

Test Plan

Умный дом

Version 1.0.0

Revision History

Date	Version	Description	Author
07.12.2018	1.0.0	Creating	Иващенко А.В.
08.12.2018	1.1.0	MODIFY	Иващенко А.В.
08.12.2018	2.0.0	MODIFY	Иващенко А.В.
09.12.2018	2.1.0	MODIFY	Иващенко А.В.

Содержание

1. Описание User Story;
2. Задача
3. Критерии оценки
 - 3.1. Качественные
 - 3.2. Версии продукта:
4. Тест стратегия
5. Результаты

Описание User Story:

- В квартире есть датчики: умная розетка, холодильник, входная дверь, инфракрасный датчик движения, чайник, роутер
- Каждый датчик имеет ID, настраиваемый пользователем
- Умные розетки, инфракрасный, входная дверь - пассивные (к ним нужно постучаться с запросом, чтобы получить данные)
- Остальные активные (в них есть параметры листенера, они сами стучатся и присылают данные)
- Для начала работы с датчиками листенер слушает входящие подключения от активных датчиков или коннектится сам к пассивным датчикам и регистрирует данные по датчикам у себя
- зарегистрировать можно только те датчики, тип которых среди разрешённых
- при регистрации от датчика приходит ID и тип
- листенер проводит мониторинг статуса зарегистрированных датчиков с заданной периодичностью (период настраивается для каждого типа в конфиге)
- на экран выводится Online (статус датчика ещё не потерял актуальность по времени), Offline (датчик не прислал подтверждение статуса или не ответил на последний запрос)
- При обмене данными датчик может присылать дополнительные метрики:
- розетки - наличие потребителя, напряжение и ток
- холодильник - температура камер
- входная дверь - открыта/закрыта, дата последней смены статуса
- инфракрасный датчик - наличие подвижных объектов, дата последней смены статуса
- чайник - статус (выключен, кипячение, поддержание температуры), температура
- роутер - наличие Интернет, статистика приёма/передачи за период
- Дополнительные метрики могут служить основой для нотификации, если таковая предусмотрена конфигом.

Задача:

Вам будет предоставлен продукт, который будет развиваться с каждой новой итерацией. Необходимо обеспечить качество продукта на всех итерациях.

Язык программирования – Python 3.7

На чем писать тесты – ограничений нет

Количество версий продукта – 4 (подробнее см. ниже)

Продукт будет предоставляться в виде архива исходников

Когда будут выкатываться новые версии продукта:

- в пятницу вечером
- в субботу утром
- в субботу вечером
- в воскресенье утром.

Критерии оценки:

Количественные:

- покрытие кода
- покрытие требований
- найденное число ошибок и их описание
- найденное число дефектов

Качественные:

- CI/CD, время его прохождения, сбор статистики и прочие полезности
- тест-анализ, тест-дизайны (да\нет)
- наличие регресса и удобство его изменения (да\нет)
- производительность,
- нагрузка,
- масштабируемость,
- безопасность (да\нет)
- автоматизация

Версии продукта:

1.0.0 – Реализован сервер мониторинга, к которому могут подключаться активные датчики

- Реализована обработка и хранение данных от датчиков «чайник» и «роутер»
- Реализована работа с файлом конфигурации (адрес сервера, разрешённые типы датчиков, таймауты актуальности данных)
- Реализован переход датчика в offline при неактивности больше заданного таймаута

1.1.0 – Добавлена поддержка дополнительных метрик датчиков

- реализовано хранение метрик датчиков «чайник» и «роутер»

2.0.0 – Реализована нотификация (запись в файл) при срабатывании правила, заданного

для типа датчиков

2.1.0 – Добавлен тип датчика «холодильник»

Тест стратегия

Приведенный ниже план функционального тестирования является формальным, так для построения развернутого плана необходимо понимание текущего состояния проекта. В результате первого прогона функциональных тестов в тест-план будут внесены изменения. Первый прогон функциональных тестов даст нам четкое представление об уровне стабильности системы и будет четко определен набор тестов, которые будут выполнены в каждой конфигурации.

Такой подход даст возможность получить развернутый отчет по тестируемому продукту и сосредоточить максимальное внимание на узких местах.

В результате тестирования заказчик получит набор документов с полным результатом тестирования. При предоставлении заказчиком доступа в багтрекер все обнаруженные дефекты будут занесены в виде отдельных багов для последующего исправления.

В процессе тестирования будет применено ad-hoc тестирование и wite box ввиду отсутствия строгой спецификации, а также ввиду ограниченности ресурсов на формализацию тестов. Исходя из того, что количество различных параметров и возможных сочетаний этих параметров может быть довольно большим и такая проверка требует достаточно длительных временных затрат, мы предлагаем ограничиться типовым готовым набором, состоящим из различных конфигураций.

Тестируемые типы

На первом этапе тестирования:

1. Проверка новых фич:
- 1 Проверка реализации сервера который слушает активные датчики (Feature-1, Feature-4)
- 2 Проверка зачитывания параметров из конфига (Feature-3)
- 3 Проверка работы с датчиками типов "чайник" и "роутер" (Feature-8, Feature-10)
- 4 Проверка реализации сброса датчика в offline при срабатывании таймаута на тип датчика (Feature-5)
- 5 Проверка реализации нотификации в файле на основе правил конфига (Feature-6)
2. Тестирование функциональности подключение к серверу активных датчиков, переход датчиков в офлайн при неактивности больше заданного таймаута. Обработка ошибок.
3. Тестирование локализации в рамках проекта Windows, Linux.

4. Позитивное, негативное тестирование сервера мониторинга.
5. Компонентное тестирование сервера, интеграционное тестирование сервера с датчиками.
6. Дымное тестирование, ad-hoc тестирование работы сервера.

На втором этапе тестирования:

1. Будет проведен обзор тестовых сценариев и будут выявлены часто повторяющиеся операции, которые довольно легко автоматизировать.
2. Проводится проверка новых фич: поддержка дополнительных метрик датчиков реализовано. Реализация хранения метрик датчиков чайника и роутера.
3. Тестированию UI/UX.
4. Тестирование функциональности активных датчиков чайника и роутера. Для чайника статус режимов работы выключен, кипячение, поддержание температуры. Для роутера наличие отсутствие интернета.
5. Проведение регрессионного тестирования фич. первого этапа.
6. Дымное тестирование, ad-hoc тестирование работы сервера.

На втором третьем этапе тестирования:

1. Проводится проверка новых фич: нотификация (запись в файл) при срабатывании правила, заданного для типа датчиков.
2. Проверка регрессионного тестирования.
3. Проведение ad-hoc тестирования.

На четвертом этапе тестирования:

1. Проводится проверка новых фич: добавление типа датчика «Холодильник» а так же других возможных датчиков.
2. Проверка регрессионного тестирования.

Результаты:

Конечным результатом проведения тестирования должен стать развернутый результат тестирования с найденными дефектами. Итогом процесса тестирования будут следующие материалы: отчет о результатах тестирования текущего покрытия, типовые сценарии использования/ОС/конфигурации, найденные баги; рекомендации относительно мест в продукте и условий работы (окружения, сценариев использования), которые, по мнению команды тестирования.

