#### Условия

Юные фермеры решили автоматизировать свою теплицу, установив в нее автоматизированную систему контроля и управления климатом.

Автоматизированная система состоит из:

1. Четыре равноудалённых друг от друга датчика температуры и влажности воздуха, что позволяет корректно определять среднюю температуру воздуха и влажности в теплице
2. Шесть датчиков влажности почвы, каждый из которых установлен в свою собственную тепличную бороздку
3. Автоматизированный привод удаленного открытия форточек теплицы для обеспечения проветривания
4. Шесть систем автоматизированного полива грядки, каждая из которых установлена в собственную тепличную бороздку по аналогии с датчиками влажности почвы
5. Едина система увлажнения воздуха в теплице.

Автоматизированная система представляет из себя комплекс специальных измерительных и управляющих приборов, взаимодействие с которыми осуществляется по средствам API:

Получение измерений с датчиков температуры и влажности осуществляется запросом GET на адрес https://dt.miet.ru/ppo\_it/api/temp\_hum/<number>, где <number> – порядковый номер устройства от 1 до 4.

Пример корректного ответа датчика температуры и влажности воздуха в формате JSON:

*{*

*"id": device\_id,*

*"temperature": temp\_value,*

*"humidity": hum\_value*

*}*

Где device\_id – номер устройства, temp\_value – температура воздуха в моментзапроса, hum\_value – влажность воздуха в момент запроса

Получение измерений с датчиков влажности почвы осуществляется запросом GET на адрес https://dt.miet.ru/ppo\_it/api/hum/<number>, где <number> – порядковый номер устройства от 1 до 6.

Пример корректного ответа датчика влажности почвы в формате JSON:

*{*

*"id": device\_id,*

*"humidity": hum\_value*

*}*

Где device\_id – номер устройства, hum\_value – влажность воздуха в момент запроса

Управление автоматизированным приводом открытия форточек осуществляется запросом PATCH на адрес https://dt.miet.ru/ppo\_it/api/fork\_drive/ со следующими параметрами:

PATCH это метод! Адрес получается https://dt.miet.ru/ppo\_it/api/fork\_drive?state=1

*{*

*"state": state\_value*

*}*

Где state\_value принимает значение 0 – закрыть форточку, 1 – открыть форточку

Управление автоматизированным поливом грядки осуществляется запросом PATCH на адрес https://dt.miet.ru/ppo\_it/api/watering со следующими параметрами:

*{*

*"id": device\_id*

*"state": state\_value*

*}*

Где device\_id – номер устройства от 1 до 6, state\_value принимает значение 0 – закрыть полив, 1 – открыть полив

Управление единой системой увлажнения осуществляется запросом PATCH на адрес https://dt.miet.ru/ppo\_it/api/total\_hum со следующими параметрами:

*{*

*"state": state\_value*

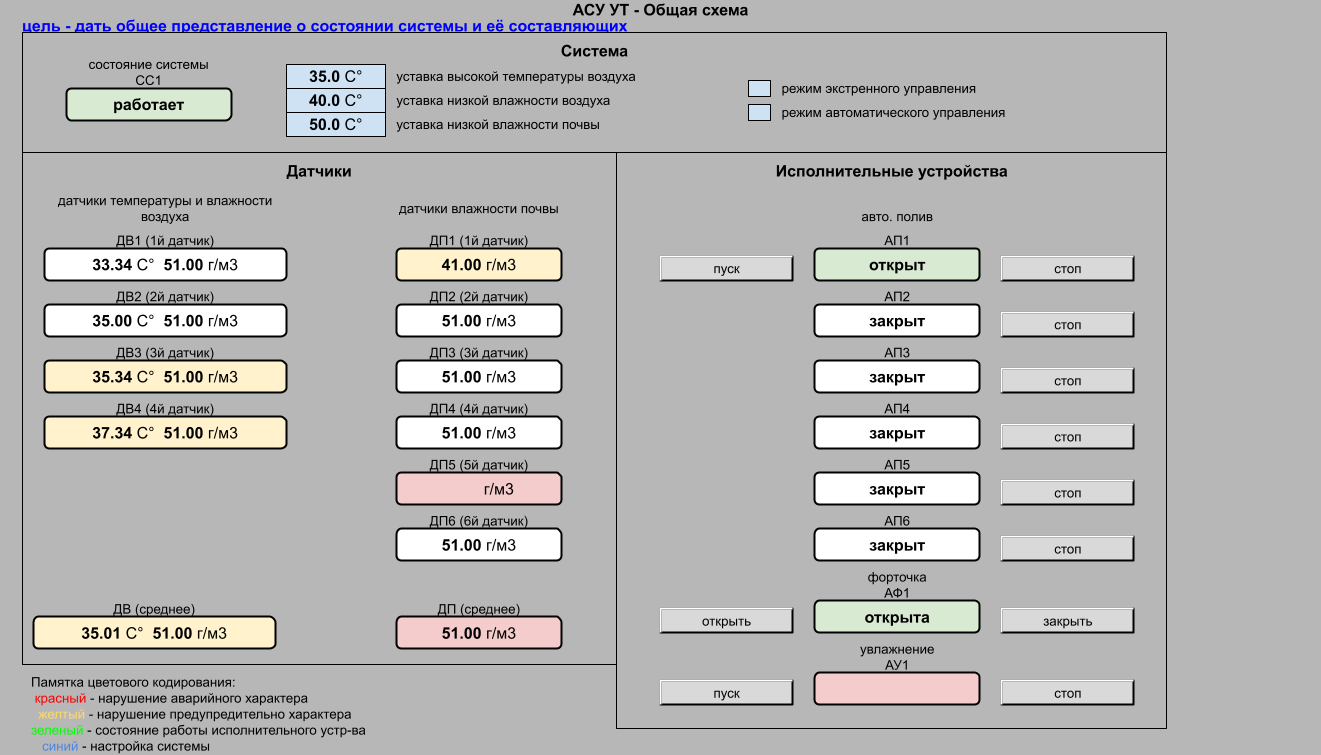
*}*

Где state\_value принимает значение 0 – закончить увлажнение, 1 – начать увлажнение

#### Техническое задание

Имеется программный интерфейс, с помощью которого можно взаимодействовать с аппаратной частью автоматизированной системы управления теплицей. Для взаимодействия необходимо выполнить GET и PATCH запросы к методам API, описанным выше. В заголовках запроса необходимо указать токен авторизации, выданный организаторами: {“X-Auth-Token”: “<token>”}

Необходимо разработать программный продукт, который позволит:

1. Отображать динамику и последние данные с каждого датчика температуры и влажности, как в табличном виде, так и в виде графиков.  
   ЧМИ составим из нескольких мнемосхем:  
   Общая схема АСУ УТ - вид на котором отображены последние показания всех датчиков (скаляр – одно значение на датчик) на прямоугольном фоне для цветового кодирования статуса нарушения (жаль, что статус датчика получить не можем).  
     
   История показаний АСУ УТ – в виде графиков и таблиц + средние значения.  
     
   Данные формировать в рамках текущей рабочей сессии.  
   Как это понимать? Т.е. только после запуска приложения в БД появятся данные?  
   Период демонстрации определяется участником  
   Участник это разработчик?  
   Слово период используется в описании циклических процессов. Период демонстрации это время автоматического обновления экрана?  
   Отображать динамику и последние данные средней температуры и средней влажности в теплице, как в табличном виде, так и в виде графиков
2. Открыть или закрыть форточки в теплице нажатием на специальные кнопки, а также просматривать текущий статус
3. Отображать динамику и последние данные с каждого датчика влажности почвы, как в табличном виде, так и в виде графиков
4. Открыть или закрыть полив конкретной бороздки в теплице нажатием на специальные кнопки, а также просматривать текущий статус
5. Открыть или закрыть общее увлажнение теплицы нажатием на специальные кнопки, а также просматривать текущий статус  
   Наверно пуск и стоп для оборудования увлажнения

Предусмотреть также выполнение следующих условий:

1. Если средняя температура в теплице больше, чем **T градусов**, кнопка открытия форточек активна, в противном случае нажатие невозможно
2. Если средняя влажность в теплице меньше, чем **H %**, кнопка открытия общей системы увлажнения активна, в противном случае нажатие невозможно
3. Если средняя влажность почвы в конкретной бороздке теплицы меньше, чем **Hb %**, кнопка открытия системы полива активна, в противном случае нажатие невозможно  
   Средней влаж. почвы нет!

Предусмотреть режим настройки системы и указания параметров: **T, H, Hb.**

Также предусмотреть переход в режим экстренного управления, при котором возможно нажатие всех описанных выше кнопок даже если условия не соблюдены.

#### Рекомендации к выполнению

* Программный продукт должен работать или на мобильных устройствах, или иметь веб-интерфейс.
* Полезной будет возможность внесения новых данных от датчиков вручную в интерфейсе пользователя.
* Необходимо предусмотреть сохранение данных, получаемых от датчиков в системе управления базами данных (СУБД). Выбор СУБД не регламентируется.
* Разработку рекомендуется вести с помощью системы контроля версий git.
* Рекомендуется использовать unit-тестирование при разработке продукта.

#### Требования к документации

* Титульный лист (с указанием названия кейса и перечислением членов команды);
* Анализ технических требований;
* Обоснование выбора языка программирования и используемых программных средств;
* Структурная и функциональная схемы программного продукта;
* Блок-схема работы основного алгоритма;
* Схема базы данных;
* Описание проведенных испытаний в соответствии с регламентом кейса (снимки экрана и/или запись экрана с работой);
* Программный код (ссылка на репозиторий).

#### Регламент испытаний

* Производится запуск программного продукта
* Отображение показателей всех датчиков в виде графиков и таблиц
* Отображение текущего состояния все устройств управления теплицей
* Настройка изменяемых параметров для работы системы
* Демонстрация работоспособности всех кнопок с учётом настроенных
* параметров
* Демонстрация режима экстренного управления
* Демонстрация unit-тестов

#### Примерный перечень средств и инструментов для выполнения задания

* https://www.python.org/
* https://nodejs.org/
* https://sqlite.org/
* https://www.postgresql.org/
* https://pypi.org/project/requests/
* https://flask.palletsprojects.com/en/2.0.x/
* https://reactjs.org/
* https://vuejs.org/
* https://matplotlib.org/