**План тестирования облачного сервиса и приложения по управлению роботом - пылесосом.**

1. **Unit - тесты.** Пишутся по мере написания кода на классы и методы API облачного сервиса, API обновления, авторизации и мобильного приложения.
2. **E2E - тестирование.** Проводим после завершения работы над кодом. Тестируем каждую систему на работоспособность. При этом воздействуем на систему через ее самые внешние интерфейсы и проверяем ожидаемую реакцию системы через эти же интерфейсы.

Таблица 1. Протокол тестирования API облачной системы.

| **№ п/п** | **Вызываемый метод** | **Ожидаемый результат** |
| --- | --- | --- |
| 1.1 | Register | Происходит регистрация нового пользователя с добавлением его в базу данных. Метод возвращает новый токен доступа и новый токен обновления. |
| 1.2 | LoginByPassword | Происходит авторизация пользователя по паролю. Метод возвращает новый токен доступа и новый токен обновления. |
| 1.3 | LoginByAccessToken | Происходит авторизация пользователя по токену доступа. Метод ничего не возвращает. |
| 1.4 | LoginByRefreshToken | Происходит авторизация пользователя по токену обновления в случае, если срок действия токена доступа истек. Метод возвращает новый токен доступа и новый токен обновления. |
| 2.1 | GetRobotsInGroup | Метод извлекает из базы данных сведения о всех роботах, которые прикреплены к определенной группе и возвращает коллекцию данных о роботах в виде json строки. |
| 2.2 | GetRobot | Метод извлекает из базы данных сведения о конкретном роботе и возвращает данные о нем в виде json строки. |
| 2.3 | DeleteRobot | Метод удаляет указанного робота из базы данных и возвращает флаг выполнения операции. |
| 2.4 | UpdateRobot | Метод обновляет сведения об указанном роботе и возвращает флаг выполнения операции. |
| 2.5 | AddRobot | Метод добавляет робота в базу данных и возвращает его идентификатор в базе данных. |
| 3.1 | GetSchedule | Метод извлекает из базы данных расписание уборки помещений и возвращает его в виде json строки. |
| 3.2 | UpdateSchedule | Метод обновляет указанное расписание и возвращает флаг выполнения операции. |
| 3.3 | DeleteSchedule | Метод удаляет расписание уборки из базы данных и возвращает флаг выполнения операции. |
| 3.4 | AddShedule | Метод добавляет расписание уборки в базу данных и возвращает его идентификатор в базе данных. |
| 4.1 | UpdateSoft | Метод обращается к API обновления и получает путь к файлу обновления, если обновление необходимо. Метод возвращает путь обновления. |

1. **UAT - тестирование.** Готовый программный продукт тестирует ограниченный круг пользователей. Тестируется каждый кейс из User case диаграммы. При этом группа людей изучает эффективность сервиса, его функционала. UAT нужен для того, чтобы понять:

* как ведет себя продукт в реальных условиях, соответствует ли результат задумке;
* выявить, были ли добавлены все возможные функции;
* проверить, есть ли ошибки, которые будут мешать пользователю.

Таблица 1. Протокол тестирования клиентского приложения.

| № п/п | User case | Действие | Ожидаемый результат | Соответствие |
| --- | --- | --- | --- | --- |
|  | Подключение | Подключить пылесос к мобильному устройству и следовать указаниям. | При подключении пылесоса к мобильному устройству, мобильное устройство просит пользователя скачать и установить приложение для управления роботом-пылесосом. |  |
|  | Уборка экстренная | Выбрать план помещения, указать область уборки и нажать кнопку “EMERGRENCY CLEANING”. | При запуске экстренной уборки в указанной области робот - пылесос должен совершить там уборку и вернуться на место подзарядки. |  |
|  | Уборка автоматическая | Выбрать список доступных роботов и возле нужного робота нажать кнопку “>” для запуска автоматической уборки | При запуске автоматической уборки робот - пылесос должен совершить уборку и вернуться на место подзарядки. |  |
|  | Уборка по расписанию | Выбрать расписание и установить, когда робот должен убрать комнату. Дождаться начала уборки. | При достижении времени уборки, робот - пылесос должен совершить уборку и вернуться на место подзарядки. |  |
|  | Ремонт | При заполнении шкалы “Next service” произвести техническое обслуживание робота в сервисе. Проконтролировать успешность сброса шкалы технического обслуживания. | При заполнении шкалы “Next service” робот должен стать на место зарядки и подать сигнал о том, что нужно техническое обслуживание. После проведения ТО, робот должен работать так же как работал до проведения ТО. Шкала “Next service” должна обнулиться. |  |
|  | Обновление | Выбрать робота и нажать кнопку “UPDATE”. Проконтролировать обновление системы. | Если обновление уже вышло, робот должен вернуться на место зарядки, обновить свое ПО и продолжить уборку. На странице робота в приложении должно появиться уведомление о том, что ПО обновлено и его актуальная версия.  Если обновление не вышло, робот должен не прерывать уборку, а на странице робота в приложении должно появиться информационное сообщение о том, что установленное ПО последней версии. |  |
|  | Чистка | При заполнении шкалы “Robot Pollution” произвести очисткку рабочего узла робота. Проконтролировать поведение датчика загрязнения после очистки. | При заполнении шкалы “Robot Pollution” робот должен стать на место зарядки и подать сигнал о том, что нужна очистка рабочего узла. После проведения очистки, робот должен работать так же как работал до очистки. Шкала “Robot Pollution” должна обнулиться. |  |
|  | Чистка контейнера с мусором | При заполнении шкалы “Garbage Container” произвести очистку контейнера с мусором и фильтра. Проконтролировать поведение датчика уровня заполненности контейнера после очистки. | При заполнении шкалы “Garbage Container” робот должен стать на место зарядки и подать сигнал о том, что нужно очистить контейнер. После очистки контейнера, робот должен работать так же как работал до очистки. Шкала “Garbage Container” должна обнулиться. |  |