**GeekBrains 2023**

**Архитектура ПО (семинары)**

**Номер урока:**

Урок 12. Принципы тестирования приложений

**Задание:**

Разработать тесты под проект системы управления роботом пылесосом (достаточно один протокол):

1. Предложить и разработать E2E тесты;
2. Предложить и разработать приёмочные тесты (UAT) для сценариев UseСase
3. **\***Разработать Unit Test для разработанного для проекта кода.

Задание со звездочкой ( \* ) - задание по желанию. Не обязательное.

**Формат сдачи домашнего задания:**

Тесты сдаются в виде протокола (pdf, doc, docx, xls) или ссылка на документ на платформах:

* <https://www.postman.com/>
* <https://www.gurock.com/testrail/>

**Выполнил:**

Ман Виталий

**Группа:**

ЦП | Разработчик. Программист | 19.11.22 | 3623

**Выполнение домашней работы 12**

**Описание видов тестирования:**

**E2E (end-to-end) тестирование** — это разновидность функционального тестирования, в ходе которого проверяется функциональность системы от начала до конца, имитируя реальные сценарии использования и взаимодействия со всеми компонентами системы. Оно выполняется для проверки правильности работы системы в целом и выявления возможных проблем, связанных с интеграцией и взаимодействием различных частей системы. E2E тесты помогают убедиться, что весь процесс работы системы, начиная от входных данных или пользовательского взаимодействия и заканчивая выходными результатами, функционирует должным образом.

**UAT (user acceptance testing)** — это вид тестирования, проводимый конечными пользователями или представителями бизнеса для проверки соответствия системы или продукта их требованиям, ожиданиям и бизнес-целям. UAT выполняется после завершения разработки и перед выпуском системы в эксплуатацию. Основная цель UAT - убедиться, что система полностью соответствует ожиданиям пользователей и может быть успешно принята ими для использования. В ходе UAT тестирования пользователи проверяют различные функциональные, операционные и производительностные аспекты системы, а также документацию и инструкции пользователей.

Оба эти тестирования играют важную роль в обеспечении качества программного обеспечения и помогают выявить проблемы, которые могут возникнуть на разных уровнях системы, от интеграции компонентов до соответствия требованиям пользователя.

**Е2Е-тесты**

Таблица 1 - Тестовые сценарии для методов авторизации пользователя

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Описание сценария** | **Шаги** | **Ожидаемый результат** |
| 1. | Авторизация пользователя с корректными данными | 1. Вызвать метод **authorizeUser** с правильным логином и паролем.  2. Проверить возвращенное значение. | Получен валидный токен авторизации |
| 2. | Авторизация пользователя с некорректными данными | 1. Вызвать метод **authorizeUser** с неправильными логином и паролем.  2. Проверить возвращенное значение. | Возвращено сообщение об ошибке авторизации |
| 3. | Проверка валидности токена авторизации | 1. Вызвать метод **checkTokenValidity** с корректным токеном.  2. Проверить возвращенное значение. | Возвращено подтверждение валидности токена |
| 4. | Проверка валидности недействительного токена авторизации | 1. Вызвать метод **checkTokenValidity** с неправильным токеном.  2. Проверить возвращенное значение | Возвращено сообщение об ошибке валидации токена |
| 5. | Выход пользователя | 1. Вызвать метод **logout** с активным токеном.  2. Проверить возвращенное значение. | Пользователь успешно вышел из системы |

Таблица 2 - Тестовые сценарии для методов работы с пользователями

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Описание сценария** | **Шаги** | **Ожидаемый результат** |
| 1. | Получение пользователя по идентификатору | Вызвать метод **getUserById** с существующим идентификатором пользователя.  2. Проверить возвращенное значение. | Получен пользователь с заданным идентификатором |
| 2. | Получение пользователя по имени | 1. Вызвать метод **getUserByName** с существующим именем пользователя.  2. Проверить возвращенное значение. | Получен пользователь с заданным именем |
| 3. | Получение пользователя по логину | 1. Вызвать метод **getUserByLogin** с существующим логином пользователя.  2. Проверить возвращенное значение. | Получен пользователь с заданным логином |
| 4. | Получение пользователя по номеру телефона | 1. Вызвать метод **getUserByPhone** с существующим номером телефона пользователя.  2. Проверить возвращенное значение. | Получен пользователь с заданным номером телефона |
| 5. | Получение пользователя по адресу электронной почты | 1. Вызвать метод **getUserByEmail** с существующим адресом. электронной почты пользователя  2. Проверить возвращенное значение. | Получен пользователь с заданным адресом электронной почты |
| 6. | Создание нового пользователя | 1. Вызвать метод **createUser** с корректными данными нового пользователя.  2. Проверить возвращенное значение. | Создан новый пользователь с заданными данными |
| 7. | Обновление информации о пользователе | 1. Вызвать метод **updateUser** с существующим идентификатором пользователя и новыми данными.  2. Проверить возвращенное значение. | Информация о пользователе успешно обновлена |
| 8. | Удаление пользователя | 1. Вызвать метод **deleteUser** с существующим идентификатором пользователя.  2. Проверить возвращенное значение. | Пользователь успешно удален |

**UAT-тесты**

Таблица 1 - Тестирование мобильного приложения для робота-пылесоса

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Кнопка** | **Шаги** | **Ожидаемый результат** |
| 1. | Сбросить пыль на станцию сбора пыли | 1. Пользователь нажимает кнопку "Сбросить пыль на станцию сбора пыли".  2. Робот-пылесос направляется к станции сбора пыли и опустошает контейнер для пыли. | Содержимое контейнера для пыли робота-пылесоса переносится в контейнер станции сбора пыли |
| 2. | Запуск / Остановка уборки | 1. Пользователь нажимает кнопку "Запуск / Остановка уборки".  2. Робот-пылесос начинает или останавливает уборку. | Робот-пылесос начинает или останавливает уборку в соответствии с выбором пользователя |
| 3. | Начать зарядку | 1. Пользователь нажимает кнопку "Начать зарядку".  2. Робот-пылесос ищет и направляется к станции сбора пыли и заряда. | Робот-пылесос успешно находит станцию сбора пыли и заряда и начинает процесс зарядки |
| 4. | Режим | Пользователь выбирает желаемый режим уборки. | Выбранный пользователем режим уборки активируется на роботе-пылесосе |
| 5. | Карта | Пользователь настраивает параметры карты (например, запретные зоны, виртуальную стену). | Выбранные настройки карты сохраняются и применяются во время уборки |
| 6. | Имя устройства | Пользователь вводит новое имя для устройства. | Имя устройства успешно изменяется на введенное пользователем |
| 7. | Настройка подключения к Wi-Fi | Пользователь вводит данные для подключения к Wi-Fi сети. | Устройство успешно подключается к указанной Wi-Fi сети |
| 8. | Управление общим доступом | Пользователь включает или отключает общий доступ к устройству. | Общий доступ к устройству включается или отключается в соответствии с выбором пользователя |
| 9. | Обновление прошивки | Пользователь запускает процесс обновления прошивки | Прошивка устройства успешно обновляется до последней доступной версии |
| 10. | Справка и отчёт | Пользователь нажимает на кнопку "Справка и отчет" | Открывается окно с информацией и отчетом об использовании устройства |
| 11. | Настройка безопасности | Пользователь выбирает и настраивает параметры безопасности устройства | Выбранные пользователем настройки безопасности устройства сохраняются и применяются |
| 12. | Настройка робота | Пользователь настраивает параметры робота-пылесоса | Выбранные пользователем настройки робота-пылесоса сохраняются и применяются |
| 13. | Настройка станции сбора пыли | Пользователь настраивает параметры станции сбора пыли | Выбранные пользователем настройки робота-пылесоса сохраняются и применяются |