



Planering och implementering av rapporter till programvaran Granlund Manager

Wilhelm Kinos

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informationsteknik
Identifikationsnummer:	
Författare:	Wilhelm Kinos
Arbetets namn:	Planering och implementering av rapporter till programvaran Granlund Manager
Handledare (Arcada):	Jonny Karlsson
Uppdragsgivare:	Granlund Oy
<p>Arbetet beskriver planeringen och utvecklingen av ett nytt rapporteringssystem till programvaran Granlund Manager. Planeringen består av en kravanalys där man samlar in företagets och kunders krav. Detta görs för att få en bättre bild på arbetets krav och hur man vill att lösningen skall se ut. Från kravanalysens krav var det också lättare att fortsätta med utvecklingen av det nya rapporteringssystemet. Man valde att börja med det nya rapporteringssystemet med ett bibliotek som företaget redan har licens till, som heter Syncfusion. Med Syncfusion Dashboard Designer designade man en rapport enligt kraven från kravanalysen. Rapporten behövde filtreringsmöjligheter samt en graf. När rapporten var färdig designad implementerade man den till programvaran Granlund Manager. Det gjorde man med Syncfusions skrip, som läste rapporten så den blev synlig i programvaran. För att få data från databasen kopplade till rapporten använde man sig av SQL. Med SQL gjorde man en databasförfrågning som kopplades till rapporten och från den kunde man koppla rätta data till grafen och filtreringsmöjligheterna, som var rullgardinsmenyer. När man hade ett fungerande rapporteringssystem med data från databasen kunde man göra en utvärdering av lösningen. Man samlade in data genom att skicka en bilaga, med hjälp av Google Forms. Bilagan skickades till personer som varit med och hjälpt med arbetet. Lösningen blev inte helt som man ville ha den, för man kunde inte ändra på databasförfrågan som var kopplad till rapporten. Detta gjorde att rapporten inte blev så dynamisk som önskat och man kunde inte fortsätta att utveckla denna lösning. Företaget var ändå tillfreds med arbetet, för man hade Syncfusion som ett alternativ för ett nytt rapporteringssystem. Företaget fick nu också en bättre bild av vad som behövs till rapporteringssystemet. I fortsättningen kommer företaget att satsa på större rapporteringssystem, som Microsoft Power BI.</p>	
Nyckelord:	Syncfusion, rapporteringssystem, Granlund Manager, kravanalys
Sidantal:	26
Språk:	Svenska
Datum för godkännande:	01.06.2019

DEGREE THESIS	
Arcada	
Degree Programme:	Information Technology
Identification number:	
Author:	Wilhelm Kinos
Title:	Planning and implementation of reports to the software Granlund Manager.
Supervisor (Arcada):	Jonny Karlsson
Commissioned by:	Granlund Oy
<p>The thesis describes the planning and development of a new report system to the software Granlund Manager. The planning consists a requirement analysis where you gather clients and company requirements. This is done to get a better picture of the thesis requirements and the solution for it. From the requirement analysis requirements was it also easier to continue with the development of the new report system. To develop the new report system a library that the company already had a license to, named Syncfusion was chosen. With Syncfusion Dashboard Designer a report was designed with the requirements from the requirement analysis. The report needed filter options and a graph. When the report was designed, it was implemented to the software Granlund Manager. It was made with Syncfusions scripts, that read the report, so it became visible in the software. To get the data from the database connected to the report a query was made with SQL. The query was added to the report and from the query the correct data got connected to the graph and filter options that were dropdown lists. When a working report system with real data was made, it was possible to do an evaluation from the solution. Data was collected by sending an annex with the help of Google Forms. The annex was sent to people that had helped with the work. The solution didn't become as expected, since it wasn't possible to change the query that was connected to the report. That meant that the report wasn't as dynamic as hoped and it wasn't possible to continue to develop on the solution. The company was still satisfied with the work because Syncfusion was on their list for an option for a new report system and they now got a better picture of what is needed for a report system. In the future the company will invest in bigger solution like Microsoft Power BI.</p>	
Keywords:	Syncfusion, Report System, Granlund Manager, Requirement analysis
Number of pages:	26
Language:	Swedish
Date of acceptance:	01.06.2019

INNEHÅLL

1	Inledning.....	6
1.1	Bakgrund	6
1.2	Syfte och målsättningar	7
1.3	Metoder	7
1.4	Avgränsningar	8
1.5	Struktur	8
2	Kravanalys	9
2.1	Vad är en kravanalys?.....	9
2.2	Kravanalys på det nya Rapporteringssystemet.....	10
3	Utvecklingen av Rapporteringssystemet	11
3.1	Val av bibliotek	11
3.1.1	<i>Syncfusion</i>	12
3.2	Uppbyggnaden av rapporten.....	13
3.3	Implementering av en testrapport i Granlund Manager.....	14
3.4	Tillämpning av data till rapporten	16
4	Utvärdering	18
4.1	Datainsamlingsmetod.....	18
4.2	Utvärderingen av Syncfusion Dashboard.....	19
4.3	Utvärderingen av rapporteringssystemets lämplighet för Granlund Manager.....	19
5	Slutsatser	20
	Källor	22
	Bilagor	23

Figurer

Figur 1. Ett exempel på hur en Syncfusion Dashboard kan se ut	15
Figur 2. En Syncfusion Dashboard implementerad till Granlund Manager	17
Figur 3. En bild som visar hur man kunde dra en kolumn från sin data till ett visst fält i Syncfusion Dashboard Designer	18

1 INLEDNING

1.1 Bakgrund

Granolund är ett stort företag inom fastighets- och byggnadsbranschen och deras största mål är att satsa på fastighetens funktionalitet, samt personernas välmående i miljön. Granlund grundades år 1960 och har flera hundra arbetare i kontor runt om i Finland (Granolund). Företaget har också en programvara Granlund Manager som är ett kontrollsystem för fastigheters underhåll. Jag har jobbat med att hjälpa och utveckla denna programvara och examensarbetet beskriver utvecklingen av en del av programmet. Programvaran behövde ett nytt rapporteringssystem och därför vill man utreda bästa lösningen för detta. Med ett nytt och välfungerande rapporteringssystem är det möjligt att få mera kunder att använda programvaran.

Rapporterna var för tillfället gjorda och lästes med Microsoft ReportViewer som är ett rapporteringssystem för ASP.NET applikationer för att göra och läsa rapporter i programvaran (Got ReportViewer). Rapporterna sparas i egna filer och kan sedan öppnas från applikationen. Denna lösning är ganska gammal och rapporten öppnas i ett nytt fönster, som gör att lösningen är långsam.

Rapporter tilläggs i program för att få en mer analytisk bild av data i program. Med bra rapporter kan kunden t.ex. göra lättare beslut med hjälp av analys av rapporten och det går snabbare att se en översikt av datan från en graf, än från en lång tabell. Ett välfungerande rapporteringssystem för ett program förstärker dess konkurrenskraft och leder till mera kunder.

De flesta rapporter i andra system görs oftast på samma sätt som det gamla rapporteringssystemet som Granlund använde. Man använder sig av ett bibliotek som man kan göra rapporter med och designar dem som man vill ha dem och efter det kan man tillägga data till den. Dessutom kan man också programmera ett eget rapporteringssystem med ett UI Framework som man kan göra grafer med.

1.2 Syfte och målsättningar

Syftet med den praktiska delen av examensarbetet var att göra en ”konceptvalidering (Eng. Proof Of Concept (POC))” på en rapport. POC innebär att man har en idé och försöker förverkliga den och till sist kollar man om man är nöjd med resultatet. Med det nya rapporteringssystemet vill man att rapporterna blir mer moderna, att de laddas snabbare och att de är mer dynamiska. Dynamisk blir den med att rapportens graf eller tabell laddas genast upp på samma sida och om man ändrar på något av filtreringsalternativen, så uppdateras grafen eller tabellen genast på samma sida med nytt data. För att få rapporten mer dynamisk vill man att man skall kunna filtrera i rapporten vad som vill visas. Ett filtreringsalternativ kan ex. vara att man kan välja mellan vilka dagar data är synligt dvs. då man väljer vilket data som skall vara synligt, skall grafen uppdateras med ny data. Detta gör också att det nya systemet skall vara snabbare än det förra, för nu görs allting på samma sida. Medan tidigare kom rapporten upp i en ny flik och om man ville ändra på datan till rapporten måste man gå tillbaka och söka efter en ny rapport, som sedan öppnades igen i en ny flik.

Syftet med den teoretiska delen av examensarbetet är att beskriva om olika rapporteringslösningar i allmänhet samt på vilka olika sätt det är möjligt att göra rapporter och vad för program man kan göra detta med. I examensarbetet utvärderas även resultatet dvs. huruvida rapporteringssystemet är lämpligt för företaget och om det uppfyller kraven. De olika valen för rapportens förverkligande kommer även att beskrivas och motiveras.

1.3 Metoder

För detta examensarbete har två väsentliga metoder tillämpats. Den första metoden är en kravanalys var man söker information om vad som fattas i det nuvarande rapporteringssystemet och vad man önskar att skulle komma med i det nya systemet. För att få svar på detta ser man vad kunder har sagt och vad man inom företaget har önskat och tycker att skall komma med till det nya rapporteringssystemet. Från dessa krav kan man sedan gå igenom bättre vad de slutliga kraven för detta rapporteringssystem är.

När lösningen var färdig gjordes en utvärderingsrapport var man frågade personer i företaget om hur resultatet blev. Man fick veta om resultatet blev tillräckligt bra och om det är som önskat sig, samt om det går att användas i programvaran. Denna utvärdering skall alltså berätta om man vill använda denna lösning eller om man måste hitta en annan möjlighet till detta problem. Personerna som kommer att svara på denna rapport är personer inom företaget som har att göra med programvaran och varit med och hjälpa i detta arbete.

1.4 Avgränsningar

För att inte ha ett för brett ämne gjordes det en del avgränsningar. I programvaran Granlund Manager finns det flera rapporter, men för att avgränsa detta examensarbete behandlar det endast om att förnya en gammal rapport, än att förnya alla rapporter i programvaran. Det är också onödigt att förnya alla rapporter i detta examensarbete eftersom det endast är en POC och man inte ännu är säker om resultatet kommer att användas.

1.5 Struktur

I Kapitel 2 av detta arbete presenteras kravanalysen för rapporteringssystemet och där beskrivs vad en kravanalys är och vilka olika typers krav det finns för det nya rapporteringssystemet. Från arbetets krav kan man sedan börja med utvecklingen av rapporteringssystemet i Kapitel 3. I kapitlet behandlas vilket bibliotek som valdes för att göra rapporteringssystemet, samt så beskriver man steg för steg hur man designade rapporten och hur man fick den implementerad till programvaran Granlund Manager. Till sist utvärderas resultatet i Kapitel 4 av svar som man fått av arbetare i företaget, som haft något med arbetet att göra. Från utvärderingen ser man om resultatet uppnådde kraven och vad man tyckte om resultatet, samt om resultatet går att användas i produktionen eller måste man hitta på en annan lösning för ett nytt rapporteringssystem.

2 KRAVANALYS

2.1 Vad är en kravanalys?

I en kravanalys försöker man se om programvaran har någonting som fattas, så man har en så bra programvara som möjligt. Utan en kravanalys kan det hända att man försöker lösa fel problem och då förlorar man tid och pengar, samt gör kunderna besvikna. Kravanalysen är det första tekniska steget i programutvecklingsprocessen (Pressman och Ince, 2000 s. 288).

Kravanalysen börjar med att man försöker få fram (Eng. Elicit) alla krav av kunderna samt programutvecklarna, därefter analyserar man kraven och ser om de är klara och går att lösas. Efter detta kan man börja med projektet och utvecklingen. Det är inte en bra idé att göra hela projektet färdigt genast för då kan det uppstå problem. Det kan hända att man missförstått helt och inte gjort alls det som ex. kunden ville ha, dessutom är det enligt Pressman och Ince (2000 s. 289) också många gånger inte möjligt att specificera ett problem i början av ett projekt. För att hitta problemen eller inte göra fel saker kan det vara bra att använda sig av Agil systemutveckling (Eng. Agile Software Development) eller att göra en prototyp av projektet. Dessutom är också kommunikationen mellan alla kravgivare väldigt viktiga, för att projektet skall gå smidigt utan problem framåt. I Agil systemutveckling finns det flera olika metoder man kan använda sig av, men en av de mest kända är Scrum. Scrum är ett ramverk för projektledning som framhäver grupparbete, ansvarighet och iterativa processer mot ett väl definierat mål (Rouse, 2017). Om man använder sig av Scrum i sitt projekt betyder det att man delar upp projektet i mindre delar och gör på t.ex. endast några funktionaliteter under en vecka. Med regelbundet samarbete och möten gör man delmål som gör arbetet lättare att lösa än att genast försöka nå kravet av hela arbetet. Detta gör det också lättare att i en kravanalys se om kraven uppnås och om man har förstått rätt vad kunden vill ha. En prototyp gör man däremot för att kunden kan testa och se om det är så som kunden hade tänkt sig att det skulle se ut. En prototyp är bra för att man kan visa den till kunden och fråga om båda parterna hade tänkt lika. Då vet man redan i ett tidigare skede om kunden vill ha några förändringar och man slösar inte tid. Dessutom är en prototyp bra om t.ex. kunden inte är

helt säker på vad den vill ha, så kan man göra en prototyp för att ha någonting och visa och planera vidare därifrån.

Det är även vanligt att man bryter upp kraven i olika funktionella krav (Bergfeldt-Trotzig, 2017). Ett funktionellt krav kan vara t.ex. att man tillägger en knapp i programvaran som öppnar en ny sida när man klickar på den. Om man använder sig av Scrum skulle ett funktionellt krav vara en user story. User stories är de kraven som man har till projektet och man vill att skall göras och varje user story är en funktion som man vill ha med i programmet. Medan icke funktionella krav igen är mer tekniska och i exemplet ovan skulle det kunna vara att den nya sidan skall laddas upp inom x antal sekunder.

2.2 Kravanalys på det nya Rapporteringssystemet

De tekniska krav som kom på Granlund Managers nya rapporteringssystem kom från personer i företaget som har med programutvecklingen att göra, samt företagets kunder. Under möten planerade vi hur vi vill att resultatet skall se ut när det är färdigt och hur systemet skall fungera. De funktionella kraven till systemet var att det finns en graf var datan visas på och bredvid grafen skall det finnas filtreringsmöjligheter var man kan fritt filtrera data i grafen. I det gamla rapporteringssystemet kunde man inte ändra på någonting i rapporten när den var öppen, utan endast före man öppnade rapporten. Därför önskar man nu att man skulle kunna filtrera och sortera informationen i rapporten när den är synlig. Man skall också kunna ändra på tidsplanen att mellan vilka datum man vill att datan skall komma ifrån och att man skall ha en möjlighet att välja om man vill att datan visas t.ex. enligt varje månad eller år. Man vill också att rapportens data skall presenteras i realtid, för att till det gamla rapporteringssystemet fick man inte data från idag, utan med en dags fördröjning. En kund som använder programvaran kan ha flera byggnader i programmet och varje byggnad har egen data. Därför vill man att när man byter till en annan byggnad skall datan i rapporten också ändras. Ett icke funktionellt krav som fanns var att försöka att inte det nya rapporteringssystemet skulle göra den nuvarande programvaran långsammare och att när man väljer ett filtreringsalternativ skall man snabbt få den uppdaterade datan fram.

Till uppbyggandet av rapporteringssystemet användes prototypmodellen. Denna typ av arbete passade bättre för att jag jobbade ensam och då kunde man efter ett skede i arbetet visa prototypen och få kommentarer om det är bra eller om någonting skall ändras på, samt vad som behövs göras i fortsättningen av arbetet.

Kraven på rapporteringssystemet var i korthet följande:

Som användare vill man ha möjlighet att göra detta i Granlund Manager:

- Filtrera och sortera rapportens data
- Välja till rapporten vilken nivå som helst av objekthierarkin
- Filtrera och ändra på rapportens tidsplan dynamiskt och fritt
- Se rapportens information i realtid och inte med en dags fördröjning

Som Granlund Managers programutvecklare vill man:

- Inte göra programvarans funktioner långsammare med rapporteringssystemet
- Att rapporten öppnas och uppdateras snabbt

Enligt dessa krav från kravanalysen planerades det att målet skulle vara att man skulle ha en graf där mängden data ses under en viss tid. På sidan av grafen skall det finnas filtreringsmöjligheter. Filtreringsmöjligheterna skall vara i rullgardinsmenyer, så man kan välja vad för data man vill att skall visas i grafen. Dessutom skulle det komma en datumväljare att kravet med att kunna filtrera och ändra på tidsplanen skulle bli möjligt.

3 UTVECKLINGEN AV RAPPORTERINGSSYSTEMET

3.1 Val av bibliotek

Första steget i utvecklingen av rapporteringssystemet var att hitta på en lösning för att kunna göra rapporter med, samt att denna lösning möjliggör att skapa och hantera rapporter enligt de krav som ställts under kravanalysen. Lösningen bör också kunna implementeras till programvaran Granlund Manager, som är uppbyggt med ASP.NET och C#. Till

rapporteringsystemet ville man ha en graf och för att kunna göra en graf behöver man ett JavaScript ramverk som man kan göra detta med eller använda sig av ett bibliotek som man kan designa och visualisera rapporter med.

Om man väljer att använda sig av ett bibliotek som man kan designa och visualisera rapporter med behöver man inte göra alla funktioner eller visualisera rapporten själv, för det sköter biblioteket oftast om. Därför är det lättare att använda sig av ett bibliotek för det gör nästan allt färdigt för dig. Medan om man använder sig av ett JavaScript ramverk skulle man t.ex. i detta arbete behöva programmera alla rullgardinsmenyer själv, samt alla funktioner som gör att grafen uppdateras efter val av filter. Ofta är det i alla fall det bättre alternativet för med ett JavaScript ramverk får man göra allt så som man själv vill ha det och att alla funktioner gör de sakerna som man vill att de skall göra. Medan med ett bibliotek som man kan designa och visualisera rapporter med kan det vara svårt att ändra på saker till sådana som man själv vill ha dem, för att en sådan lösning inte finns i biblioteket.

Uppdragsgivaren hade flera alternativ på val av tillvägagångssätt för förnyandet av rapporterna, men man beslöt att använda sig av ett bibliotek som man redan har licens till och kolla vad biblioteket har att erbjuda. Biblioteket heter Syncfusion och har några alternativ som skulle passa för att göra ett rapporteringssystem. Därför ville man se om detta bibliotek skulle vara ett bra alternativ för att få ett fungerande rapporteringssystem för programvaran Granlund Manager.

3.1.1 Syncfusion

Syncfusion är en ledande leverantör av företags-klassad utveckling av komponenter och bibliotek, som levererar en bred mängd av UI, rapportering och funktionalitet för företagsintelligens på varje större Windows plattform (ComponentSource). Syncfusion var biblioteket som valdes att använda för att göra rapporteringssystemet i detta examensarbete till programvaran Granlund Manager. Syncfusion erbjöd två olika produkter som man kunde göra rapporter med. Ett av dem var Report Platform och den andra var Dashboard Platform. Report Platform är ett rapporteringsverktyg som man kan göra rapporter med, men det var liknande som Microsoft ReportViewer som redan fanns i programvaran

Granlund Manager. I Syncfusions Report Platform designar man sin rapport och sparar den som en fil, som man sedan läser till din programvara och därför ville man inte använda denna produkt. Istället valde man Dashboard Platform som är ett verktyg med vilket man kan designa sin egna instrumentbräda med. Dashboard Platform verkade som ett bättre alternativ för man kunde tillsätta filtreringsalternativ till instrumentbrädan, samt så verkade denna produkt som ett mera dynamiskt alternativ. Filtreringsmöjligheter och ett dynamiskt rapporteringssystem var det som man var ute efter, så därför ville man se vad man kan få ut av denna produkt som man redan hade licens till.

Till Syncfusion Dashboard Platform hördes en Dashboard Designer med vilken man designar instrumentbrädan till sådan man vill ha den. I designern fanns det flera olika sorters grafer, rullgardinsmenyer, tabeller och mycket mera, som man kunde lägga till instrumentbrädan. Efter man designat instrumentbrädan kunde man implementera den till programvaran med hjälp av en Dashboard Viewer som också tillhörde produkten Dashboard Platform. Denna process beskrivs noggrannare i nästa kapitel var man designade och sedan tillämpade rapporten till programvaran Granlund Manager.

3.2 Uppbyggnaden av rapporten

Syncfusion Dashboard Designer användes för att göra rapportens design och tillämpning av data till den. Syncfusions Dashboard Designer kan laddas ner från Syncfusions hemsidor (Syncfusion), om man har en licens. Med Syncfusion Dashboard Designer försökte man få en rapport som skulle fungera enligt kraven. Designern hade flera olika alternativ för grafen, men man valde en kolumngraf eftersom det verkade som det bästa sättet för att man lätt skulle kunna se värdena i grafen. Det var också möjligt att ha rullgardinsmenyer i rapporten, samt en datumväljare som behövdes för att kunna filtrera under vilken tid datan skall ses i instrumentbrädan. Rullgardinsmenyerna och datumväljaren var nödvändiga för att få instrumentbrädan att fungera så som man ville.

Nästa steg var att försöka få någon data kopplad till instrumentbrädan. Man ville alltså att grafen uppdateras efter att man valt ett filter och för att detta var endast för att testa om allt fungerar gjordes en JSON fil som innehöll testdata. Denna data kopplades till rapporten för att kunna få något att synas i fälten. Bra med designern var att efter man hade

kopplat en datakälla till instrumentbrädan kunde man bara dra kolumnens namn till t.ex. en rullgardinsmeny och efter det kunde man se denna kolumnens data i rullgardinsmenyn (se figur 3). Syncfusions Dashboard Designer var lätt att använda och detta var ett exempel på varför det var det. På grund av att designern var så lätt att använda gjorde att man snabbt fick gjort en testrapport för att se hur en rapport kunde se ut och i figur 1 kan man se ett exempel på hur denna Syncfusion Dashboard ser ut efter att man gjort den i designern och tillämpat data till den.

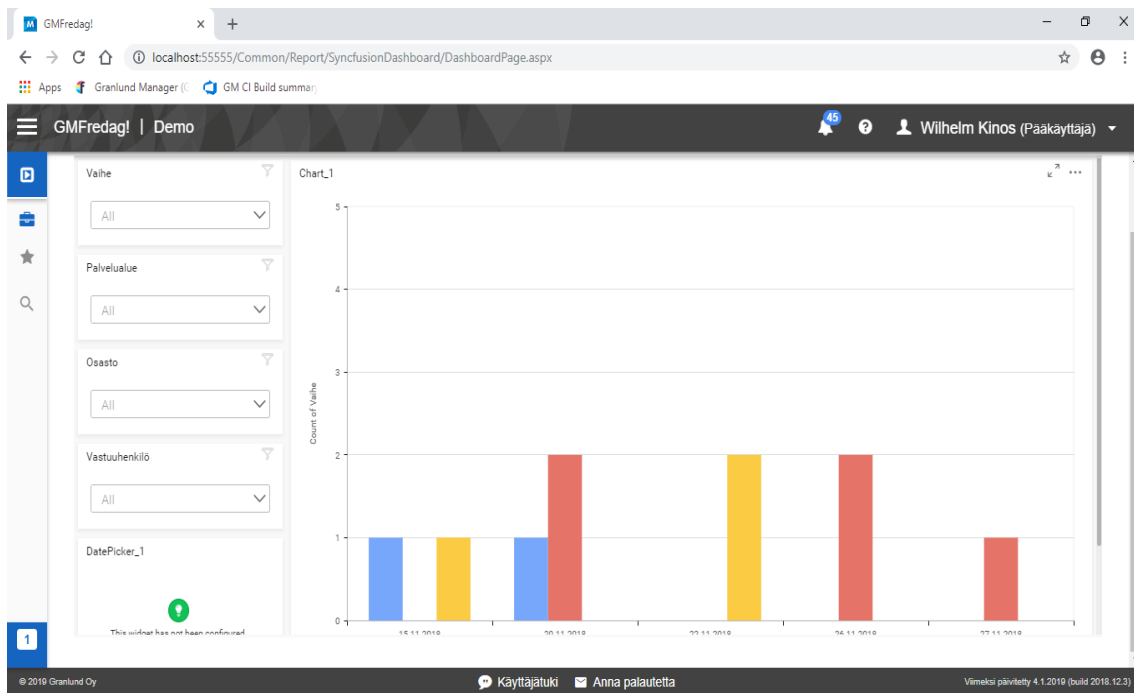


Figur 1. Ett exempel på hur en Syncfusion Dashboard kan se ut

3.3 Implementering av en testrapport i Granlund Manager

För att få testrapporten implementerad i Granlund Manager behövdes Syncfusion Dashboard Platform SDK som inkluderar behövliga skript och teman för att få instrumentbrädan att vara synlig i programvaran. Syncfusion Dashboard Platform SDK kan laddas ner från Syncfusions hemsidor (Syncfusion), men man måste ha en licens för att kunna göra det. Skripten och teman laddades ner och tillsattes till en ny mapp i programvarans kod för att ha tillgång till dessa i ett senare skede.

För att få testrapporten implementerad i programvaran var det lättast att börja från en helt tom sida var testrapporten sedan skulle komma och som man kom till från en länk i programvaran. Programvaran Granlund Manager är uppbyggd med ASP.NET och C# vilket man använde för att få rapporteringssystemet med i programvaran. ASP.NET är ett gratis webbramverk för att bygga websidor och webapplikationer med att använda HTML, CSS, JavaScript (Microsoft, 2010). ASP.NET är bra för att man kan ha en masterfil som innehåller layout och funktioner som man vill att skall finnas på alla sidor i programvaran, som t.ex. menyn. Därför började man med att göra en ASP.NET Page som är kopplad till masterfilen för att få menyn och annat utseende man vill ha med på rapportsidan. Detta kan man göra automatiskt när man gör en ny fil, då finns det ett alternativ för att göra en ASP.NET Page fil som är kopplad till en masterfil. När man gör en ny ASP.NET Page gör den automatiskt också en annan fil, som har serversidans kod och är skriven med C#. Denna fil brukar kallas ”code behind” och där brukar finnas applikationens logik t.ex. om man vill att efter en knapptryckning skall en ny sida öppnas, då finns funktionen till detta i denna fil. På serversidan tillsattes de nya skripten och teman till sidans header, så de laddas när sidan öppnas. Ofta har man också skript i masterfilen, men det är inte en bra idé att laga dem där för då laddas de på alla sidor som använder sig av masterfilen fast de inte används. Då kan stora skript också göra funktionaliteten av programmet långsammare. Ett skript som läser och gör instrumentbrädan skall komma i en ny ASP.NET User Control fil, som skall innehålla HTML koden för detta. För att User Control filens HTML kod skall synas i en browser måste den kopplas till en ASP.NET Page. Därför lade man också till i Page filens serversida att den skall läsa User Control filen för att få HTML koden. HTML koden bestod av en div-tag var instrumentbrädan skulle komma och av ett skript där man söker instrumentbrädan och bygger upp den. I skriptet tillsattes också testrapportens höjd och bredd, för att testrapporten skulle ha en minimum höjd och bredd för att vara synlig. Utan dessa fanns det en möjlighet att det inte uppstod fel, men instrumentbrädan ändå inte blev synlig.



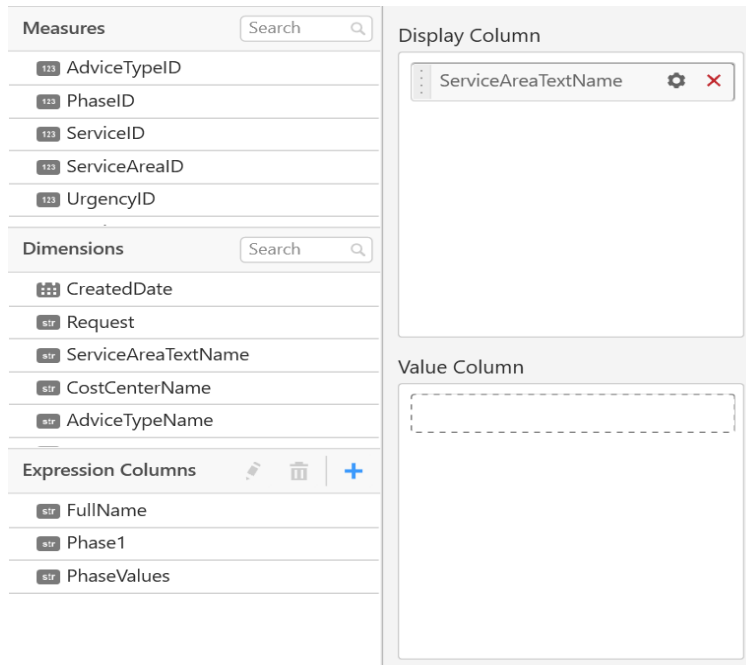
Figur 2. En Syncfusion Dashboard implementerad till programvaran Granlund Manager

I figur 2 kan man se hur en Syncfusion Dashboard ser ut när den är implementerad till programvaran Granlund Manager när alla steg är gjorda. På vänstra sidan finns alla rullgardinsmenyer som gör att man kan filtrera datan som finns i grafen som synas på högra sidan.

3.4 Tillämpning av data till rapporten

Testrapporten använde sig endast av en självgjord JSON fil för att få någon data till rapporten, men nu vill man ha data från databasen istället. I Syncfusions Dashboard Designer hade man möjlighet att välja hur man vill koppla data till rapporten. Som alternativ kunde man välja mellan en Microsoft Excel fil, SQL, Web API eller en JSON fil som tidigare och på flera andra sätt. Till detta examensarbete valde man att använda sig av SQL för att få data från databasen. Till sökningen av data användes SQL Server Management Studio (SSMS), för att göra queryn och se att den returnerar rätt data. SSMS används för att göra queryn, designa och hantera din databas (Microsoft, 2018). När man hade gjort queryn och sett att den returnerar den rätta datan var det endast att lägga till den färdiga queryn till Syncfusions Dashboard Designer. För att få data synligt i rapporten kunde man välja

vilken av databaskolumnerna som skulle visas i vilket fält, med att dra rätt kolumn till rätt komponents fält (se figur 3).



Figur 3. En bild som visar hur man kunde dra en kolumn från sin data till ett visst fält i Syncfusion Dashboard Designer

Målet med examensarbetet var att få en dynamisk rapport till programvaran Granlund Manager. Därför ville man ha möjlighet att ändra på datan som är kopplad till rapporten för att få ny data synlig i rapporten. Det visade sig att de inte var möjligt att koppla en helt ny query till rapporten. Man kunde endast ändra på ett värde i queryns *WHERE* sats, vilket gör att man får andra rader från databasen, men endast från samma query. Detta var bra i flera sökningar, men hjälpte inte med alla, som gjorde att man inte fick rapporten så dynamisk som man ville ha den. Man ville också ha möjligheten att skicka data från koden till rapporten, för att inte göra samma databassökningar om och om igen. Det visade sig att endast vara möjligt att göra databassökningen i designern och få data från den. Därför ville man prova om det skulle lyckas med någon av de andra metoderna att få data till rapporten. Man valde att använda sig av Web API och få data till rapporten på det sättet. För att få fram datan i rapporten från API:n måste man koppla sig till den med OAuth. OAuth är ett simpelt sätt att publicera och koppla till en skyddad data (OAuth). Efter detta

var det bara att koppla rätt data till rätt ställe i rapporten som när man kopplade datan från SQL.

Med Web API:n märkte man igen att man inte får rapporten så dynamisk som man ville ha den. Man ville ha möjlighet att ändra på parametern som var i API url:en för att kunna ändra på datan som var kopplad till rapporten, men detta visade sig inte vara möjligt. Man kunde endast ha en och samma url hela tiden kopplad till rapporten. Detta gjorde att resultatet inte blev helt som man ville fast rapporten på alla andra sätt fungerade helt som man ville ha den.

4 UTVÄRDERING

4.1 Datainsamlingsmetod

För att få en utvärdering på detta rapporteringssystem skickade jag till några personer i företaget några frågor, som de sedan kunde fritt svara på (Bilaga 1). För att göra frågorna och lätt få svar på dem använde jag mig av Google Forms. Jag beslöt att använda mig av Google Forms för att lättare kunna se vad alla har svarat på en viss fråga och få svaren direkt när de har svarat på dem. Alla frågor var öppna fält var man fick svara med egna ord. Det var ett bättre alternativ för att få noggranna svar, än om det skulle ha varit ja/nej frågor.

Personerna som jag skickade frågorna till jobbar alla inom systemutvecklingen av programvaran Granlund Manager och har haft på något sätt att göra med detta examensarbete. Två stycken jobbar med ledningen av systemutvecklingen och de har gett kraven till detta arbete och en är programmerare för programvaran och har hjälpt mig när jag har undrat något med programmeringen eller behövt hjälp. Med att skicka frågor till personer som jobbar med olika saker får jag svar från flera synvinklar och vissa kan svara på en fråga bättre än den andra.

4.2 Utvärderingen av Syncfusion Dashboard

De första frågorna i enkäten handlade om Syncfusion Dashboard och vad man hade för åsikter om det valda biblioteket. Man tyckte att bibliotekets designer Syncfusion Dashboard Designer var bra och att man kunde göra det mesta med den och att rapporterna ser bra ut. Förutom detta fanns det ändå saker som man inte gillade med Syncfusion. För att få rapporten att synas i programvaran Granlund Manager behövdes en väldigt stor JavaScript fil. Dokumentationen för Syncfusion Dashboard var inte den bästa och det var svårt ibland att få en lösning gjord på grund av detta. Dessutom ville man använda sig av databasförfrågningar som redan fanns i programvarans kod, men det var inte möjligt. Man kunde endast ha en skild databasförfrågning som gjordes i Syncfusions Dashboard Designer.

När man använder sig av tredje partens komponenter måste man ofta göra kompromisser med kraven för att få ett bra resultat. Detta var svaret till frågan om man med Syncfusion Dashboard kunde göra allt vad man ville göra. Dessutom var ett av de största kraven i detta examensarbete att rapporteringssystemet skulle vara dynamiskt och för att man inte kunde ändra på datan som var kopplad till rapporten var detta inte möjligt. Detta ville man absolut ha med i rapporteringssystemet och detta var orsaken till att man inte ville fortsätta att utveckla på detta rapporteringssystem eller använda denna lösning i produktionen. I alla fall tyckte man att Syncfusion var ett bra val, för att man ville veta om Syncfusion som man redan hade licens till skulle gå att användas som ett rapporteringssystem.

4.3 Utvärderingen av rapporteringssystemets lämplighet för Granlund Manager

Rapporteringssystemet blev inte det som man ville för att man inte fick rapporten att bli så dynamisk som man ville ha den. Detta var den största orsaken till att man inte fortsatte att vidareutveckla på detta rapporteringssystem. Majoriteten tyckte ändå att examensarbetet inte var onödigt, för att arbetet var en POC och man sökte vad allt Syncfusion kan göra och om man skulle kunna använda Syncfusion som ett rapporteringssystem.

Dessutom fick företaget också en bra bild över kraven för ett rapporteringssystem i alla möten under examensarbetets gång där man gick igenom vad kraven är för ett nytt rapporteringssystem i programvaran Granlund Manager.

Slutresultatet av utvärderingen var att Syncfusion inte var ett bra alternativ eftersom det inte uppfyller alla krav på rapporteringssystem. I fortsättningen kommer företaget att satsa på större rapporteringssystem som t.ex. MS Power BI. Power BI är en business analytisk service som levererar insikter för att göra det möjligt att göra snabba och informerade beslut (Microsoft, 2019).

5 SLUTSATSER

Bakgrunden med detta arbete var att Granlund ville ha ett nytt rapporteringssystem till deras programvara Granlund Manager. Deras nuvarande rapporteringssystem är föråldrat samt långsamt och man vill förnya det till en mer modern och snabbare lösning. Målsättningen med detta examensarbete var att göra en POC på ett rapporteringssystem med hjälp av ett bibliotek. Man ville ha ett dynamiskt rapporteringssystem där man med hjälp av filter skulle kunna ändra på datan i rapporten och att rapporten ändras automatiskt efter att man valt ett filter.

Utvecklingsprocessen började med att samla kraven av företaget för det nya rapporteringssystemet med hjälp av en kravanalys. Kraven fick man från företaget, samt deras kunder och från dem kunde man på möten gå igenom bättre vad de slutliga kraven är till detta arbete. De slutliga kraven var att få en dynamisk graf som uppdateras då man väljer ett av filtren som skulle finnas med i rapporteringssystemet.

Från kraven kunde man sedan välja hur man skulle göra det nya rapporteringssystemet och man valde att välja Syncfusion. Företaget hade redan licens till Syncfusion och man ville därför veta om man med Syncfusion Dashboard Platform kunde göra ett rapporteringssystem till programvaran Granlund Manager. Man designade rapporten och tillsatte data till den med Syncfusion Dashboard Designer som fanns med i produkten Dashboard Platform. Efter att rapporten var gjord tillämpade man den in i programvaran Granlund Manager enligt Syncfusions dokumentation. Detta visade sig också vara det svåraste i

detta arbete, för Syncfusions dokumentation inte var den bästa och för att Granlund Manager redan färdigt använder sig av andra skript, som gjorde att de lite krockade och man måste få dem att köras i rätt ordning.

Rapporteringsystemet blev visuellt som man ville ha det och rullgardinsmenyerna gjorde att man kunde filtrera datan i rapporten. Man kunde inte ändra datan till rapporten som man ville göra utan samma data var kopplad till den hela tiden. Detta gjorde att rapporten inte blev så dynamisk som man ville ha den. Ett mer dynamiskt rapporteringssystem var det största kravet man hade till rapporteringssystemet, vilket gjorde att man inte kunde fortsätta att utveckla på detta rapporteringssystem eller använda det i produktionen. I utvärderingen över arbetet frågade man personer i företaget om vad de tyckte och det visade sig att arbetet i alla fall var nödvändigt. Arbetet var endast en POC och man ville veta om Syncfusion skulle gå att användas som ett rapporteringssystem och som resultat av detta examensarbete fick man svar på denna fråga. Nu kunde man stryka över Syncfusion som ett alternativ för ett rapporteringssystem och fortsätta att utveckla på ett nytt rapporteringssystem på något annat sätt.

Syncfusion visade sig inte vara det bästa för att kraven inte uppfylldes. Syncfusion verkade vara bra med att få bra och fina rapporter visuellt, men det visade sig vara svårt att få datan som var kopplad till rapporten ändrad. Det skulle eventuellt ha varit bättre med ett annat bibliotek eller användningen av ett JavaScript bibliotek som man kan göra grafer med. I fortsättningen kommer man i företaget att försöka nå bättre resultat med att använda sig av ett JavaScript bibliotek och med hjälp av det visa grafer i programvaran Granlund Manager. Man kommer också att prova större rapporteringssystem som MS Power BI för att se om man får bättre resultat med detta bibliotek.

KÄLLOR

- Bergfeldt-Trotzig, S., *Knowit Blogs*, 2017. Tillgänglig: <https://blogg.knowit.se/experience-se/digitalkommunikation/2017/02/kravhantering-i-ett-scrum-projekt> Hämtad: 11.2.2019
- ComponentSource, Tillgänglig: <https://www.componentsource.com/brand/syncfusion> Hämtad: 26.02.2019
- Got ReportViewer, Tillgänglig: <http://www.gotreportviewer.com/> Hämtad: 3.5.2019
- Granlund, Tillgänglig: <https://www.granlund.fi/granlund/meista/> Hämtad: 3.5.2019
- Microsoft, 2010, Tillgänglig: <https://docs.microsoft.com/en-us/aspnet/overview> Hämtad: 18.02.2019
- Microsoft, 2018, Tillgänglig: <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017> Hämtad: 1.3.2019
- Microsoft, 2019, Tillgänglig: <https://powerbi.microsoft.com/en-us/> Hämtad: 3.5.2019
- OAuth, Tillgänglig: <https://oauth.net> Hämtad: 18.4.2019
- Rouse, M., *SearchSoftwareQuality*, TechTarget, 2017. Tillgänglig: <https://searchsoftwarequality.techtarget.com/definition/Scrum> Hämtad: 14.2.2019
- Syncfusion, Tillgänglig: <https://www.syncfusion.com> Hämtad: 1.3.2019
- Pressman, R.S. & Ince, D., 2000, *Software Engineering*, Fifth Edition, McGraw-Hill International (UK) Limited, London

BILAGOR

Bilaga 1. Frågeformuläret som skickades för utvärderingen

4/3/2019

Lopputyö ja Syncfusion Dashboard

Lopputyö ja Syncfusion Dashboard

Arviointi lopputyöhöni: Syncfusion Dashboardin mielipiteet ja sen soveltuvuus Granlund Manageriin

1. Hyvät ja huonot puolet Syncfusion Dashboardista?

2. Pystyikö Syncfusion Dashboardilla tekemään kaikkea mitä haluttiin?

3. Oliko Syncfusion oikea valinta tähän työhön?

4. Miksi valittiin että emme jatkokehitä tätä tulosta tai käytetä tätä tulosta tuotannossa?

5. Oliko työ silti hyödyllinen vaikka emme jatkettu kehittämään sitä?

6. Muuta?
