# Спецификация требований к программному обеспечению

## Клиент-серверное приложение “SmartHouse”

### Содержание

1. Введение  
   1.1. Назначение  
   1.2. Соглашения, принятые в документах  
   1.3. Границы проекта  
   1.4. Ссылки
2. Общее описание  
   2.1. Общий взгляд на продукт  
   2.2. Классы и характеристики пользователей  
   2.3. Операционная среда  
   2.4. Ограничения дизайна и реализации  
   2.5. Предположения и зависимости
3. Системные функции  
   3.1. Решение задач пользователя
4. Требования к данным  
   4.1. Логическая модель данных  
   4.2. Словарь данных  
   4.3. Отчеты  
   4.4. Целостность, сохранение и утилизация данных
5. Требования к внешним интерфейсам  
   5.1. Пользовательские интерфейсы  
   5.2. Интерфейсы ПО  
   5.3. Интерфейсы оборудования  
   5.4. Коммуникационные интерфейсы
6. Атрибуты качества  
   6.1. Требования по удобству использования  
   6.2. Требования к производительности  
   6.3. Требования безопасности  
   6.4. Требования к защите  
   6.5. Требования к доступности  
   6.6. Требования к надежности  
   Приложение А. Модели анализа  
   Бизнес-правила

## 1. Введение

### 1.1. Назначение

Данная спецификация описывает требования к выпуску 1.0 программного комплекса “SmartHouse” — системы управления устройствами умного дома. Документ предназначен для разработчиков и тестировщиков, реализующих и поддерживающих систему. Все указанные требования, если не отмечено иного, имеют высокий приоритет для релиза 1.0.

### 1.2. Соглашения, принятые в документах

В этой спецификации термины используются в общепринятом смысле: - **Пользователь**: Владелец или член семьи, использующий систему для управления устройствами, мониторинга состояний и получения уведомлений. - **Администратор**: Пользователь с расширенными правами (например, настройка пользователей, сценариев автоматизации, доступа). - **Система**: Программно-аппаратный комплекс “SmartHouse”, включающий клиентское приложение, сервер и оборудование дома.

### 1.3. Границы проекта

Приложение SmartHouse позволяет: - Управлять устройствами (освещение, климат, розетки и т.д.)  
- Создавать и выполнять сценарии автоматизации  
- Просматривать состояние устройств и журнал событий  
- Получать уведомления о событиях и авариях  
- Администраторам — управлять пользователями, устройствами, сценариями и просматривать статистику использования

### 1.4. Ссылки

1. Документация Qt — https://doc.qt.io
2. Документация Docker — https://docs.docker.com
3. Документация Git — https://git-scm.com/doc
4. Документация Doxygen — https://www.doxygen.nl/manual/index.html
5. Спецификация протокола MQTT — https://mqtt.org/documentation
6. Спецификация REST API — https://restfulapi.net/
7. Документация SQLite — https://www.sqlite.org/docs.html

## 2. Общее описание

### 2.1. Общий взгляд на продукт

SmartHouse — это система автоматизации дома, предоставляющая: - Управление и мониторинг устройств (освещение, климат, датчики, розетки, камеры) - Создание и исполнение сценариев автоматизации (например, “Утро”, “Температура”) - Просмотр журнала событий и статистики - Получение уведомлений о событиях (например, включение, отключение девайса) - Использование устройств и сценариев

### 2.2. Классы и характеристики пользователей

| Класс пользователя | Описание |
| --- | --- |
| Пользователь | Управляет устройствами, просматривает их состояние, использует сценарии. Работает через мобильное или десктопное приложение. |
| Администратор | Имеет доступ к расширенным настройкам: добавление/удаление пользователей, настройка сценариев, управление оборудованием, просмотр полной статистики и журнала. |
| Системный администратор | Технический специалист, обслуживающий серверную часть, базу данных, безопасность и интеграцию с внешними сервисами. |

### 2.3. Операционная среда

* Клиентское приложение работает на:
  + Windows 10/11
  + macOS Big Sur и выше
  + Linux (Ubuntu, Fedora)

### 2.4. Ограничения дизайна и реализации

* Код документируется согласно стандартам Doxygen.
* Архитектура оформляется в виде UML-диаграмм классов, компонентов, последовательностей.
* Вся документация хранится в системе контроля версий (Git).
* Для UI используется Qt (кроссплатформенность).
* Серверная часть должна поддерживать REST API и/или MQTT.
* Поддержка ролевой модели доступа.
* Все автоматизированные сценарии должны быть протестированы для исключения аварийных ситуаций.

### 2.5. Предположения и зависимости

**Предположения:** - Система используется в жилых и малых коммерческих помещениях. - Пользователи имеют базовые навыки обращения с десктопными приложениями.

- Пиковая нагрузка обычно утром и вечером.

**Зависимости:** - Стабильная работа локальной сети или интернет-соединения. - Корректная работа базы данных. - Надёжная работа MQTT-брокера (или другого канала обмена с устройствами). - Безопасное хранение и передача данных (шифрование, TLS).

## 3. Системные функции

### 3.1. Решение задач пользователя

#### 3.1.1. Описание

Пользователь может: - Просматривать и управлять состоянием устройств - Получать уведомления о событиях и авариях - Создавать, запускать и изменять сценарии автоматизации.

Администратор дополнительно: - Управляет пользователями, устройствами, настройками сценариев, просматривает расширенную статистику и журнал.

#### 3.1.2. Функциональные требования

**Авторизация**

| Этап | Описание | Действие системы | Возможные варианты ответа |
| --- | --- | --- | --- |
| Вход пользователя | Проверка учётных данных | Система подтверждает личность по логину и паролю | Успешно/Ошибка |
| Вход администратора | Проверка прав | Система подтверждает права администратора | Успешно/Ошибка |
| Регистрация | Создание новой учётной записи | Система добавляет пользователя с выбранной ролью | Успешно/Ошибка (например, логин занят) |

**Управление устройствами**

| Этап | Описание | Действие системы | Возможные варианты |
| --- | --- | --- | --- |
| Просмотр состояния | Пользователь видит список устройств и их статусы | Система отображает актуальное состояние | Список устройств и их состояния |
| Управление устройством | Включение/выключение, настройка параметров | Система отправляет команду устройству, обновляет статус | Подтверждение/Ошибка |

**Сценарии автоматизации**

| Этап | Описание | Действие системы | Возможные варианты |
| --- | --- | --- | --- |
| Создание сценария | Пользователь задаёт условия и действия | Система сохраняет сценарий | Подтверждение/Ошибка |
| Запуск сценария | Пользователь вручную или по расписанию запускает сценарий | Система исполняет действия | Подтверждение/Ошибка |

**Уведомления и журнал событий**

| Этап | Описание | Действие системы | Возможные варианты |
| --- | --- | --- | --- |
| Получение уведомлений | Система информирует о событиях (авария, достижение порога) | Показывается уведомление в приложении/по email/Push | Список уведомлений |
| Просмотр журнала | Пользователь или администратор просматривает историю событий | Система отображает журнал с деталями | Список событий |

## 

## 4. Требования к данным

### 4.1. Логическая модель данных

**Основные сущности:** - Пользователь: - id - login - password\_hash - role - contact\_info (email, телефон) - настройки уведомлений - Устройство: - id - type (розетка, лампа, датчик и т.д.) - name - state (включено/выключено, значение датчика) - location (комната) - параметры (например, яркость, температура) - last\_active - Сценарий: - id - name - условия (триггеры) - действия - enabled/disabled - last\_run - Журнал событий: - id - user\_id - device\_id - action\_type (включение, изменение параметра, ошибка) - timestamp - result - Уведомление: - id - user\_id - type (push, email) - message - timestamp - status (доставлено/прочитано)

**4.2. Словарь данных**

| Элемент данных | Описание | Тип данных | Длина/Ограничения |
| --- | --- | --- | --- |
| Логин | Уникальный идентификатор пользователя | Алфавитно-числовой | 50 |
| Пароль | Хэш пароля | Строка | SHA-256 |
| Роль | Роль пользователя (user/admin) | Строка | 16 |
| ID пользователя | Уникальный идентификатор | Целое | авто |
| Email, телефон | Контактные данные | Строка | 100 |
| ID устройства | Уникальный id | Целое | авто |
| Имя устройства | Название | Строка | 50 |
| Тип устройства | Класс (лампа, датчик и пр.) | Строка | 32 |
| Параметры | Индивидуальные параметры | JSON | - |
| Сценарий | id, имя, триггеры, действия | Структура | - |
| Журнал событий | id, user\_id, device\_id, action\_type, timestamp, result | Структура | - |
| Уведомление | id, user\_id, type, message, timestamp, status | Структура | - |

### 

### 4.3. Отчеты

**Пример отчёта для администратора по событиям и использованию системы:**

| Параметр | Значение |
| --- | --- |
| Идентификатор отчёта | SH-RPT-1 |
| Заголовок | Сводная статистика по событиям и использованию устройств |
| Цель | Анализ активности пользователей и устройств, выявление частых сценариев и аварийных ситуаций |
| Пользователи | Администраторы системы |
| Источники данных | Журнал событий, данные устройств, сценариев, пользователей |
| Частота | По запросу |
| Время отклика | Не более 5 секунд |
| Макет | Таблица: пользователь, устройство, количество операций, частые сценарии, аварии, время последней активности |
| Фильтры | Диапазон дат, пользователь, тип события |
| Экспорт | CSV, PDF |

### 4.4. Целостность, сохранение и утилизация данных

* Все события и действия хранятся не менее года.
* Данные пользователей и устройств резервируются ежедневно.
* При удалении пользователя все его сценарии и настройки могут быть утилизированы или анонимизированы.

## 

## 5. Требования к внешним интерфейсам

### 5.1. Пользовательские интерфейсы

* Экран авторизации/регистрации
* Главный экран: список устройств, управление ими
* Экран сценариев: создание, просмотр, запуск
* Экран уведомлений и журнала событий
* Административная панель: управление пользователями, устройствами, сценариями, просмотр отчётов

### 5.2. Интерфейсы ПО

* Сервер предоставляет REST API для обмена с клиентами, поддержка WebSocket/MQTT для реального времени.
* Внутренний API для интеграции с оборудованием и внешними сервисами.
* Взаимодействие с БД через ORM или прямые SQL-запросы.

### 5.3. Интерфейсы оборудования

* Поддержка протоколов (MQTT, Zigbee, Z-Wave, Wi-Fi, Bluetooth LE).
* Возможность подключения модулей сторонних производителей через API-шлюзы.

### 5.4. Коммуникационные интерфейсы

* Все команды и статусы передаются через защищённые каналы (TLS).
* Пользователи получают push-уведомления через десктопные клиенты.

## 6. Атрибуты качества

### 6.1. Требования по удобству использования

* Интуитивно понятный интерфейс.
* Быстрый доступ к основным функциям.
* Простой процесс регистрации и входа.
* Гибкая настройка уведомлений и сценариев.

### 6.2. Требования к производительности

* Время отклика при управлении устройствами — не более 2 секунд.
* Поддержка одновременной работы не менее 20 пользователей.

### 6.3. Требования безопасности

* Все пользователи проходят аутентификацию.
* Пароли хранятся в виде хэшей.
* Передача данных только по защищённым каналам (TLS).

### 6.4. Требования к защите

* Защита от несанкционированного доступа и атак (SQL injection, XSS).
* Разграничение прав доступа по ролям.
* Логирование всех административных действий.

### 6.5. Требования к доступности

* Система должна быть доступна 24/7.
* В случае сбоя — автоматическое восстановление из резервных копий.

### 6.6. Требования к надёжности

* Резервное копирование БД и журналов событий.
* Автоматическая диагностика и оповещение при неполадках.

## Приложение А. Модели анализа

* Диаграммы классов и компонентов размещены на Wiki проекта или по ссылке:  
  https://github.com/lanskypavel/Smart-Home-/wiki

## 

## Бизнес-правила

| Идентификатор | Определение правила | Тип правила | Статичное/Динамичное | Источник |
| --- | --- | --- | --- | --- |
| ID-1 | Пользователь может управлять только теми устройствами, к которым у него есть доступ. | Ограничение | Динамическое | Разработчики SmartHouse |
| ID-2 | Запрещено создавать сценарии, приводящие к зацикливанию команд. | Ограничение | Статическое | Разработчики SmartHouse |
| ID-3 | Все изменения состояния устройств должны логироваться. | Факт | Динамическое | Разработчики SmartHouse |
| ID-4 | Только администратор может добавлять или удалять пользователей и устройства. | Ограничение | Статическое | Разработчики SmartHouse |
| ID-5 | При потере связи с устройством система должна уведомить пользователя. | Факт | Динамическое | Разработчики SmartHouse |
| ID-6 | Пароли пользователей должны храниться только в виде хэшей. | Ограничение | Статическое | Разработчики SmartHouse |
| ID-7 | Удаление пользователя приводит к удалению или анонимизации его персональных данных. | Ограничение | Динамическое | Разработчики SmartHouse |
| ID-8 | В журнале событий не должно быть возможности изменения ранее зафиксированных записей. | Ограничение | Статическое | Разработчики SmartHouse |
| ID-9 | Все сценарии автоматизации должны быть протестированы перед активацией. | Факт | Динамическое | Разработчики SmartHouse |
| ID-10 | В случае попытки неавторизованного доступа фиксируется событие безопасности. | Ограничение | Динамическое | Разработчики SmartHouse |