LUT University

Running a Software Project

Vision system configuration tool

0401495 Joel Salminen

0438345 Nidal Abu Raed

0387849 Teemu Hokkanen

0403299 Ville Tele

ABSTRACT

This report will describe the process of implementing a vision system configuration tool. This tool will enable the end user to update the different hardware connected to the system. It also enables the user to create custom configurations, that can be ordered from the company. This report describes the interactions with the customer and the different phases of planning, implementing and testing the product.

Vision system configuration tool
Project Report

13 pages, 1 figures, 1 tables, 1 appendix

Keywords: vision system, configuration tool, Qt, multiplatform

ACKNOWLEDGEMENTS

Thank you for Savox and Veli-Matti for cooperation with this project. We’d also like to thank Jari and Jayden for understanding our special circumstances and making it possible for us to take part in the course.

LIST OF SYMBOLS AND ABBREVIATIONS

ACL	Access Control List
CSV	Comma-Separated Values
NHL	National Hockey League
Wi-Fi	Wireless Fidelity

(All symbols and abbreviations are listed on this page in alphabetical order. Remember to introduce the abbreviation when it is used in the text for the first time.)

1 INTRODUCTION

1.1 Customer and the problem to be solved

The customer is Savox which specializes in creating equipment for challenging conditions like on which for example police officers, firefighters and soldiers face. The product we were creating our solution to was a system where you could add for example screens and different kinds of cameras. Our project was a configuration tool, which gave to user the possibility to some of the step themselves. In this case the steps were updating the firmware of the connected hardware and planning the configurations for the vision system.

Without this new configuration tool the customers have to rely on the company to take care of the actual updates on their hardware. The customers had to also describe and plan the configuration without any visual aid. The software would be on a computer which would be connected to a switch where all of the hardware is connected. This way the updates could be made to the hardware.

The basic concept of the tool is an application, which provides a possibility to update different hardware connected to the system. This application should also enable naming different inputs and outputs. The inputs in this context mean cameras of some sort and the outputs mean screens. The application should also enable configuring what inputs are shown in the outputs. This configuration could be exported as a CSV file, which could be emailed to the company to get the right configurations to update to the hardware.

1.2 Vision and delimitations

The first version of this configuration tool will enable the two major features of updating firmware and creating CSV configurations. However this will be only the start of this application. The software could be able to communicate directly to the switch and actually create and upload the configurations to the devices. This way the customer could make all the current steps required to make changes to the vision system. This could also enable new features unlocked to the customer and configuration tool, since these deliveries wouldn’t always have to be hand made.

There were also talks about creating dummy functionalities with a existing interface for transferring commands to the switch. This however was one of the things we had to decline. Our project team was not familiar with the used technology, so we had to focus on the most important thing. We decided with the customer to focus on creating this application which could be used to do the main functionalities mentioned before. This meant that we purely focused on a functional user interface and left out everything else.

1.3 Structure of the report

Section 2 will describe in detail what was the starting situation before the actual implementation of the project. This includes describing the team and the way we worked, the requirements we got from the customer and the implementation plan we made before the actual implementation. Section 3 will describe what steps we took to help create a more user friendly design for the application. Section 4 will go through the actual implementation of the software. Section 5 describes the testing we did during the project and what the future will hold for this application. The last section will have conclusions about the project. This means that we reflect on how our project went and what we could’ve done better and also there will be a conclusion about the whole report.

2 PROJECT

2.1 Project team and the management

Our team consists of four members. We are all full time employed and we live near Helsinki. We did not know each other before the course. Our team never decided any roles formally. However as the project went on, some roles did emerge. Joel was more in charge of the actual implementation of the project. Nidal had also a small part in that, but he did more of the general work and testing. Teemu's role was also general work and he also looked after some of the arrangements, such as keeping track of deadlines and arranging meetings with customer, course holders and the team itself.

The most important tool for our communication was whatsapp. This is how we kept in touch and arranged all of our meetings and a lot of decisions too. Close second was skype, since we had all of our meetings with it. We did all of our documentation to a google drive folder, that we had access to. We also had trello set up in the earlier parts of the course, but after maybe half of the course we had less meetings and this is why we stopped updating it. Our implementation had gitlab as version control.

For about half of the course our team didn't have different roles. We balanced the work and divided it depending on the different schedules we all had. This can be seen in the appendix weekly reports from 1 to 5. After we got the second customer meeting and had much better understanding about the project, we split into two groups. Nidal and Joel took upon themselves to learn about the implementation technology and to do the actual implementation. Teemu and Ville worked on the architecture and the improvements on the user interface. We kept these roles mostly until end of the course.

2.2 Requirements

Our team gathered the most of the requirements in two customer meetings. We had some difficulties understanding the customers needs, so we had to have a second meeting in person. After the second meeting we could gather most of the requirements.

ID	Requirement	Accepted/Declined
1	The application should enable updating the firmware of devices. The devices have different firmwares, so the updates should be done to specific device. Savox will create the actual firmware.	Accepted
2	The application should enable the possibility to add a new output (screen). There should also be a possibility to update firmware of the new device.	Accepted
3	The application should enable configuring the quad view in a way, that you can choose which inputs are represented on the screen in the desired order.	Accepted
4	The application should enable to configure screens to other outputs with firmware update.	Accepted
5	The application should enable offline use, meaning that the configuration can be made and sent to Savox in a different occasion.	Accepted
6	The application should enable the naming of inputs and outputs for clarity.	Accepted
7	The application should enable adding a new camera or screen to a existing configuration. This would be done by taking firmware from another device and trying to update it to the new one.	Declined
8	The application should enable the user to show something on the selected screen, so that the screen could be recognized.	Declined
9	The application should enable to change the quad view to a new screen	Accepted

10	The application should enable the quadview to be configured to have different options with two small screens and one larger.	Accepted
11	The application should enable sending email to Savox if the firmware update would fail for some reason.	Declined
12	The application should work on windows and unix based operating systems.	Accepted
13	The applications colors should be configurable.	Declined
14	The application should enable creating a CSV from the applied configuration.	Accepted
15	The application should enable factory reset to the connected devices.	Accepted
16	The application should show the order of the configured views in the CSV file.	Accepted
17	The application should be scalable to smaller screens, since the end users devices might vary.	Accepted
18	The application should enable keyboard only use.	Accepted
19	The application should enable adding pictures on top of the camera feed.	Declined

2.3 Implementation plan

As mentioned before, before the implementation phase our group split in to half. The other part started to study the technology needed to create the application. The other half of the group started to create a more user friendly user interface and started to plan the architecture of the application. With these preparations we got a common understanding of the direction we should go with this implementation. At this point we had been planning for too long and we had to start doing the actual implementation.

3 USER EXPERIENCE DESIGN

User experience is about understanding users properly when they are interacting with application. It is about getting to know users emotions, environment, preferences and demographics, in order to design perfect experience for the end user. In our project, we started mapping user experience by profiling the users with different personas. This gave us some indication what the users expect from the system, and that way provided us good reasons to make decisions about some non-functional requirements. After personas, we created journey maps, which, in turn, provided us more detailed information about the users thoughts. From these two artifacts we did derive some type of prototype which we presented to the customer for fast feedback, and to raise out some uncertainty.

3.1 Personas

Picture: Persona 1. Officer who configures screens, but doesn’t use the device in practise.

Picture: Persona 2. Squad leader, who wants to see all relevant information from screen, in order to make decisions.

Picture: Persona 3. Soldier, who uses the device. Changes views as commanded.

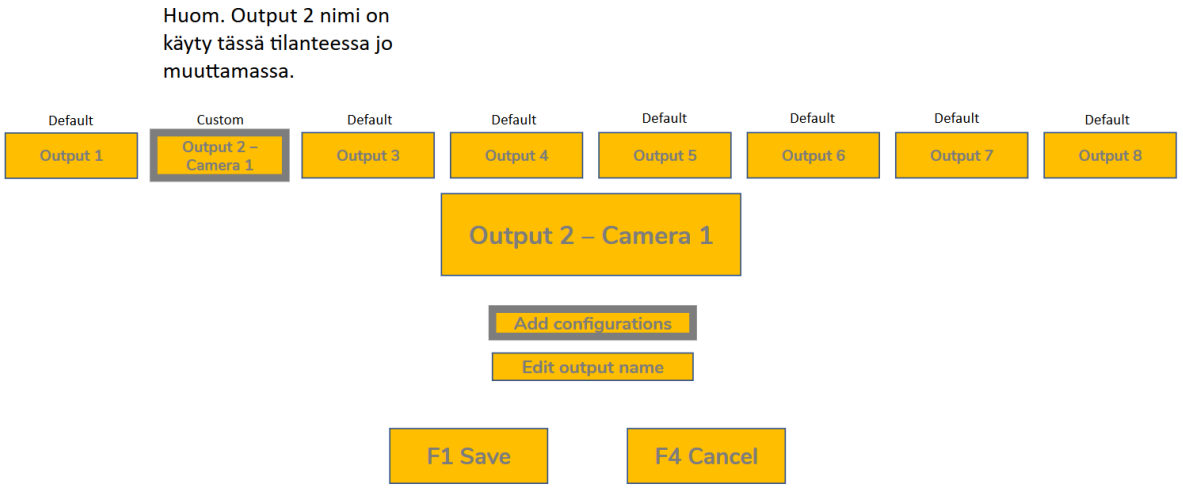
3.4 Journey Maps

	Open Software	Navigate to create new configuration	Choose output (screen) to configure	Choose the outputs's "view" (configuration) to modify	Choose input for selected view (single)
Actions	Find laptop, start software	From home screen, choose "create configuration" (F1)	Navigate to wanted output, select "add configurations" (F1)	Edit "view" with (F2)	Navigate to "edit input" (F1), select input (1-8) → save (F1)
Questions	How to use this?	Can I modify existing configuration?	How do I know which is my preferred screen? What is the current configuration of the screen?	Which inputs are configured to this "view"?	
Happy moments		This seems easy and fast!	This is simple! Oh, here i can edit output name!	Oh, here I can choose type of the "view"! (single/multi)	Oh, here I can change input name!
Pain points	"There is not a lot of space"		All the outputs are named in a confusing manner!	Still cant figure out how the outputs are named!	
Opportunities			Allow naming of outputs	Allow naming of outputs	

	Open Software	Navigate to create new configuration	Choose output (screen) to configure	Choose the outputs's "view" (configuration) to modify	Choose view type multiview and modify it	Choose layout for the "view"	Map inputs to desired positions of the layout
Actions	Find laptop, start software	From home screen, choose "create configuration" (F1)	Navigate to wanted output, select "add configurations" (F1)	Edit "view" with (F2)	Navigate to "view type" (F1), select multiview, select "edit multiview"	Scroll between layout designs (F2,F3)	Select "edit layout" → "select input" → save
Questions	How to use this?	Can I modify existing configuration?	How do I know which is my preferred screen? What is the current configuration of the screen?	Which inputs are configured to this "view"?		What is suitable for me?	
Happy moments		This seems easy and fast!	This is simple! Oh, here i can edit output name!	Oh, here I can choose type of the "view"! (single/multi)			
Pain points	"There is not a lot of space"		All the outputs are named in a confusing manner!	Still cant figure out how the outputs are named!			I'm not sure which input is which camera, since they haven't been named!
Opportunities			Allow naming of outputs	Allow naming of outputs			

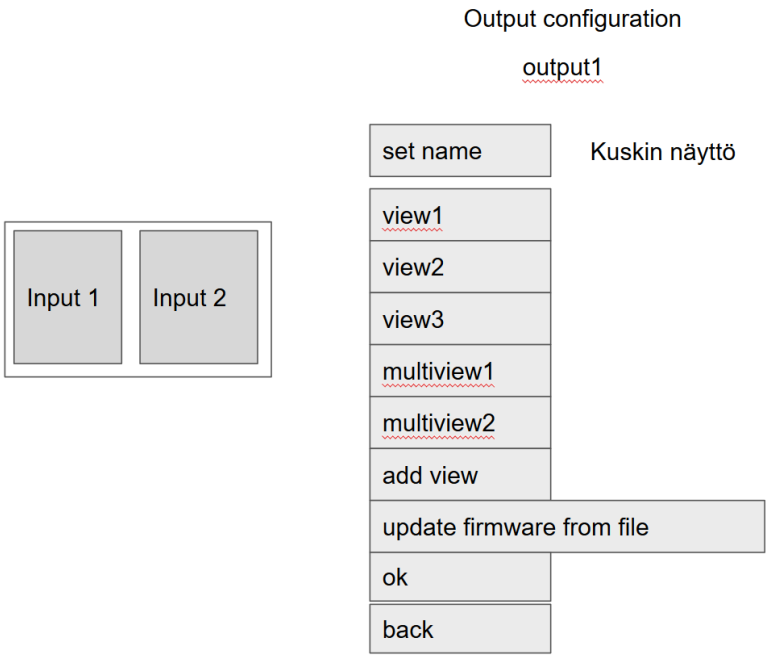
3.5 Prototypes

First prototyping was done with Marvelapp, in order to have something concrete to discuss with customer. We knew the layout and menus weren't final, but we wanted to make something to get clear feedback what is good and what requires improvement. There was lots of issues with communication, as given specs and requirements were not clear, and they seemed to change all the time depending what was asked and how. Initial prototype was made to anchor something, so we could get started

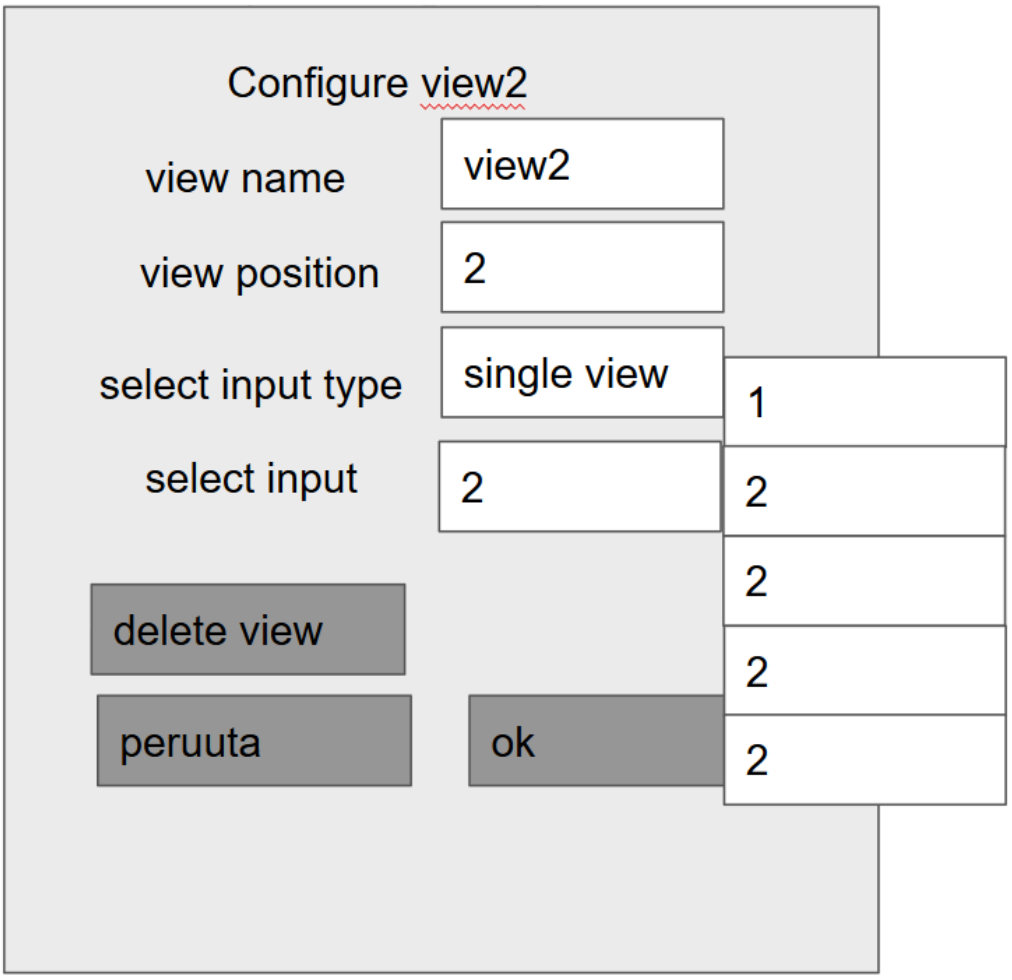


Picture: Original design. Lots of buttons, navigation with arrows and function keys.

After the feedback, we revamped almost whole UI using PowerPoint to represent different views and windows on the program.. We structured the key info differently, and went more in detail. Based on that plan, we created the first Qt program, which we demoed to the customer with good success. There were only few minor things to change, which was pretty quick operation.



Picture: Different layout. Navigation with just arrow keys.



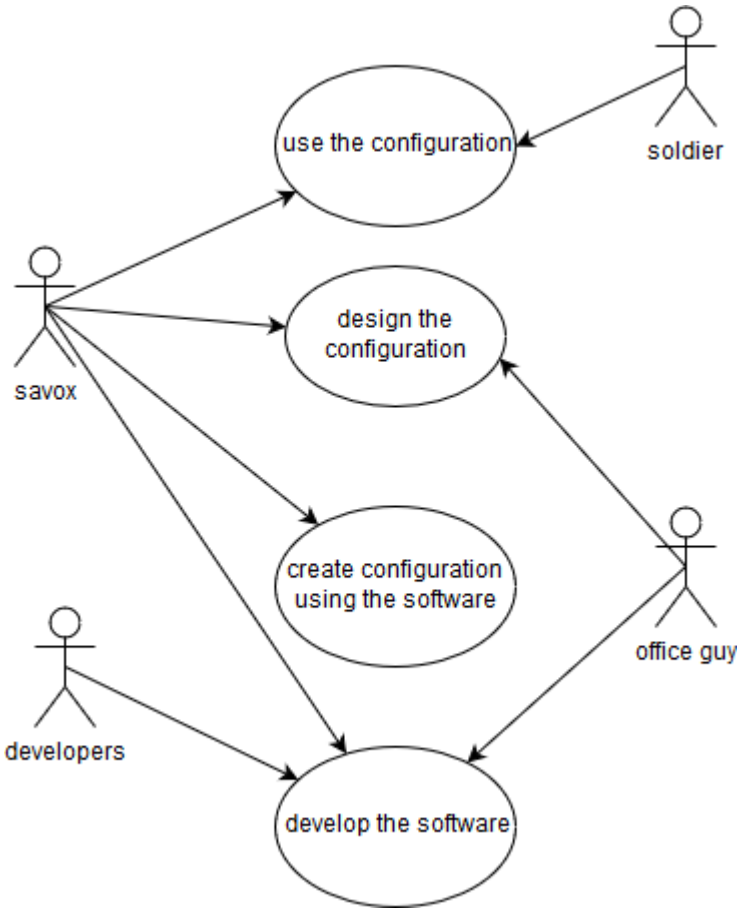
Picture: Configuration is done in pop-up windows, when view is selected.

4 DEVELOPMENT

The development process was a combination of Lean development and some aspects of extreme programming (planning, testing, small releases). The development process focused on increasing feedback, delivering fast, and eliminating waste, but at the same time emphasizing on quality and the core functionalities. In terms of testing, test plan (test cases) was created for different functionalities and then manually executed.

4.1 Application overview

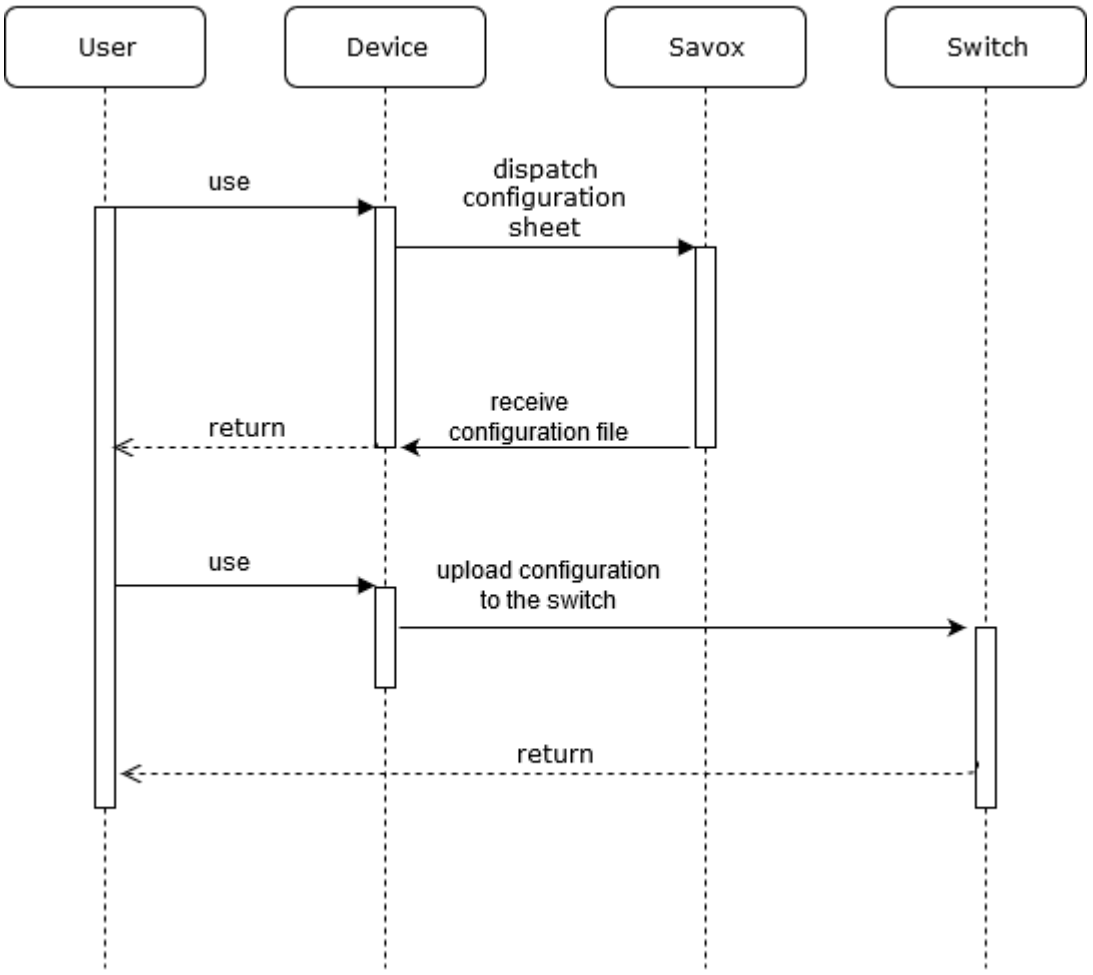
The vision system configuration tool for creating customized views. Configure input sources to create your own monitoring display. Create configurations easily and fast almost anywhere. Export configurations to your colleagues, import requested configurations and modify your own!



Picture: Use case diagram

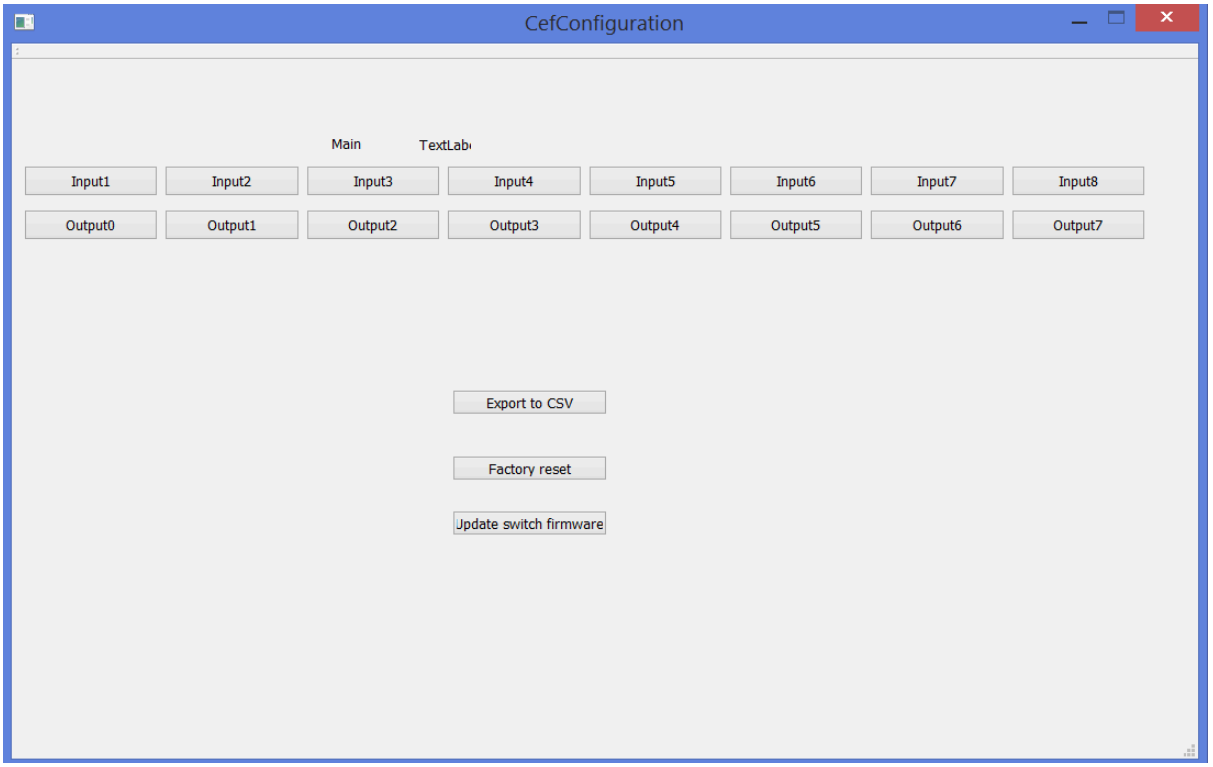
4.1.1 Architecture

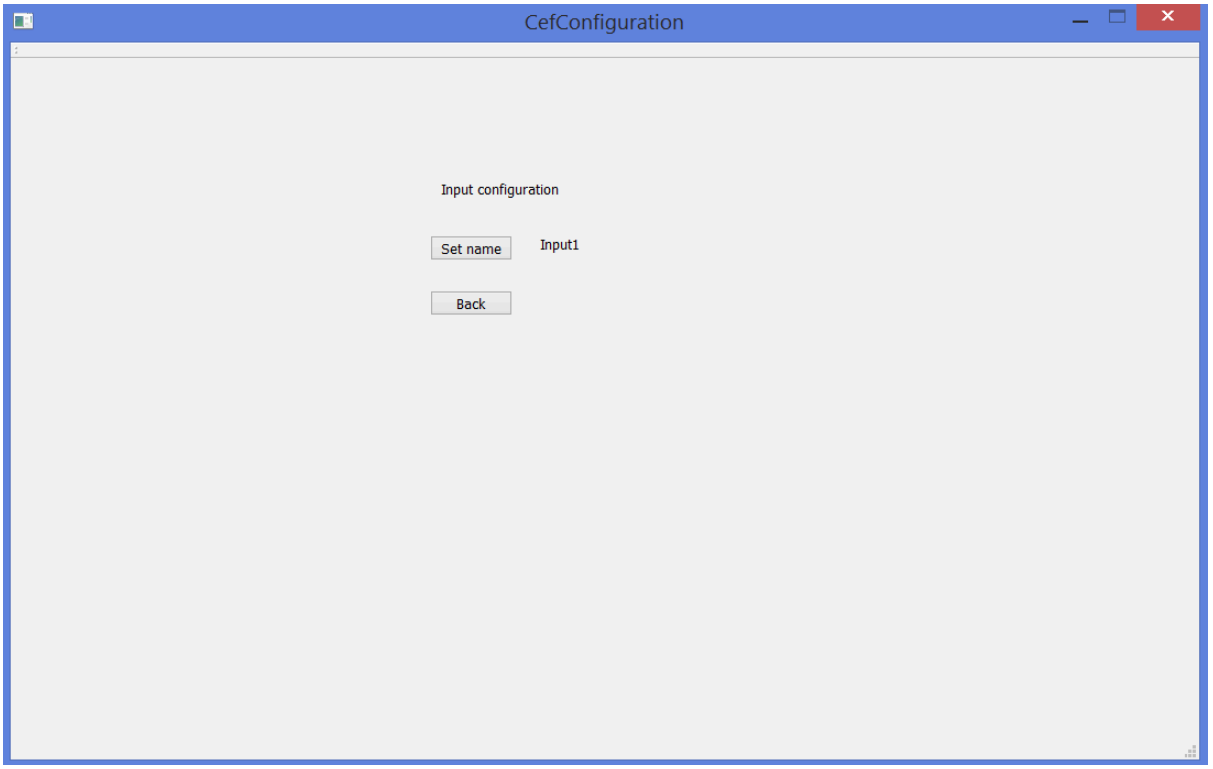
The application mainly consists of Graphical User Interface and core. These were requested by the customer, and found out as best solution for their use case. In order to provide simplicity and portability, we chose to implement these components using Qt. See the sequence diagram below.

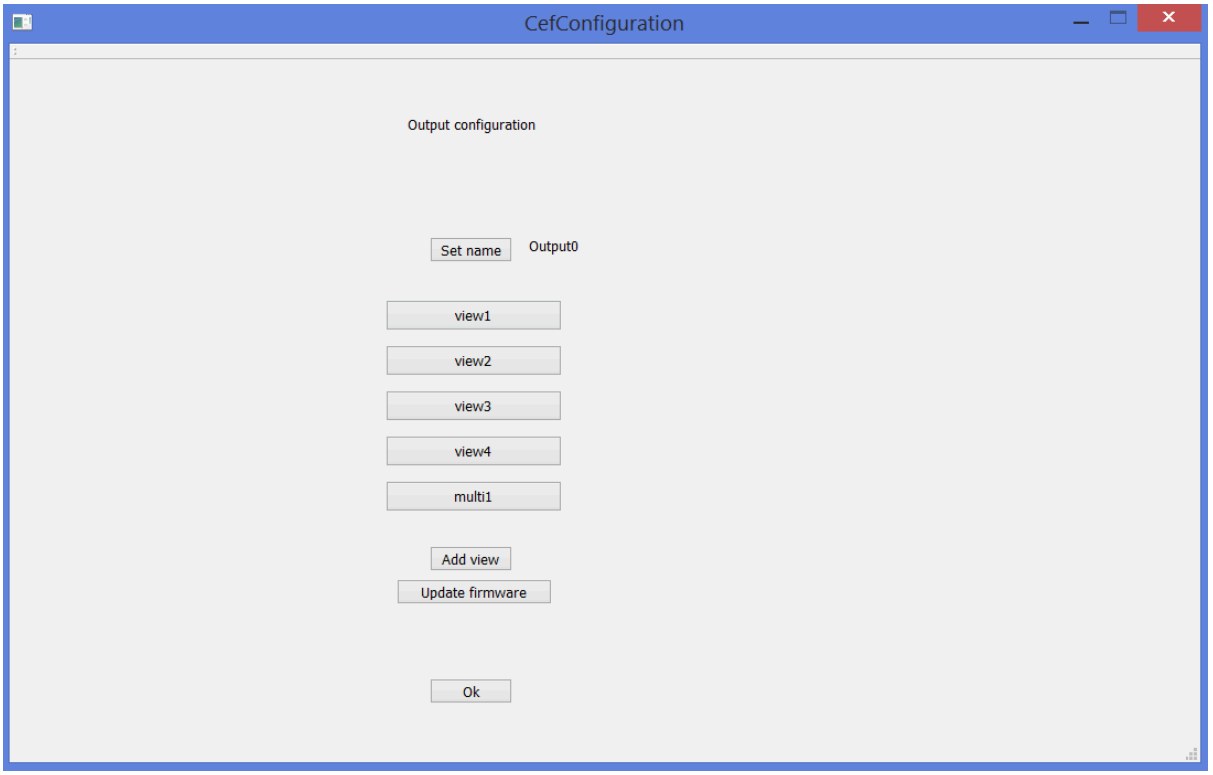


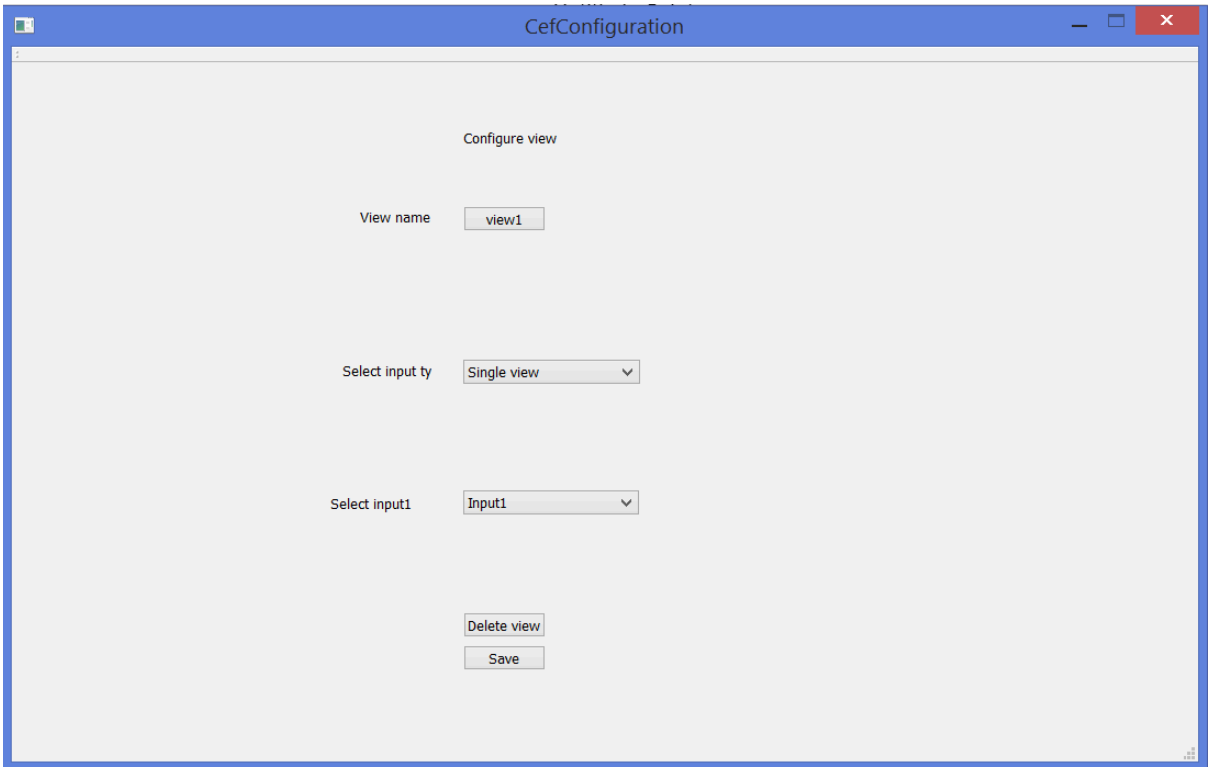
Picture 10. Sequence diagram

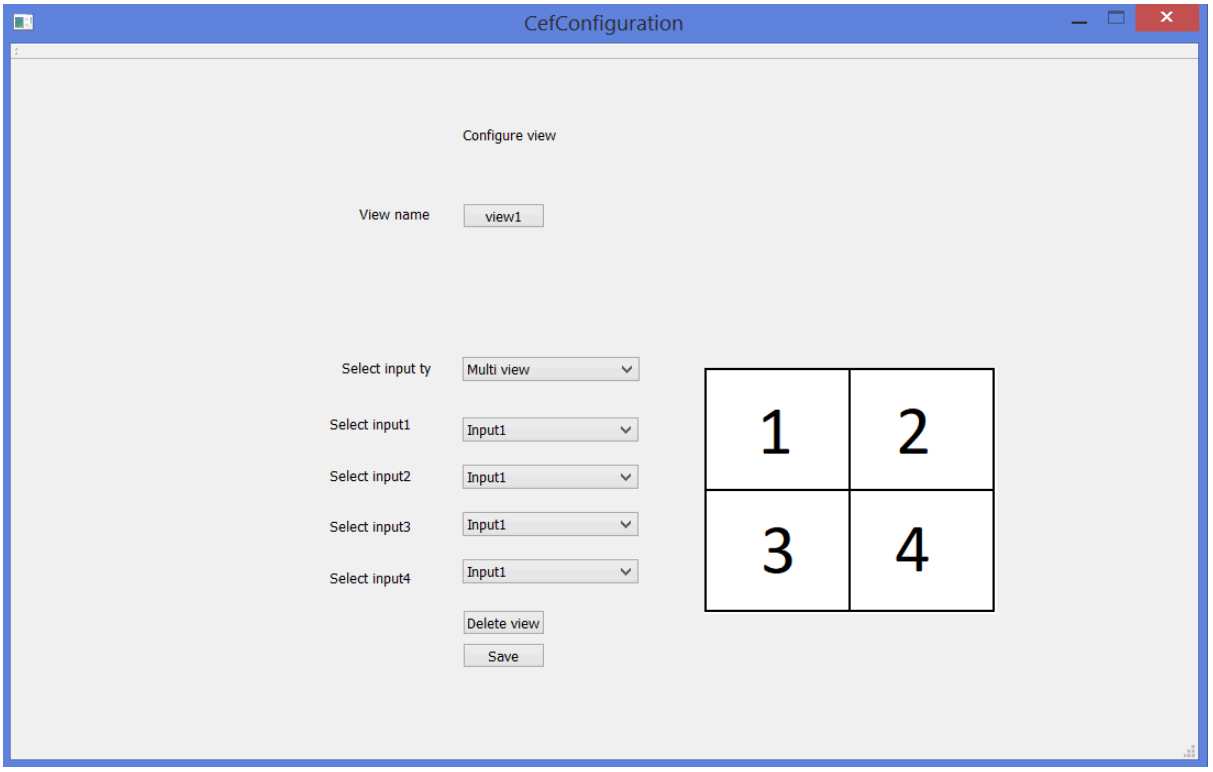
4.1.2 Application screenshots






CefConfiguration
Output configuration
Set name
Output0
view1
view2
view3
view4
multi1
Add view
Update firmware
Ok


CefConfiguration
Configure view
View name
view1
Select input ty
Single view
Select input1
Input1
Delete view
Save



4.1.3 User Engagement

This vision system configuration tool is designed to be essentially used by military staff. The end users will be using the application in the field and also in the office. The implementation of the system is customized for these specific use cases, so the users are expecting simplicity, modifiability and exportability. The application allows end users to work in small and crowded places where the user is not left with complex and long procedures. The UI is very simple where you can navigate easily with keyboard and mouse. Also this configuration tool allows organizing things by naming outputs and inputs, saving different views, importing and exporting configurations. Savox, as a maintenance “users”, are capable of sending some configurations for different end users, which helps Savox to easily respond to different requests from customers.

5 TESTING AND MAINTENANCE

5.1 Public Testing Feedback

Overall feedback was positive. UI is simple, and software is pretty easy to use. Negative feedback was due to lack of any instructions or help within the software, even though every action and button is named properly. There is also no pictures, the UI is really simple.

Software is meant to be used by handful of people, who are in charge of configuring the device. Thus every end user is not doing the configuration. Software would have short manual/quick start guide, in order to tell users how to configure the views. Still, testing was pretty successful with little or no instructions.

5.2 What features did you improve/change after the public testing experience?

After first concept, we used valuable feedback to create second UI mockup, and actual software. When it was ready and we demoed it, it had just minor flaws. It proved to be a good way to start from something that you can show, and then modify it according to customers input.

5.3 Customer Testing Feedback

The first draft of the implementation was presented for the customer for testing purposes, and in general, they were happy about it. There were some relevant things still missing, so they gave feedback on these things: bring option to configure switch, update firmware, file explorer and exporting csv files. These issues were expected to be mentioned, but due to uncertainty we left these features for this stage.

The customer was sent the source code and an executable version of the final configuration tool to test it on their. The following table includes feedback of acceptance testing in regards to the original requirements provided by the customer at the beginning of the project.

Acceptance testing results

ID	Requirement	Customer feedback
1	The application should enable updating the firmware of devices. The devices have different firmwares, so the updates should be done to specific device. Savox will create the actual firmware.	Buttons exists, back-end functionality will be implemented by Savox.
2	The application should enable the possibility to add a new output (screen). There should also be a possibility to update firmware of the new device.	Skipped
3	The application should enable configuring the quad view in a way, that you can choose which inputs are represented on the screen in the desired order.	After save and come back default config is shown.
4	The application should enable to configure screens to other outputs with firmware update.	Not available
5	The application should enable offline use, meaning that the configuration can be made and sent to Savox in a different occasion.	Not available
6	The application should enable the naming of inputs and outputs for clarity.	OK, do not remember after closing
7	The application should enable adding a new camera or screen to a existing configuration. This would be done by taking firmware from another device and trying to update it to the new one.	No highlight
8	The application should enable the user to show something on the selected screen, so that the screen could be recognized.	Skipped
9	The application should enable to change the quad view to a new screen	Not available, some bugs that cannot modify multiview type

10	The application should enable the quadview to be configured to have different options with two small screens and one larger.	Not available, some bugs that cannot modify multiview type
11	The application should enable sending email to Savox if the firmware update would fail for some reason.	Can export CSV, but not directly use any email client. Export format needs to adjusted.

6 CONCLUSION

6.1 Learning / reflections to the development process

6.1.1 Reflections to the project management

Project management was key issue for us. We did not manage time well, we had problems communicating, and meetings were not held often enough. This lead to uninformed team members. Part of it also came from the fact that we started the project late, as we formed the group and got the topic later than other groups. Other reason was bad course description, as there was no clear list of things wanted. Plans was required, but there were no info what is actually required. Graphs, diagrams, journey maps, architectural plan, class diagram, flow maps or something else? This also lead to unefficient work, as goal was to do “something” that might be suitable, without actual knowledge.

6.1.2 Reflections to the customer interface

Another big issue was customers specs. They did not know what they wanted, and as the project went on, demands changed. Most of our questions were not answered properly, and it seemed that they answered how they felt at that very moment, without any bigger consideration.

If you work independently, you or your group usually has idea what they are doing. This time we didn’t have this, as we did not exactly know what customer wanted.

However, the real customer gave us some insight how to communicate with customers, in order to come up with consensus with the vision. Also requirements gathering and grooming became more transparent as we get to discuss about the uncertainties in different stages of the development cycle.

Moreover, a real customer also put some time pressures on work and the need of delivering something every once in a while.

6.1.3 Reflections to the technical development aspects?

We learned completely new language, Qt, which ended up being pretty useful for software like this. Secondly, we learned more on using diagrams and planning the development better. Also, we got some insight on what is really important and core things in development of an application. Having just list of requirements, wouldn't be enough, and communication is major key. However, creating minimum viable product with core functionalities is a good starting point to develop on top of by communicating and testing in iterations. In addition, some revision on C and C++ was done.

6.1.4 What are the future plans for your project?

On our side, project will be seized. We have made prototype that is accepted by customer, and according to them it will be useful for their further integration of the system to the hardware. It will remain to be seen if the UI sees daylight in real world solutions.

6.1.5 Reflect what did you learn as a team from the course

As a team, we learned how to manage software projects. Also we got insight into how to develop applications with close collaboration with real life customers in practise. Taking user experience design as part of the development process was also a new thing for us.

6.2 Summary

In this report we have described the implementation of vision system configuration tool. We started the process by gathering some initial requirements from the customer and by having discussions in the team. After that we did map some user experience by profiling users with personas and analyzing them through journey maps, then creating a prototype of the product. The first prototype generated some feedback on which we based some of our architectural decisions. We designed the

architecture of the system, and developed minimum viable product to demo for the customer. At the end, we did a test plan and manually tested the application.

REFERENCES

We advise to use Zotero or Mendeley to handle your references.

1. Gonzalez, R.C., Woods, R.E., Digital Image Processing, 3rd edition, Prentice-Hall, Inc., USA, 2002.
2. Hamouz, M., Kittler, J., Kamarainen, J.-K., Paalanen, P., Kälviäinen, H., Matas, J., Feature-based Affine-invariant Localization of Faces, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol. 27, No. 9, September 2005, pp. 1490-1495.
3. Liu, Y., Handroos, H., Alkkiomäki, O., Kyrki, V., Kälviäinen, H., Development of a Hybrid Position/Force Controlled Hydraulic Parallel Robot for Impact Treatment, *Proceedings of the 7th International Conference on Machine Automation (ICMA2008)*, September 24-26, Awaji, Japan, 2008, pp. 41-47.
4. Jang, J.-S. R., Sun C.-T., Mizutani E.: Neuro-Fuzzy and Soft Computing, 1st edition, Prentice-Hall, Inc., USA, 1997.

(References are either in alphabetical order (according to the first author) or in the order of citing. Typically, for the reader, the alphabetical order is easier and more informal. All references are in one list only. All bibliographic information is included. If a reference is a project then the type of the project is included in the reference, i.e. mention either “bachelor’s project”, “master’s project” or “doctoral project”

APPENDIX

Test cases

Test case id	1
Test priority	High
Module name	Input configuration
Test Designed by	THO
Test Designed Date	22.4.2019
Test Executed By	Nidal Abu Raed
Test Execution Date	25.4.2019
Test Title/Name	Input naming
Test Summary/Description	This test consists of naming an input
Pre-conditions	The input hasn't been modified before
Dependencies	-
Test steps	1. Choose a input 2. Select "Set name" 3. Give the input name and click ok
Test data	-

Expected result	The input name is saved to the selected input and is displayed on the other screens
Post-condition	The input name should have the given value
Actual result	Inserted value for input is saved and is displayed on the other screens
Status (Pass/Fail)	Pass
Notes/Comments/Questions	There should be a few different test runs with different inputs and different names (special characters etc.)

Test case id	2
Test priority	High
Module name	Update
Test Designed by	THO
Test Designed Date	22.4.2019
Test Executed By	Nidal Abu Raed
Test Execution Date	25.4.2019
Test Title/Name	Update firmware
Test Summary/Description	This test involves choosing a file which upgrades the firmware to the chosen device
Pre-conditions	You have to be in the device configuration screen to be able to choose the firmware upgrade
Dependencies	-

Test steps	1. Select a device (input/output/switch) 2. Select “Update firmware” 3. Select a file and press ok
Test data	-
Expected result	The program should return a message from a successful selection of upgrade file
Post-condition	-
Actual result	The system allows users open a upgrade file and returns message of successful selection
Status (Pass/Fail)	Pass
Notes/Comments/Questions	This is the most important feature for the customer

Test case id	3
Test priority	Medium
Module name	Main screen
Test Designed by	THO
Test Designed Date	22.4.2019
Test Executed By	Nidal Abu Raed
Test Execution Date	25.4.2019
Test Title/Name	Export to CSV
Test Summary/Description	
Pre-conditions	Configurations made by the software
Dependencies	-

Test steps	1. Change a few input and output names 2. Configure some inputs to views 3. Go to the main screen and choose “Export ot csv” 4. Compare the choices made in the program to the ouput csv
Test data	-
Expected result	The choices match the output file
Post-condition	-
Actual result	The output file contains correct configurations
Status (Pass/Fail)	Pass
Notes/Comments/Questions	

Test case id	4
Test priority	Medium
Module name	Output configuration
Test Designed by	THO
Test Designed Date	22.4.2019
Test Executed By	Nidal Abu Raed
Test Execution Dated	25.4.2019
Test Title/Name	Output naming
Test Summary/Description	This test involves the naming of different outputs

Pre-conditions	-
Dependencies	-
Test steps	1. Choose an output 2. Select “Set name” 3. Change the output name and choose ok
Test data	-
Expected result	The name should be saved for the time the program is running
Post-condition	-
Actual result	The output name is saved
Status (Pass/Fail)	Pass
Notes/Comments/Questions	

Test case id	5
Test priority	Low
Module name	Output configuration
Test Designed by	THO
Test Designed Date	22.4.2019
Test Executed By	Nidal Abu Raed
Test Execution Date	25.4.2019
Test Title/Name	Add view
Test Summary/Description	This test involves adding new views for output

Pre-conditions	-
Dependencies	-
Test steps	1. Navigate to some outputs configuration 2. Select “Add view”
Test data	-
Expected result	The program should add a new view the the end of the queue (starts from top to the bottom)
Post-condition	-
Actual result	When clicking “Add view”, a new view at the end of the queue appears
Status (Pass/Fail)	Pass
Notes/Comments/Questions	This test should also include different situations with the existing views when doing multiple tests.

Test case id	6
Test priority	Low
Module name	Output configuration
Test Designed by	THO
Test Designed Date	22.4.2019
Test Executed By	Nidal Abu Raed
Test Execution Date	25.4.2019
Test Title/Name	Delete view

Test Summary/Description	This test includes the functionality of deleting a view.
Pre-conditions	Existing views to delete
Dependencies	-
Test steps	1. Navigate to some outputs configuration and select a view 2. Select “Delete view”
Test data	-
Expected result	The selected view should be deleted
Post-condition	
Actual result	Only the first output’s (from left) views can be deleted
Status (Pass/Fail)	Fail
Notes/Comments/Questions	This test should also include different situations with the existing views when doing multiple tests.

Test case id	7
Test priority	Low
Module name	Configure view
Test Designed by	THO
Test Designed Date	22.4.2019
Test Executed By	Nidal Abu Raed
Test Execution Date	25.4.2019

Test Title/Name	Set view name
Test Summary/Description	This test involves changing the views name
Pre-conditions	There are existing views to be selected
Dependencies	-
Test steps	1. Navigate to a view from some of the outputs 2. Select “Set name” 3. Change the name and press OK
Test data	-
Expected result	The name should be changed until the program has been restarted
Post-condition	-
Actual result	The name of the view is saved succesfully
Status (Pass/Fail)	Pass
Notes/Comments/Questions	What happens if the name is changed multiple times?

Test case id	8
Test priority	Low
Module name	View configuration
Test Designed by	THO
Test Designed Date	22.4.2019
Test Executed By	Nidal Abu Raed
Test Execution Date	25.4.2019

Test Title/Name	Views input configurations
Test Summary/Description	This test involves the configurations of views inputs
Pre-conditions	Existing view
Dependencies	-
Test steps	1. Navigate to view configuration 2. Select single or multiview 3. Configure input(s) to the view
Test data	-
Expected result	The given choices should be saved as long as the program is running
Post-condition	
Actual result	The view configuration did not save properly
Status (Pass/Fail)	Fail
Notes/Comments/Questions	