**E2E**-**тестирование**

Сквозное тестирование (End-to-end, E2E, Chain testing) — это вид тестирования, используемый для проверки программного обеспечения от начала до конца, а также его интеграцию с внешними интерфейсами. Цель сквозного тестирования состоит в проверке всего программного обеспечения на предмет зависимостей, целостности данных и связи с другими системами, интерфейсами и базами данных для проверки успешного выполнения полного производственного сценария.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | Описание | Тестируемые методы | Тестируемые таблицы в БД | Входные данные | Ожидаемый результат | Полученный результат | Статус |
| 1 | Регистрация пользователя | setUser(User u, Robot r): Token  showRobotsByUser(): void  loadMaps(int robot\_id): List<int> | После подтверждения email в таблицах «User», «Group» БД создается новая запись. Если создалась новая группа, то создается новая запись в таблицах «Robot», «Token». | Email,  Пароль, серийный номер пылесоса | Пользователь вводит email, пароль и серийный номер пылесоса. При вводе данных в окне ввода осуществляется проверка вводимых данных: email - на соответствие шаблону, пароля - на соответствие правилам безопасности.  После нажатия на кнопку «Регистрация» пользователю отправляется письмо для подтверждения email. После подтверждения email пользователю на устройство передается токен. В приложении открывается окно со списком роботов-пылесосов.  Автоматически синхронизуется информация по картам. |  |  |
| 2 | Авторизация пользователя | getAuthToken(User u): Token  showRobotsByUser(): void  loadMaps(int robot\_id): List<int> |  | Email  Пароль | После нажатия на кнопку «Вход» запускается метод **getAuthToken(User u): Token** для проверки введенных данных. Если клиент не зарегистрирован в системе, то выдается ошибка «Неверный логин или пароль». Если клиент зарегистрирован, то на его устройство передается токен. В приложении открывается окно со списком роботов-пылесосов.  Автоматически синхронизуется информация по картам. |  |  |
| 3 | Восстановление пароля | restorePsw(String email): int  setPsw(String email, int hashPsw): Boolean | Hash нового пароля сохраняется в таблице «User». | Email | При нажатии на кнопку «Восстановить» отправляется письмо для сброса пароля на электронный ящик пользователя. Клиент переходит по ссылке, вводит новый пароль и нажимает «Сохранить». В приложении открывается окно авторизации. |  |  |
| 4 | Запуск/Пауза/Остановка работы робота-пылесоса | checkToken(Token t): boolean  startRobot(int robot\_id): Boolean  pauseRobot(int robot\_id): Boolean  stopRobot(int robot\_id): boolean | В таблице «Робот» меняется статус робота на «working», «paused» или «stopped». | Id робота | При нажатии на кнопку «Запуск», робот-пылесос начинает уборку помещения по установленной схеме помещения и типу уборки.  При нажатии на кнопку «Пауза» робот-пылесос приостанавливает уборку и останавливается на месте, где находился в момент приостановки программы.  При нажатии на кнопку «Остановка» робот-пылесос завершает выполняемую программу уборки и возвращается на место подзарядки. |  |  |
| 5 | Добавление нового робота-пылесоса | searchRobot(int serial\_num): Robot  addRobot(Robot r): Token | Если добавили нового робота-пылесоса, то в таблицах «Робот», «Токен», «Группа» появляется новая запись. Если роботом уже пользовались другие члены семьи, то добавляется запись в таблице «Группа». | Серийный номер пылесоса,  QR-код пылесоса | Клиент вводит в поле поиска серийный номер пылесоса, либо сканирует QR-код пылесоса и нажимает на кнопку «Поиск». Выделяет строку с найденным пылесосом и нажимает на кнопку «Добавить». На приложение пользователя отправляется токен. |  |  |
| 6 | Удаление робота-пылесоса из приложения | deleteRobotByUser(Robot r): boolean | Пользователь удаляется из таблицы «Группа». | Robot\_id | При нажатии на кнопку «Удалить» робот-пылесос перестает отображаться в приложении пользователя. |  |  |
| 7 | Просмотр карточки робота-пылесоса | showRobot(int robot\_id): void |  | Robot\_id | При нажатии на значок пылесоса в окне списка открывается карточка робота-пылесоса в окне «Карточка». |  |  |
| 8 | Изменение названия робота-пылесоса | renameRobot(Robot r): boolean | В таблице «Робот» меняется название пылесоса (поле name). | Robot\_id | При нажатии на кнопку редактирования становится доступным для редактирования поле названия пылесоса. После введенных изменений пользователь нажимает кнопку «Сохранить». |  |  |
| 9 | Добавление воды в контейнер робота-пылесоса | changeWaterLevel(Robot r): boolean | В таблице «Робот» меняется значение поля «water\_level». | Robot\_id | После добавления воды в контейнер индикатор уровня воды в приложении должен измениться, а индикатор на пылесосе перестать мигать. |  |  |
| 10 | Очистка мусорного контейнера | cleanRubbishBag(Robot r): boolean | В таблице «Робот» меняется значение поля «rubbish\_bag». | Robot\_id | После очистки мусорного контейнера индикатор уровня заполнения контейнера в приложении должен измениться, а индикатор на пылесосе перестать мигать. |  |  |
| 11 | Замена щеток | changeBrushes(Robot r): boolean | В таблице «Робот» меняется значение поля «brushes». | Robot\_id | После замены щеток индикатор количества использования щеток в приложении должен измениться, а индикатор на пылесосе перестать мигать. |  |  |
| 12 | Выбор режима работы робота-пылесоса | changeOptions(Robot r, int option): boolean | После выбора нового режима работы пылесоса в таблице «Расписание» меняется идентификатор режима работы пылесоса. | Robot\_id | Пользователь с помощью радиокнопок выбирает необходимый режим работы пылесоса. |  |  |
| 13 | Обновление ПО робота-пылесоса | updatesSW(Robot r): int | В таблицу «Робот» записывается новая версия ПО. | Robot\_id | Клиент нажимает на кнопку «Обновить» и получает статус обновления «Обновление ПО не требуется», либо «ПО обновлено до версии N. Необходима перезагрузка приложения». Запускается метод, который по серийному номеру робота-пылесоса и текущей версии ПО проверяет наличие обновлений. |  |  |
| 14 | Установка расписания работы робота-пылесоса | setShedule(Robot r, List<DateTime> dates, int map\_num, int option): boolean | В таблице «Расписание» сохраняются даты и время начала работы робота-пылесоса, а также номер схемы помещения и тип уборки. | Robot\_id | В окне «Расписания работы» пользователь выбирает дни недели и часы начала работы робота-пылесоса и нажимает на кнопку «Сохранить». Запуск возможен 1 раз в сутки. |  |  |
| 15 | Выбор помещения для уборки | chooseMap(Robot r, int map\_num): Boolean | В таблицу «Расписание» записывается номер схемы, которая отображается на экране. | List<Map\_num> | По стрелкам слайдера пользователь выбирает схему помещения для уборки. |  |  |
| 16 | Обновление схемы помещения | updateMap(Robor r): int | В таблице «Расписание» для данного робота обновляется номер схемы помещения. | Map\_num | Для обновления схемы помещения пользователь нажимает на кнопку «Обновить». При этом робот-пылесос заново сканирует помещение, в котором находится, сохраняет схему в своей памяти и отправляет схемы на устройство пользователя. |  |  |
| 17 | Удаление схемы помещения | deleteMap(int map\_num): Boolean | В таблице «Расписание» для всех роботов, имеющих удаляемую схему помещения, записывается номер предыдущей схемы помещения. | Map\_num | Отображается предыдущая схема помещения. Если схем нет, то - только кнопка «Обновить схему помещения» |  |  |
| 18 | Просмотр логов работы робота-пылесоса | showLogs(Robot r): void |  | Robot\_id | Пользователь может просматривать логи работы пылесоса, листая страницы логов. |  |  |
| 19 | Ручное управление | toRight(Robot r): void  toLeft(Robot r): void  forward(Robot r): void  back(Robot r): void |  | Robot\_id | Приложение отслеживает местоположение робота-пылесоса. При нажатии вперед пылесос движется вперед до препятствия. При нажатии назад пылесос движется назад до препятствия. При нажатии вправо пылесос поворачивает направо. При нажатии влево пылесос поворачивает налево. |  |  |
|  |  |  |  |  |  |  |  |