SEP4 Project Requirements

# General

### DevOps

All groups must employ DevOps methods/technologies from the DevOps course. The exact nature and extent of using DevOps techniques will depend on the type of the individual project (IoT, backend, frontend), and will be decided in consultation between the groups and the DevOps supervisor(s). But all projects must:

* Perform unit tests of (parts of) their application.
* Maintain their code base using a distributed Git setup, which must include the use of a central Git hub such as GitHub or GitLab.
* Demonstrate an appropriate and systematic use of Git features such as tags and branches.
* Setup automated regression testing using the features available in GitHub/GitLab
* Use containers to encapsulate their development setup and, where applicable, the runtime environment of their application.

### Process

You must utilize SCRUM & AUP for the development process.

The project must be under version control through the entire project period. The commit history must be clear. For pair programming you must write the names of the participators in the commit message.

### Handin

You must hand in a project- & process report. The project report must include authors for each section.

The project report can be no longer than 60 pages (1 page is considered 2400 characters).  
We expect the project report to be one coherent unit up until the design section. That is, you hand in one list of user stories, one domain model, etc. The design section should start out with an overview, and then branch out into 3 sub-sections – one for each team. The report should merge again for results and discussion, conclusions & future work. Here is a diagram illustrating the structure:

|  |  |  |
| --- | --- | --- |
| Introduction | | |
| Analysis | | |
| Design Overview | | |
| IoT Design | Cloud Design | Frontend Design |
| IoT Implementation | Cloud Implementation | Frontend Implementation |
| IoT CI/CD | Cloud CI/CD | Frontend CI/CD |
| IoT Test | Cloud Test | Frontend Test |
| Results and Discussion | | |
| Conclusions | | |
| Future Work | | |

Notice the added continuous integration and continuous delivery sections, that each team must include.

You must hand in your source code and links to any used repository.

You must also hand a video with your group presentation for the project (30 minutes). The video should also show the complete system in action with all 3 parts working together.

# IoT

Design and implement a FreeRTOS based application in C for an AVR2560 MCU that interfaces to several sensors and actuators

Use LoRaWAN (IoT-Network) for transmitting and receiving data. Drivers to sensors and actuators will be given.

Minimum requirements to the embedded software:

* You must use at least five tasks.
* Some data must be used by more than one task.
* You must use semaphores, mutexes and queues.
* Your software must run on your Arduino Mega2560 board with VIA-shield and make use of the following sensors and actuators:
* Temperature/Humidity
* CO2
* LoRaWAN transceiver
* RC-servo
* Light (optional)
* Sound (optional)
* PIR sensor (optional)

# Cloud

The backend must store and manage sensor data and other relevant information for the system within a database, and subsequently make the information accessible via a RESTful web service. Moreover, it must leverage cloud computing, encompassing:

* Hosting the software with a public cloud provider
* Utilizing containers for deploying the software
* Running part of the software as serverless workloads

# Frontend

*These are the essential requirements for developing the web application. Feel free to add more:*

* The application must be developed using React
* The application must be hosted and accessible online
* The application must retrieve, parse and display relevant sensor data
* The application must include options for visualizing the sensor data history
* The application must have a responsive user interface
* All customer related features of the system must be managed through the application