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, machine learning, frontend), **and** will be decided in onsultation 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.

Security

In the project you can consider implementing encryption in the connection between the IoT-device and the cloud. This encryption can both be in the form of asymmetric encryption or symmetric encryption. Likewise, you can consider protecting your endpoints for the frontend with e.g. JSON Web Tokens.

Handin

You must hand in a project and 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 4 sub-sections. The report should merge again for results and discussion, conclusions and future work. Here is a diagram illustrating the structure:

Introduction		
Analysis		
Design Overview (inc. cloud)		
IoT Design	ML Data Exploration	Frontend Design
IoT Implementation	ML Preprocessing	Frontend Implementation
IoT CI/CD	ML Models	Frontend CI/CD
IoT Test		Frontend Test
Results and Discussion		
Conclusions		
Future Work		

Notice the added DevOps (continuous integration and continuous delivery) sections, that each team must include. A check list for stuff to think about for the contents of these sections could look like this:

- What kind of general DevOps considerations have you done, and how have you tried to plan DevOps into the project?
- Which DevOps tools have you used and why (or why not)?
- How, and to what extent, have you managed to make DevOps an integrated part of the general workflow?
- What effect has the DevOps tools/methods made in the project, and which parts worked well/less well?

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. You will have a separate hand in on WiseFlow for this, giving you some extra time to create the presentation during the exam period.

IoT

Design and implement an embedded application in C for an atmega2560 MCU that communicate over WIFI and interfaces to several sensors and actuators such as:

- Air Temperature
- Air Humidity
- Soil Humidity
- Light sensor
- Proximity sensor
- PIR sensor
- (CO2 sensor)
- Buttons

- Servo motor
- Water pump
- 7-segment display
- LEDs

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

Machine learning

The machine learning system should import and preprocess sensor data along with other pertinent information. Subsequently, the processed data must be fed into an ML model to generate descriptive insights or predictive outcomes regarding the system. The ML part should:

- Import, clean, and preprocess data.
- Utilize an appropriate ML model for classification, regression or clustering
- Export pertinent results

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 and predictions
- The application must include options for visualizing the sensor data history
- The application must have a responsive user interface (adapt well to at least the following screen widths: 576px, 768px and 1200px)
- All customer related features of the system must be managed through the application