение загруженного аватара: Проверка, что после успешной загрузки аватара он отображается корректно в профиле пользователя.

Свойства изображения: Проверка того, что загруженные изображения соответствуют параметрам (размер, разрешение и т.д.), заданным приложением.

3. Проверка обработки ошибок при загрузке изображения неподдерживаемого формата

Аспекты функциональности для проверки:

Обработка ошибок: Проверка, что пользователю выводится ясное и информативное сообщение об ошибке при попытке загрузить изображение в неподдерживаемом формате (например, BMP, TIFF и т.д.).

Ограничение выбора файлов: Проверка, что диалог загрузки файлов ограничивает выбор пользователей только поддерживаемыми форматами, если это возможно.

Функциональность после ошибки: Проверка, что после возникновения ошибки система остается доступной и функциональной для повторной попытки загрузки изображения.

Эти проверки помогут убедиться, что раздел 'Профиль пользователя' работает корректно, удобно для пользователей и безопасен для данных. Также важно учитывать пользовательский опыт, чтобы каждый аспект взаимодействия был интуитивно понятен и предоставлял необходимые инструкции в случае ошибок или проблем.

3. Дополнительное задание: Анализ покрытия тестами

Отслеживание покрытия тестами — это критически важный аспект процесса тестирования, который помогает определить, насколько полно тестируемая программа или система охвачена тестами. Возможности отслеживания покрытия включают как функциональные, так и нефункциональные аспекты, что позволяет разработчикам и тестировщикам оценить качественное выполнение тестов и принять информированные решения на основе полученных данных.

Почему важно отслеживать покрытие тестами?

Выявление не протестированных участков кода: При помощи анализа покрытия тестами можно обнаружить участки кода, которые не были протестированы. Это помогает предотвратить ситуации, когда ошибки остаются незамеченными, что может привести к сбоям или неправильной работе программного обеспечения.

Улучшение качества программного обеспечения: Высокое покрытие тестами обычно коррелирует с высокой надежностью и качеством ПО. Поэтому регулярный анализ покрытия может помочь снизить уровень ошибок и дефектов в продуктах.

Оптимизация ресурсов: С помощью анализа покрытия можно оптимизировать процесс автоматизированного тестирования, сосредоточив внимание на наиболее уязвимых местах и избегая избыточного тестирования ненужных участков.

Повышение уверенности в релизе: Зная, что большинство кода было протестировано, команды разработки могут с большей уверенностью выпускать новые версии программного обеспечения.

Примеры влияния недостаточного покрытия на качество программного обеспечения:

Недоработанные сценарии: Если тесты не покрывают определенные функциональные возможности, такие как крайние случаи или редкие сценарии использования, это может привести к незамеченным ошибкам. Например, если не протестировать возможность заказа товара с отрицательным количеством, пользователь может столкнуться с опасной уязвимостью.

Ошибки производительности: Неполное покрытие также может привести к ошибкам в производительности системы. Например, если не протестировать систему при максимальной нагрузке, она может не выдержать реальных условий эксплуатации, что вызовет сбои во время пиковых нагрузок.

Упущенные требования: Если тесты не проверяют все требования, указанные в спецификациях, некоторые особенности могут остаться невыполненными. Например, если спецификация требует возможности фильтрации данных в отчетах, но тестирование этого не включает, конечный пользователь может столкнуться с отсутствием критически важной функциональности.

Проблемы с безопасностью: Недостаточное покрытие тестами может привести к уязвимостям в безопасности. Например, если не протестированы сценарии, связанные с вводом данных от пользователей, могут возникнуть SQL-инъекции или другие виды атак.

Таким образом, регулярный анализ покрытия тестами и обеспечение достаточного уровня охвата во время тестирования является важным условием для достижения высокого качества, надежности и безопасности программного обеспечения.

4. Аналитическая часть: Анализ рисков, связанных с недостаточным тестированием функций аутентификации и авторизации

Недостаточное тестирование функций аутентификации и авторизации в программном обеспечении может привести к серьезным рискам, которые могут повлиять на безопасность системы и доверие пользователей. Аутентификация - это процесс подтверждения идентичности пользователя, а авторизация - это процесс предоставления разрешений на доступ к ресурсам системы. Вот анализ возможных рисков и проблем, которые могут возникнуть из-за недостаточного тестирования этих функций:

1. Уязвимости в безопасности

Недостаточное тестирование может привести к уязвимостям, которые могут быть использованы злоумышленниками для несанкционированного доступа к системе. Например:

Простые пароли: Если нет тестов на сложность паролей, пользователи могут использовать легкие для угадывания пароли, что упрощает атаки методом подбора.

Уязвимости SQL-инъекций: Неправильное тестирование ввода данных может привести к SQL-инъекциям, при которых злоумышленники могут манипулировать базой данных.

2. Перепутанные роли и разрешения

Неэффективное тестирование может привести к ситуации, когда пользователи имеют доступ к функциям и данным, которые им не должны быть доступны. Например:

Расширенный доступ: Пользователь с ролью «обычного пользователя» может получить доступ к административным функциям, что может привести к нежелательным изменениям или утечке чувствительных данных.

Отсутствие разграничения доступа: Если тестирование не охватывает сценарии, при которых разные пользователи должны иметь разные уровни доступа, это может привести к ошибкам в бизнес-логике и несанкционированному доступу к конфиденциальной информации.

3. Нарушение конфиденциальности данных

Недостаточное тестирование функций аутентификации и авторизации может привести к нарушениям конфиденциальности данных. Например:

Утечка пользовательских данных: Если функции авторизации не функционируют должным образом, злоумышленники могут получить доступ к личным данным пользователей, что приведет к утечке и потере доверия.

Неправильное хранение сессий: Если сервер не проверяет авторизацию на каждом этапе, несанкционированные сессии могут оставаться активными, что позволяет злоумышленникам воспользоваться предыдущими сессиями.

4. Подрыв доверия пользователей

Проблемы с аутентификацией и авторизацией могут негативно повлиять на репутацию компании и доверие пользователей. Например:

Негативный опыт пользователей: Частые проблемы с входом в систему или доступом к функциями могут вызвать недовольство пользователей и побудить их покинуть сервис.

Утрата клиентов: Нарушения конфиденциальности и безопасности данных могут привести к частичной или полной утрате клиентов, особенно в финансовых, медицинских и других секторах, где безопасность данных критически важна.

5. Юридические последствия

Недостаточное тестирование функций аутентификации и авторизации может привести к юридическим последствиям, если будет нарушен закон о защите данных, например:

Штрафы: Компании могут подвергаться крупным штрафам за нарушение правил защиты данных в соответствии с Общим регламентом защиты данных (GDPR) или другими аналогичными законами.

Иски: Потерпевшие пользователи могут подать иски против компании за ненадлежащее управление их данными и безопасность.

Вывод

Правильное и полное тестирование функций аутентификации и авторизации является критически важным для поддержания безопасности, конфиденциальности данных и доверия пользователей. Недостаточное внимание к этим аспектам может привести к серьезным уязвимостям, негативным последствиям и даже к юридическим рискам. Компании должны инвестировать в тщательное тестирование этих функций, чтобы защитить как свои активы, так и данные пользователей.

5. Заключение

Тестовые сценарии и тестовые пакеты играют ключевую роль в процессе тестирования программного обеспечения, обеспечивая систематический и структурированный подход к выявлению дефектов и проверке функциональности системы.

Значение тестовых сценариев:

Тестовые сценарии представляют собой четкие и последовательные шаги, которые тестировщики должны выполнять для проверки конкретной функции или требования. Они обеспечивают:

Стандартизацию тестирования: Тестовые сценарии помогают гарантировать, что тестирование выполняется последовательно, независимо от того, кто его проводит, что значительно снижает риск пропуска важных проверок.

Документирование требований: Они служат документом, который фиксирует, какие функции были проверены и как, что помогает в дальнейшем анализе и понимании тестового покрытия.

Упрощение процесса выявления ошибок: Четко оформленные сценарии позволяют тестировщикам быстро определять, где могла произойти ошибка, и сосредотачиваться на конкретных областях, требующих внимания.

Обеспечение воспроизводимости тестов: Их использование позволяет легко воспроизводить тесты и проверять, были ли исправлены ранее выявленные дефекты.

Значение тестовых пакетов:

Тестовые пакеты, объединяющие несколько тестовых сценариев, также имеют важное значение в процессе тестирования:

Эффективность управления тестированием: Группировка связанных тестов в тестовые пакеты позволяет тестировщикам управлять и организовывать процесс тестирования более эффективно, особенно в крупных проектах с множеством функций.

Упрощение отчетности: Тестовые пакеты предоставляют удобный способ отслеживания статуса тестирования и выявленных дефектов, что позволяет быстро оценивать общую картину качества ПО.

Системный подход: Использование тестовых пакетов помогает обеспечить, что все функциональные области приложения покрыты, и позволяет видеть, какие части приложения уже протестированы, а какие - нет.

Повышение поверенности в релиз: Тестовые пакеты помогают сделать процесс выпуска более надежным, обеспечивая, что все критически важные функции были протестированы перед выпуском новой версии программного обеспечения.

**Вывод**:Таким образом, тестовые сценарии и тестовые пакеты являются необходимыми инструментами в процессе обеспечения качества программного обеспечения. Они не только упрощают процесс тестирования, но и способствуют повышению надежности и безопасности разработанного продукта. Инвестирование времени и ресурсов в их создание и поддержание является важным шагом к достижению высокого уровня качества ПО и удовлетворенности пользователей.