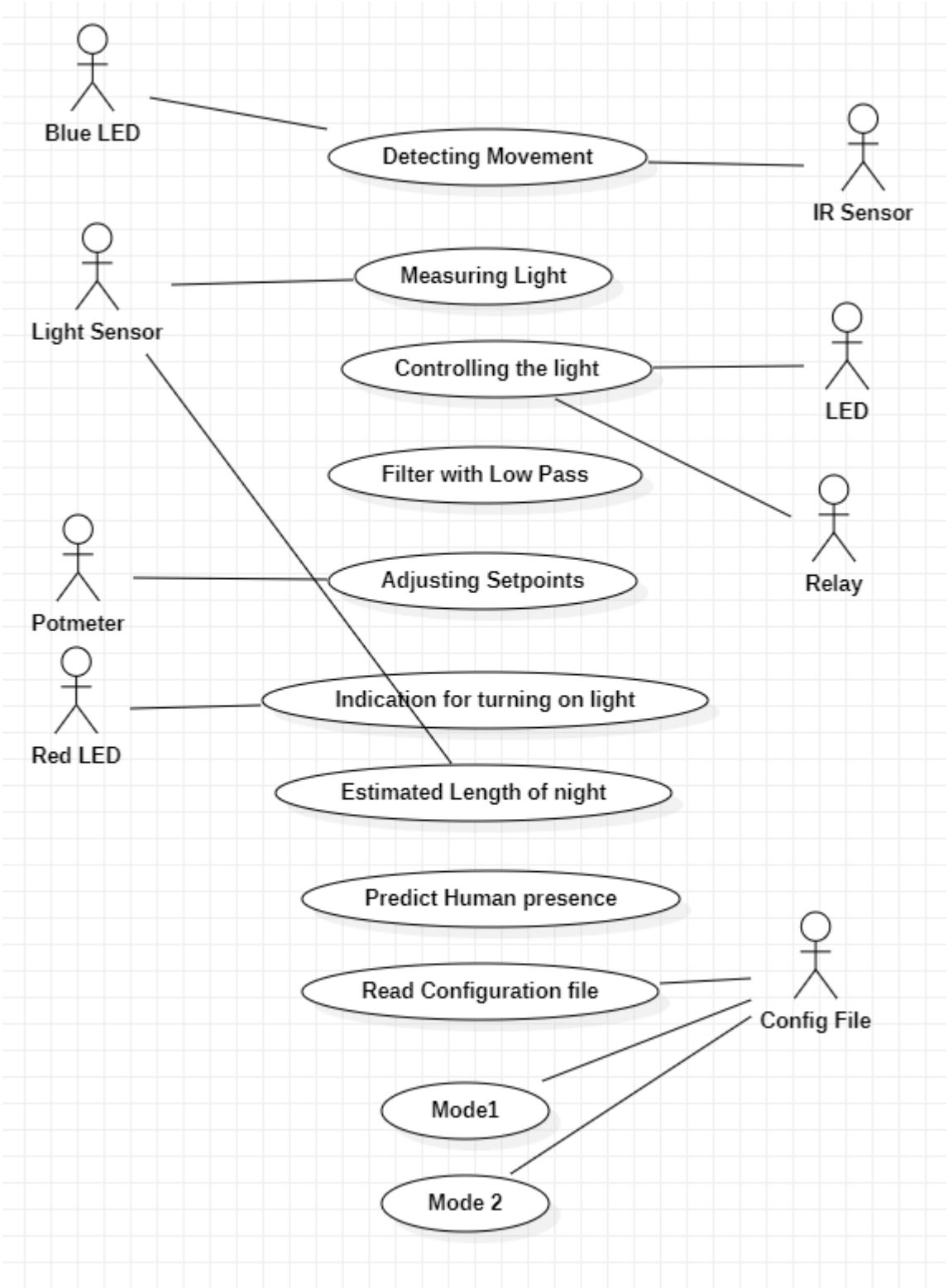
Exam 2022 Software Engineering

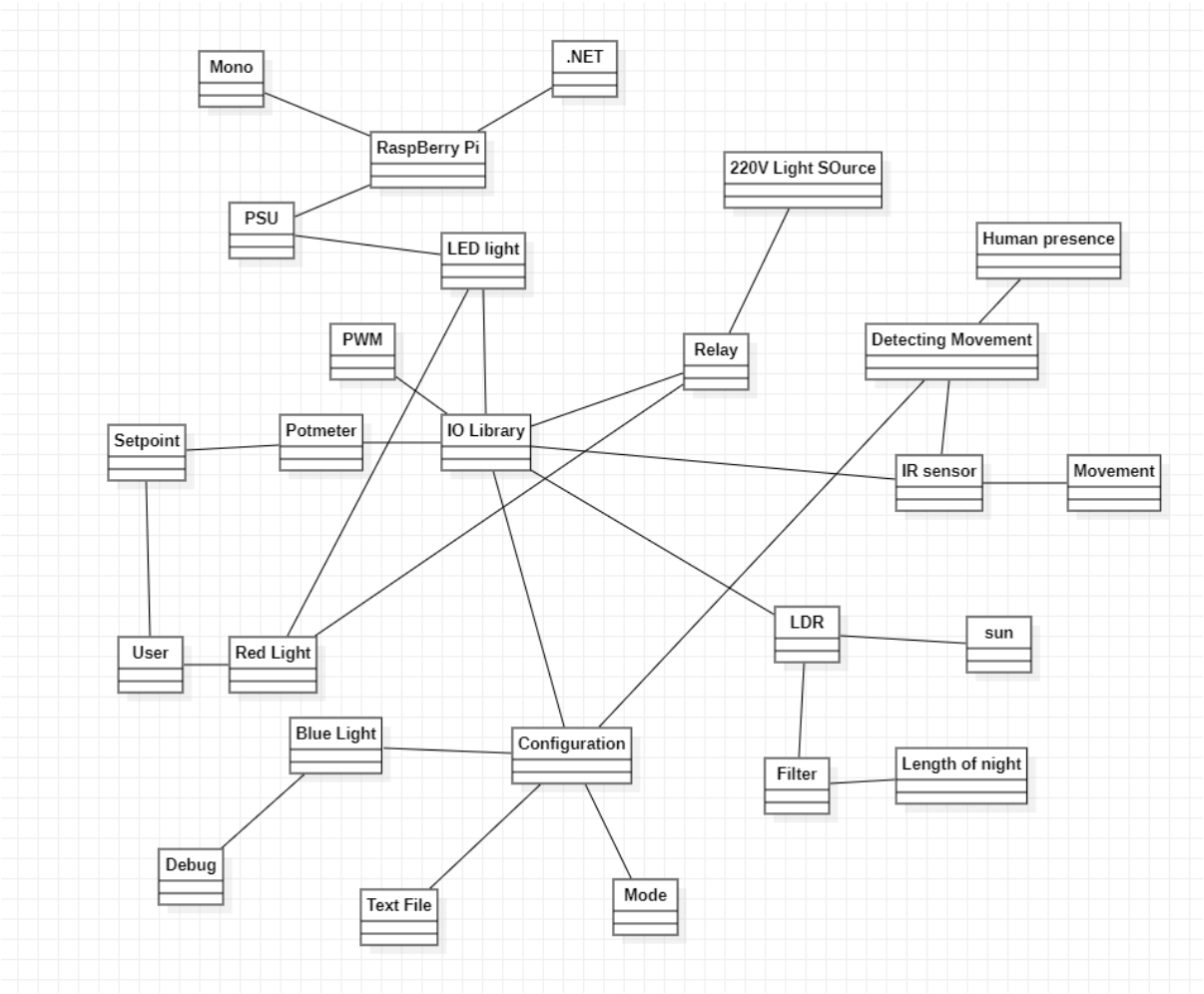
1. The requirements are collected using FURPS. The verbs are found in the specification and turned into the functional requirements.

Functional	Detecting Movement Measuring Light Controlling the light Filter with low pass Adjusting Setpoints Indication for turning on light Indication when movement is detected Estimated Lenght of night Predict human presence Read configuration from Text file
Usability	Setpoint adjustment with Potmeter Red and Blue LED indication
Reliability	Needs to read input data 24/7 Correctly scaled output values Needs to be handling being outside
Performance	Acts with a 15 min delay Hystereris with 10% margin for Setpint and light reading
Supportability	Gets config file from a text file Change operating mode
Implementation	Deployed to Raspberry PI

2. The use cases are from the functional requirements in the FURPS, the actors are made up from the non functional requirements. Mode1 and Mode 2 are also added as use cases



3. The domain model is supposed to fulfill the requirements. Most of the nouns in the specification is used to create the domain model



4. The Exam gives us the specification for the System. We can therefore jump directly to the Elaboration phase, and start with analysis. The first step of the analysis is to collect the requirements with FURPS, as executed in task 1. Then the use case diagram is created, based on the requirements. And the domain model is created based on the Specification, as seen in task 3. When the Elaboration phase is done, we move on to the construction phase. From the use case diagram. Theres 11 use cases in the use case diagrams, some of these can be combined when going through the iterations in the construction phase. The most important use case i Mode 2, FDUCD, interaction diagram and ssd is created for each iteration. Then Mode 1, Detecting Movement and predicting human presence combined, Measuring the light, filter and setpoint combined. Night duration estimation, controlling the light, and then read config file. The Iterations are ordered by the most critical first.
5. FDUCD

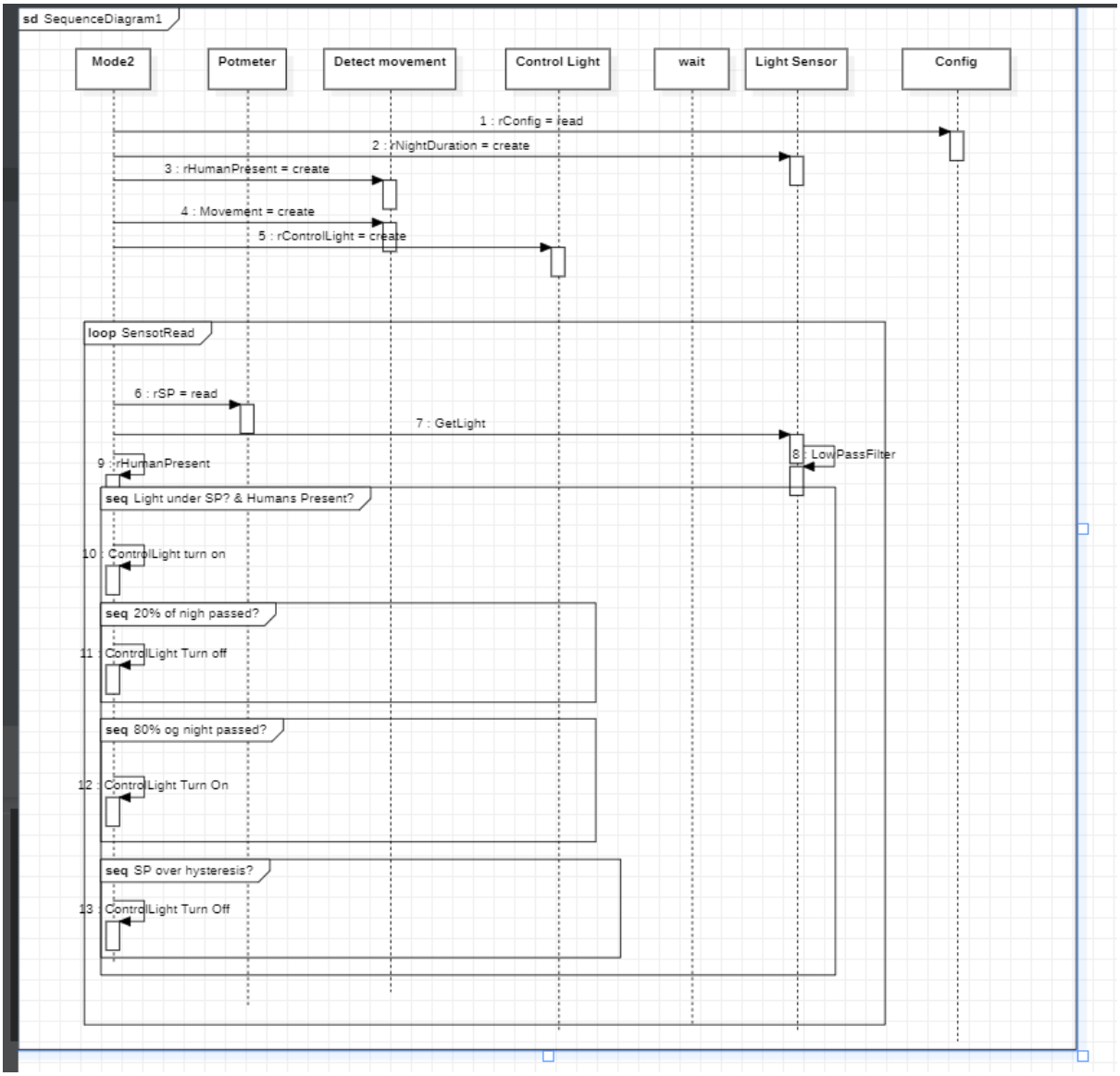
Use Case Name	Mode 2
Scope	Dynamic control
Level	Logical Layer
Primary Actors	
Stakeholders and Interests	
Preconditions	RP is set up
Succes Guarantee	The program detects when people are present, calculated duration of night, controls the light accordingly

Main Success Scenario	1. Get mode from configuration 2. Potmeter SPO 3. Gather Night duration (calculated in separate module) 4. Detect movement (3x?) within a certain time 5. If movement detected 3 times, humans are present 6. Humans present, turn on light after 15min delay of SP reached 7. Humans present, a. turn off after approx. 20% of night has passed b. Movement detected, Light 75% for 15 min 8. Humans Present, Turn on after approx. 80% of night has passed 9. Turn light off after reaching hysteresis of SP 10. Save duration of Night duration (Defined by the Setpoints, when night starts and stops)
Extensions	1. Cannot read file, goes to Mode 1 2. Not Connected, go to default in config 3. Night duration deviation? Keep old 4. Sensor fault? Mode 1 5. No apparent source of error 6. Unable to turn on light, Blue LED indication 7a. 7b. Movement not detected, External fault. 8. Timer is faulty, external fault 9. Hystereris too low, Use mode 1, blue light. Light sensor faulty, External fault 10. Duration is too far off other measurements. Keep old
Special Requirements	
Technology and Data Variations	Text file for config
Frequency of occurence	
Miscellaneous	

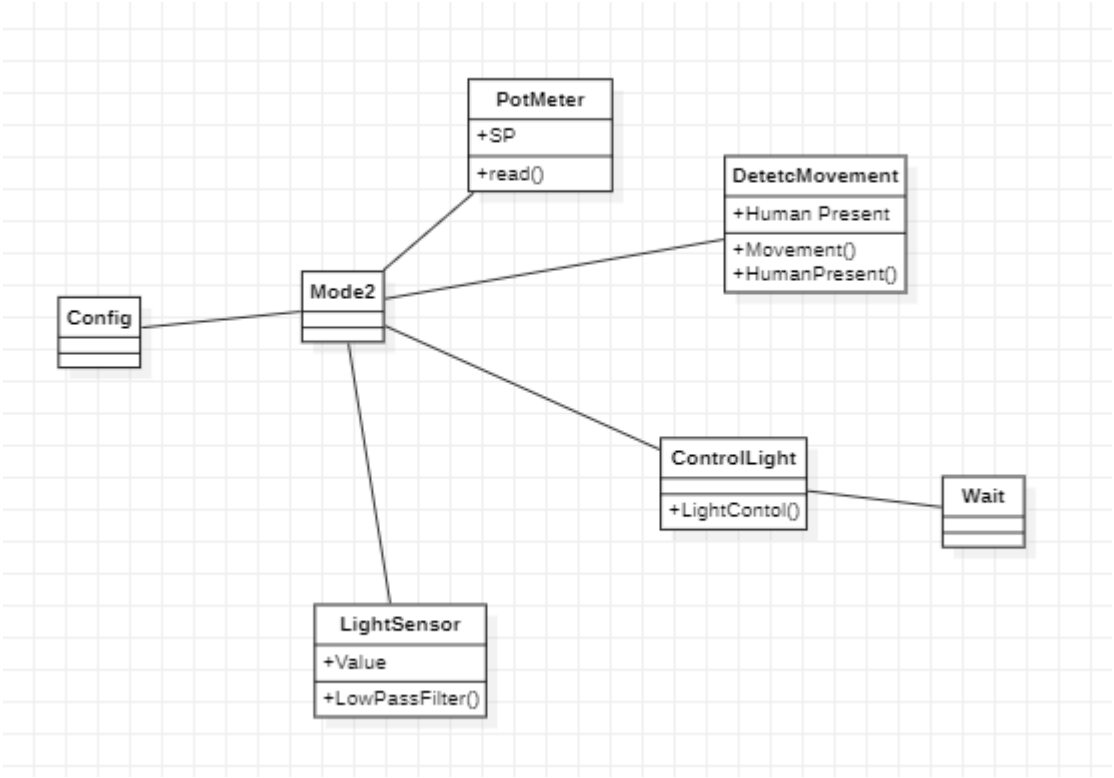
6. A SSD defines the use case as one object and shows how messages are being sent between the use case object and the actors for the system. While the Sequence diagram in

construction, shows in detail the way each use case works, which objects it contains and the flow of data within it.

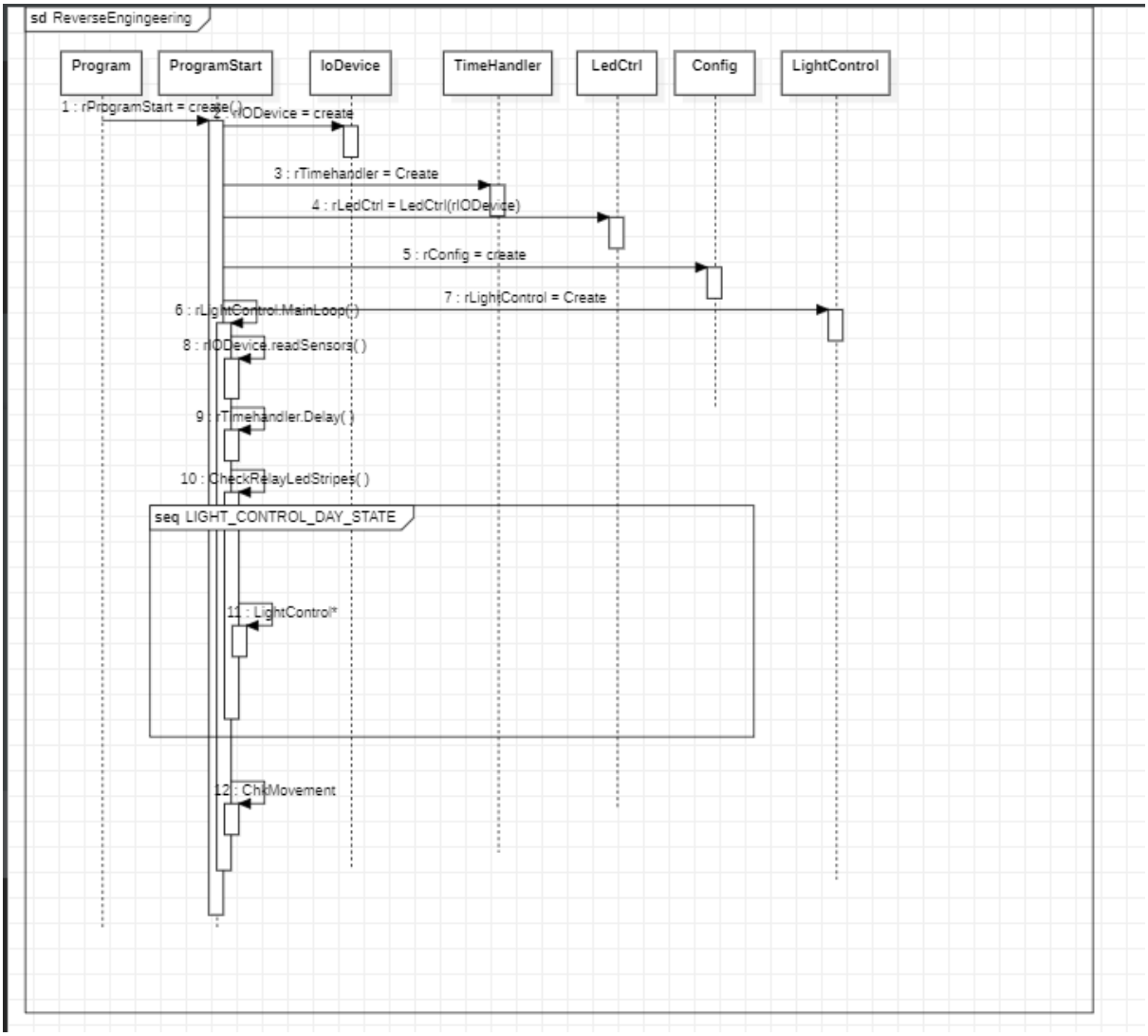
7. Due to the size of the Use case, only the main success scenario will be implemented. Ideally mode 2 should be embedded with threading and interrupts, this is hard to visualize in a sequence diagram.



8. When looking for superclasses, one need to look after objects that are reused. This keeps the code “dry”, meaning you write the code only once. Making it more clean and better structured. Although not very present in the design of this system. A Superclass for controlling the light should be implemented, with subclasses being what type of light source that’s being controlled.
9. Class diagram base on the Interaction diagram



10. There will be one class diagram for the entire system, while there will be one interaction diagram for each use case. In this case, there are 11 use cases, which means theres going to be 11 Interaction diagrams.
11. Interaction Diagram. Within the sequence of the LIGHT_CONTROL_DAY_STATE. If statements and switch cases are used to check IO according to the mode and control the lights accordingly. The logic contained in the sequence should be drawn in a separate sequence diagram. As the Interaction diagram drawn here, shows the structure of the application.



12. If you disregard the program class, as this is common for all the use cases. Then ProgramStart is the Controller, using the controller pattern. Is also using controller expert with the IODevice and Config. These classes have a responsility to obtain information. Low Coupling is used, the classes are only connected to program start class, meaning its easy to make changes to one class without having to change other classes. High cohesion is used, this is done by using a lot of methods within the classes, specifically for LightControl and IoDevice. Having a number of methods with descriptive naming. The program is also using Creator pattern, with the ProgramStart Class creating instances of the other classes.
13. To keep the config file as a text file. The CSV format could be used. Ideally xml or json is used for configuration, but this changes the format of the file from text. CSV can be handled as a normal text file in C#. The config file needs to include Which mode to use, IO addresses and scaling, Timezone, information about the light setup (LED, relay or both).
14. The existing use case diagram can eas