Exam 2021

# 1.2 Task

1. Reguirements
   1. Recieve Modbus protocoll
   2. Lowpass filter
   3. Deadband
   4. Read config files
   5. Always listen for data
   6. Connect to several nodes
   7. Create CSV file for data
   8. UI
2. Use Case Diagram

Diagram

Description automatically generated

1. Domain Model

Diagram

Description automatically generated

1. To find a superclass, one needs to look at the classes needed for the system, and the ones that is used several times but with small differences, the subclasses will be the variations form this superclass. For this system, the class recieveing and hadling the data from the nodes would be a superclass, as the nodes wary in type and amount of data produced.
2. I would use a Three layer architecture. The application needs a Data layer to handle the incoming data from the nodes. A Logic layer to handle the data, filtering and foramtting. And a presentation layer where the User interface resides.
3. The Specification is given from the exam paper. Then theres the analysis step where the requirements are layed out, the use case diagram, and domain model is created. When the analysis step is done. We start in the construction phase, with one iteration for each of the use cases in the use case diagram. Additionally, through each iteration in the construction phase, the cases are tested. For each construction iteration, a fully dressed used case is created, before a interaction diagram is created. Assuming each iteration in the construction phase takes the same amount of time, one can estimate this to be 1 week. Making the project 6 weeks in length. Need to discuss the order in the construction phase for each use case.
4. No
5. Needs to Include Adress of nodes, timestamp, and sensor values.
6. Class structures. For Sensor Data. A superclass is created, for every node. Each sensor is also a class struct, wich contains value (float) and name(string), quality (bool). Node class contains one instance of each sensor. Number of sensor (int), quality (bool), desciption (string).
7. Fully dresses Use Case document, for recieving ModBus Data

|  |  |
| --- | --- |
| Use Case Name | Reading Configuration |
| Scope | Setup system for recieving data |
| Level | Logical Layer |
| Primary Actors |  |
| Stakeholders and Interests | norges energi og vassdragsdirektorat |
| Preconditions | XML config files are available |
| Succes Guarantee | Configurations form XML are stored in the system |
| Main Success Scenario | 1. Folder for config files are scanned 2. Map number of Config files (One for each node) 3. Save Name for each file 4. Read through one by one, and save |
| Extensions | 1. Folder is not found, or cannot be opened   3. Wring Filename  4. a) Cant open file  b) Wrong format of file |
| Special Requirements |  |
| Technology and Data Variations | XML files |
| Frequency of occurence | At startup |
| Miscellaneous |  |

1. Interaction Diagram

Diagram

Description automatically generated

Design patterns?

1. I would use GIT for version control. For each new use case developed a branch is created in the repo. The Use case is tested before its merged with the main branch
2. Class Diagram

Diagram

Description automatically generated

1. The use case will be tested by creating a test XML file, then see if the Class structure gets populated witht he right information.
2. Reverse engineering

The dataLayer doesnt read from a config file.