



Universitetet i Sørøst-Norge

Bacheloroppgave

SFH03201 Karoline M.

Predefinert informasjon

Startdato:	15-05-2020 09:00	Termin:	2020 VÅR
Sluttdato:	25-05-2020 12:00	Vurderingsform:	Norsk 6-trinns skala (A-F)
Eksamensform:	Bacheloroppgave		
SIS-kode:	222 SFH03201 1 BO 2020 VÅR		
Intern sensor:	Karoline Moholth Mcclenaghan		
Intern sensor:	Henning Gundersen		

Deltaker

Navn:	Erik Sung No Iuersen
Kandidatnr.:	8026
USN id:	134680@usn.no

Gruppe

Gruppenavn:	Gruppe 3 - AWC
Gruppenummer:	7
Andre medlemmer i gruppen:	Petter Riiber Elquin, Even Tyrholm, Marie Brettingen Johansen, Kristian Tørseth, Nina Patricia Heglund



Abstract

The petroleum industry is a tough and rigorous field, to stay in the lead technical advancements are crucial to reduce cost, increase efficiency, and ensure safety and stability.

HoloLens 2 and its advanced Augmented Reality technology is a brand-new field applicable to many existing technologies.

This project applies Augmented Reality to a new kind of user interface created by the team to control and test oil manufacturing technology and to test whether this application concept should be explored further.

Acknowledgments

First of we would like to thank TechnipFMC with Svein-Aage in the lead for an exciting and challenging task and for all the feedback, guidance, help and motivation we have gotten during the project. We are so thankful that you were as into the project and product as we were.

We would also like to thank Arnt-Ole from TechnipFMC, for great guidance with different technical tasks. Jørgen and Eivind have helped us out with understanding and setting up TechnipFMC's applications and we are very grateful for your fast replies and support.

Karoline and Henning have been big contributors to help shaping and commenting on the documentation throughout the process. You have always answered when we needed you and been flexible to our wishes. We thank you.

Table of contents

Abstract.....	2
Acknowledgments	3
Table of contents.....	4
Figure List.....	12
Abbreviations	17
1 Introduction.....	18
1.1 Background	18
1.2 Group Members	19
1.3 Roles and Responsibilities.....	21
1.3.1 Project Lead	21
1.3.2 Product Owner	21
1.3.3 Scrum Master	22
1.3.4 Technical Lead.....	22
1.3.5 Design Lead.....	22
1.3.6 Web Architect	23
1.3.7 Technical Advisor	23
1.3.8 Test Lead.....	23
1.3.9 Risk Lead	23
1.3.10 Document Lead	24
1.4 Technip FMC	25
1.5 Document Structure	26
1.5.1 References	26
1.5.2 Structure	27
1.5.3 Virus Scan.....	27

2	The Problem Domain	28
2.1	Scenario	28
2.2	Tools Provided by TechnipFMC.....	29
2.2.1	InControl	29
2.2.2	API.....	29
2.2.3	Simulator.....	29
2.3	Our Assignment.....	30
3	Tools.....	31
3.1	Unity	32
3.2	Microsoft HoloLens 2.....	32
3.3	Mixed Reality Toolkit.....	33
3.3.1	Input System	33
3.3.2	Standard Shader.....	34
3.3.3	Scene System	34
3.3.4	UX Building Blocks.....	34
3.4	Slack.....	36
3.5	Scrum	36
3.6	Teams.....	37
3.7	Messenger.....	37
3.8	Azure DevOps.....	37
3.9	Git.....	38
3.10	Adobe XD.....	39
3.11	LucidChart.....	39
3.12	Visual Studio and C#	39
3.13	Microsoft Office Packet.....	40
3.14	Blender	40

3.15	Adobe Photoshop	40
4	Methodology	41
4.1	Agile Development: SCRUM	42
4.2	Daily Scrum	43
4.3	User Stories	44
4.4	Product Backlog.....	45
4.5	Sprint	45
4.5.1	Sprint Planning	46
4.5.2	Sprint Backlog	47
4.5.3	Sprint review	47
4.5.4	Sprint Retrospective	48
4.5.5	Burndown Chart	48
4.6	Minimal Viable Product MVP	49
4.7	How Scrum Has Been Used.....	49
4.8	Group Dynamic.....	50
4.9	Meetings with Internal Supervisor.....	51
4.10	Risk Management	51
4.10.1	Identifying Risks	52
4.10.2	Risk Analysis and Evaluation	53
4.10.3	Risk Response Strategies	54
4.10.4	Monitor and review risk.	55
4.11	Testing.....	56
4.11.1	Acceptance criteria testing.....	56
4.11.2	Usability test	58
4.12	UX Design	59
4.12.1	AR best practise – colour and light	60

- 5 Modelling the Product61
 - 5.1 Requirements.....61
 - 5.1.1 Non – functional Requirements61
 - 5.1.2 Functional Requirements61
 - 5.2 Hierarchy.....62
 - 5.3 Use Case Diagrams63
 - 5.3.1 Choose User63
 - 5.3.2 Manage XT64
 - 5.3.3 XT Operation65
 - 5.3.4 Hand Menu66
 - 5.4 User stories67
 - 5.4.1 Choose admin67
 - 5.4.2 Connect to XT67
 - 5.4.3 Display XT Schematic.....68
 - 5.4.4 Open/Close Valves68
 - 5.4.5 Sensor Data68
 - 5.4.6 Hand Menu69
 - 5.4.7 Display HCR Schematic69
 - 5.5 Sequence diagram70
 - 5.5.1 Choose Admin – Feature 1.1.....70
 - 5.5.2 Manage XT – Feature 2.1.....71
 - 5.5.3 See XT Schematic - Feature 3.1.....72
 - 5.5.4 Valve Operation – Feature 3.273
 - 5.5.5 Sensor Information – Feature 3.4, 3.5 and 3.674
 - 5.5.6 Hand menu - Feature 4.1, 4.3 and 4.4.....75
 - 5.6 Flow Chart76

6	The Implementation	77
6.1	Technical	79
6.1.1	Global Manager.....	79
6.1.2	Choose user.....	80
6.1.3	Manage XT	84
6.1.4	XT Schematic.....	94
6.1.5	Valve Control.....	100
6.1.6	Sensor Information.....	110
6.1.7	Hand Menu	119
6.2	User Interface.....	128
6.2.1	Design	128
6.2.2	Interactable Elements	130
7	Results.....	132
7.1	Have we met the requirements?	132
7.1.1	Usability Test with Stakeholder	133
7.2	Challenges.....	134
7.2.1	New Technology.....	134
7.2.2	Covid – 19 Virus.....	135
7.2.3	GIT	136
7.2.4	Only one HL2 - testing	137
7.2.5	Building in UNITY to HoloLens2	137
7.2.6	Building and exporting models from Blender to Unity	138

8	Conclusion	141
8.1	Future Work	143
8.1.1	True multi-user.....	143
8.1.2	Implement an observer pattern for the Software communication.	143
8.1.3	Implement more MRTK features	143
8.1.4	Expand application to include PCR-X and D	144
8.1.5	Oil Flow	144
8.1.6	Indication if values is critical.....	144
8.1.7	Drawing program for json configuration	144
9	APPENDIX A - DESIGN DOCUMENTATION	145
9.1	Persona	146
9.2	UX Design	147
9.3	UI Design Suggestions.....	148
9.4	Final Sketch for UI Design	149
9.5	UX design ideas – Brainstorming	160
9.6	Suggestions of UI – Brainstorming	161
9.6.1	First UI Design Suggestion	161
9.6.2	Second UI Design Suggestion.....	164
9.6.3	Third UI Design Suggestion	169
9.6.4	Fourth UI Design Suggestion.....	170
9.6.5	Fifth UI Design Suggestion	172
10	APPENDIX B – TEST DOCUMENTATION	176
10.1	Acceptance Criteria Test templates.....	177
10.2	Completed tests.....	178
10.3	Rejected tests	192
10.4	Usability test.....	193

10.4.1	Information given to the participant before the test starts:.....	193
10.4.2	Test 1, 08.05.2020, Participant 1:	195
10.4.3	Test 2, 22.05.2020, Participant 2:	199
11	APPENDIX C – DECISIONS.....	202
11.1	Schematic Decisions.....	203
11.1.1	Procedural drawing of Christmas tree schematic.....	203
11.1.2	SplineMesh Asset	208
11.1.3	Why we chose static pipes with json file configuration.....	208
11.2	Sensor Considerations	209
11.2.1	History.....	210
11.3	Decision on where to place the hand menu	211
11.4	Hashing algorithm: BCrypt	212
12	APPENDIX D - RISK DOCUMENTATION	213
13.1	Risk Template	214
13.2	Risk Evaluation – Matrix:.....	215
13.3	Sprint 1	217
13.4	Sprint 2	230
13.5	Sprint 3	231
13.6	Sprint 4	233
13.7	Sprint 5	234
13.8	Sprint 6	235
13.9	Sprint 7	236
13.10	Risk Register	237

- 14 APPENDIX E - SPRINT DOCUMENTATION..... 242
 - 14.1 Project Plan.....243
 - 14.2 Sprint retrospective – Sprint 1 244
 - 14.3 Sprint retrospective – Sprint 2246
 - 14.4 Sprint retrospective – Sprint 3 248
 - 14.5 Sprint retrospective - Sprint 4249
 - 14.6 Sprint retrospective - Sprint 5 251
 - 14.7 Sprint retrospective - Sprint 6252
 - 14.8 Sprint retrospective - Sprint 7 253
 - 14.9 Sprint retrospective - Sprint 8254
- 15 APPENDIX F – APPLICATION USER GUIDE 255
 - 15.1 Choose User.....256
 - 15.2 Manage XT 256
 - 15.3 XT Schematic256
 - 15.4 Hand Menu256
- 16 APPENDIX G - WORKING HOURS 257
- 17 References..... 259

Figure List

Figure 1 - PCR-X, PCR-D, HCR	29
Figure 2 - Simulator	29
Figure 3 – System Overview.....	30
Figure 4 - Unity Icon [3]	32
Figure 5 -Mixed Reality Logo [6]	33
Figure 6 - Button prefabs	35
Figure 7 - Slack logo [12].....	36
Figure 8 - Scrum Icon [13].....	36
Figure 9 - Microsoft teams logo [14]	37
Figure 10 - Facebook messenger logo [15].....	37
Figure 11 - Azure DevOps logo [16].....	37
Figure 12 - git logo [18].....	38
Figure 13 - Git workflow [18]	38
Figure 14 - Adobe XD logo [19]	39
Figure 15 - Lucidchart logo [20]	39
Figure 16 - Development tool and language logo [21] [22].....	39
Figure 17 - Office packet [23] [24] [25] [26]	40
Figure 18 - Blender logo [52].....	40
Figure 19 -Adobe Photoshop logo [53].....	40
Figure 20 – Methodology approach	41
Figure 21 - Scrum process model [28]	43
Figure 22 - Example of a burndown chart used in scrum	48
Figure 23 - Risk management process model.....	51
Figure 24 - Qualitative determination of level of risk matrix	53
Figure 25 - A User Story with coherent acceptance criteria from taskboard in DevOps	56
Figure 26 - Example user story that has been tested and fulfilled its acceptance criteria	57
Figure 27 - Colour palette in HL2 view	60
Figure 28 - Snapshot from DevOps of Epics and associated Features	62
Figure 29 - Use Case: Choose User	63
Figure 30 - Use Case: Manage XT	64

Figure 31 - Use Case: XT Operation	65
Figure 32- Use Case: Hand Menu	66
Figure 33 - Sequence Diagram: Choose User.....	70
Figure 34 - Sequence Diagram: Manage XT	71
Figure 35 - Sequence Diagram: XT Schematic.....	72
Figure 36 - Sequence Diagram: Valve Operation	73
Figure 37 - Sequence Diagram: Sensor Information	74
Figure 38 - Sequence Diagram: Hand Menu	75
Figure 39 - Flow Chart: System Overview.....	76
Figure 40 - Technical Areas	77
Figure 41 - Class Diagram: GlobalManager.....	79
Figure 42 - Choose User Menu.....	80
Figure 43 - Pin Code Menu.....	81
Figure 44 - Class Diagram: PinCode.....	82
Figure 45 – Sequence Diagram: Pin is typed in	83
Figure 46 - Manage XT View with XT Prefabs	84
Figure 47 - Class Diagram: XmasTrees.....	85
Figure 48 - Class Diagram: PrefabXT.....	86
Figure 49 - Sequence Diagram: Manage XT	87
Figure 50 - Structure of XT list text file.....	87
Figure 51 - Add New XT Menu	88
Figure 52 - Sequence Diagram: Add XT	88
Figure 53- Edit XT Menu	89
Figure 54 - Class Diagram: Edit Menu.....	90
Figure 55 - Sequence Diagram: Edit XT.....	91
Figure 56 - Sequence Diagram: Remove XT	92
Figure 57- Sequence Diagram: Connect To XT.....	93
Figure 58 - Populated XT Schematic (Pyxis).....	94
Figure 59 - Class Diagram: XT Schematic	95
Figure 60 - Empty XT Schematic.....	96
Figure 61 - Template piping from TechnipFMC software.....	96
Figure 62 - Json structure	97

Figure 63 - Populated XT Schematic (Cassiopea)	98
Figure 64 - Sequence Diagram: XT Schematic.....	99
Figure 65 - Class Diagram: ValvePrefab	100
Figure 66 - Class Diagram: ChokePrefab.....	101
Figure 67 - Class Diagram: CovPrefab.....	102
Figure 68 - Class Diagram: CIMV	103
Figure 69 - Flow Diagram Open/Close Valve	104
Figure 70 - Flow Diagram Adjusting Choke.....	105
Figure 71 - Valve Model Closed.....	106
Figure 72 - Valve Model Open.....	106
Figure 73 - Valve Menu.....	106
Figure 74 - Security Face Plate	106
Figure 75 - Choke Prefab Model.....	107
Figure 76 - Choke Menu.....	107
Figure 77 - Security Face Plate	107
Figure 78 - CIMV Model	108
Figure 79 - CIMV Menu	108
Figure 80 - Security Face Plate	108
Figure 81 - COV Model.....	109
Figure 82 - COV Menu.....	109
Figure 83 - Class Diagram: Sensor	111
Figure 84 - Sequence Diagram: Sensor Data.....	112
Figure 85 - Active Values Small	113
Figure 86 - Active Values Small Detailed	113
Figure 87 - Active Values Small Detailed History	113
Figure 88 - Active Values Medium Detailed	114
Figure 89 - Active Values Medium	114
Figure 90 - Active Values Medium Detailed History	114
Figure 91 - Active Values Large	115
Figure 92 - Active Values Large Detailed	115
Figure 93 - Active Values Large Detailed History	115
Figure 94 - Sequence Diagram: SensorManager.....	116

Figure 95 - Sequence Diagram: DetailedInfo	117
Figure 96 - Sequence Diagram: History	118
Figure 97 - Hand Menu seen after connecting to XT	119
Figure 98 - Hand Menu seen before connecting to XT.....	119
Figure 99- Class Diagram: Hand Menu	120
Figure 100 - Sequence Diagram: Clear View.....	121
Figure 101 - Sequence Diagram: Home	122
Figure 102 - ReadMe displayed.....	122
Figure 103 - Populated HCR Schematic	124
Figure 104 - Class Diagram: HCR Schematic	125
Figure 105 - Empty HCR Schematic	126
Figure 106 - Sequence Diagram: Population of HCR Schematic	127
Figure 112 - Indications open/close valves.....	129
Figure 113 - showing table on how FPS drops when polygons increase [39]	138
Figure 114 - schematic piping. Colours are for separating in Blender	139
Figure 115 - showing model of finished valve	140
Figure 116 - Choke design.....	140
Figure 117 - Visualization of oil flow	144
Figure 118 – Sketch of choose user.....	149
Figure 119 - Sketch of Pin Code is typed in.....	150
Figure 120 - Sketch of correct Pin is typed in and add XT button is pressed in	151
Figure 121 - Sketch of form is filled in and XT is added.....	152
Figure 122 - Sketch of edit button is pressed	153
Figure 123 - Sketch of the edit form	154
Figure 124 - Sketch of the connect button is pressed.....	154
Figure 125 - Sketch of the XT Schematic	155
Figure 126 - Sketch of choke is being pressed	156
Figure 127 - Sketch of valve being pressed	157
Figure 128 - Sketch of security plate for valve operation	158
Figure 129 - Sketch of sensor is being pressed	159
Figure 130 - Sketch of Hand Menu.....	159
Figure 131 - Sketch of icons	160

Figure 132 - Sketch of first UI suggestion 1	161
Figure 133 - Sketch of first UI suggestion 2	162
Figure 134 - Sketch of first UI suggestion 3	163
Figure 135 - Sketch of second UI suggestion 1	164
Figure 136 - Sketch of second UI suggestion 2	165
Figure 137 - Sketch of second UI suggestion 3	166
Figure 138 - Sketch of second UI suggestion 4	167
Figure 139 - Sketch of second UI suggestion 5	168
Figure 140 - Sketch of third UI suggestion 1.....	169
Figure 141 - Sketch of forth UI suggestion 1.....	170
Figure 142 - Sketch of forth UI suggestion 2.....	171
Figure 143 – Sketch of fifth UI suggestion 1	172
Figure 144 - Sketch of fifth UI suggestion 2.....	173
Figure 145 - Sketch of fifth UI suggestion 3.....	174
Figure 146 - Sketch of fifth UI suggestion 4.....	175
Figure 147 - 2D Array Coordinates	204
Figure 148 - how a simple lvi design can be drawn	205
Figure 149 - Connection Intersection	206
Figure 150 - thought process when making procedural HMI	207
Figure 151 - left hand showing "zones" for hand menu position	211
Figure 152 - Working Hours	258

Abbreviations

AR – Augmented Reality

API - Application Programming Interface

ASD - Acoustic Sand Detector

CIMV - Chemical Injection Metering Valve

COV – Change Over Valve

EFAT - Extended Factory Acceptance Criteria

EP - Erosion Protection Sensor

HCR – Hydraulic Control Router

HL2 - HoloLens 2

MPFM - Multiphase Flow Meter

MRTK - Mixed Reality Toolkit

PCR-D – Power communication router - Down Hole

PCR-X – Power communication router

PT/TT – Pressure Transmitter/Temperature Transmitter

SIT – System Integration Testing

UI – User Interface

UX – User Experience

VR – Virtual Reality

VM – Virtual Machine

XT – Xmas Tree

PBI – Product Backlog Item

P&ID – Piping and Instrumentation Diagram

1 Introduction

This chapter introduces the reader to the team and their roles in this project, employer and documentation structure.

1.1 Background

The team consists of six students studying computer engineering with a specialization in virtual computer systems at the University of South-East Norway campus Kongsberg. We got in touch with Svein-Aage Opsal from TechnipFMC who has a great interest for AR. He created a project description and at the end of august 2019 we had our first meeting with TechnipFMC.

1.2 Group Members



Even Tyrholm

Project Lead and Product Owner

even.tyrholm@augmentedwellcontrol.com



Erik Iversen

Technical Advisor and Website Architect

erik.iversen@augmentedwellcontrol.com



Petter Riiber Elgvin

Technical Lead

petter.elgvin@augmentedwellcontrol.com



Nina Heglund

Test Lead and Budget Lead

nina.heglund@augmentedwellcontrol.com



Marie Brettingen Johansen

Risk Lead and Design Lead

marie.johansen@augmentedwellcontrol.com



Kristian Tørseth

Scrum Master and Documentation Lead

kristian.torseth@augmentedwellcontrol.com

1.3 Roles and Responsibilities

1.3.1 Project Lead

Oversees the management of the project on a high level. He is also responsible for communicating with the external and internal people of interest. His role is not to make high level decisions but guide the group towards the final goal and foresee obstacles. He is also responsible for contacting the internal or external supervisor for help.

1.3.2 Product Owner

Responsible for maximizing the value the scrum team delivers. In a normal scrum process the product owner is solely responsible for putting tasks in the product backlog. In our case, since the product owner is a fellow student and not a representative from the company, we've decided that we will make user stories for the backlog as a group. These will later on be pitched for the company where we will ask for feedback. The product owner is responsible for ensuring that the team is working on the right things at the right time. This means that the product owner must rank the different user stories in the product backlog and make sure that every task is completed with a certain quality. By keeping an organized and ranked product backlog the product owner is telling the team what should be prioritized, so that the team will be successful in delivering the project on time.

1.3.3 Scrum Master

The role is to enforce the agile framework during the project. The scrum master should also have significant more knowledge about how scrum works and should help educate and teach the team members about the scrum values, educates the team about practices like stand-up meeting or the sprint retrospective.

It's important for the scrum master to make sure every idea and voice are heard during the project. They need to facilitate so that the quiet one in the room has their opportunity to raise their voice and pitch the ideas they have, as well as the more influential ones, on the team

1.3.4 Technical Lead

Responsible for making sure that the technical aspects of the project are used in accordance with their purpose and in a proper technical manner. His tasks may include: Enforcing technical standards such as coding standards, tools, and platforms. Reviewing code to ensure quality, clarity and have an overall view of the application, in addition to ensure that the technical documentation is well documented, especially that the system and software architecture is clear and easy to follow.

1.3.5 Design Lead

Responsible for overseeing the UX design work of the system throughout the project period. This includes both the Graphical User Interface and Usability. Ensuring that the user is able to execute features of the system in an easy and efficient manner. Design lead is also responsible to analyse results from UX test and Usability test in order to get a better understanding of what factors of UX that needs more attention and how easy it is to use the application. Design choices must be documented, and all members of the group needs to have the same view of point at what the customer needs.

1.3.6 Web Architect

Responsibilities include coordinating the development, implementation and maintenance of the team website. Accountable for continuous publishing of content throughout the project period.

1.3.7 Technical Advisor

Responsible for making a prototype and gaining extensive knowledge on the following technical topics MRTK, Unity and HL2. Expected to provide training, assistance, and advice on these topics during the project. Technical Advisor is also responsible for building and deploying the application to HL2.

1.3.8 Test Lead

Have a holistic view over the different tests that must be conducted throughout the project life. Through testing we can verify and validate that the product is built right and that the right product is being built. Managing verification consists of traceability in documentation from requirements and throughout the whole process. Managing validation is making sure the product meets the requirements and the design specifications. We are going to conduct acceptance criteria testing of our user stories and usability tests of our UX design.

1.3.9 Risk Lead

Responsibility is to analyse identified risks and based on the findings implement a strategy to mitigate those risks. The implemented strategy will also help to monitor and to prioritize the different risks and reduce the likelihood of occurrence and/or consequence. The risk manager needs to take the whole project into consideration when identifying risks and all members has the responsibility to notify the risk manager when new risks is found.

1.3.10 Document Lead

Responsible for document structure in form of defining standards, making sure a hierarchical structure is maintained in Microsoft Teams archive and defining templates that will be used for documentation.

The document lead also has to quality check documents before they are considered done, and make sure that templates and standards are correctly followed and used properly. The document manager should not have the sole responsibility, but it is the document managers job to make sure that this is completed.

1.4 Technip FMC

TechnipFMC is one of the global leaders in oil and gas projects, technologies, systems, and services. They provide their clients with deep expertise across onshore/offshore and subsea projects. The company has approximately 37 000 employees spread across more than 45 countries.

TechnipFMC Norway Controls and Automation develops all the subsea systems where signal, data, automation, and user interface are the most important points of interest. TechnipFMC has been a leading supplier of control systems for subsea installations ever since oil production started on the Norwegian continental shelf.

The latest generation of technology that TechnipFMC's been developing, has been under development for the past three years and is soon ready for projects in Norway, Italy, Brazil and Australia. As part of the generational shift in technology, emphasis has been making the systems more efficient, robust, easier to maintain and not least, easier and safer to operate in a new way.

1.5 Document Structure

1.5.1 References

References will be referenced in the IEEE style to keep the focus on the content, but with the option to investigate the sources further. IEEE is a reference type typically used in engineering and information documentation. [1]

1.5.2 Structure

1 – Introduction

This chapter introduces the reader to the team and their roles in this project, employer and documentation structure.

2 - The Problem Domain

Explains the given assignment and the given tools from TechnipFMC. It is meant to give the reader a deeper understanding of the system environment.

3 - Tools

This chapter will explain what tools, processes, and technologies we use to accomplish the goal of this project.

4 – Methodology

Describes why and how we used the tools and processes we used.

5 – Modelling the Product

The purpose of this chapter is to present a high-level view of the application.

6 – The Implementation

The purpose of this chapter is to explain how the application is built and assembled.

7 – Result

In this chapter we will discuss the challenges and obstacles we encountered and how we worked to overcome those obstacles.

8 – Conclusion

Discusses how the project went, challenges, our handling of issues and what future implementations can be added to improve the application.

1.5.3 Virus Scan

Before every delivery it is mandatory to scan the file delivered electronically for malware. This will be done using Windows Defender which scans the file for malware. A report will be handed in with the documents.

2 The Problem Domain

This chapter will explain the given assignment and the given tools from TechnipFMC. It is meant to give the reader a deeper understanding of the system environment.

2.1 Scenario

The control system has a critical role in oil and gas production and system failures or misunderstandings of functionality can in worst case lead to major environmental disasters and loss of life.

A project delivery for Controls & Automation goes through a time-consuming and complicated process that deals with extensive tests and verifications regulated by the authorities, customers, and suppliers. The control system is made up of various technologies such as software, electronics, power management, networking, hydraulics, and mechanics. There are also many disciplines involved to make a complete system work according to all requirements.

It goes without saying that the process requires a lot of equipment, configuration, and expertise. TechnipFMC constantly strives to improve all processes so that implementation can be as secure, efficient and as cost optimized as possible. This is most often done through process improvements, but also through access to new technology. An important milestone in deliveries is to run a complete system test, in which all functions and instruments connected to the control system operate according to the customer's specification.

2.2 Tools Provided by TechnipFMC

2.2.1 InControl

InControl is an app with GUI made by TechnipFMC to test against real life components. It is also made to be able to test software, control and set values to see if, and how this impacts the system. InControl can in theory be used to control a XT but is in real life practices used on smaller subsystems of the XT.

2.2.2 API

TechnipFMC made an API to make it easier to communicate with the XT and the different computers on the XT.

2.2.3 Simulator



Figure 1 - PCR-X, PCR-D, HCR

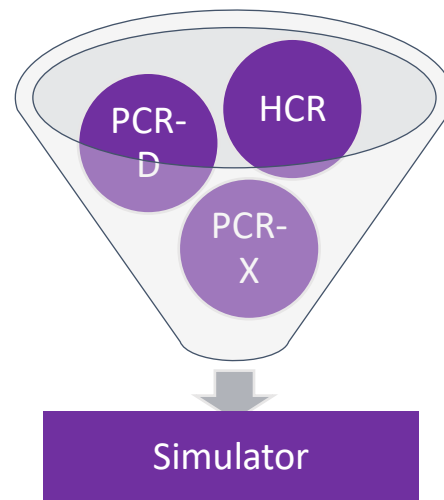


Figure 2 - Simulator

TechnipFMC has a XT Simulator. The simulator represents the three computers usually mounted on the XT and can therefore simulate how an XT behaves. We will use the simulator instead of a XT because of physical restrictions.

2.3 Our Assignment

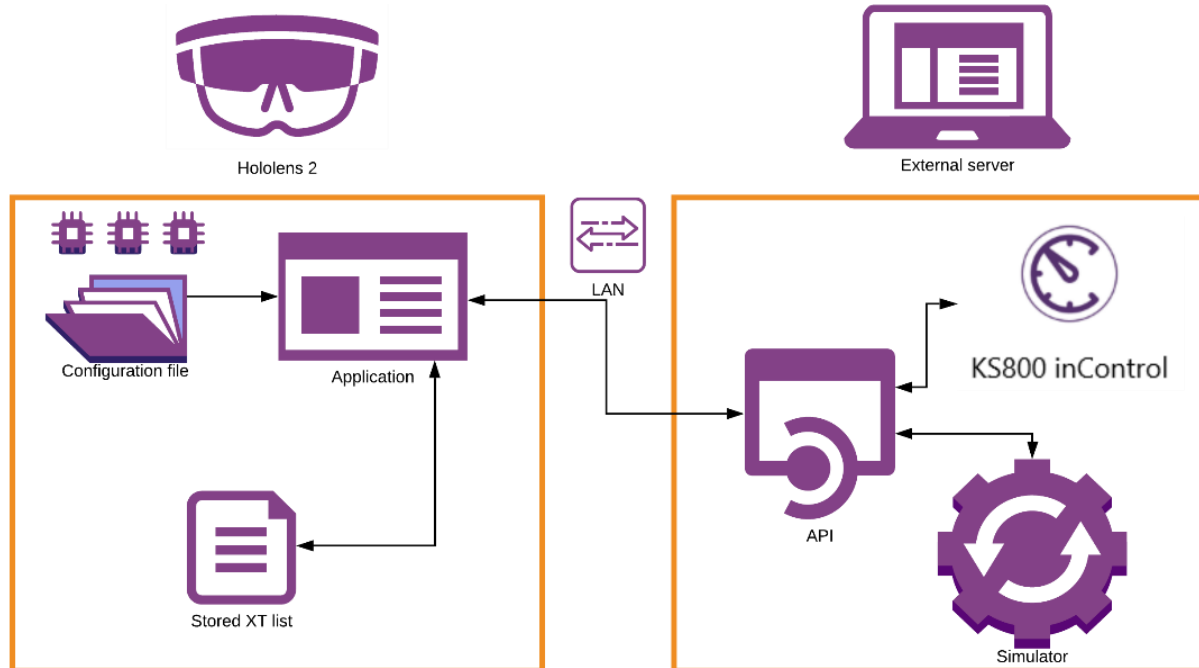


Figure 3 – System Overview

The goal of our assignment is to develop an alternative user interface application using Augmented Reality technology to operate the control system and read connected component information. In addition, the user interface must be capable of retrieving and visualizing historical data for selected components.

The intention is to test whether this technology can make testing more intuitive for the user without being reliant on specific computers or configurations.

TechnipFMC has given us access to a simulator connected to an API which we are going to use to access simulated data of a XT. KS800 inControl is a UI application connected to the API which we can use to verify that changes we make with our application takes effect and are correct.

Our HL2 application therefore has some requirements to function; a LAN connection, a chosen XT and a configuration file for translating the Simulator components into something visual that the user can interact with.

3 Tools

This chapter will explain what tools, processes and technologies we use to accomplish the goal of this project.

Tools used for Communication:

- Teams
- Messenger
- Slack
- Discord

Tools used for Development:

- Unity
- Teams
- Visual Studio
- MRTK
- Lucidchart
- Photoshop
- Blender
- Git

Tools used for Documentation:

- Word
- Power Point
- Excel
- Teams
- DevOps
- Git

3.1 Unity

Unity is a real-time development platform. It will be the main platform for developing our application. The code used in our application and run in Unity will be written in C# with Visual Studio. This code will be stored on GitHub for storage and version control.

“Unity Technologies is the creator of Unity, a flexible and high-performance end-to-end development platform used to create rich interactive 3D and 2D experiences. Unity's powerful graphics engine and full-featured editor serve as the foundation to develop beautiful games or apps and easily bring them to multiple platforms: mobile devices, home entertainment systems, personal computers, and embedded systems. Unity also offers solutions and services for creating games, boosting productivity, and connecting with audiences including the Unity Asset Store, Unity Cloud Build, Unity Game Performance Reporting, Unity Ads, and Unity Everyplay. Unity Technologies serves over 700,000 monthly active developers including large publishers, indie studios, students and hobbyists around the globe. For more information, visit [Unity](#) for more information [2]

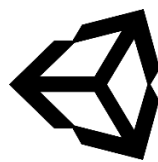


Figure 4 - Unity Icon [3]

3.2 Microsoft HoloLens 2

HoloLens 2 is Microsoft newest mixed reality glasses. HoloLens 1 is the old version, capable of mixed reality functions like a virtual object floating in the room you are currently in and interacting with this object. It has its own computer and operating system, which means that it's completely wireless.

HoloLens 2 improves on nearly all aspects of its predecessor, increasing speed, accuracy and field of view as well as adding eye tracking and eye recognition. Meaning that once you put the headset on you are immediately recognized and logged in. More intuitive gestures are also included in HL2 based upon what users expect will happen when you reach out to touch an object.

3.3 Mixed Reality Toolkit

The Mixed Reality Toolkit (MRTK) is a cross-platform toolkit for building Mixed Reality experiences for Virtual Reality (VR) and Augmented Reality (AR). [4]

The Mixed Reality Toolkit is a package for Unity which aids developers using the HoloLens 2 in developing mixed reality experiences. [5]



Figure 5 -Mixed Reality Logo [6]

Being a Microsoft-driven project the MRTK provides a set of components and features, that allows for easier and faster development. We are using several features and components in our project. The MRTK is well documented with feature descriptions and tutorials. Everything is available online, so we will only provide a short summary of the most important components and features used by us. MRTK consists of several Unity packages

- Foundation
- Extensions
- Examples
- Tools

The most important one is the Foundation package which include the core functionality.

3.3.1 Input System

A feature of the MRTK is the input system. The input system controls how the user interacts with the objects. HL2 gives you the ability to control an application with your bare hands. You can also control with your voice, and eyes. There's also the ability to use far pointers to control objects that are far away. The system is quite extensive, with a lot of customization. When an event is fired, the system detects which object was interacted with and triggers the Input Handler attached to that [7]

3.3.2 Standard Shader

MRTK comes with its very own shader. HoloLens 2 is not nearly as powerful as an ordinary computer, that's why limiting resource use is essential. The MRTK Standard Shader is specially designed for mixed reality devices like the HoloLens 2 and will provide optimal performance for our application. [8]

3.3.3 Scene System

Unity bases its development around scenes. A scene consists of several Game Objects, scripts etc. MRTK comes with a scene management system which enables the user to easily switch between scenes. We will use this system to navigate our application. There will be one default manager scene which contain the main MRTK Game Objects. Then you can load and unload scenes containing other content. Our prototype uses this system. There's one scene for login – and one scene for the XT display. [9]

3.3.4 UX Building Blocks

MRTK provides several UX building blocks which are especially designed for AR. There's no haptic feedback when touching models so, providing feedback in the form of visuals, animations and sound is important.

There are several button prefabs which we will use. They are well designed, and they provide visual and audio feedback. [10]

Collider based buttons

<p>PressableButtonHoloLens2</p> <p>HoloLens 2's shell-style button with backplate which supports various visual feedback such as border light, proximity light, and compressed front plate</p>	<p>PressableButtonHoloLens2Unplated</p> <p>HoloLens 2's shell-style button without backplate</p>	<p>PressableButtonHoloLens2Circular</p> <p>HoloLens 2's shell-style button with circular shape</p>
<p>PressableButtonHoloLens2_32x96</p> <p>Wide HoloLens 2's shell-style button 32x96mm</p>	<p>PressableButtonHoloLens2Bar3H</p> <p>Horizontal HoloLens 2 button bar with shared backplate</p>	<p>PressableButtonHoloLens2Bar3V</p> <p>Vertical HoloLens 2 button bar with shared backplate</p>
<p>Radial</p>	<p>Checkbox</p>	<p>ToggleSwitch</p>
<p>Radial button</p>	<p>Checkbox</p>	<p>Toggle switch</p>

Figure 6 - Button prefabs

Prefab buttons contain an interactable component. The interactable component is a part of the input system and allows an object to be interactable and responsive to different types of input. Objects can be interacted with and you can also add different event receivers.

These event receivers allow different actions to be carried out when they trigger. [11]

3.4 Slack

Slack is used to communicate with our supervisors at TechnipFMC and is used to send and receive files. This is where we can get quick feedback when we pitch ideas, or when we're stuck on a problem that we need our supervisors to look at.



Figure 7 - Slack logo [12]

3.5 Scrum

Scrum is a project framework that we use for our project. Scrum is an agile framework that encourages us to learn from experiences while working on a project. To read more about how we've used scrum read section 4.1 under methodology.

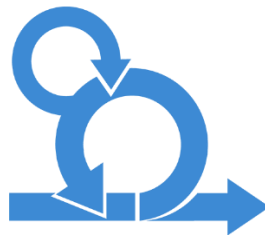


Figure 8 - Scrum Icon [13]

3.6 Teams

Microsoft teams is our file organizer and is an integrated part of the office ecosystem that Microsoft offers. We've chosen to use word as our main text editor for writing this thesis and teams helps us synchronize and organize our files. Teams also allows us to collaborate when using other Microsoft products and this helps when we're all writing on a big document.



Figure 9 - Microsoft teams logo [14]

3.7 Messenger

We use Facebook messenger for communicating internally in the group, as we use this messaging app for everyday use.



Figure 10 - Facebook messenger logo [15]

3.8 Azure DevOps

Azure DevOps offers a wide variety of scrum tools for our project. This is where we make our product back log items, have our digital task board and where we plan our sprints. Azure DevOps also gives us access to make repositories that can be shared easily with the use of git.



Figure 11 - Azure DevOps logo [16]

3.9 Git

“Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.” [17]



Figure 12 - git logo [18]

With git we have used a git workflow as inspiration, see Figure 13

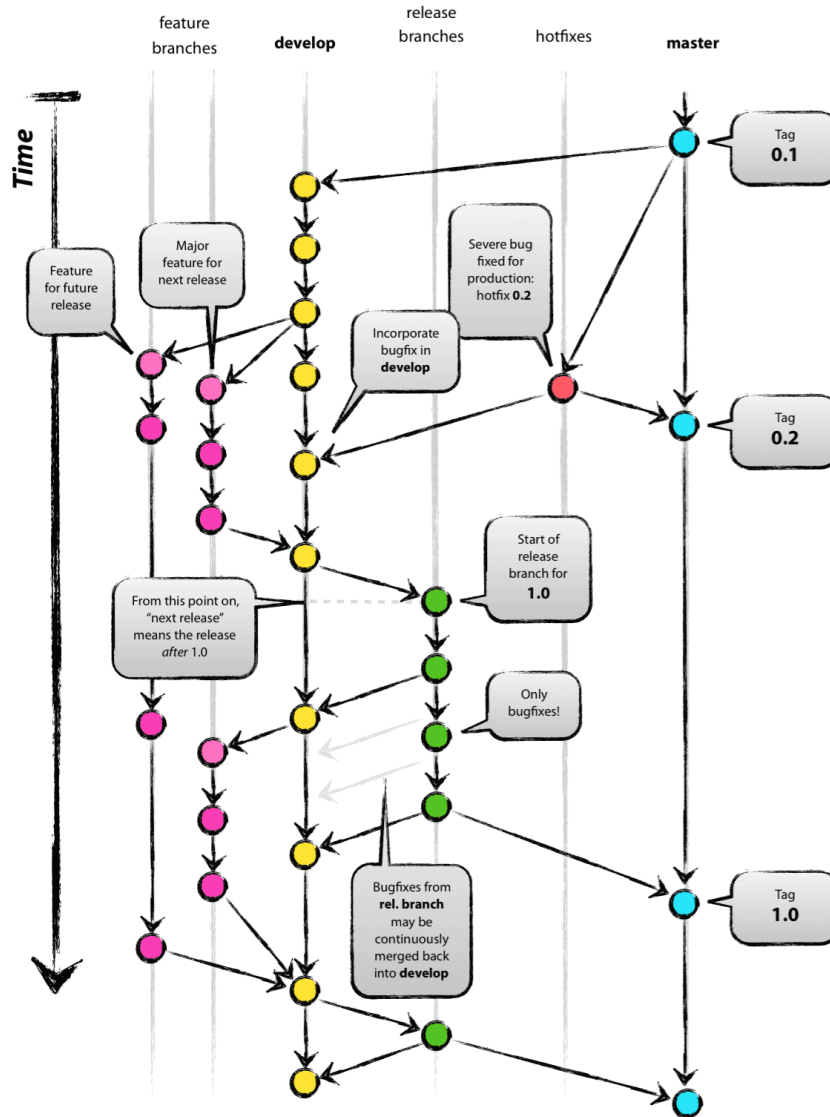


Figure 13 - Git workflow [18]

3.10 Adobe XD

Adobe XD is a mock-up tool we use for our UX/UI design. During the project duration we've used Adobe XD for making wireframes, 2D prototypes and screen design. Adobe XD was useful for mapping all the functionality



Figure 14 - Adobe XD logo [19]

3.11 LucidChart

Used in the context of creating diagrams. This includes among other things flow charts, use case diagrams, class diagrams and sequence diagrams.



Figure 15 - Lucidchart logo [20]

3.12 Visual Studio and C#

C# is desired programming language from TechnipFMC and Unity standard programming language while Visual Studio is desired development tool for C# integration.



Figure 16 - Development tool and language logo [21] [22]

3.13 Microsoft Office Packet

Word is the program used to document this thesis and is our main collaboration tool when documents need to be written.

Power Point is a tool that is helping us make good and professional presentations. During our thesis period we will have a total of three presentations made in powerpoint.

Excel is used to tracking work hours, and to making a gantt diagrams that shows milestones and deadlines.



Figure 17 - Office packet [23] [24] [25] [26]

3.14 Blender

Blender is a free 3D graphic software that is used in the project for 3D modelling, making textures, UV unwrapping. The reason for making models inside of Blender instead of in unity itself is that team members are more familiar with the workflow of Blender. Blender will also be used to prototype animations and experiment with shaders using nodes.



Figure 18 - Blender logo [52]

3.15 Adobe Photoshop

Adobe Photoshop is a paid 2D graphic software that lets us edit photos, make 2d shapes for logos and icons inside of the application.



Figure 19 -Adobe Photoshop logo [53]

4 Methodology

Describes why and how we used the tools and processes we used.

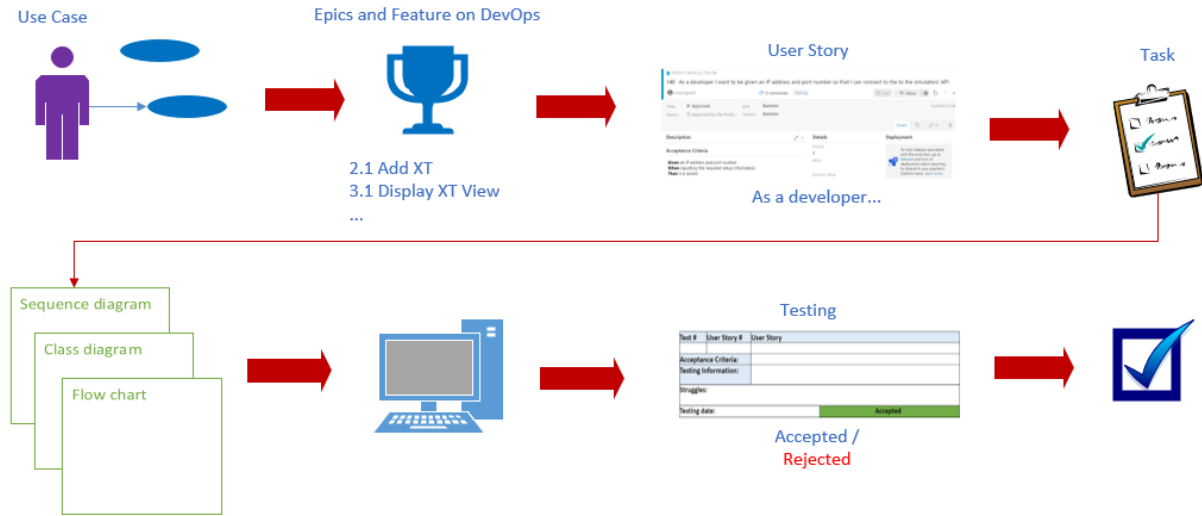


Figure 20 – Methodology approach

When choosing a methodology approach, it's important that it fits the project and provides a guidance path that helps the project's progression and form. Figure 20 is a representation on how Scrum has been a tool for our approach.

4.1 Agile Development: SCRUM

The process model chosen for our bachelor project is based on the agile framework and is called scrum. To meet this year's requirement for writing a bachelor thesis we had to implement an agile process into our project. The benefits of using an agile process model is that we will continuously deliver functioning prototypes of software, this will help us both test and validate that we have delivered useful software after every sprint. After every sprint we will have a sprint review or sprint demo with our internal- and external supervisors where we showcase what we have done in the sprint. With this demo/review we will get critical feedback from our supervisors telling us if we're on the right path or if what we have delivered is not what our supervisors had envisioned. The process model will help us fail early, and this is a good thing. With this agile framework no requirements are final, and they can and will likely change during the project.

Using the scrum process model means that the scrum team is given a lot of freedom. The key difference between this process model and a normal waterfall model is that the scrum team does not have a leader that decides what everyone should do. The scrum process encourages the team to learn from experience, and a commonly used term when talking about scrum is self-organizing, this means that the team members are self-motivated and seeks to take a greater responsibility for the project. By being self-organized the team members takes a shared ownership of the project.

Scrum has four formal events that happens during a sprint. These events are for testing and making adaptations quickly to the project if needed. The four events will be described further in a later section of this chapter. [27]

- Sprint planning
- Daily scrum
- Sprint review
- Sprint retrospective

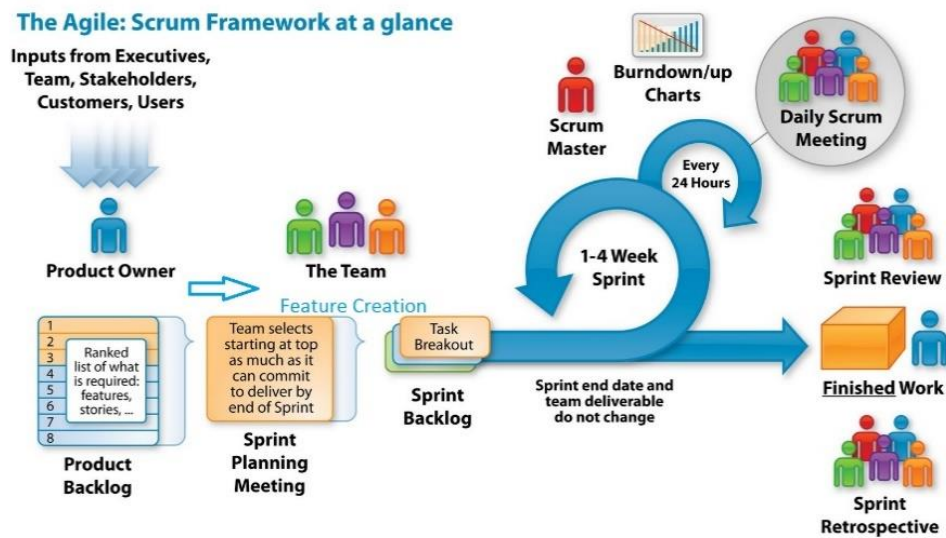


Figure 21 - Scrum process model [28]

4.2 Daily Scrum

The daily scrum is a 15-minute scrum event held every day for the development team where the team plans what they are going to do for that day. The event is meant to enhance team cooperation by looking at what they have done since the last daily scrum meeting. The daily scrum is set up to give every team member an insight into what the others are working on. The daily scrum is a set event that is held at the same place every day. This is to make the meetings orderly and not complex. You are not allowed to sit during the daily scrum. When sitting, meetings tend to go on over intended time, because the participants get comfortable. With scrum this is intended to be an efficient meeting with a maximum length of 15-minutes.

The Daily scrum has three topics that we discuss.

- What did I do yesterday?
- What am I doing today?
- Are there any obstacles in my way that require more attention?

4.3 User Stories

User stories is a scrum term used for what we normally would call requirements. User stories should be easily understood by every team member, and a user story describes a feature that a specific user wants. User stories usually follow a specific template where they describe who is the user of the feature, then they talk about what value the new feature gives and why this value is important for the user. User stories follow some criterion for example: a user story must give some customer value. They need to be estimable and by having a user story that is estimable you've made sure that the story has been broken down into an understandable story and makes it ready to start working on it. They also need to be testable and if a user story isn't testable then we might need to rethink or rephrase the user story.

When user stories are made, they get divided into tasks that each team member will be working on during the sprint. These tasks should be relatively small and get put on the scrum board. For our project we've chosen to both use a physical board with post it notes and a digital one. We also have a term that is called an epic and an epic and a user story are very similar but not the same. An epic is a big complex story that has several user stories connected to them. You can think of an epic as an orange and the user stories are the wedges that make the whole orange.

When user stories have been validated and tested, they can be set as done/completed. If a user story is not completed during its original sprint it should be marked as not completed and then get carried over to the next sprint/sprint planning and during the sprint retrospective the issues surrounding the not completed user story and the bad sprint planning involved.

4.4 Product Backlog

The Product Backlog is like a huge to-do list for the scrum project model. It is here the product owner puts the user stories.

4.5 Sprint

Having a complex project and breaking this into more manageable sizes that is worked on in iterations is called sprints. A sprint is a time period during the project development where the scrum team decided an amount of work that should be done during a set time period. For our project we will work in sprints of 10-days (two working weeks).

4.5.1 Sprint Planning

The sprint planning is where product owner, scrum master and the scrum team gets together to plan what is going to happen for the next sprint. When sprint planning the product backlog should be up to date and be there to provide clarity for the team and the product owner. It's up to the product owner to rank the importance of the items in the product backlog and up to the team to play estimation poker to see what each team member thinks of the magnitude of the user story.

This estimation poker is helpful for planning. If you have two people that picks very different numbers, let's say one picks low and one picks high then they get the opportunity to discuss their different views of the task, and by doing this it will help the team get a broader understanding of the issue and a common understanding.

- 1 What type of item this is (in this case this is a PBI with id 164)
- 2 Who the item is assigned to.
- 3 Shows if the item has been approved or revoked from the project entirely.
- 4 What sprint the item is a part of (at the moment of this picture this is only an item in the PBI and not assigned to any sprint).
- 5 Simple description of the item.
- 6 Acceptance Criteria.
- 7 Discussion about the item, here we can @team members, you can also see a description of this in the photo.
- 8 This is where we put in our estimation on how long the item will take.
- 9 The items priority.

4.5.2 Sprint Backlog

The sprint backlog is a set of items selected from the product backlog that is planned for the sprint. The sprint backlog shows stakeholders and team members what the group are planning to do during the sprint. It's normal to display the sprint backlog on a task board whether this is a physical that's in the room visible for everyone, or if it's digital and online on Azure DevOps as we have it. The task board is to give a representation of what the development team identifies as necessary work that must be done for the sprint goal to be met. The daily stand-up meeting is used to give the development team an understanding on how progress is going.

4.5.3 Sprint review

During a sprint review product owner calls in internal- and external supervisors for a demo of what the team has done during the latest sprint. This is where the supervisors get a live demonstration of the product that has been worked on and this is where the product owner show of what the team has done. It's also important to discuss the things that wasn't finished so that the team shows that they are transparent with their supervisors. During the sprint review supervisors will come with constructive feedback on what they like and what they don't like. This is also where you get feedback if the team is heading in the right direction with the project or if they need to rethink their solutions. During a sprint review it's important that the developing team is all present if questions rise about their work. So, the presenter of the demo should have a good overview of what the team has done during the sprint, but they don't need to know all the specifics.

It's important for team members that don't present to take notes on feedback given from internal- and external supervisors and post this on a "*heard*" board for the group to see. The "*heard*" board is feedback that was heard during the sprint review from supervisors, and it's important that these notes are considered for the next sprint planning process and rises in priority.

4.5.4 Sprint Retrospective

In the sprint retrospective the Scrum team is supposed to pause, reflect and analyse the way they work. This phase comes after the sprint review and the purpose is to identify areas of improvement to the process of creating the product. Anything that affect how the team creates the product is open for discussion and review. This is where the team finds ways to work better, faster and more efficient. After areas of improvement are identified, the team needs to plan how to implement improvements. The sprint retrospective is a way to ensure continuous improvements to the Scrum framework by inspecting and adapting. [29]

4.5.5 Burndown Chart

Burndown chart is a standard chart in a scrum process that shows work left vs time left. The burndown chart is helpful for predicting when all the tasks for a sprint will be finished.

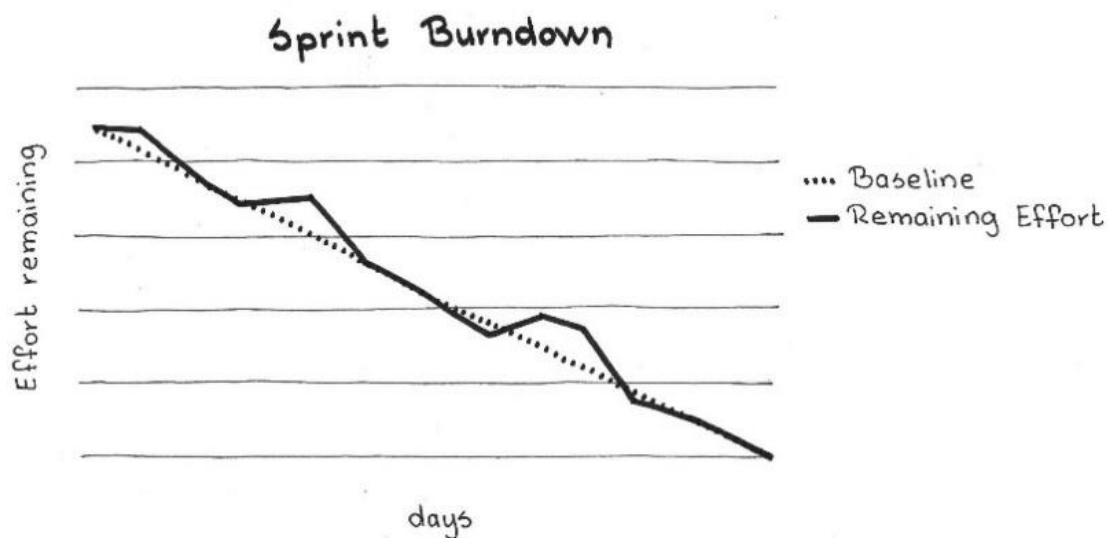


Figure 22 - Example of a burndown chart used in scrum

The vertical axis shows how much work is left and the horizontal axis shows how many days have passed.

4.6 Minimal Viable Product MVP

“A minimum viable product (MVP) is a development technique in which a new product or website is developed with sufficient features to satisfy early adopters. The final, complete set of features is only designed and developed after considering feedback from the product’s initial users.” [30]

This is what we are developing using a dev ops method on working. Dev stands for design and Ops stands for implementation and we use this to make sure that our design is doable and within our reach. By doing a little design and then implementing it, we can early see if there is some major problems that will occur and that we need to take position to already in a first iteration, or let’s say that the implementation was done with success we can then decide if that or another concept is what’s best for our project. By working Dev Ops, we don’t have the arguments about what the best solution is after the design part. By implementing the design that we think is best in an MVP fashion we can take our designs and showcase it for the group and the necessary stakeholder that will give us good and conclusive feedback on what design they want. Also, by working in a Dev Ops manner we can easily try out our design before committing to any design at all. Working this way also gives us an advantage in finding errors and doing R&D.

4.7 How Scrum Has Been Used

We have been using scrum very careful by following every feature that scrum has to offer. The daily stand-up (will be called daily scrum from here on out) has been a great success and important for our team communication. In the beginning of the project we had some problems keeping the daily scrum to 15 minutes and we were also speaking when it wasn’t our turn to speak. Now after a few sprints it’s become very autonomous having the daily scrum. People are listening to each other and asking question that are appropriate to the person speaking. We are more and more coordinating what we’re working on and asking group members for feedback and help. Since we are such a small group, we try to show team members what we are working on during the meeting unless this takes too much time, or the group has questions about implementations etc. then we take it after the daily scum.

When it comes to the sprint planning then we have a whole other story. We started of making way to big user stories for our project, we were so focused on details that we forgot the big pictures sometime and we didn't not only redact or remake the user stories once we did it twice and now after the last remake we come to a great and easily understandable method on deciding what epics, features, user stories and task look like. The fact that we have failed two times now in the beginning has just strengthen our project immensely in its robustness and overall development. To be clear we did not throw away our projects 100% two times, we keep all the good stuff and did mostly rename and remake some user stories that was to begin with one big into smaller, more understandable and implementation friendly user stories.

4.8 Group Dynamic

When we're working the way we do, it's obvious that we have decided to divide us into smaller groups internally. There is no use that six members are working on the concept when we have new technology to experiment with. We have divided us into two people working on R&D on the HoloLens2. Their responsibility is to find a stable version of the software that we need to use and to explore the possibilities available to us.

The other section is the design team which has the responsibility to start making the concept with exceptional exploration on the topic UX/UI. This team is further divided into two people working only on the User Experience or UX for short. It's the UX team to find a concept that the customer wants and give the customer the functionalities they need for the product to be a success. Our task is very user experience heavy because we are dealing with a HoloLens2 which can be compared to just a big screen, in a matter of fact I believe at this moment it's the world's biggest screen available at the time. The other part of the UX/UI team is doing the UI work. They are exploring the different colour possibilities and finding out what is nice and comfortable for the HoloLens2 user to look at. They are making sure to make a colour palette that will be used when development of the final HMI (human-machine interface) that the user will be looking at.

Our last team member is working on a configuration file for mapping the subsea Christmas tree (XT).

4.9 Meetings with Internal Supervisor

Meetings with internal supervisors Henning Gundersen and Steven Bos are held every Monday at 14:00.

4.10 Risk Management

Risk management is used to prepare for the unexpected and is important process that needs to be included in project management in order to accomplish success. Risks are possible events that may affect the project positively or negatively. If it is positive, it's called an opportunity and if it's negative it's called a threat. Most risk are negative, and prior knowledge of the identified risks helps the project team develop and implement a strategy that among other things can mitigate the likelihood of occurrence and/or the consequence of the risks. In this project, for each identified risk, the risk management process model is followed.

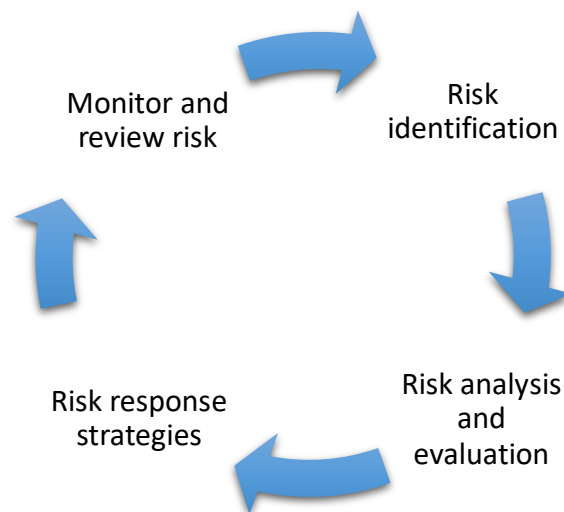


Figure 23 - Risk management process model

4.10.1 Identifying Risks

Risk identification is an iterative process where the objective is to identify and record all possible risks that may affect the outcome of the project negatively. Once a risk is identified, evaluation and analysis need to be taken into consideration which is essential for developing a strategy to reduce their impact. [31]

Identified risks will be mapped into the following categories:

- Technical
- Non - Technical

4.10.1.1 Methods to identify risks

Several methods can be used to map and identify the different risks that may have an impact on the project. Brainstorming, looking at history and identifying functions/factors that is critical to success is some approaches that we are going to use throughout this project period. Other known methods are checklist, root cause analysis, flow chart and interview.

Brainstorming

Brainstorming is a method where the members of a group implicitly thinks out loud without any judgment. Throughout a brainstorming session a list of risks can be generated.

History

By looking at history, risks can be found and identified easily. Many projects have several risks in common.

Critical functions/factors for the system to operate successful

Identifying all main functionalities and factors that influence the system, by finding them we can work backwards to see what happens if they're not performing and by doing it this way we can identify important risk. [32]

4.10.2 Risk Analysis and Evaluation

All risks identified must be analysed and evaluated in order to get a better understanding of their possible impact on the project. By doing this, we can prioritize those who need immediate attention and those who don't. Risk analysis identifies the possible causes of a risk occurring and its potential impact on the project. Risk evaluation helps with defining the severity of the consequence. This is determined qualitatively through a risk matrix. The severity is calculated by giving the likelihood and the consequence a rank and then multiply them together. **Likelihood x Consequence = severity.**

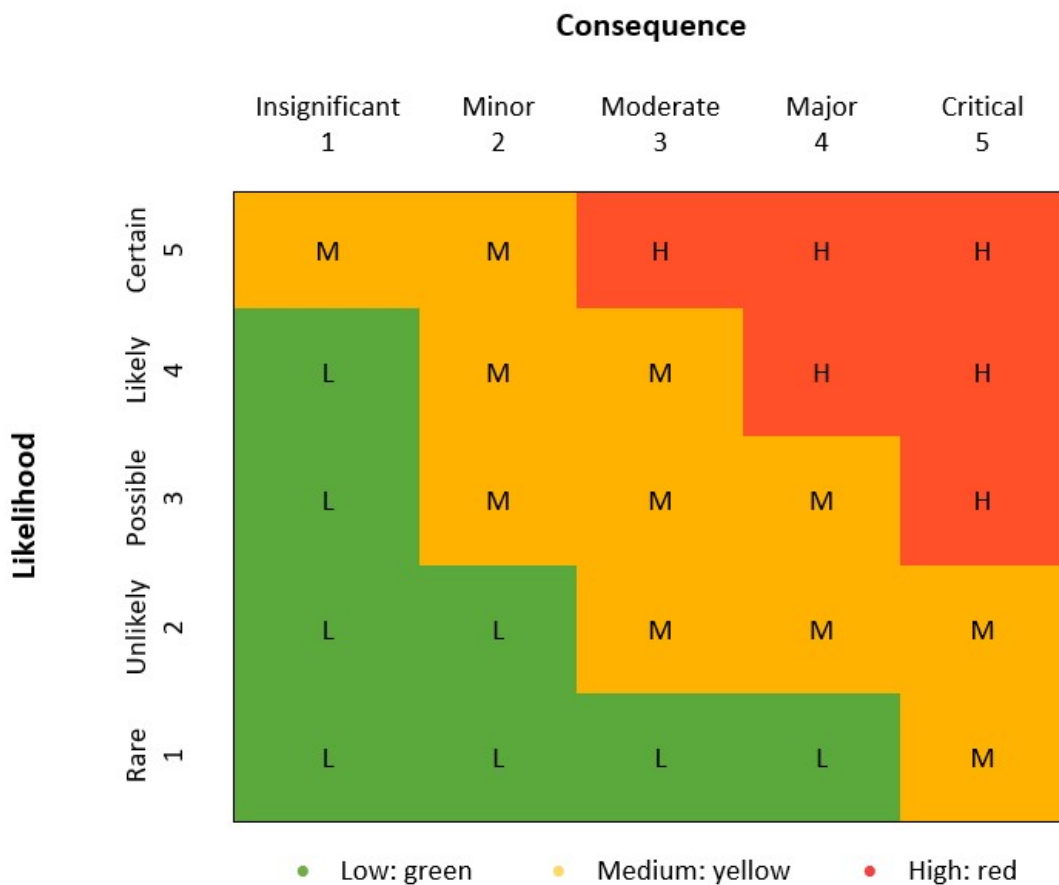


Figure 24 - Qualitative determination of level of risk matrix

Description of the priority

Priority	Description
High	Immediate action is required
Medium	Action required
Low	Action not required

4.10.3 Risk Response Strategies

Based on the findings from the analysis and evaluation a risk response strategy is implemented. There are different ways to respond to negative risks and the method used depends on the risk. However, mitigation is probably the most common one to use out of the four listed below.

Avoid

Avoidance is an approach where the activities that causes the risk is stepped away from. Meaning that you stop a risk before it happens by changing for example scope, functionalities or schedule. This strategy is often used when the threat has a great impact on the project, and the best option is to eliminate the root cause of the risk completely.

Transfer

Transferring of a risk to a third party is done when it's not manageable. Then, if the risk occurs someone else is responsible to manage it. Some reasons for choosing this strategy can be lack of experience, resources or skills.

Mitigate

Risk mitigation reduce the likelihood of occurrence and/or the impact it has on the project. This is the most frequently strategy used. Most risks are unavoidable but there are always measures that can be done to prevent the risk from happening or to reduce the consequence of it.

Accept

When a risk is accepted, it means that the risk is worth taking and the only thing done is to acknowledge it. This strategy is often used for non-critical risk or if the effort involved does not outweigh the benefit. Acceptance can either be passive or active. Passive includes only acknowledge it and adding it to the risk register while active includes having another plan if it occurs. [33]

4.10.4 Monitor and review risk.

From the moment the risk response plan is implemented, it is important to keep track of the effectiveness of it. When it is discovered that changes need to be made, a new risk response plan is to be developed. This can be done at the end of the sprint in the risk meetings, or if required, as soon as possible. Also, to monitor the risks, the risks are prioritized by having one table for all the ranks (high, medium, low) and categorized into technical and non – technical. Risks that is no longer a risk will be registered in a table at the end of each sprint.

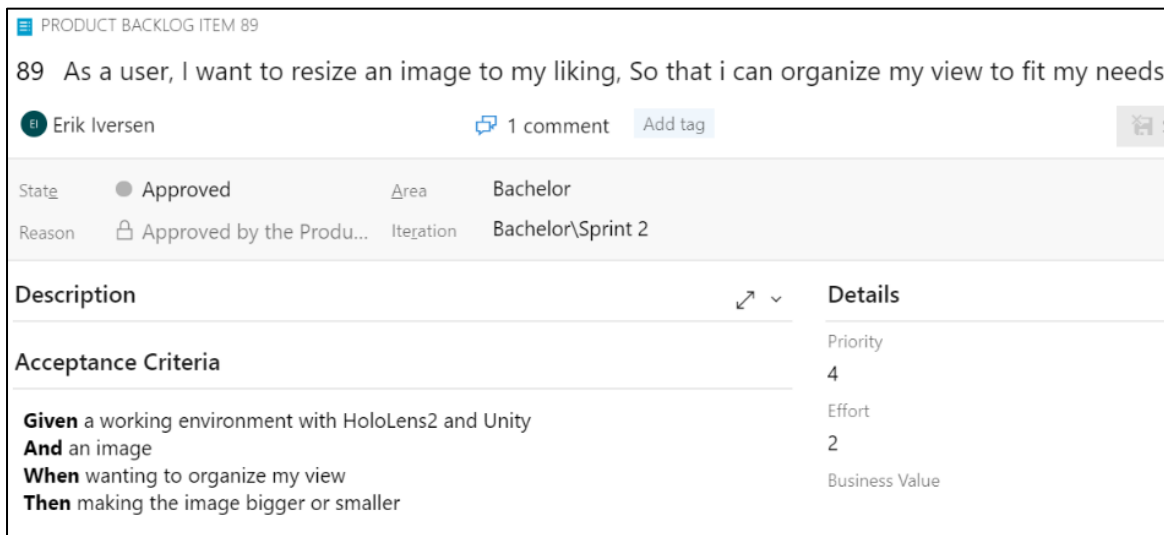
4.11 Testing

Testing is important during the whole lifecycle of the project to confirm that the product we are making complies with the customer's needs and wishes.

4.11.1 Acceptance criteria testing

Testing of user stories is conducted by making sure that the acceptance criteria is fulfilled, and the testing template is accepted.

Attached to every user story there's an Acceptance Criteria which can be viewed below in Figure 25. This defines the parameters that must be fulfilled in order for the user story to be accepted.



The screenshot shows a Jira user story card for item 89. The story text is: "As a user, I want to resize an image to my liking, So that i can organize my view to fit my needs". The card is assigned to Erik Iversen and has 1 comment. The state is "Approved" and the area is "Bachelor". The reason is "Approved by the Produ...". The iteration is "Bachelor\Sprint 2". The card is divided into two main sections: "Description" and "Acceptance Criteria". The "Acceptance Criteria" section lists three conditions: "Given a working environment with HoloLens2 and Unity", "And an image", and "When wanting to organize my view". The "Then" condition is "making the image bigger or smaller". The "Details" section shows the priority is 4, effort is 2, and business value is present.

Figure 25 - A User Story with coherent acceptance criteria from taskboard in DevOps

After a sprint where a user story usually is considered done by a team member, we test it against the acceptance criteria and fill out the test template. If the team member and tester agree that the acceptance criteria are fulfilled, then the test template is marked with the green "Accepted"

Test #	User story #	Feature #	User story
1.0	46		
Acceptance Criteria:		Given data from simulator When operating in a system (InControl/Unity/HoloLens2) Then view the same data	
Testing Information:			
Struggles: <u>Struggle 1:</u> Connection issues. Could not connect HoloLens 2 to computer. Resolved it by connecting using USB C to USB A cable (USB 3.0) first. Then paired HoloLens 2 to the computer and finally remote debugged in Visual Studio. <u>Struggle 2:</u> HoloLens 2 could not communicate with simulator over Wifi. Resolved it by using following project settings in unity under Project Settings - Player - Capabilities: Internet Client Internet Client & Server PrivateNetwork Client & Server			
Testing date: 13.02.2020		Accepted	

Figure 26 - Example user story that has been tested and fulfilled its acceptance criteria

If the acceptance criteria are not fulfilled then it is marked with a red “Rejected” and the user story must be revisited, altered, or improved before or during the next sprint.

4.11.2 Usability test

The usability test is used to map what can be improved in the application. The test is to be completed both by stakeholder and a naïve user to achieve a more versatile response and is done at the end of the project period.

The usability test consists of two parts:

Part A - Tasks: Include tasks which the user is to perform without help or interference in order to evaluate if the application design come as “natural” for the user and to find out whether if it easy to use and helps the user reach the goals. The user is to talk out loud through the whole test session and say what’s ever on their mind and how he/she feels when completing the tasks.

Part B - Survey: Consists of a ten-item questionnaire with five response options to verify (or not) that our system has a better usability than the already existing system, and uses the System Usability Scale (SUS) [34].

4.12 UX Design

A major part of this project is to make it a better User Experience than the currently existing system has. This means a good UX is essential and should be carefully thought through. The user must be in the centre of our decision-making process. A good UX design consists of mainly two building blocks that is important to differentiate: UI and usability. UI represent the visual parts of the design, colour, font, size of text, buttons, screen and other graphical components. While usability refers to how easy or convenient the user finds it to use the product; do you find the information you are looking for? Does the app do what you expect it to do? Does it help you reach your goal? What confuses you? Etc. UX is about the overall experience with the product, from start to finish. The experience can be positive, negative or neutral and is based on how the user feels and what thoughts arise when interacting with the product.

The HoloLens2 glasses give the user the opportunity to interact with a system on a whole new level in the terms of user experience. The glasses put virtual interactable objects out in the real world as it where real objects that the user can touch, move and scale. Audio effects can be added, and the glasses register where the user look and points. All these gestures help improving the user experience and it is important to take advantage of these visual aspects when making a system to make it feel as real as possible. Also, visual effects help the user to get a clear understanding of how the system works without much effort.

4.12.1 AR best practise – colour and light

After testing HL2 we notice some red and purple curved lines which was very noticeable when looking at white backgrounds. It also seemed that HL2 had a hard time rendering small text. This and subjects listed below must be considered when designing the UI.

Colour and light: Since holographic devices have additive displays, whites will appear super brightly and black will appear transparent. By using a dark grey instead of black we can work around the see-through problematics and by using white in moderation we can alleviate discomfort. Large sections of solid colour may become blotchy, so avoid if possible, to achieve colour uniformity. [35] [36]

Contrasts: When using the HL2 in a room with diverse lighting conditions we risk that the contrasts in our environment create an unclear overview ex: your focus will go to the brightest colour and the other objects will seem muted. By creating appropriate levels of contrast the whole view will seem clearer.

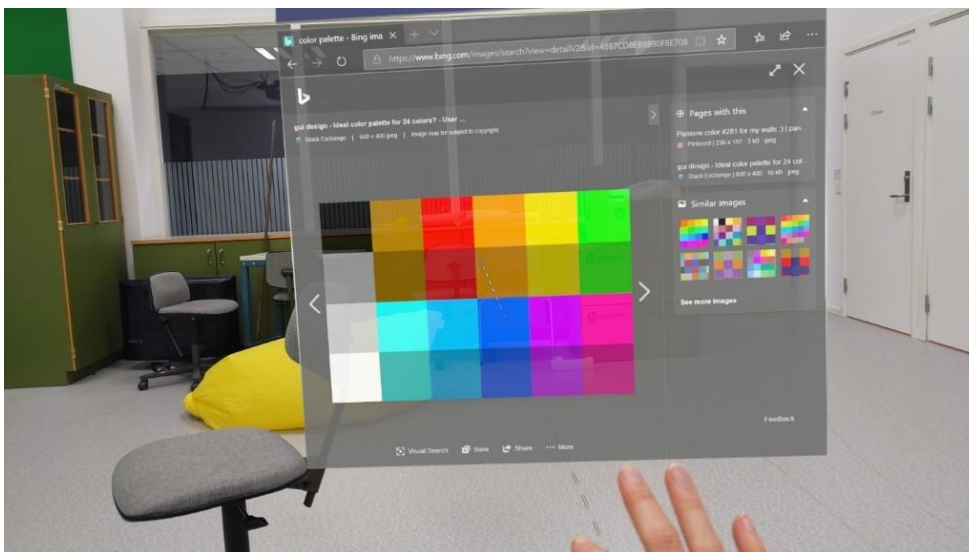


Figure 27 - Colour palette in HL2 view

Dynamic Lighting and Colour Uniformity: Dynamic lighting is best to avoid, but if required Microsoft recommends using the Mixed Reality Toolkit Standard shader.

5 Modelling the Product

The purpose of this chapter is to present a high-level view of the application.

5.1 Requirements

The requirements given by TecnipFMC were broken down into functional and non-functional requirements. A functional requirement is a function that the system must perform, while a non-functional requirement describes how the system works.

Colour coding see Figure 40 is implemented from use cases down to test documentation for traceability and readability.

5.1.1 Non – functional Requirements

- Make an application using AR.
- Use SW given by TechnipFMC (InControl, Simulator).
- Multiple users (multiple HL2).
- There should be similarities between this application and existing TechnipFMC applications.
- The application should be more, or as user friendly as the existing system.

5.1.2 Functional Requirements

- Secure valve operation.
- Connect to different XT's, preconfigure only once.
- Display XT information.
- Display HCR information.
- Read valve and sensor information.
- Display sensor history.

5.2 Hierarchy

The system is divided into five high level sections, categorized as Epics in DevOps.

The five Epics:

1. Choose User
2. Manage XT
3. XT Operation
4. Hand Menu
5. Software Communication

Each Epic, except Software Communication, is made from and represent a Use Case diagram. The Epics has Features associated with them, which represent the use cases (bubbles) inside the use case diagrams. The hierarchy in DevOps is displayed in Figure 28. Each feature has one or more user story attached to it with acceptance criteria. When all user stories related to their features is tested and accepted, the associated requirement is then validated.

👑 1. Choose User	👑 3. XT Operation	👑 4. Hand Menu
🏆 1.1 Choose Admin	🏆 3.1 See XT Schematic	🏆 4.1 Go to Manage XT
🏆 1.2 Choose Guest	🏆 3.2 Control Valves	🏆 4.2 See HCR Schematic
👑 2. Manage XT	🏆 3.3 Read Valve Status	🏆 4.3 Read Me
🏆 2.1 Add XT	🏆 3.4 Read Sensor Information	🏆 4.4 Close Application
🏆 2.2 Edit XT	🏆 3.5 Display Sensor History	🏆 4.5 Clear View
🏆 2.3 Connect to XT	🏆 3.6 Display Sensor Trend	👑 5. Software Communication

Figure 28 - Snapshot from DevOps of Epics and associated Features

Software Communication between the simulator and the applications is a huge part of our system. Therefore, this is categorized as an Epic but does not represent a Use Case Diagram nor does it have associated features as these represent use cases inside the use case diagrams.

5.3 Use Case Diagrams

In this chapter we will explain a high-level view of the functionality in our application.

5.3.1 Choose User

Anyone with the password can login as an admin user. Everyone else must choose Guest login.

After choosing user you will be sent to the view - 2 Manage XT.

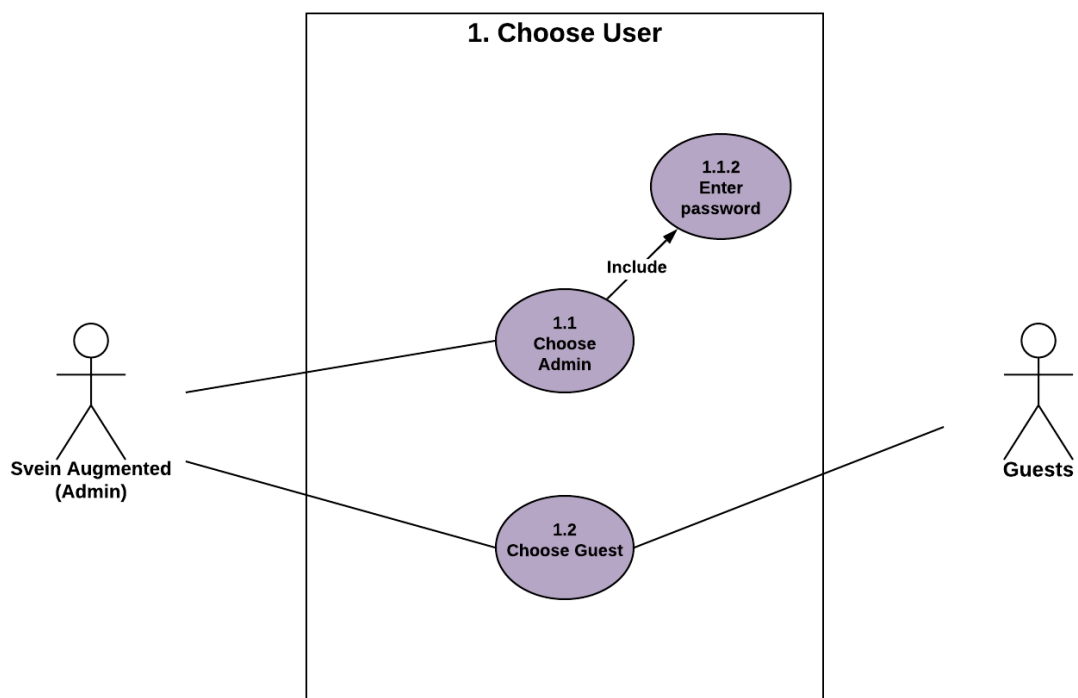


Figure 29 - Use Case: Choose User

5.3.2 Manage XT

Regardless of what user you are logged in as, you can choose between adding a new XT, edit an existing XT or connect to an existing XT. Editing an existing XT will also give you the option to delete. Connect XT sends you on to the next view - 3 XT Operation

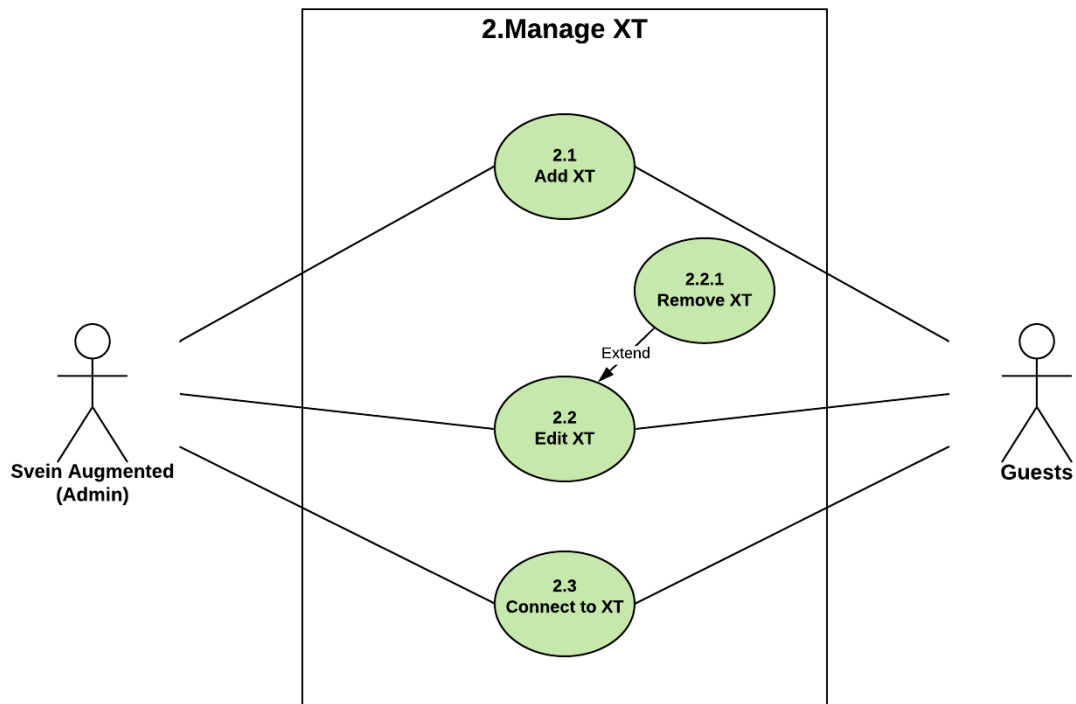


Figure 30 - Use Case: Manage XT

5.3.3 XT Operation

In this view the user can see the different schematics populated with components i.e. valves and sensors. When pressing a component, the user can get more information and operate the component depending on what type.

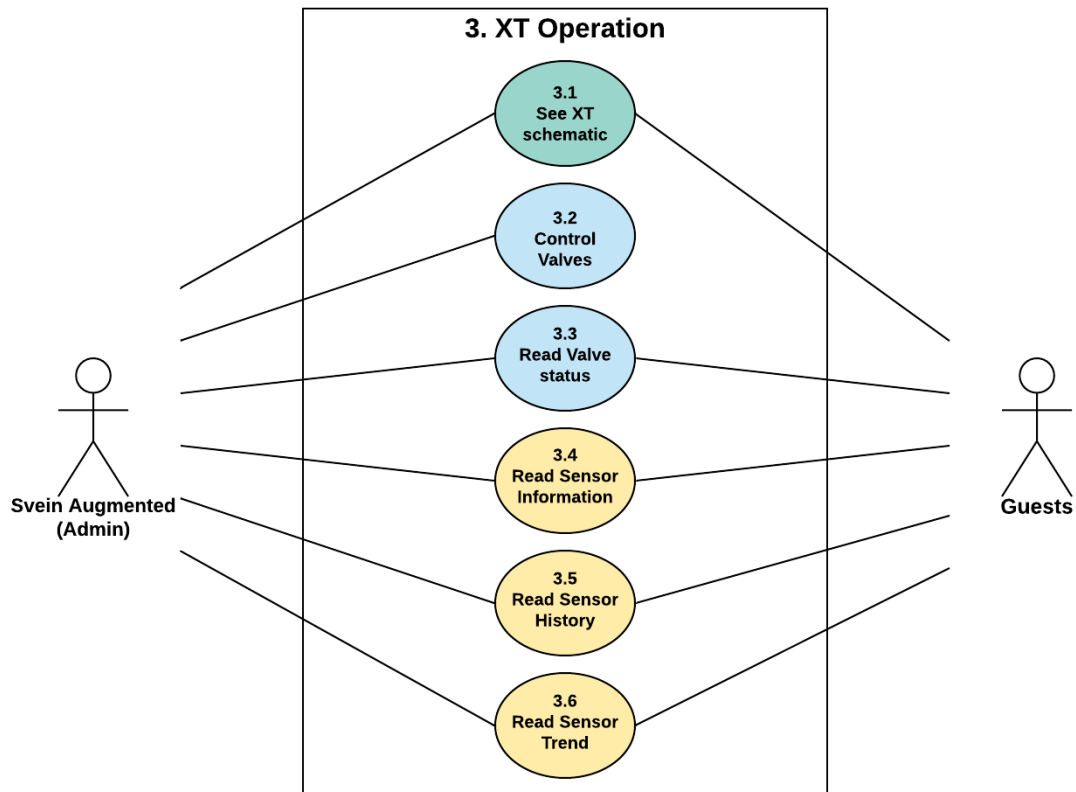


Figure 31 - Use Case: XT Operation

5.3.4 Hand Menu

The hand menu is a navigation tool used in the application. This can be accessed by turning the palm of your hand towards you. The functions in the hand menu differentiates depending on what view you are in.

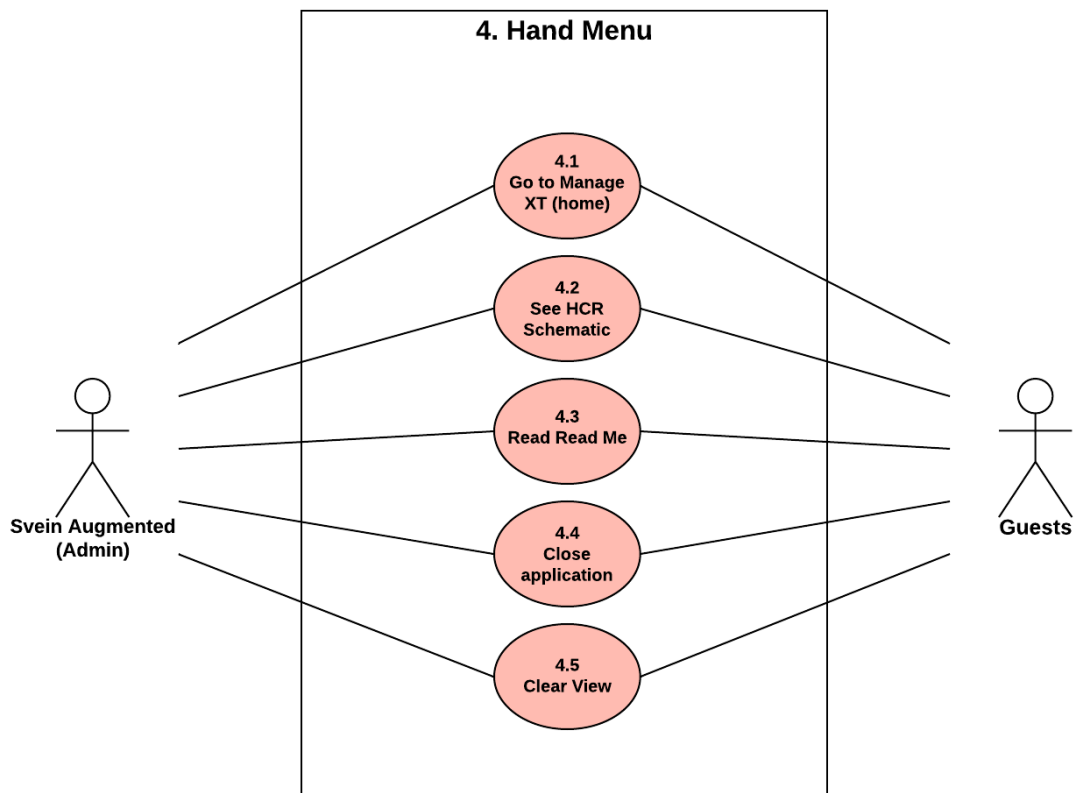


Figure 32- Use Case: Hand Menu

5.4 User stories

This is a sample of some of the user stories created throughout the project. All user stories have Acceptance Criteria which is tested to make sure the requirements are satisfied.

5.4.1 Choose admin

NR: 139

As a user I want to select admin mode So that I have the authority to open/close valves

ACCEPTANCE CRITERIA:

Given selection between admin and guest mode

AND having a PIN

When correct PIN is typed in

Then having the authority to open/close valves

5.4.2 Connect to XT

NR: 171

As a user I want to connect to an already added XT by selecting it from a list with my finger,
So that I can operate the XT

ACCEPTANCE CRITERIA:

Given a list of XT's.

When choosing a XT by point clicking on it

Then I connect to the XT and open the XT view

5.4.3 Display XT Schematic

NR: 169

As a user I want to see the valves and sensors displayed on a pipe that is configurable for different XT's so that I can intuitively see how the XT is performing

ACCEPTANCE CRITERIA:

Given a config file with component types, names, placement, and addresses

When opening the XT view

Then a XT is drawn in the right order with corresponding values on components

5.4.4 Open/Close Valves

NR: 142

As a user I want to open/close valves in HL2. So that I can regulate the flow

ACCEPTANCE CRITERIA:

Given the valve I want in HL2 is pushed

And a face plate pops up

When pushing open/close button in the face plate

Then valve status changes in the simulator

5.4.5 Sensor Data

NR: 156

As a user I want to visually read sensor data, so that it is possible to display it as single entities

ACCEPTANCE CRITERIA:

Given Sensor data

And XT view

When XT view is open

Then Sensor data is displayed in XT view

5.4.6 Hand Menu

NR: 238

As a user I want to be able to use a hand menu to navigate between the different views in AWC application

ACCEPTANCE CRITERIA:

Given selection between different scenes as displayed in photo

And a working hand menu

When user raises left hand

And press one of 5 different buttons

Then scenes/view should change seamless

5.4.7 Display HCR Schematic

NR: 169

As a user I want to see the valves and sensors from the HCR displayed on a pipe that is configurable for different HCR's and XT's so that I can intuitively see how the HCR is performing

ACCEPTANCE CRITERIA:

Given a config file with component types, names, placement and addresses

When opening the HCR view through the hand menu

Then a HCR diagram is drawn in the right order with corresponding values on components

5.5 Sequence diagram

In this chapter we will show a high-level sequence of some of the different features.

5.5.1 Choose Admin – Feature 1.1

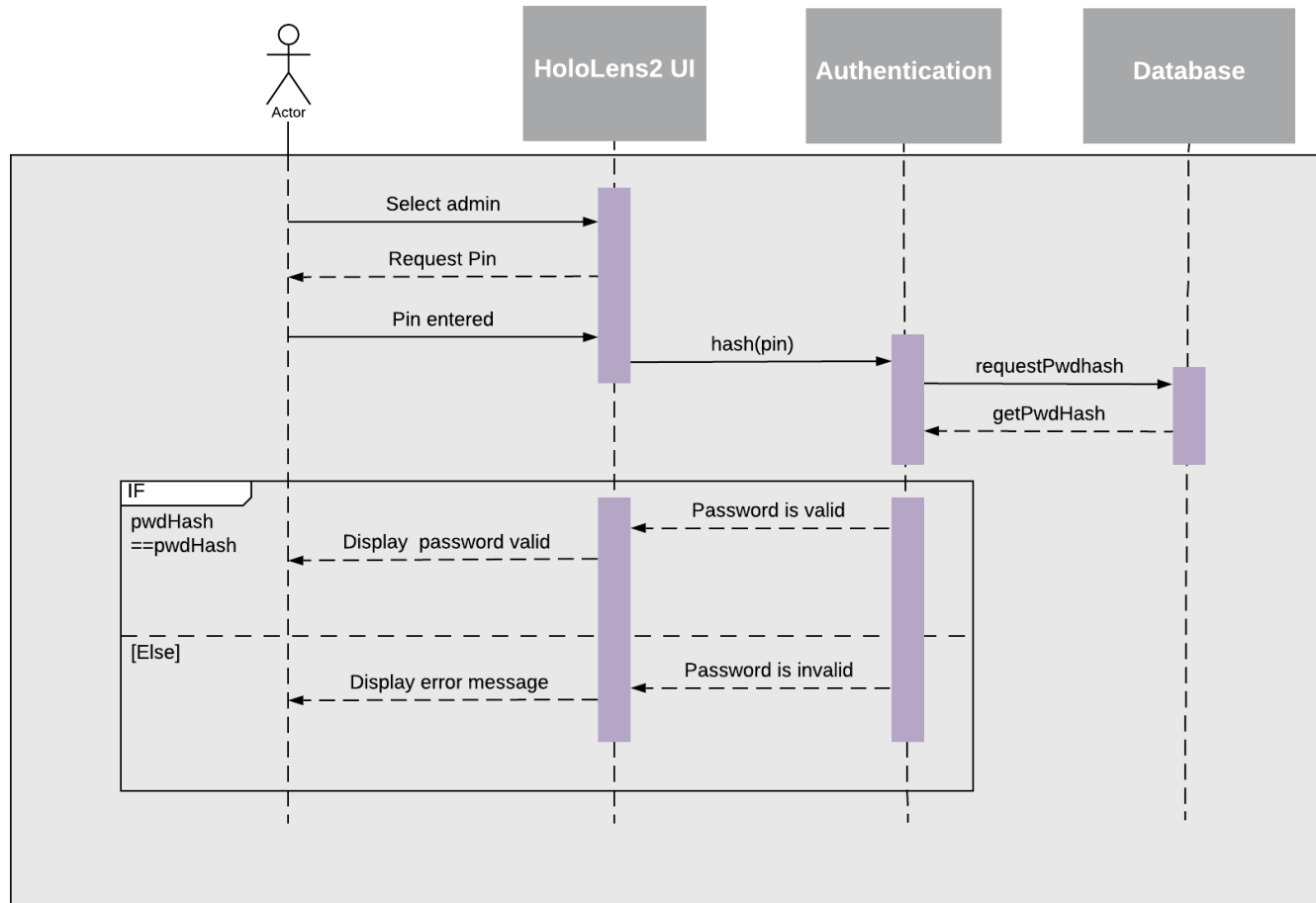


Figure 33 - Sequence Diagram: Choose User

5.5.2 Manage XT – Feature 2.1

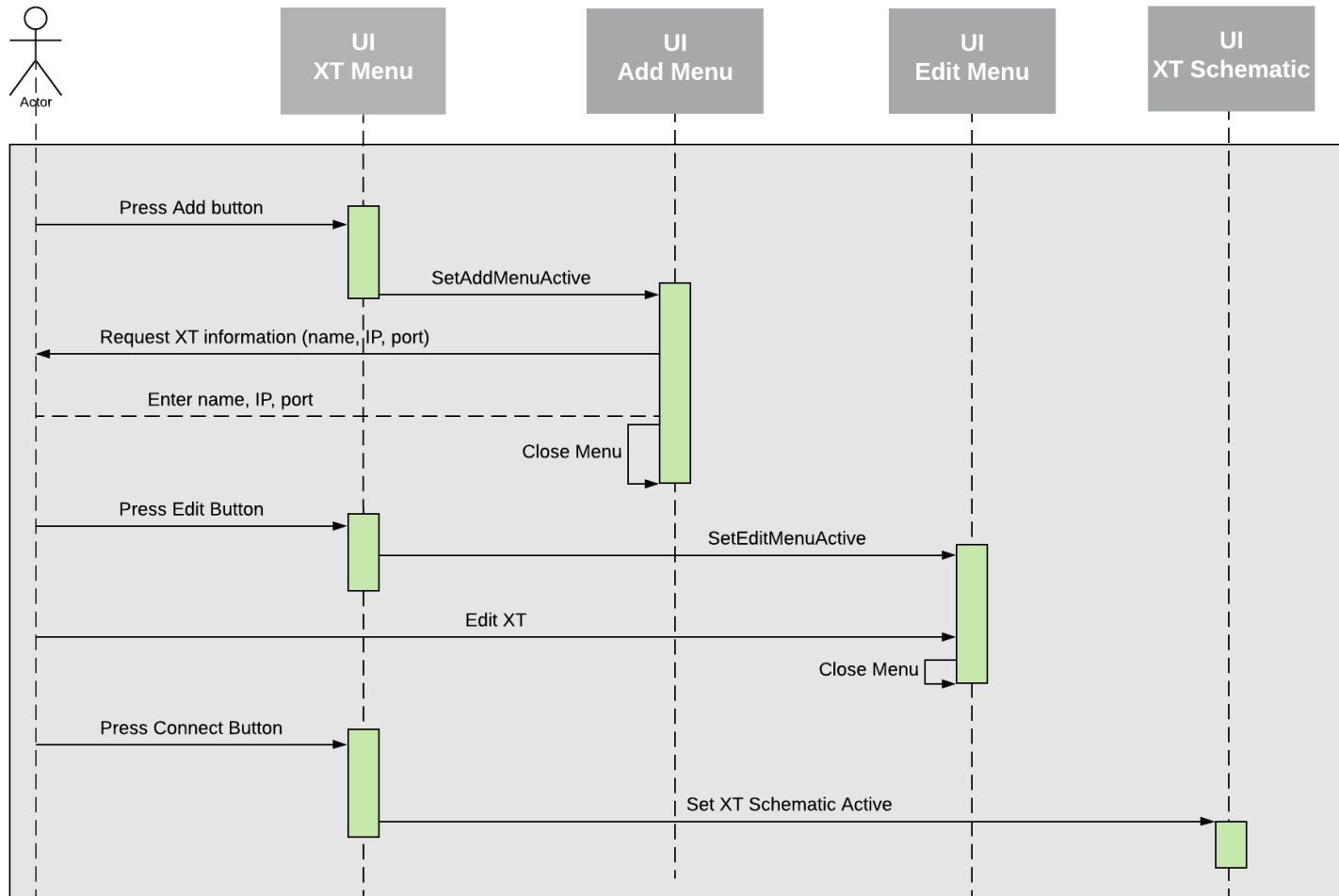


Figure 34 - Sequence Diagram: Manage XT

5.5.3 See XT Schematic - Feature 3.1

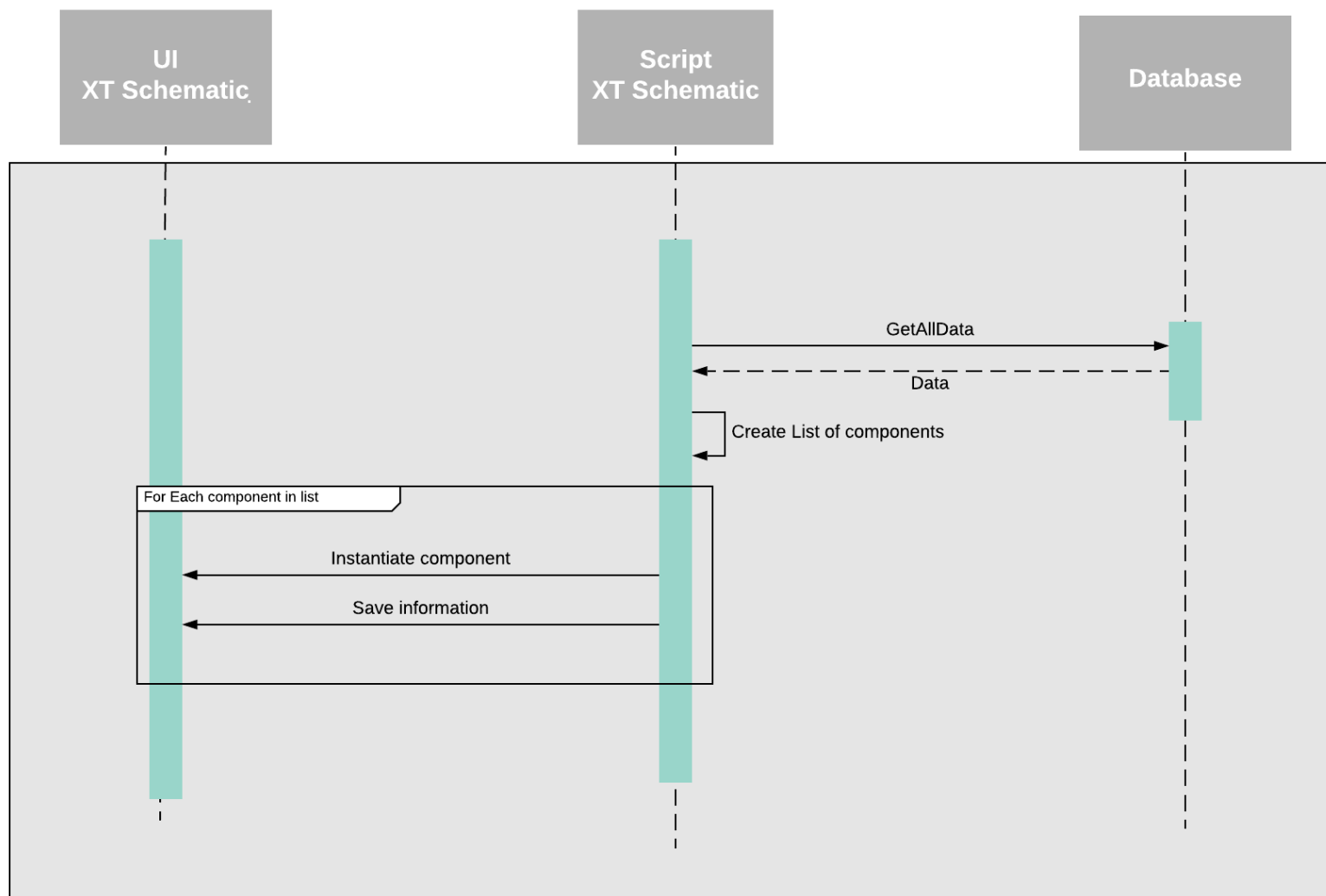


Figure 35 - Sequence Diagram: XT Schematic

5.5.4 Valve Operation – Feature 3.2

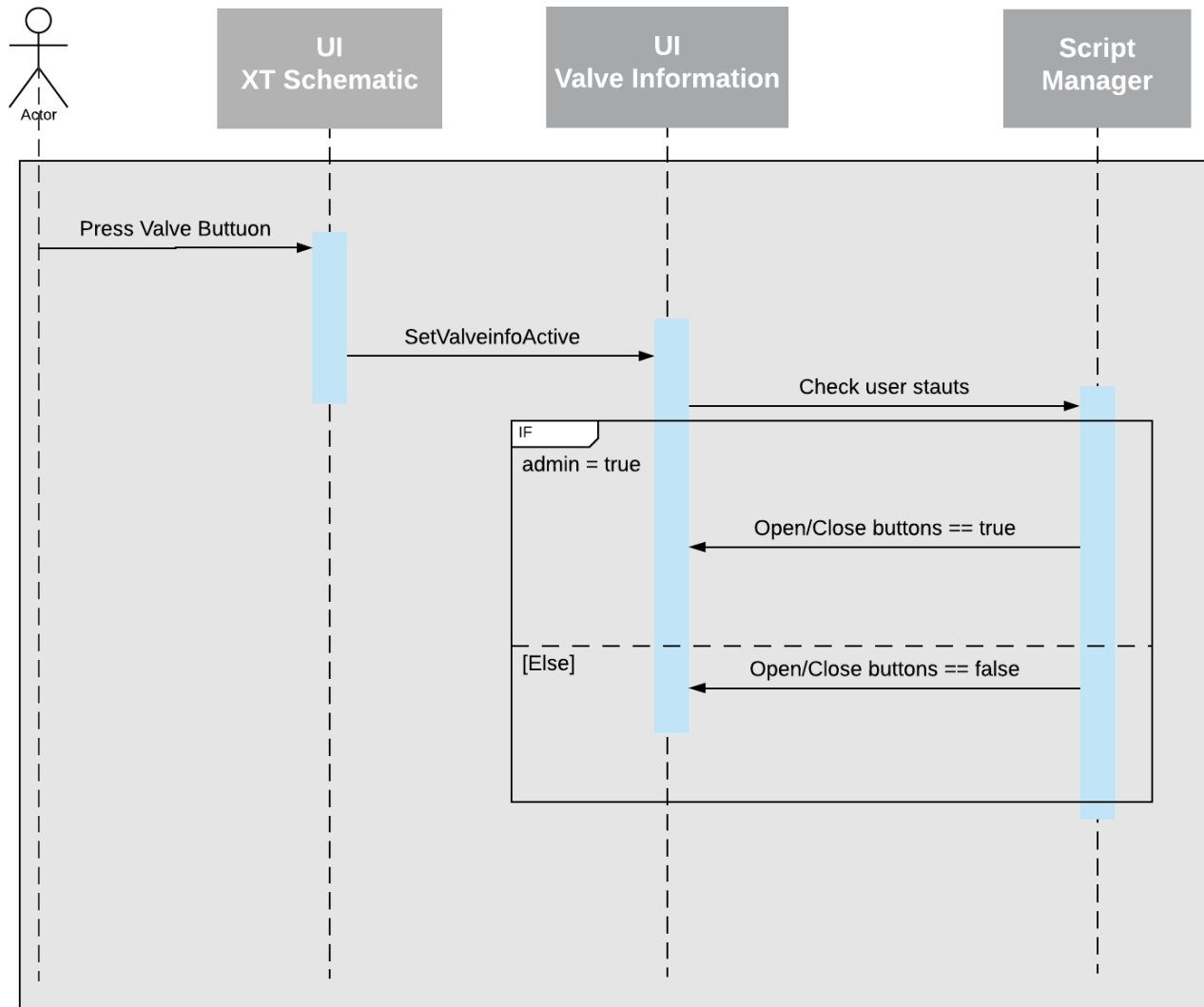


Figure 36 - Sequence Diagram: Valve Operation

5.5.5 Sensor Information – Feature 3.4, 3.5 and 3.6

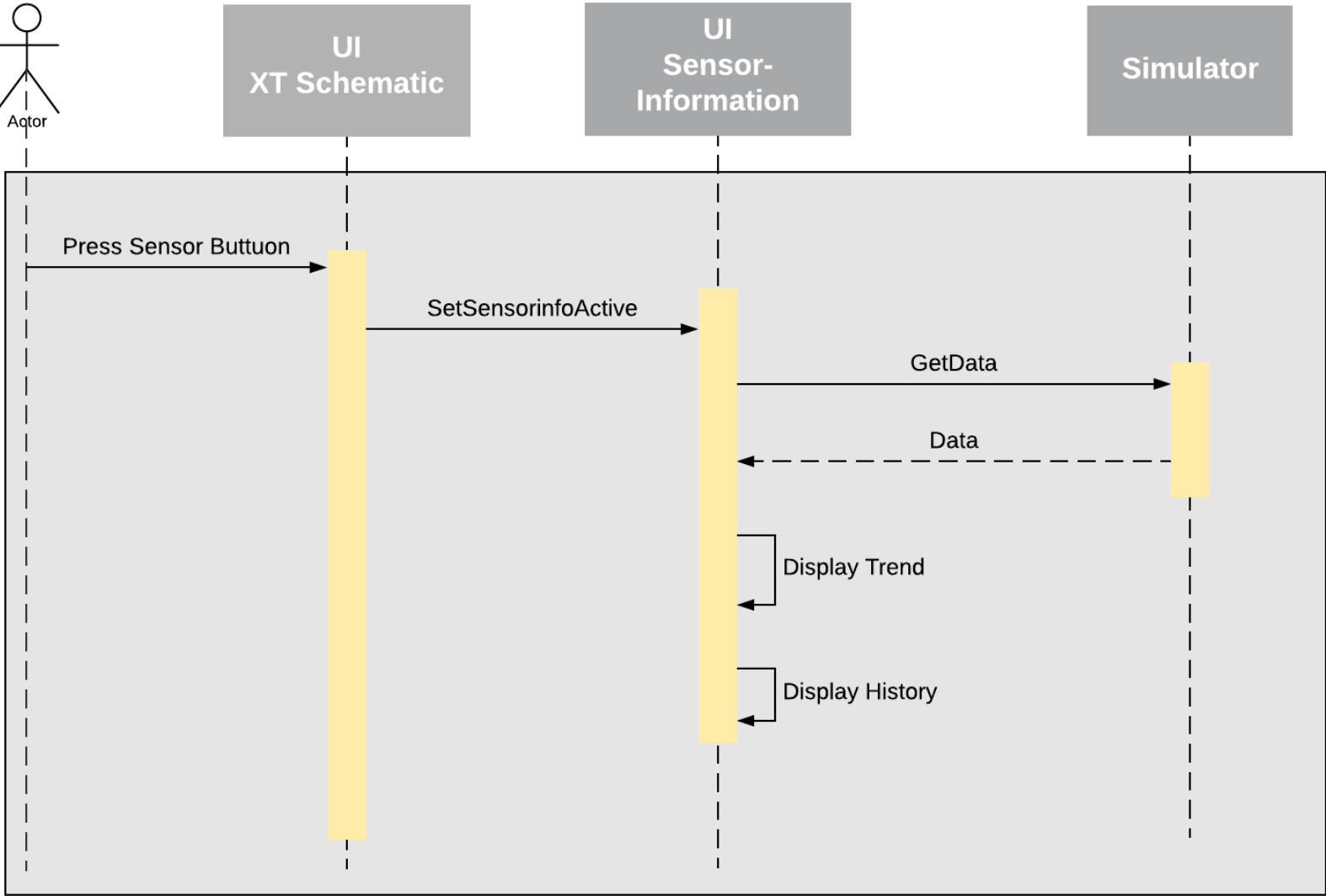


Figure 37 - Sequence Diagram: Sensor Information

5.5.6 Hand menu - Feature 4.1, 4.3 and 4.4

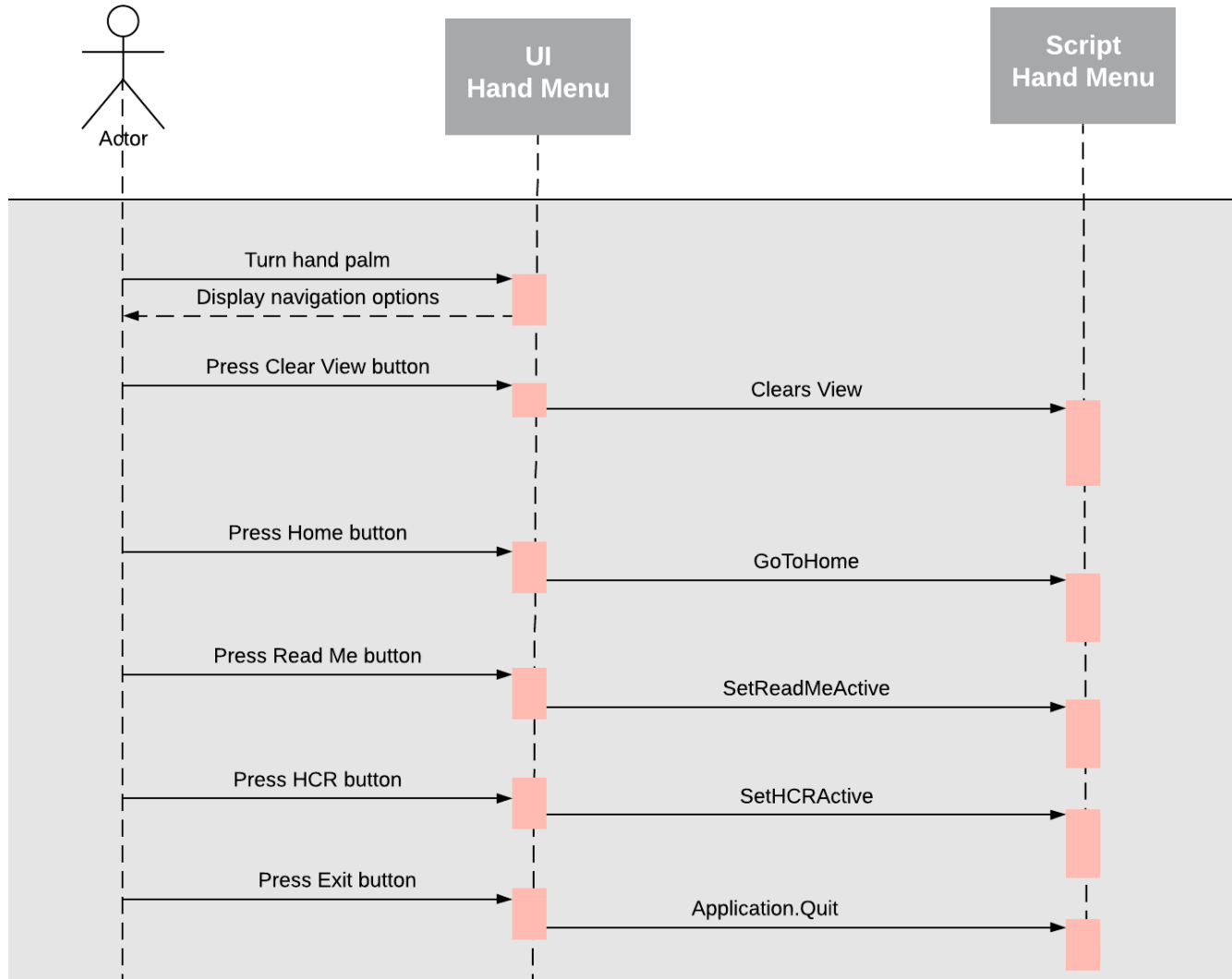


Figure 38 - Sequence Diagram: Hand Menu

5.6 Flow Chart

The flow chart diagram is used to map and explain what choices the user can encounter during the use of the application. It also describes a high-level relationship between different views in the application.

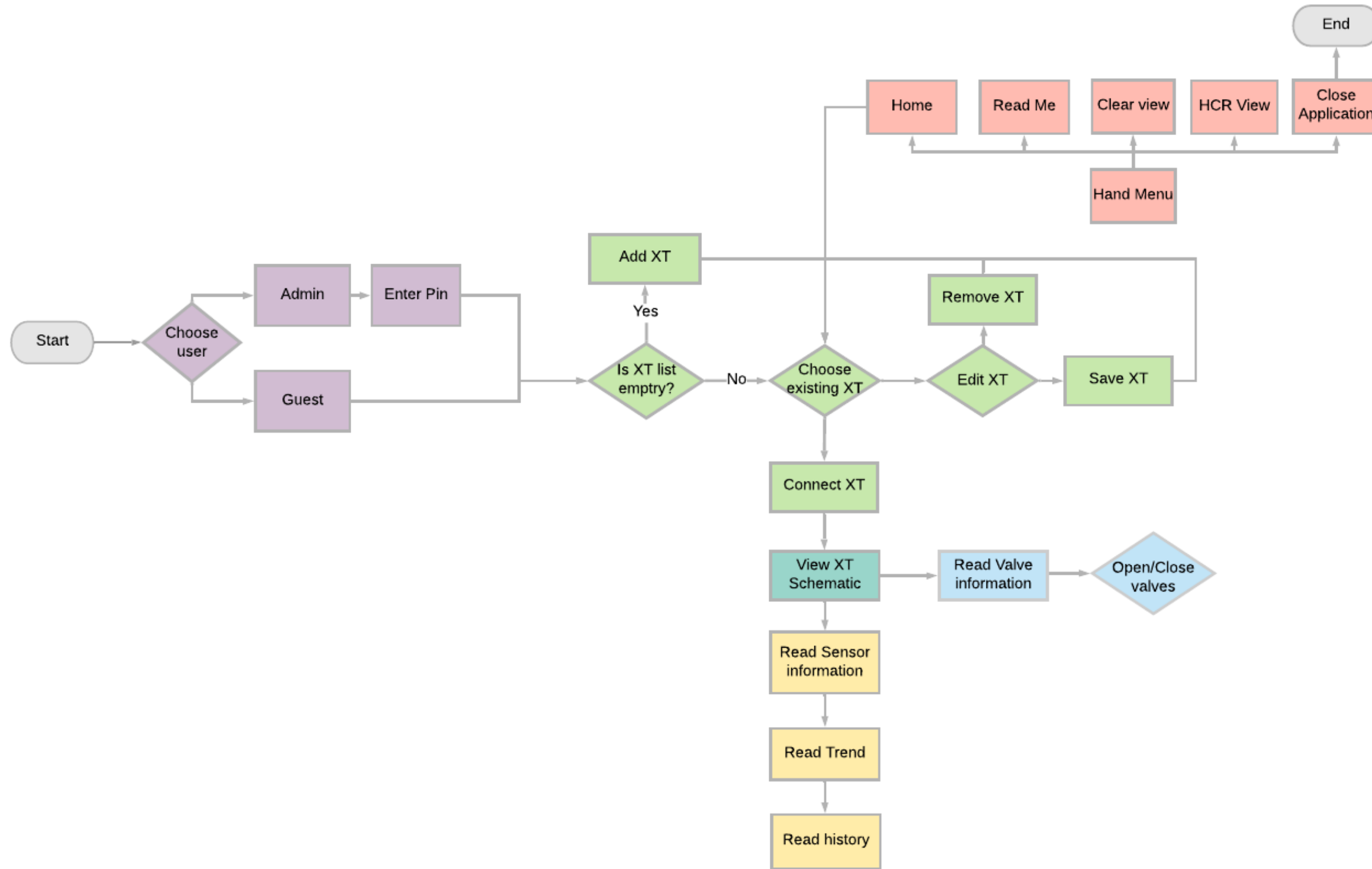


Figure 39 - Flow Chart: System Overview

6 The Implementation

In this chapter we will explain how the application is built and assembled. For more detailed documentation see the attached Doxygen document with code comments and source code.

The implementation is divided into two parts. Technical and User Interface. In the technical part the focus is on the functionalities in the application, and in the User Interface implementation the focus is on the aesthetic and the visual elements that the user is enabled to interact with.

Technical

The technical areas were divided into five main areas, which was distributed between the group members. XT Operation is not a part of the colour code.

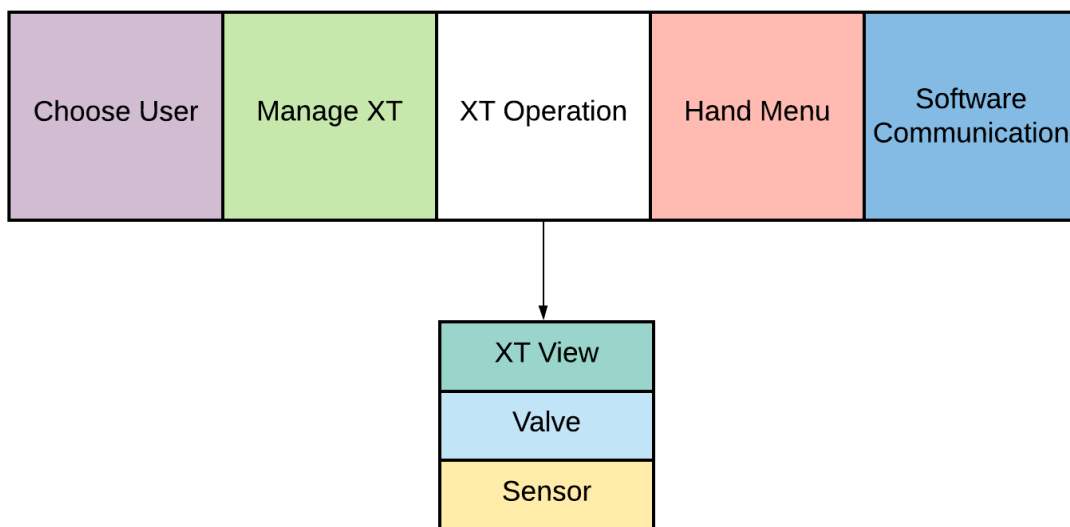


Figure 40 - Technical Areas

User Interface

All technical areas follow the same standard for UI. The implementation of UI is divided into two main areas.

- Interactable elements
- Design

Interactable elements is implemented at the same time as the functionalities as this area cover buttons and views, while design is done after the technical is implemented as this cover among other things background-colour, what icons to use and where the button is placed.

6.1 Technical

6.1.1 Global Manager

Global Manager is a static class used to hold information used throughout the system.

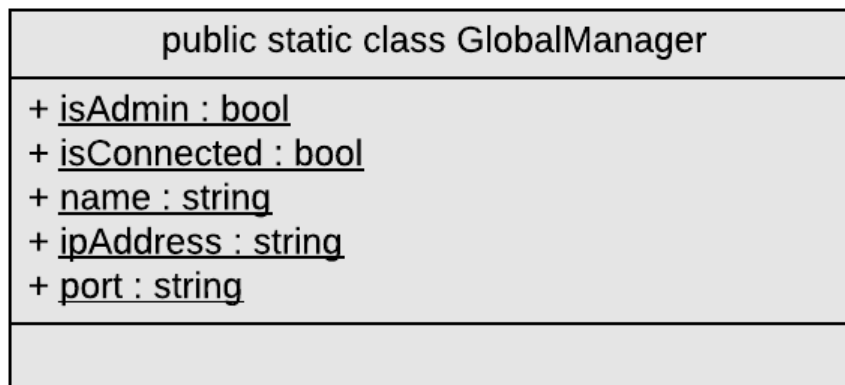


Figure 41 - Class Diagram: GlobalManager

6.1.2 Choose user

The choose user menu is the first part of the application that the user interact with. The menu has two options; Admin and Guest.



Figure 42 - Choose User Menu

The purpose of this feature is to make sure that only an authorized user can do secure valve operation. As a solution, an Admin is required to use a PIN code to log in. Otherwise, both users have access to the same information and functionalities.

6.1.2.1 Interactable PIN Code Menu

The Pin Code is six digit long and take the user directly to the next view of the application when the right pin is typed in. The Pin Code menu consists of:

- Pin Buttons, 0-9
- Delete Pin Button
- Input Field
- Close button



Figure 43 - Pin Code Menu

The Pin Buttons in the Pin Code Table is organized in a grid system. This is implemented by adding the script component: Grid Object Collection, to the game object PinCodeButtons. The Grid system makes it easier to organize the buttons after custom demands as the system lets the user choose the number of columns and cell width/height.

The pin code in the InputField is censored with the character "*" because it should not be possible for others to see the typed in pin code. This is done in Unity by choosing Pin in Content type under the scrip TMP_InputField which is a component of the object TMP InputField.

6.1.2.2 Class Diagram: PinCode

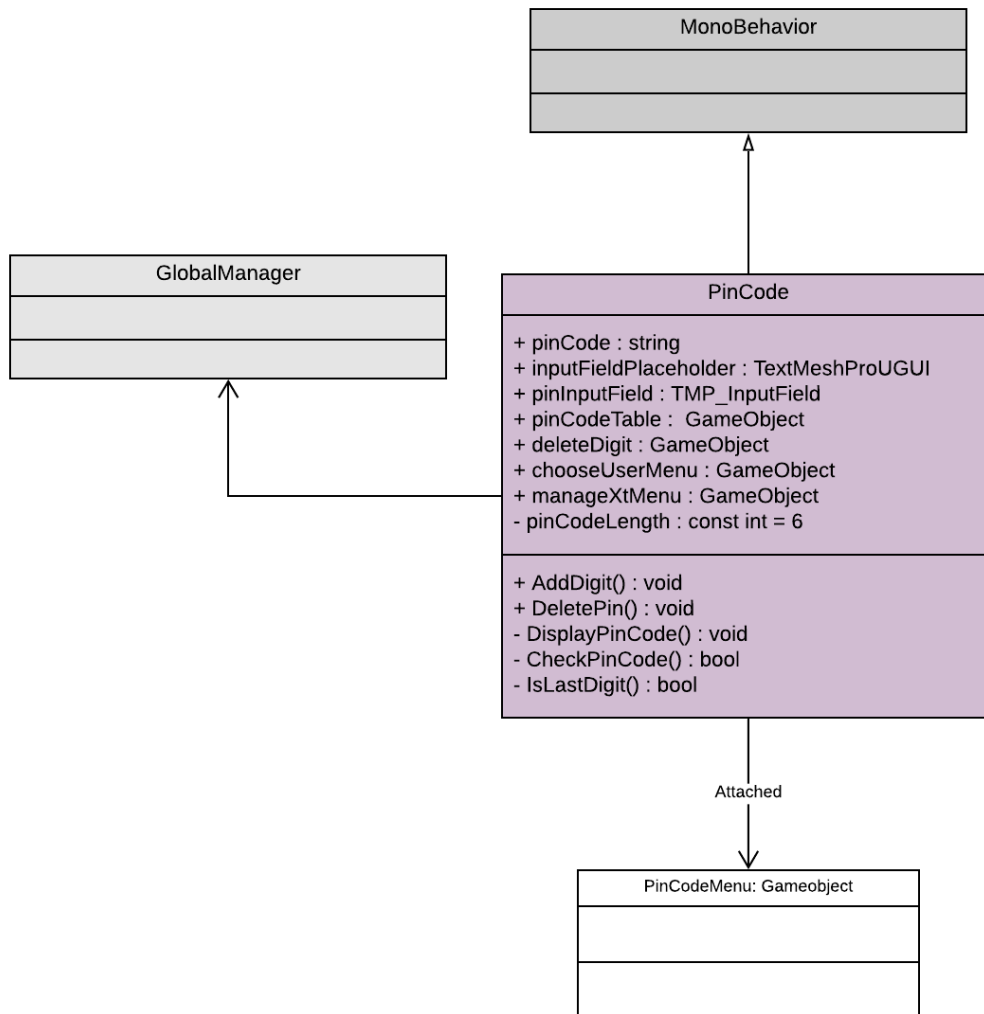


Figure 44 - Class Diagram: PinCode

When a pin button is pressed the pin digit is added to an empty string and displayed in the input field. When the last pin is added, the typed in pin is hashed and verified against the stored hashed pin. If the Pin is valid the application takes the user to the next part of the application, Manage XT. This is showed in the sequence diagram, Figure 45

The pin is hashed with salt using BCrypt. The NuGet BCrypt-Official was used. For detailed instructions implementation, see [37] [38].

6.1.2.3 Sequence Diagram: PinCode

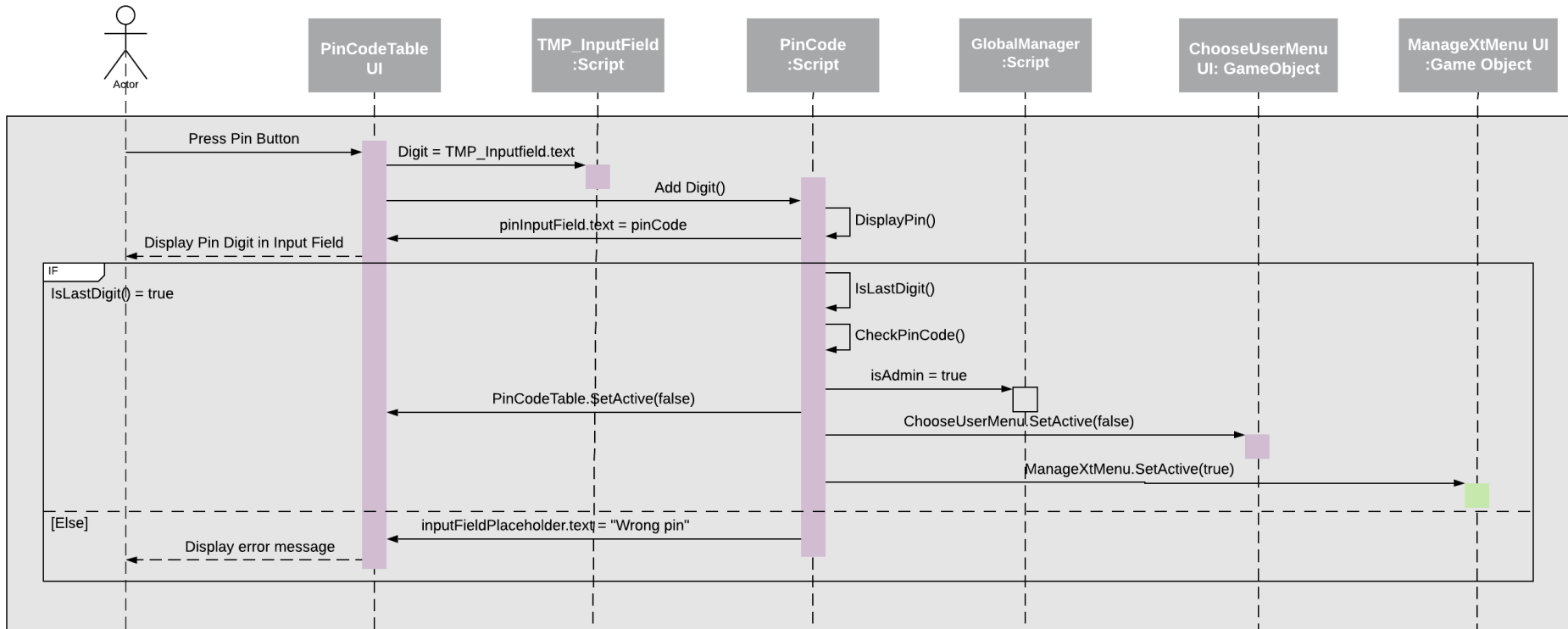


Figure 45 – Sequence Diagram: Pin is typed in

6.1.3 Manage XT

The Manage XT menu allows the user to manage available XTs. Opening the application for the first time, the Manage XT menu will be empty. After adding a new XT, it will be displayed in the Manage XT menu as a XT prefab as seen in Figure 46

The user can then choose to edit, remove or connect. All the XTs are stored in a text file in json format, so the user only must configure once. Figure 50 shows the structure of the text file.



Figure 46 - Manage XT View with XT Prefabs

The manage XT Menu consist of

- Add Button
- Edit Button
- Remove Button
- Connect Button

6.1.3.1 Class Diagram Manage XT

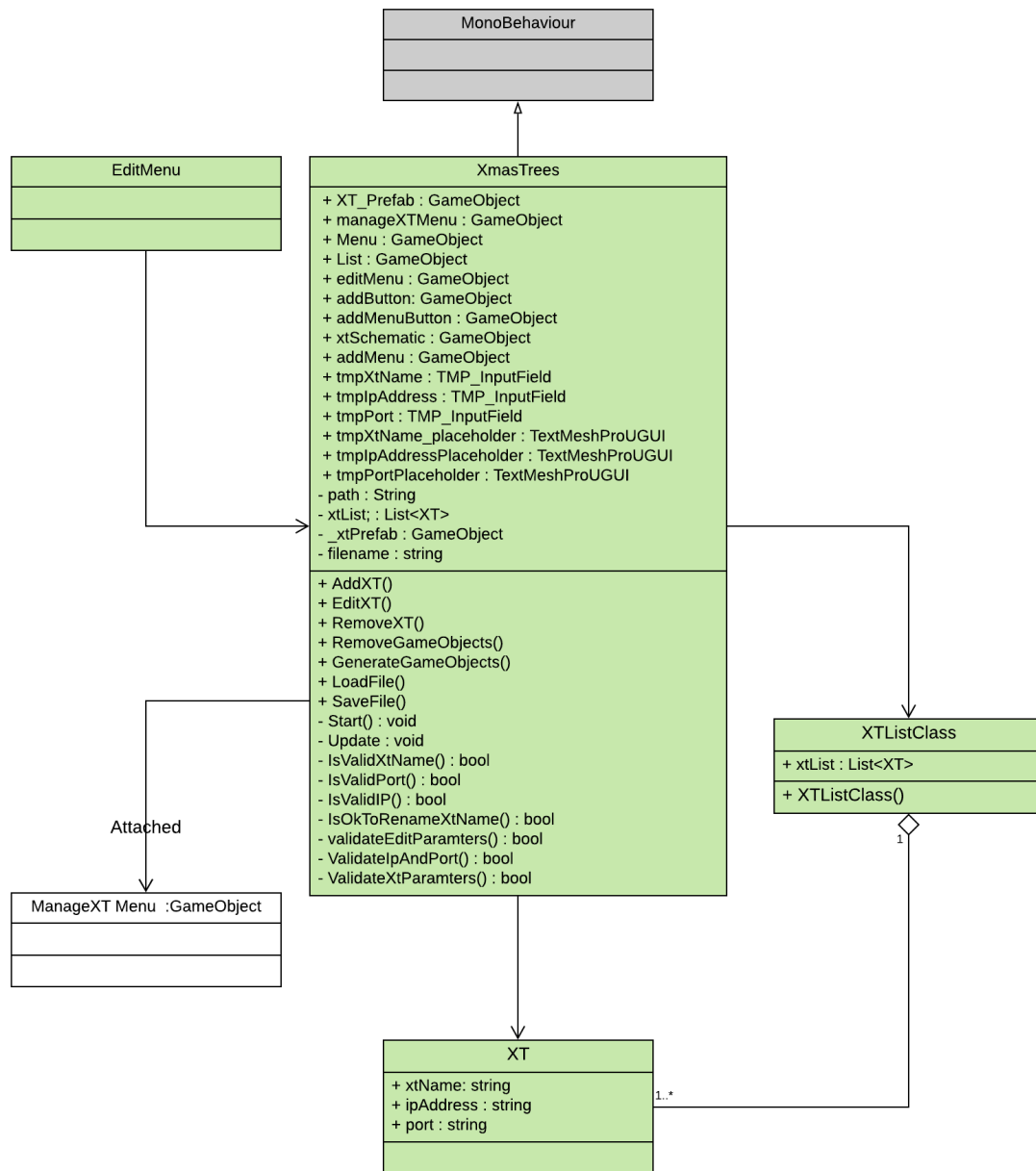


Figure 47 - Class Diagram: XmasTrees

6.1.3.2 Class Diagram PrefabXT

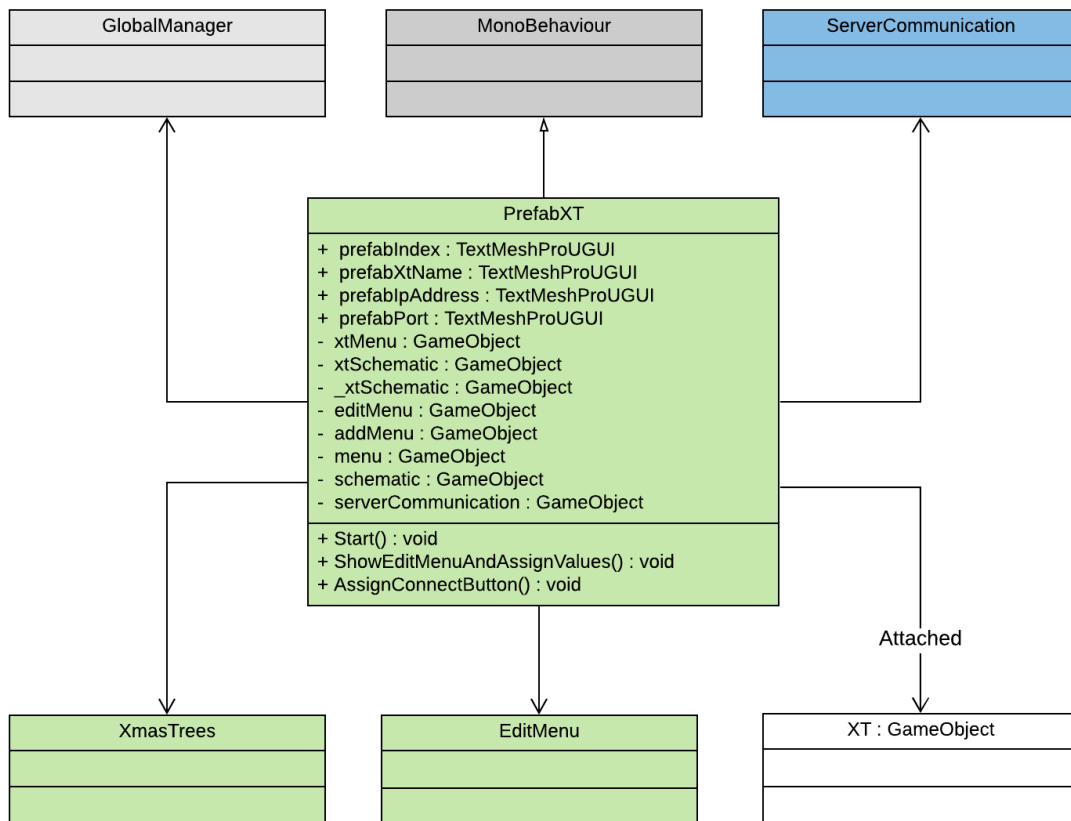


Figure 48 - Class Diagram: PrefabXT

The XmasTrees script is meant as the main class to manage the XT. It allows for edit, remove and add operations in addition to functionality for validating input parameters, saving/loading to text file and generating the list of XT objects in the Manage XT menu. Each XT object will be instantiated as a XT prefab. A XT prefab will contain the following.

- Edit button.
- Connect button.
- Properties (Index, Name, IP, Port).

The PrefabXT script is also attached to assign functionality to the buttons, the class diagram can be seen in Figure 48.

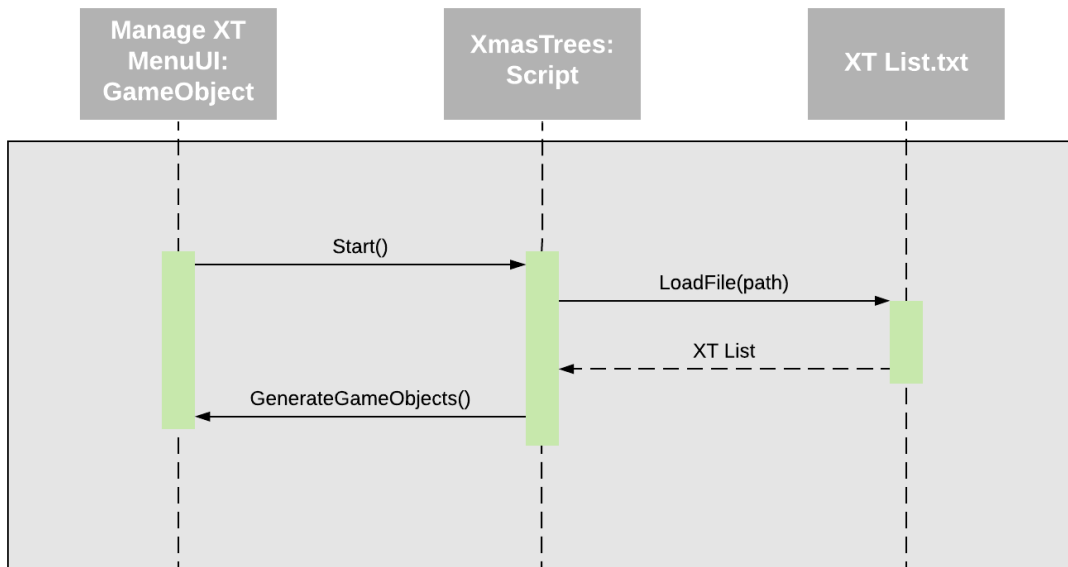


Figure 49 - Sequence Diagram: Manage XT

```

{
  "xt_list": [
    {
      "xt_name": "Pyxis",
      "ip_address": "██████████",
      "port": "██████"
    },
    {
      "xt_name": "Cassiopea",
      "ip_address": "██████████",
      "port": "██████"
    }
  ]
}

```

Figure 50 - Structure of XT list text file

6.1.3.3 Add XT

Pressing the Add XT button in the Manage XT menu opens the Add XT menu as seen in Figure 51. Pressing Add XT in this menu will start the sequence in Figure 52.



Figure 51 - Add New XT Menu

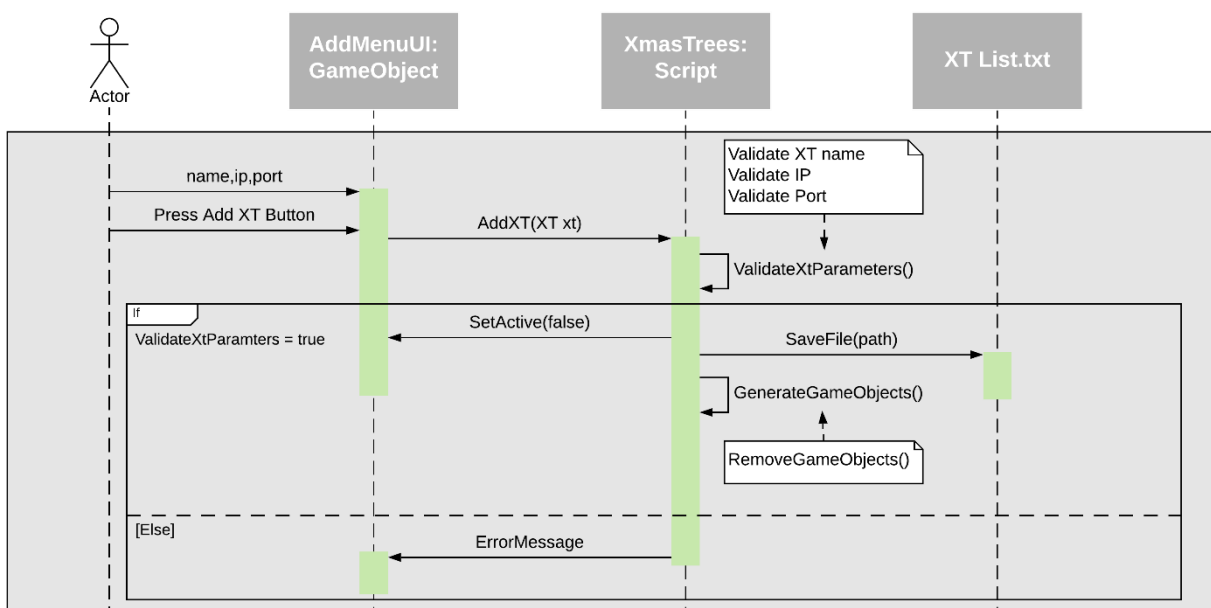


Figure 52 - Sequence Diagram: Add XT

6.1.3.4 Edit XT

Opening the Edit menu in Figure 53 allows the user to edit or remove already existing XTs. The menu will have the properties of the chosen XT attached. Attached to the Edit menu GameObject is the EditMenu script which controls the functionality of the menu. Pressing save will start the sequence in Figure 55.



The image shows a screenshot of a mobile application interface for editing an XT. The interface is titled "Edit" and has a dark blue background. At the top right, there is a close button with an "X" icon and the word "Close" below it. The main content area contains three text input fields, each with a label above it: "Name", "IP Address", and "Port". Each input field contains the placeholder text "Enter text...". At the bottom of the form, there are two buttons: "Remove" on the left and "Save" on the right.

Figure 53- Edit XT Menu

6.1.3.5 Class Diagram: EditMenu

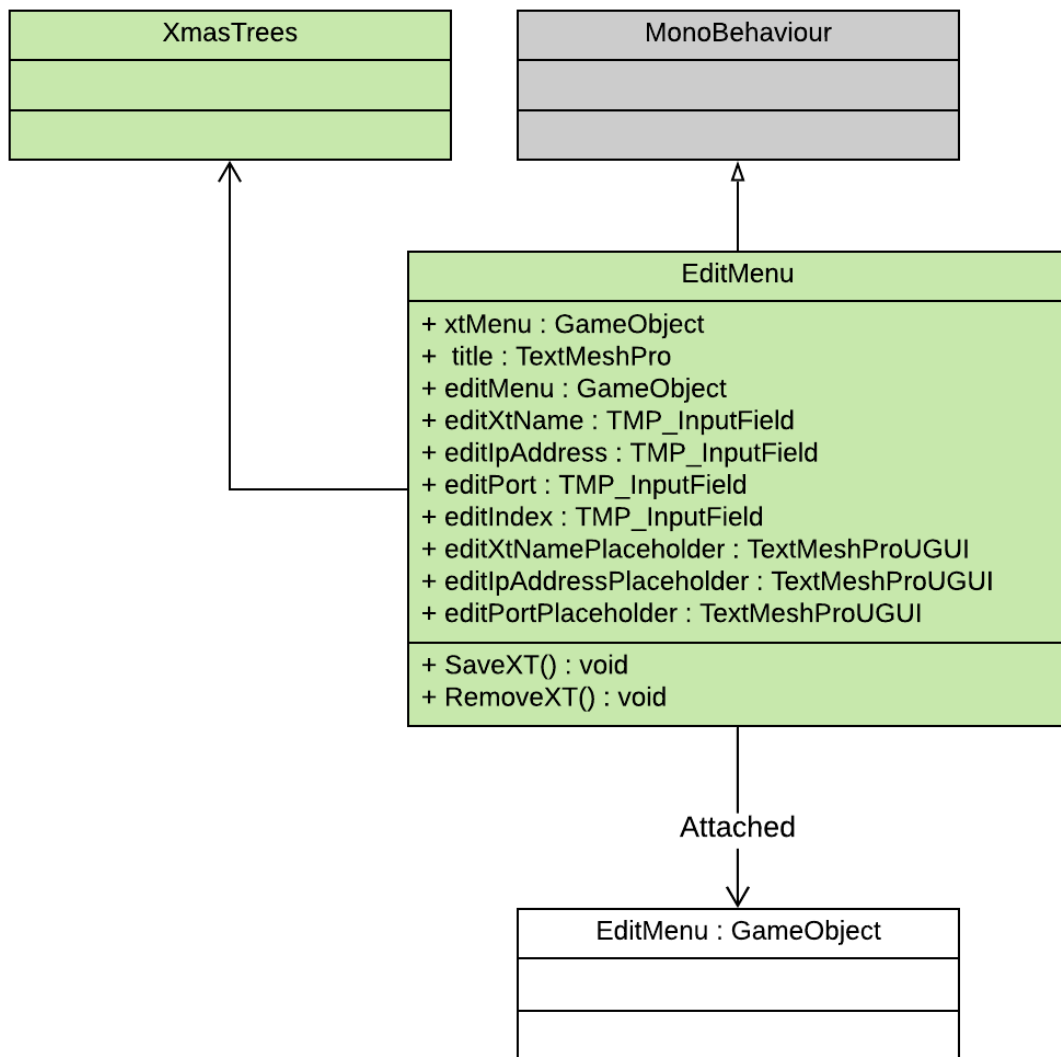


Figure 54 - Class Diagram: Edit Menu

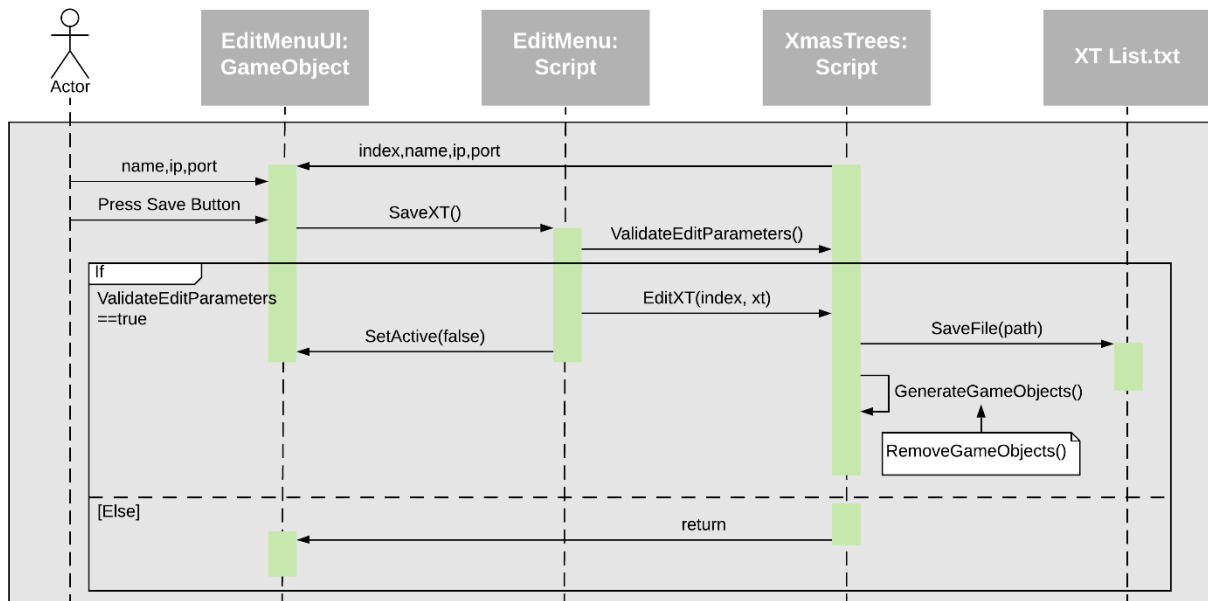


Figure 55 - Sequence Diagram: Edit XT

6.1.3.6 Sequence Diagram: Remove XT

Explains the sequence when removing a XT from the list.

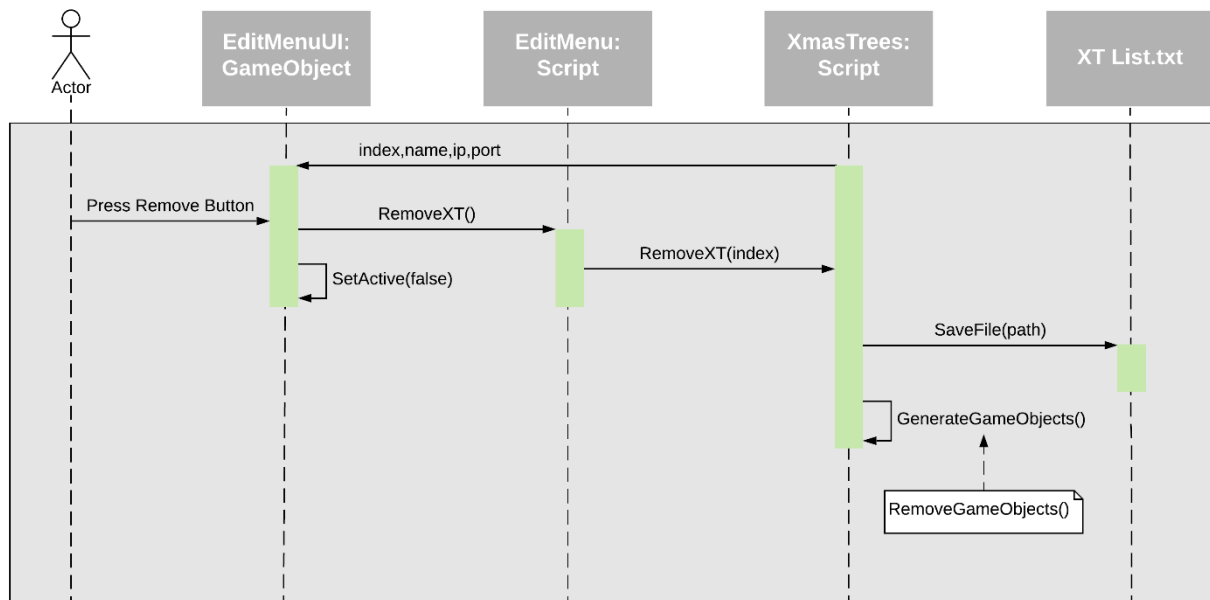


Figure 56 - Sequence Diagram: Remove XT

6.1.3.7 Connect XT

Figure 57 shows the sequence after the connect button on a XT is pressed.

- Set GlobalManager variables to selected XTs variables (name, IP, port).
- Enable ServerCommunication script (see Software Communication for more details).
- Instantiate the XT Schematic (see XT Schematic).
- Deactivate the Manage XT Menu.

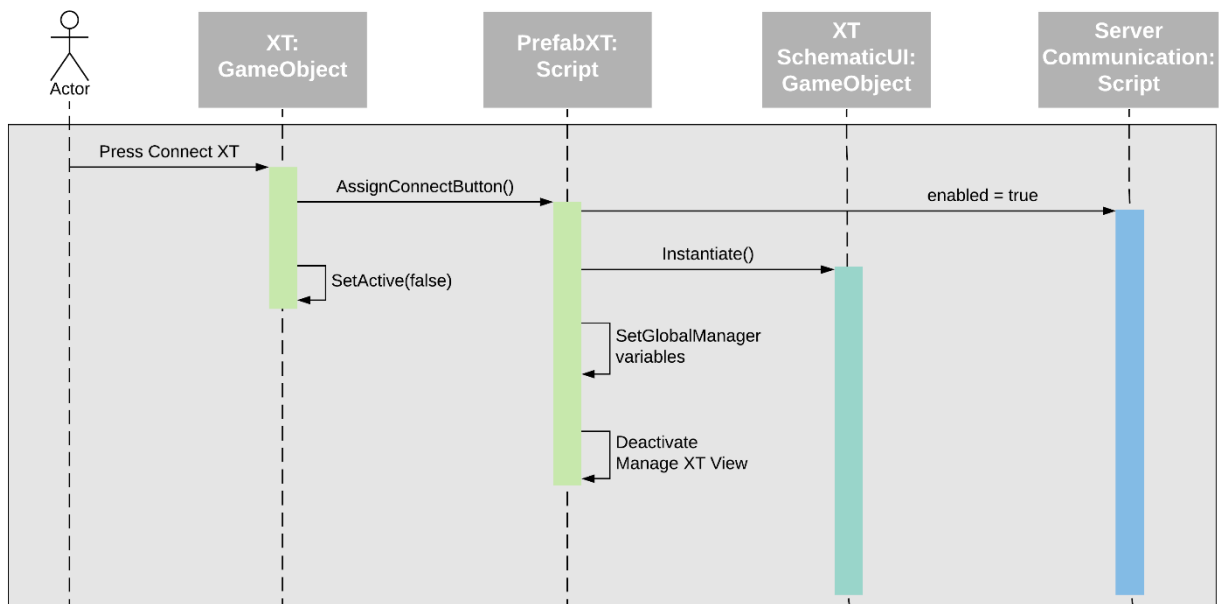


Figure 57- Sequence Diagram: Connect To XT

6.1.3.8 Error handling

Error messages has been implemented to check if:

- XT Name, IP and Port is empty
- XT Name already exist in XT list
- IP is invalid
- Port number is invalid
- Maximum numbers of XT's in XT list

6.1.4 XT Schematic

The XT schematic is a solution made to display the P&ID, see Figure 58. The sub system instantiates every component located in the P&ID. Including, but not limited to:

- Valves
- Sensors
- Chemical Injection Trees

The solution is made to satisfy the functional requirement of being able to display XT information, and the non-functional requirement of being able to see similarities between this application and existing TechnipFMC applications.

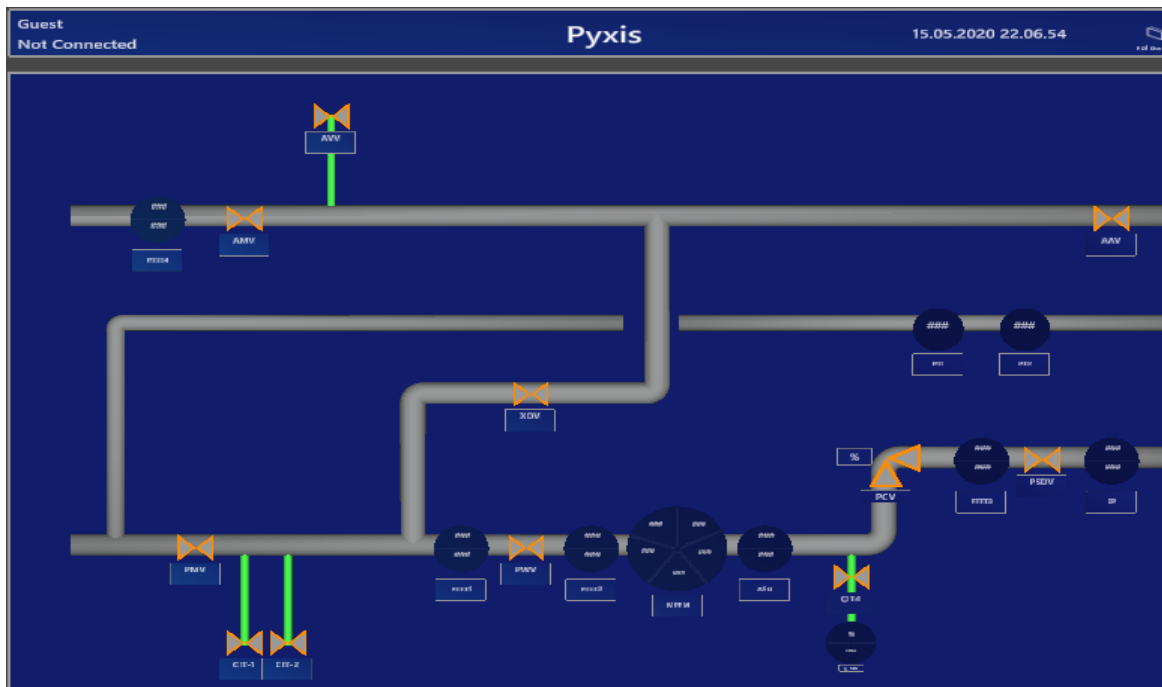


Figure 58 - Populated XT Schematic (Pyxis)

6.1.4.1 Class Diagram: XT Schematic

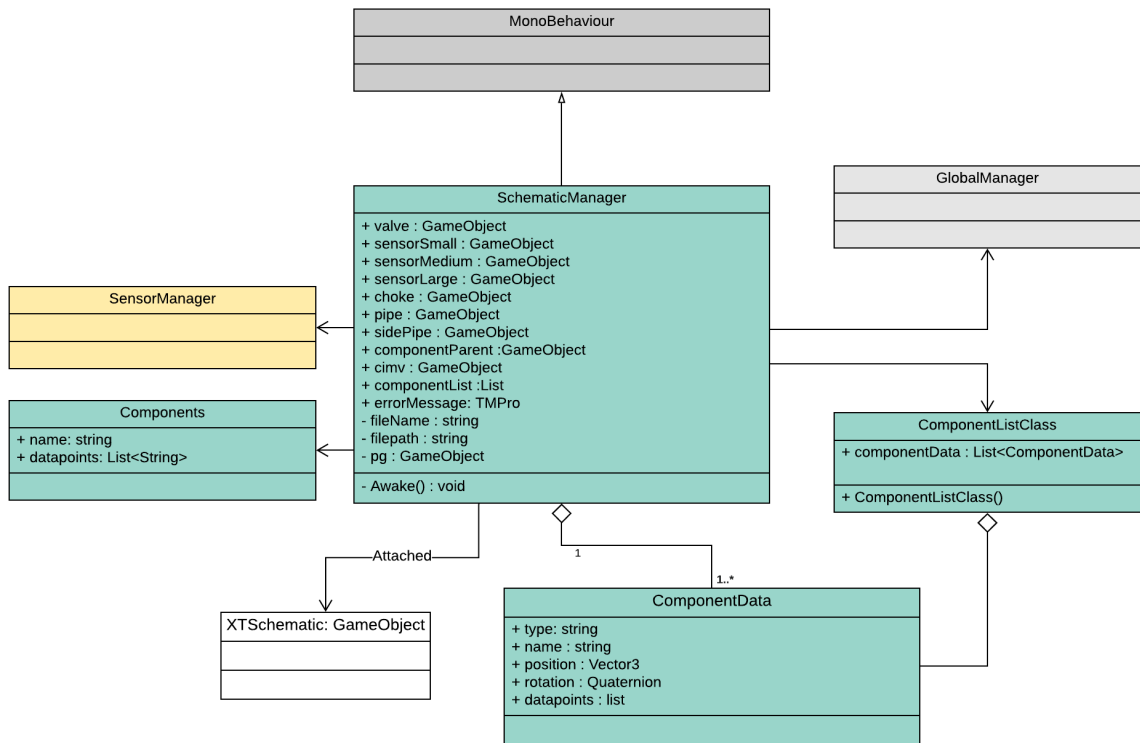


Figure 59 - Class Diagram: XT Schematic

6.1.4.2 Piping

There are a lot of different ways to satisfy the requirement of displaying XT Information. Because AR have unlimited 3D space with the HL2 technology, and TechnipFMC wanted to see similarities between existing applications, the decision fell on making a 3D model of the static pipes that every XT is based upon, see Figure 60. TechnipFMC have a program designed to draw out XT Schematics in 2D see, Figure 61

The pipes are a background to spawn the different components, i.e. valves sensors and injection pipes, based on what configuration is needed for the specific XT.

To see different implementation options for pipe drawing see, Schematic Decisions

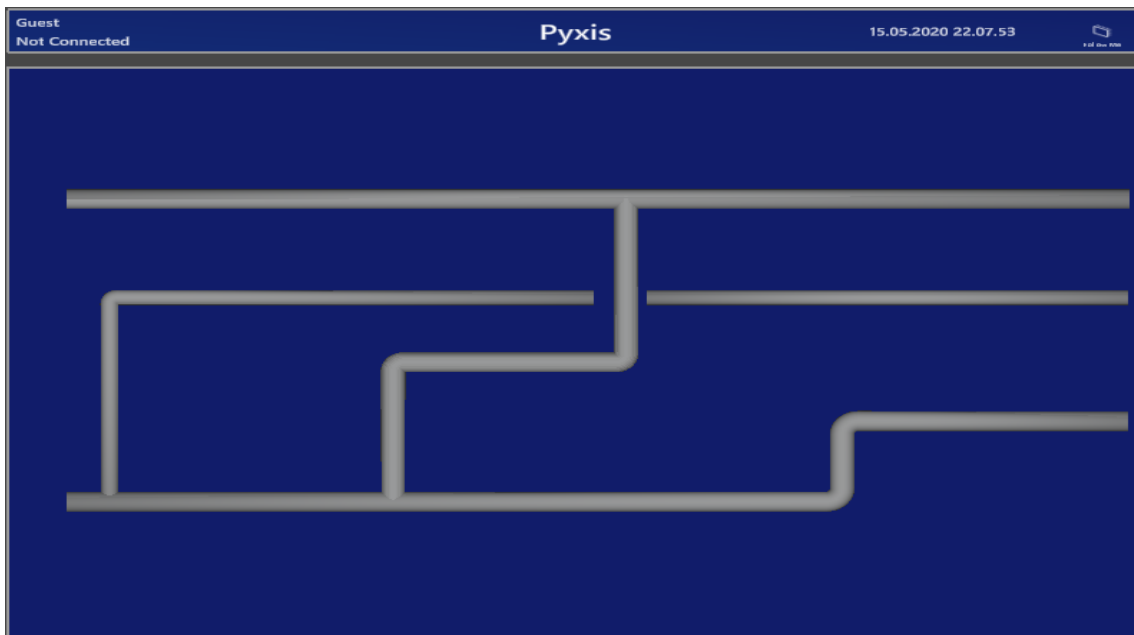


Figure 60 - Empty XT Schematic

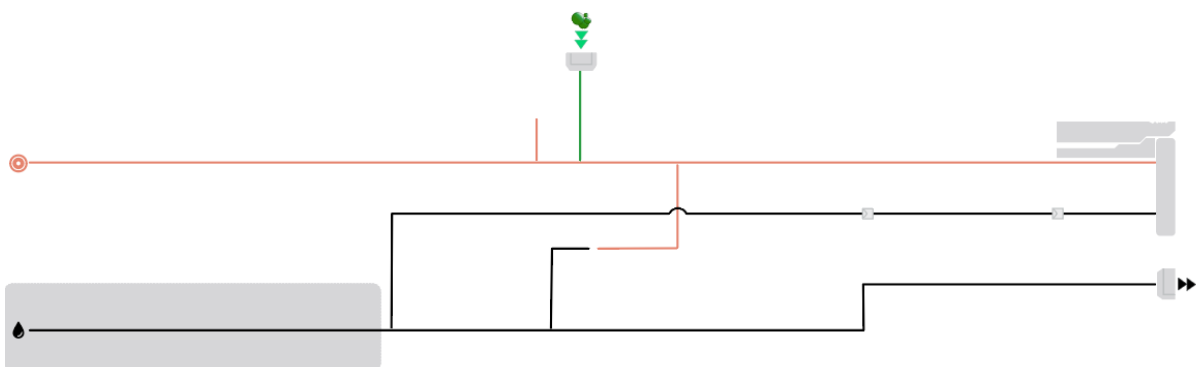


Figure 61 - Template piping from TechnipFMC software

6.1.4.3 Json Structure

One of the biggest challenges in this section was figuring out how to place the components. The decision fell on using a json file to store information about the type, name, position, rotation and datapoints see Figure 62.

This made it able to configure a json file to move the components around and have multiple json files corresponding to the number of XT's added to the system. How these files can be added and made can be seen in **Feil! Fant ikke referansebildene..**

Json is also the preferred format at TechnipFMC. Using Json makes it easier for them to implement it with their systems later.

```
{
  "componentData": [
    {
      "type": "Valve",
      "name": "PMV",
      "position": {
        "x": 0.1436,
        "y": 0,
        "z": -0.01
      },
      "rotation": {
        "x": 0,
        "y": 0.0,
        "z": 0,
        "w": 0
      },
      "datapoints": [
        "/XT_HCR.ks800fb/Valve_102.cmdStatus",
        "/XT_HCR.ks800fb/Valve_102.cmd",
        "/XT_HCR.ks800fb/Valve_102.pressure"
      ]
    }
  ],
}
```

Figure 62 - Json structure

By changing up the position values in the Json file a completely different Schematic can be displayed, see Figure 63.

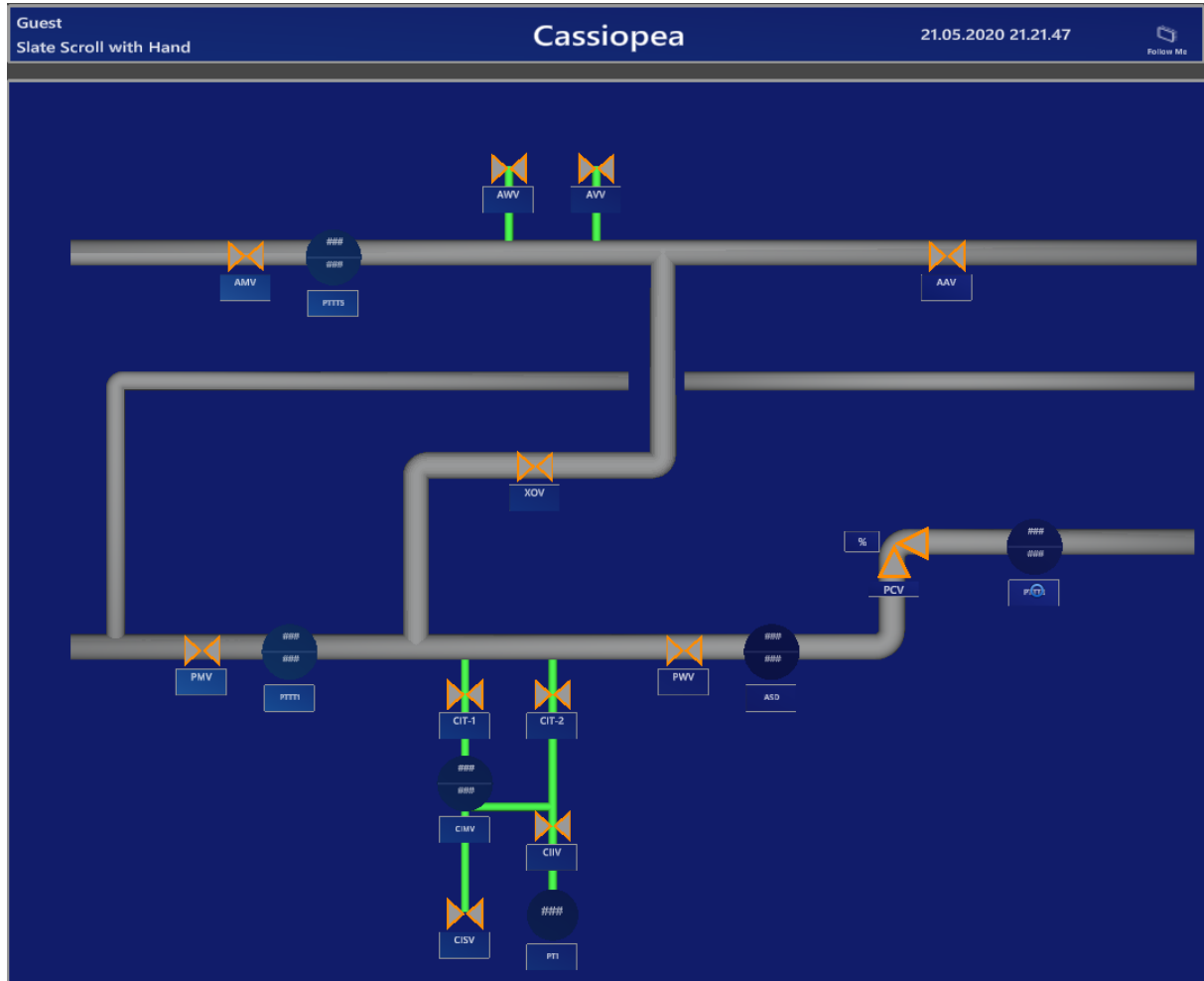


Figure 63 - Populated XT Schematic (Cassiopea)

6.1.4.4 Sequence Diagram: Population of XT Schematic

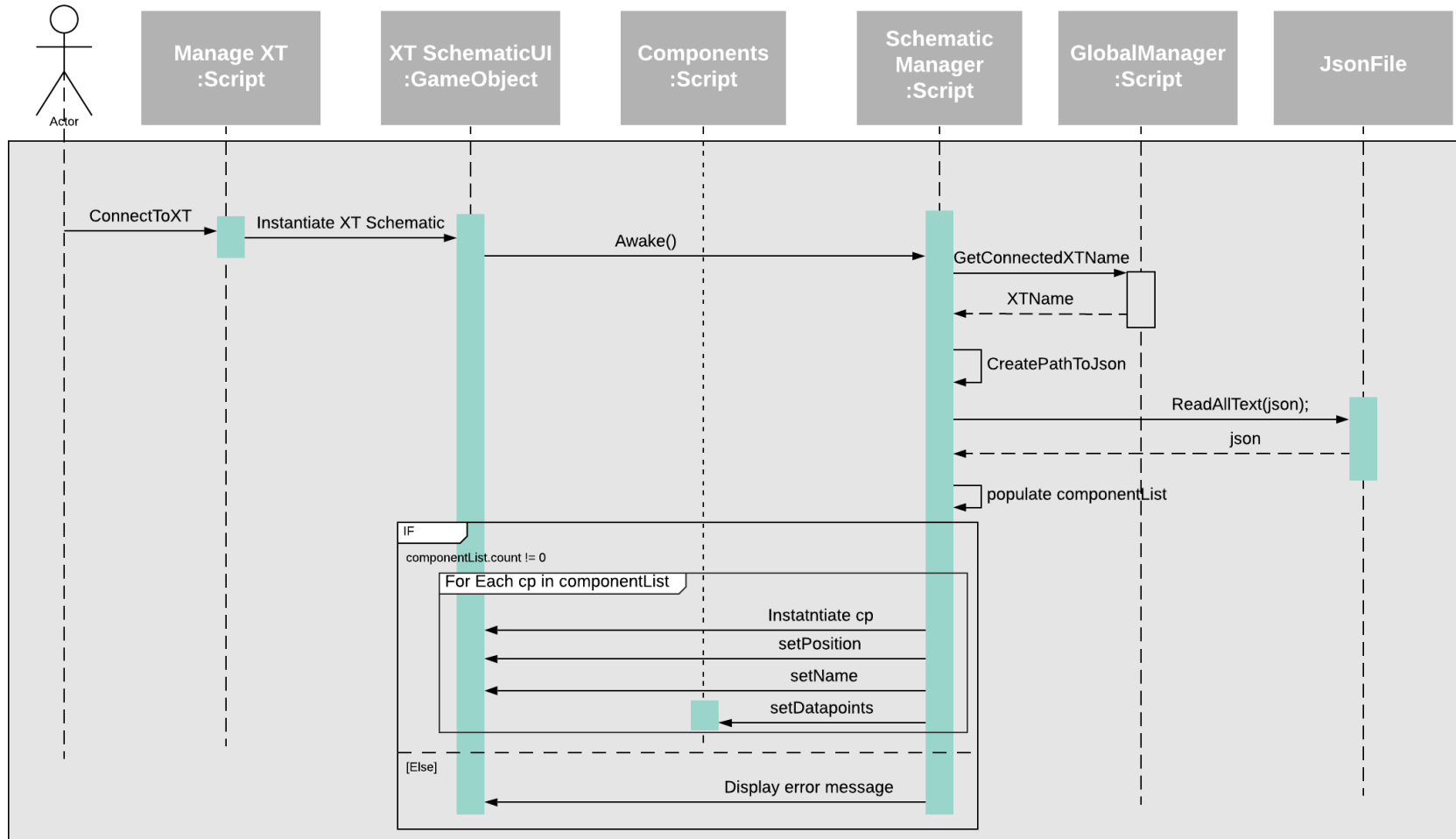


Figure 64 - Sequence Diagram: XT Schematic

6.1.5 Valve Control

This section will explain the implementation of Valve Control. The section derives from the functional requirements “Secure valve operation” and “Read valve information”.

6.1.5.1 Class Diagram: ValvePrefab

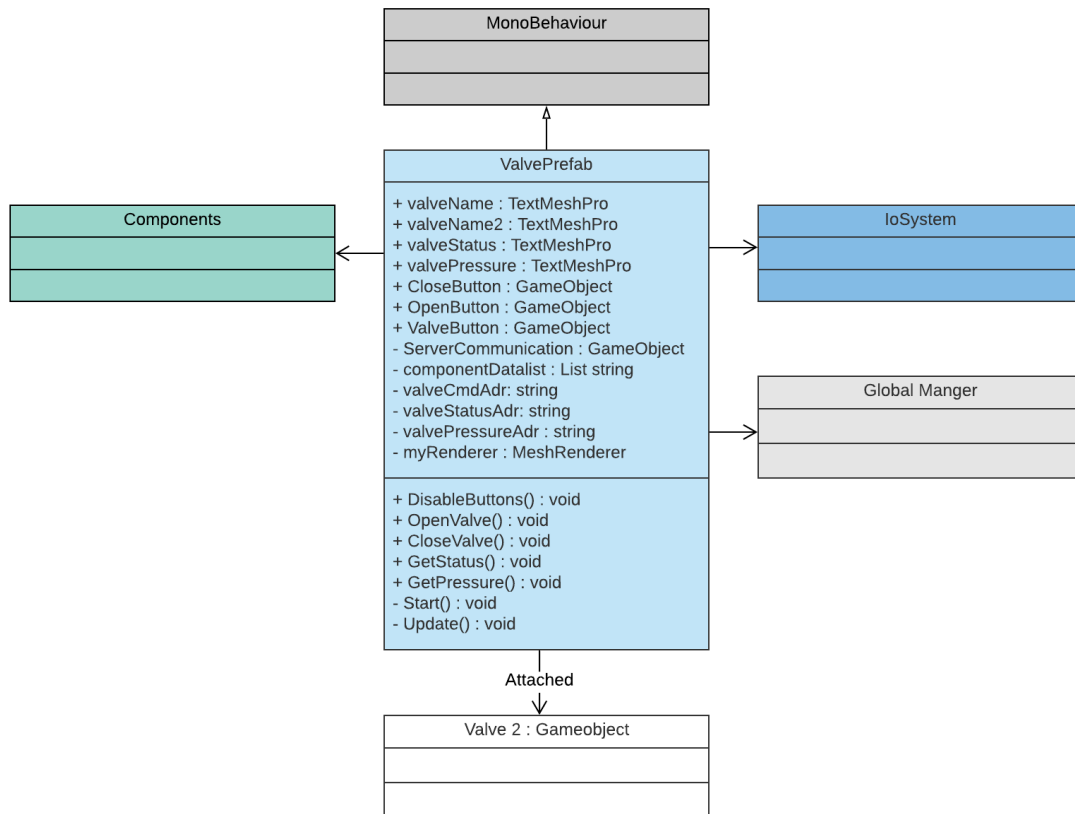


Figure 65 - Class Diagram: ValvePrefab

6.1.5.2 Class Diagram: ChokePrefab

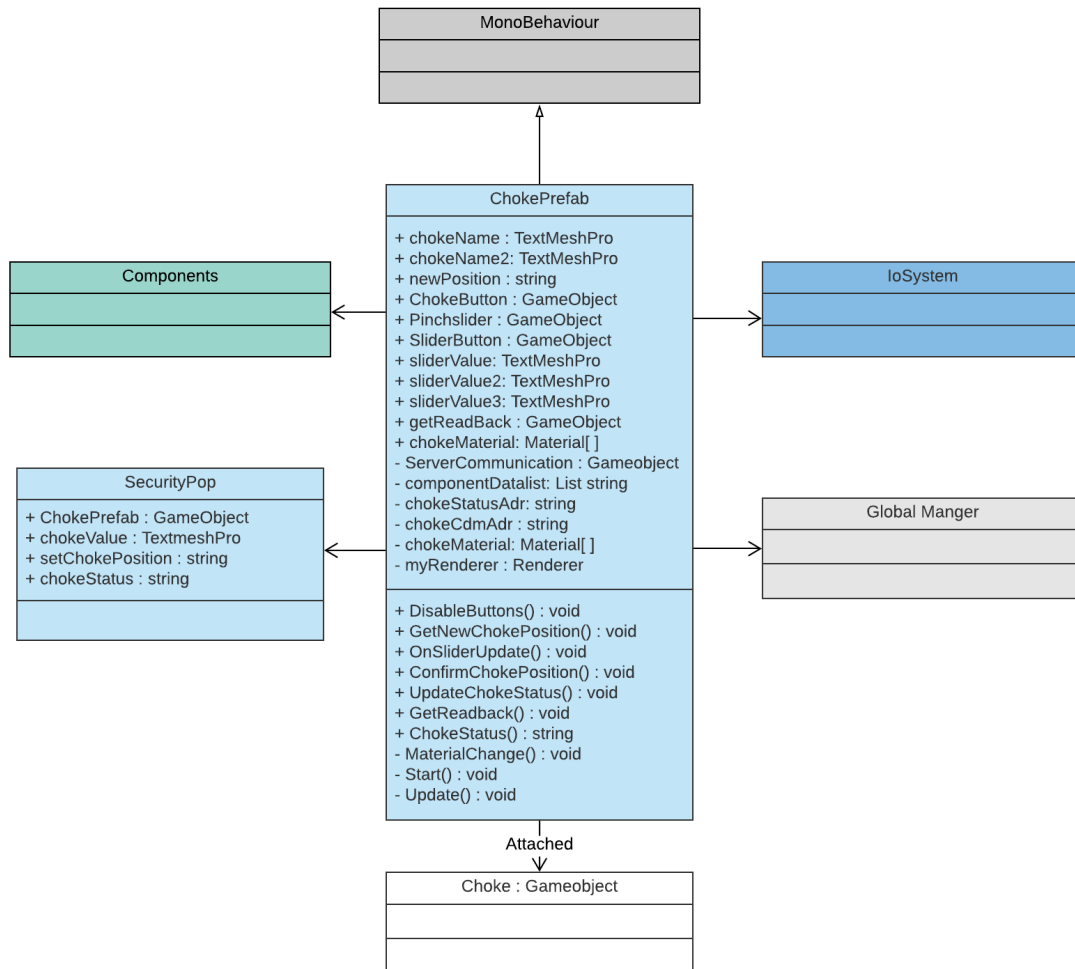


Figure 66 - Class Diagram: ChokePrefab

6.1.5.3 Class Diagram: CovPrefab

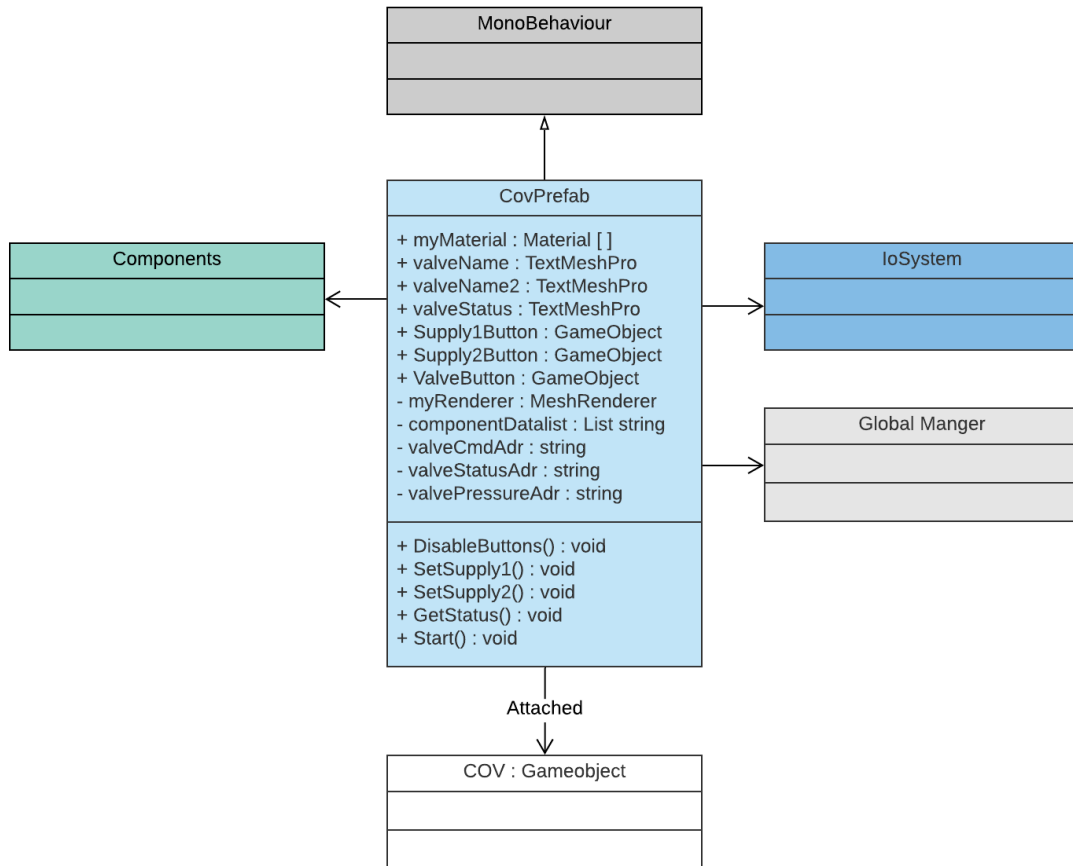


Figure 67 - Class Diagram: CovPrefab

6.1.5.4 Class Diagram: CIMV

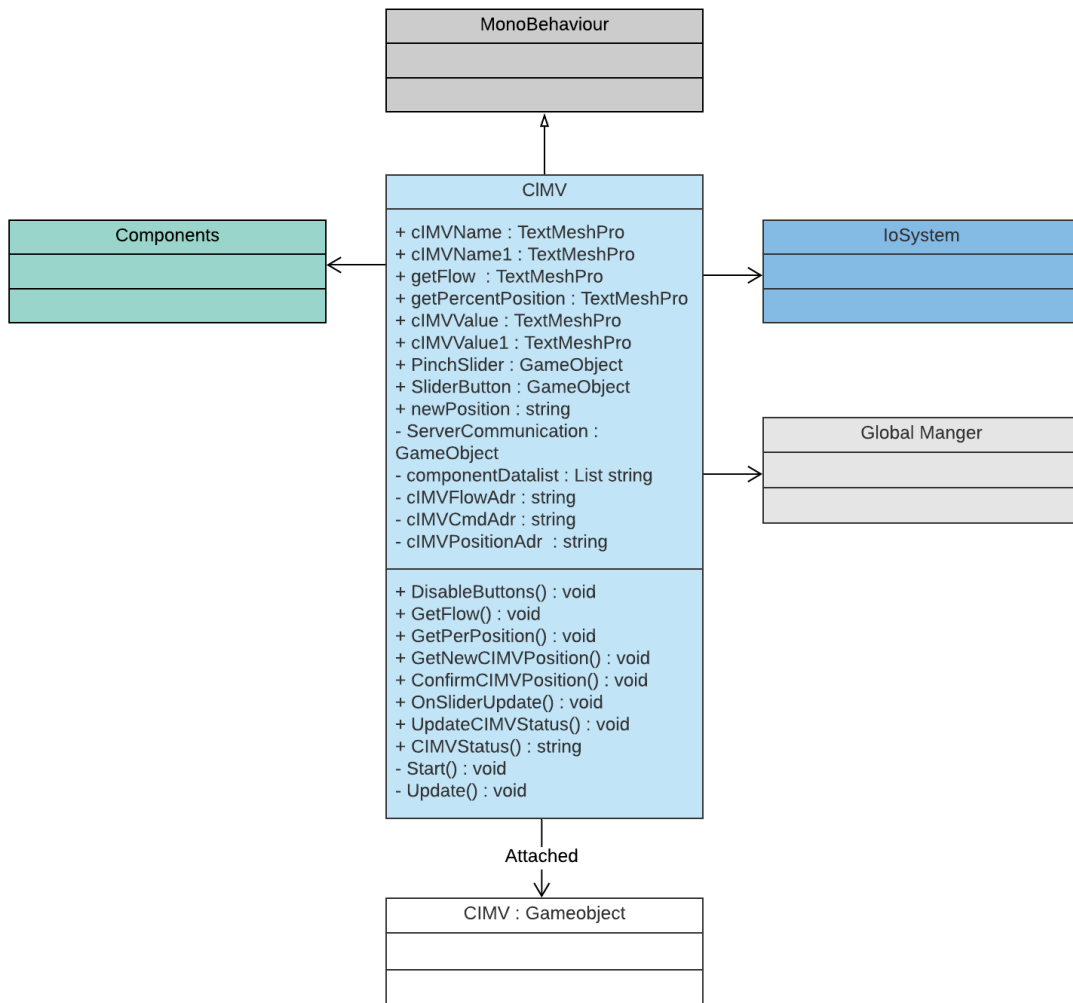


Figure 68 - Class Diagram: CIMV

6.1.5.5 Flow Diagram: Open/Close Valve

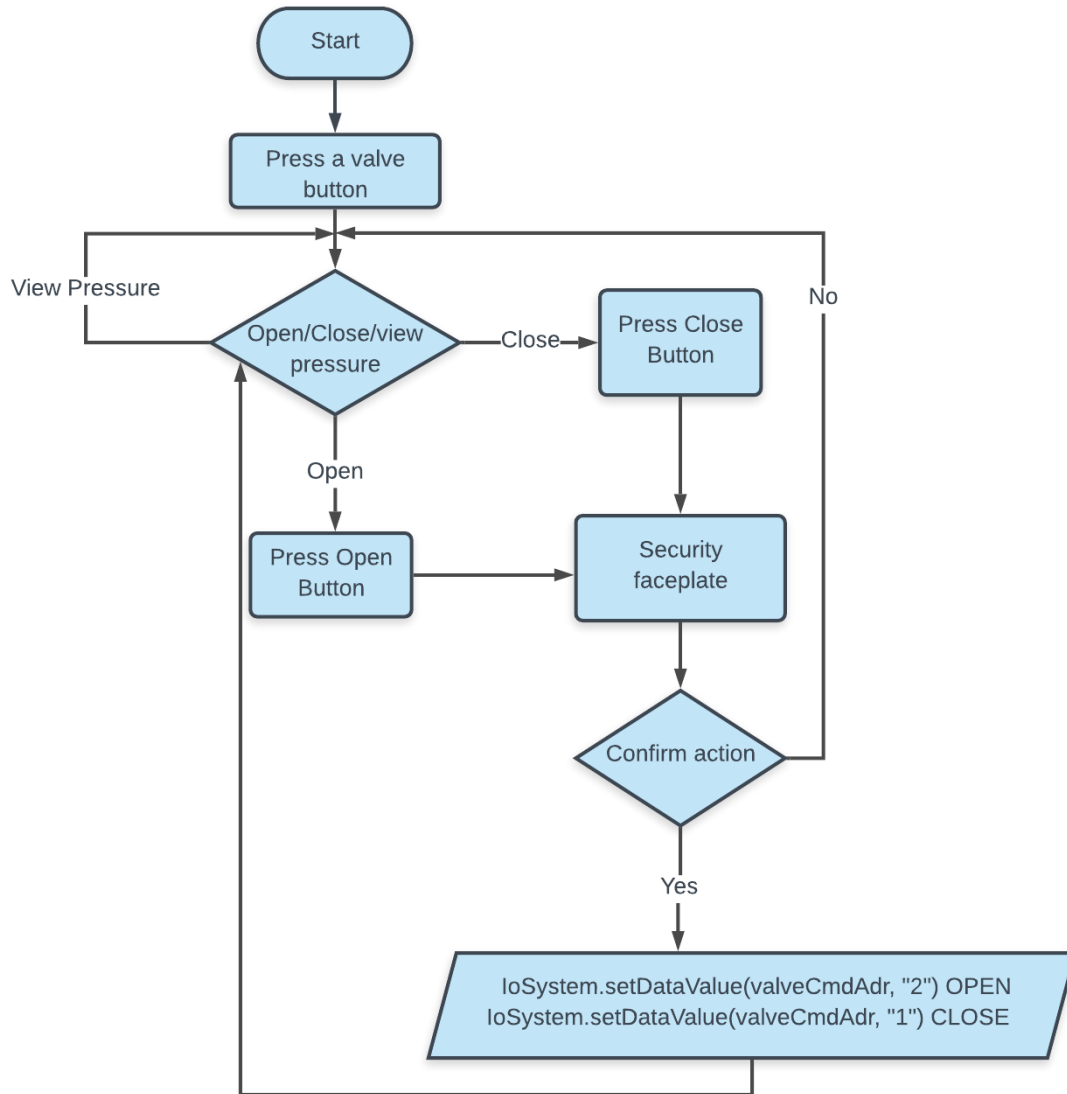


Figure 69 - Flow Diagram Open/Close Valve

6.1.5.6 Flow Diagram: Adjusting the choke

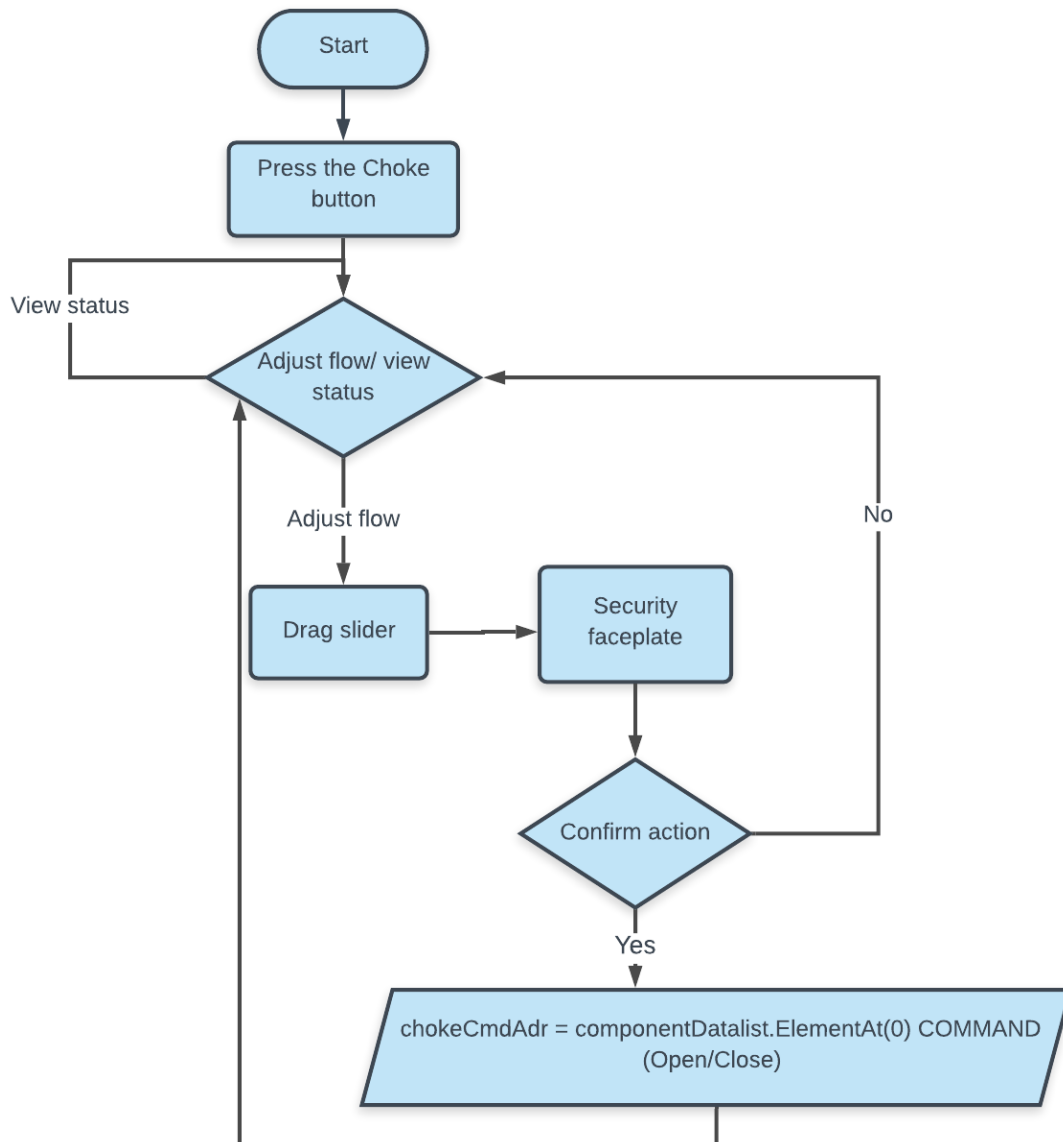


Figure 70 - Flow Diagram Adjusting Choke

6.1.5.7 Visuals and description

Valve prefab

Model:

- Name field
- Colour
- Shape



Figure 71 - Valve Model Closed



Figure 72 - Valve Model Open

The valve prefab consists of the model designed as a classic P&ID valve symbol. The status of the valve is represented by the colour. When fluid is running through the valve, the valve is filled in and the status is open. When only the outline is visible, fluid is not flowing through and the valve is closed. The model also showcases the name of the valve in a square namespace.

Valve Menu:

- Name field
- State field
- Pressure field
- Exit button
- Guest/Admin



Figure 73 - Valve Menu

The valve menu consists of a field for name, state, and pressure. It also includes an “Open” and a “Close” button. When either of the command buttons are pressed, a face plate pops up. This face plate works as a security feature, so that a valve doesn’t open or close by accident. This feature derives from a requirement for safe operation when using AWC. When a user is logged in as a guest will the command buttons be set to a false active, and not appear on the menu. The pressure and status view will still be visible.

Security face plate:

- “Confirm action”
- Yes button
- No button



Figure 74 - Security Face Plate

Choke prefab:

Model:

- Name field
- Colour
- Shape
- Status field

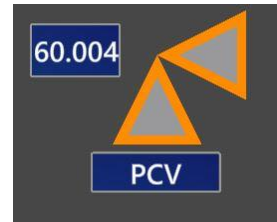


Figure 75 - Choke Prefab Model

The model is also a classic P&ID symbol for an angle valve. Since the choke always sits in an angle in our schematics was the P&ID symbol a natural choice model. The model also displays the name of the choke and the status of opening in percent in the XT-view.

Choke menu:

- Name field
- State in percentage
- Slider
- Exit button
- Guest/Admin

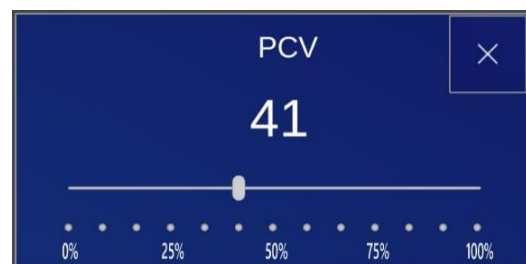


Figure 76 - Choke Menu

The choke menu contains the usual name and exit button in addition the state that the choke is currently in. As soon as the slider button is dragged to a new position the state percentage will change accordingly. This alone doesn't change the state in the simulator.

For the choke menu there is also a safety pop up, where the user can officially confirm the state change, or disregard it. This is done by pressing the "yes" button to confirm or the "No" button, to disregard it on the new face plate. If it's confirmed the value will be set in the simulator and the readback of the change-in-state is displayed in the readback field. If the "No" button is pressed and the state change is disregarded, the face plate pop-up will close, and the percentage display will return to its actual state. If the user is a guest, the slider will be set to a false active, and not appear in the menu.

Security face plate:

- Percentage of the adjusted state
- Readback field
- Yes button
- No button
- Exit button



Figure 77 - Security Face Plate

CIMV prefab:

Model:

- Name field
- Shape
- Status in percentage
- Flow



Figure 78 - CIMV Model

The CIMV valve is a chemical dosing valve and regulates the dosage flow of the chemical fluid to flush the system. The dosage is regulated with a slider which again results in adjustment showed in percentage. The shape of the valve is the same as a sensor in our schematics because that is what TechnipFMC is familiar with and uses in their P&IDs. The flow is also displayed in the XT-view and shows the direct consequence of the regulation of dosage flow.

CIMV menu:

- Name field
- State in percentage
- Slider
- Exit button
- Guest/Admin

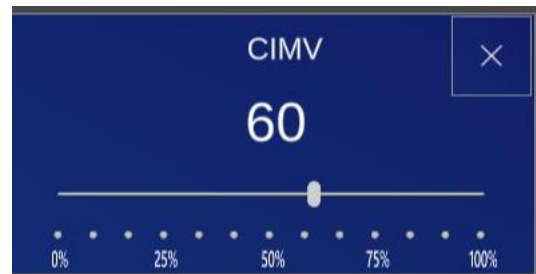


Figure 79 - CIMV Menu

Similar to the choke menu.

Security face plate:

- Percentage of the adjusted state
- Yes button
- No button
- Exit button



Figure 80 - Security Face Plate

The security face plate for the CIMV does not contain a readback because TechnipFMC mainly wanted to use Command MCS's datapoints due to that this is the UI a master control station would have used. Otherwise does CIMV's security face plate contain the same functionality as Choke's security face plate.

COV prefab:

Model:

- Name field
- Colour
- Shape

*Figure 81 - COV Model*

The model is the same as the regular valve but differ in colour to make it clear that it is not a usual valve. The COV can decide which supply line to use in the HCR. The two supply lines are there to ensure operational drift if anything happens to one of them. The operator can then switch supply line to the “back-up”.

COV Menu:

- Name field
- Supply1 button
- Supply2 button
- Exit button
- Guest/Admin

*Figure 82 - COV Menu*

In addition to the usual name field, exit button and the Guest/Admin mode, COV Menu consists of the supply buttons where the user can decide which supply line they want to use.

6.1.6 Sensor Information

Sensor data details how sensor values are presented in the application.

Sensor data is a subsection of the use case XT Operation. Sensor data consist of the three user stories read sensor data, read trend data, and read history data.

Each user story has been given its own script:

- Sensor data - SensorManager
- Trend – DetailedInfo
- History – History
- GraphChanger is a complementary script to History.

6.1.6.1 Class Diagram: Sensor Information

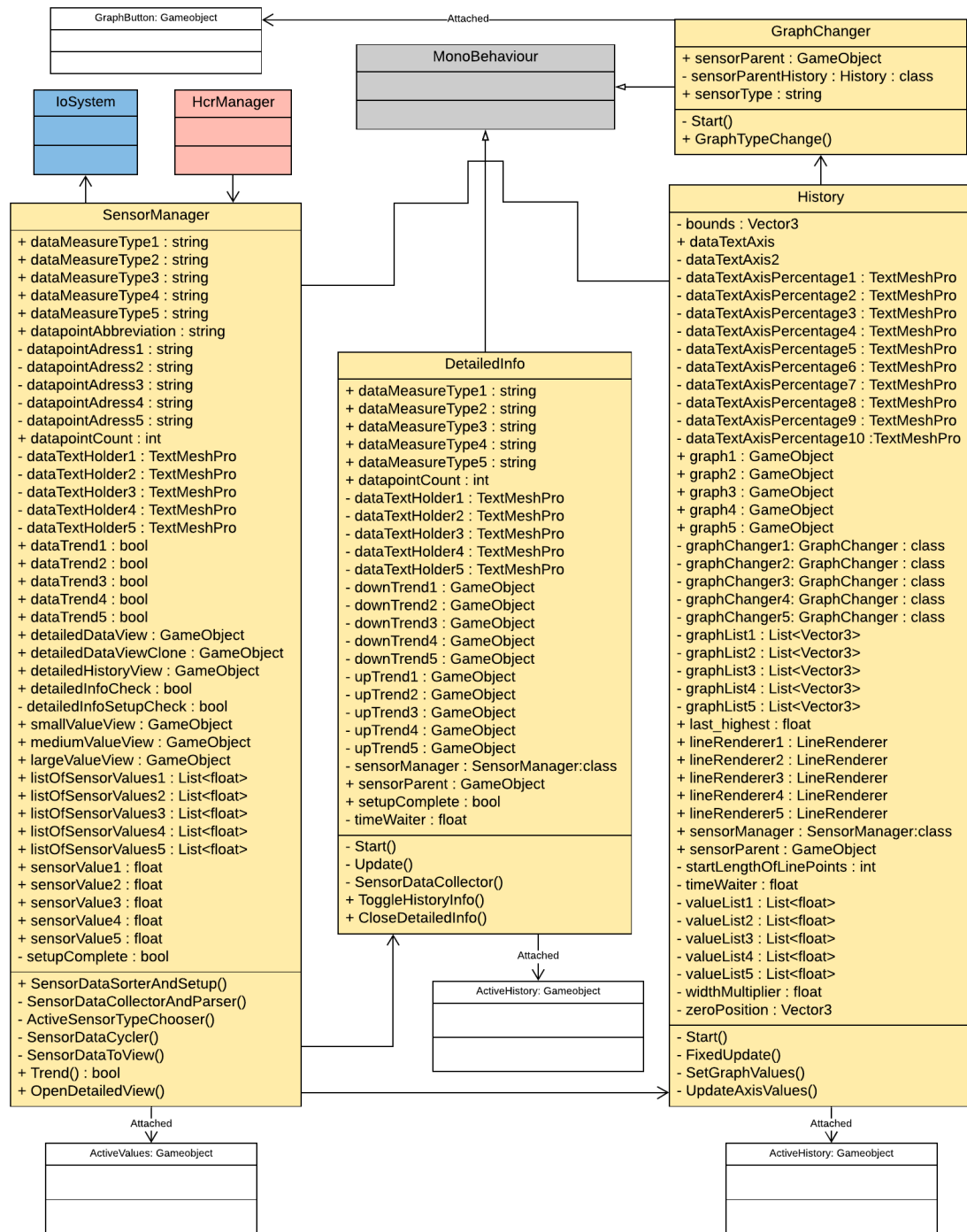


Figure 83 - Class Diagram: Sensor

6.1.6.2 Sequence Diagram: Sensor Information

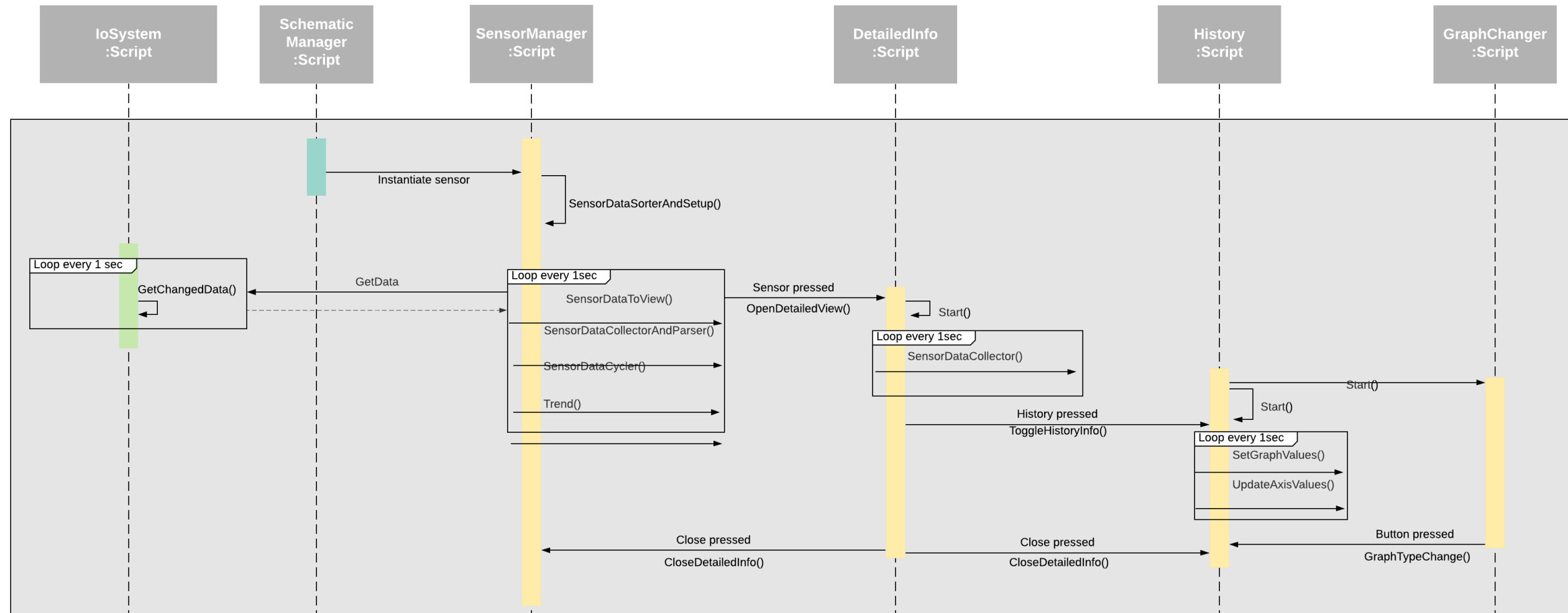


Figure 84 - Sequence Diagram: Sensor Data

6.1.6.3 Prefabs: Sensor Information

Sensor data is divided into three sizes, one, two and five. The small display sensors have one data address, medium has two, large has five.

Active Values Small

Figure 85 is the first of 3 views for sensors.

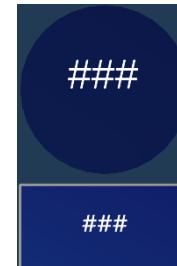


Figure 85 - Active Values Small

A detailed faceplate see Figure 86 for this sensor data can be accessed by a press from your finger, the information is now shown on a line and with an up/down arrow indicating if the values are currently rising or falling. In the top left and right corners of this figure there are two buttons, the left one opens history, see Figure 87, which is a graph displaying the last 200 seconds of data.



Figure 86 - Active Values Small Detailed

The y axis is capped at the current maximum value of the current sensor type and the x axis will always display time in seconds. The history button can be pressed again to toggle the history view. The close button will close the detailed view and the graph.

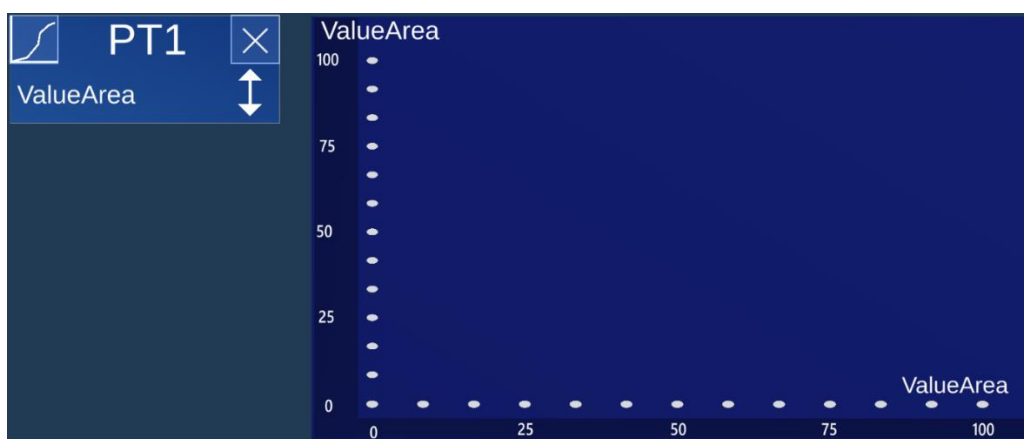


Figure 87 - Active Values Small Detailed History

Active Values medium

Are now displaying two values instead of one, a change graph function is therefore added on Figure 89, however the graph on Figure 90 will still be displaying one line of values. To access the values of the other sensor type, the user must press the area of “ValueArea”. That will result in a change in the graph, where the y axis also changes to reflect the sensor type change.

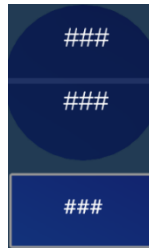


Figure 89 - Active Values Medium



Figure 88 - Active Values Medium Detailed

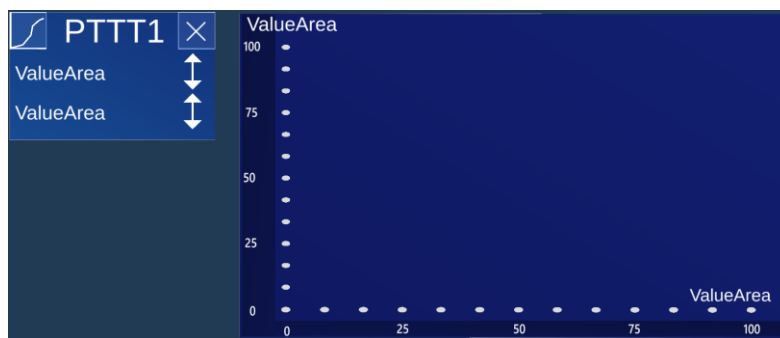


Figure 90 - Active Values Medium Detailed History

Active Values Large

Has no difference in functionality compared to the medium size, except for how many sensor types are displayed.

As seen on Figure 91, Figure 92 and Figure 93.

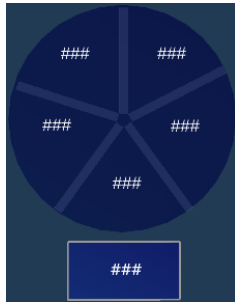


Figure 91 - Active Values Large



Figure 92 - Active Values Large Detailed

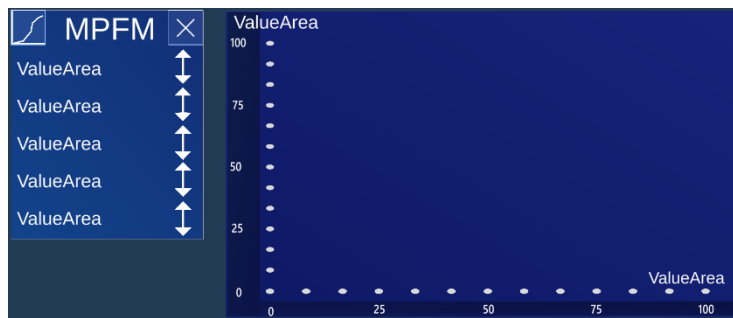


Figure 93 - Active Values Large Detailed History

6.1.6.4 Sequence Diagram: SensorManager

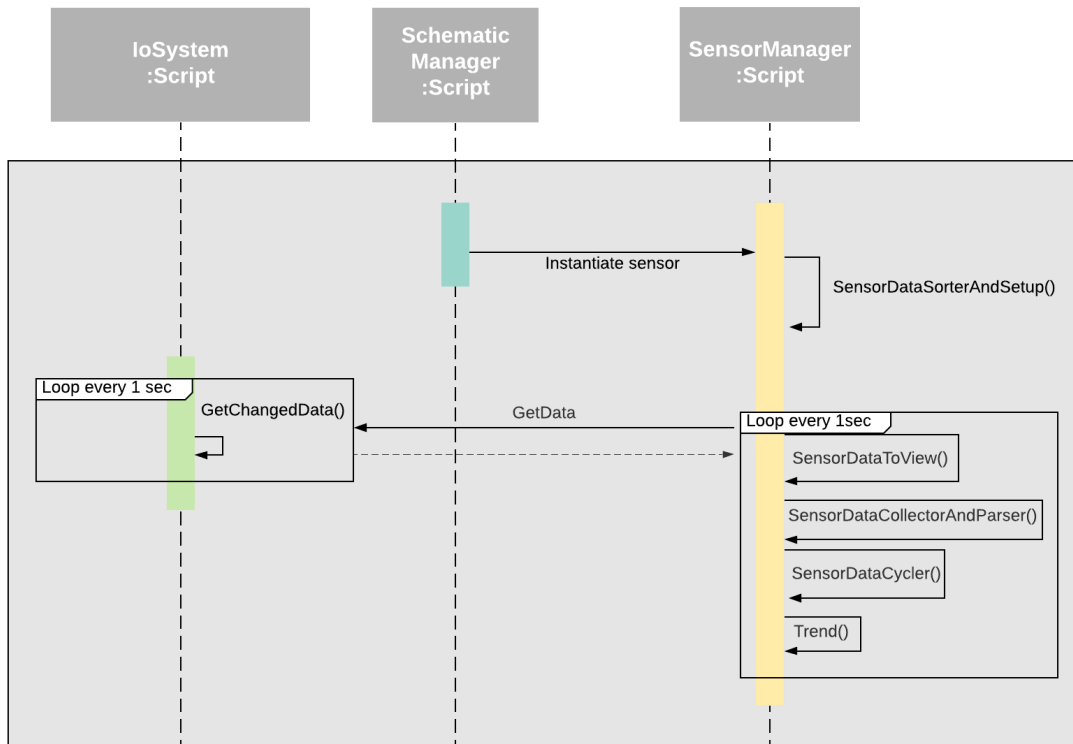


Figure 94 - Sequence Diagram: SensorManager

6.1.6.5 Sequence Diagram: DetailedInfo

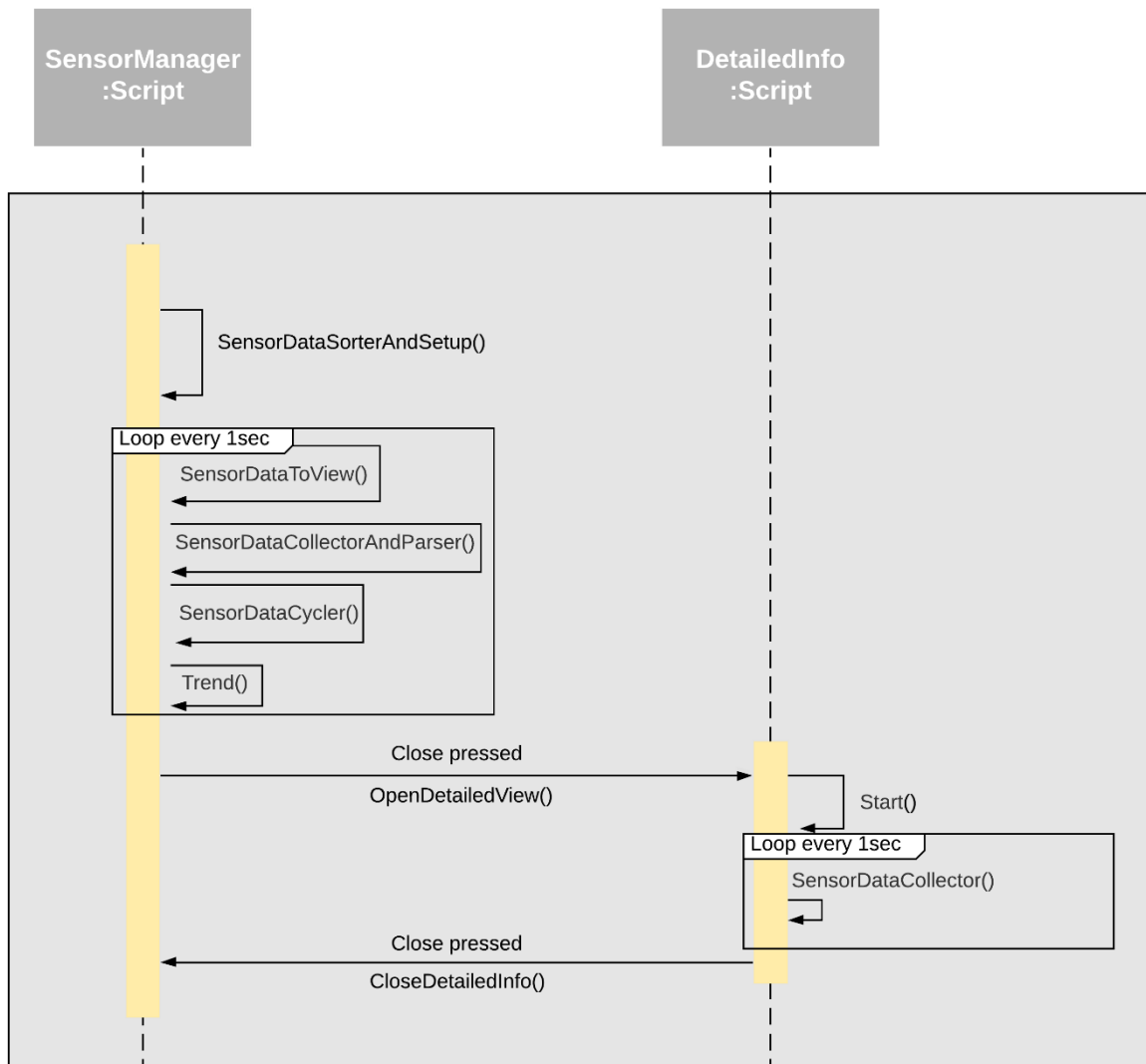


Figure 95 - Sequence Diagram: DetailedInfo

6.1.6.6 Sequence Diagram: History

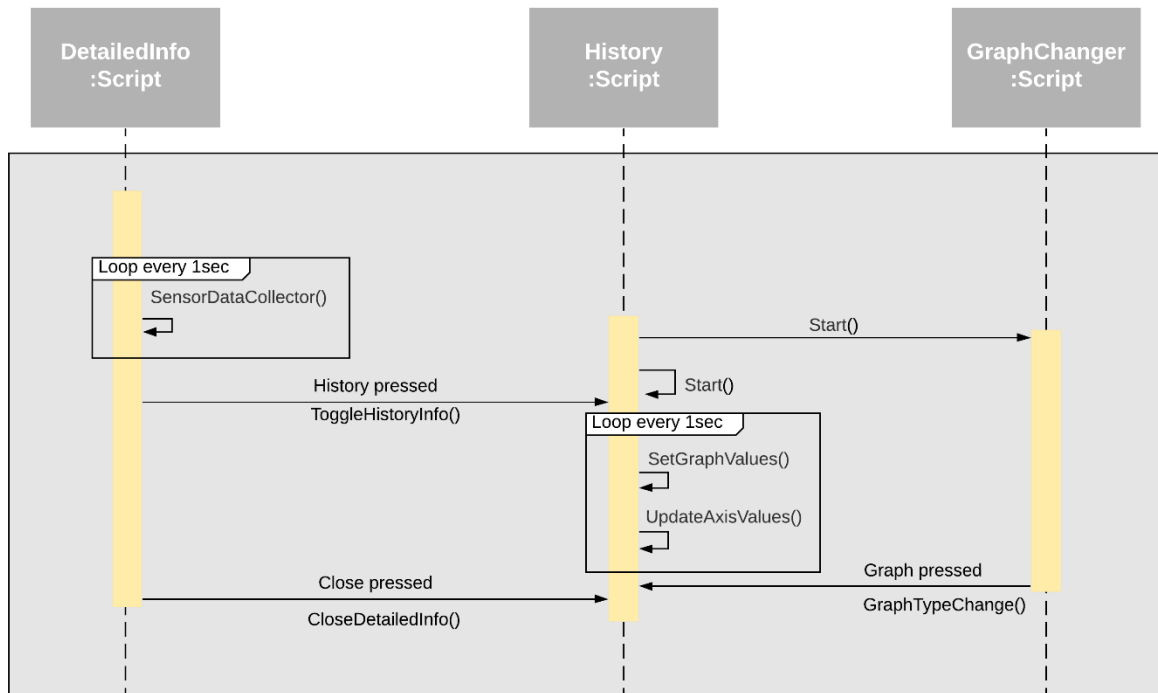


Figure 96 - Sequence Diagram: History

6.1.7 Hand Menu

The hand menu is used to navigate through the application and can be activated any time during user session.

The hand menu feature is implemented as a solution to a requirement given that the application should be more intuitive and user-friendly than the existing system. The hand menu takes advantage of MRTK script called Hand Constraint Palm Up which identifies when the hand is visible.

The hand menu is separated into two different hand menus. The one we see in Figure 97 is the hand menu that is available after the user has pressed the connect button inside of the XT menu. Figure 98 is a simpler hand menu that is available from the moment user launches the application until connect is pressed. This hand menu is a simple but important menu that consist of two options. The user can open the Read Me or the user may quit the application.

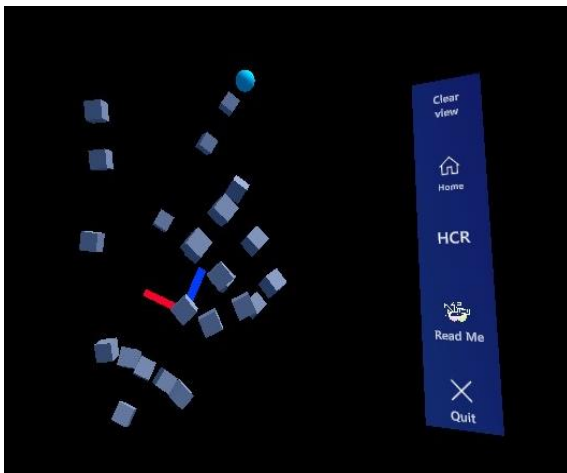


Figure 97 - Hand Menu seen after connecting to XT



Figure 98 - Hand Menu seen before connecting to XT

The Hand menu consists of five buttons:

- Clear view
- Home
- README
- Quit
- HCR

6.1.7.1 Class Diagram: Hand Menu

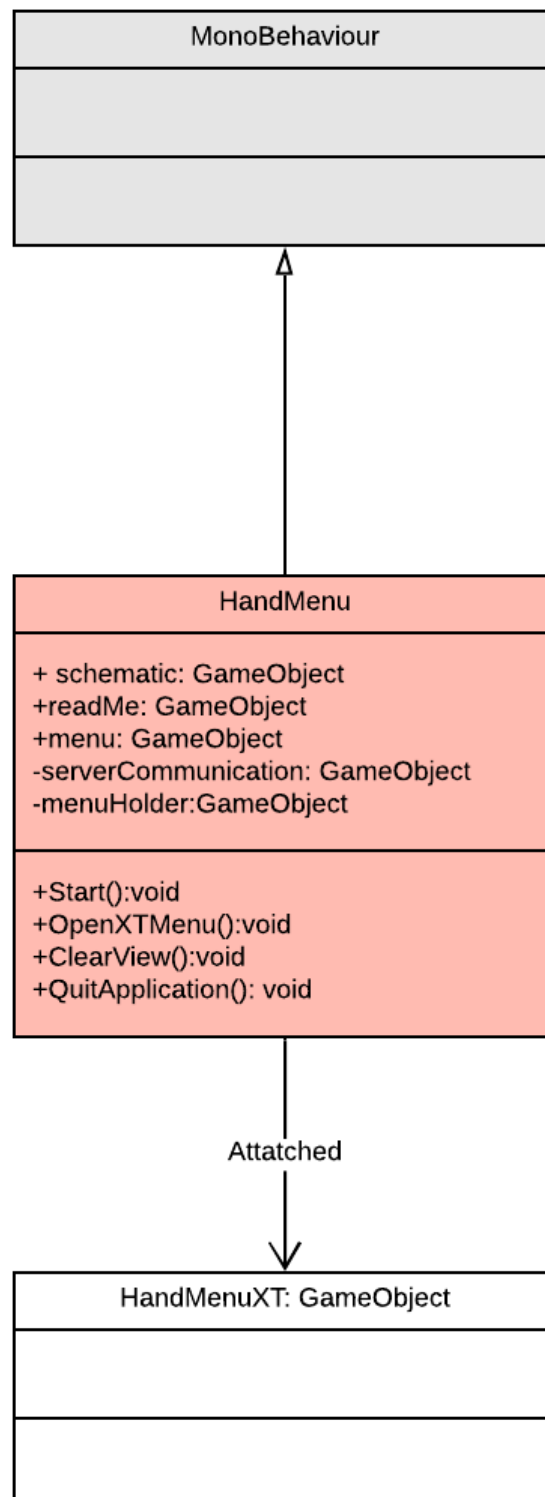


Figure 99- Class Diagram: Hand Menu

6.1.7.2 Clear view

The clear view button is a quick and easy way to unclutter the users view when having multiple sensor and valve information faceplates open at once. By pressing the button sensor and valve faceplates will be reset to their local position corresponding to parent object.

6.1.7.3 Sequence Diagram: ClearView

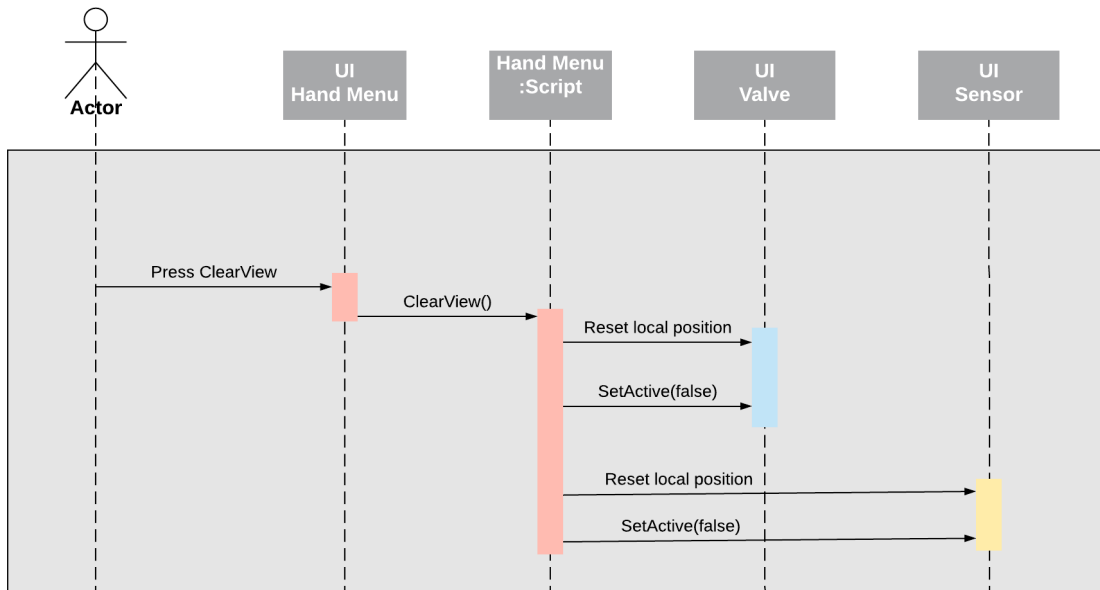


Figure 100 - Sequence Diagram: Clear View

6.1.7.4 Home

Home will take the user back to the Manage XT menu. This is where the user can connect, edit, remove, or add a new XT to the application. The button needs to search through the scene hierarchy for the Manage XT component and set this to true, while turning of the server communication and destroying the current schematic that is in view of the user.

6.1.7.5 Sequence Diagram: Home

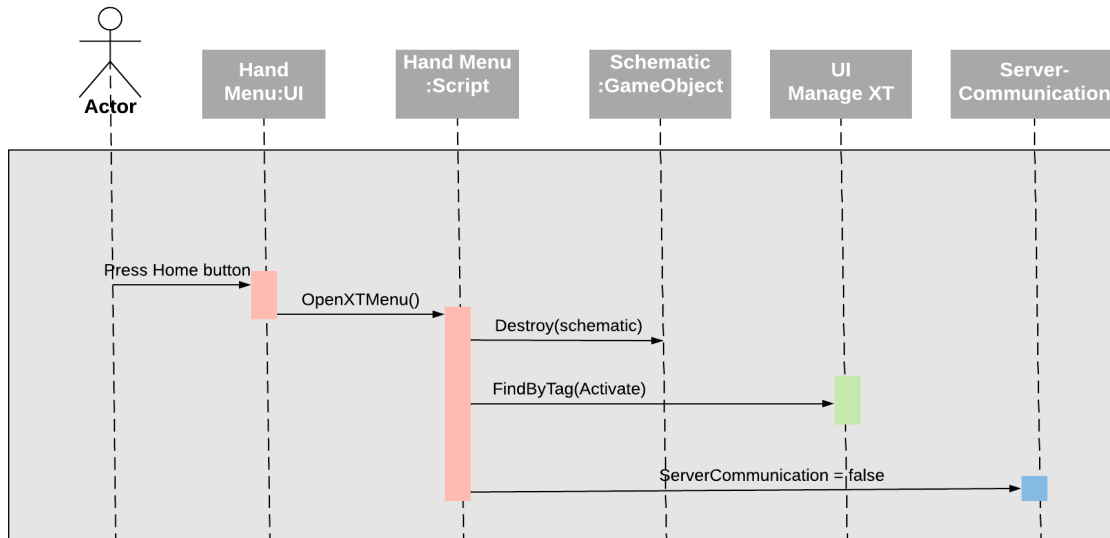


Figure 101 - Sequence Diagram: Home

6.1.7.6 Read Me

The read me button functions by toggling itself on and off. The read me includes a short manual on how the application is used. This calls a script called isToggled. The readme is also represented as an appendix. See APPENDIX F – APPLICATION USER GUIDE

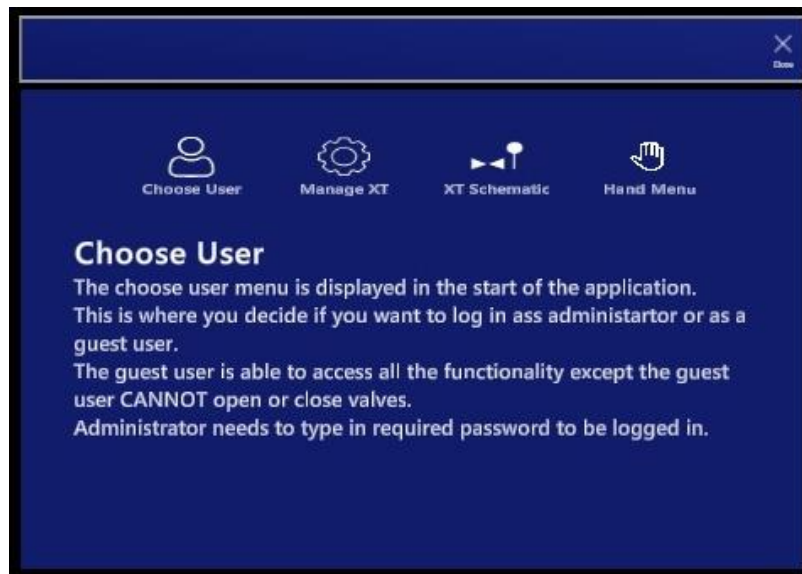


Figure 102 - ReadMe displayed

6.1.7.7 Quit application

Calls two functions that will quit the application when pressed. First is to quit the unity application. This is done by `Application.Quit()`. There is also a need to quit the Universal Windows Platform application. This is called by this:

```
Windows.ApplicationModel.Core.CoreApplication.Exit();
```

6.1.7.8 HCR

The HCR schematic is a solution made to display the P&ID in the HCR, see Figure 103. The sub system instantiates every component located in the P&ID. Including, but not limited to:

- Valves
- Sensors
- Chemical Injection Trees

The HCR Schematic section of the technical system is a solution made to satisfy the functional requirement of being able to display HCR information. The HCR is the component in the XT what controls the valves placed on the XT. The HCR is therefore a critical component to monitor. Valves placed on the HCR only have the ability to read status, not be operated.

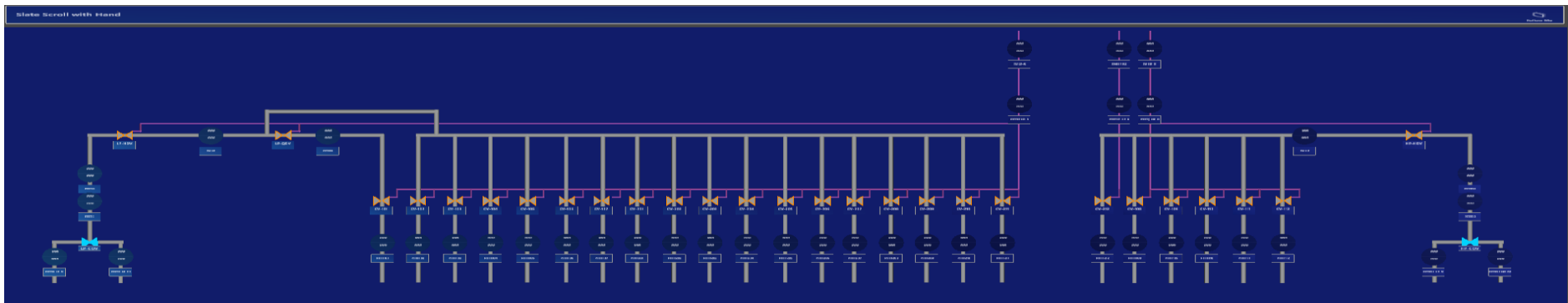


Figure 103 - Populated HCR Schematic

HCR Button

The HCR button will toggle the XT corresponding HCR view. The script on the button will check if the HCR is active and either set it active/not active depending if the view is visible.

6.1.7.9 Class Diagram: HCR Schematic

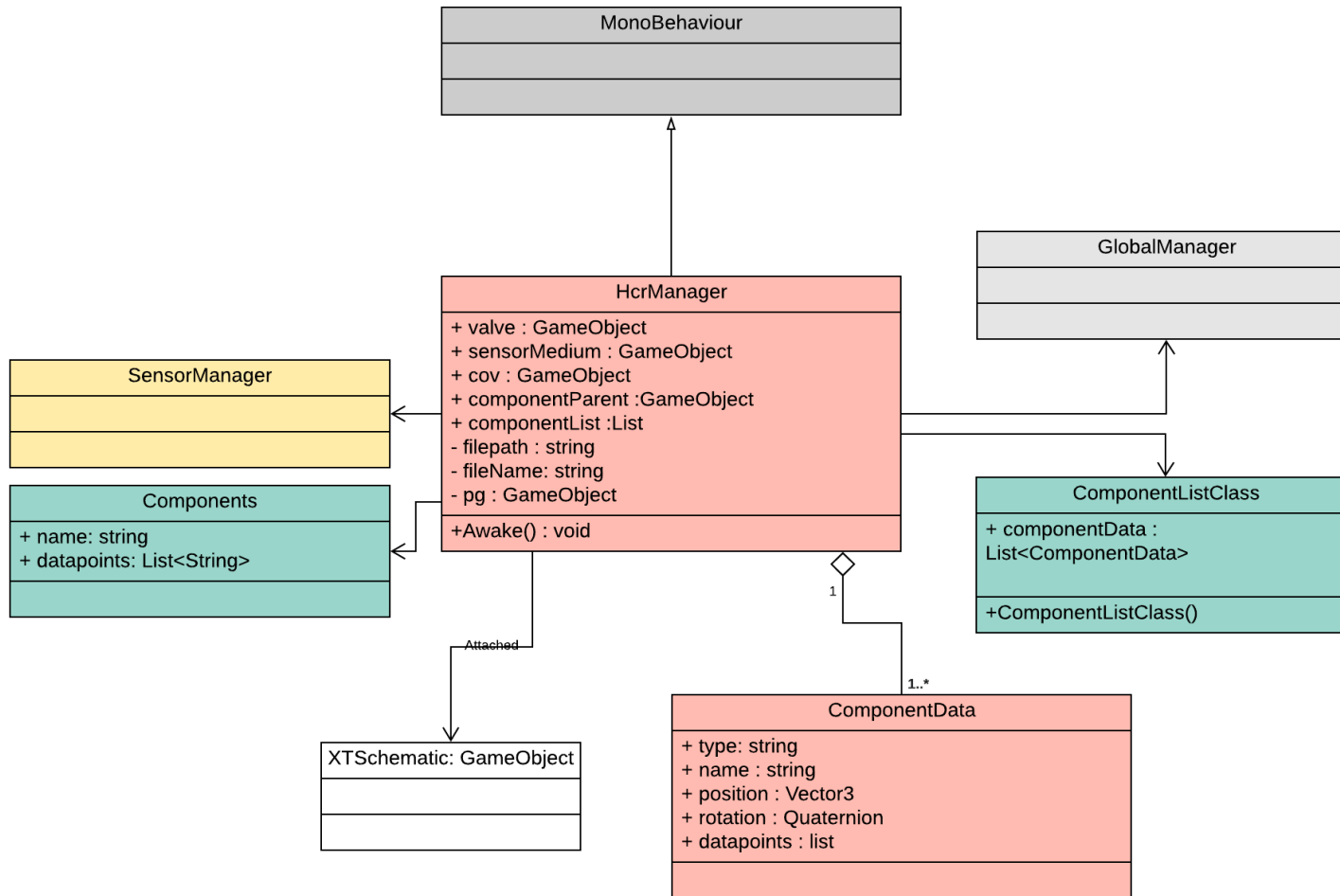


Figure 104 - Class Diagram: HCR Schematic

6.1.7.10 Piping

The HCR piping is based on the same principals as the XT Schematic, the main difference is that the HCR Schematic requires return lines. The return lines are pink.

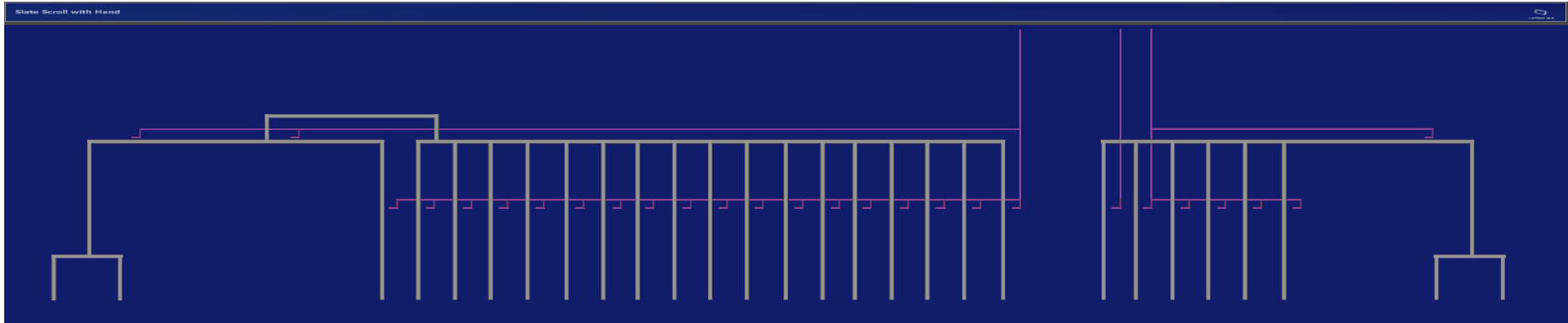


Figure 105 - Empty HCR Schematic

6.1.7.11 Populating the Schematic

The HCR and XT schematic are populated in the same way, to see more details XT Schematic.

6.1.7.12 Population of HCR Schematic – Sequence Diagram

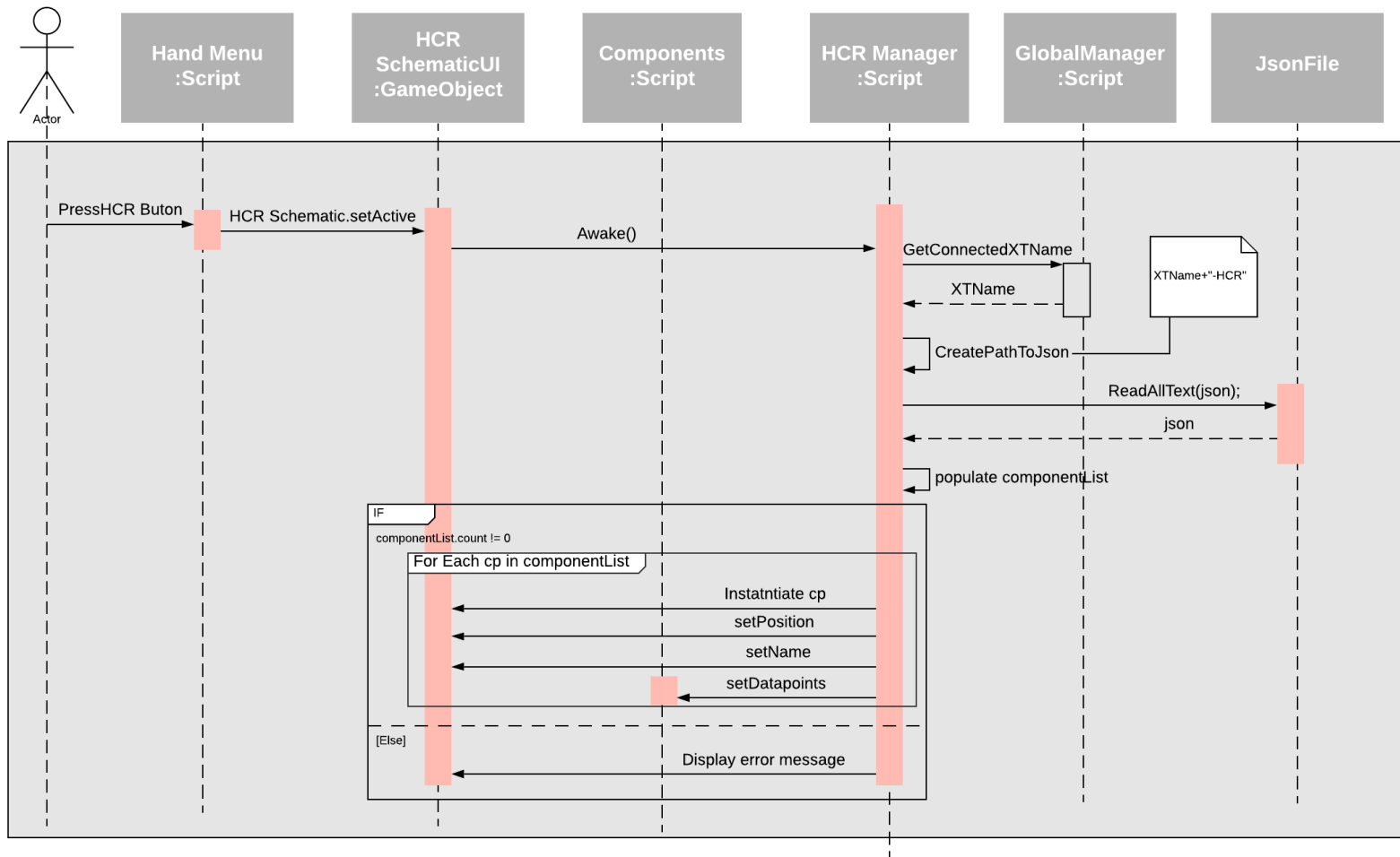


Figure 106 - Sequence Diagram: Population of HCR Schematic

6.2 User Interface

The user interacts with a system in an environment where visual design and effects have a great influence on the full user experience. It is therefore important to take advantage of the visual elements and effects that come with using HoloLens2 and develop a product where the visual effects help the user get a clear understanding of the system flow without much effort. The goal is to make a simple design that allows the user to navigate quickly through the different functionalities and to find the information needed easily.

6.2.1 Design

Dull screen was an important factor in the design decision-making due to the user work environment. Control rooms have a critical role in the oil industry and the user must not exhibit fragmented attention from extra features but focus on what is currently happening in the system. Having this in mind, the application has a clean user interface with few buttons and uses visual elements as indications to unclutter the user view.

6.2.1.1 Visual indicators

There are different ways to distinguish information. Colours, icons, forms and figures in addition to text is used in this application.

Colours

The colours used on the faceplates is dark blue and white text is used as a contrast and to make the text readable. The pipes (production, annulus, service) are grey as these do not need much attention, and the injection pipes are coloured neon green because this is also used by TechnipFMC in their schematics. It is the valve and sensor information that is most important. The standard valves have therefor been given the colour orange which gives a good contrast to the background colour, and the HCR valve uses the colour light blue to differentiate between the types of valve. In addition, two different colours were used to indicate whether the valve is open or closed, but this posed a challenge because red and green could not be

used. This meant that we had to come up with two different colours that provided the same intuitive meaning to the user.

The sensors are dark blue, which is not too conspicuous and complies to the dull screen aspect.

Icons

The various icons provide a quick understanding of the functionality that the icons represent and help the user navigate through the application. The icons are a mix of both standard and custom icons. In addition to text, the Choose User icons use the letter A or G to make it easier to identify the different users. The XT symbol represent an XT and the XT Schematic icon has a symbol of a valve and a sensor to represent what information the user can expect to view. In the Read Me, the icons also change form to bold when the tab is activated.

Figure

To indicate whether the valve is open or closed, filled vs outlined indication was used as showed in Figure 107. The uppermost proposal in the Figure 107 was implemented to further distinguish between the valve's shape and sensor's shape.



Figure 107 - Indications open/close valves

6.2.2 Interactable Elements

All faceplates have the same canvas and UI components attached to them so that the user will experience an overall similarity between the different faceplates when interacting with them. The faceplates are categorized into three different groups depending on their main functionality.

When designing the application, elements from MRTK were chosen to use. Microsoft has already done a lot of research on UI, and there was no reason to make our own buttons, sliders and faceplates from scratch when slates, materials and prefabs already existed. All used prefabs have been modified to fit for use of this application.

Game Objects used from MRTK:

- Interactable
- Buttons
- Title Bar Slate

Game Objects designed by AWC:

- Pipes
- Valves
- Sensors
- Graphs

Materials used from MRTK:

- Close icon
- Follow me icon
- Add icon
- Read Me icon
- Home icon
- Hand Menu icon

Materials designed by AWC:

- AWC Logo
- Delete Pin Icon
- User icons
- XT icon
- Trend icons
- XT Schematic icon

The title bar is used in two different contexts, depending on what view it's located in. For the Main views, the title bar gives the information about who the user is logged in as, what XT the user is connected to and if the connection is active or inactive. For the Medium views the title bar only holds the close button for that particular view.

6.2.2.1 Faceplates in the application

The different faceplates are categorized into three groups. The large ones consist of the three main views in the application. Medium consists of all additional information and tables that are required to be filled in or can be modified. Small covers the information about the valves and sensors.

Large (main):

- Choose User
- Manage XT
- XT Schematic

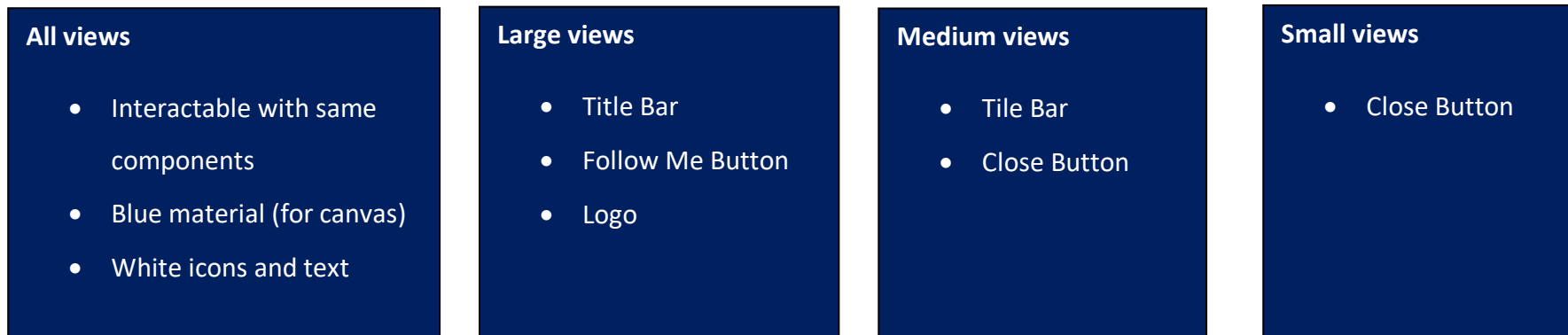
Medium:

- Read Me
- Pin Code Table
- Add XT
- Edit XT
- HCR

Small:

- Sensor Information
- Valve Information

Design/Functions given to the different views:



7 Results

In this chapter we will discuss the challenges and obstacles we encountered and how we worked to overcome those obstacles. We will also see if we met the requirements.

7.1 Have we met the requirements?

Non – functional Requirements

- Make an application using AR
- Use SW given by TechnipFMC (InControl, Simulator)
- Multiple user (multiple HL2)
- Want to see similarities between this application and TechnipFMC's existing application
- The application should be more or as user friendly as the existing system

Functional Requirements

- Secure valve operation
- Connect to different XT's, preconfigure only once
- Display XT information
- Display HCR information
- Read valve and sensor information
- Display sensor history

By implementing colour coded documentation, it is easy to verify that we have met all the requirements from TechnipFMC. The end product uses AR together with inControl and the simulator from TechnipFMC to give a user-friendly UI application that provides XT, HCR, sensor and valve information in a way that is recognizable for TechnipFMC. It also allows secure valve operation, multiple users and connection to multiple XTs with little pre-configuration.

7.1.1 Usability Test with Stakeholder

To further verify and validate that we had met the requirements and visually represented the client's wishes, a user test was conducted on Svein-Aage. The user test took place at the university towards the end of our implementation phase. Svein-Aage was asked to perform a series of tasks, explain his likes and dislikes with the application and answer a short survey. For more information about the process of the user test go to Test 1, 08.05.2020, Participant 1:

The UX design-related results from this test indicated small but important adjustments. Comments on the depth in the view, colours, and uniformity were noted and adjustments were made shortly after. When it came to the functionality and user-friendly specified results, he gave the score "Strongly Agree", on the question "I would like to use this app more frequently than the existing system". In addition, he finds the app to be very user-friendly.

7.2 Challenges

7.2.1 New Technology

HL2 was released November 7, 2019 – Only 2 months before our bachelor project started. None of us were familiar with HoloLens1 and had limited experience with Unity, which was the tool TechnipFMC urged us to use.

We received HL2 06.02.2020, a month into our project and immediately began to familiarize us with it.

Because of how new and exclusive this product is, there are no tutorials online other than the ones from the owner, Microsoft. We were among the first students in the world, and likely the first in Norway to develop an application for the HL2.

To deal with this challenge, our strategy in the first weeks was to develop a prototype to test the viability of the project and to further develop on. Two members of the group was set to develop this prototype. The rest of the group focused on detailed planning of the project, so that individual technical tasks could be allocated immediately following our second presentation. As a result of our strategy, one of the group members now had enough knowledge to help the others more effectively initiate the setup of their technical areas and to provide assistance throughout the implementation period. Our strategy proved to be both successful and effective.

7.2.2 Covid – 19 Virus

Due to the virus Covid – 19, strict restrictions by the government were implemented to stop the spread of the virus. This included closing universities and setting restrictions on how many people could be gathered in one room. The whole country was urged to practise social distancing from early March. This had a huge impact on our project working process and we had to adapt to the new regulations and start working from home. This introduced new challenges as we no longer could meet face to face and work together in our group room.

Close communication and following each other's progress were some of the main challenges. Another major challenge periodically was motivation and good work ethics. It is easy to be distracted when working from home and it is hard to define the line between work and home becomes blurred.

The new challenges included:

- Communication
- Following each other's progress
- Technical problems

The followed measures where made:

- Extended morning meetings – to improve communication
- Sharing screen through TEAMS – to follow each other's progress
- Remote control through TEAMS – to help each other with technical problems

We also distributed technical areas in such a way that it was possible to work independently. This allowed us to work on our technical part without relying on the components from the other team members.

7.2.3 GIT

The team members had little to no experience with using GIT and that proved to be problematic. The first GIT repo was set up with a gitignore that did not work properly. Secondly, it was altered and added to after each team member had branched out with the files that should not have been included. Alteration to the gitignore works for some files, but not all.

As this problem was not solved right away, the branches became six independent projects, which leads us to the next challenge, merging. The files that were included in the six branches contained static paths specific to the computers used.

Outside GIT expertise was sought and by making the GIT issues a priority, the root to some of the most problematic files was located. Some MSbuild files generated Nuget files with the conflicting paths. Unfortunately, this meant resolving at least thirty errors per branch, which proved to be a time consuming and inefficient procedure.

Instead we decided to share asset files and combine the split project. The prototype of the project was then uploaded to the new GIT repo with a much simpler and accurate gitignore. In addition, strict limitations on actions were established to reduce possible file conflicts. This eliminated the previous issues.

7.2.4 Only one HL2 - testing

HL2 is an exclusive product that we were fortunate enough to borrow from TechnipFMC for the duration of the project, not only in periods. Initially, having only one HL2 was not a problem, but after the Covid – 19 outbreak it became an issue as only the person with the HL2 had the opportunity to test the implemented application. This limited our technical testing functionalities. With the new rules and regulations regarding Covid-19, sharing the HL2 to test became more problematic. HL2 were not to be worn by different people due to possible transmission of the virus.

This was partly solved by how we planned our project and the fact that we had only just began the implementation phase. This resulted in a testing period when the restrictions were less strict, and the university opened for groups with special needs to use distributed group rooms. The only function that needed to be tested early was the one regarding the user interface. This included all the elements that a user should interact with. It was important to make a usable UI early so this could be implemented to all the different views.

7.2.5 Building in UNITY to HoloLens2

Throughout the project there have been issues when developing and deploying to the HL2. Prototyping using holographic remoting or play mode in Unity is effective, but the transition to a fully deployed build is not always straightforward. As we experienced quite a lot, functionality in Unity play mode did not work correctly after building and deploying the application. Due to lack of information and inexperience it has been a struggle to find the correct settings and the correct way to build. To get a fully working build you need to have correct VS2019 build settings, Unity settings and MRTK profile settings. Building and deploying is also time consuming, it takes around 10 minutes to build and deploy a fresh application to HL2.

A “How to guide” was made to make this process simple to follow, and it is available in **Feil!**
Fant ikke referanseilden.

7.2.6 Building and exporting models from Blender to Unity

The schematic had to be made in 3D to give a sense of 360-degree immersion. The decisions were made to make this static and from a 3D file (in this case an fbx. file).

Microsoft's guide when developing and designing objects to HL2 says that performance drops after 100 000 polygons. This means that there ought to be a stable 60-fps at all times. It is not required that the application should run at 60-fps, but to keep things as optimal as possible, a decision was made to make the static schematic model as a low polygon model. If the application were to be scaled up and included large and well-made 3D models of for example, a real XT or other large models that include millions of polygons (before optimization), then the limitations of the HL2 would start to show. From Figure 113 we can see that at around 2 million polygons the HL2 will perform at around 25-fps or lower [39]

Viewing, Moving Hand Mesh and Object Interaction

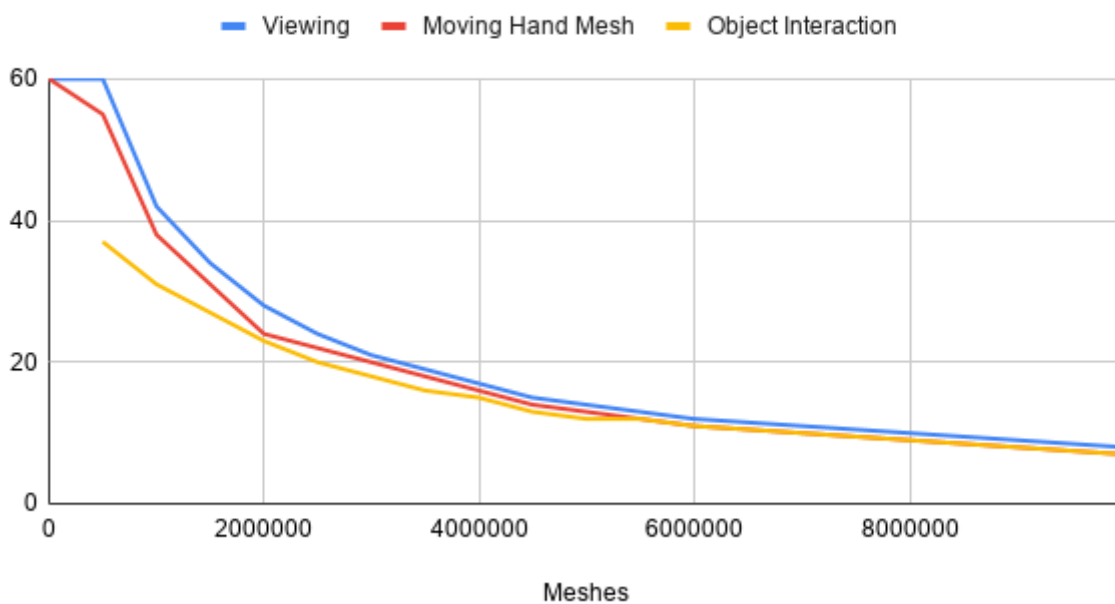


Figure 108 - showing table on how FPS drops when polygons increase [39]

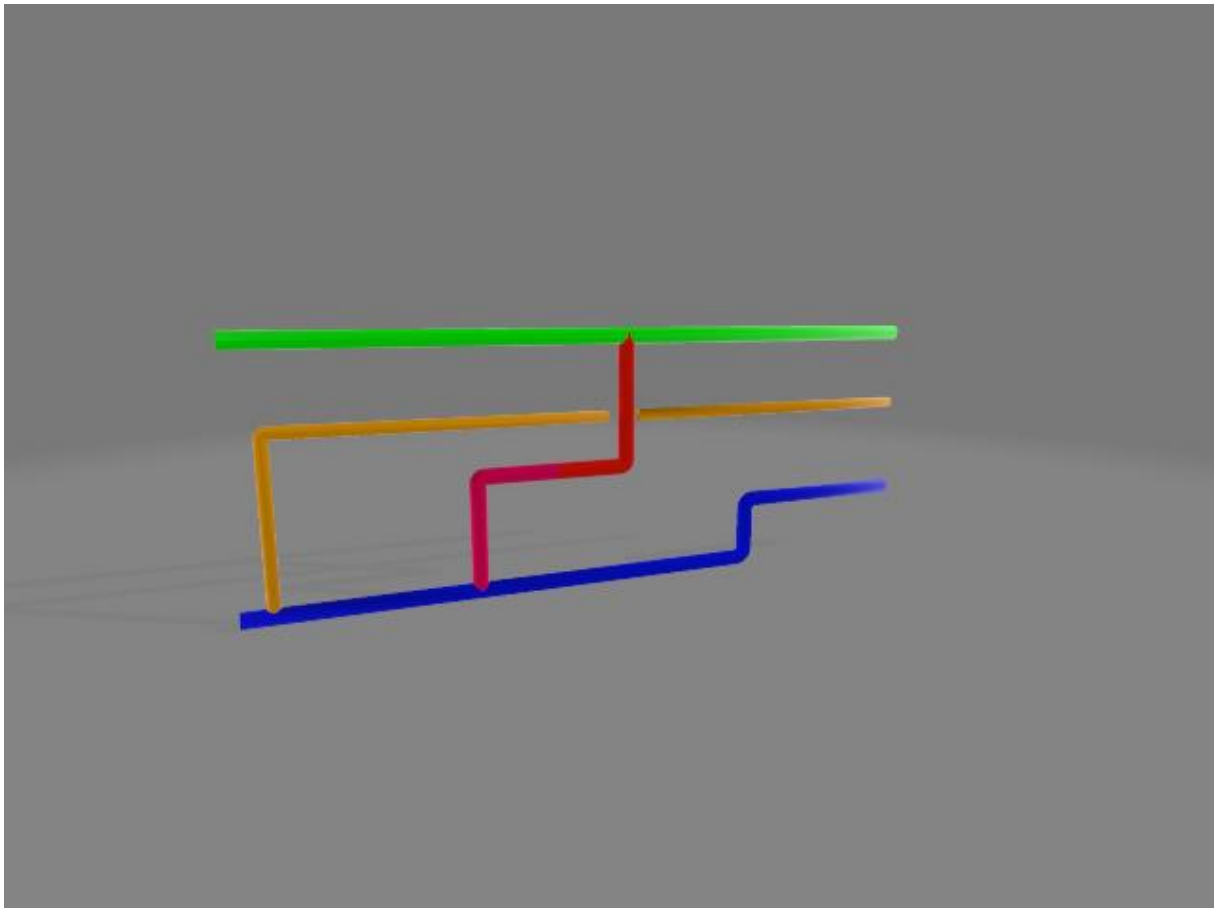


Figure 109 - schematic piping. Colours are for separating in Blender

A smart solution in Unity is making it difficult to render the front and back of the models. Because of this, it is necessary to add thickness to the figure to be able to display the whole figure. Thickness was added by using the thickness modifier in Blender.

Shading is added to give the object a cleaner look when light reflects off the figure. What is also important before exporting is to set the figure to a 1-1 scale in alignments. By doing this it will be easier to set the scale in Unity using numbers.

While exporting the model there were some issues regarding the difference in coordinate systems. In Unity x, y and z is different from 3D Blender's. Although x – axis is the same in both, y – coordinate in Unity is z – coordinate in Blender. This was an issue when exporting figures where models were out of position.

Making the Valve and Choke

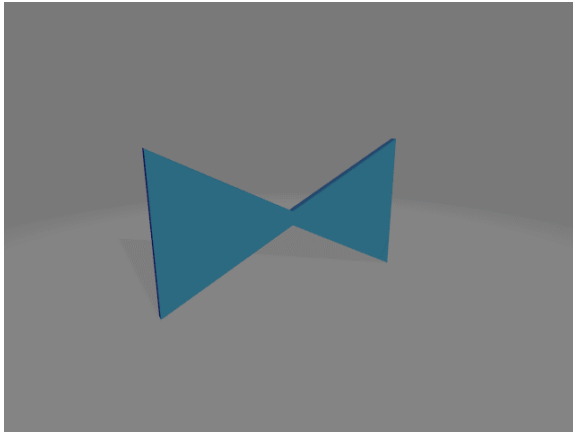


Figure 110 - showing model of finished valve

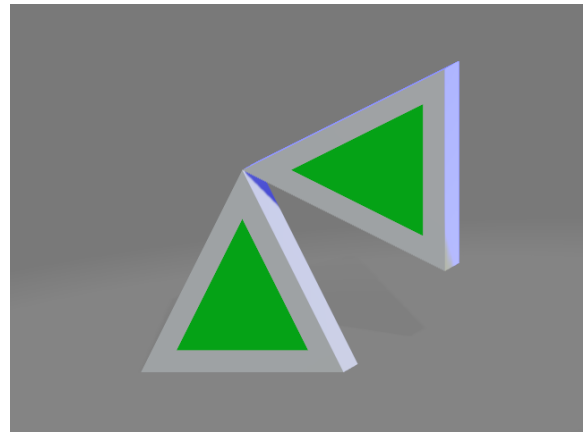


Figure 111 - Choke design

Making the valve 3D figures was done in Blender 3D as well. The valves were made using a cube then centring vertices on one side together to make a triangle. A mirror modifier aided symmetry. The thickness of the valve was set to be slim. This was to minimize the rendering strain for a scenario with many valves. The colours used in the figure are not the same colours used in the application. These are just colours used in Blender for rendering an example image. The valve is also given an outline Figure 116 this outline is used for valve status.

8 Conclusion

Discusses how the project went, challenges, our handling of issues and what future improvements can be made.

The requirements given to us by TechnipFMC were no small task: Make an application using a technology that is not officially released in Norway with minimal to no documentation and to do so in an industry that is brand new to the whole team.

The oil industry is highly competitive, and new strategies are tested frequently to land customers. Our product should not only be more efficient, safer, and more intuitive than conventional methods, but it should also have a notable level of character that impresses their clients.

We quickly decided that to create an outstanding product we needed to separate the application into different sections. One group focused on the UX design and functionality, while another group worked on making the “engine” of the system work. This led to some group members gaining expertise in how all the sub systems work together, as well as how the communication between our application and TechnipFMC’s systems interact.

This is probably our most important decision during the project. Because of this decision we had team members ready to assist other members with their respective technical task and its connection to the rest of the system. This resulted in faster and more precise error handling and gave us more time to add more features.

As to the aesthetical part of the project, without a group solely focusing on the aesthetical design and ease of manoeuvrability throughout the application, the application would not have been as intuitive and aesthetically pleasing.

One of our greatest achievements was how the group handled the Covid-19 pandemic. On Thursday March 12, we were to meet TechnipFMC and the school to present our work up to that point. On Wednesday 12, at 9.30 pm we received message that the presentation could not be held due to national lockdown and major restrictions. This lead us on a journey throughout the night recording, editing, and publishing the presentation so that we could present it digitally the next day. The determination and commitment by the group is what defines us and enables us to complete the tasks we set out to do.

8.1 Future Work

During the project period we have thought of and evaluated different features that could be implemented in the application to further improve the functionality. Due to time limitations these features were not implemented.

8.1.1 True multi-user

In our application the multiuser part of the system is handled by the simulator. With two HL2 connected and running the same application they, both receive the same data from the simulator. A true multi-user system would allow the users have a shared Mixed Reality space in run time either by sharing location data or using a server like Photon. This would allow collaboration on the same interface. [40]

8.1.2 Implement an observer pattern for the Software communication.

An improvement to the current system is to implement an observer pattern design on the sensors side of the IoSystem. Now every sensor is doing a polling loop every second. This is a waste, if the datapoint value are not updated from the simulator side within the 1 second timeframe. In our current application there are very few occurrences of this but implementing an observer pattern would still be an improvement. A system with callbacks that notifies sensors only when their values updates will get a better resource efficient system.

8.1.3 Implement more MRTK features

To take full advantage of all the capabilities of the HL2 we recommend implementing more features supplied by the MRTK toolkit, like eye gaze and voice commands. Iris scanning can also be added as a replacement for Pin code.

8.1.4 Expand application to include PCR-X and D

The ultimate goal for this application would be to control the whole process system from well to topside production. This is a daunting task when existing systems can contain up to 70 000 data points. The simulator updates around 8500 data points, and our application uses no more than 500 data points.

8.1.5 Oil Flow

In our application the valves changes colour depending on if the valve is opened or closed. Instead we propose to animate an oil flow showing where the valves are opened or closed to make the application even more intuitive to use.

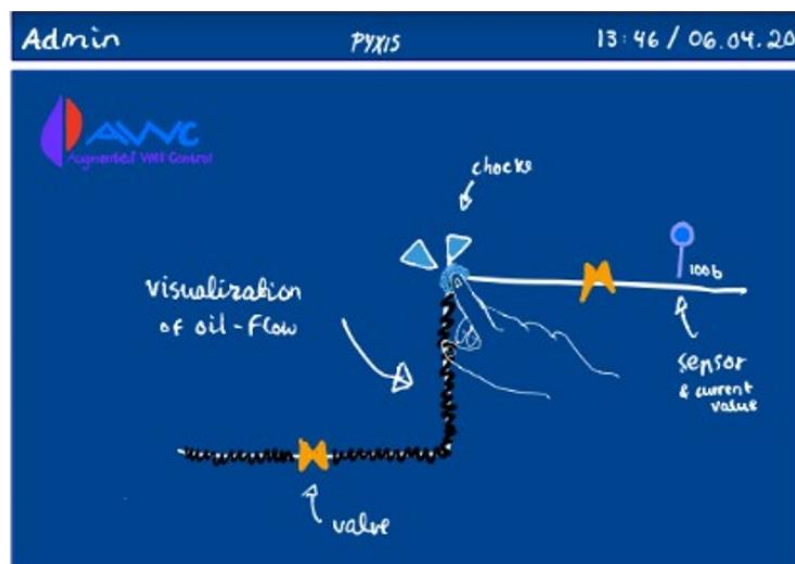


Figure 112 - Visualization of oil flow

8.1.6 Indication if values is critical

An important feature that could be implemented is an indication if the sensor or valve values is in a critical level. Per now, there is nothing in the application saying if for example the current pressure value is normal or unnormal/critical low or critical high.

8.1.7 Drawing program for json configuration

Instead of configuring the json files manually for every XT we suggest making a drawing application where you can drag and drop different valves and sensors, autogenerating a json file with the right positions and data addresses.



9 APPENDIX A - DESIGN DOCUMENTATION


Abstract

This document includes our design concept and how we plan to ensure to meet the customer's needs and desires. Also shows different implementation of different 3D models and thought behind decisions made.

9.1 Persona

Persona is a representation of a specific user which help standardize needs and get solutions faster. It is not generic, and it is a way to identify common issues users have to ensure empathy between the potential user and the designer. [41]

Augmented Well Control – Persona



Name
Svein – Augmented

Occupation
Master's degree in computer science. Currently working as a test engineer within the field of subsea technology.

Demographic

- 28 years old
- Lives in Kongsberg, Norway
- Father to three girls
- Married

Goals and needs

Svein is a busy man with a lot of deadlines and therefore needs to work efficient. His goal for each day is to do as many tests as possible and document them. After a day at work his attention is drawn to his wife and three children. When the girls are in bed he sometimes answers emails and other administrative work he can do from home.

Pain Points

Svein feels he is always in a hurry and doesn't always get through with all the tests he needs to do throughout the day. The process is slow and in order to complete one test there are too many screens and papers he needs to look at. At the same time, the things he tests are heavy and he doesn't always feel safe even though they have good safety routines. Also, he needs to walk around a lot as nothing is gathered in one place.

Personality

Interested in new technology as it makes new and more efficient systems. Takes inspiration in future movies and games.

9.2 UX Design

To map the different functions and views needed in our application, a whiteboard and sticky notes were used.

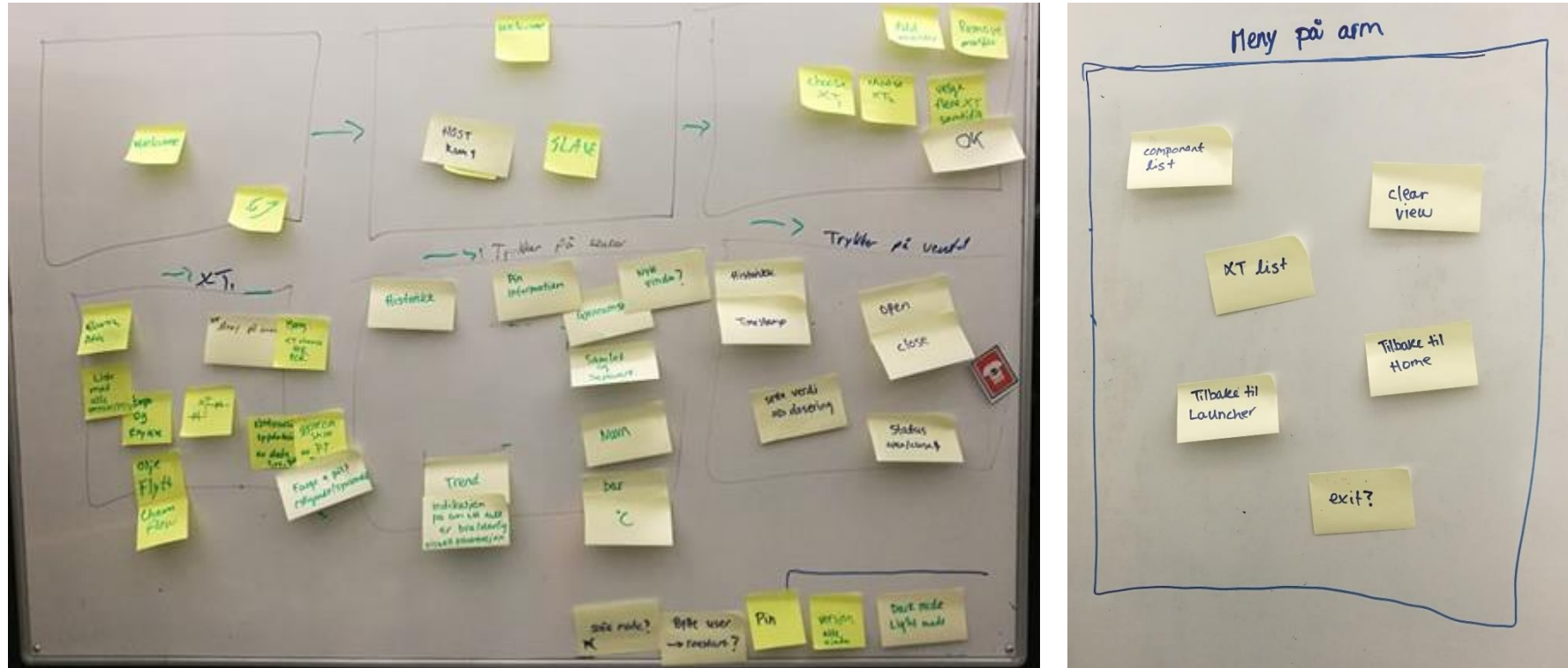


Figure 1 – Picture of functions and views needed

9.3 UI Design Suggestions

Different suggestions of UX design where made from different members. By pretending seeing the application for the first time and having the HoloLens 2 on, group members though of expectations of UX design and the flow of the application they had in mind when coming up with suggestions. Both usability and user experience were taking into consideration, but the main focus was on the flow and what “views” the user would see, not for example where the buttons is located.

Three of the suggestions of UX design for edit XT is shown below:

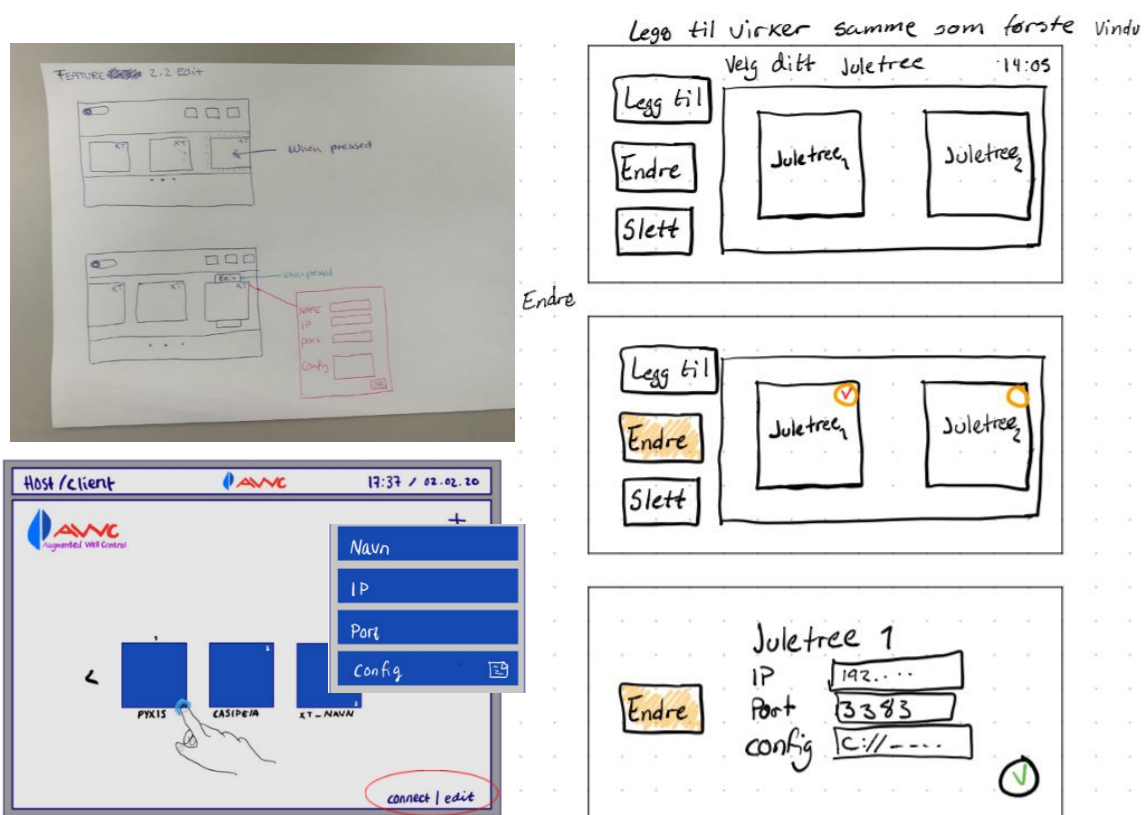


Figure 2 – Three different suggestions of the edit-xt UX design.

9.4 Final Sketch for UI Design

1. Welcome page → bring up the icon in HE2



starting up
icon

2. choose between admin and guest. Admin mode requires a PIN code. Once user is selected → remain throughout the session.



Figure 113 – Sketch of choose user

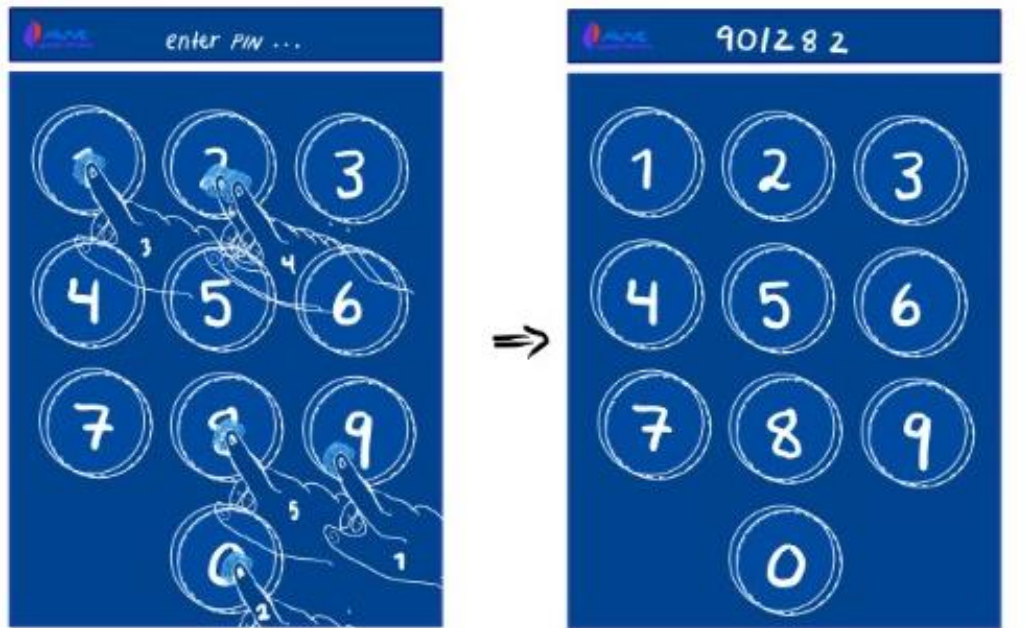


↑ chooses admin in this example



↑ new table is opened: user enters the PIN

Figure 114 - Sketch of Pin Code is typed in



chosen user
↓



Figure 115 - Sketch of correct Pin is typed in and add XT button is pressed in

3 Fill out XT form and connect to chosen XT.

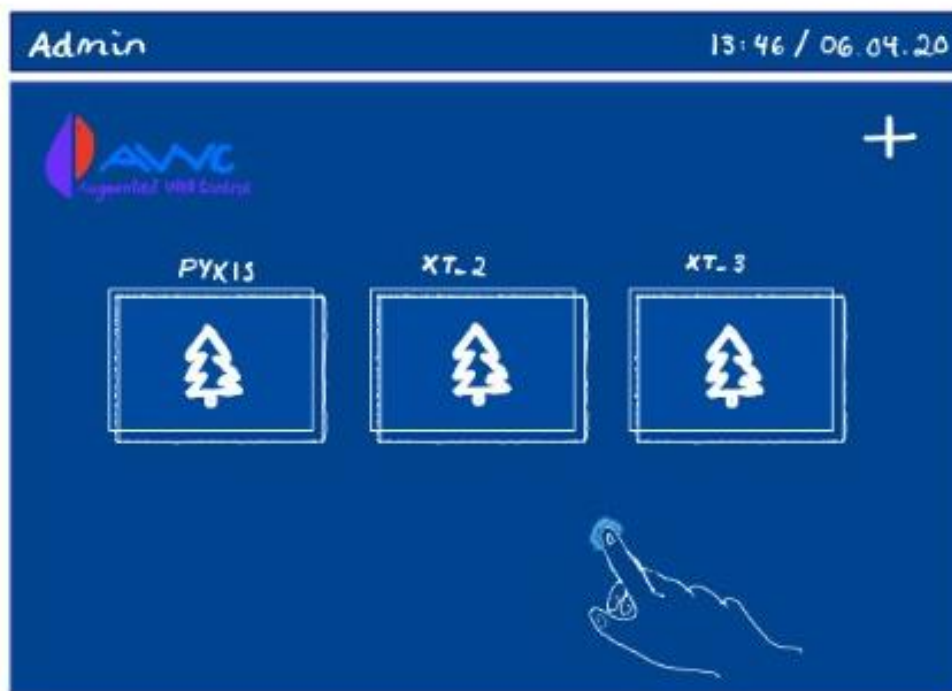
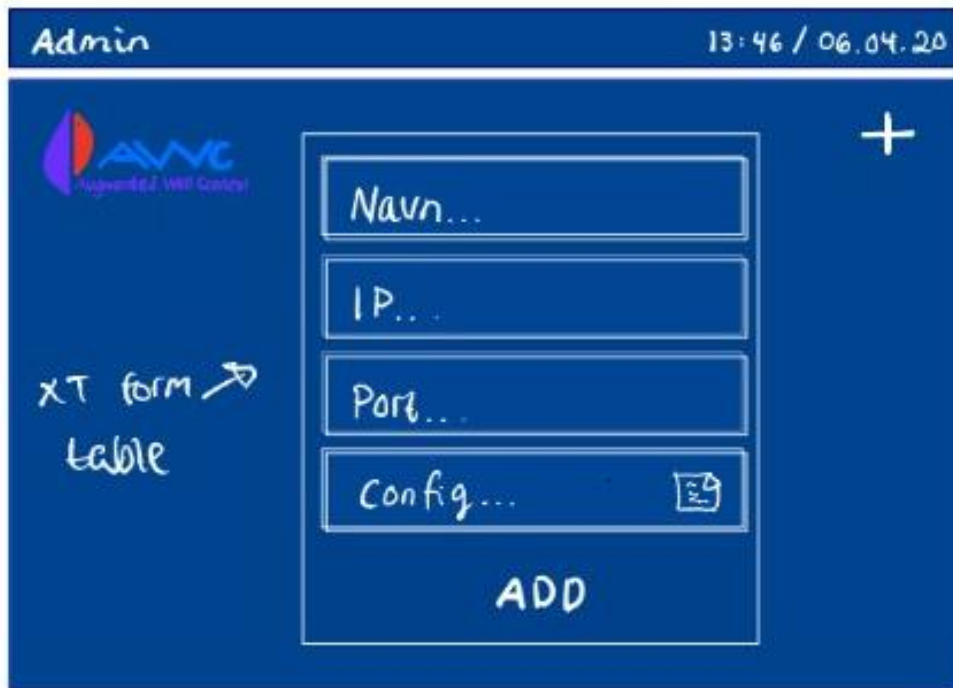


Figure 116 - Sketch of form is filled in and XT is added

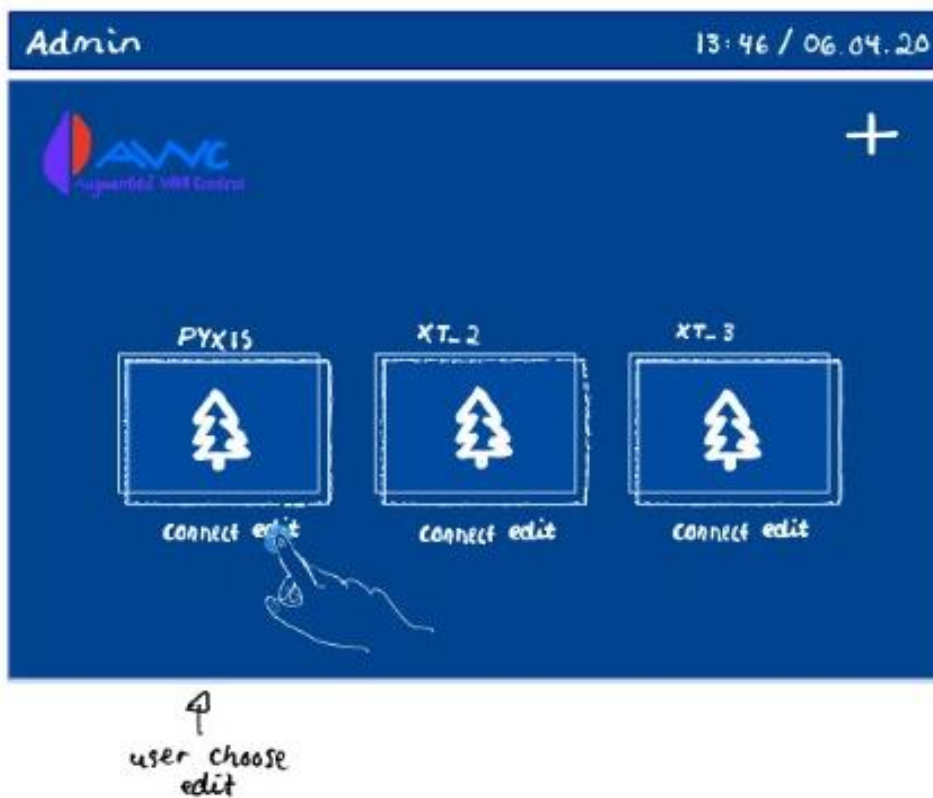
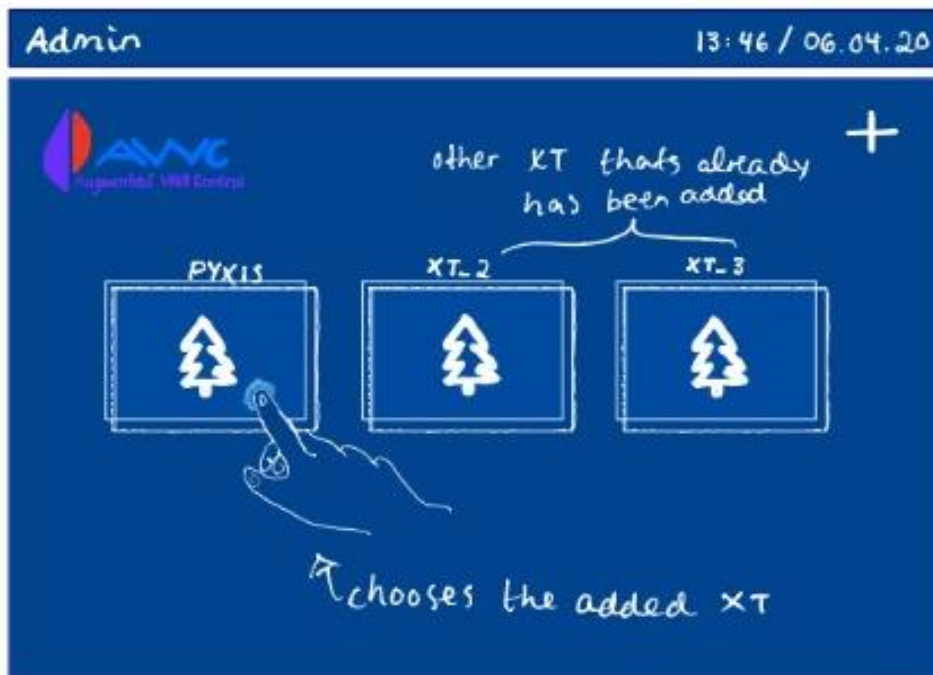
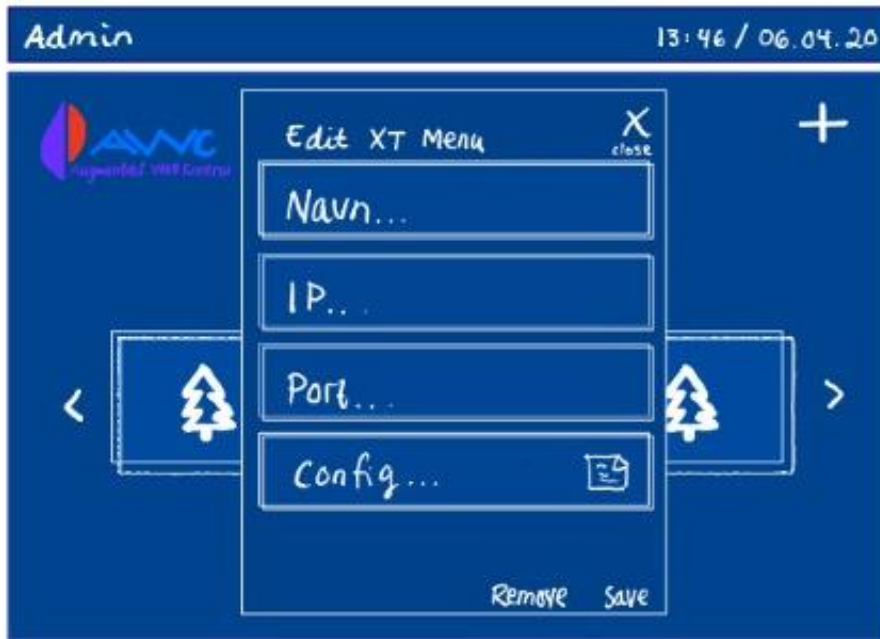


Figure 117 - Sketch of edit button is pressed



↑ edit form
⇒ edit or remove
XT from list

Figure 118 - Sketch of the edit form



Figure 119 - Sketch of the connect button is pressed

4. auto generated schematic of choosen and connected XT



name of connected XT

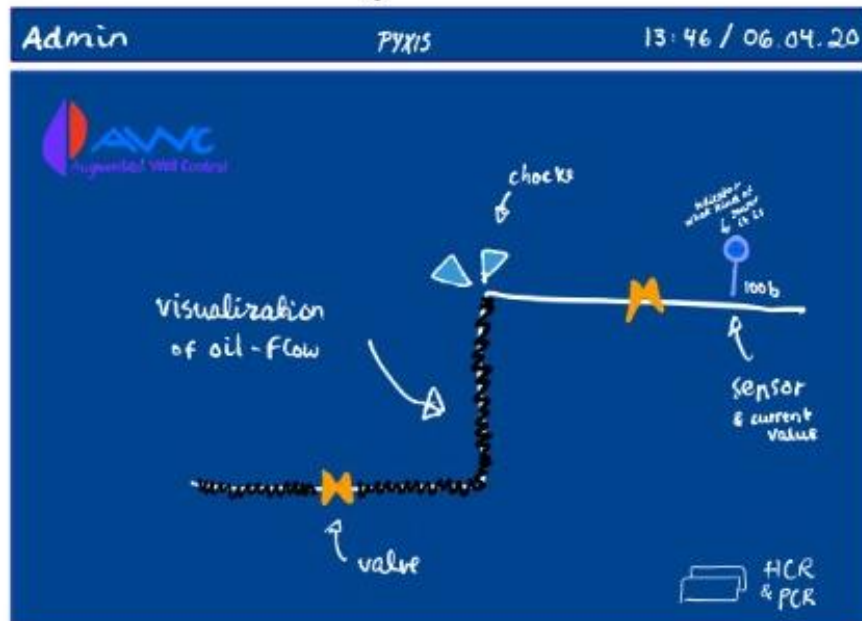


Figure 120 - Sketch of the XT Schematic

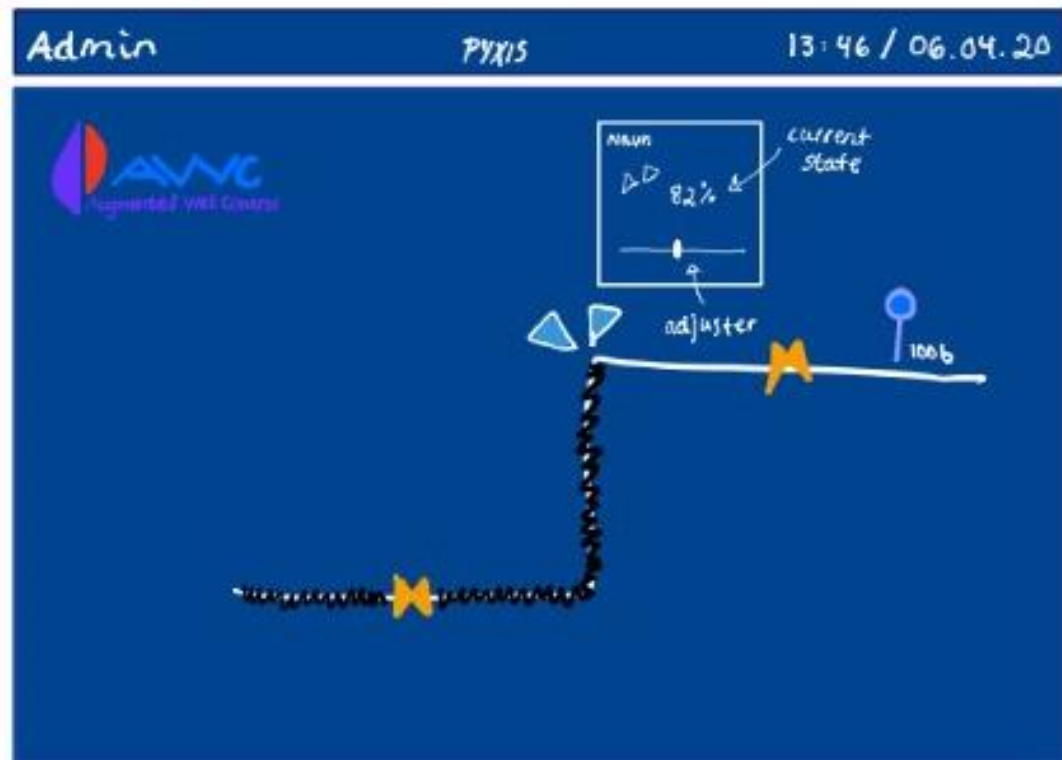
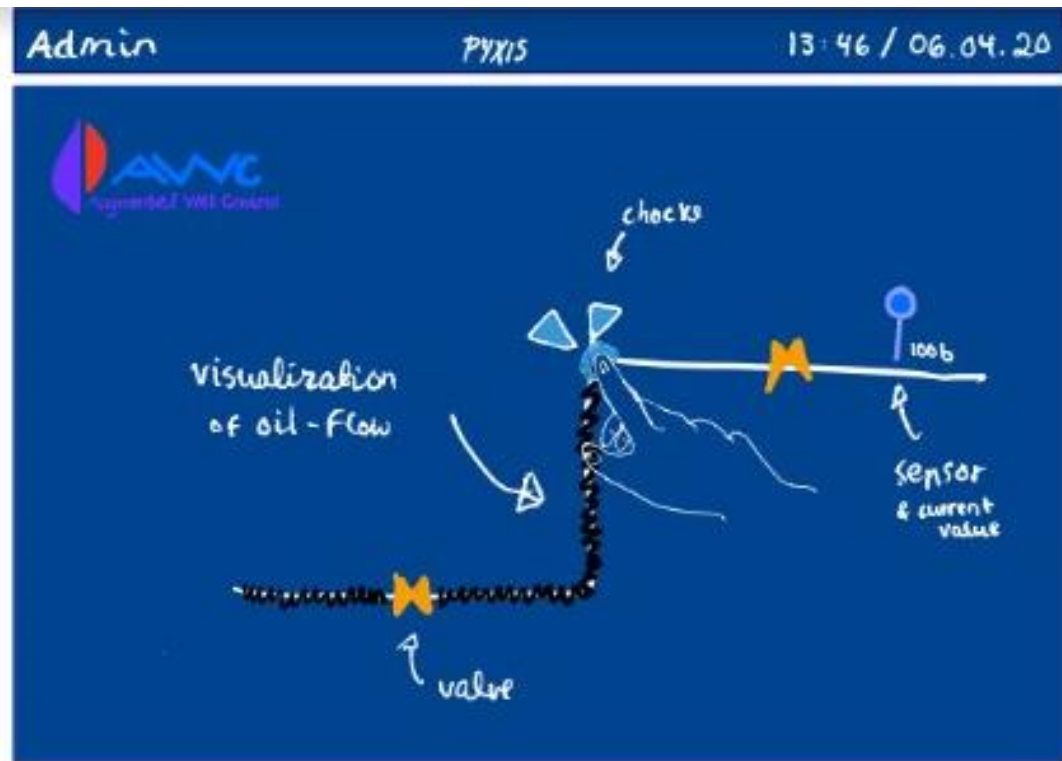


Figure 121 - Sketch of choke is being pressed

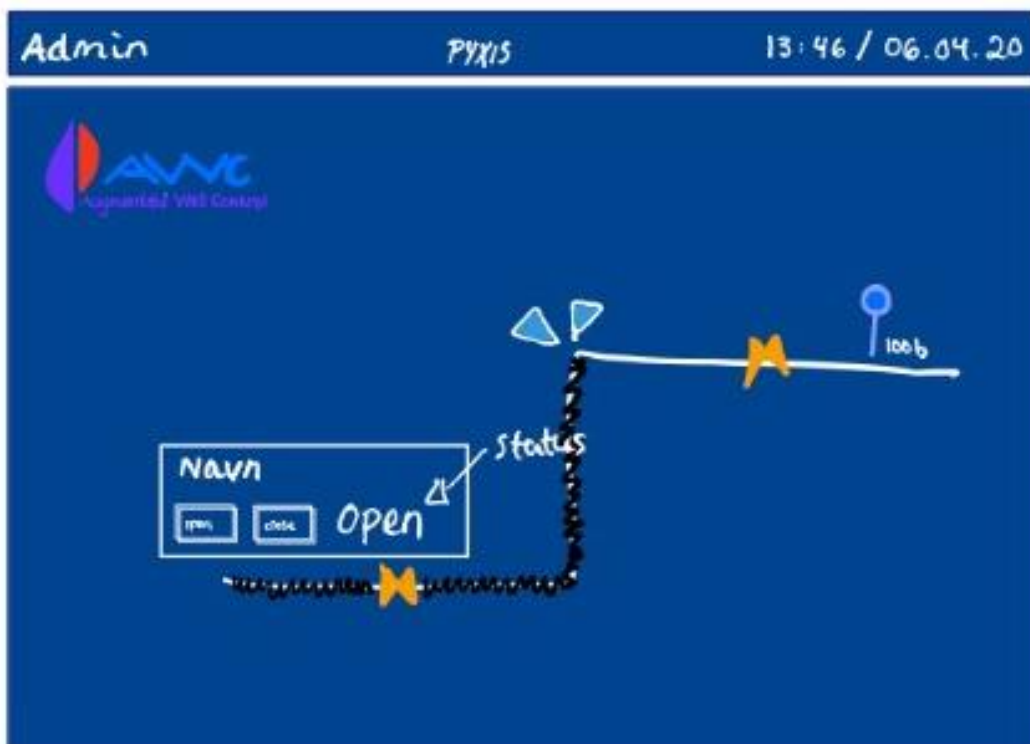
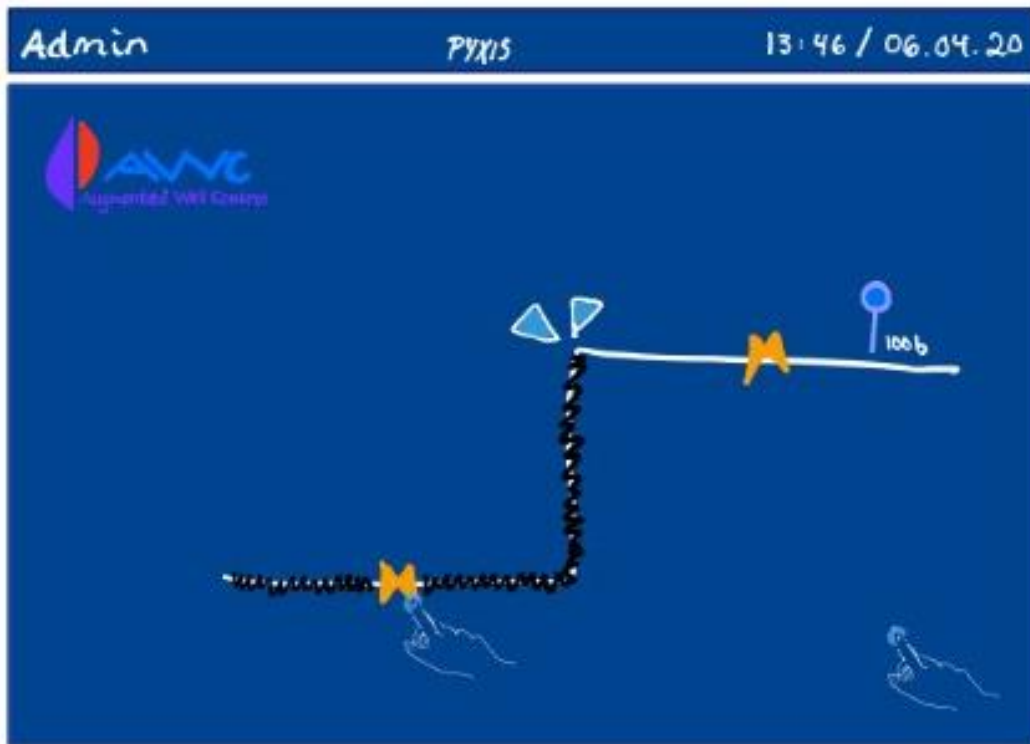


Figure 122 - Sketch of valve being pressed

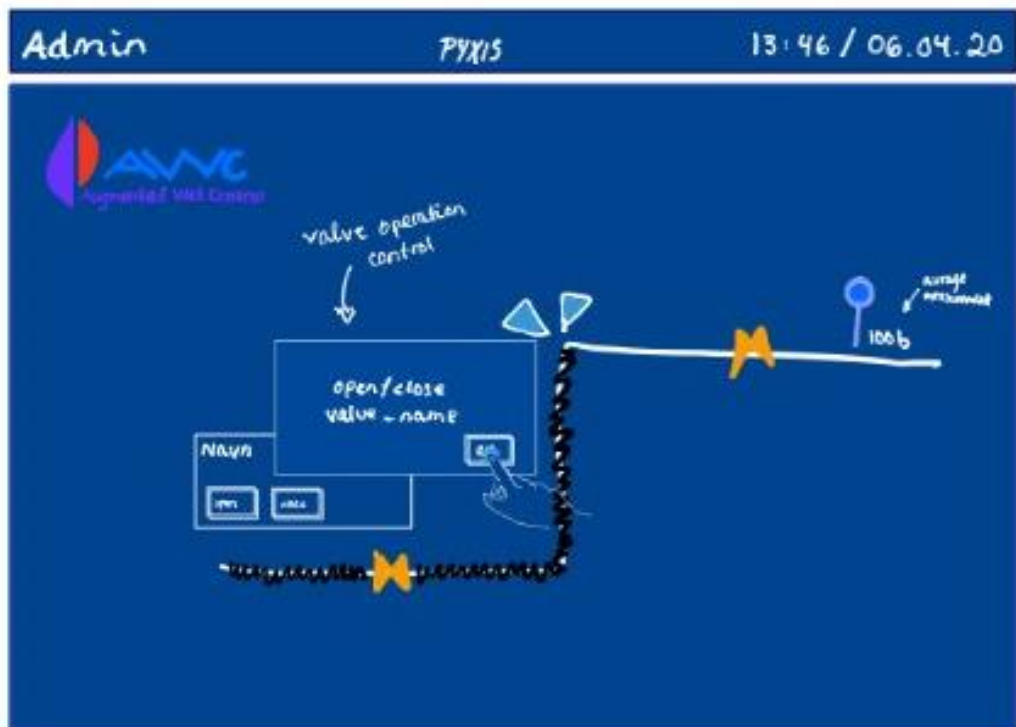
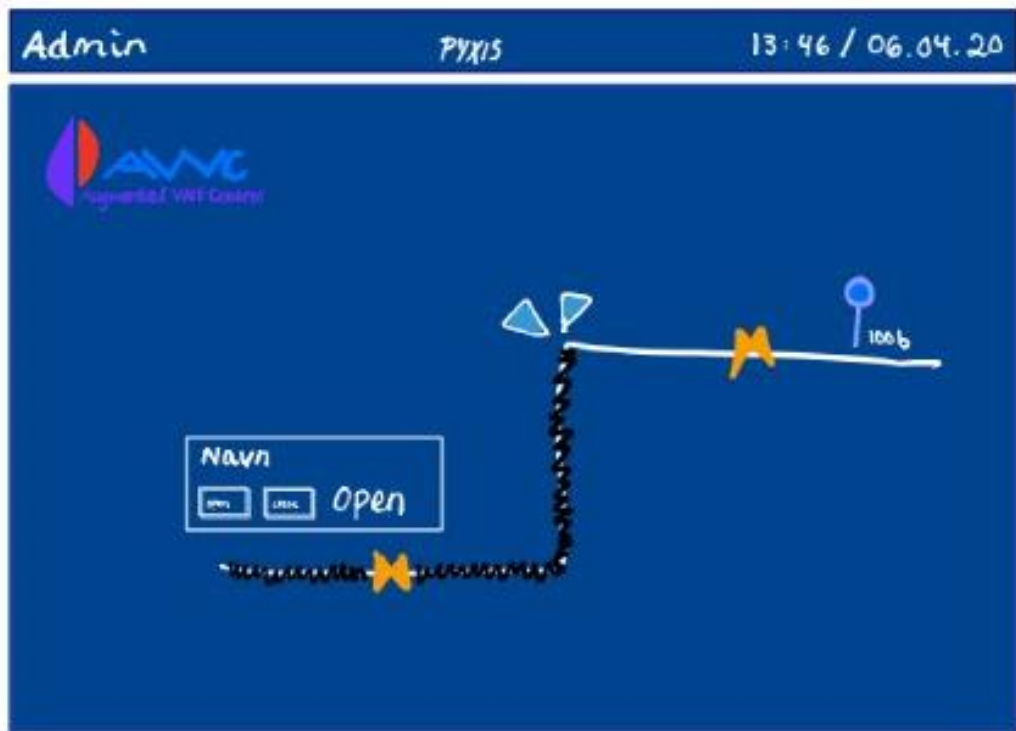


Figure 123 - Sketch of security plate for valve operation

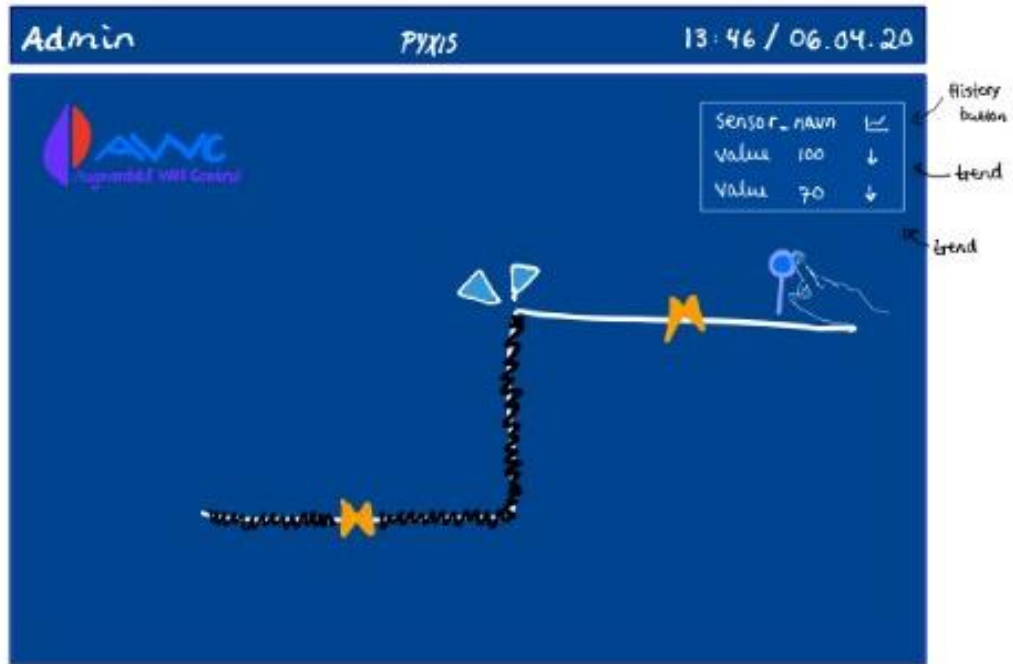


Figure 124 - Sketch of sensor is being pressed

5. Hand - menu

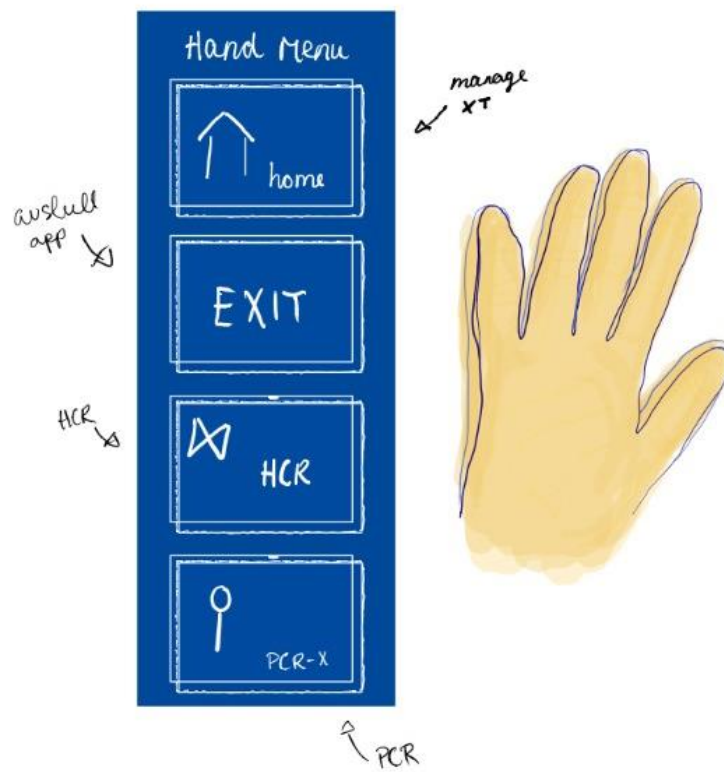


Figure 125 - Sketch of Hand Menu

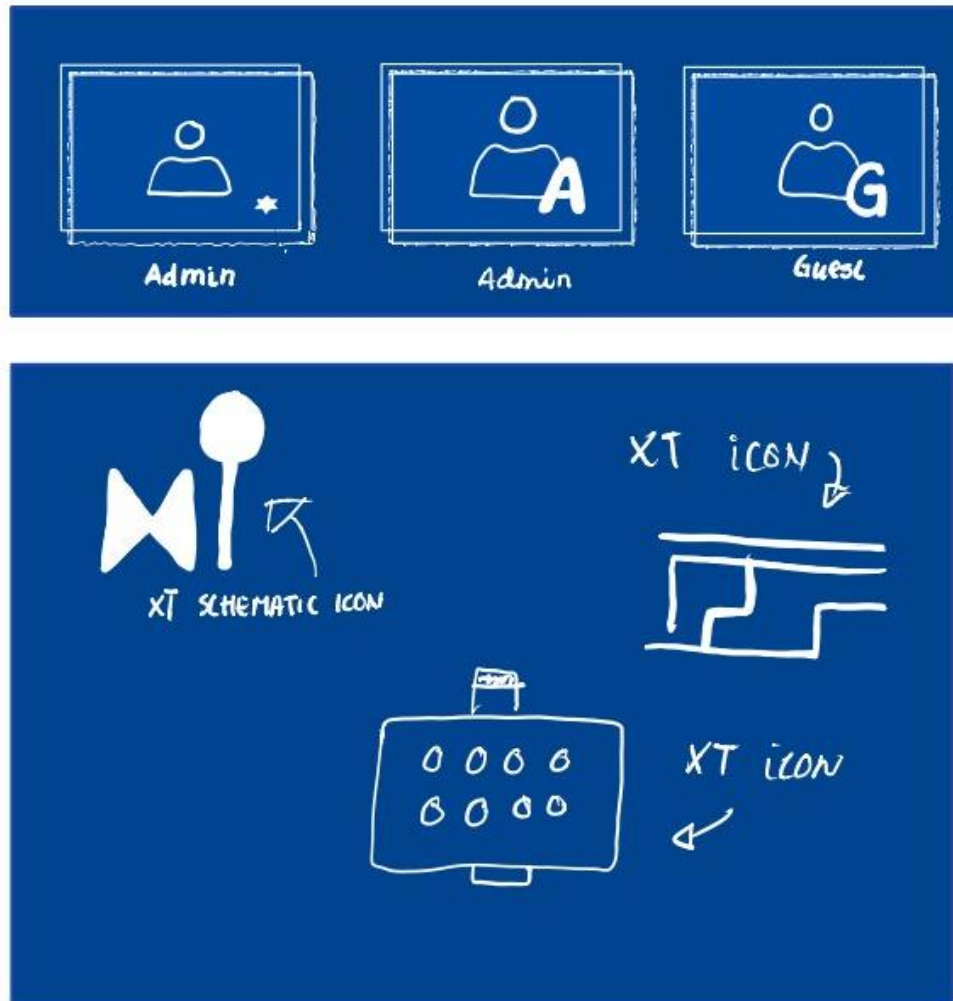


Figure 126 - Sketch of icons

9.5 UX design ideas – Brainstorming

- Hide/view function – Only show sensor/valves
- Day/Night mode – different colours depending on the room you are located in
- Component list – another way to access sensor and valve information.
- Different colours on opened and closed valves – indication of where there is oil flow
- Animation of oil through the pipes – visualizing which valves that are open and which that is closed
- Arrows - pointing the direction of the oil flow
- Trend arrows – next to names of the sensor in the XT view
- Hide/View function of names
- Read Me
- Pin to surface – use the walls and surfaces of the room to attach the different views.
- Tool tip – a little description of the functionality

9.6 Suggestions of UI – Brainstorming

9.6.1 First UI Design Suggestion

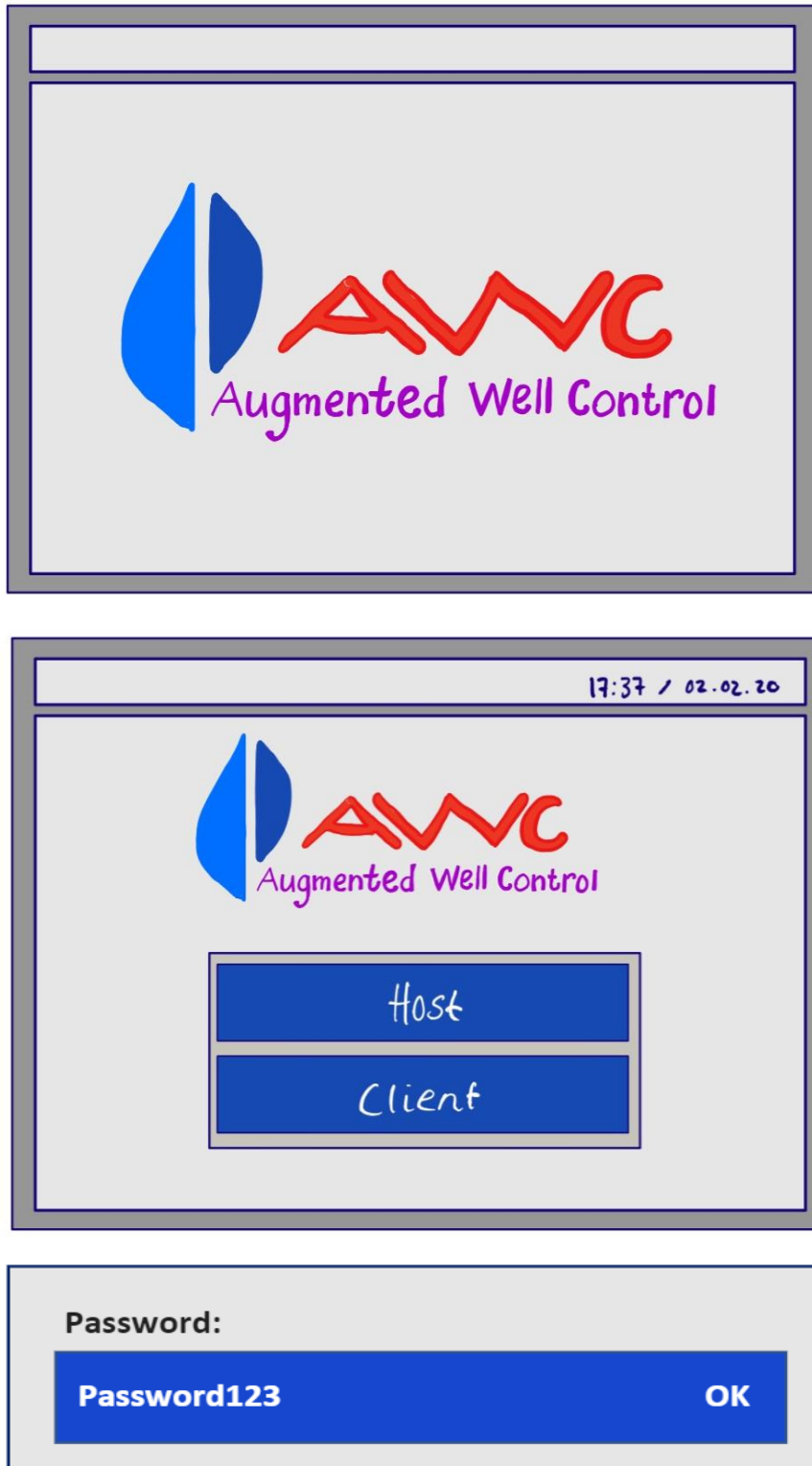


Figure 127 - Sketch of first UI suggestion 1

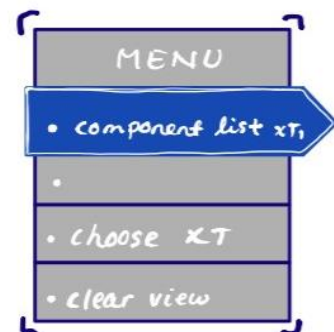
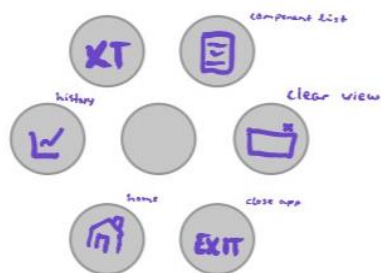
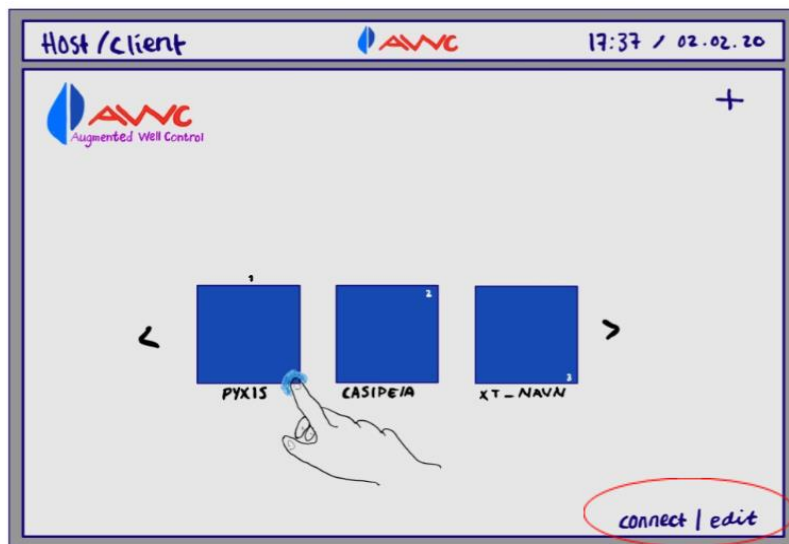
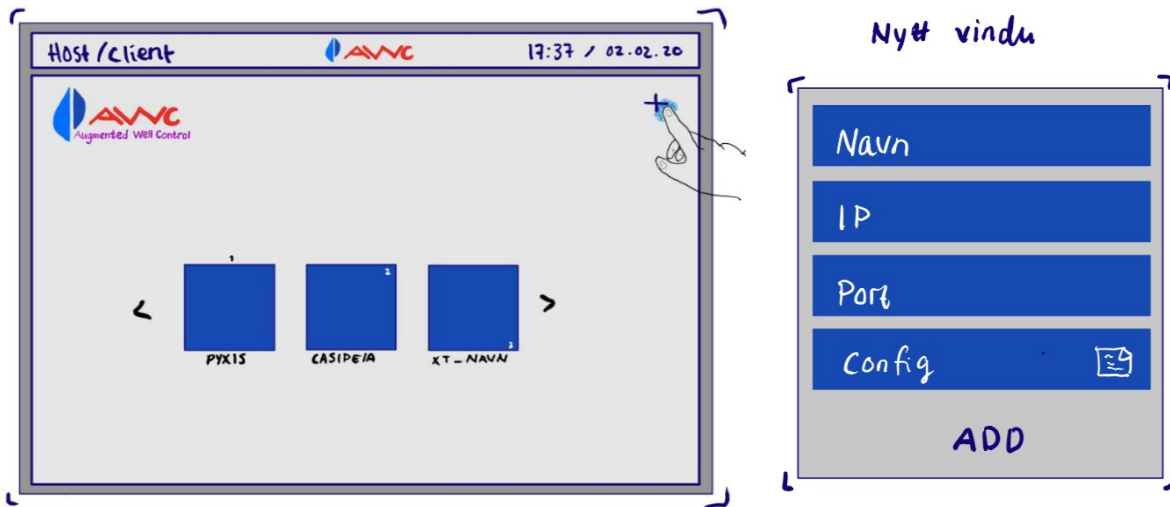


Figure 128 - Sketch of first UI suggestion 2

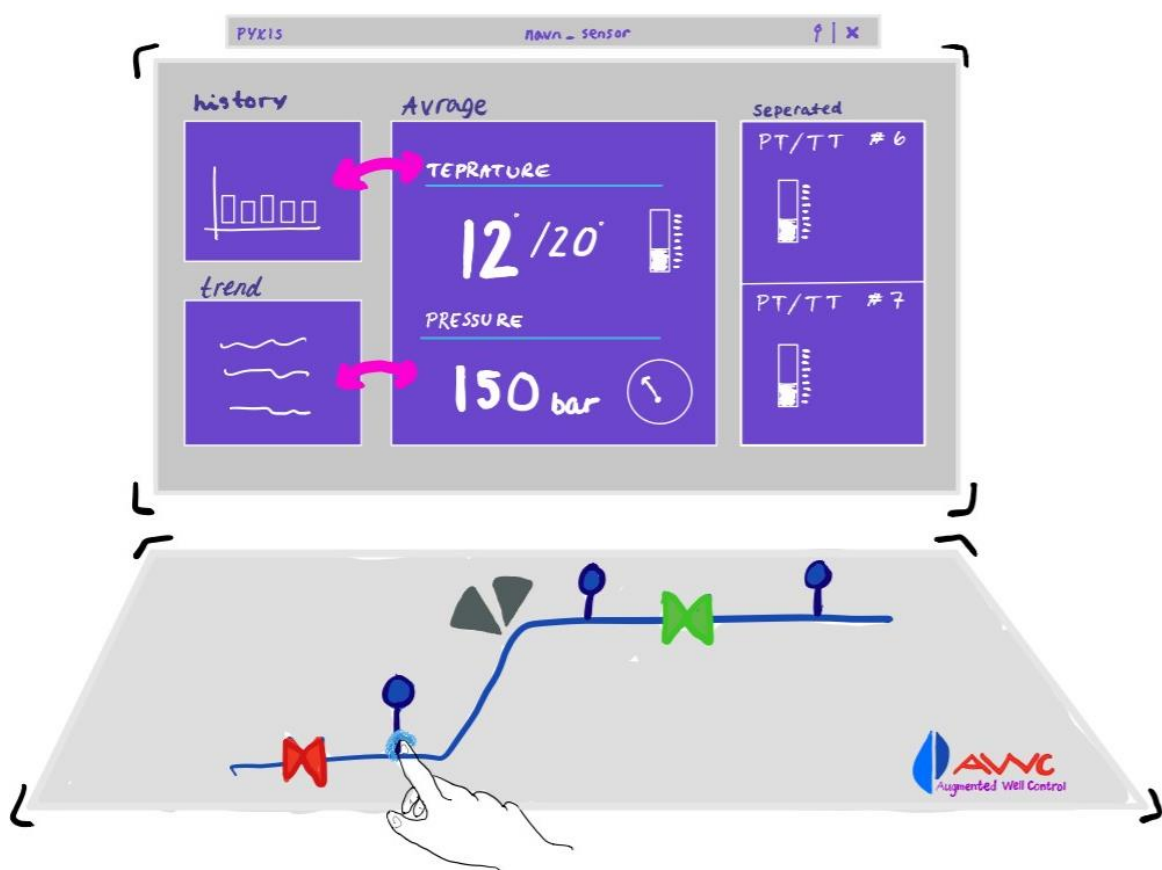
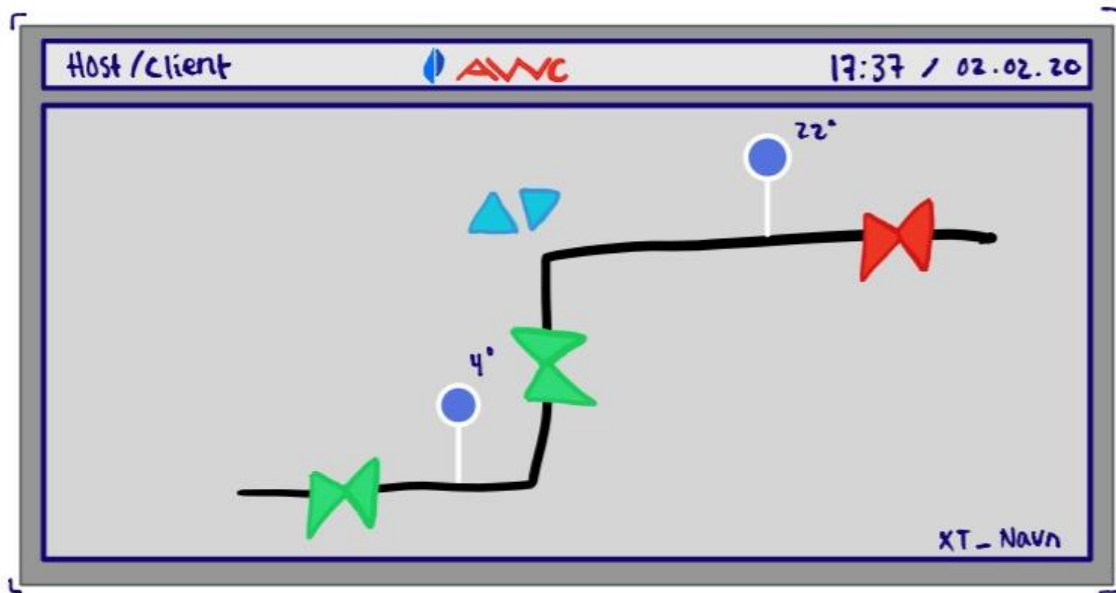


Figure 129 - Sketch of first UI suggestion 3

9.6.2 Second UI Design Suggestion

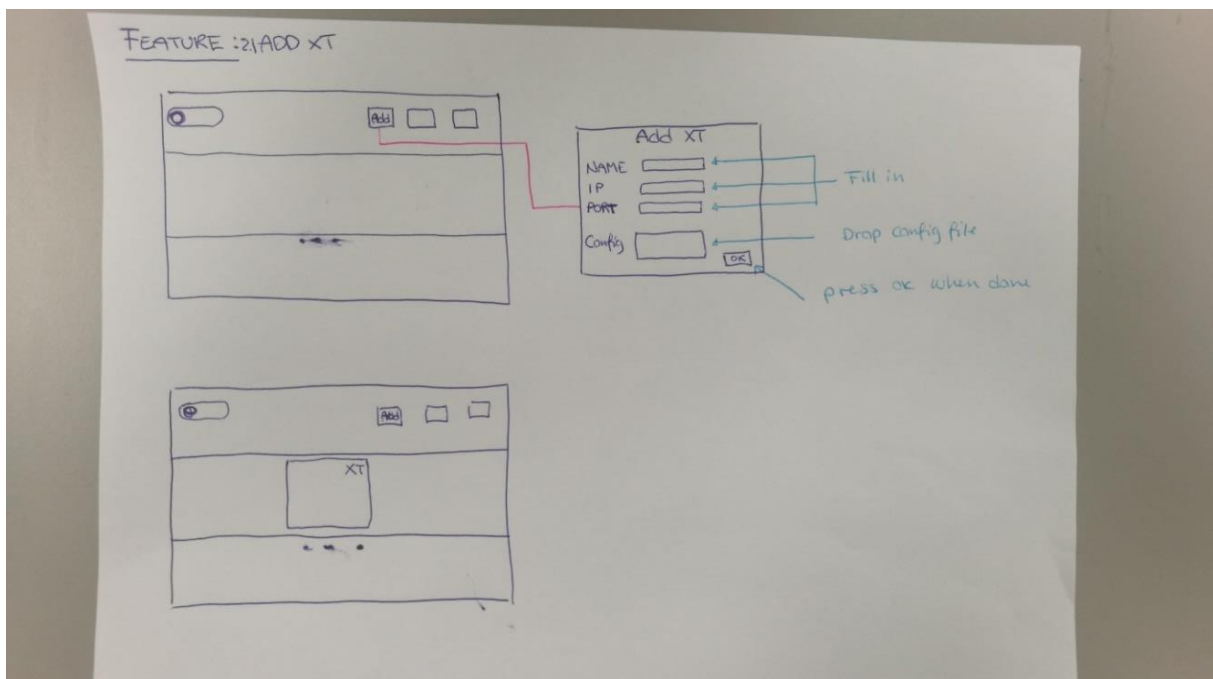
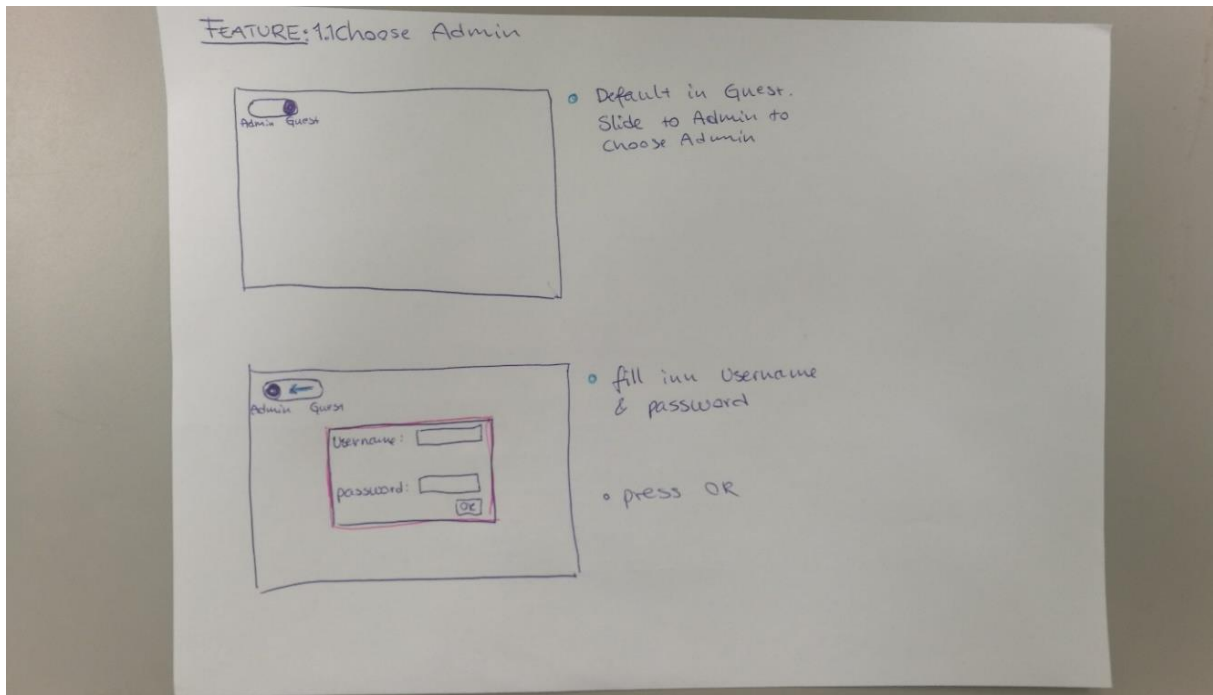


Figure 130 - Sketch of second UI suggestion 1

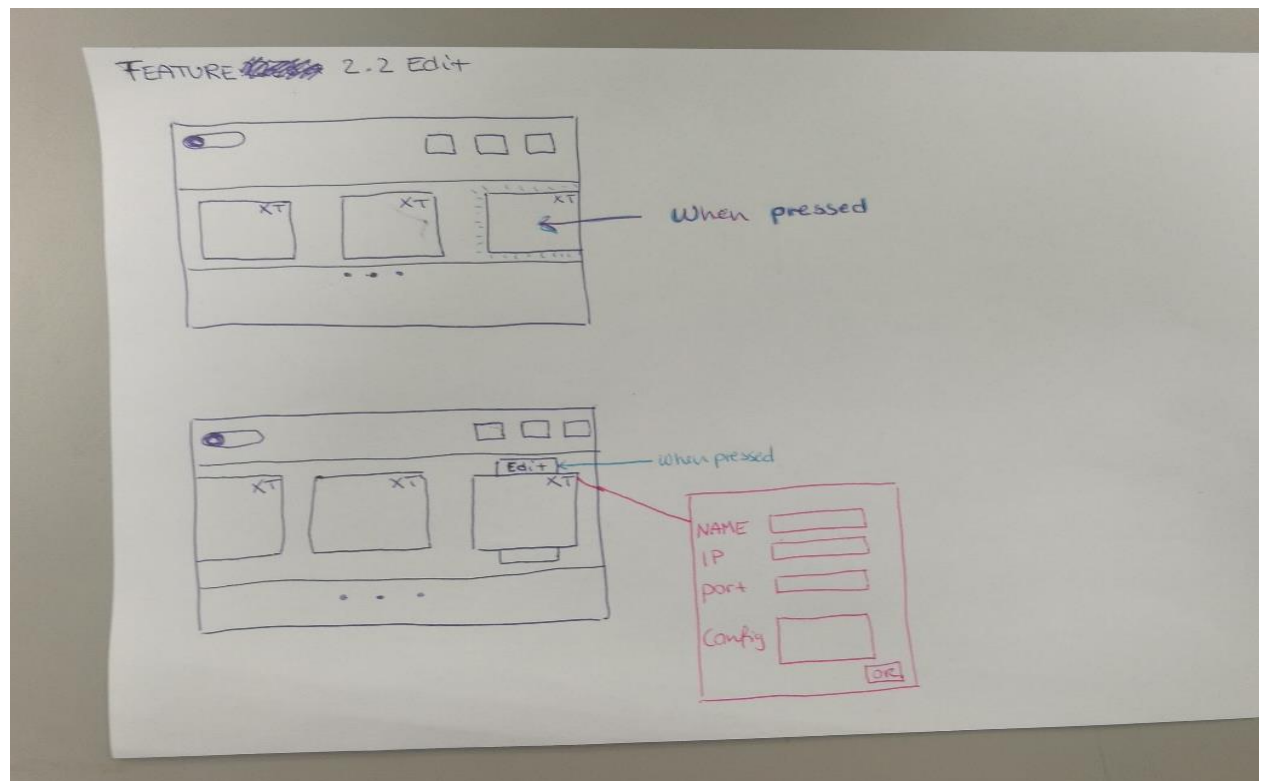
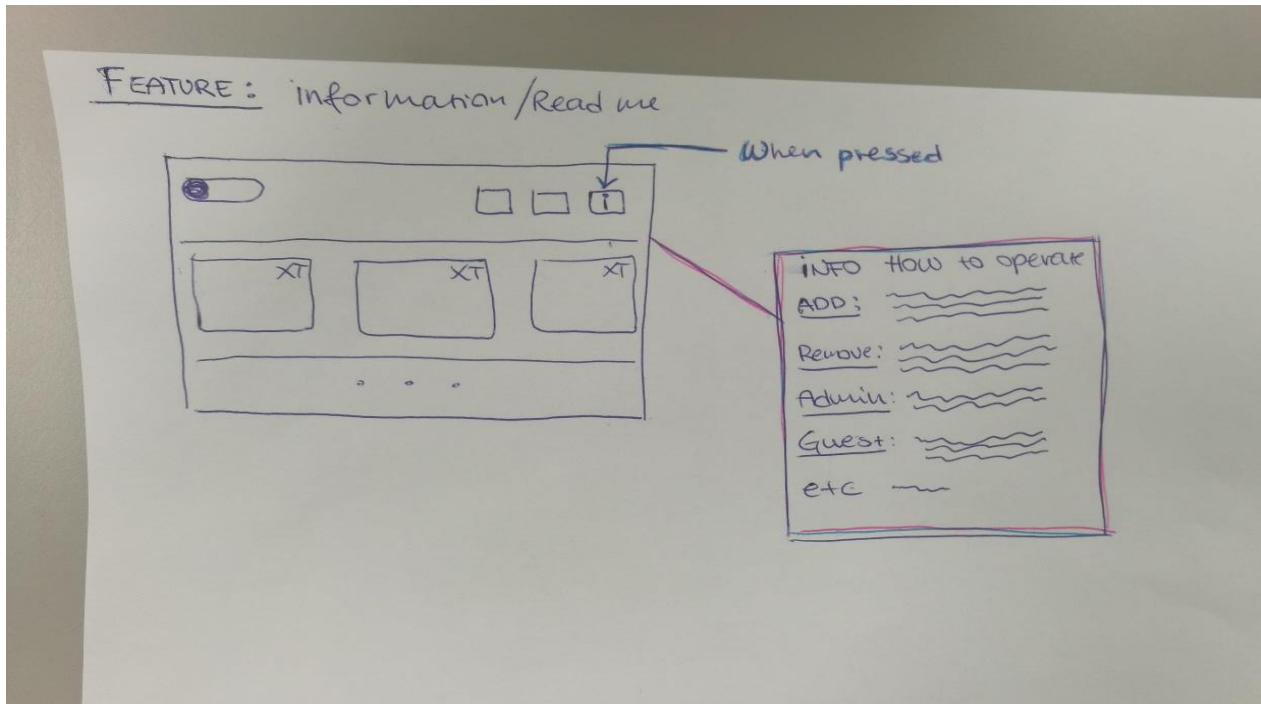


Figure 131 - Sketch of second UI suggestion 2

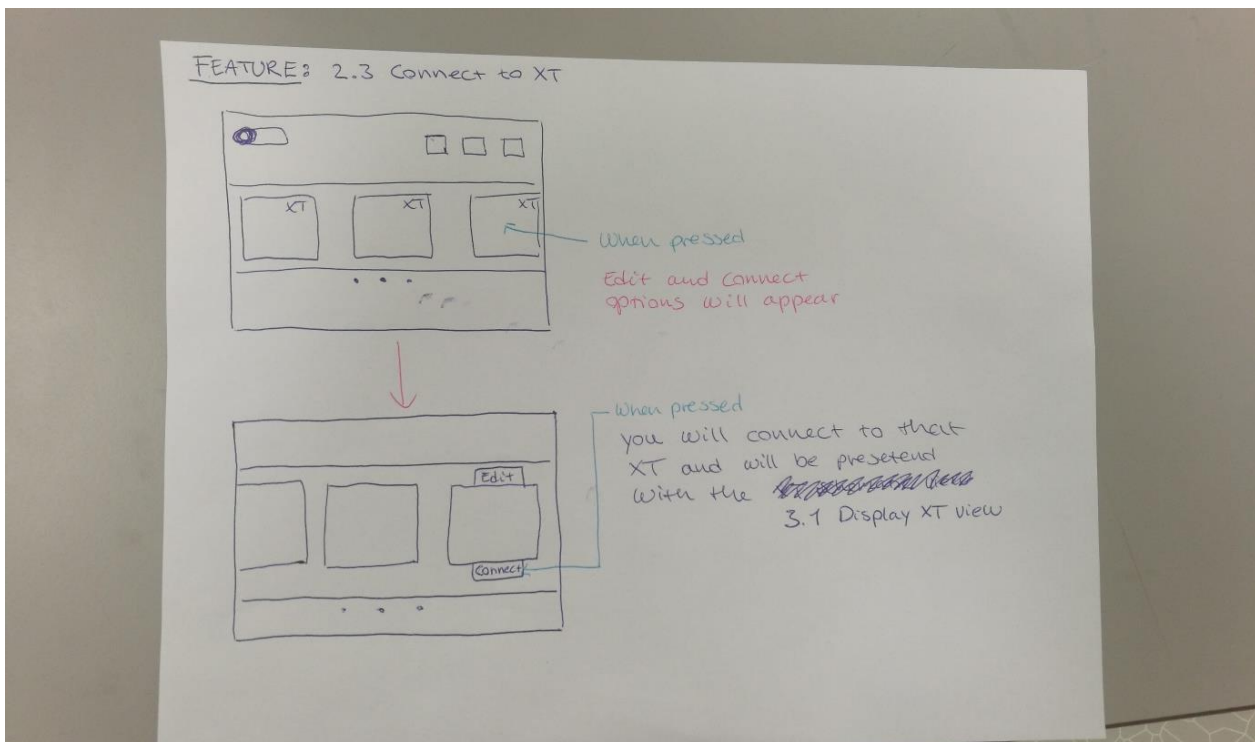
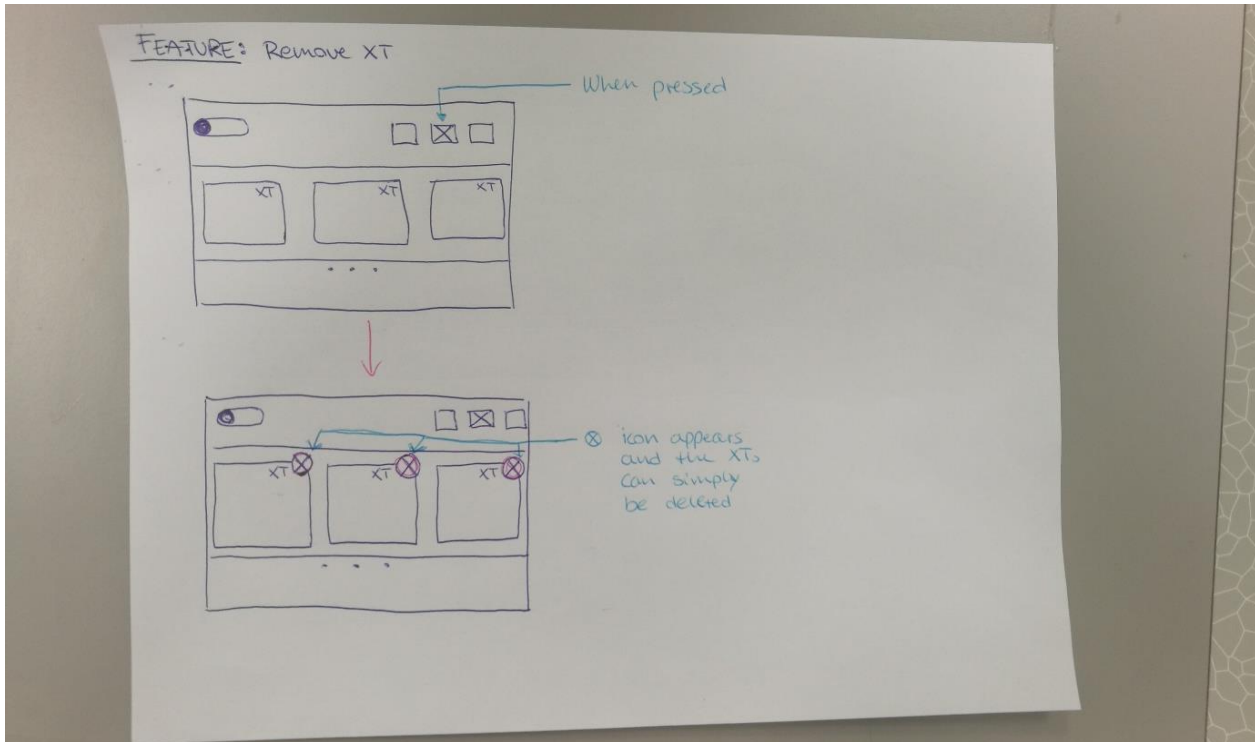


Figure 132 - Sketch of second UI suggestion 3

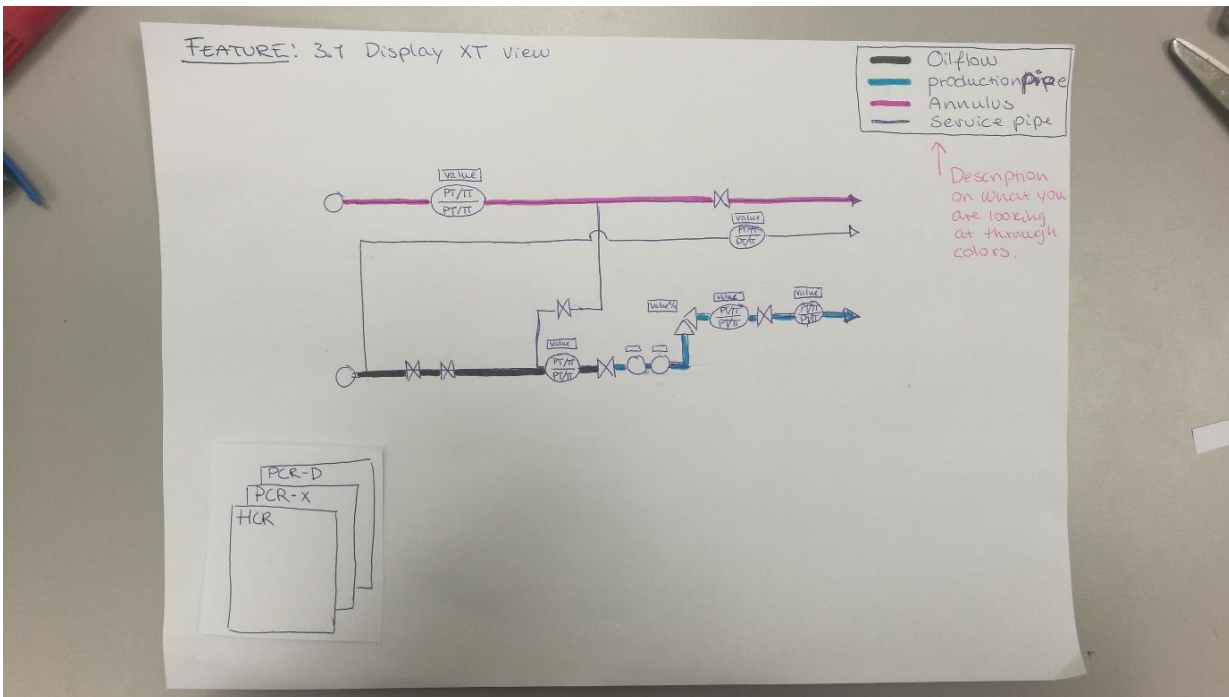
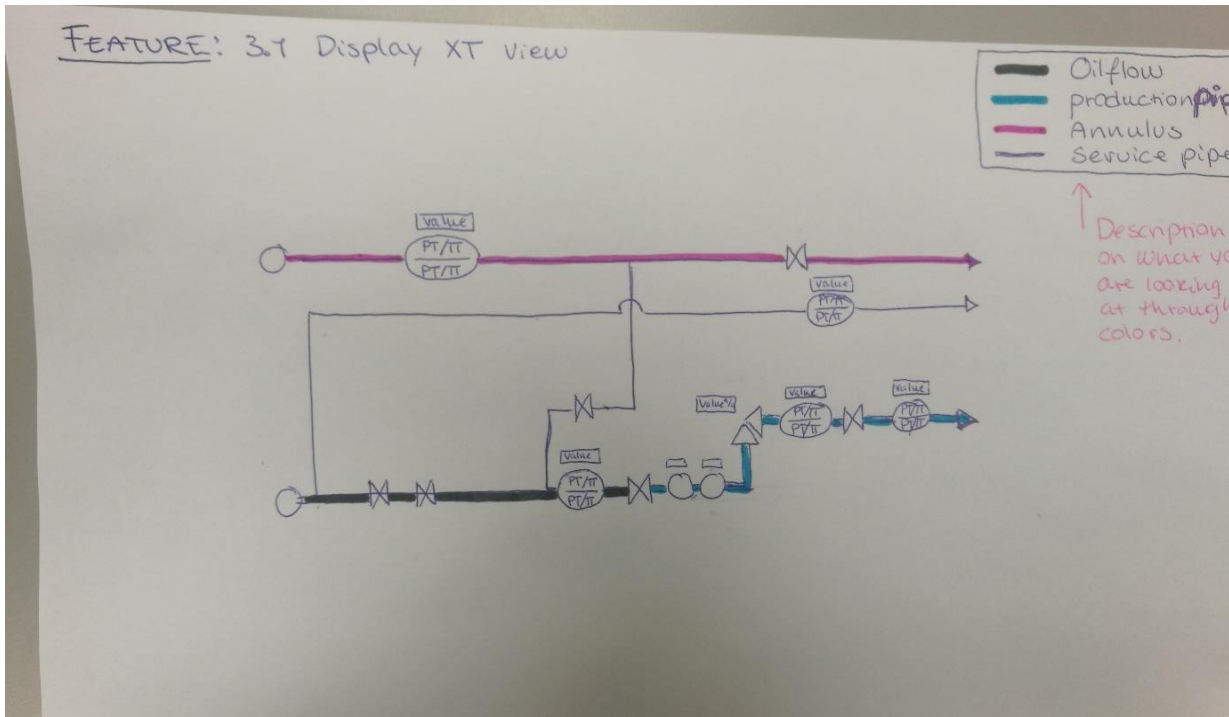


Figure 133 - Sketch of second UI suggestion 4

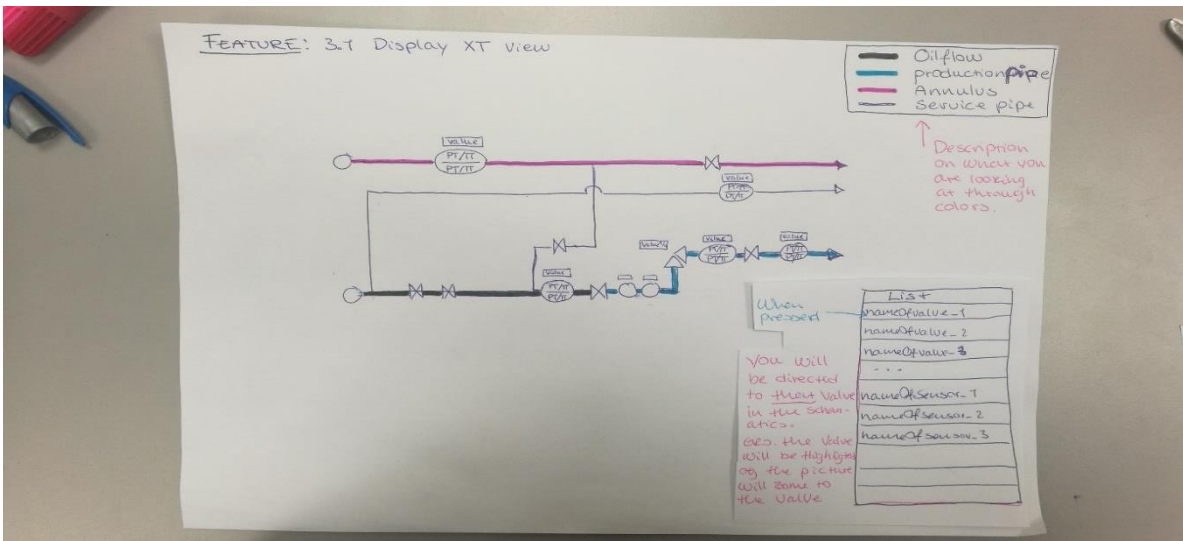
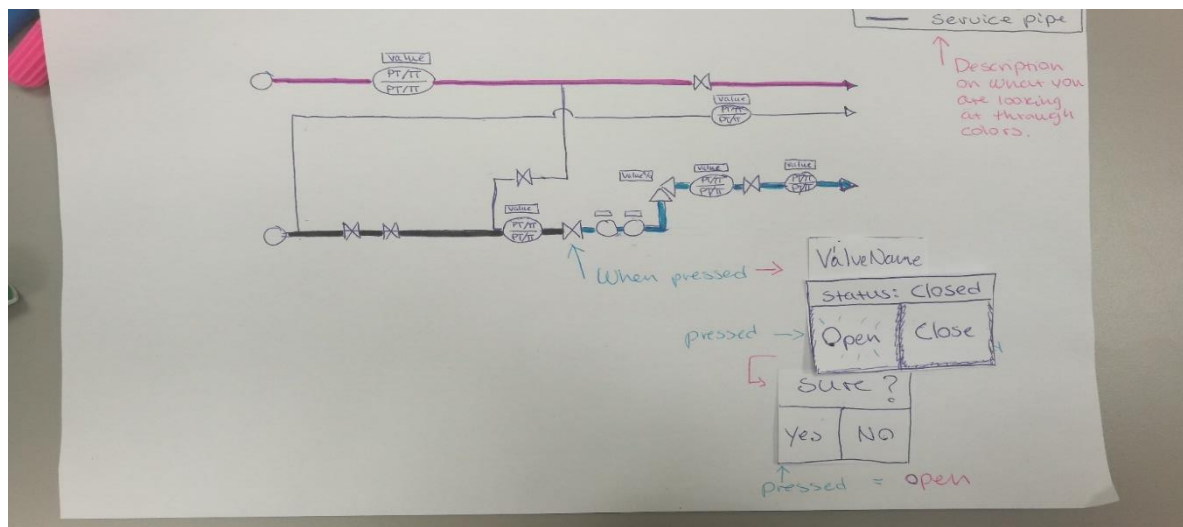
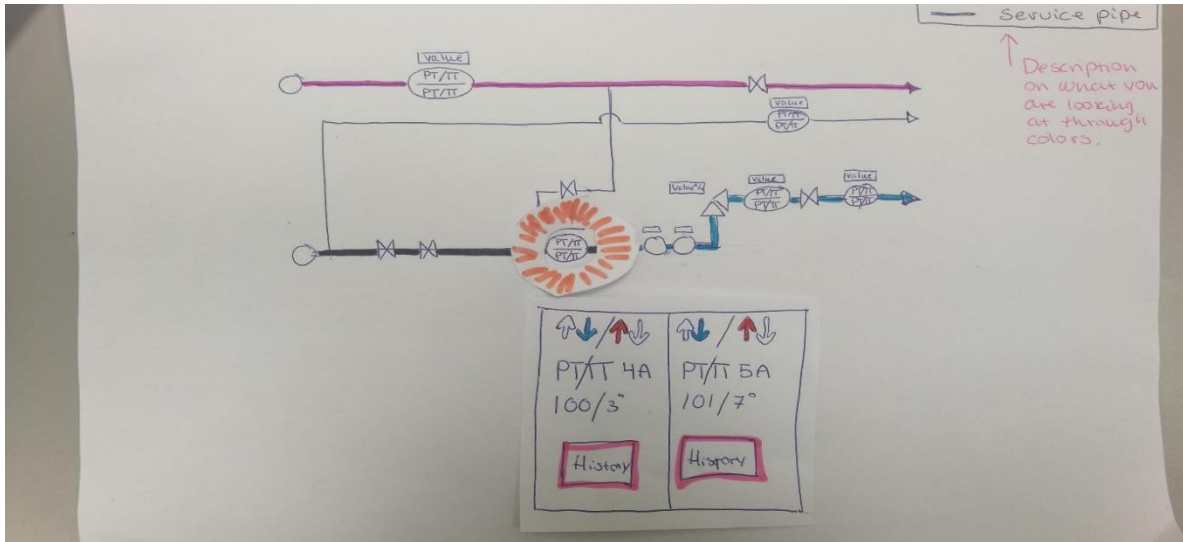


Figure 134 - Sketch of second UI suggestion 5

9.6.3 Third UI Design Suggestion

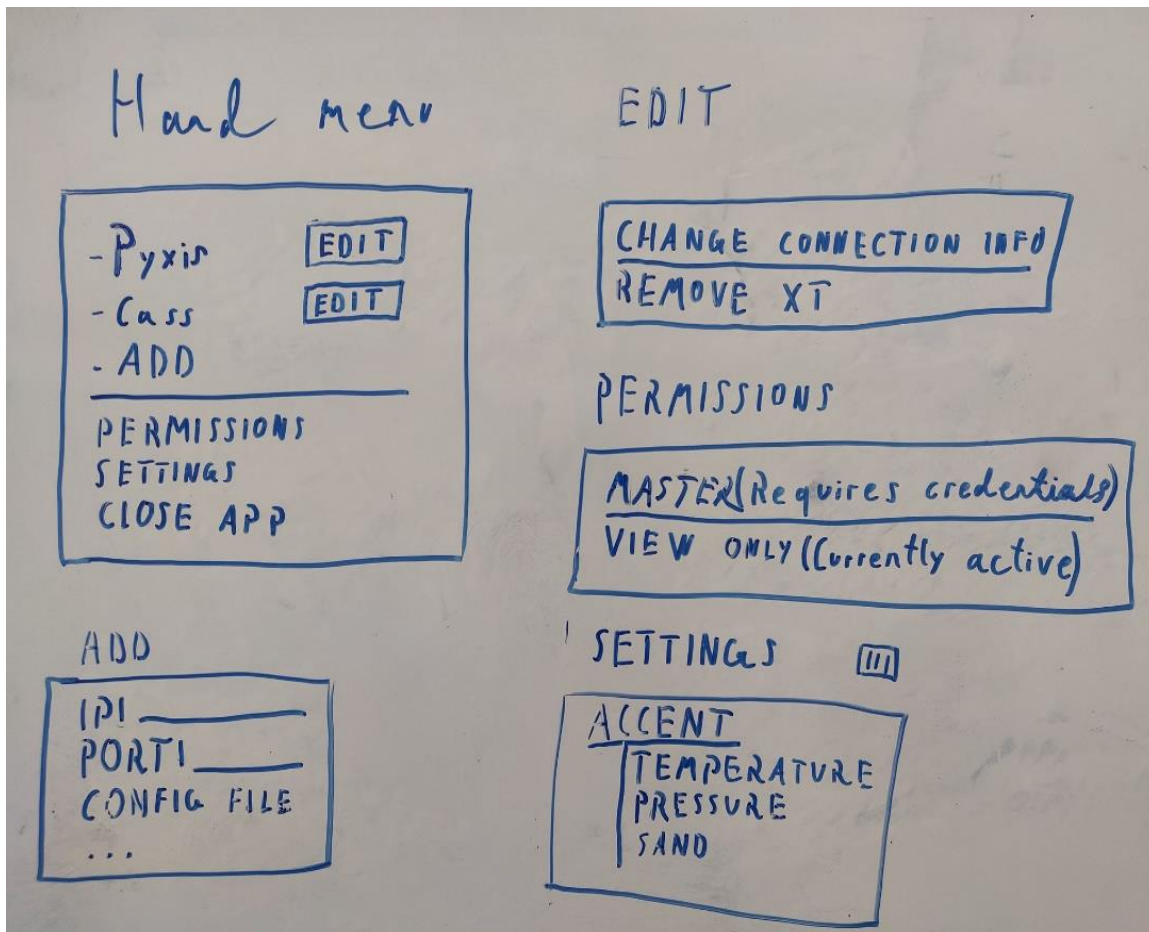
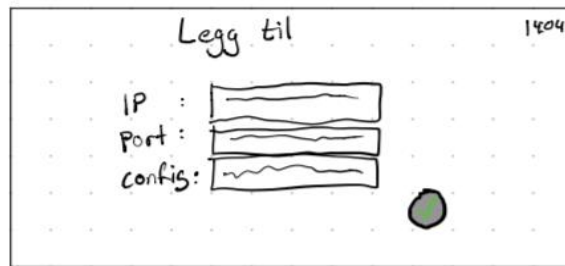


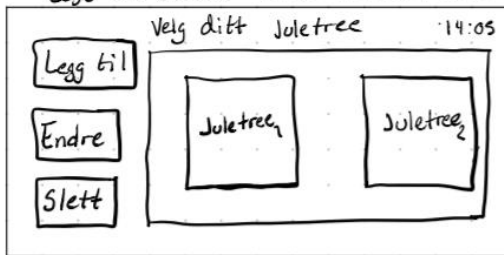
Figure 135 - Sketch of third UI suggestion 1

9.6.4 Fourth UI Design Suggestion

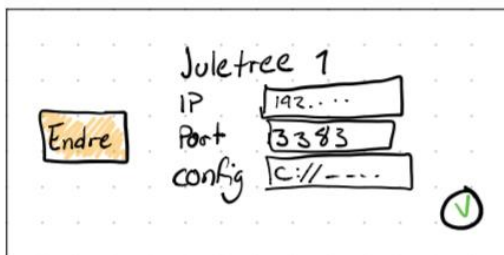
Jafop Første gang
connection vindu første gang



Legg til virker samme som første vindu



Endre



Slett

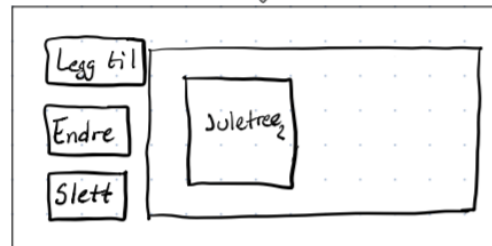
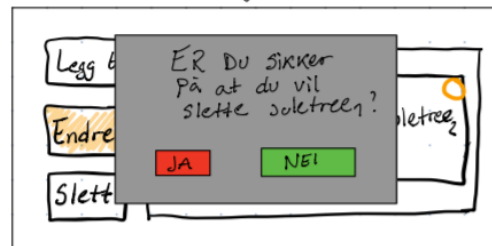
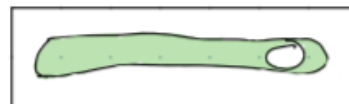
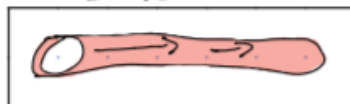


Figure 136 - Sketch of forth UI suggestion 1

Trykkes på et tree



Få opp en slider / gjøre bekreftelse for å Koble seg på.
slide to connect hvilke andre løsninger



Koble på xt.

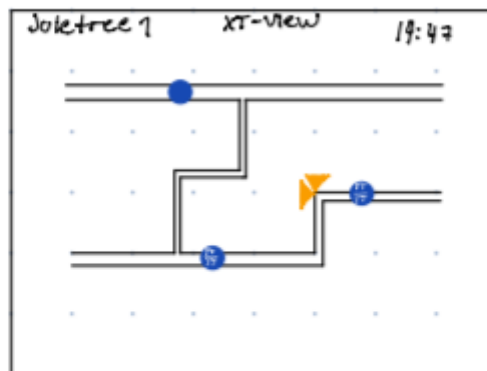


Figure 137 - Sketch of forth UI suggestion 2

9.6.5 Fifth UI Design Suggestion

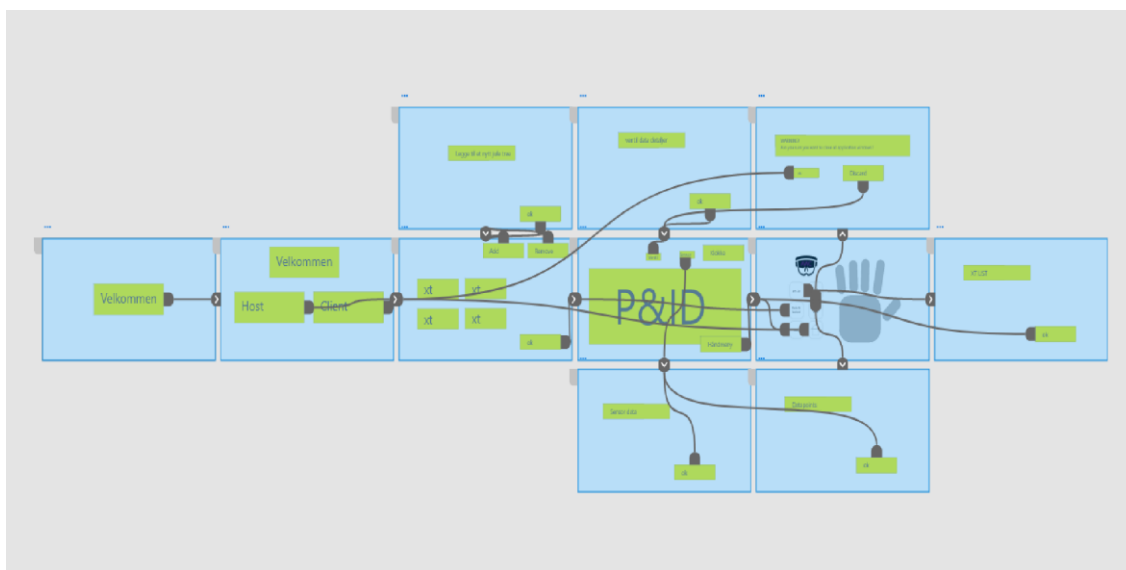


Figure 138 – Sketch of fifth UI suggestion 1

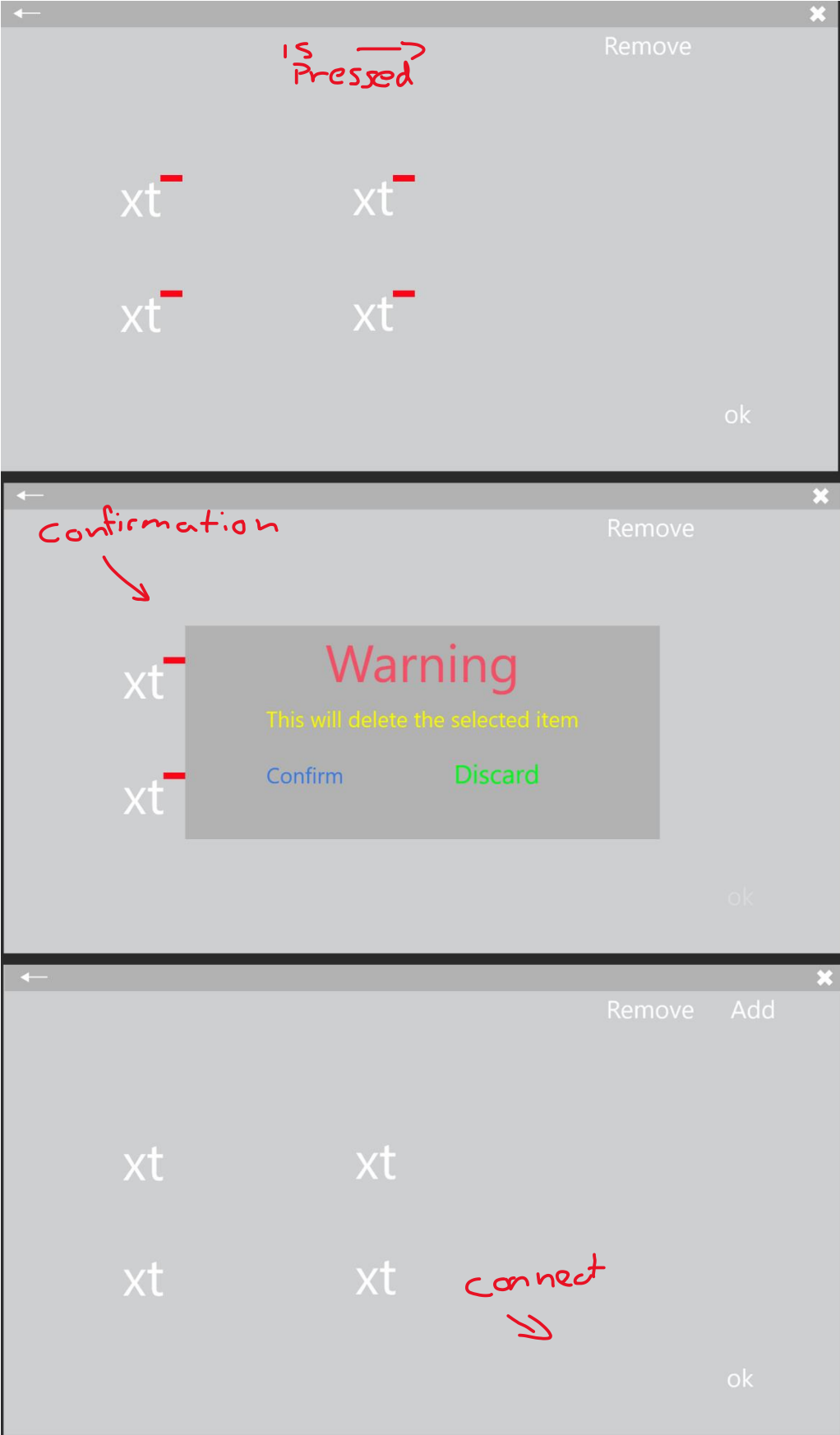


Figure 139 - Sketch of fifth UI suggestion 2

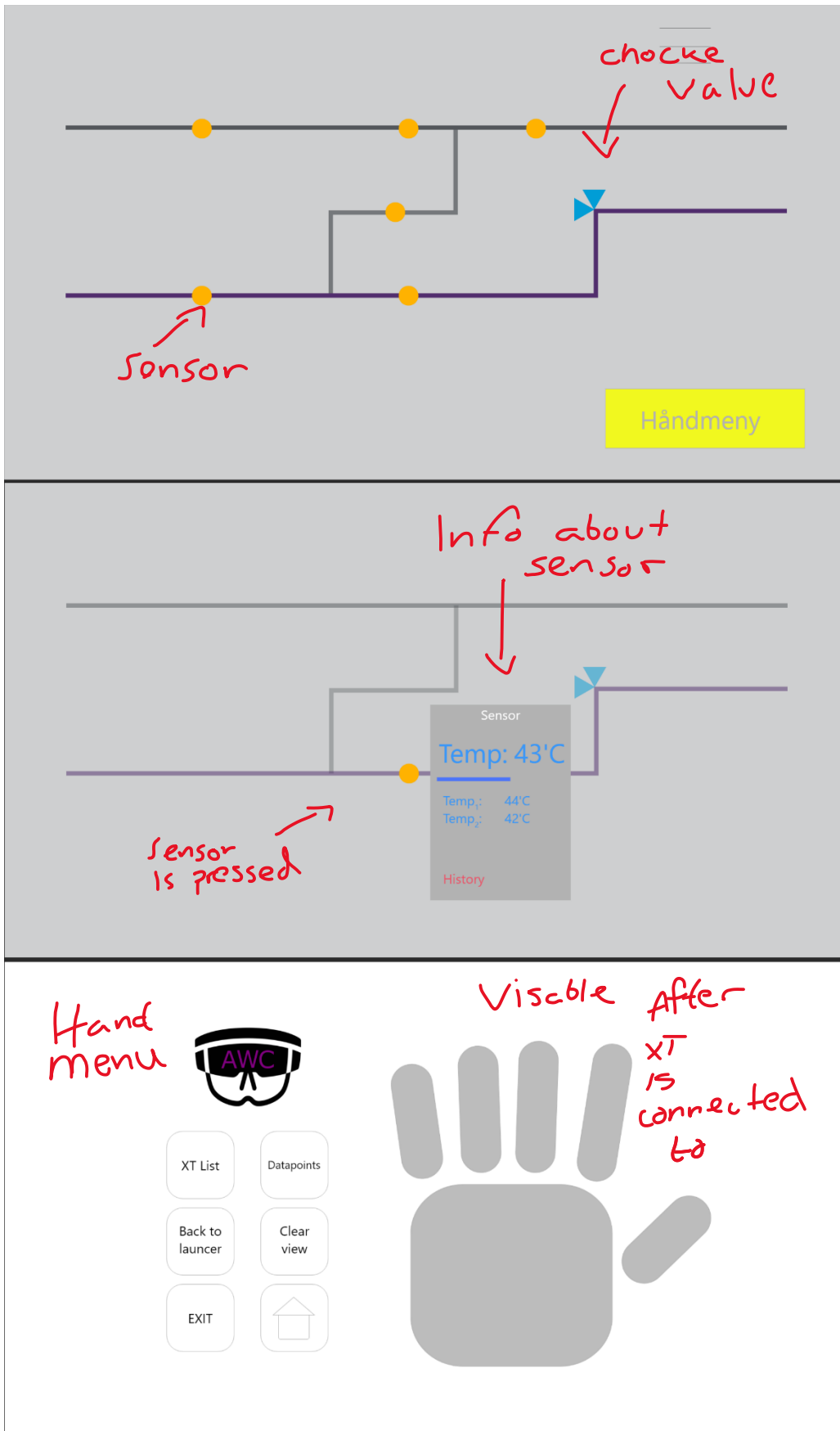


Figure 140 - Sketch of fifth UI suggestion 3

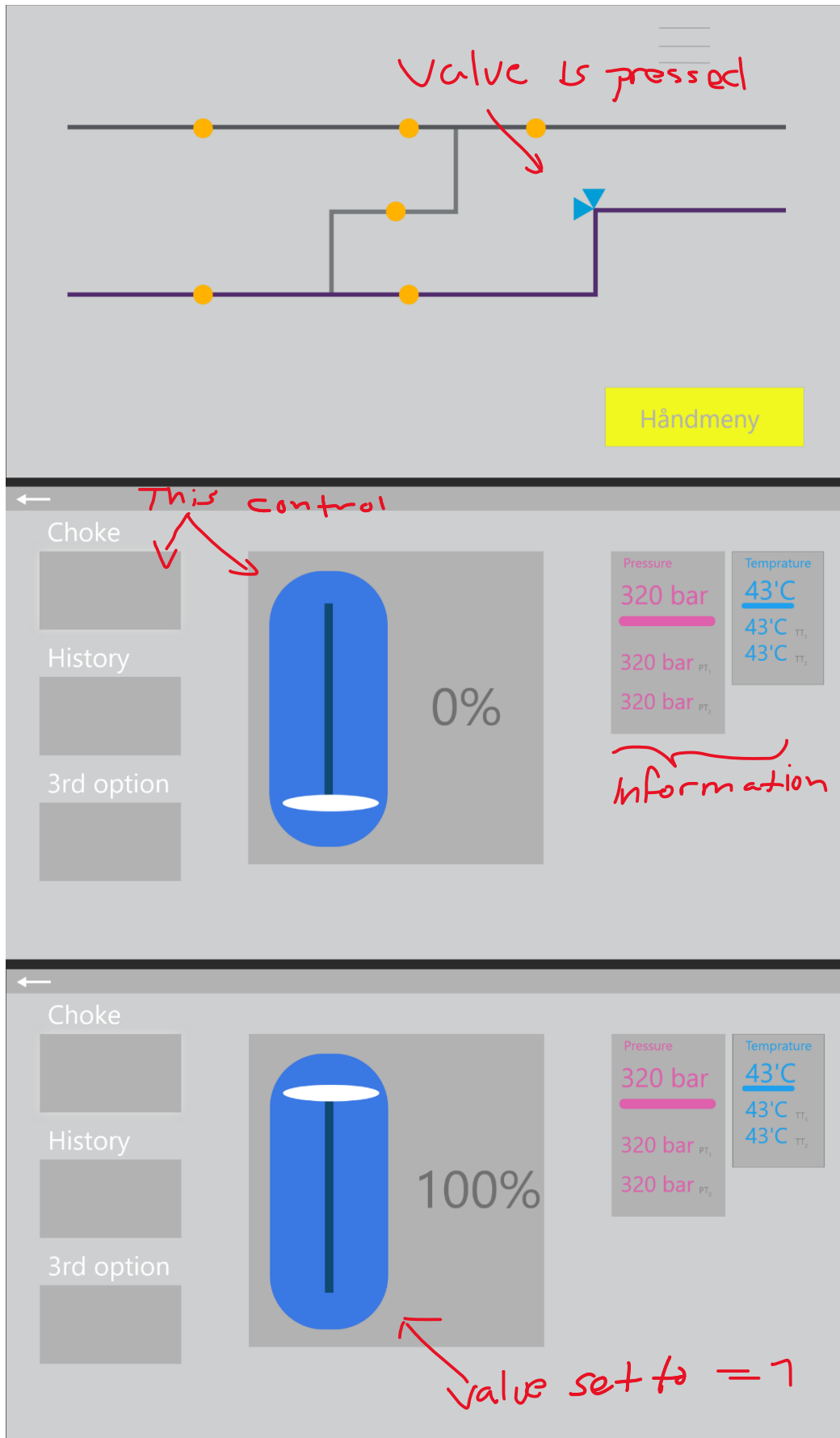


Figure 141 - Sketch of fifth UI suggestion 4



10 APPENDIX B – TEST DOCUMENTATION

Abstract

This document will contain test related documentation.

1. Acceptance Criteria Test information and templates
2. Completed user story tests
 - Accepted tests
 - Rejected Tests
3. Usability test information and template
4. Completed usability tests
 - Participant 1
 - Participant 2

10.1 Acceptance Criteria Test templates

Test #	User Story #	Feature #	User Story
Acceptance Criteria:			
Testing Information:			
Struggles:			
Testing date:		Accepted	

Test #	User Story #	Feature #	User Story
Acceptance Criteria:			
Testing Information:			
Struggles:			
Testing date:		Rejected	

Test#: The number assigned to the test will work as an “id” to the test and should be traceable when mentioned other places in our documentation. A test can have multiple user stories or only one. If one test has multiple user stories it will be marked by adding an increasing letter to the second slot ex. Test 1.0 has two user stories. The tests will be marked as 1.A and 1.B. They will have the same User Story# and acceptance criteria.

User Story#: If one user story has multiple tests the User Story# will be the same on different ID'ed tests ex: User story# 89 has two tests: test# 3.0 and 4.0. The acceptance criteria will differ, so the tests will be different.

Feature#: Which feature the test belongs to.

Testing information: Is a field where a more detailed list of the testing process comes through.

Struggles: The Struggles section is where the tester fills out the problems that occurred when trying to fulfil the acceptance criteria.

Testing date: The date the test was executed.

10.2 Completed tests

Test #	User story #	Feature #	User story
1.0	46	Prototype	As a developer I want to extract data from the simulator/InControl. So, I can see that our future system is possible.
Acceptance Criteria:			Given data from simulator When operating in a system (InControl/Unity/HoloLens2) Then view the same data
Testing Information:			Connect to API, connect unity to API, connect app for HL2 to API then upload the app in HL2 and display.
Struggles:			
<p><u>Struggle 1:</u> Connection issues. Could not connect HoloLens 2 to computer. Resolved it by connecting using USB C to USB A cable (USB 3.0) first. Then paired HoloLens 2 to the computer and finally remote debugged in Visual Studio.</p> <p><u>Struggle 2:</u> HoloLens 2 could not communicate with simulator over Wifi. Resolved it by using following project settings in unity under Project Settings - Player - Capabilities:</p> <ul style="list-style-type: none"> Internet Client Internet Client & Server PrivateNetwork Client & Server 			
Testing date: 13.02.2020			Accepted

Test #	User Story #	Feature #	User Story
2.0	162	2.1	As a developer I want to implement the functionality to the menu that takes an IP address, Name and Port so that the user can fill in the necessary details
Acceptance Criteria:			Given the add XT menu And input slots for the data When filled in Then the data is stored, and the input slots are linked correctly
Testing Information:			When pressing add XT in the add menu, the new XT is stored and displayed.
Struggles:			
<p><u>Struggle1:</u> When adding a new XT and generating a new list. The old list would still be there. Solved by deleting the old list first.</p>			
Testing date: 03.04 2020			Accepted

Test #	User Story #	Feature #	User Story
3.0	171	2.1	As a developer I want to implement the functionality to the menu that takes an IP address, Name and Port so that the user can fill in the necessary details
Acceptance Criteria:			Given the add XT menu And input slots for the data When filled in Then the data is stored and the input slots are linked correctly
Testing Information:			When pressing add XT in the add menu, the new XT is stored and displayed.
Struggles:			
<u>Struggle1:</u> When adding a new XT and generating a new list. The old list would still be there. Solved by deleting the old list first.			
Testing date: 03.04 2020			Accepted

Test #	User Story #	Feature #	User Story
4.0	159	2.2	As a user I want to change a XT's IP-address and port number so that I can easily configure it on a new network
Acceptance Criteria:			Given an existing XT When editing its IP address and port number Then the new values are saved
Testing Information:			When pressing save in the edit menu the new values are saved.
Struggles:			
<u>Struggle1:</u> Storing list of XT objects correctly in the textfile using JsonUtility. Resolved by adding a wrapper class to serialize correctly.			
Testing date: 03.04 2020			Accepted

Test #	User Story #	Feature #	User Story
5.0	13	2.3	As a developer I want to implement the functionality to the menu that takes an IP address, Name and Port so that the user can fill in the necessary details
Acceptance Criteria:			Given a list of XT's When choosing a XT by point clicking on it Then I connect to the XT and open the XT view
Testing Information:			When pressing add XT in the add menu, the new XT is stored and displayed.
Struggles:			
<u>Struggle1:</u> When adding a new XT and generating a new list. The old list would still be there. Solved by deleting the old list first.			
Testing date: 03.04.2020			Accepted

Test #	User Story #	Feature #	User Story
6.0	171	2.3	As a developer I want to implement the functionality to the menu that takes an IP address, Name and Port so that the user can fill in the necessary details
Acceptance Criteria:			Given a list of XT's When choosing a XT by point clicking on it Then I connect to the XT and open the XT view
Testing Information:			When pressing add XT in the add menu, the new XT is stored and displayed.
Struggles:			
<u>Struggle1:</u> When adding a new XT and generating a new list. The old list would still be there. Solved by deleting the old list first.			
Testing date: 03.04 2020			Accepted

Test #	User Story #	Feature #	User Story
7.0	159	2.2	As a user I want to change a XT's IP-address and port number so that I can easily configure it on a new network
Acceptance Criteria:			Given an existing XT When editing its IP address and port number Then the new values are saved
Testing Information:			When pressing save in the edit menu the new values are saved.
Struggles:			
<u>Struggle1:</u> Storing list of XT objects correctly in the textfile using JsonUtility. Resolved by adding a wrapper class to serialize correctly.			
Testing date: 03.04.2020			Accepted

Test #	User Story #	Feature #	User Story
8.0	162	2.1	As a developer I want to implement the functionality to the menu that takes an IP address, Name and Port so that the user can fill in the necessary details
Acceptance Criteria:			Given the add XT menu And input slots for the data When filled in Then the data is stored, and the input slots are linked correctly
Testing Information:			When pressing add XT in the add menu, the new XT is stored and displayed.
Struggles:			
<u>Struggle1:</u> When adding a new XT and generating a new list. The old list would still be there. Solved by deleting the old list first.			
Testing date: 03.04.2020			Accepted

Test #	User Story #	Feature #	User Story
9.0	171	2.3	As a developer I want to implement the functionality to the menu that takes an IP address, Name and Port so that the user can fill in the necessary details
Acceptance Criteria:			Given a list of XT's When choosing a XT by point clicking on it Then I connect to the XT and open the XT view
Testing Information:			When pressing add XT in the add menu, the new XT is stored and displayed.
Struggles:			
<u>Struggle1:</u> When adding a new XT and generating a new list. The old list would still be there. Solved by deleting the old list first.			
Testing date: 03.04.2020			Accepted

Test #	User Story #	Feature #	User Story
10.0	157	2.2	As a user, I want to be able to edit the name of an already existing XT, So that I can easier manage my XT's
Acceptance Criteria:			Given an existing XT When Editing its name Then The new name is saved
Testing Information:			When save button is pressed in Edit menu, new XT parameters are stored.
Struggles:			
Testing date: 03.04.2020			Accepted

Test #	User Story #	Feature #	User Story
11.0	159	2.2	As a user, I want to change a XT's IP-adress and port nr, So that i can easily configure it on a new network
Acceptance Criteria:			Given a existing XT When editing its IP-Adress and Port-nr Then the new values are saved
Testing Information:			When save button is pressed in Edit menu, new XT parameters are stored.
Struggles:			
Testing date: 03.04.2020			Accepted

Test #	User Story #	Feature #	User Story
12.0	171	2.3	As a user I want to connect to an already added XT by selecting it from a list with my finger, So that I can operate the XT
Acceptance Criteria:			Given a list of XT's When choosing a XT by point clicking on it Then I connect to the XT and open the XT view
Testing Information:			When save button is pressed in Edit menu, new XT parameters are stored.
Struggles:			
Testing date: 03.04.2020			Accepted

Test #	User Story #	Feature #	User Story
13.0	171	2.1	As a developer I want to implement the functionality to the menu that takes an IP address, Name and Port so that the user can fill in the necessary details
Acceptance Criteria:			Given the add XT menu And input slots for the data When filled in Then the data is stored and the input slots are linked correctly
Testing Information:			When pressing add XT in the add menu, the new XT is stored and displayed.
Struggles:			
<u>Struggle1:</u> When adding a new XT and generating a new list. The old list would still be there. Solved by deleting the old list first.			
Testing date: 03.04.2020			Accepted

Test #	User Story #	Feature #	User Story
14.0	137	2.1	As a user I want to select a config file with information about valves and sensors, so that I don't have to manually add it every time.
Acceptance Criteria:			Given a config file that contains: names, XT-addresses and placement index When selected Then the directory path is stored
Testing Information:			When XT schematic is loaded, the json file with the corresponding name is loaded.
Struggles:			
<u>Struggle1:</u> Referring to a valid datapath after the application was built. Resolved by using persitentDataPath.			
Testing date: 03.04.2020			Accepted

Test #	User Story #	Feature #	User Story
15.0	141		
Acceptance Criteria:		Given a XT menu When wanting to add a new XT And the add XT button is touched Then a XT input menu appears	
Testing Information:			When Add XT button is pressed in Manage XT menu Add New XT menu appears.
Struggles:			
Testing date: 03.04.2020			Accepted

Test #	User Story #	Feature #	User Story
16.0	157	2.2	As a user I want to be able to delete existing XT's, So that I can keep my view from being cluttered
Acceptance Criteria:			Given an existing XT When deleting it Then The XT disappears from my XT list
Testing Information:			When remove button is pressed. XT disappears.
Struggles:			
<u>Struggle1:</u> When adding a new XT and generating a new list. The old list would still be there. Solved by deleting the old list first.			
Testing date: 03.04.2020			Accepted

Test #	User Story #	Feature #	User Story
17.0	166	3.2	As a developer I want to implement the functionality of regulating the choke with a slider so that I can regulate the flow of oil
Acceptance Criteria:		Given a choke with a slider to operate When dragging a slider to regulate the oil Then the oil flow is regulated (data)	
Testing Information:		Using MRTK slider and ChokePrefab script	
Struggles: transform simulator values to match slidervalue text			
Testing date: 04.04.2020			Accepted

Test #	User Story #	Feature #	User Story
18.0	164	3.2	As a developer I want to open/close a valve in the simulator through Unity and HL2
Acceptance Criteria:		Given a valve in unity When touched Then the valve in the simulator opens or closes	
Testing Information:		The visuals; faceplate, buttons, events	
Struggles: making the button with correct press distance and pivot to be pressable			
Testing date: 04.04.2020			Accepted

Test #	User Story #	Feature #	User Story
19.0	142	3.2	As a user I want to implement the functionality of the valve so that when pressed it is either open or closed (sends correct data)
Acceptance Criteria:		Given a "valve" When touched Then the links work and it actually opens or closes in sim	
Testing Information:		Used a pressable button from MRTK. After extensive research and learning about how unity, HL2 and the simulator communicates The first action was to open and close PMV valve 102.	
Struggles: Catching up to Erik with the knowledge of Unity, simulator, HL2 and how they work together.			
Testing date: 07.04.2020			Accepted

Test #	User Story #	Feature #	User Story
--------	--------------	-----------	------------

20.0	169	3.1	As a user I want to see the valves and sensors displayed on a pipe that is configurable for different XT's so that I can intuitively see how the XT is performing
Acceptance Criteria:			<p>Given a config file with component types, names, placement, and addresses</p> <p>When opening the XT view</p> <p>Then a XT is drawn in the right order with corresponding values on components</p>
Testing Information:			Connecting to an XT from the Manage XT menu sends you to the XT Schematic view where you see the schematic with components spawned
Struggles: Placing in right position due to differences in local and global position.			
Testing date: 19.04.2020			Accepted

Test #	User Story #	Feature #	User Story
21.0	139	1.1	As a user I want to select admin mode So that I have the authority to open/close valves
Acceptance Criteria:			<p>Given selection between admin and guest mode</p> <p>And having a PIN</p> <p>When correct PIN is typed in</p> <p>Then having the authority to open/close valves</p>
Testing Information:			When correct PIN is typed in and singleton variable isAdmin is set true open/close/adjust valve buttons is visible.
Struggles:			
<u>Struggle 1:</u> Learning Unity and MRTK			
<u>Struggle 2:</u> Write to an InputField. MRTK keyboard is under development. Solved it by making my own keyboard and connect the interactable buttons to the InputField.			
<u>Struggle 3:</u> Add BCrypt NuGet packet. Worked in Visual Studio but Unity could not find it. Solved it by adding the right .dll file in the asset map PLUGGINGS and added an XML file to Unity.			
Testing date: 26.04.2020			Accepted

Test #	User Story #	Feature #	User Story
22.0	138	1.2	As a user I want to select guest mode So that I can view and do the same as admin except open/close valves.
Acceptance Criteria:			Given selection between admin and guest mode When selecting guest mode Then the system gives me the rights to all functionalities except open/close valves.
Testing Information:			When choosing Guest mode, the variable isAdmin is set to false and open/close/adjust valve buttons is not visible.
Struggles:			
Testing date: 28.04.2020			Accepted

Test #	User Story #	Feature #	User Story
23.0	309	4.5	As a user I want to be able to clear view so that I can quickly reset position and unclutter my view
Acceptance Criteria:			Given hand menu is visible AND I press clear view button When there are many faceplates open Then faceplates should snap back to position and close down
Testing Information:			When clear view is pressed it should unclutter the view for the user and snap faceplates back to coherent parent in the hierarchy.
Struggles:			
<u>Struggle1</u> : first bug found is that object had wrong local position (setting this to 0,0,0 inside a new vector3 needed to be tweaked). This was easily adjusted			
Testing date: 29.04.20			Accepted

Test #	User Story #	Feature #	User Story
24.0	313	4.4	As a user I want to be able to clear view so that I can quickly reset position and unclutter my view
Acceptance Criteria:			Given hand menu is visible When I press close application Then application should shut down
Testing Information:			By pressing the button, the application should shut down
Struggles:			
<u>Struggle1</u> : Application only quit inside of the unit application and not the whole UWP. Needs to be added. After this is fixed then button works as intended			
Testing date: 29.04.20			Accepted

Test #	User Story #	Feature #	User Story
25.0	313	4.4	As a user I want to close AWC application
Acceptance Criteria:			Given hand menu is visible When I press close application Then application should shut down
Testing Information:			By pressing the button, the application should shut down
Struggles:			
<u>Struggle1:</u> Application only quit inside of the unit application and not the whole UWP. Needs to be added. After this is fixed then button works as intended			
Testing date: 29.04.20			Accepted

Test #	User Story #	Feature #	User Story
26.0	309	4.5	As a user I want to be able to clear view so that I can quickly reset position and unclutter my view
Acceptance Criteria:			Given hand menu is visible AND I press clear view button When there are many faceplates open Then faceplates should snap back to position and close down
Testing Information:			When clear view is pressed it should unclutter the view for the user and snap faceplates back to coherent parent in the hierarchy.
Struggles:			
<u>Struggle1:</u> first bug found is that object had wrong local position (setting this to 0,0,0 inside a new vector3 needed to be tweaked). This was easily adjusted			
Testing date: 29.04.20			Accepted

Test #	User Story #	Feature #	User Story
27.0	313	4.4	As a user I want to be able to clear view so that I can quickly reset position and unclutter my view
Acceptance Criteria:			Given hand menu is visible When I press close application Then application should shut down
Testing Information:			By pressing the button, the application should shut down
Struggles:			
<u>Struggle1:</u> Application only quit inside of the unit application and not the whole UWP. Needs to be added. After this is fixed then button works as intended			
Testing date: 29.04.20			Accepted

Test #	User Story #	Feature #	User Story
28.0	313	4.4	As a user I want to close AWC application
Acceptance Criteria:			Given hand menu is visible When I press close application Then application should shut down
Testing Information:			By pressing the button, the application should shut down
Struggles:			
<u>Struggle1:</u> Application only quit inside of the unit application and not the whole UWP. Needs to be added. After this is fixed then button works as intended			
Testing date: 29.04.20			Accepted

Test #	User Story #	Feature #	User Story
29.0	319	4.1	As a user I want to be able to go back to manage xt screen so that I can edit or connect to a new xt
Acceptance Criteria:			Given hand menu is visible When I press home button Then the user should be taken to the Manage XT screen
Testing Information:			Should be able to go back to Manage XT screen by pressing button on the hand menu. This should destroy the XT schematic that was active and kill server communication while schematic window is not active.
Struggles:			
<u>Struggle1:</u> couldn't change screens properly and schematic couldn't be deactivated (needs to be destroyed).			
<u>Struggle2:</u> need to kill server communication properly. -fixed after some time			
Testing date: 30.04.20			Accepted

Test #	User Story #	Feature #	User Story
30.0	319	4.1	As a user I want to be able to go back to manage xt screen so that I can edit or connect to a new xt
Acceptance Criteria:			Given hand menu is visible When I press home button Then the user should be taken to the Manage XT screen
Testing Information:			Should be able to go back to Manage XT screen by pressing button on the hand menu. This should destroy the XT schematic that was active and kill server communication while schematic window is not active.
Struggles:			
<u>Struggle1:</u> couldn't change screens properly and schematic couldn't be deactivated (needs to be destroyed). <u>Struggle2:</u> need to kill server communication properly. -fixed after some time			
Testing date: 30.04.20			Accepted

Test #	User Story #	Feature #	User Story
31.B	251	4.1	As a user I want to read information about the application so that I can get a better understanding of the different functionalities
Acceptance Criteria:			Given a Hand Menu When ReadMe button is pressed Then Information about the app is displayed
Testing Information:			When Read Me is opened, the different tabs take the user to the different information about the application.
Struggles: Make an indication of what tab the user is at when navigating through the different tabs. Tried with making the icon "glowing"/bold but did not make much difference.			
Testing date: 08.05.2020			Accepted

Test #	User Story #	Feature #	User Story
32.0	156	3.4	As a user I want to visually read sensor data, so that it is possible to display it as single entities
Acceptance Criteria			Given sensor data And XT-view When XT-view is open Then sensor data is displayed in XT-view
Testing Information:			Values displayed in the test environment is equal to the values in InControl.
Struggles: InControl communication			
Testing date: 12.05.2020			Accepted

Test #	User Story #	Feature #	User Story
33.0	147	3.5	As a user I want to read historical sensor data from a graph so that i can monitor its behaviour
Acceptance Criteria:			Given sensor menu And available history data for the last 200 seconds When history button is pressed Then history view appears
Testing Information:			In the detailed view, when the history button is pressed a history graph opens, displaying values for the last 200 seconds, in accordance with the inControl values.
Struggles: InControl communication			
Testing date: 12.05.2020			Accepted

Test #	User Story #	Feature #	User Story
34.0	178	3.6	As a user I want to see a trend indicator to see if my values are decreasing or increasing. This will help me do a quick evaluation if the system is behaving as intended
Acceptance Criteria:			Given enough data necessary to calculate a trend is available When I press a sensor I want to see the trend displayed clearly and noticeable Then A trend indicator is displayed where I can do a quick glance at the indicator to see how my system is behaving
Testing Information:			Pressing a sensor opens detailed information on the sensor including an up/down arrow indicating whether the trend is currently rising or falling.
Struggles: InControl communication			
Testing date: 12.05.2020			Accepted

Test #	User Story #	Feature #	User Story
35.0	176	4.2	As a user I want to see the valves and sensors from the HCR displayed on a pipe that is configurable for different HCR's and XT's so that I can intuitively see how the HCR is performing
Acceptance Criteria:			Given a config file with component types, names, placement, and addresses When opening the HCR view through the hand menu Then an HCR diagram is drawn in the right order with corresponding values on components
Testing Information:			Pressing HCR button from the hand menu activates the HCR Schematic view where you see the schematic with components spawned
Struggles: Placing in right position due to differences in local and global position. Huge diagram with a lot of components			
Testing date: 13.05.2020			Accepted

10.3 Rejected tests

Test #	User Story #	Feature #	User Story
31.A	251	4.3	As a user I want to read information about the application so that I can get a better understanding of the different functionalities
Acceptance Criteria:			GIVEN a Hand Menu WHEN ReadMe button is pressed THEN Information about the app is displayed
Testing Information:			When Read Me is opened, it should be possible for the user to scroll down the read me information.
Struggles:			
<u>Struggle1:</u> Scroll does not work in the HL2 glasses but in Unity. Tried using and testing prefab from MRTK but this did not work either. Concluded that the prefab has a bug and chose to solve the user story without the scroll function.			
Testing date: 29.04.20			Rejected

10.4 Usability test

10.4.1 Information given to the participant before the test starts:

The application is made in collaboration with TechnipFMC and has the purpose of controlling valves and read sensor information of a subsea XT, using augmented reality. The test is done in the HoloLens2 glasses. Testing usability does not require pre-knowledge about XT's or augmented reality, but the app is made for people who have knowledge about the system already existing so having that experience is an advantage. Using HoloLens2 is new for most users and have some different/new features:

- Hand Menu – Hold up your hand to see the menu.
- Face plate – The “windows” is called face plate and they are both scalable and movable.

The application allows you to choose between two users. Admin and Guest. The only difference between the two user options is that in Admin mode you have the authority to open/close and adjust valves. To do operations and read information about a XT, a config is needed. For now, only the XT's Pyxis and Cassiopeia has this.

The usability test includes part A: tasks to complete and part B: a survey. It is important to say whatever you feel and think during the test and there are no wrong or rights. Everything you feel and think is valuable information, and no one is judging. Notes through the session will be taken and as a tester you have the rights to stop or take a break whenever you want.

10.4.1.1 Part A – Tasks:

1. Find “Read Me” and read it.
2. Choose User
3. Find the AVV valve.
4. Find the HCR.
5. Choose Admin.
6. Add a new XT
7. Edit the XT name.
8. Connect to Cassiopeia.
9. Connect to Pyxis again.
10. Read the trend of PTTT.
11. Adjust the choke valve.
12. Change the status of AVV valve.

10.4.1.2 Part B – Usability Survey:

Usability Test Template

Usability Survey – AWC in HL2	Strongly Disagree				Strongly agree
	1	2	3	4	5
1. I think that I would like to use this app frequently .					
2. I found the app unnecessarily complex .					
3. I thought the app was easy to use .					
4. I think that I would need the support of a technical person to be able to use this app.					
5. I found the various functions in this app were well integrated .					
6. I thought there was too much inconsistency in this app.					
7. I would imagine that most people would learn to use this app very quickly .					
8. I found the app very inefficient to use.					
9. I felt very confident using the app.					
10. I needed to learn a lot of things before I could get going with this app.					

To **improve** the app, I would **change the following 3 things**:

1. **Ex:** Add more gestures to for example close the view by swiping down
2. **Ex:** Make settings its own page, add a button.
3. **Ex:** Describe the graphs a bit more clearly

The following **3 things should not be changed** in the next iteration:

1. **Ex:** The trend simulation is very intuitive
2. **Ex:** I liked the ...
3. **Ex:** Good flow

10.4.2 Test 1, 08.05.2020, Participant 1:

1. Find “Read Me” and read it.

Had some trouble finding it as the user is not used to a hand menu. Notice that the depth is different. “Hard to see which tab you are reading, no visual indications, but because there is not too much text it is easy to understand”. “Use numbers on all tabs or none of the tabs”.

2. Choose User

Does not hesitate when choosing user. “I don’t have a password; I therefore choose Guest”.

3. Find ACC valve.

Immediately press connects. “Know that I’m not supposed to edit anything, pressing connect”. The face plate is too close, “Needs to be respawn in a more comfortable distance, not pressed up in the face”. First impression of contrasts is good. “Like the contrasts, but not sure about the valve colours and not sure if I like that the valves are floating on top of the pipes”.

The tester is familiar with the XT schematic and knows where the AVV is located. “The names of the valves and sensors are only visible when hovering over the valves/sensors. As a naive user I would have used time to search through all the valves to find the AVV. The names should be visible all the time”. “I like the way the name is displayed but would have liked to have it below the valve”. The user presses the right valve easily and gets busy with the pressure information right away.

4. Find the HCR

The user is looking at the XT view. Can’t see any HCR icons, concludes that the HCR must be at the Hand Menu. Did not use too much time to locate it, but the Hand Menu was not his first go-to. “The HCR has much more valves and sensors. Still need to work with depth and alignment”. “If I look at the schematic from 90 degrees, the names and valve positions is moved to the wrong pipe because of the alignment. Make sure to get everything lined up correctly”.

5. Choose Admin

As the tester is not naive and knows that the User can’t be changed during the session, he immediately use the AWC hand menu to close the application, and not through windows hand menu. Starts the application and launches the app. Choose Admin and types in the given PIN.

“I like that I don’t have to confirm the typed in PIN. Very user friendly and require one less button to reach his goal”. Questions whether that’s secure or not but likes it.

6. Add a new XT

See the add button right away, recognizes the icon. Tries to add a new XT by writing invalid inputs, did not work. Types in right information and add the new XT. Tries to add several XT’s. Discovers the known fact that it can’t fit more than four XTs. He deletes one XT before adding another to the list.

7. Edit the XT

Tries to enter wrong information. Not possible.

8. Connect to Cassiopeia.

Connects to Cassiopeia.

9. Connect to Pyxis.

Starts to get familiar with the application. Uses the Hand Menu to go back to the Manage XT view where you connect to the different trees. “Like the animation used to draw the XT schematic. Cool effect”. Notes that “The XT was spawned even closer to the face this time”.

10. Read the trend of PTTT.

Know where the sensor is located. The “flashlight effect” of the face plate is too much. Can’t read the information when pointing at the face plate, because the text is also white.

11. Adjust the choke valve

The slider is nice. Likes that you must confirm if you want to make a change. “Percentage status should also be visual outside in the XT view”.

12. Change the statues of the AVV valve

Did not notice the status text field when he looked at the valve for the first time, as he was too busy with looking at the pressure. “I like the layout”. Understands better this time why the pressure is showed as the value changes when open/close button is pressed and indicates that the valve status is changing and is either “opening or closing”.

Part B – Usability Survey:

Usability feedback – AWC in HL2	Strongly Disagree				Strongly agree
	1	2	3	4	5
1. I think that I would like to use this app frequently (more than existing system)				x	
2. I found the app unnecessarily complex .	x				
3. I thought the app was easy to use .					x
4. I think that I would need the support of a technical person to be able to use this app.	x				
5. I found the various functions in this app very well integrated .					x
6. I thought there was too much inconsistency in this app.				x	
7. I would imagine that most people would learn to use this app very quickly .					x
8. I found the app very inefficient to use.	x				
9. I felt very confident using the app.					x
10. I needed to learn a lot of things before I could get going with this app.	x				

To **improve** the app, I would **change the following 3 things**:

1. Give the valves and sensors name in XT-view
2. Improve the depth on the face plates and pop up face plates
3. Priority should be to create a completeness-feel, make it look the same throughout the app
4. Scratch the flashlight effect or dim it. Follow through the effects on the whole app.

The following **3 things should not be changed** in the next iteration:

1. Piping
2. Sensor icons.
3. The colour scheme (just dim the colours of the valves)

Short summary:

- Adjust the depth of the schematics
- Face plate distance to face
- Finalize the completeness of the application
- Svein-Aage's opinion is that the application is very user friendly and would rather use this than the existing system.

10.4.3 Test 2, 22.05.2020, Participant 2:

1. Find “Read Me” and read it.

Didn't quite understand and closes the application. The participant got some guidance to use the hand menu but struggled with the hand gestures and the placing of the “Read Me” face plate. The Read Me didn't work correctly so the participant didn't get to read it. This may have an impact on later tasks.

1. Choose User

“I'll have to choose guest since I don't have a password” And pressed “Connect” right after. Pushed the connect button because that was a natural choice.

2. Find the PMV valve

Found the PMV valve quickly. Added a comment “it was a bit small and hard to view from afar but when you moved closer it became clear”

3. Find the HCR

Tried to look for it in the view but needed a little reminder of the hand menu again to find it. “It is nice to have a hand menu, but it could have been bigger, it is a little difficult to press the correct buttons”

4. Log in as Admin

Finds the hand menu and closes the whole application. “Don't know if there's a “home” button here or not”.

5. Add a new XT

Finds the add button and types in an invalid port and IP address. “Nice to have error messages”

6. Edit XT

Finds the edit button without any trouble and edits the name. Saves it and edits it back to “Pyxis”. “Don't know if I like the keyboard, you really need to know where your hands are at all time, easy to press the wrong button” Looks like the participant presses underneath the keyboard.

7. Connect to Cassiopeia

Closes the app again, so now we inform him about the “Manager” button on the hand menu. When opening the application again “the play sign only appears if you point to it, shouldn't it be visible all the time?”

8. Connect to Pyxis.

Getting the hang of connecting now very quickly.

9. Read trend from PTTT

Finds the PTTT and reads the values and trend. “It’s a very small face plate, hard to push the buttons”

10. Adjust the choke

First instinct now is to look in the hand menu, but after some guidance on what the choke is and its name, it’s located and pressed. “Like the faceplate’s size and slider. The security pop-up is good to have too, because the slider is very sensitive.”

11. (Changed to) Open PMV

Finds the valve and opens it, presses “Open” and presses “Yes” in the security face plate. He notices the status change in the left corner and the valve colour change

Usability Survey – AWC in HL2	Strongly Disagree				Strongly agree
	1	2	3	4	5
1. I think that I would like to use this app frequently.					x
2. I found the app unnecessarily complex.				x*	
3. I thought the app was easy to use.				x*	
4. I think that I would need the support of a technical person to be able to use this app.	x*				
5. I found the various functions in this app were well integrated.				x	
6. I thought there was too much inconsistency in this app.				x	
7. I would imagine that most people would learn to use this app very quickly.					x
8. I found the app very inefficient to use.					x
9. I felt very confident using the app.					x
10. I needed to learn a lot of things before I could get going with this app.				x	

To **improve** the app, I would **change the following 3 things**:

1. Larger x button on sensor menu
2. Larger menu menus on sensors
3. More intuitive “Home” button
4. Thought the colours were a little transparent and bland, liked the strong colours on the injection pipes.

The following **3 things should not be changed** in the next iteration:

0. Don’t make anything smaller
 1. Admin/Guest
 2. Choose XT, add, edit – not a fan of the keyboard
 3. XT view – good with the colour change on the valves

* Added a grade down because he felt that he wasn’t used to operate the HL2, and because of that he felt a bit restricted.

Short summary:

- Change icons on the hand menu to improve an intuitional user experience
- Scale up the sensor menus, so that it’s easier to press the correct buttons
- Overall the participant felt that with a little experience using AR, it would be much easier to operate our system.



11 APPENDIX C – DECISIONS

Abstract

Application decisions and decisions for removed features.

11.1 Schematic Decisions

To decide how we were supposed to draw the schematic of the XT and the HCR a few different options were explored, in this chapter we will explain a few different approaches and why we decided to use the final version.

11.1.1 Procedural drawing of Christmas tree schematic

One of the requirements that we got from TechnipFMC was that they wanted the XT schematic to be procedural/automatically drawn from an external configuration file. This problem was worked on from two different angles. This angle of the solution is about how research was done to automatically draw the HMI from a json file, although work on this solution was scrapped before json file was ever implemented.

One of the solutions for building a dynamic HMI is to make the pipes and valves drawn automatically/procedural from a json configuration file. This is one of two solutions that was explored for the application. The concept of this method has inspiration from a maze builder in 2D. The solution started with making a 2D array.

The idea behind making a 2D array is that we need to have access to four different directions (-x), (x), (-z) and (z) direction to be able to populate the HMI with pipes. Depth will not be taken into consideration for this solution so there will be no Y – axis for this HMI solution.

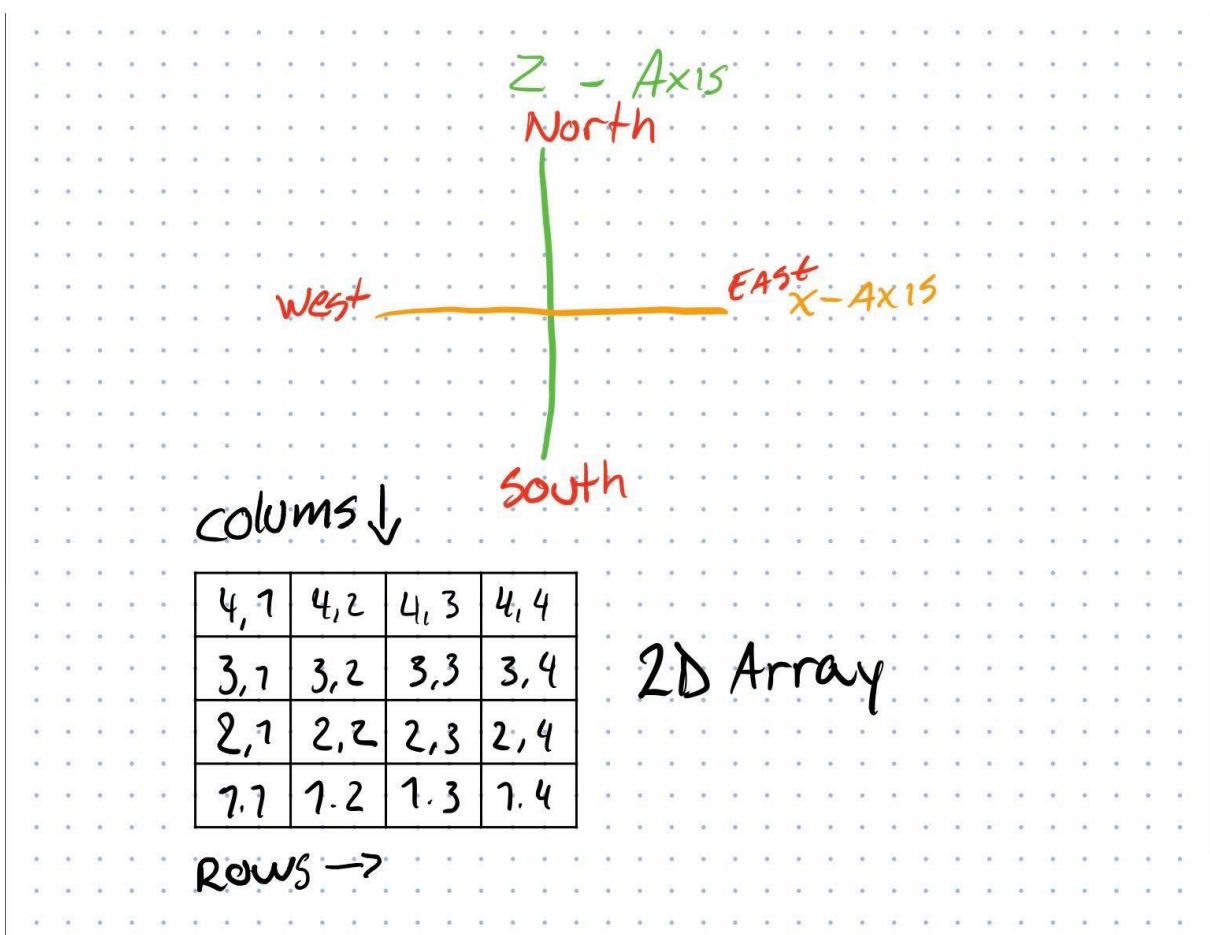


Figure 142 - 2D Array Coordinates

With a 2D grid we can now look at different ways to populate the grid. The first steps that was taken was to do research to find a good solution to do this with unity. Reason for this is that we want to populate the grid with Unity GameObjects which can hold up to multiple scripts with different functionality. The concept of populating a 2D array with numbers contra with GameObjects will in theory work. The grid system will give our GameObjects their respectfully position where the grid is spawned, so It does not rely on being spawned in 0,0,0 in the world, but instead it gets a offset from set spawn location. Grid (1,1) is spawned +1 in x and +1 in z from where the grid system is spawned in unity

First the solution started with generating a dynamic HMI from a simple .txt file. The text file was populated with 0, 1 and 2's to differentiate between three different game objects. The algorithm will differentiate between spaces and new lines to be able to populate in the four directions that the grid has set up.

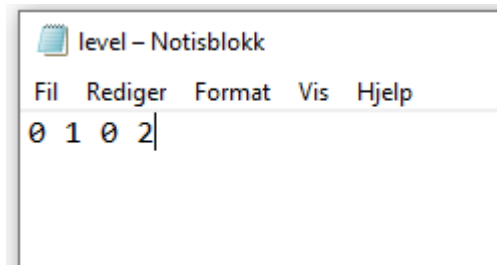


Figure 143 - how a simple lvl design can be drawn

To be able to connect pipes coming from different directions a connection system is needed from Figure 144 we can see the thought process behind this solution. This is needed to be able to specify where the pipe should connect or branch off. Inspiration for this solution is from a simple maze game made by unity. By implementing a method like this, pipes can be connected by saying “connect to pipe that is north (+1 in z axis)” from the desired location in the 2D array. We can do the same for the three other directions which we call west (-1 x axis), east (+1 x axis) and south (-1 z axis).

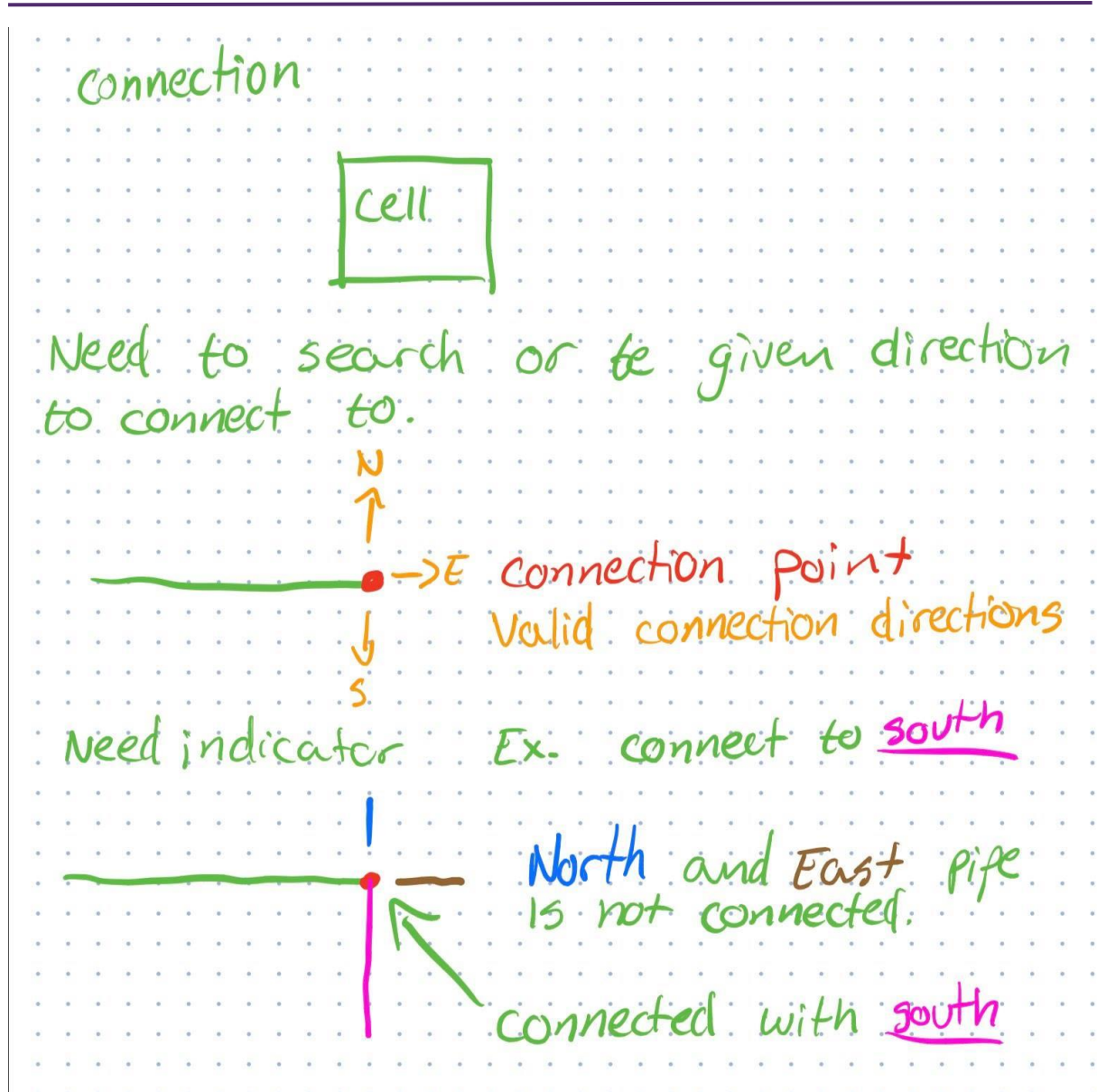


Figure 144 - Connection Intersection

One smart feature that would be nice to implement is the spawning of one large game object instead of ten smaller one. To do this a search algorithm who searches for next neighbour in same direction is thought of to be used. The algorithm would have a start point and search for neighbours of the same type (ex. +1 in x axis). When the algorithm stops finding neighbours in the corresponding axis, then it would take how many steps taken, spawn a game object of the type and multiply the given axis with the number of steps. We can see from Figure 145 that instead of having five game objects the algorithm will spawn in one game object and set its width to original width of a cell multiplied by how many steps. The new width of the game object will be 5

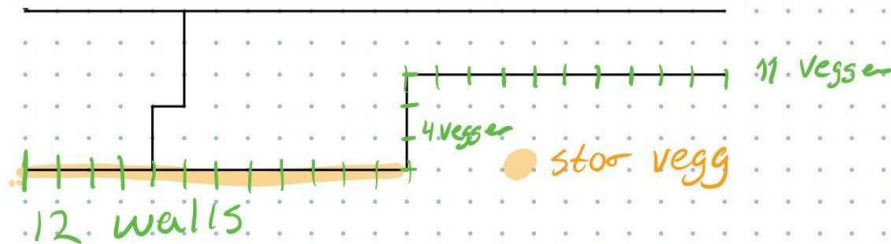
Make function to draw pipes between 2 coordinates.

Using:



Viscosity

Exs:



Estimerer 1M Per blokk

Lag liste med hvor mange vegger

Antall vegger x Lengde vegg = stor vegg

Da får vi 12M lang vegg

Algoritme som sjekker nabo

BFS? Må få samlet alle på samme nivå

Figure 145 - thought process when making procedural HMI

After working on this solution for a couple of days the team sat down and scrapped this solution for implementing the HMI. With more knowledge on how to solve the problem team members had found out that the pipes could be statically drawn within unity itself. This would mean that populating the HMI with valves and sensors could be easily placed in unity and by writing a save script the positions of every game object placed could be generated in a .json file that was almost finished.

11.1.2 SplineMesh Asset

Splinesh is an asset free to download in the Unity Asset store. The concept of splinesh is to use Bezier curves to draw out different curved contents in real time [42].

The splinesh uses nodes to draw shapes between them. The idea was to place nodes throughout the scene and then draw pipes between the nodes. A good amount of time was used learning how the asset worked and exploring with prototypes. In the end it was decided that the procedural drawing of pipes was too complex for its use.

11.1.3 Why we chose static pipes with json file configuration

The pipes in the HCR and XT is static it is therefore no use in drawing it every time the application is started. This would only cost more recourses developing, implementing and running. We therefore decided to go with the solution explained in XT Schematic

11.2 Sensor Considerations

- Sensor are divided into three sizes because of how sensor data is delivered from the simulator.
- Active Values Small and its size counter parts are named because of their sizes and because their values are actively changing and should be monitored.
 - PT and FM sensors takes one data address.
 - PTTT, ASD, EP takes 2 data addresses.
 - MPFM takes 5 data addresses.
- Their measure types are not the same, which is what prompted this sensor system.
- The amount of variables In the three main scripts, SensorManager, DetailedInfo and History could have been reduced if performance wasn't an issue, but the sensor part of the application displays data every second, and so by preparing the scripts for this process we can reduce the amount of calls necessary and thereby increase performance.
- The measure type is determined by the name, this is currently hardcoded, if new sensors with new names are added, which are not indigenous to the old names this function will become deprecated.
- Data values are parsed into floats to be manipulated into 2 decimal floats and for creating the graph.
- Update () is used on detailed info because only numbers and trends are shown every second. [43]
- FixedUpdate () is used on History because a visual graph is presented and shown to the user and a fixed fps rate is especially preferable to an unstable one. [44]

11.2.1 History

There are many open sourced examples of graphs already created for in Unity and C#, the ones that were applicable to our whole application were not free and there would be no way of telling how HoloLens 2 would handle these rather impressive creations at a update rate every second.

A free one was attempted but it was quickly scrapped because of its performance and how difficult it was to implement into the application properly.

By creating the graphs our self we had full control over the presentation, the performance and its features, allowing us to make an optimised graph view with just the functions we needed and the realization that the amount of work would have been roughly the same: synchronization, positioning, sizing, graphs, changing the graph, and automatically adjusting the data inputted and its alphabetical representations.

11.3 Decision on where to place the hand menu

With little to no tutorials on how to implement a hand menu in HL2. Much of the implementation of the hand menu is left up to how our users wants the hand menu to work and function. By examining and playing with MRTK's example scenes implementation can start with a requirement in mind.

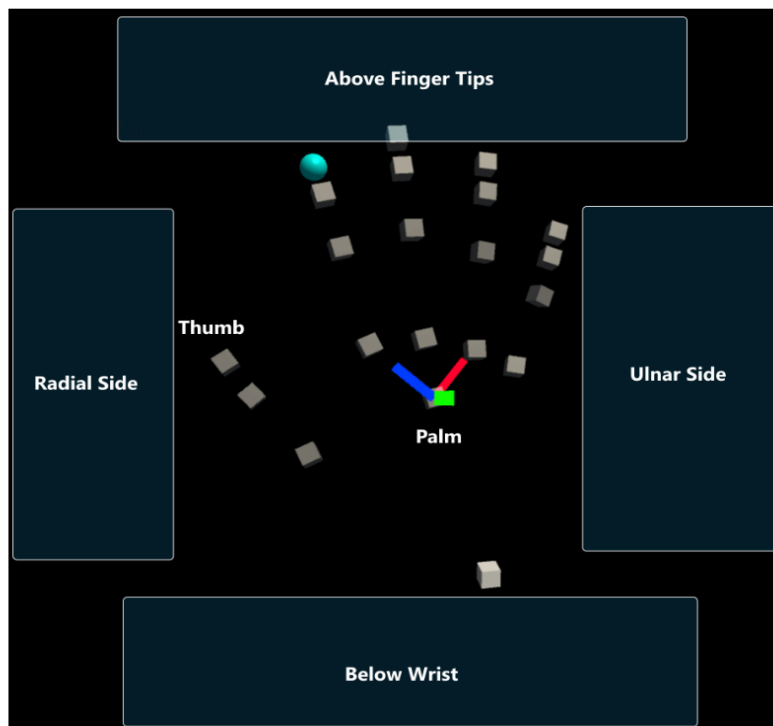


Figure 146 - left hand showing "zones" for hand menu position

A requirement given is that the application should be more user-friendly than existing systems. The team figured out that the hand menu would be best appropriate as a tool to navigate within the application. The menu should be an easy and practical addition for the application, and not something that will clutter your view or interrupt workflow. To understand h

From Figure 151 we can see four different positions suitable for the menu. Microsoft recommends building the hand menu in a "safe zone". The safe zone is on the ulnar side of the left hand. This is the side that is closest to opposite shoulder. By having it on ulnar side Both hands will not intervene with each other and hands crossing each other will not be a problem. If hands were to cross the HL2 will have problems detecting the shape of the hand and the hand menu will feel unresponsive or not even work.

11.4 Hashing algorithm: BCrypt

The BCrypt algorithm was chosen to hash the pin code. This algorithm uses salt to protect against rainbow table attacks, has slow construction and is resistant to brute force attacks. The hashing algorithm is considered to be one of the most secure ones compared to among others like MD5 and SHA1/SHA 256. [45] [46]



12 APPENDIX D - RISK DOCUMENTATION

Abstract

This document includes all risk found throughout the project and a response plan. There are several methods that can be used when responding to a risk, but we are mainly going to use mitigation. Mitigate the probability of occurrence and/or the consequence of it.

13.1 Risk Template

Risk analysis template – table:

Risk id:

Likelihood:	Consequence:	Rank:
--------------------	---------------------	--------------

Description of risk:	
Cause:	
Description of Consequence:	
Tag:	

Risk response strategy:	
Response:	

Risk id: Id of the risk. All risk that is identified starts with R and then the id number. Ex. R – 000.

Likelihood: Given rank on the likelihood scale.

Consequence: Given rank on the consequence rank.

Rank: Likelihood rank multiplied with Consequence rank = Rank.

Description of risk:

Cause: Root cause of the risk.

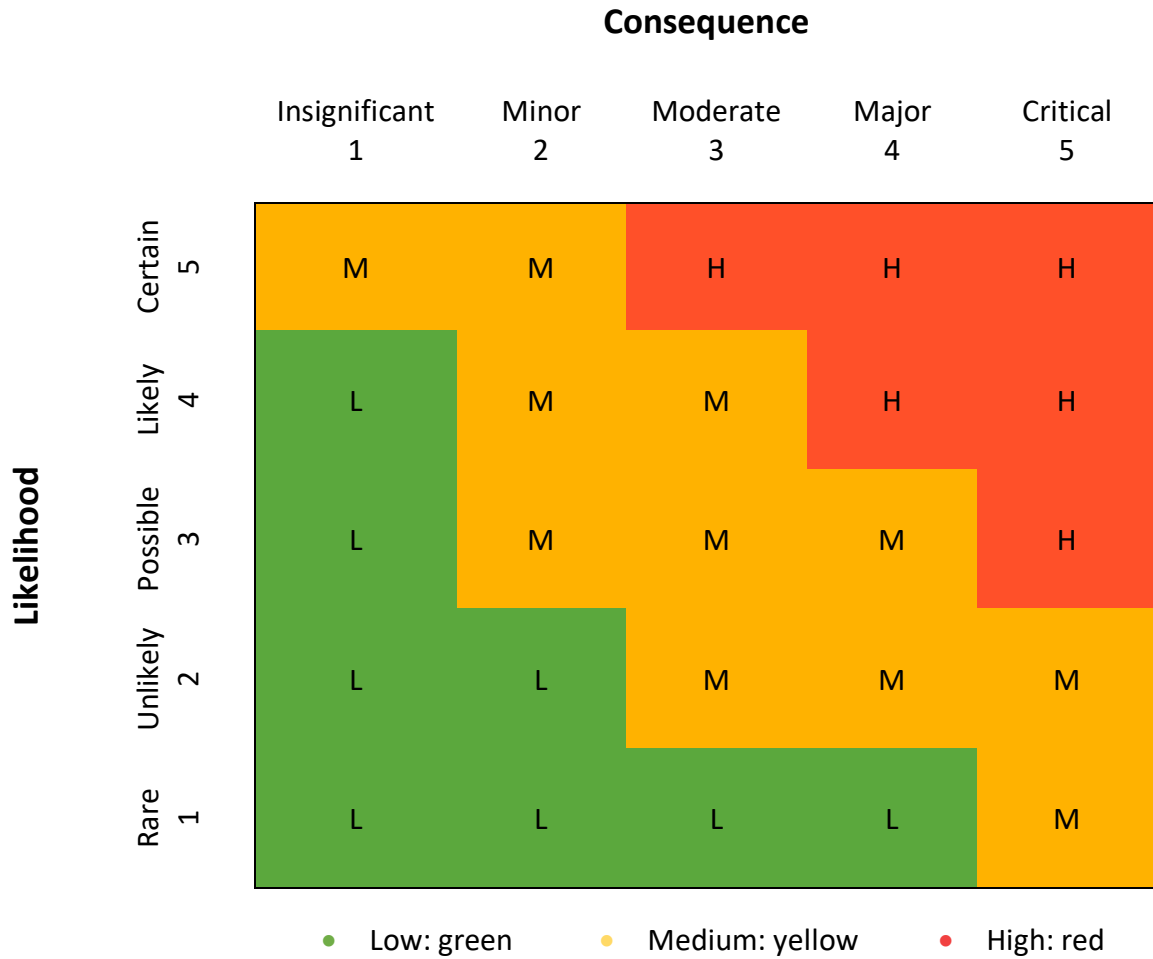
Description of Consequence: Outcome if the risk occurs.

Tag: If the risk is Technical or Non – Technical.

Risk response strategy: How we plan to respond to the risk, based on the consequence.

Response: Which of the four methods of response strategy is used.

13.2 Risk Evaluation – Matrix:



Priority	Description
High	Immediate action is required
Medium	Action required
Low	Action not required

Monitor and review risks template - table

Attached at the end of each sprint.

Risk that needs to be reviewed:

Id	Description	NEW Response plan	Implementation status

List of closed risks:

Id:	Description:	Comment:

Risk Register template – table

Priority register of risks. One is made for technical and one for non – technical risks.

High

Id	Description	Mitigation plan	Implementation status

Medium

Id	Description	Mitigation plan	Implementation status

Low

Id	Description	Mitigation plan	Implementation status

13.3 Sprint 1

Risk identification – brainstorming, history, critical functionalities of the system

- Bad communication
- Takes discussions personally
- Does not meet at time
- Lack of expertise
- Cannot be present at important date because of illness or personal reasons
- Loss of team members
- Internal conflicts between members of the team or stakeholders
- Delete content by accident
- Our goal for the project is not reached
- Lack of authority
- Unidentified risks
- Scope Creep
- User stories not formulated correctly
- Members does not take areas of responsibility
- Wrong estimation of time
- Does not have access to each other's work one of the members cannot be reached
- HoloLens 2 does not arrive throughout the project period
- TechnipFMC cannot provide us with the tool we need to manage the task
- TechnipFMC does not have the time to give feedback/answers when we need it
- Bad usage of version control
- Storage limitations in GIT
- Git Upload/download
- Bad gitignore file
- Faulty usage of GIT/Merge issues
- Ram limitations in HoloLens2
- Don't use the tools we have agreed on to use
- Does not
- Don't use the tools we have agreed on
- Connection to the API
- Use different versions of the tools
- Auto updates on tools
- Backup
- Bad software architecture
- Not strong enough computers
- Changes in the API
- Use the wrong tools
- Keep deadlines
- Does not follow chosen project model
- A member of the group quits/cannot continue to work on the project because of personal problems
- Unity has an update and the current version is not frozen

Risk id: R-001

Likelihood: 5	Consequence: 3	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Lack of expertise.
Cause:	Not enough experience or knowledge within the different fields that we touch through the project period.
Description of Consequence:	Too much time is spent on researching the same topics.
Tag:	Non – technical.

Risk response strategy:	Through the project plan, find and map the different areas of expertise and distribute responsibilities where each member of the team must take a lead and learn about this specific area.
Response:	Mitigation: Consequence.

Risk id: R-002

Likelihood: 2	Consequence: 4	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Lack of communication.
Cause:	Not enough questions are asked and/or not the right ones. Poor discussions.
Description of Consequence:	Team members uses their time on the wrong/not prioritized tasks. Misunderstandings of what needs to be done.
Tag:	Non – technical.

Risk response strategy:	Make a list of communication channels and their purpose. Read the meeting notes and after the next day stand up meeting, some time is set aside for discussion. Sprint review, planning poker and daily stand up meetings will help us mitigate communication problems. Sprint retrospective will give specific feedback on communication.
Response:	Mitigation: Likelihood, Consequence.

Risk id: R-003

Likelihood: 4	Consequence: 3	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	The group is unable to agree on decisions.
Cause:	Different opinions caused by different points of view.
Description of Consequence:	Cannot Cannot continue to work within discussion related topics of the project. Reduced mood and work moral within the group.
Tag:	Non – technical.

Risk response strategy:	Hold an objective discussion and keep it on topic. If no decision is made, write it down and go to supervisors for guidance. Find the root cause of each other's opinions, why the member thinks like he/she does. In this way the group can identify possible misunderstandings.
Response:	Mitigation: Consequence.

Risk id: R-004

Likelihood: 1	Consequence: 5	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	HoloLens 2 does not arrive in time to allow for development and testing.
Cause:	Technical errors from Microsoft. Transport or toll problems.
Description of Consequence:	Project description needs to be rewritten. Cannot make something based on technology we don't have access to.
Tag:	Technical.

Risk response strategy:	Have close communication with TechnipFMC and ask for weekly updates about status. Use HoloLens1.
Response:	Accept: Active

Risk id: R-005

Likelihood: 4	Consequence: 3	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Unidentified risks.
Cause:	Unpredictable risks. For example, weather or death in family.
Description of Consequence:	Hold up in the project. Don't know how to manage risks when they occur.
Tag:	Non – technical.

Risk response strategy:	Set aside time for identifying risks at the beginning of each sprint. Use different methods to identify risks. Be alert. Keep up with newspapers.
Response:	Acceptance: Passive, Mitigation: Likelihood.

Risk id: R-006

Likelihood: 1	Consequence: 3	Rank: Low
----------------------	-----------------------	------------------

Description of risk:	A team member can no longer take part in the project.
Cause:	Illness or personal reasons.
Description of Consequence:	Team members must use time to understand and learn the work of the lost member. Work is not uploaded and available for all members.
Tag:	Non – technical.

Risk response strategy:	Use stand-up meetings to inform each other on what you are currently working on. Upload everything regularly at an agreed platform so that all members can pick up on each other's work.
Response:	Mitigation: Consequence

Risk id: R-007

Likelihood: 1	Consequence: 5	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Loss of work related to the project.
Cause:	Malware, theft, files stored local and not in cloud, accidental.
Description of Consequence:	Work that has already been done must be done again. This is time-consuming and inefficient.
Tag:	Technical

Risk response strategy:	<p>Always work “online”.</p> <p>Save all important and related files at agreed platform.</p> <p>Make sure of that only a limited group of authorized people have access to your working space.</p>
Response:	Mitigation: Likelihood, Consequence

Risk id: R-008

Likelihood: 5	Consequence: 2	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Incorrect estimation of time required to complete a user story.
Cause:	Not enough experience.
Description of Consequence:	<p>Acceptance criteria cannot be tested at the end of the sprint.</p> <p>Hard to measure progress and to know when something is finished.</p>
Tag:	Non – technical.

Risk response strategy:	<p>Make smaller tasks that is easier for the development team to estimate and relate to, which makes it easier to measure progress as well.</p> <p>Use sprint retrospective to review and discuss what went wrong.</p>
Response:	Mitigation: Likelihood, Consequence

Risk id: R-009

Likelihood: 3	Consequence: 3	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Not able to meet the client's requirements
Cause:	Incorrect prioritization of user stories and tasks.
Description of Consequence:	Does not manage to meet the stakeholder's requirements by doing "nice to have" before "need to have". Affects the time schedule.
Tag:	Non – technical.

Risk response strategy:	Use daily stand up meetings to make sure that everyone is working on the right thing. Close communication with TechnipFMC. Before each sprint, the Product Owner prioritizes relevant user stories and tasks. The group then discuss if they agree or not.
Response:	Mitigation: Likelihood, Consequence.

Risk id: R-010

Likelihood: 2	Consequence: 4	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Internal conflicts between members of the team.
Cause:	Disagreement on decisions. Lack of respect. Bad communication.
Description of Consequence:	Reduced work moral on the team. End of friendship and/or members quit.
Tag:	Non – technical.

Risk response strategy:	Write a group contract that includes guidelines about how we want to work together as a group.
Response:	Mitigation: Likelihood.

Risk id: R-011

Likelihood: 1	Consequence: 2	Rank: Low
----------------------	-----------------------	------------------

Description of risk:	Group members does not meet at agreed time, or not at all.
Cause:	Commute to work. Low self-discipline.
Description of consequence:	Members may not be included in important decisions.
Tag:	Non – technical.

Risk response strategy:	Make a group member contract including a consequence for members that are late. Have stand-up meetings over slack or teams. Make an agreement for workplace that suits everyone.
Response:	Mitigation: Likelihood, Consequence

Risk id: R-012

Likelihood: 2	Consequence: 4	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	TechnipFMC fails to provide the development team with the tools needed to manage the project.
Cause:	Hard to reach the responsible persons. Their software is not ready for use.
Description of consequence:	Cannot validate output data is correct. Does not get the datapoints simulated to extract data.
Tag:	Technical, Non – Technical.

Risk response strategy:	Have a close dialogue with the TechnipFMC from the beginning of the project to make sure of that everything needed is in place for the project.
Response:	Mitigation: Likelihood.

Risk id: R-013

Likelihood: 1	Consequence: 4	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	TechnipFMC does not have the time to give feedback/answers on important decisions.
Cause:	Hard to reach because they are too busy.
Description of consequence:	Decisions must be our self that we may not have time to change when first implemented.
Tag:	Non – technical.

Risk response strategy:	Schedule time to discuss decisions and progress. Ask stakeholders about specific dates and time they are not available. Use sprint review for feedback.
Response:	Mitigation: Likelihood, Consequence.

Risk id: R-014

Likelihood: 2	Consequence: 2	Rank: Low
----------------------	-----------------------	------------------

Description of risk:	Members of the team fail to uphold their responsibilities in their areas of expertise.
Cause	Too difficult. Members are not interested in assigned areas of expertise.
Description of consequence:	Other members must take responsibility.
Tag:	Non – technical.

Risk response strategy:	Plan a meeting at the beginning of the project where each member can express their thoughts and discuss what areas of expertise they want to cover. Use sprint retrospective to decide if
Response:	Mitigation: Likelihood

Risk id: R-015

Likelihood: 2	Consequence: 5	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	HoloLens 2 does not have adequate performance for running the application.
Cause:	Inefficient code Lack of processor power.
Description of consequence:	A less enjoyable user experience.
Tag:	Technical.

Risk response strategy:	Use premade HoloLens2 standards. Read documentation about the HoloLens2 before making the application. Write clean code. Prioritize what is needed to be included of features.
Response:	Acceptance and Mitigation: Likelihood, Consequence

Risk id: R-016

Likelihood: 1	Consequence: 2	Rank: Low
----------------------	-----------------------	------------------

Description of risk:	The team members computers performance is too weak to run the application developed.
Cause:	Weak GPU or CPU
Description of consequence:	Cannot run application whenever a team member wants to.
Tag:	Technical

Risk response strategy:	Use computers provided by TechnipFMC if possible or buy new.
Response:	Mitigation: Consequence.

Risk id: R-017

Likelihood: 4	Consequence: 3	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Software architecture is badly structured.
Cause:	Not enough knowledge.
Description of consequence:	Starts programming before we know how the different parts of the system communicate, both internally and externally causing that parts of coding must be done all over again.
Tag:	Technical.

Risk response strategy:	Use resources provided by TechnipFMC and USN.
Response:	Mitigation: Likelihood, Consequence.

Risk id: R-018

Likelihood: 4	Consequence: 5	Rank: High
----------------------	-----------------------	-------------------

Description of risk:	Members use different versions of software.
Cause:	Bad communication.
Description of consequence:	When merging error occurs. Tasks must be done over again.
Tag:	Technical.

Risk response strategy:	Make a list of agreed tools and versions at the beginning of the project.
Response:	Mitigation: Likelihood, Consequence

Risk id: R-019

Likelihood: 3	Consequence: 3	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Scope Creep.
Cause:	Scope not properly defined, documented, or controlled.
Description of consequence:	Continuous or/and uncontrolled growth in the project.
Tag:	Non – technical.

Risk response strategy:	Define the scope early in collaboration with TechnipFMC. Make a project plan.
Response:	Mitigation: Likelihood.

Risk id: R-020

Likelihood: 4	Consequence: 4	Rank: High
----------------------	-----------------------	-------------------

Description of risk:	User stories not correctly defined with acceptance criteria.
Cause:	Lack of experience.
Description of consequence:	Must re-define user stories. Cannot test the user stories by their acceptance criteria (accept/reject).
Tag:	Non - Technical

Risk response strategy:	Talk to supervisors to get feedback before we move on with tasks.
Response:	Mitigation

Risk id: R-021

Likelihood: 5	Consequence: 2	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Faulty usage of GIT/Merge issues.
Cause:	Errors in committed code, faulty gitignore file.
Description of consequence:	Nonrunning code, memory issues and or slow upload/download.
Tag:	Technical

Risk response strategy:	Do not commit anything that contains errors causing the application to not be runnable. Ensure the gitignore file includes relevant files. Ensure committed code is compatible if it must match other code.
Response:	Mitigation

Risk id: R-022

Likelihood: 2	Consequence: 3	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Team members does not follow the chosen process model correctly.
Cause	Lack of knowledge and discipline.
Description of consequence:	Start working in a waterfall model which does not give time or space for changes in design.
Tag:	Non – Technical.

Risk response strategy:	Decide one platform where it is easy to follow the process model and progress. Use sprint retrospective and daily meetings.
Response:	Mitigation: Likelihood.

Risk that needs to be reviewed:

Id	Description	NEW Response plan	Implementation status
R-014	Members of the team fail to uphold their responsibilities in their areas of expertise.	Use sprint retrospective to decide new areas of responsibilities.	Done

List of closed/eliminated risks

ID:	Description:	Comment:
R-004	HoloLens 2 does not arrive in time to allow time for development and testing.	HoloLens2 has arrived.
R-018	Members use different versions of software.	Make a list of agreed tools and versions

13.4 Sprint 2

Risk identification - brainstorming

- Loss of memory

Risk id: R-023

Likelihood: 2	Consequence: 3	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Lag in application.
Cause:	Garbage collection. Not deleted generated game objects.
Description of Consequence:	Bad user experience.
Tag:	Technical.

Risk response strategy:	Pre-defined set of rules to ensure clean code.
Response:	Mitigation: Likelihood.

Risk that needs to be reviewed:

Id	Description	NEW response plan	Implementation status

List of closed/eliminated risks

ID:	Description:	Comment:
	.	

13.5 Sprint 3

Risk identification - brainstorming

- Corona virus
- Changes in the API

Risk id: R-024

Likelihood: 2	Consequence: 5	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Infection of coronavirus.
Cause:	Illness.
Description of Consequence:	Postponing of bachelor thesis.
Tag:	Non - Technical

Risk response strategy:	Wash hands often. Stay away from sick people. Avoid travelling.
Response:	Mitigation: Likelihood

Risk id: R-025

Likelihood: 5	Consequence: 4	Rank: High
----------------------	-----------------------	-------------------

Description of risk:	Changes in the API.
Cause:	Version upgrade.
Description of Consequence:	Does not have time to familiarise with the new API.
Tag:	Technical

Risk response strategy:	Close dialogue with TechnipFMC.
Response:	Accept: Passive

Risk that needs to be reviewed:

Id	Description	NEW response plan	Implementation status

List of closed/eliminated risks

ID:	Description:	Comment:

13.6 Sprint 4

Risk identification – brainstorming

- Third presentation

Risk id: R-026

Likelihood: 5	Consequence: 2	Rank: Medium
----------------------	-----------------------	---------------------

Description of risk:	Visual presentation of our final product.
Cause:	Corona virus (covid-19).
Description of Consequence:	Cannot show our final product in the HL2 as planned when having our third presentation.
Tag:	Non-Technical.

Risk response strategy:	Perform demonstration to stakeholder in smaller groups before presentation day - if possible. Make a video, similar to second presentation.
Response:	Accept: Active

Risk that needs to be reviewed:

Id	Description	NEW response plan	Implementation status

List of closed/eliminated risks

ID:	Description:	Comment:

13.7 Sprint 5

Risk identification – brainstorming

- Limitations in Unity.

Risk id: R-027

Likelihood: 2	Consequence: 2	Rank: 4
----------------------	-----------------------	----------------

Description of risk:	Limitations in Unity.
Cause:	Tools in Unity not fulfilling our demands.
Description of Consequence:	Must solve our problem in another way.
Tag:	Technical.

Risk response strategy:	Remove the risk by dropping that particular feature in Unity that has limitations.
Response:	Accept: Passive.

Risk that needs to be reviewed:

Id	Description	NEW response plan	Implementation status

List of closed/eliminated risks

ID:	Description:	Comment:

13.8 Sprint 6

Risk that needs to be reviewed:

Id	Description	NEW response plan	Implementation status
R-021	Faulty usage of GIT/Merge issues	Make a new GIT repository Accept and make a .unitypacket that includes all files from the different projects	Done

List of closed/eliminated risks

ID:	Description:	Comment:

13.9 Sprint 7

Risk that needs to be reviewed:

Id	Description	NEW response plan	Implementation status

List of closed/eliminated risks

ID:	Description:	Comment:

13.10 Risk Register

Technical risks

High

Id	Description	Mitigation plan	Implementation status
R-018	Members use different versions of software.	Make a list of agreed tools and versions.	Done.
R-025	Changes in the API.	Close dialogue with TechnipFMC.	Done.

Medium

Id	Description	Mitigation plan	Implementation status
R-004	HoloLens 2 does not arrive in time to allow time for development and testing.	Have close communication with stakeholders and ask for weekly updates about status. Use HoloLens1	Done.
R-007	Loss of work related to the project.	Always work "online". Save all important and related files at agreed platform Make sure of that only a limited group of authorized people have access to your working space.	Done.
R-012	TechnipFMC fails to provide the development team with the tools needed to manage the project.	Have a close dialogue with the TechnipFMC from the beginning of the project to make sure of that everything needed is in place for the project success.	Done.

R-015	HoloLens 2 does not have adequate performance for running the application.	<p>Use premade HoloLens2 standards.</p> <p>Read documentation about the HoloLens2 before making the application.</p> <p>Write clean code.</p> <p>Prioritize what is needed to be included of features.</p>	Done.
R-017	Software architecture is badly structured.	Use resources provided by TechnipFMC and USN.	Done.
R-021	Faulty usage of GIT/Merge issues.	<p>Do not commit anything that contains errors causing the application to not be runnable.</p> <p>Ensure the gitignore file includes relevant files.</p> <p>Ensure committed code is compatible if it must match other code.</p>	Done.
R-023	Lag in application.	<p>Garbage collection.</p> <p>Not deleted generated game objects.</p>	Done.

Low

Id	Description	Mitigation plan	Implementation status
R-016	The team members computers performance is too weak to run the application developed.	Use computers provided by TechnipFMC if possible or buy new.	Not started.
R-027	Limitations in Unity.	Remove the risk by dropping that particular feature in Unity that has limitations.	Done

Non – Technical risks**High**

Id	Description	Mitigation plan	Implementation status
R-020	User stories not correctly defined with acceptance criteria	Define acceptance criteria and test user story	Done.

Medium

Id	Description	Mitigation plan	Implementation status
R-001	Lack of expertise.	Through the project plan, find and map the different areas of expertise and distribute responsibilities where each member of the team must take a lead and learn about this specific area.	Done.
R-002	Lack of communication	<p>Make a list of communication channels and their purpose.</p> <p>Read meeting notes and after the next stand up meeting, some time is set aside for discussion.</p> <p>Sprint review, planning poker and daily stand up meetings will help us mitigate communication problems. Sprint retrospective will give specific feedback on communication.</p>	Done.
R-003	The group is unable to agree on decisions.	<p>Hold an objective discussion and keep it on topic. If no decision is made, write it down and go to supervisors for guidance.</p> <p>Find the root cause of each other's opinions, why the member thinks like he/she does. In this way the group can identify possible misunderstandings.</p>	Done.

R-005	Unidentified risks.	<p>Set aside time for identifying risks at the beginning of each sprint.</p> <p>Use different methods to identify risks.</p> <p>Be alert. Keep up with newspapers.</p>	Done.
R-008	Incorrect estimation of time required to complete a task.	<p>Make smaller tasks that is easier for the development team to estimate and relate to, which makes it easier to measure progress.</p> <p>Use sprint retrospective to review and discuss what that went wrong.</p>	Done.
R-009	Not able to meet the customers' requirements.	<p>Use daily stand up meetings to make sure that everyone is working on the right thing.</p> <p>Close communication with TechnipFMC.</p> <p>Before each sprint, the Product Owner prioritize relevant user stories and tasks. The group then discuss if they agree or not.</p>	Done.
R-013	TechnipFMC does not have the time to give feedback/answers on important decisions.	<p>Schedule time to discuss decisions and progress.</p> <p>Ask stakeholders about specific dates and time they are not available.</p> <p>Use sprint review for feedback.</p>	Done.
R-014	Members of the team fail to uphold their responsibilities in their areas of expertise.	<p>Plan a meeting at the beginning of the project where each member can express their thoughts and discuss what areas of expertise they want to cover.</p> <p>Use sprint retrospective.</p>	Done.
R-019	Scope Creep.	<p>Define the scope early in collaboration with TechnipFMC.</p> <p>Make a project plan.</p>	Done.

R-022	Team members does not follow the chosen process model correctly.	Decide one platform where it is easy to follow the process model and progress. Use sprint retrospective daily meetings.	Done.
R-024	Infection of coronavirus.	Wash hands often. Stay away from sick people. Avoid travel.	Done.
R-026	Visual presentation of our final product.	Perform demonstration to stakeholder in smaller groups before presentation day - if possible. Make a video, similar to second presentation.	Done.

Low

Id	Description	Mitigation plan	Implementation status
R-006	A team member can no longer take part in the project.	Use stand-up meetings to inform each other on what you are currently working on. Upload everything regularly at an agreed platform so that all members can pick up on each other's work.	Done
R-010	Internal conflicts between members of the team.	Write a group contract that includes guidelines about how we want to work together as a group.	Done.
R-011	Group members does not meet at agreed time, or not at all.	Make a group member contract including a consequence for members that are late. Have stand-up meetings over slack or teams. Make an agreement for workplace that suits everyone.	Done



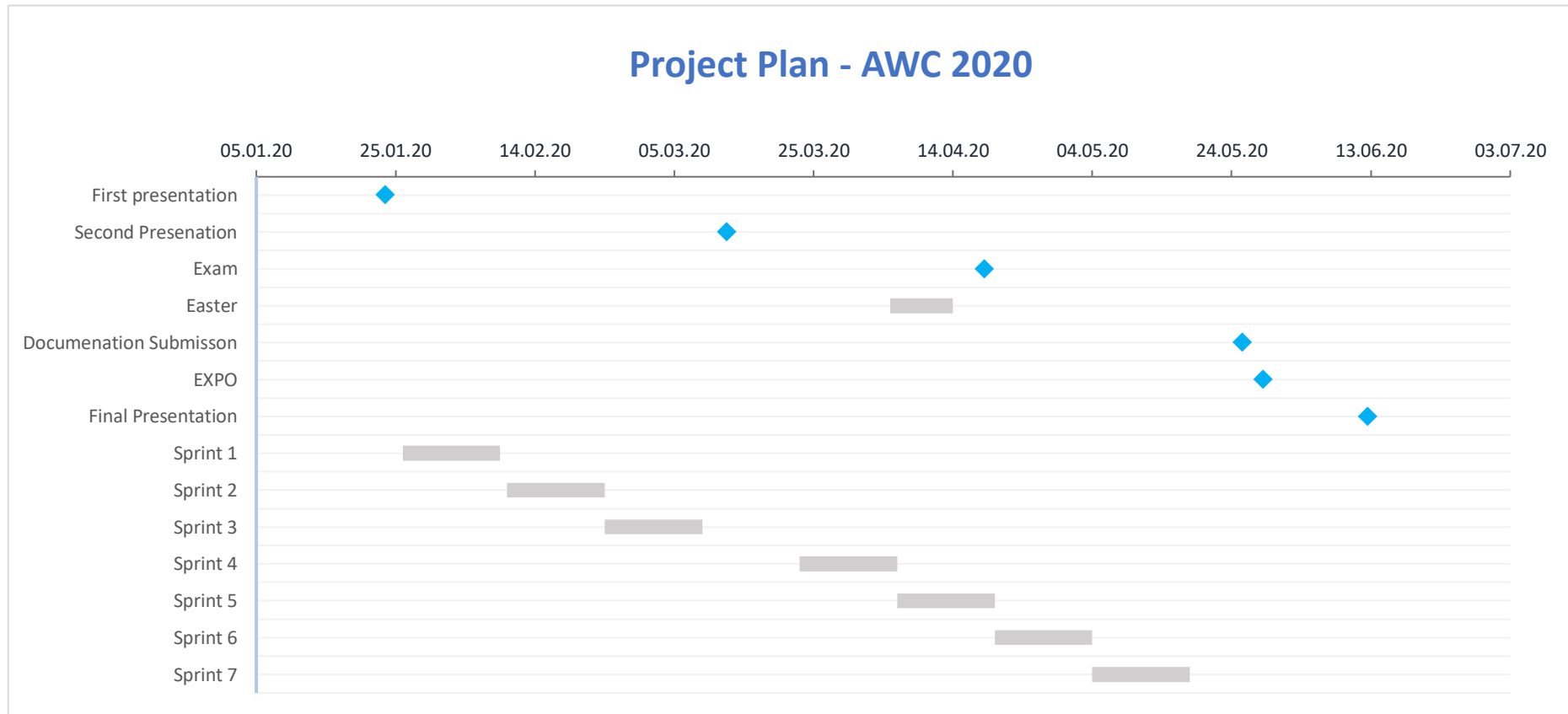
14 APPENDIX E - SPRINT DOCUMENTATION

Abstract

This document includes the project plan and all sprint retrospective made at the end of each sprint.

- Sprint 1
- Sprint 2
- Sprint 3
- Sprint 4
- Sprint 5
- Sprint 6

14.1 Project Plan



14.2 Sprint retrospective – Sprint 1

What worked well.

Feel that people were good at taking tasks. Really happy that TechnipFMC was pleased with the work we did. Attendance became gradually better after we implemented cake punishment. We believe that we are the bachelor team with the highest attendance rate! Been good at doing research and discovering new things. Things might seem inefficient in the beginning, but by having good objective discussions at the start will give us a better understanding of what each team member thinks. The scum model is working, daily stand-up is going smooth, but the rest of the process is still under work. Nina did a really good job about acceptance criteria for each user story. Petter and Erik has done a lot of trying and failing involving the HoloLens2 and by doing this they save the rest of the team members valuable time is the setup phase. We've had good communication with TechnipFMC they have been super active to help and guide the group. Steven has been a good asset for the group and have help us understand different types of problems.

- Scrum is working well.
- Better attendance after cake disciplinary.
- Group is pleased with guidance from TechnipFMC.

What could be improved?

People are not active at assigning tasks in Azure DevOps, and some group members are working on several tasks in different user stories. Team members should focus on one user story at a time. Need to write acceptance criteria when user stories are made. We made way to large user stories for the first sprint, and we need to break them into smaller stories. We did not use the description field well so there was some confusion on different user stories. Low efficiency, but we hope this was due to the start-up phase of the project. Need to be better at documenting research and document why decisions were made/ not made. User stories did not have tasks made for them, so this was also a reason for confusion during sprint planning. We were bad at debugging, this has to be improved.

Do not be uninterested during discussions, turning around and continue working is not a solution. Need to be better at taking breaks, too few breaks are not good for the mood of the team. Kristian and Even must be better at log hours.

- Confusion around user stories because of bad wording and missing tasks
- Not great at debugging
- People not interested in discussing

Focus areas for the next sprint:

Be better at taking breaks. Be better at doing light documentation during sprints

14.3 Sprint retrospective – Sprint 2

What worked well?

Good that we started doing some simple UX design. Good that we had such a bad experience with the sprint planning this early in the project, this gave us the “zoom-out” we needed for our project to make sure that we are heading in the right direction again. External supervisor liked our UX propositions and the prototype we made, but reminded us that we need to focus on what elements the systems needs and not elements that is “nice to have”. Test templates is done and ready for use, same with risk.

- Good to fail early in the project.
- Started with flow and mock-up of the application.
- Prototype up and running.

What could be improved?

UX had many elements that was nice to have and not so many need to have. People started to scope outside of the sprint, and this is not how sprints should be done. When people are doing whatever they want then it is no reason for the product owner to plan sprints.

Marie had no tasks assigned to her; every task was already taken. Bad at planning poker and at estimating how long a task/user story takes. We still have way to big user stories for the sprint and we need to reconsider our whole sprint planning Monday 24.02.2020. We think the reason this sprint went this bad was because we only had 2 out of 6 people working on design and user experiences when several user stories were about design and concept. Been using different unity versions and now we need to freeze a version. Everyone needs to set up their git.

- Not enough tasks for everyone.
- People focused on other things than what was planned during this sprint.
- Still too large user stories.

Focus areas for the next sprint:

We need smaller user stories and we need to be come together as a group and write better and smarter user stories. Need to break the assignment into features from use cases and make user stories smaller from features.

14.4 Sprint retrospective – Sprint 3

What worked well?

Finally found the right size of how big our user stories should be broken down into and made a structure that fits our project (epic-feature-user story-task). Helped a lot to start with use cases and plan from this level. Our epic is now on the same level as our use case diagrams and each feature is a use case, and user story was made from this. Completed our UX concept. Feeling that we are ready for second presentation and prototype is now able to open close a valve from the HL2. Got help with software architecture from TechnipFMC.

- Structured our work hierarchy and flow of the system.
- UX concept is done.
- Prototype is enhanced.

What could be improved?

More efficient work strategy. Still not many tasks assigned to the user stories, need to find a way to document chores that's not connected to a user story. Be more involved in meetings and ask questions to make sure there are no misunderstandings. If we are insecure about anything, regarding how the system works, ask on Slack instead of using too much time on arguing! Remember to document research.

- Better work structure
- Don't be afraid to ask silly questions (no question is silly)
- Be better at documentation.

Focus areas for the next sprint:

The two next weeks is dedicated for second presentation and preparing for simulation and modelling exam. We need to prepare and plan the second presentation and write documentation. Otherwise, for the next sprint: Divide into smaller groups when working on one big task like UX/user stories, to be more efficient. Not everyone needs be part for every small decision.

14.5 Sprint retrospective - Sprint 4

What worked well?

Daily meetings on Teams to update every team member on their work progress. Technical areas have been given out and each team members have started with their responsibility. Our strategy, where one group member took the responsibility to play and learn with MRTK toolkit and our system, has been proven to be a success. The possibility to give remote support through TEAMS. Thinking especially when configuring and set all settings that is needed when building in Unity with MRTK. This has also done it easy to ask for help when one of the team members has problem with Unity in general and needs to learn how the system works. Really nice that we managed to keep our time limit to second presentation when Corona broke out. Also looking at multiple solutions for solving a problem. Divided the technical areas into independent areas. Doing this we are not dependent at others team members to finish our tasks.

- Keeping time schedule for second presentation
- Technical strategy is a success
- Everyone got their technical area and has started to develop

What could be improved?

Not using DevOps. Things are difficult due to corona and we're struggling to adapt to this new everyday situation. It's hard to work efficient when there are so much chaos and distractions. Need to find a way to structure our days. Not the same amount of communication as we are used to have. Difficult to see each other's progress even though we talk every day at scrum, but it's not the same. GIT is not working. Another team member is waiting for it to work, this is not good. Even is not going to have another job interview during our final documentation time.

- DevOps and GIT is not going well.
- Communication needs to be improved, not enough with morning meetings.
- Motivation and daily structure are difficult due to the pandemic.

Focus areas for the next sprint:

Use DevOps! Change our agile process and adapt it to the new situation. Find out what workflow suits you the best. Use Teams more efficient if you need to get help with technical problems. (Next sprint is only one weeklong because of exam in another unit)

14.6 Sprint retrospective - Sprint 5

What worked well?

Adapted and changed our agile working process. People managed to multitask project and exam and got further into their development process. A solution to one of our main problem was identified.

What could be improved?

GIT problems were not solved right away. Used too much time on this problem. Important to ask for help when help is needed. Still some communication issues. Need to be clearer when asking questions. Changed meeting schedule.

Focus areas for the next sprint:

Manage to get GIT working. Change meeting schedule back to normal! Develop, develop, develop. Start to set deadlines.

14.7 Sprint retrospective - Sprint 6

What went well?

Still close communication with TechnipFMC and we got a simulator that finally works. The application starts to take shape and the three first parts is close to done. We have managed to work around the fact that GIT has not been operational and made a new one with proper .gitignore files. Better use of dev ops with tasks by several team members.

- Simulator is connected.
- Application is starting to take form.
- New GIT is fixed.

What could be improved?

Deadlines is not met by some team members. Not all members pay attentions during morning meetings. It was very clear that the data points and address needed was available, but still there was a misunderstanding where members though they needed the simulator for completing their tasks. Even though we have longer morning meetings, people are still not asking for help and when asked if things are going well the answer is always yes while the reality is different. There is a lot of good communication, but we need to be better at including everyone and ask more questions about our technical task to challenge each other. Also, it is important to include yourself into others work and give feedback.

- Take responsibility and ask for help.
- Pay more attentions to all team members work.

What do we want to focus on next (last sprint) and what should we do better?

Provide as much documentation as possible before entering the final phase of the assignment. Finished asking a lot of documentation. We must focus on communication and that everyone is included.

14.8 Sprint retrospective - Sprint 7

What went well?

Usability tests were conducted by stakeholder. Technical area finalized and project was assembled. Diagrams are looking better. Everyone is putting in more workhours to finish on time. Doxygen was discovered. Communication during the last sprint has been great.

What could be improved?

Team members are sitting idle waiting for others to finish getting feedback, assembling the project. Team members still not sure how to pull and merge without errors. Starting to notice sleep deprivation, hunger, and low intake of water. Caffeine does not have effect anymore. Nina has finally converted over to the dark side where they serve frozen pizza and Fjordlands

Focus areas for the next sprint:

Finish documentation and finetune UX on the application. Remember that there are some common areas outside your technical area that need to be documented. Important that everyone participates in this and tries their best to make the whole documentation good, not just "your" part. (Sprint 8 week is only one week, as it is the last one in this project).

14.9 Sprint retrospective - Sprint 8

Documentation and application are finished and ready to submit. Everyone worked hard to finalize the final product and documentation. Happy to submit!



15 APPENDIX F – APPLICATION USER GUIDE

Abstract

In this appendix the contents of the readme that is also present within the application will be shown.

15.1 Choose User

The choose user menu is displayed in the start of the application. This is where you decide if you want to log in as administrator or as a guest user.

The guest user is able to access all the functionality except the guest user CANNOT open or close valves. Administrator needs to type in required password to be logged in.

15.2 Manage XT

In the Manage-XT view you will be able to add, edit, delete and connect to an XT. To add a XT you will need to press the "+" button and fill out the required information and then press "add". You will then see that your XT is stored in the XT list. To edit or delete a XT you can simply press "Edit" or "Remove". To connect you will need to press or click "Connect" on the stored XT. The XT list will hold 4 XTs, so if you want to add a fifth you will need to delete an existing one.

15.3 XT Schematic

In the XT-Schematic view, you will be able to see the layout of your desired XT. If you are signed in as Guest you will be able to view sensor information, names and state of each valve. You can also press or click on the sensors or valves to get more information on the specific valve or sensor. If you are signed in as Admin you will have the authority to open and close valves in addition to regulate the choke

15.4 Hand Menu

Buttons in the hand menu

- Home view - Takes you back to Manage XT where you can connect or edit XT's.
- Clear view - Unclutter your view and close all valve, sensor and other information that you have opened.
- HCR view - Toggle HCR on and off.
- Read Me - Toggles read me on and off.
- Close application – Exits to desktop.



16 APPENDIX G - WORKING HOURS

Abstract

Overview of the total hours worked during the project. Details will be on the memory stick in the document "Working Hours.xlsx". Week 0 is the total amount of hours spent on the project before Christmas.

Total working hours from Autumn 2019 until 25th of May.

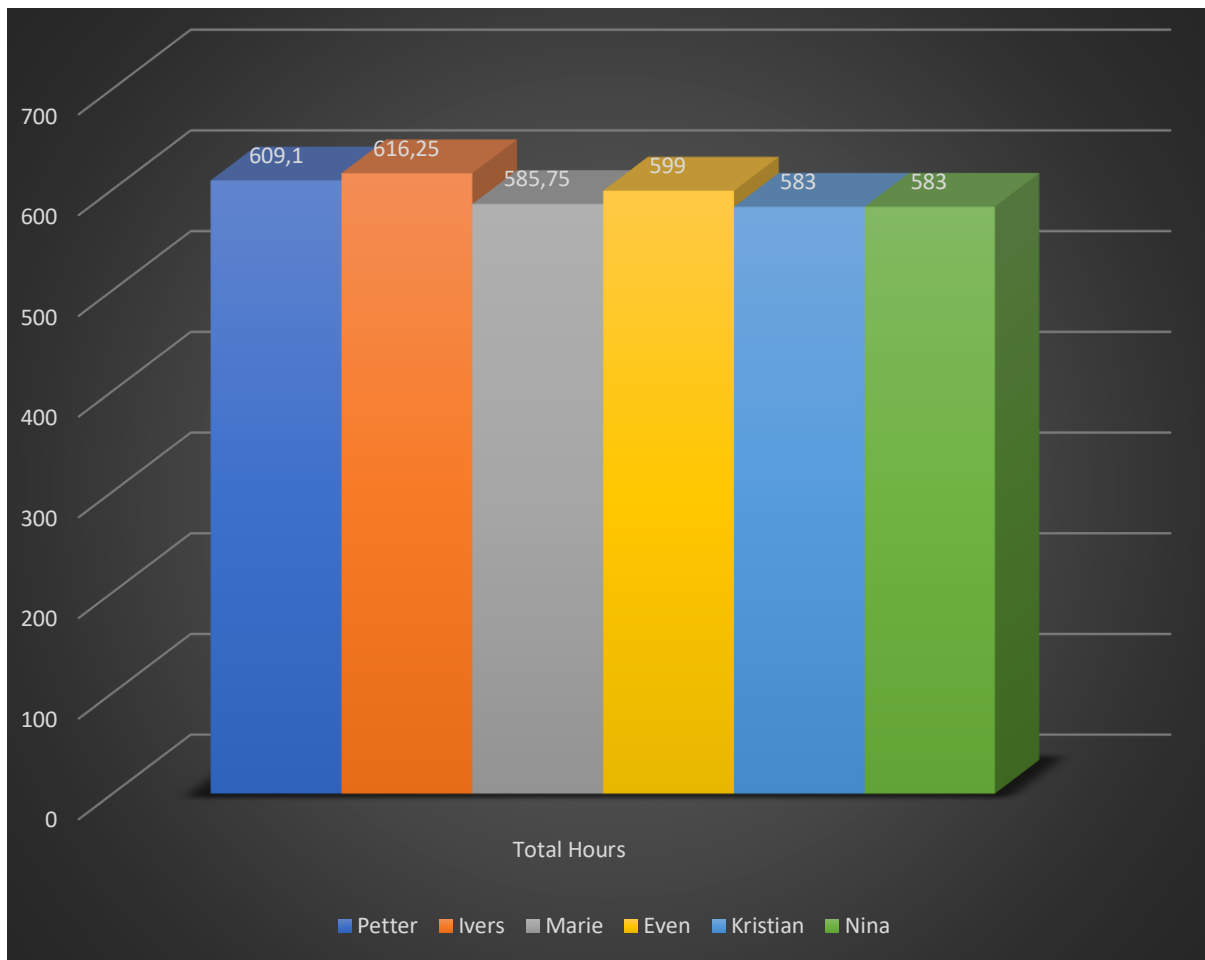


Figure 147 - Working Hours

Month	Petter	Ivers	Marie	Even	Kristian	Nina
Autumn	18	28	17	28	12	16
Jan	112,05	102,75	98	99,25	89,75	91,75
Feb	105,15	97	76,25	76,75	89,25	99
Mar	83,4	70,5	95,5	79	90,5	65,75
Apr	114,5	112	123	114	108	118
Mai	176	206	176	202	193,5	192,5
Jun	0	0	0	0	0	0
Total	609,1	616,25	585,75	599	583	583

17 References

- [1] «Kildekompasset,» 24 5 2020. [Internett]. Available: <http://kildekompasset.no/referansestiler/ieee.aspx>. [Found 24 5 2020].
- [2] Unity Technologies, «Unity Announces Support for Microsoft HoloLens,» Unity, 29 April 2015. [Internett]. Available: <https://unity.com/our-company/newsroom/unity-announces-support-microsoft-hololens>.
- [3] Unity, «IconFinder,» [Internett]. Available: https://www.iconfinder.com/icons/4691514/unity_icon. [Found 23 May 2020].
- [4] MRTK. [Internett]. Available: <https://microsoft.github.io/MixedRealityToolkit-Unity/version/releases/2.3.0/Documentation/GettingStartedWithTheMRTK.html>.
- [5] davidkline-ms, «GitHub/MixedRealityToolkit,» [Internett]. Available: <https://github.com/Microsoft/MixedRealityToolkit-Unity/releases>.
- [6] «Image of MRTK logo,» [Internett]. Available: <https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/GettingStartedWithTheMRTK.html>. [Found 09 March 2020].
- [7] «MRTK Input Events,» [Internett]. Available: <https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/Input/InputEvents.html>. [Found 09 March 2020].
- [8] «MRTK Standard Shader,» [Internett]. Available: https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_MRTKStandardShader.html?q=standard. [Found 09 March 2020].
- [9] "MRTK Scene System," [Online]. Available: <https://microsoft.github.io/MixedRealityToolkit->

- Unity/Documentation/SceneSystem/SceneSystemGettingStarted.html. [Accessed 09 March 2020].
- [10] «MRTK ButtonPrefabs,» [Internet]. Available: https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Button.html. [Funnet 09 March 2020].
- [11] «MRTK Interactable,» [Internet]. Available: https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Interactable.html. [Found 09 March 2020].
- [12] «Image of Slack logo,» [Internet]. Available: https://www.dropbox.com/nb_NO/app-integrations/slack. [Found 09 March 2020].
- [13] Scrum, [Internet]. Available: <https://apk.support/app/br.com.mk4.scrumsim>. [Found 23 May 2020].
- [14] «Image of Microsoft teams logo,» [Internet]. Available: <https://www.smartit.ch/ueber-smartit/aktuelles/blog/office-workplace/nuetzliche-tipps-microsoft-teams/>. [Found 09 March 2020].
- [15] «Image of Facebook messenger logo,» [Internet]. Available: <https://www.messenger.com/>. [Found 09 March 2020].
- [16] «Image of Azure DevOps logo,» [Internet]. Available: <https://logodix.com/azure-devops>. [Found 09 March 2020].
- [17] «Git --distributed-even-if-your-workflow-isnt,» [Internet]. Available: <https://git-scm.com/>. [Found 09 March 2020].
- [18] «Git_logo,» [Internet]. Available: <https://git-scm.com/downloads/logos>. [Found 09 March 2020].
- [19] «File:Adobe XD CC icon.svg,» [Internet]. Available: https://commons.wikimedia.org/wiki/File:Adobe_XD_CC_icon.svg. [Found 09 March 2020].

- [20] «LucidChart logo,» [Internett]. Available: <https://tigerware.lsu.edu/Discount/251/Lucidchart-for-Education#>. [Found 09 March 2020].
- [21] «Pluralsight,» [Internett]. Available: <https://www.pluralsight.com/paths/visual-studio-2019>. [Found 10 March 2020].
- [22] «Freeiconspng,» [Internett]. Available: <https://www.freeiconspng.com/img/28402>. [Found 09 March 2020].
- [23] «Pngitem,» [Internett]. Available: https://www.pngitem.com/middle/iJxbbwi_new-microsoft-word-icon-hd-png-download/. [Found 10 March 2020].
- [24] «Wikipedia,» [Internett]. Available: [https://no.m.wikipedia.org/wiki/Fil:Microsoft_Office_Excel_\(2018%E2%80%93present\).svg](https://no.m.wikipedia.org/wiki/Fil:Microsoft_Office_Excel_(2018%E2%80%93present).svg). [Found 10 March 2020].
- [25] «pngitem,» [Internett]. Available: https://www.pngitem.com/middle/ihmJTTJ_microsoft-office-icon-png-image-free-download-searchpng/. [Found 10 March 2020].
- [26] «Pngitem,» [Internett]. Available: https://www.pngitem.com/middle/hxJwTxR_microsoft-powerpoint-new-powerpoint-icon-png-transparent-png/. [Found 10 March 2020].
- [27] K. S. a. J. Sutherland, «The Scrum Guide tm - The Definitive Guide to Scrum: The Rules of the Game,» 2017. [Internett]. Available: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>. [Found 20 Januar 2020].
- [28] O. Mandryk, «iconderman,» 2016 February 2016. [Internett]. Available: <https://icoderman.wordpress.com/2016/02/23/the-agile-scrum-framework-at-a-glance/>. [Found 20 May 2020].

- [29] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, Arbor, Michigan: Addison-Wesley, 2012.
- [30] «Technopedia,» 30 August 2019. [Internet]. Available: <https://www.techopedia.com/definition/27809/minimum-viable-product-mvp>. [Found 08 March 2020].
- [31] K. Heldman, *Project Management JumpStart*, John Wiley & Sons Inc, 2018.
- [32] N. R. Council, *The Owner's Role in Project Risk Management*, Washington, DC: The National Academies Press, 2005.
- [33] F. Usmani, «Blog PM Sprout,» 29 March 2018. [Internet]. Available: <https://blog.pmsprout.com/risk-response-strategies/>. [Found 04 March 2020].
- [34] «Measuring u,» [Internet]. Available: <https://measuringu.com/sus/>. [Found 9 Mars 2020].
- [35] K. Gutttag, «Karl Gutttag on Technology,» [Internet]. Available: <https://www.kgutttag.com/2019/12/03/hololens-2-hl2-scan-lines-making-text-hard-to-read-and-quality-issues-with-waveguides/>. [Found 06 02 2020].
- [36] «Color, light and materials,» [Internet]. Available: <https://docs.microsoft.com/nb-no/windows/mixed-reality/color-light-and-materials>. [Found 13 02 2020].
- [37] M. J. Davis, «<http://davismj.me/>,» 28 January 2016. [Internet]. Available: <https://davismj.me/blog/bcrypt/>. [Found 24 April 2020].
- [38] «Microsoft,» 29 August 2018. [Internet]. Available: <https://docs.microsoft.com/en-us/visualstudio/cross-platform/unity-scripting-upgrade?view=vs-2019>. [Found 24 April 2020].
- [39] Gwyll, «FOLOGRAM,» 20 Januar 2020. [Internet]. Available: <https://community.fologram.com/t/hololens-2-polygon-count-and-frame-rate/49>. [Found 15 May 2020].

- [40] «Microsoft,» 26 02 2019. [Internett]. Available: [https://docs.microsoft.com/en-us/windows/mixed-reality/mrlearning-sharing\(photon\)-ch1](https://docs.microsoft.com/en-us/windows/mixed-reality/mrlearning-sharing(photon)-ch1). [Found 24 05 2020].
- [41] «Mockplus,» 12 Mai 2017. [Internett]. Available: <https://www.mockplus.com/blog/post/ux-persona-examples>.
- [42] B. Dumas, «Unity Assetstore,» 27 2 2020. [Internett]. Available: <https://assetstore.unity.com/packages/tools/modeling/splinemesh-104989>. [Found 22 05 2020].
- [43] Unity, «Update,» [Internett]. Available: <https://docs.unity3d.com/ScriptReference/MonoBehaviour.Update.html>. [Found 19 5 2020].
- [44] Unity, «EventFunctions,» [Internett]. Available: <https://docs.unity3d.com/Manual/EventFunctions.html>. [Found 19 5 2020].
- [45] Defuse Security, «crackstation,» [Internett]. Available: <https://crackstation.net/hashing-security.htm#ineffective>. [Found 27 March 2020].
- [46] Wikipedia, «Wikipedia,» [Internett]. Available: <https://en.wikipedia.org/wiki/Bcrypt>. [Found 27 March 2020].
- [47] «A successful Git branching model,» [Internett]. Available: <https://nvie.com/posts/a-successful-git-branching-model/>. [Found 09 March 2020].
- [49] «Unity Documentation,» Unity, 21 05 2020. [Internett]. Available: <https://docs.unity3d.com/Manual/IL2CPP-OptimizingBuildTimes.html>. [Found 22 05 2020].
- [50] «MRTK Documentation,» Microsoft, NA NA NA. [Internett]. Available: <https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/Tools/HolographicRemoting.html>. [Found 22 05 2020].
- [51] [Internett]. Available: <http://satheespractice.blogspot.com/2011/11/what-is-agile-scrum.html>. [Found 24 May 2020].

-
- [52] Blender, «wikimedia,» [Internett]. Available:
https://commons.wikimedia.org/wiki/File:Blender_logo_no_text.svg. [Found 23 May 2020].
- [53] photoshop, «worldvetrologo,» [Internett]. Available:
<https://worldvectorlogo.com/logo/adobe-photoshop-cs6>. [Found 23 May 2020].