

Resource Planning Tool for Nonconformity Projects at Wärtsilä

Alexander Rex

Degree Thesis for Bachelor of Engineering

Degree programme in Industrial Management

Vasa 2019



BACHELOR'S THESIS

Author: Alexander Rex
Degree Programme: Industrial management and Engineering
Supervisor(s): Ralf Nybond – Wärtsilä Services
Mikael Ehls - Novia University of Applied sciences

Title: Resource Planning Tool for Nonconformity Projects at Wärtsilä

Date April 25, 2019 Number of pages 42 Appendices 0

Abstract

The thesis presents the development of a Resource Planning Tool that will be used for non-conformity projects for the Customer Assistance department at Wärtsilä. Nonconformity projects are service-related projects that handle products with quality defects. The purpose of the tool is to collect project data and improve future project handling. The task is to create a tool in which it is possible to create an overview per project in which booked resources can be inserted. The result should give the user a good overview of the time schedule, but also to get a better understanding why different delays occur. There has been a demand for this type of tool for some time, but there has not been any successful implementation yet.

A type of dynamic chart will also be created, in where the users can choose to see how many bookings are actually successful and how many that are not. Also, reasons for why they were not successful should be displayed. A well-made and implemented tool could build a better connection between Customer Assistance and Field Services at Wärtsilä.

The result of this tool was an Excel-tool that fulfilled the purpose and requirements of this thesis. The development resulted in a functional tool that can be used for projects in the Customer Assistance department. Projects can be customized, bookings can be added and statuses for these bookings can be inserted. On the home page a functional chart displays the data the user chooses to display.

Language: English Key words: resources, delays, projects

EXAMENSARBETE

Författare: Alexander Rex
Utbildning och ort: Produktionsekonomi, Vasa
Inriktningsalternativ/Fördjupning:
Handledare: Ralf Nybond, Wärtsilä
Mikael Ehrs, Yrkeshögskolan Novia

Titel: Resursplaneringsverktyg för projekt angående kvalitetsdefekter vid Wärtsilä

Datum 25.4.2019 Sidantal 42

Bilagor 0

Abstrakt

Detta examensarbete presenterar utvecklingen av ett resursplaneringsverktyg som kommer att användas för projekt angående kvalitetsdefekter vid Wärtsilä. Syftet med verktyget är att samla all projektdata och förbättra framtida projekthantering. Uppgiften var att skapa ett verktyg i vilket det är möjligt att skapa en överblick per projekt där bokade resurser kan insättas. Verktyget skall ge användaren en bra överblick över tidsschemat, men även en bättre förståelse över varför olika förseningar uppstår. Det har funnits ett behov av denna typ av verktyg för en tid, men det har ännu inte gjorts en lyckad implementering av ett verktyg av denna typ. En typ av dynamisk graf kommer också att skapas, i vilken användare kan välja att se hur många av bokningarna som påbörjas i tid och hur många som inte påbörjas i tid. Även orsaker över varför bokningar inte påbörjas i tid skall synas. Ett valgjort verktyg kan bygga ett bättre samarbete mellan Customer Assistance och Field Services vid Wärtsilä.

Resultatet av detta verktyg blev ett Excel-verktyg som uppföljde syftet och kraven för detta arbete. Utvecklingen resulterade i ett funktionellt verktyg som kan användas för projekthantering vid Customer Assistance. Projekt kan skräddarsys, bokningar och olika statusar kan läggas till. På första sidan av verktyget syns en graf som visar det data som användaren väljer att se.

Språk: Engelska

Nyckelord: resurser, bokningar, projekt

Table of Contents

1	Introduction	1
1.1	Background.....	1
1.2	Problem Area	1
1.3	Purpose	2
1.4	Delimitations.....	2
2	Wärtsilä Services	3
3	Theory.....	4
3.1	Elements of a Software Development Paradigm.....	4
3.1.1	Requirement Gathering.....	4
3.1.2	Conceptualization.....	5
3.1.3	Project Representation.....	5
3.1.4	Project Implementation	6
3.1.5	Project Testing.....	6
3.2	Waterfall Software Development Process.....	7
3.3	Basics of Project Management.....	8
3.4	Time Schedule.....	8
3.5	Monitoring Progress	9
3.6	Excel VBA and Pivot Tables	9
4	First Phase of Development	11
4.1	The Basic Idea.....	11
4.1.1	Use	11
4.1.2	Users	12
4.1.3	Platform.....	12
4.1.4	Home Page.....	13
4.1.5	Project Manager Page	13
4.1.6	Project Page	13
4.1.7	Booking and Statuses	14
4.2	First Prototype	15
4.3	Continued Development.....	17
4.3.1	Code Used for Buttons	17
4.3.2	Home Page and Manager Page	22
5	Second Phase of Development	23
5.1	Data Sheet.....	23
5.2	Adding of Bookings and Statuses	23
5.2.1	Adding Bookings	24
5.2.2	Add Statuses	26

5.3	Add Managers and Projects.....	27
5.4	Navigating Between Sheets	30
5.5	Testing	31
6	Final Phase of Development.....	33
6.1	Charts for Homepage	33
6.2	Go to Function.....	34
6.3	Delete Statuses	35
6.4	Further Improvements.....	37
7	Result	38
8	Discussion	40
8.1	Summary.....	40
8.2	Conclusion.....	41
9	References.....	42

Table of Figures

Figure 1 – Waterfall software development process	7
Figure 2 - Basic layout for Project page	15
Figure 3 - Formatting of cells	16
Figure 4 - First prototype	17
Figure 5 - Form for inserting installations.....	18
Figure 6 - Code for adding installations.....	18
Figure 7 - Form for adding weeks to project.....	19
Figure 8 - Code for adding weeks	20
Figure 9 - Buttons for adding resources and bookings	21
Figure 10 - Code for adding resources	21
Figure 11 - Description of how the code for resources work.....	22
Figure 12 -Data in the data sheet.....	23
Figure 13 - Form for adding bookings or delays/statuses	24
Figure 14 - Form for adding bookings	24
Figure 15 - Code for adding booking.....	25
Figure 16 - Form for adding delays/statuses.....	26
Figure 17 - Code for inserting delay/status.....	26
Figure 18 - Form for adding managers.....	27
Figure 19 - Code for adding managers	28
Figure 20 - Form for adding projects.....	29
Figure 21 - Code for adding projects	30
Figure 22 - Code for sheet navigation.....	31
Figure 23 - Code for navigating away from project page	31
Figure 24 - Code for displaying the homepage when file is opened	31
Figure 25 - Charts displaying statuses on Home page.....	33
Figure 26 - Pivot table for charts	34
Figure 27 - Code for updating data in pivot table	34
Figure 28 - Userform for Goto function.....	35
Figure 29 - Code for Go to function.....	35
Figure 30 - Userform for deleting statuses	36
Figure 31 - Code for deleting statuses	37
Figure 32 – Home Page of the tool	38
Figure 33 - Manager Page of the tool	39
Figure 34 - Project Page with example bookings and statuses.....	39

1 Introduction

The introduction will present an overview and background of the thesis followed by a presentation of the theory and development. This thesis has been done on the behalf of Wärtsilä. The goal of the thesis is to implement a resource planning tool in order to collect data and improve future bookings, but also to improve collaboration with the Field Service department at Wärtsilä. A demand for this type of tool has existed for some time. In this chapter, background and problem area will be explained followed by a walkthrough of the purposes.

1.1 Background

This Bachelor's Thesis is made on behalf of Wärtsilä Services, specifically for the Customer Assistance Product Center Engines Project Department. The responsibilities of this department are to manage service-related projects for both the Marine Solution and Energy Solution sectors. There are currently five project managers that all have responsibility for their own projects. These projects can differ a whole lot from each other. Some projects may be small and can be quickly done, some can be very time consuming with multiple installations. The department handles various service projects in different countries all over the world. These projects need manpower, tools and parts to be done within given budget and timeframe. Manpower and tools are ordered from field service who then sends the required workforce or items to the site at the decided date. When managing a project on a vessel or ship, timing gets even more important due to the fact that these vessels have time schedules that need to be kept.

When Wärtsilä offered the opportunity to develop a resource planning tool I understood the basics of what should be needed from the tool. Because of previous work experience at Customer Assistance I had a good view of what work they were doing and who I could ask for help when I needed it. I had some experience of delays in projects from before and to some extent understood what the problem area was before given the chance to dig deeper into the problem.

1.2 Problem Area

Resources are ordered from field services, but bookings tend to be delayed by various reasons. Delayed bookings lead to the projects not being finished in time, which means

delays that lead to increased project costs. Customer Assistance cannot control if resources get delayed or not, but to improve future resource booking we can collect data from bookings in order to display where, how and if they were delayed. The main problem is that Customer Assistance often get blamed for not finishing projects in time when the delay often is beyond their reach. Customer Assistance do not have any documentation to prove this and that is the problem area that this thesis will focus on. There has been a demand for a resource documentation tool for a long time, but there has not been any successful implementation of a tool yet.

1.3 Purpose

The purpose of this thesis is to create a tool in which it is possible to create an overview per project in which booked resources can be inserted. One of the main purposes for this tool is that it should give the user a good overview of the time schedule, but also to get a better understanding why different delays occur. A type of dynamic chart will also be created in where the users can choose to see how many bookings are actually successful and how many that are not. Also, reasons for why they were not successful should be displayed. A well-made and implemented tool could build a better connection between Customer Assistance and Field Services at Wärtsilä. A closer relationship between departments will also lead to less delays and a better result. To this point there has not been any common place to gather this information. All the projects managers all have their own way in which they handle projects. Implementation of this tool could give a common ground for all users and projects to display project history, project planning, project documentation and to provide a visual overview.

1.4 Delimitations

Customer Assistance ordered this thesis to get a better view over their projects and to have a common place for all booked resources per project. The tool is custom made for Customer Assistance only and therefore no other departments will be taken into consideration when developing the tool. The tool should not be used as a booking tool and will not have any connection to field services. This tool will and should only function as a sort of database for all bookings that are made in order to then display which problems often occur and then further steps can be taken to avoid them in future projects. Managers should use this tool as a timeframe and as a helping tool for better and smoother project management in the future.

2 Wärtsilä Services

Wärtsilä is a Finnish corporation established 1834 in Tohmajärvi, Finland. It all started out as a small saw milling company, but Wärtsilä has changed its markets many times since. Today Wärtsilä is a global supplier of advanced technologies and complete lifecycle solutions towards the Energy and Marine markets. The company is well positioned on the global market due to its integrated offering of both services and products. Industry leading multiple fuel products combined with a superior global service network helps Wärtsilä reach their main objective: to achieve growth by offering their customers new and innovative solutions.

Wärtsilä Services is the biggest department within Wärtsilä with 45% of the net sales in 2017. Services provides high-quality services that enhance customers business. Every third of the worlds ships is driven by a Wärtsilä engine and every second ship is maintained by Wärtsilä Services. This means that even ships that is not driven by a Wärtsilä engine still gets maintained by Wärtsilä Services. Services consist of many smaller departments that handle different types of services and in different areas of the world. Wärtsiläs organization was set up like this in 2018, but reorganizations led to changes in early 2019. As of January 2019, Wärtsilä consists of two businesses; Marine Business and Energy Business. Services business has been incorporated into Marine and Energy Businesses.

Energy Business offer internal combustion engine-based power plants and solar power plants, as well as LNG terminals and distribution systems. Marine Business offer maneuvering systems for vessels including multiple fuel engines, thrusters and propellers, but also maintenance agreements and complete ship designs. (About Wärtsilä, 2019)

When the work for this thesis started, the department Customer Assistance was a part of Services, Delivery Management, Product Center Engines. After reorganization in the beginning of 2019 Services was integrated into Marine and Energy business units and Customer Assistance is now listed under Marine, Businesses, Assets Management, but still maintained its work tasks. Customer Assistance mainly consists of project managers, but also a couple of project engineers and a project controller. I had been a summer trainee (Project Engineer) for four months during the summer of 2018 and therefore I already had knowledge about the work and was familiar with my colleagues. During 2018, Customer Assistance was a part of Wärtsilä Services and is mainly focusing on non-conformities and quality issues for both the Marine and Energy sectors. The department is based in Runsor, Vasa and in Turku. Customer Assistance is divided into two major groups with one focusing

on operative warranty issues and the department that this thesis is made for are focusing more on single target quality issues.

3 Theory

In order to create a resource planning tool software that is user friendly and incorporates all requirements, it is necessary to have a stable theoretical foundation to stand on when the development start. The theoretical foundation is based on literature about software development, resource planning and Excel usage. This chapter will present the theoretical foundations of the tool, such as software development, project management, project planning and Excel functions.

3.1 Elements of a Software Development Paradigm

Software development is a lot like other types of engineering, you need to make a plan, follow that plan, overcome obstacles and end up with the right outcome. The software development process refers to the series of steps one goes through to develop a software system. By software development one means the entire conceptualization, representation and implementation process. A software paradigm is defined by techniques we use in developing software and getting organized. In this chapter the elements of the software development process will be presented and explained. (Stiller & Leblanc, 2002, p. 7)

Software engineering is a lot like other types of engineering, certain tasks must be accomplished in order to end up with the right product. (Stephens, 2015, p. 3)

3.1.1 Requirement Gathering

Requirements gathering is a vital part in any software project. If you get the requirements wrong, the resulting application will not solve the user's problems. You may go somewhere, but probably not where your aim was set. Requirements gathering should be the first link when developing the software. The requirements will be used as a type of foundation that involves the most critical functions of the software and ideas on how things should be solved. (Mathaisel, Manary, & Criscimanga, 2013, p. 76)

Depending on the projects scope and complexity, we might need only a few requirements, or we might need hundreds of pages of requirements. Good requirements are clear concise and easy to understand. They should not be full of management speak and confusing terms

and language. They should not either be vague or ill-defined. The requirements should be worded so that you understand what it requires, then you can build a system to satisfy it. This can seem obvious to any requirement, but it can be more difficult to guarantee than you think. For example, to use the word “best route” in a street map application. No one knows what the best route is, is it the shortest? Or the fastest? Or the one with the best roads? To try and make requirements unambiguous is something you should have in mind while setting the requirements. (Stephens, 2015, pp. 54-55) (Stiller & Leblanc, 2002, p. 73)

3.1.2 Conceptualization

Conceptualization of the project means how the software developer thinks about the problem. This could as simple as a thought or an illustration of the solved problem. The conceptualization of a system describes the mental constructs that the developer use in order to organize their thoughts and discussions of the project. (Stiller & Leblanc, 2002, p. 7)

You can make “go/no go” decisions at any step in the project. At this stage the decision you make is a big one. After concept development, you should have enough information to make an informed decision. After this point, it gets harder to cancel the project as it starts to staff up and gain momentum towards the chosen way of finishing the project. (Stephens, 2015, p. 277)

3.1.3 Project Representation

The writing down of the project conceptualization is the project representation that will be used in the software development paradigm. The representation must convey what the project is about in an efficient and unambiguous manner. When creating such a representation, we agree on a set of rules for how to represent various pieces of the project. These rules are called notation that we are using for our representation. The notations specify a view of the software to be developed. The notations function is to represent the system in a manner that is unambiguous and unaffected by the background knowledge of the person viewing the characterization. Ideally the notation should be a representation of the system that means the same thing for both the expert as for the future users of the software. (Stiller & Leblanc, 2002, p. 9)

3.1.4 Project Implementation

Project implementation addresses how the source code that will make up the software is structured. All programming languages use some sort of mechanism for creating units of programming languages instructions. These units are called modules and these modules will work together in the final software system in order to meet the needs of the users.

Modularity is a feature that is desirable in a well-made software. Ideally, the mechanisms for modularity in the project implementation are the same as the elements that comprised the project conceptualization and representation. If the software is built in stages one upon another it will allow us to move back and forth between the various phases of the object-oriented development process if needed. We may need to analyze part of the system to be developed, then design a solution for that part of the system, and then move back to analyze the next phase of the system. This ability to move back and forth between development phases carries over into the implementation and testing phases as well. In fact, there is in no way to separate the implementation phase from the testing phase. (Stiller & Leblanc, 2002, p. 10)

3.1.5 Project Testing

In a perfect world every written code works as the programmer want it to work and the system runs smoothly. Unfortunately, this is not the case in an everyday situation. The number of situations in which the system must work perfectly is very large. In order to achieve a working system, we need to test regularly to see if the code actually does what it is supposed to. The goal of testing is to eliminate all the small faults in the program before turning it over to the user and to determine how close to the ideal the system is. Any deviation from the ideal indicates the presence of a bug in the system that must be fixed. (Stiller & Leblanc, 2002, p. 263)

A test plan specifies how we will demonstrate that the system is free from faults and behaves according to the required specifications. There are several levels of testing such as unit testing, integration testing, and system testing, and the test plan is a guide to the entire testing process, just as the implementation plan is a guide to the entire implementation process. It is important to evaluate performance, uncover deficiencies, and implement corrective action while the system is still being developed. (Mathaisel, Manary, & Criscimanga, 2013, p. 400)

A good way of testing is to let someone else test your code, because obviously you did not put a bug there on purpose and therefore it might be hard to notice by yourself. When a piece

of code seems to work properly it is inserted to the system, and if not the programmer has to dive back in to figure out what is going wrong. (Stephens, 2015, p. 7)

3.2 Waterfall Software Development Process

One of the most common software development processes is the Waterfall development process. This process is a traditional development process due to its traditional essential phases namely, requirements analysis, design, programming, testing and maintenance. These are the basic phases of every software development, independent of software paradigm. Since each software development process shares the same basic phases, the difference between the various process approaches is in the duration, completeness, and sequence of the development process. Other phases such as quality assurance and risks assessment may be specified as separate phases or can be integrated in the basic phases. (Stiller & Leblanc, 2002, p. 43)

Because of the many bugs and small adjustments, a software developer has to make during the development of a new software, this method might not be the most efficient way of working. (Mathaisel, Manary, & Criscimanga, 2013, p. 65)

The Waterfall model assumes that you finish one phase completely before going on to the next one. The model is very straight forward, therefore it can be hard to get back to the right track if some mistakes are made during the “waterfall”. A more modern version of this model is the waterfall with feedback model that allows feedback after every step, in order to make changes before taking on the next step in the development. (Stephens, 2015, p. 271)

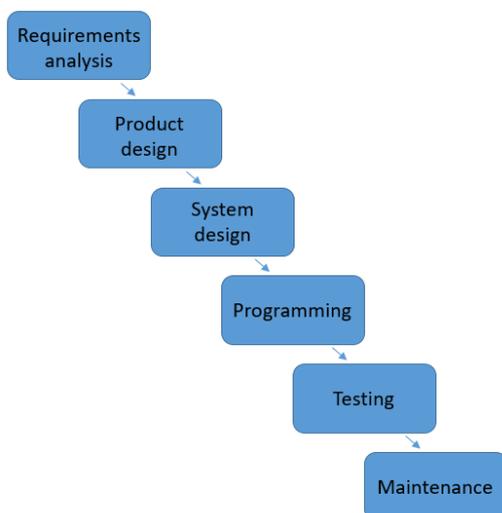


Figure 1 – Waterfall software development process

3.3 Basics of Project Management

In this chapter the basics of project management will be described. Projects can be defined as a deliverable that once delivered and accepted; the project is over. There are often project teams that take on the project. In these teams there are always a project manager that lead the project toward its goal. The project management process includes the following: planning, initiation, execution, monitoring, and closing. The objective of effective project management is to guide the development of a project within reasonable time and cost parameters, ensuring that the result of the project is of the highest quality. There are also different industries that have developed specialized project management templates that are specific to the needs of their respective fields. (Stiller & Leblanc, 2002, p. 293) (Marion, 2018, p. 4)

For example, IT project management specializes in facilitating the process of delivering technical products that pass through several life-cycle stages including development, testing, and deployment. The project manager does not necessarily need to be an expert in the field; however, these skills can be helpful because the project manager often is the connecting link between the customer and the project team. The project manager can be described as the most important role in any project, since they have several areas of responsibility the manager needs to take the primary leadership role and guide the project to be successful. (Stephens, 2015, p. 31)

3.4 Time Schedule

The earlier planning steps of activities, estimation of resources and time needed, and analysis of dependency – form the basis for drawing up the time schedule. The most common way to draw up a time schedule is by using the Gantt chart. By using a Gantt chart one can easily illustrate how a process was developing time-wise in relation to the previously set schedule. A Gantt chart presents the activities as horizontal bars, while the x-axis works as a timeline. A Gantt chart can also illustrate the dependencies between different activities by using arrows between the activities. Another useful feature of a Gantt chart is to display milestones and decision points. By using colors, texts and symbols in order to show how is doing what and how the activity is developing. (Hallin & Gustavsson, 2012, p. 87) (Stephens, 2015, p. 41)

A Gantt chart should not be too detailed but not basic either. A main point of a chart is for it to be easy to understand and to provide a good visual overview of the activities of the project.

With the creation of the Gantt chart, the project manager now has a complete schedule where activities and important milestones. (Marion, 2018, p. 128)

3.5 Monitoring Progress

The best way to ensure that progress is being made is to define milestones with associated deliverables. Milestone should be created so that the developer receives sufficient feedback at regular intervals. If the milestones are not completed in time the developer can make necessary adjustment to the project plan. By having frequent feedback sessions, the developer can avoid spending time and resources on the wrong things. By getting frequent feedback it is easy to keep the development “on track” and to also get new ideas on how to improve existing versions of the product. To get feedback from users is probably the best way forward in software development, since it is the users that will use it and they know what is required for the tool to be successfully implemented. (Stiller & Leblanc, 2002, p. 317)

3.6 Excel VBA and Pivot Tables

Microsoft Excel is a part of office 365 and is a spreadsheet program that is used to record and analyze numerical data. In order to fully understand this thesis, some knowledge about Excel VBA is needed. Excel VBA or Visual Basic for Applications is a type of programming code that Excel use to execute any number of actions. VBA is built by macros, these macros can be recorded or even written manually. Normally macros are recorded and when recording a macro Excel saves the steps that the user do manually and saves it as a code in VBA. Many people call any code in VBA a macro even if it is written manually. To record a macro is similar to saving a phone number, first you have to insert it manually, but when it is saved the user can just push a button instead of having to insert it again. The path to Visual Basics for Applications also known as VBA often starts with the need to perform tasks that cannot be performed by the standard tools in Excel. These tasks can be different for each of us. The beautiful thing about VBA is that you do not have to be an expert to start solving problems with it. You can learn just enough to solve a particular problem, or you can go further and discover ways to handle all kinds of automation scenarios. Excel programming terminology can be a bit confusing. (Introduction to Microsoft Excel, n.d.) (Alexander & Kusleika, 2016, p. 19)

Pivot tables in Excel is a very useful function when users have to handle a large amount of data and display them in a clear manner. Pivot tables will allow you to quickly organize and summarize data. You can easily filter data and choose which part of the data you want to be visible and how you want it to be displayed. You can pretty much display any type of data in a pivot table. The only requirement is that the data is in an Excel sheet somewhere on your computer. By using slicers, Excel allows the user to choose which data they want to see. A slicer is basically a list of different values that occurs in the data. For example, if you have sales data for vehicle types which includes cars and motorbikes. A slicer will have two buttons, one for cars and one for motorbikes. When the user chooses cars, the data for car sales is displayed and vice versa for motorbikes. This can also be done with dates. A timeline allows users to choose from which timeframe data should be displayed. (Benton, 2017, p. 10) (Holman & Blake, 2018, p. 175)

4 First Phase of Development

After discussions with the team and my supervisor, we came up with which aspects of the tool was the most important and how these potentially could be solved. We also came up with a type of layout of what the tool would potentially look like when finished. In this chapter a description of the development process for the basics of the tool will be presented, as well as thoughts and problems that also will be mentioned. This can be connected to the requirements gathering process described in Chapter 3.1.1.

4.1 The Basic Idea

The first step was to idealize how the software could potentially look. The basic idea consisted of three different pages that would all fulfill different purposes. The tool should consist of one main page where all project data would be inserted, one page for each manager in which projects could be added and one home page where managers could be added and statistics about projects would be displayed. There were also discussions about a forecasting function where the users could insert a forecasted view of how the bookings for the project would be set up.

4.1.1 Use

The tool would be used for forecasting projects, project follow up and displaying statistics. The statistics would display why different bookings were delayed and also what caused the delay. As described in Chapter 3.3, the purpose of project management is to guide the project within reasonable time and cost parameters, ensuring that the result of the project is of the highest quality. During this process this tool can be used as support for keeping track of bookings and how delays has affected the costs and timeframe of the project. From the beginning it was mentioned that user friendliness was important and that a complicated tool would be hard to implement. For this specific tool some guideline requirements where set at this stage to set a type of goal of what the project should lead to.

The most important requirements were:

- User friendliness
- Customization of projects
- Collecting statistics
- All project history in the same place

4.1.2 Users

The users would be all members of the department, which consists of project managers, project engineers and controllers. They are all involved in the projects that the department handles. In order for this system to function properly, all of the projects should be inserted in the tool. The user input in the tool is an essential part of the tool, especially for each project. Every step and booking should be inserted in the project for the charts to display the correct information. There are around nine people that would come in contact with the software and a requirement for them is that they understand it and can use it according to the guidelines.

4.1.3 Platform

The chosen platform was Microsoft Excel. This choice was made due to user friendliness, multi-functionality and previous experience from Excel VBA. More about Excel can be read in Chapter 3.6. Excel is an excellent software when having to deal with a large amount of data which will be needed to collect all inserted statuses and delays for projects. Other platforms were also taken into consideration in the beginning of the development. The strongest candidates to be used for a platform were Retain Planner and Microsoft Projects.

Research about Retain was done by having a discussion with a represent from Energy Solutions. One of the main reasons why Retain was not used was because of the often-occurring time changes of bookings. When a delay occurs the normal way in Retain is to just drag and drop the booking further forward on the timeline. For this specific tool the need of knowing the history of the booking was needed. In retain there would not be any “trace” of the booking on how it has moved and from where. Also, to implement a statistic function that could display all the data at the same time could have been a challenge. Retain planner would also include costs. This means that it would have to work almost perfectly in Customer Assistance for it to be worth the costs. Due to the fact that Retain planner is a complete software it can be hard to customize it in a way that would suit the purpose of this thesis.

Microsoft Projects is a very popular planning software that is currently in use in multiple departments in Wärtsilä. One of the main reasons Projects was not used was because of the user friendliness requirement that was mentioned in Chapter 4.1.1. Projects has a lot of functions that were not necessary for this specific tool and would involve a lot of implementation before it would work as it should. Projects also focuses a lot on costs and details of the projects. This did not fulfill the requirements of an easy to use and straight

forward tool that the aim of this tool was. There was also a plus side to projects which was that you could link different bookings which meant that they would be dependable of each other which could have been a useful feature.

Due to the fact that multiple users need to have access to the file, there will be only one user at a time that can use the tool. The Excel file will be accessible through a software called Teams where the department members can access the file. When a user has updated the file, the updated version will then be available on Teams.

4.1.4 Home Page

A part of the basic idea to create a simple and easy to use “home page” where all project managers are listed. This page would be used as a “home page” that opens every time that someone opens the file. When adding a new manager, a link to that specific manager will be created and the link would lead to their own project page. This page would also have charts displaying how often different delays occur depending on the type of booking. This page should have an active dashboard where users easily can choose what they want to be displayed. This involves all managers, all statuses, types of booking and within which time window. For example, a manager should be able to see how many weeks of delay a missing item caused to a specific project between June and October.

4.1.5 Project Manager Page

When choosing your project manager from the homepage the user will be navigated to the “Project Manager Page”. Here the idea was that the manager could add their own projects by a button. Here the project will be given a name and a link to the empty “project page” is created. All the chosen manager projects will be visible and also charts displaying how often different delays occurs depending on the type off booking for this specific manager.

4.1.6 Project Page

The most complicated page is the “project page” which is accessed from the “project manager” page. This was also the most complicated page which had to include the most programming and customization. The most important feature is to add bookings and statuses for these bookings. The project page had to be easy to customize by installations, resources and weeks. Because of the many functions, a lot of weight will be set on this page to make everything working and to avoid bugs. From the “project page” data must be sent to a data

sheet where all inserted statuses will be listed. Right from the beginning it was understood that this page would be a big challenge due to the fact that a lot of buttons and codes would have to work in harmony with each other. The area where bookings should be inserted was taken from the foundations of theory, Chapter 3.4, about time schedules. Gant charts is a useful set up where it is easy to visualize project progress. The idea of using a type of Gantt chart looking foundation for the “project page” was something that we decided to keep working with.

4.1.7 Booking and Statuses

In the “project page” the users will be able to insert bookings and also statuses for these bookings. There will be two different types of bookings “hard bookings” and “soft bookings”. These are the typical bookings that Customer Assistance makes to order workforce or tools for the projects. “Hard bookings” are bookings that a date has been decided and the booking is important for the whole project to succeed. A “soft booking” is more of an estimate on when the job should be done. Statuses or delays is the current situation for the booking. For example, if a booking gets delayed due to customer delay the status for this booking is delayed due to customer delay. Other statuses can be done, cancelled and other types of delays. These statuses are the foundation for the information that will be collected. Every time a status is inserted a new line with all the relevant will be added to the data sheet. These statuses can easily be changed if there is demand for that to happen. You can delete a status that never really is used, and you can easily add a status if there is need to.

The current statuses for bookings are the following:

- Customer Delay
- Logistic Delay
- Missing Items Delay
- Reassigned Delay
- Visa Application Delay (Visa App.)
- Holidays Delay
- Sick leave Delay
- Other Delay
- Done
- Cancelled

4.2 First Prototype

In this chapter, a prototype of the software was developed. Conceptualization as described in Chapter 3.1.2, “Conceptualization of the project addresses how the software developer thinks about the problem to be solved”, is the approach of this chapter. First part of development was to create a visually good-looking project page where the project interface and functionality will be important factors. In this page there should also be the possibility to create a booking which has to contain the essential booking information. To create a new booking, we need to specify some important information in order to separate bookings from each other. After discussions with the future users and my supervisor we decided which information was crucial for each booking. The information consists of; type of resource, specification of resource, optional comments, type of booking and status of booking. Because of the fact that every project will start at different week the tool also had to include an eligible start date for each project. At this stage this was called “projektstart”. The basic layout for the project page was created as in Figure 2.

Projektstart:		Vecka		1	2
Projektstart:		43		43	44
Installation 1		Specification	Comment, Estimated duration of work (weeks)	Column1	Column2
Team 1				0	0
Team 2				0	0
Team 3				0	0

Figure 2 - Basic layout for Project page

At this moment everything had to be inserted manually which was not efficient and could easily lead to mistakes and miss understandings. To insert information manually into every cell would also be very time consuming. The positives from this prototype is that the Excel function conditional formatting helped a lot in order to use colors for different types of values in cells. This would lead into a valuable input for the visual aspect of a solid project overview. In order to create conditional formatting that suited the tool, all statuses with colors were inserted to the worksheet in a section that would later be hidden for users. The formatting was done by selecting the area where booking would be inserted and then using the code: =OR(“Top left cell of area”=”Range where reasons where listed”). This function can be seen in Figure 3 on the following page.

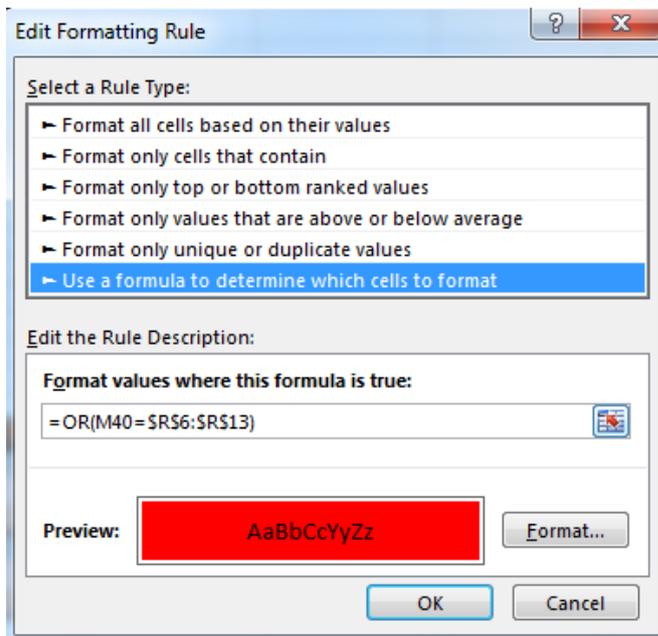


Figure 3 - Formatting of cells

At this point the project was not customizable at all from the user point of view. No projects, resources or weeks could be added. As the next step in the development, customization of projects was created.

The main reasons why customization had to be available

- Installations – The number of installations can vary from hundreds to just one for different projects.
- Weeks – The duration of the project often varies from a month up to several years.
- Resources – Under one installation, multiple resources can be required

It was decided that these functions would have to be as easy to use as possible, therefore it was decided that Excel's Visual Basic Application (VBA) would be used. Thanks to my previous knowledge about VBA I knew that this problem could easily be solved in VBA. Here buttons were added in order to display pop up user forms in which number of weeks, installations or weeks could be added.

Form buttons for adding installations, teams (later resources) and project weeks were added as can be seen in Figure 4 on the following page.

Add installations here					
Add project weeks here					
	Projects first week	35			
					35
		Installation	Specification	Comments, Estimated duration of work (w	1
		Team			0
Add Team	Bookings				
	Reasons				
		Team			0
	Bookings				
	Reasons				
		Installation	Specification	Comments, Estimated duration of work (w	1
Add Team		Team			0
	Bookings				
	Reasons				
		Installation	Specification	Comments, Estimated duration of work (w	1
Add Team		Team			0
	Bookings				
	Reasons				

Figure 4 - First prototype

4.3 Continued Development

After the first prototype was made, discussions of further development started. The most important functions that needed to be added was the functionality to the buttons that was inserted in the first prototype. A home page and a manager page were also important to get into the system. At this point the development turned to more of a project representation phase as described in Chapter 3.1.3. Here we have decided the ground rules and a foundation for the project is set. At this stage the development was more about setting the system alive with codes and functions.

4.3.1 Code Used for Buttons

By using the userform function in Excel VBA, a userform for how many installations the users wanted to add was created as in Figure 5. Here users could decide how many installations that should be added. When clicking Ok the inserted number of installations is added to the project.

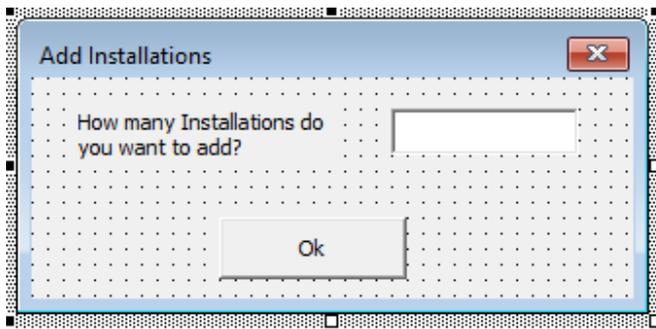


Figure 5 - Form for inserting installations

Code for adding installations

```

Private Sub OK_Click()

Application.Calculation = xlCalculationManual
Application.ScreenUpdating = False
Application.DisplayStatusBar = False
Application.EnableEvents = False
'Shuts down automatic updates
Dim x As Long

For x = 1 To TextBox1.Value
    Debug.Print x & Rows("38:41").Copy
    Range("B39").End(xlDown).Offset(1, 0).EntireRow.Insert

Range("B39").End(xlDown).Offset(-1, 8).Select
ActiveCell.Offset(-2, 0).ClearContents
ActiveCell.Offset(-1, 1).Resize(1, 200).ClearContents
ActiveCell.Resize(2, 200).ClearContents
ActiveCell.Offset(-2, 3).Value = Range("M38")

    Unload InstalationsAdd

Next x
'Copies first installation and inserts it in the right place and clears data

Application.Calculation = xlCalculationAutomatic
Application.ScreenUpdating = True
Application.DisplayStatusBar = True
Application.EnableEvents = True
'Re-enables automatic updates

End Sub

```

```

Private Sub TextBox1_Change()

If TextBox1.Value <> "" And Not IsNumeric(TextBox1) Then
    MsgBox "Please enter Numbers only"
ElseIf TextBox1.Value > 20 Or TextBox1.Value < 1 Then
    MsgBox "Number Outside limits ""1-20""
End If
'Keeps the users input between 1-20 installations

End Sub

```

Figure 6 - Code for adding installations

The code in Figure 6 basically copies the top row of the table and pastes it below the last table row and the deletes all data that was previously inserted. The code had to be able to handle up to ten repetitions without crashing or lagging. In order to prevent this the following code was used in the installation buttons code.

- *Application.Calculation = xlCalculationManual*

Prevents the system to make automatic calculations as the code runs.

- *Application.ScreenUpdating = False*

Prevents the system to update the screen as the code runs.

- *Application.DisplayStatusBar = False*

Prevents the system from displaying everything that is happening on the Statusbar

- *Application.EnableEvents = False*

This line disables events before a file is saved. After the code had run these functions were re-enabled. These lines proved to be crucial for the tool to run as smooth as possible. When trying to run without them, the process was slow when this code had to be able to handle multiple repetitions. To prevent crashes of the system a limitation of amounts of installations that could be added had to be added. As displayed in Figure 6 under Textbox1_Change, the limitations were set in such a way that only between one and ten projects could be added at a time. By using a similar code, a user form to add weeks was created which can be seen in Figure 7.

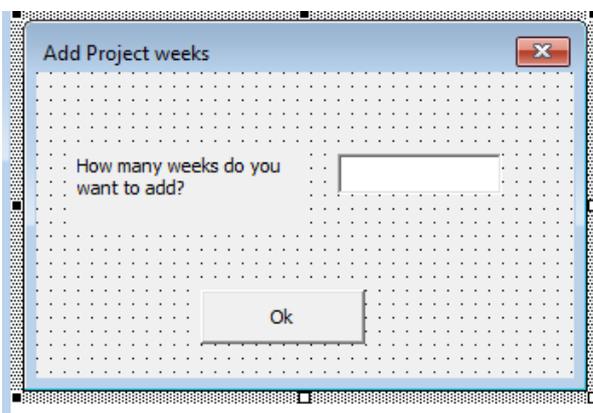


Figure 7 - Form for adding weeks to project

The code for the userform in Figure 8 below.

```

Private Sub Ok2_Click()
|
Dim lastColumn As Long

    For y = 1 To TextBox2.Value
        lastColumn = ActiveSheet.Range("U39").End(xlToRight).Column

            Columns(lastColumn + 1).Insert Shift:=xlToRight
            Columns(lastColumn).Copy Destination:=Columns(lastColumn + 1)
'Copies the last column and inserts it to extend the table

    Next y

Unload WeeksAdd

End Sub
-----
Private Sub TextBox2_Change()

If TextBox2.Value <> "" And Not IsNumeric(TextBox2) Then

    MsgBox "Please enter Numbers only"
End If

    If TextBox2.Value > 15 Or TextBox2.Value < 1 Then

        MsgBox "Number Outside limits ""1-15""
End If
'Keeps the user input between 1-15
End Sub

```

Figure 8 - Code for adding weeks

The code finds and copies the last column in the table and then pastes it to the right of the last column in the table.

This function proved to be a bit challenging. If data were inserted in the last column of the table before new weeks were added. This data would be copied as well resulting in a misleading project overview. The best way to avoid this would be to change the code in a way that it clears data after being pasted, but this would require some complicated coding due to the fact that dates in the new columns should not be deleted. Another way to get around this problem is to inform users that weeks should be added before data is inserted in the last column. For example, a pop-up that displays a warning text could show up if users tried to insert data in the last column.

Resources which the user also had to be able to insert were created as a button for each added installation. Each installation had its own “Add resource button” and when pressing the button one resource would be added to the specific project. Bookings and statuses are later inserted for each resource. Resources can be manpower or tools. Projects with two installations with two resources in the first installations and three resources in the second installation in Figure 9.

		Project weeks			
					1
		Installation	Specification	Comments, Estimated duration of work (weeks)	1
<input type="button" value="Add Team"/>	<input type="button" value="Add booking"/>	Team			0
		Bookings			
		Status			
		Team			0
	<input type="button" value="Add booking"/>	Bookings			
		Status			
		Installation	Specification	Comments, Estimated duration of work (weeks)	1
<input type="button" value="Add Team"/>	<input type="button" value="Add booking"/>	Team			0
		Bookings			
		Status			
		Team			0
	<input type="button" value="Add booking"/>	Bookings			
		Status			
		Team			0
	<input type="button" value="Add booking"/>	Bookings			
		Status			

Figure 9 - Buttons for adding resources and bookings

Code for adding resources under specific project in Figure 10.

```

Sub Button103_Click()

    Rows("42:44").Copy
    Set R = ActiveSheet.Buttons(Application.Caller).TopLeftCell
    R.Offset(0, -5).End(xlDown).Offset(1, 0).Select
    ActiveCell.Insert
    ActiveCell.Offset(1, 9).Resize(2, 200).ClearContents

End Sub

```

Figure 10 - Code for adding resources

In order to get the resource code to work I had to find a way so the system understands where it should insert projects and where it should insert resources. Projects can always be inserted at the bottom of the table, but resources had to be inserted at the bottom of the specific project. If the same function was used as for adding projects, it would lead to resources

always being added at the bottom of the table not depending to which installation it should actually go. To get around this problem values in columns A and B was used. In Figure 11 below one can see how the system knows where to insert the different types of functions. In Column A the code for resource adding searches for the next empty space and inserts it there. The code for installations searches in the same way for the next empty cell in column B which always will be at the bottom of the table and inserts it there. Since the code adds the complete rows these numbers will also be copied and therefore the system automatically updates where the next installation or resource should go.

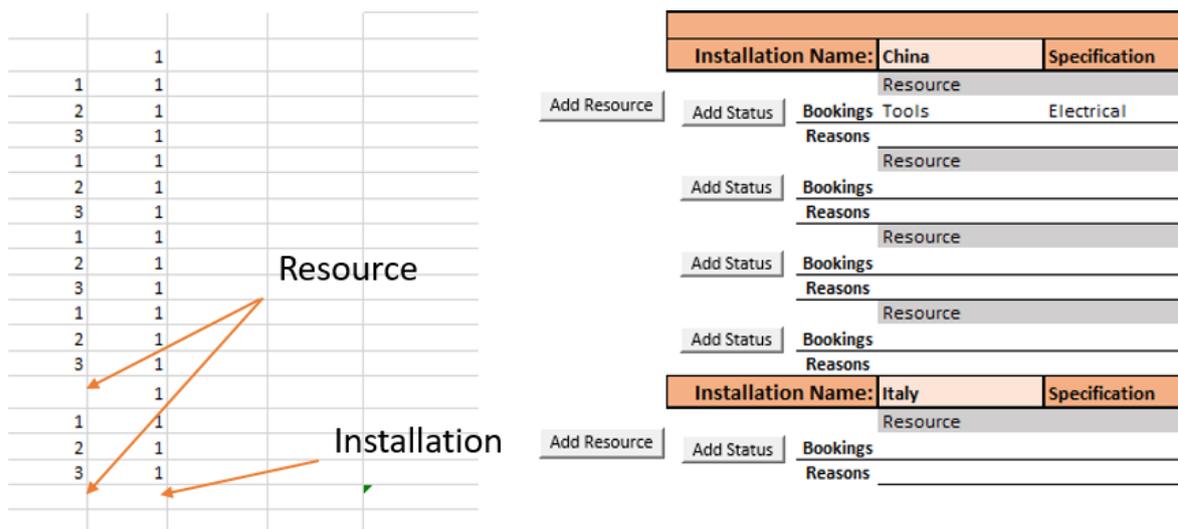


Figure 11 - Description of how the code for resources work

4.3.2 Home Page and Manager Page

As described in Chapters 4.1.4 and 4.1.5 a “home page” and a “manager page” have to be created to make a structural and understandable system. The three types of pages (home page, manager page and project page) had to be easy to navigate between and have connections to one another. Because of that, the system always will start up showing the “home page”, this page has to be easy to understand and the only real way to get out of this page is to click on an existing manager link or to create a new one. At this stage of the development, these pages only had a link connecting them to each other. From the “home page” a link was created to the “manager page” and from there a link was created to access the “project page”. Also, a “back” button was inserted so this also worked the other way around.

5 Second Phase of Development

After the first phase of development all the functions for customization of the project was done and also a basic structure of how the system would be built up. From here the bookings and statuses have to be added and also sent to a data sheet where all statuses would be collected. Another important function was to be able to add managers and projects from the home respective manager page.

5.1 Data Sheet

A data sheet was created in order to collect all data from all projects and later display them on the “home page”. This page worked as a type of database that an Excel pivot table later could work with. All the important information is saved in the data sheet. All cells that are inserted in a project gets an own row in the sheet. As shown in Figure 12 below, a delay that depended on the customer led to a five-week delay. The following week the same resource was done in three weeks.

0	Manager	Project	Resource	Resource Typ	Comments	Installation	Delay/Status	Week	Type of bookin
1	Alexander R	Italy	Tools	Mechanical	Tools	Crankshaft	H Customer Delay	4.3.2019	Hardbooking
2	Alexander R	Italy	Tools	Mechanical	Tools	Crankshaft	H Customer Delay	11.3.2019	Hardbooking
3	Alexander R	Italy	Tools	Mechanical	Tools	Crankshaft	H Customer Delay	18.3.2019	Hardbooking
4	Alexander R	Italy	Tools	Mechanical	Tools	Crankshaft	H Customer Delay	25.3.2019	Hardbooking
5	Alexander R	Italy	Tools	Mechanical	Tools	Crankshaft	H Customer Delay	1.4.2019	Hardbooking
6	Alexander R	Italy	Tools	Mechanical	Tools	Crankshaft	H Done	8.4.2019	Hardbooking
7	Alexander R	Italy	Tools	Mechanical	Tools	Crankshaft	H Done	15.4.2019	Hardbooking
8	Alexander R	Italy	Tools	Mechanical	Tools	Crankshaft	H Done	22.4.2019	Hardbooking

Figure 12 -Data in the data sheet

5.2 Adding of Bookings and Statuses

The main function of the whole thesis is to be able to add bookings and statuses to these bookings. This function had to as easy and fast as possible where misunderstandings should be considered next to none. A button for each resource was created that opened a userform as is shown in Figure 13, where users could choose to either add a booking or add a status for an existing booking. If no existing booking was inserted for that resource the function would automatically go straight to “add booking”. This because a status should not be added before a booking.

Figure 13 - Form for adding bookings or delays/statuses

5.2.1 Adding Bookings

When adding a booking the user must fill in a userform which can be seen in Figure 14. The userform consists of six fields that all must consist of a value or text. If not everything is filled in when the user clicks “add” there will be a message box saying that every field must have something inserted. After everything is properly inserted in the userform and the user clicks “add” a booking will be inserted in the project table. The booking will be displayed as a blue cell with either hard booking or soft booking depending of the choice of booking.

Figure 14 - Form for adding bookings

In the first field from the left there should be a type of order chosen, here the alternatives are manpower or tools. Basically, this option is to select if the order involves manpower or tools. In the specification column there are three alternatives mechanical, electrical or PLC. Here the users define the type of order by choosing one of the alternatives. The comments box is a free text box where the user can describe the booking further. The type of booking column is either a hard- or soft-booking which was explained in Chapter 4.1.7. The first Monday of booking column allows the user to select the first week of the booking. The column consists of a dropdown with each Monday as a date. In the last column the user selects the length of

the booking with a number of weeks. If there is a delay on the booking the user will have to insert a new one. When the new booking is inserted the system saves the data from the three first columns of the userform (Type of order, Specification and Comments) to facilitate the process. The code for adding a booking is presented in Figure 15 below.

```
Private Sub ComboBox4_Change()

Set r = ActiveSheet.Buttons(Application.Caller).TopLeftCell
    r.Offset(0, 4 + ((ComboBox4.Value - Range("E4")) / 7) + 1).Select
    'Finds the correct cell for the first week of the booking
End Sub
```

```
Private Sub CommandButton1_Click()
    If TextBox2.Value = "" Then
        MsgBox "Length of booking can not be empty"
    End If

    If ComboBox1.Text = "" Then
        MsgBox ("Type of order can not be empty")
    Else
        Set r = ActiveSheet.Buttons(Application.Caller).TopLeftCell
        r.Offset(0, 2) = ComboBox1.Text
        'Inserts type of booking
    End If

    If ComboBox2.Text = "" Then
        MsgBox ("Specification can not be empty")
    Else
        Set r = ActiveSheet.Buttons(Application.Caller).TopLeftCell
        r.Offset(0, 3) = ComboBox2.Text
    End If
    'Inserts specification of booking
    If TextBox1.Text = "" Then
        MsgBox ("Comments can not be empty")
    Else
        Set r = ActiveSheet.Buttons(Application.Caller).TopLeftCell
        r.Offset(0, 4) = TextBox1.Text
        End If
    'Inserts comments for booking

    Set A = r.Offset(0, -7).End(xlUp)
    If A.Offset(-1, 9).Value = "" Then
        MsgBox "Give Installation Name"
        GoTo Last
        'Checks if there is an installation name
    End If
    Dim addr As String, rng, cell As Range
    Application.DisplayAlerts = True
    Set rng = ActiveCell.Resize(1, TextBox2.Value)
    'Selects the duration of the vooking as a range
    For Each cell In rng
        cell.Value = ComboBox3.Value
        'Inserts booking
    Next cell
    If ComboBox3.Text = "" Then
        MsgBox ("Cannot be empty")
    End If
    Last:
    Unload BookingAdd

End Sub
```

Figure 15 - Code for adding booking

5.2.2 Add Statuses

When the booking is inserted the either add a new booking or a status. To add a status the user clicks on the “Delay/Status” button instead of booking. This action will display an userform which is shown in Figure 16 below.

Figure 16 - Form for adding delays/statuses

Here the users have to insert the basic definitions of the delay. In the first column the user must choose if this status is for a hard- or soft-booking. Secondly, the user must choose the type of delay which are listed in Chapter 4.1.7. The two last columns are the same as in the bookings userform which means the first Monday of the status and length of the status. The best option when having a delay on the first booking is to insert a status first and then move the booking by marking it and dragging it to the new decided week. This way there will be a type of trace visible for the user to better understand how the booking have changed due to the delay. The code for adding a “delay/status” in Figure 17 below.

```

Set r = ActiveSheet.Buttons(Application.Caller).TopLeftCell
    r.Offset(1, 4 + ((ComboBox6.Value - Range("E4").Value) / 7) + 1).Select
'Chooses the correct cell depending of the start date
End Sub

Private Sub CommandButton1_Click()
|
    Dim SelectRange As Range

Set SelectRange = ActiveCell.Resize(1, TextBox1.Value)
SelectRange = ComboBox2.Value

For Each cell In SelectRange

Sheets("DATA").Range("A1").End(xlDown).Offset(1, 0).Value = 1 + Sheets("DATA").Range("A1").End(xlDown).Value
cell.Offset(-2, 0).Value = Sheets("DATA").Range("A1").End(xlDown).Value
Sheets("DATA").Range("J1").End(xlDown).Offset(1, 0).Value = ComboBox1.Value
Sheets("DATA").Range("I1").End(xlDown).Offset(1, 0).Value = Columns(cell.Column).Rows(38).Value
Sheets("DATA").Range("H1").End(xlDown).Offset(1, 0).Value = ComboBox2.Value
Sheets("DATA").Range("B1").End(xlDown).Offset(1, 0).Value = Range("L20").Value
Sheets("DATA").Range("C1").End(xlDown).Offset(1, 0).Value = ActiveWindow.ActiveSheet.Name
Sheets("DATA").Range("D1").End(xlDown).Offset(1, 0).Value = ActiveSheet.Shapes(Application.Caller).TopLeftCell.Offset(0, 2).Value
Sheets("DATA").Range("E1").End(xlDown).Offset(1, 0).Value = ActiveSheet.Shapes(Application.Caller).TopLeftCell.Offset(0, 3).Value
Sheets("DATA").Range("F1").End(xlDown).Offset(1, 0).Value = ActiveSheet.Shapes(Application.Caller).TopLeftCell.Offset(0, 4).Value
Set abcd = ActiveSheet.Shapes(Application.Caller).TopLeftCell.Offset(0, -7)
Sheets("DATA").Range("G1").End(xlDown).Offset(1, 0).Value = abcd.End(xlUp).Offset(-1, 9).Value

Next cell

ThisWorkbook.RefreshAll
Unload DelayAdd
'Adds the status to the project page and to the data sheet
End Sub

```

Figure 17 - Code for inserting delay/status

5.3 Add Managers and Projects

A function to add managers and projects was essential in order for the system to work properly. Starting with the “home page”. From here the user should be able to add a manager which also creates a “manager page” and a link to it. This would result in a list of all the managers on the homepage. When clicking on a manager you will go to that specific managers own page, in where his projects are listed. This page is called the “manager page” and should consist of a list of projects, a button to add more projects, and a go back button to return to the “home page”. The foundations for this to work is to have an empty manager page and an empty project page in the system, these are called “OriginalPM” and “OriginalProject”. When adding a page one of the basic pages is copied and named after the name that is inserted in the userform. This way all the new projects and managers will get an unused basic page.

In the homepage of the software, a button where the user can add managers is created and when clicked on displays userform in which a manager can be named and inserted. This userform can be seen in Figure 18 below.

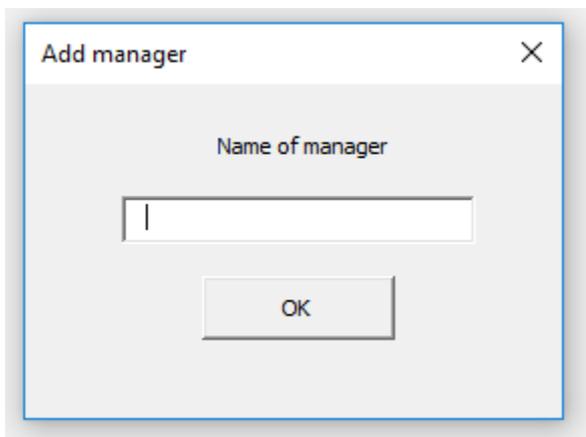
The image shows a standard Windows-style dialog box titled "Add manager". It features a close button (an 'X' icon) in the top right corner. The main content area contains a label "Name of manager" positioned above a single-line text input field. Below the input field is a rectangular button labeled "OK". The dialog box has a light gray background and a thin blue border.

Figure 18 - Form for adding managers

When clicking “OK” the system creates a new manager page with the name the user inserted. Also, a link on the homepage is created which leads to that managers “project page”. If there are previous inserted managers, the code will insert the link below the last link. There are also requirements that the same manager should not be able to be inserted twice, Excel searches through all sheets and checks this before it allows the user to add the manager.

Code for this function in Figure 19.

```

Private Sub CommandButton1_Click()

Application.Calculation = xlCalculationManual
Application.ScreenUpdating = False
Application.DisplayStatusBar = False
'Shuts down automatic updates

Dim s As String
A = TextBox1.Text
Dim ah As Worksheet, nah As Worksheet
Dim nrng As Range
Dim cont As Worksheet
Dim oRng As Range
'Defines variables
If A = "" Then
MsgBox ("Cant be empty")
Exit Sub
End If

ActiveSheet.Range("C7").End(xlDown).Offset(1, 0).Select
'Selects cell where link should be placed
TextBox1.Value = ActiveCell.Value
ActiveCell.Font.Size = 14
'names cell
Set oRng = ActiveCell
Set ah = ActiveSheet
For rep = 1 To (Worksheets.Count)
If LCase(Sheets(rep).Name) = LCase(A) Then
MsgBox "This manager already exists"
Exit Sub
End If
'Searches if there already is a sheet whit this name
Next

Sheets("OriginalPM").Visible = True
Sheets("OriginalPM").Copy After:=Sheets("HomePage")
ActiveWindow.ActiveSheet.Name = A
Range("L10").Value = ActiveWindow.ActiveSheet.Name
'Places manager name in the right cell

ah.Activate
ah.Hyperlinks.Add oRng, "", "" & A & "!L3", _
"Go to " & A, A
'Activates the link
ActiveCell.Font.Size = 14
Set oRng = Nothing
Unload AddManager
Application.Calculation = xlCalculationAutomatic
Application.ScreenUpdating = True
Application.DisplayStatusBar = True
'Re-enables the automatic updates
End Sub

```

Figure 19 - Code for adding managers

In the manager page there will be also be a button that adds a new project page that is named after the name the user chooses. The userform can be seen below in Figure 20.

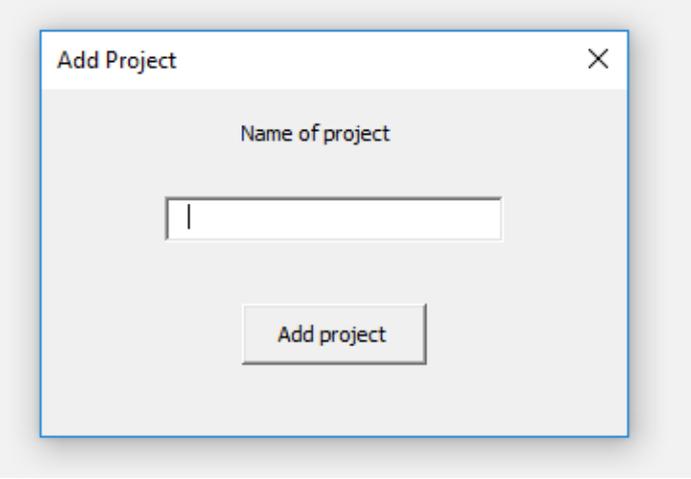


Figure 20 - Form for adding projects

This code works the same way as the adding of manager. The new page and a link to it will be created. Code for this function in Figure 21.

```
Private Sub CommandButton1_Click()

Application.Calculation = xlCalculationManual
Application.ScreenUpdating = False
Application.DisplayStatusBar = False
'Shuts down automatic updates

Dim s As String
s = TextBox1.Text
Dim sh As Worksheet, nsh As Worksheet
Dim nrng As Range
Dim cont As Worksheet
Dim oRng As Range
'Sets variables

If s = "" Then
MsgBox ("Cant be empty")
Exit Sub
End If

ActiveSheet.Range("C10").End(xlDown).Offset(1, 0).Select
TextBox1.Value = ActiveCell.Value
Set oRng = ActiveCell
Set sh = ActiveSheet
'Names project and places it below previous projects

For rep = 1 To (Worksheets.Count)
If LCase(Sheets(rep).Name) = LCase(s) Then
```

```

MsgBox "This sheet already exists"
Exit Sub
End If
'Searches the workbook if there is another sheet with this name
Next

Sheets("OriginalProject").Visible = True
Sheets("OriginalProject").Copy After:=Sheets(Sheets.Count)
ActiveWindow.ActiveSheet.Name = s
sh.Activate

Sheets(Sheets.Count).Range("L20").Value = Range("L10").Value

Sheets(Sheets.Count).Range("J37").Value = s
sh.Hyperlinks.Add oRng, "", "" & s & "!L20", _
"Go to " & s, s
'Adds the link

ActiveCell.Font.Size = 14
Set oRng = Nothing

Unload ProjectAdd

Application.Calculation = xlCalculationAutomatic
Application.ScreenUpdating = True
Application.DisplayStatusBar = True
'Re-enables the automatic updates
End Sub

```

Figure 21 - Code for adding projects

5.4 Navigating Between Sheets

There has already been mentioned in Chapter 5.3 how the user can go from the “home page” to the “manager page” and on to the “project page”. This all worked fine, but there had to be a way for the user to be able to go back from the “project page” to the “manager page” and then finally back to the “home page”. To make a solution between the “manager page” and the “home page” was relatively simple as there could be “Go back” link created on the “OriginalPM” page that always lead to the home page. But between the “project page” and the “manager page” things got a little more complicated because of that the system had to understand to which manager page it should go back to. This was solved by using a code in VBA that saved the previous page that was open (The manager page if the user is currently in a project page). A “Go back” button was created on the “project page” that lead the user back to the page that was previously open. The code for these functions in Figure 22 and 23.

```
Private Sub Workbook_SheetDeactivate(ByVal sh As Object)
    MyPrevSheet = sh.Name
End Sub
```

Figure 22 - Code for sheet navigation

This code was set in under Microsoft Excel objects in “This worksheet” and remembers the previous worksheet.

The button on the “project page” that allowed users to go back to their “manager page” had this code. If there for some reason had not been a change between before the button was clicked there would be a message saying that there has not been a change of sheets.

```
Sub Button46_Click()

    If Len(MyPrevSheet) > 0 Then
        Sheets(MyPrevSheet).Activate
    Else
        MsgBox "You have not switched sheets yet since opening the file!"
    End If

End Sub
```

Figure 23 - Code for navigating away from project page

To minimize errors and mistakes the code in figure was created to always open the homepage first when someone opens the Excel file. The line of code for this to work can be seen in Figure 24 below.

```
Private Sub Workbook_Open()

    Sheets("HomePage").Select

End Sub
```

Figure 24 - Code for displaying the homepage when file is opened

5.5 Testing

Since testing is a vital part of any software development as described in Chapter 3.1.5. At this point the system was presented to the future users and a discussion of what could be improved was held. There was a request to make the first booking of a resource clearly visible compared to later booking in the same resource. This was made to easily being able

to see where and how long the original booking was. For example, if a booking of three weeks is inserted and then gets delayed by one week there will be one more week of booking inserted because the booking was “moved” one week forward. This means there is now four weeks of instead of three. Therefore, it would be important to have the original booking to look different to later bookings. Sometimes a delay on one resource can delay all the other resources as well, there was a request about how to easily add the same delay to multiple resources since it might take some time to do it for one resource at a time. By monitoring the progress and getting feedback from the future users was a valuable input which can be read more about in Chapter 3.5, monitoring progress.

6 Final Phase of Development

In the final phase of the development charts will be created for the homepage. There will be a button created that allow users to directly go to a specific date. This function will be helpful for very long-lasting projects in order to more easily navigate within the sheet. To slim down the room for error there will also be function for users to delete wrongly input statuses. This function will be used as a solution if a mistake happens.

6.1 Charts for Homepage

The best way to create charts where the user can choose what to display is by using the pivot table function. Pivot tables is a great way of handling large amounts of data and display them in a visually appealing way. The theory about pivot tables can be found in Chapter 3.6. Because of the importance of collecting booking data in this thesis there must be a user friendly and customizable chart that can display the chosen data. To achieve this there had to be a pivot table, slicers for the chart and also a timeline for the chart. The data was taken from the data sheet and displayed as a chart on the home page as seen in Figure 25. By using a combination of the pivot table, slicers and a timeline the overview in figure was created. Pivot table settings can also be seen in Figure 26.



Figure 25 - Charts displaying statuses on Home page

Figure 25 above shows the completed home page with slicers to the left, these slicers allow the users to display for example only delays for hard bookings, only delays for one manager or only delays for one project. The timeline at the bottom also allows

users to select a time window, for example between which months they want the chart to show the delays for. The chart is displayed in the middle with the numbers for the chart at the right. This way the chart is fully dynamic as it is easy to see the basic overview as well as the detailed numbers.

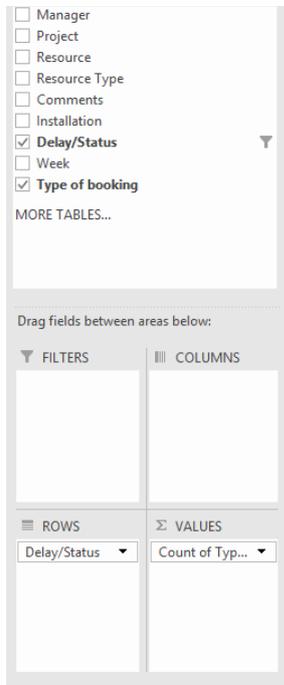


Figure 26 - Pivot table for charts

Since the table had to be updated every time a new status had been inserted there had to be a code inserted to the system. The code in Figure 27 below always kept the data in the pivot table updated.

```
Private Sub Worksheet_SelectionChange(ByVal Target As range)
ThisWorkbook.RefreshAll
End Sub
```

Figure 27 - Code for updating data in pivot table

6.2 Go to Function

This function does not have any important role in this system but was instead made as a helpful tool for very long multi installation projects. The function allows users to “Go to” a certain week in the project without having to scroll all the way. This function bases on the calendar weeks. The user form shows a drop box where all weeks are listed, and the user

simply chooses to which they want to go and then click “ok”. Userform and code for this function in Figure 28 and 29 below.

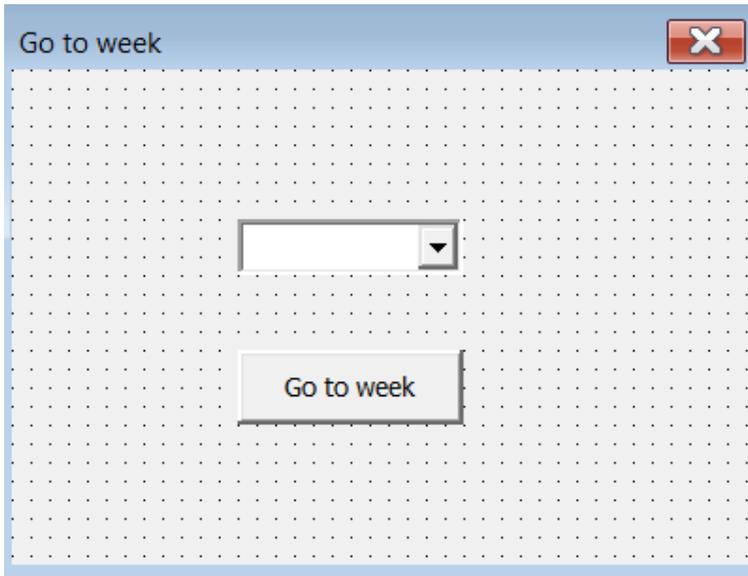


Figure 28 - Userform for Goto function

```
Private Sub CommandButton1_Click()
    Dim dateVal As Long
    Dim cell As Range

    With ActiveSheet

        dateVal = ComboBox1.Value
        For Each cell In .Range("M38:MM38").Cells
            If dateVal = cell.Value Then
                Application.GoTo cell, True
                Exit For
            End If
        Next
    End With

    Dim ColVis As Single

    ColVis = ActiveWindow.VisibleRange.Columns.Count
    Application.GoTo Reference:=ActiveCell, Scroll:=True
    ActiveWindow.SmallScroll Toleft:=ColVis \ 2
    ActiveWindow.SmallScroll Up:=ColVis \ 2

    'Searches for the date that is chosen and siplays it in the middle of the sheet

    Unload GoToWeek
End Sub
```

Figure 29 - Code for Go to function

6.3 Delete Statuses

To be able to delete statuses is also a very important function to avoid mistakes and incorrect data. At some point there will probably be a mistake when inserting a status, for example the user choses the wrong week or wrong type of delay and realizes this after they have inserted it. The data is then already in the data sheet where the user does not have access. To make

this as easy as possible a code was made but there were some problems on how to make it work in the right way. To solve this problem a new button with a user form that had a “refedit” function in it. The “refedit” function allowed the users to select the cell that should be deleted and then when the “ok” button is clicked the status is deleted from the project page and from the data sheet. In order to make this function there had to be a code above each status so that the system finds the right row in the data sheet to delete. When deleting from the data sheet the system then deletes the status from the “project page”. Due to complicated coding, this function had to be restricted to one cell at a time. The goal was to allow users to delete multiple cell at a time, but problems lead to the solution being to allow users to delete only one status at a time. An error handling code was also inserted in case users tried to choose multiple cells to delete. This line stops the code and display a message that only on cell can be chosen at a time. Userform and code for this function in Figure 30 and 31 below.

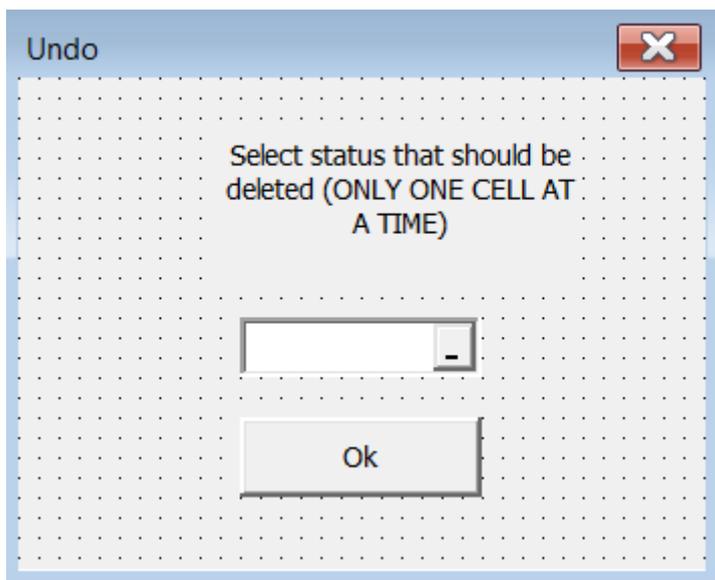


Figure 30 - Userform for deleting statuses

To inform users that only one cell at a time can be selected the label in Figure 30 was inserted to minimize errors. This is an area where future improvements could be made to increase user friendliness.

```

Sub undook_Click()
Application.ScreenUpdating = False
Application.DisplayStatusBar = False
'Shuts down automatic updates
On Error GoTo Last
'If there is an error this code cancels the code

Dim FindString As String
Dim rng As Range
'Sets variables

FindString = Range(RefEdit5).Offset(-2, 0).Value
If Trim(FindString) <> "" Then
Range(RefEdit5).ClearContents
Sheets("DATA").Select
With Sheets("DATA").Range("A:A")
Cells.Find(What:=FindString, After:=ActiveCell, LookIn:=xlValues, LookAt:= _
xlWhole, Searchorder:=xlByColumns, Searchdirection:=xlNext, MatchCase:= _
False, Searchformat:=False).EntireRow.Delete
'Selects the code above the status that should be deleted and finds it in the data sheet to the delete the entire row
Sheets(MyPrevSheet).Activate
Range(RefEdit5).Offset(-2, 0).ClearContents
End With
End If
Undo.Hide
'Clears the code from project page
Exit Sub

Last:
Undo.Hide
MsgBox "Something went wrong"
'Error message
Application.ScreenUpdating = True
Application.DisplayStatusBar = True
're-enables automatic updates

End Sub

```

Figure 31 - Code for deleting statuses

6.4 Further Improvements

To create as little room for error as possible the cells that users should not be able to change will be locked and can only be changed through the functions in the buttons. The only cells that should really be open for users is the rows where bookings are. This due to the fact that the users can move bookings and add length to them or to shorten them without using buttons. The tool has been checked for bugs, but as these are hard to notice and can occur while doing small changes to other functions, not all of them have been handled. Especially in the error handling department there are room for improvement. If the users use the tool correctly as presented to them there should not be any bugs and faults, but since mistakes can happen it would be good to improve the error handling for the tool. Since I will continue to work at this department during the summer of 2019, I will have the chance to improve the tool further to make it more complete and improve especially the error handling.

7 Result

The final Excel file that was produced and fulfilled the purpose and requirements of this thesis. The development process which was based on the theoretical foundations resulted in a functional tool that can be used for projects in the Customer Assistance department. The tool can be used for the tasks it was developed around. Projects can be customized, bookings can be added and statuses for these bookings can be inserted. On the “home page” a functional chart displays the data the user chooses to display. For future improvements there could be improvements in bug detection and handling. Overall the personnel in Customer Assistance was happy with the way the tool functioned and the possibility to customize projects easily. They also found it useful for future projects. Because of that one of the main requirements was user friendliness there should not be a lot of problems with implementation. Basically, the end product consisted of three basic pages that could be copied to form a system of projects and managers. In the final version of the tool the home page was displayed as in Figure 32.

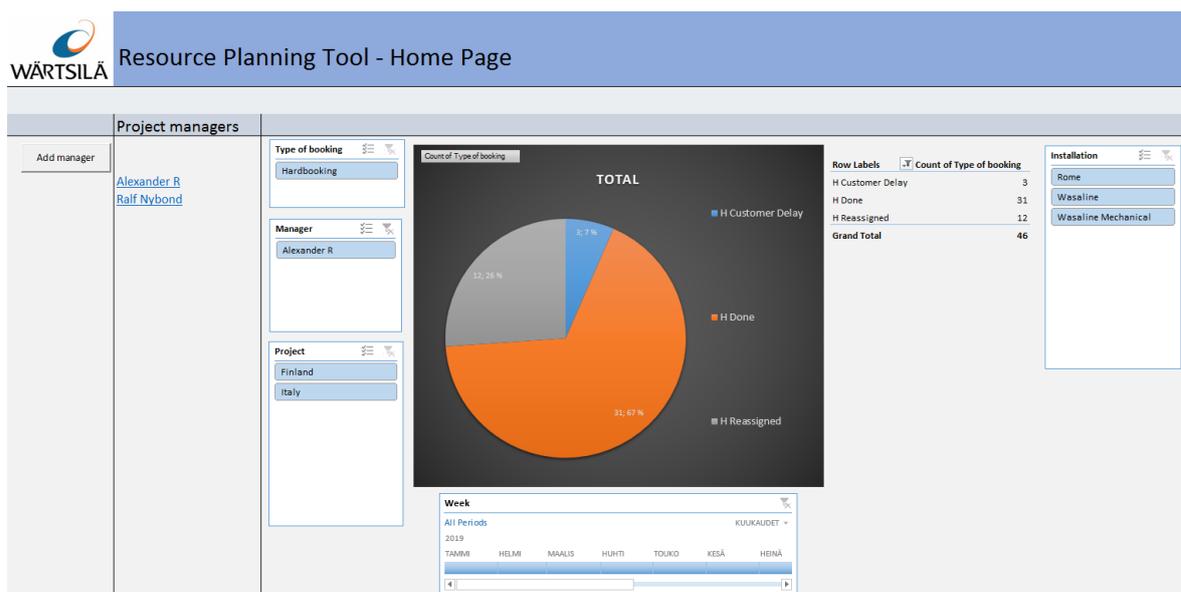


Figure 32 – Home Page of the tool

The “home page” had all the necessary functions and worked like it should. Managers could be added and charts for displaying statuses worked like they should. The Wärtsilä logo was added for a better visualizing display.

The final version of the “manager page” was displayed as in Figure 33.

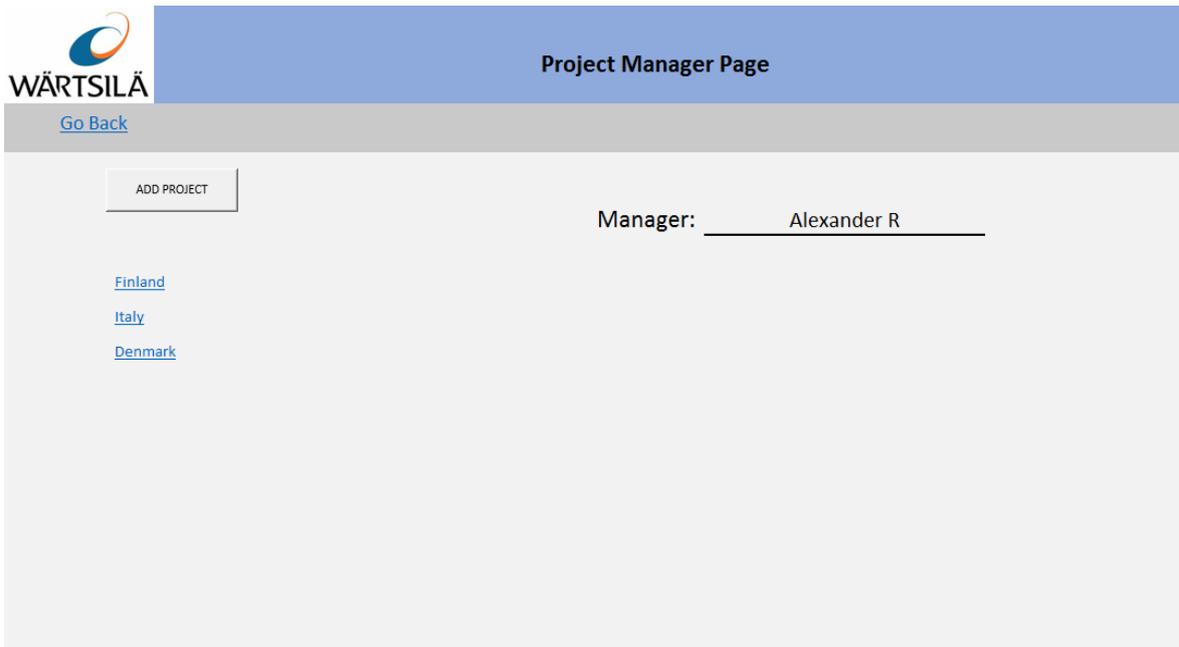


Figure 33 - Manager Page of the tool

This page was quite simple, but still useful to increase user friendliness. Here the specific managers projects are listed, and projects can easily be added.

The most crucial page; the “project page” was displayed as in Figure 34.

Project Name: Finland				Project Weeks					
Installation Name: Wasaline				1	2	3	4	5	6
Resource				10.4.2019	17.4.2019	24.4.2019	1.5.2019	8.5.2019	15.5.2019
Add Resource	Add Status	Bookings	Tools Electrical Tools for automation	H Customer Delay	Hardbooking H Done				
		Status							
		Resource							
Add Status	Bookings	Manpower	Electrical Kalle automation	H Customer Delay	Hardbooking H Done				
		Status							
		Resource							
Add Status	Bookings	Manpower	Electrical Bo automation	H Customer Delay	Hardbooking H Done				
		Status							
Installation Name: Wasaline Mechanical				10.4.2019	17.4.2019	24.4.2019	1.5.2019	8.5.2019	15.5.2019
Add Resource	Add Status	Bookings	Tools Mechanical tools mechanical						
		Status							
		Resource				H Reassigned	H Reassigned	H Reassigned	H Reassigned
Add Status	Bookings	Manpower	Mechanical mechanical manpower x2						
		Status				H Reassigned	H Reassigned	H Reassigned	H Reassigned

Figure 34 - Project Page with example bookings and statuses

Here all the functions that has been introduced in chapters 4, 5, 6 are working and the users can easily add bookings and customize their projects to their specific needs. The tool got some good feedback from the future users, but since there has not been any proper testing with many different projects, there might still be some small adjustments to the tool.

8 Discussion

Since the thesis fulfilled the requirements and the result was a tool that can work well with the projects that Customer Assistance handles, this thesis can be described as successful even if that there are some minor bugs that could be fixed quite easily. Personally, I think the tool functions better than expected and that it really could help the department in the future. This due to that I feel that all the necessary functions are in the tool and that they work together well. The most important factor of the tool was that the users was happy with the tool and that they found it to be useful for future projects.

8.1 Summary

Since I did not have that much experience of Visual Basic Application from before the development process was quite hard in the beginning when I had to learn a bit more complicated code and therefore the progress was slow. I had a lot of help from the department members and my supervisor to set the requirements for the tool and how it could be set up. The programming for this tool was completely self-made with help only from researching solutions that could help my case and then customizing them to make them work in the tool. By researching these solutions and making small changes to them so that they would fit the system I quickly learned how the different codes worked and could start doing more complicated lines. Basically, I just learned by my mistakes and tried to use all the help I could find on the internet. Since Excel is such a popular software is was quite easy to find people with similar problems as I had and therefore it was also easy to find solutions to these problems. The biggest challenge for me was definitely how to make the projects customizable by being able to add installations resources and weeks. I had to create these functions in the early phases of the development when I did not have that much experience of coding in VBA. To make these three to work together without any bugs was very time consuming, but made me understand VBA better. As mentioned in Chapter 6.4 there could still be some improvements to the tool. Especially, in order to minimize the possible user mistakes even more.

By using this tool correctly, it could help to create a better understanding of why bookings get delayed and what could be done to increase successful bookings. This could lead to less delays in projects, which will reduce project costs for Customer Assistance.

This thesis was an interesting test for me since I had to learn too much before actually being able to make decent progress. Before I got the chance to make this Thesis, I did not imagine

doing anything like this, but now after I am really happy that I did and that I now have a much better image of what can be done in Excel. Because of the fact that Excel is a very common tool to use for different tasks in industries this thesis has given me a great advantage for future tool developments in Excel.

8.2 Conclusion

To conclude this Thesis, I would describe it as successful, but with room for minor improvements especially in error handling. From now on when the tool will be properly used and tested, we will see if other problems and issues occur and how they can be fixed. From here on there will probably be a need for some adjustments. Since I will continue to work at Customer Assistance, I will be able to maintain the tool further, if needed. Continued research from this point is to test the tool in real projects and to study the delays and discuss possible solutions to reduce delayed bookings. I want to thank both my supervisor Ralf Nybond from Wärtsilä and Mikael Ehrs from Novia for the help and feedback they have given me.

9 References

- About Wärtsilä.* (2019). Retrieved from Wärtsilä: <https://www.wartsila.com/about>
- Alexander, M., & Kusleika, D. (2016). *Excel 2016 Power Programming with VBA*. Indianapolis: John Wiley & Sons, Inc.
- Benton, C. (2017). *Microsoft Excel Pivot Tables & Introduction to Dashboards The Step-By-Step Guide*.
- Hallin, A., & Gustavsson, T. K. (2012). *Project management*. Stockholm: Ola Håkansson.
- Holman, J. H., & Blake, K. (2018). *What Every Engineer Should Know About Excel Second Edition*. Boca Raton: Taylor & Francis Group.
- Introduction to Microsoft Excel.* (n.d.). Retrieved from Guru99: <https://www.guru99.com/introduction-to-microsoft-excel.html>
- Marion, J. W. (2018). *Project Management A Common Sense Guide to the PMBOK, Part One-Framework and Schedule*. New York: Momentum Press, LLC.
- Mathaisel, D. F., Manary, J. M., & Criscimanga, N. H. (2013). *Engineering for Sustainability*. Boca Raton: Taylor & Francis Group.
- Stephens, R. (2015). *Beginning Software Engineering*. Indianapolis: John Wiley & Sons, Inc.
- Stiller, E., & Leblanc, C. (2002). *Project-based software engineering*. Addison-Wesley.