

# Självständigt arbete på grundnivå

*Independent degree project - first cycle*

Datateknik  
*Computer Engineering*

***Dashboard***  
*Realtidsvisualisering av flödet mellan skog och industri*

**Andrea Jacobsson**



**Mittuniversitetet**

MID SWEDEN UNIVERSITY

Campus Härnösand Universitetsbacken 1, SE-871 88. Campus Sundsvall Holmgatan 10, SE-851 70 Sundsvall.

Campus Östersund Kunskapens väg 8, SE-831 25 Östersund.

Phone: +46 (0)771 97 50 00, Fax: +46 (0)771 97 50 01.

**MITTUNIVERSITETET**  
**Avdelningen för informations-och kommunikationssystem (IKS)**

**Examinator:** Ulf Jennehag, ulf.jennehag@miun.se

**Handledare:** Magnus Eriksson, magnus.eriksson@miun.se

Carl Lundin, Sogeti, carl.lundin@sogeti.se

**Författare:** Andrea Jacobsson, anja1403@student.miun.se

**Utbildningsprogram:** Datateknik, 180 hp

**Termin, år:** 6, 2018

## Sammanfattning

Big data är ett begrepp som idag används för att beskriva stora datamängder. Målet med arbetet har varit att utveckla ett effektivt system som analyserar stora mängder data. I det här fallet transportflödet av virke mellan skog och industri. Genom att hantera den här mängden data får användaren snabbt fram information om vart virket befinner sig och möjlighet till att jämföra historiska data. Den här rapporten beskriver arbetet med att implementera en dashboard som i realtid sammanställer Big data på ett begripligt sätt. Projektet använder sig av scrum som projektverktyg. Det betyder ett iterativt arbete där man arbetar mot delmål. Rapporten innehåller även utvärderingar som genomförts med användbarhet i fokus samt en sammanställning av forskning inom området datavisualisering och BI. Utvärderingen visade att med den utvecklade dashboarden så tog det för en normal datoranvändare i snitt 3,4 sekunder att hitta den eftersökta information. Tidigare så krävdes det programmeringskunskaper och betydligt längre tid för att få fram motsvarande information.

**Nyckelord:** Dashboard, .NET, användbarhet, BI, datavisualisering

## Abstract

Big data is a term used to describe large amounts of data. The aim of the work has been to develop an efficient system that analyzes large quantities of data. Which in this case is the transport of wood between forest and industry. By managing this amount of data, the user can quickly receive information about the location of the timber and compare historical data regarding it. This report describes the work of implementing a dashboard that compiles "Big data" in real time and in a comprehensible way. The project uses scrum as a project tool. This means an iterative development where you work towards smaller sub-goals. The report also includes evaluations conducted with a focus on usability and also a compilation of research within the subject of data visualization and BI. The evaluation indicated that by using the developed dashboard, it took a standard computer user an average of 3.4 seconds to find the requested information. Before, programming skills would be required and it would take much longer to obtain corresponding information.

**Keywords:** Dashboard, .NET, usability, BI, data visualization

## Förord

Det här examensarbetet har genomförts hos Sogeti AB i Sundsvall under tidsperioden januari 2018 till juni 2018. Jag vill börja med att tacka alla de anställda på Sogeti som stöttat mig i mitt arbete, framförallt Hans Gagner, Judith Karlsson och Kristina Löfbom Englund. Jag vill även tacka min handledare på skolan Magnus Eriksson för all hjälp med rapportskrivningen.

Jag vill avsluta med att särskilt tacka min handledare på Sogeti, Carl Lundin. Som med stort engagemang väglett mig under hela arbetet.

# Innehållsförteckning

<b>Sammanfattning</b> .....	<b>iii</b>
<b>Abstract</b> .....	<b>iv</b>
<b>Förord</b> .....	<b>v</b>
<b>Terminologi</b> .....	<b>viii</b>
<b>1 Inledning</b> .....	<b>1</b>
1.1 Bakgrund och problemmotivering.....	1
1.1.1 Sogeti.....	1
1.1.2 GPS Assisted Transport Announcement.....	1
1.2 Högnivåproblemformulering.....	2
1.3 Avgränsningar.....	2
1.4 Konkreta och verifierbara mål.....	3
1.5 Översikt.....	3
1.6 Författarens bidrag.....	3
<b>2 Bakgrundsmaterial</b> .....	<b>4</b>
2.1 .NET-ramverk.....	4
2.2 Backbone.js.....	4
2.3 C#.....	4
2.4 Nyckeltal.....	4
2.5 LINQ.....	4
2.6 Microsoft Azure Application Insights.....	5
2.7 Scrum.....	5
2.8 Datavisualisering.....	6
2.8.1 Närhetens lag.....	6
2.8.2 Statiska och dynamiska modeller.....	6
2.9 Dataöverföring.....	7
2.9.1 AJAX.....	7
2.9.2 RESTful webbtjänst.....	7
2.9.3 WebSockets.....	7
2.10 Cachelagring.....	7
2.11 Tidigare arbeten.....	8
2.11.1 Stephen Few.....	8
2.11.2 Updated DeLone and McLean IS Success Model.....	9
2.11.3 Evgeniy Gorodov och Vasiliy Gubarev.....	10
<b>3 Metod</b> .....	<b>11</b>
3.1 Designprocess.....	11
3.2 Bedömningskriterier.....	12
3.3 Eyetracking.....	12
<b>4 Konstruktion</b> .....	<b>13</b>
4.1 Kravspecifikation.....	13
4.1.1 Förutsättningar.....	13
4.1.2 Information som ska visas i systemet.....	13

---

4.2	Systemmodell.....	14
4.2.1	Dataöverföring.....	14
4.2.2	Klienten.....	17
4.2.3	Användargränssnitt.....	18
<b>5</b>	<b>Resultat.....</b>	<b>20</b>
5.1	Empirisk utvärdering.....	20
5.1.1	Användbarhetstest 1.....	20
5.1.2	Användbarhetstest 2.....	22
5.2	Teoretisk utvärdering.....	22
5.3	Sammanfattning.....	23
<b>6</b>	<b>Slutsatser.....</b>	<b>24</b>
6.1	Slutsatser.....	24
6.2	Samhälleliga och etiska aspekter.....	25
6.3	Framtida arbete.....	25
	<b>Källförteckning.....</b>	<b>26</b>
	<b>Bilaga A: Resultat från användbarhetstest 2.....</b>	<b>29</b>

# Terminologi

## Akronymer

AJAX	Asynchronous JavaScript and XML
APM	Application Performance Management
BI	Business Intelligence
CSS	Cascading Style Sheets
EIS	Executive Information System
GATA	GPS Assisted Transport Announcement - Ett system framtaget av Sogeti för att förbättra och effektivisera styrningen av SCA Skogs transporter.
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LINQ	Language-Integrated Query
MVC	Model View Controller
OLAP	Online Analytical Processing
REST	Representational State Transfer
SQL	Structured Query Language
TCP	Transmission Control Protocol
VIOL	Virke Online
XML	eXtensible Markup Language

# 1 Inledning

I det här kapitlet presenteras projektets bakgrund, kort information om SCA's befintliga system samt konkreta och verifierbara mål.

## 1.1 Bakgrund och problemmotivering

Big data är ett begrepp som idag ofta används för att beskriva stora datamängder. Data av den storleken är svårbehandlad vilket kräver mer av dagens system. Det vi idag talar om som en dashboard (sv. instrumentpanel) kommer i grunden från EIS, Executive Information System [1]. EIS är ett system tänkt att fungera som stöd inom beslutsfattning [2]. Systemet presenterar, för företaget, relevant information om organisationen och lanserades under 1980-talet. Vidare under 1990-talet utvecklades BI, Business Intelligence, Online Analytical Processing (OLAP) och Data warehouse som exempel på andra system vilka alla samlar, analyserar och presenterar data för användaren [1]. Few [1] beskriver en dashboard som en display vilken lyfter fram den viktigaste informationen på ett enkelt och överskådligt sätt.

Hjärnan hos en människa har kognitiva begränsningar. Detta ihop med att anställda idag förväntas kunna läsa och dra slutsatser från allt större mängder data kan medföra en känsla av otillräcklighet hos den anställde. Detta då det blir för svårt för det mänskliga sinnet att ta in och förstå.

### 1.1.1 Sogeti

Sogeti [3] är ett dotterbolag till Capgemini SE. Vilka arbetar med att leverera konsult-, teknik- och outsourcingtjänster. I Sverige har Sogeti 1150 anställda och 21 kontor.

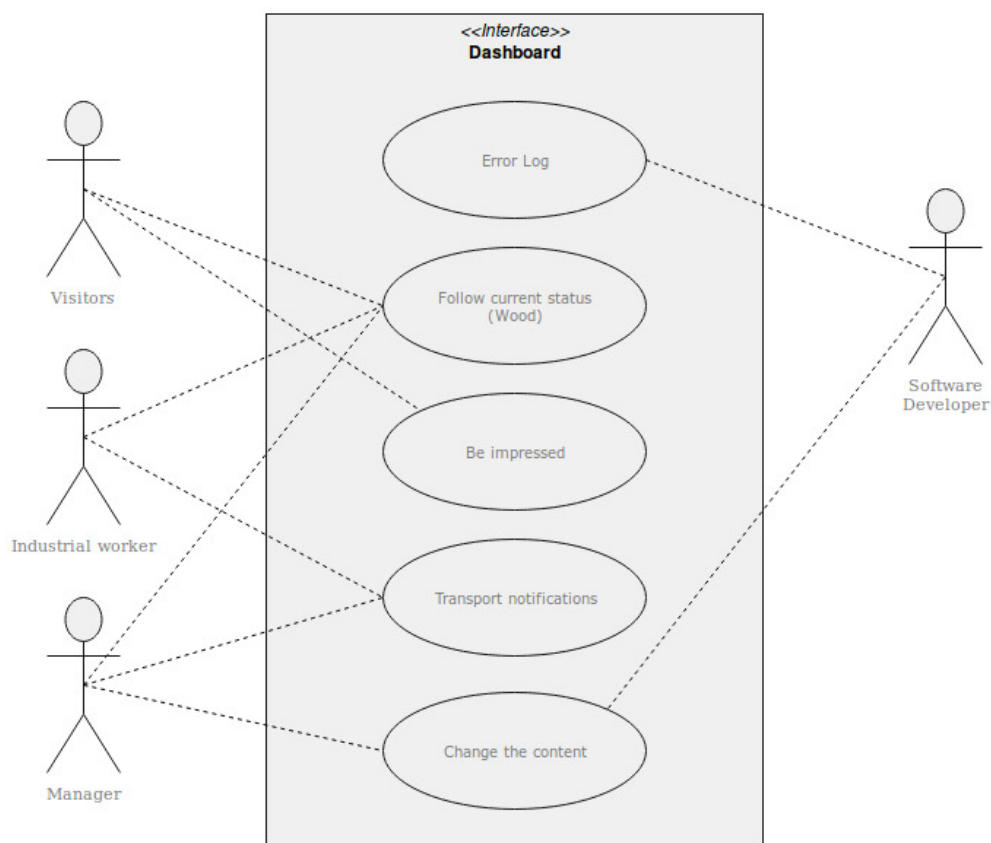
### 1.1.2 GPS Assisted Transport Announcement

Sogeti har utvecklat datorsystemet GATA på uppdrag av SCA. GATA tillhandahåller information om flödet mellan skog och industri. Exempel på data en finner i systemet är volym virke som fraktas, lossas samt finns efter väg och i skog. Genom att i realtid kunna följa genomloppstider har inblandade parter inte bara möjlighet att styra processen utan också dirigera om [4]. Detta medför att en timmerbil på väg till en mottagningsplats med lång väntetid istället kan välja att åka till en med kort väntetid. Den största fördelen med GATA är att alla användare har tillgång till samma information samtidigt [4].

## 1.2 Högnivåproblemformulering

Det här projektets mål är att undersöka möjligheten att vidareutveckla GATA. GATA saknar idag ett verktyg som sammanställer de stora datamängder som samlats in.

Målsättningen var att utveckla ett effektivt system, hjälpmedel, som analyserar, filtrerar samt visar information via en dashboard. Denna dashboard är sedan tänkt att placeras i entrén vid SCA's huvudkontor alternativt ute i industrin eller i ett kontorslandskap, se figur 1. Undersökningen har vidare som mål att verifiera att lösningen uppfyller användbarhetskriterier, samt eventuella krav på vald teknisk plattform.



Use Case Diagram - Functional requirements

Figur 1: Användningsfall, de olika målgrupperna med krav på system.

## 1.3 Avgränsningar

Projektet kommer i design av gränssnitt samt implementation av system inte att ta hänsyn till eventuella hinder med General Data Protection Regulation (GDPR). Etiska aspekter med GDPR diskuteras under slutsatskapitlet.

## 1.4 Konkreta och verifierbara mål

Undersökningen har som mål att besvara följande frågor:

- P1:** Undersöka hur och med vilken teknik systemet bör hämta data.
- P2:** Målgruppsanalys samt analysera intressenternas behov.
- P3:** Undersöka vilken information som ska visas i systemet och hur den presenteras.
- P4:** Utveckla ett användbart system som ger användaren en positiv upplevelse.
- P5:** Utvärdera systemet enligt användbarhetskriterier med fokus på den tid användaren behöver för att läsa information alternativt mata in och redigera. Bidrar systemet med nytta till verksamheten?

## 1.5 Översikt

I kapitel 2 *Bakgrundsmaterial* presenteras teorin som ligger till grund för arbetet. Så som programspråk, ramverk och tidigare arbeten inom området. Vidare under kapitel 3 *Metod* beskrivs val av tillvägagångssätt samt tekniker vid genomförande av användbarhetstest. Lösningalternativ finns i kapitel 4 *Konstruktion* tillsammans med arbetets kravspecifikation. I kapitel 5 *Resultat* presenteras en sammanfattning av användbarhetstesterna och utvärdering av systemet. Resultatet diskuteras sedan i kapitel 6 *Slutsatser*.

## 1.6 Författarens bidrag

I figur 9 presenteras en övergripande systemöversikt. Filerna som är markerade med rött har jag med hjälp av handledare Carl Lundin skapat, med undantaget *dashboardController.js* som Carl skapat.

## 2 Bakgrundsmaterial

I det här kapitlet presenteras rapportens bakgrundsmaterial tillsammans med valda arbetsmetoder och programspråk. Slutligen presenteras tidigare arbeten inom området datavisualisering och BI med fokus på ett företags framgångsfaktorer

### 2.1 .NET-ramverk

.NET [5] är ett ramverk utvecklat av Microsoft och används för att skapa applikationer av olika slag. Förutom programspråket C# som använts i det här projektet går det att skriva kod i Visual Basic eller F# [5]. .NET är ett ramverk som innehåller klassbiblioteket Base Class Library (BCL), exekveringsmiljön Common Language Runtime (CLR) och ett IDE<sup>1</sup> [6].

### 2.2 Backbone.js

Backbone.js är ett JavaScript bibliotek som används för att bygga webbapplikationer skapat av Jeremy Ashkenas [7]. Backbone.js tillhör vad man kallar ett MVC-ramverk, Model View Controller, likt konkurrenten Angular.js. I ett MVC-ramverk använder man sig av en välordnad filstruktur innehållande mappar som views, templates och models. Exempel på tillämpning visas i figur 6.

### 2.3 C#

C#, uttalas C-Sharp, är ett programspråk som en del av Microsofts ramverk .NET. C# är objektorienterat och liknar syntax en finner i Java och C++ [8]. C# stödjer abstrakta datatyper [9].

### 2.4 Nyckeltal

Key performance indicator (KPI) är ett kvantitativt värde som används för att uppvisa hur väl ett företag eller en verksamhet når sina mål [10]. Genom att definiera viktiga nyckeltal blir företagets framgång mätbart.

### 2.5 LINQ

Language-Integrated Query (LINQ) är en gren under Microsofts ramverk .NET framtaget för att integrera query-funktioner i C# [12]. En av fördelarna med LINQ är att man som programmerare inte behöver använda sig av olika språk när man hämtar data från olika typer av källor [12]. Vilket medför att till exempel XML- och SQL-formaterad data fångas med samma språk. Språket bygger på så kallade query-operationer [12].

<sup>1</sup> Integrated Development Environment (IDE)

Utvecklingsmiljö som används för att producera körbara datorprogram. Består ofta av en kompilator, debugger (sv. felsökningsprogram) samt editor (sv. ordbehandlare). [11]

Dessa operationer gör det möjligt att med färre rader kod analysera och sortera data. Se exempel på kod nedan:

```
void Main() {  
  
    FordonTrpEntities m_context = new FordonTrpEntities();  
  
    var Transportledaren = (from Transportledare in m_context.Transportledare select  
        Transportledare.Id).ToList();  
  
    Transportledare.Dump();  
  
}
```

Det här kodavsnittet hämtar information om alla transportledare.

## 2.6 Microsoft Azure Application Insights

Azure är en samling molntjänster utvecklad av Microsoft som används för att skapa samt underhålla främst webbapplikationer [13]. Application Insights som en del av Azure är en APM-tjänst, Application Performance Management, som hjälper utvecklare att samla in och övervaka data. Application Insights [14] bevakar bland annat:

- Begärandefrekvens, svarstider och felfrekvens
- Prestandaräknare
- Sidvyer och inläsningsprestanda

Utöver ovan nämnda har en möjlighet att skapa egna förfrågningar med hjälp av ett RESTful API [14]. På så sätt är det möjligt att skräddarsy vad i applikationen som mäts och sedan visas på dashboarden. Användaren når dashboarden genom Azure portalen [14].

## 2.7 Scrum

Inom agilsystemutveckling finns olika arbetsmetoder, scrum är idag den mest vanliga. Att arbeta iterativt samt inkrementellt är hjärtat i scrum. Att arbeta inkrementellt innebär att man fokuserar på de mest betydelsefulla och grundläggande funktionerna först [15]. För att sedan arbeta sig vidare. Iterativ-utveckling innebär istället att en arbetar mot mindre releaser. Första målet är då att få prototypen att fungera, nästa steg blir att förbättra den befintliga versionen.

I scrum arbetar en med sprintar. En sprint inleds och avslutas alltid med ett möte. Ett startmöte kallat sprint planning och ett avslutade möte kallat sprint review [16]. Under en sprint review presenteras och visas arbetet upp tillsammans med eventuell prototyp [16].

## 2.8 Datavisualisering

Datavisualisering, informationsvisualisering, är en metod som används för att åskådliggöra stora mängder data i bild på ett sätt som är enklare för människan att förstå samt tolka [17]. Beroende på hur data ska presenteras kan det vara lämpligt att innan göra en interpolation [17]. Interpolation innebär att man anpassar data till en matematisk funktion, vilket resulterar i en typ av kurvanpassning.

### 2.8.1 Närhetens lag

Närhetens lag används, enligt [18], för att göra användaren uppmärksam på vilket material som hör ihop och hur materialet ska läsas. Detta genom enhetlighet, placering och vit yta mellan objekt [18]. Figur 2 visar på hur objekt placeras enligt denna princip.



*Figur 2: Materialet i box 1 samt box 2 hör ihop och ska läsas från vänster.*

### 2.8.2 Statiska och dynamiska modeller

En statiska modell ger användaren en bild av organisationens nuvarande status. En dynamisk modell utgår istället från organisationens tidigare resultat och beräknar utifrån dessa kommande [19]. Genom att använda sig av dynamiska modeller upptäcks eventuella störningar tidigt.

## 2.9 Dataöverföring

Vid överföring av data mellan klient och server tillämpas olika metoder och tekniker. Nedan presenteras tre av de tekniker som ofta används.

### 2.9.1 AJAX

Asynchronous JavaScript and XML (AJAX) är en teknik som används för att uppdatera delar av en interaktiv webbapplikation [20]. Det medför att applikationen kan köras utan att behöva ladda om hela webbsidan. För att hämta data används oftast dataformatet XML<sup>2</sup> eller JSON<sup>3</sup>. AJAX använder sig av ett API som heter XMLHttpRequest för att transportera XML- alternativt JSON formaterad kod från servern till klienten [20].

### 2.9.2 RESTful webbtjänst

Representational State Transfer (REST) är en annan teknik som används vid kommunikation mellan server och klient. POST, GET, PUT och DELETE är HTTP-variabler som skickas vid förfrågan och anger vilken typ av operation som ska utföras [23]. REST använder sig precis som AJAX oftast av dataformatet XML eller JSON.

### 2.9.3 WebSockets

WebSocket är relativt ny teknik, som en del av HTML5, vilket tillåter webbapplikationer att använda sig av full-duplex<sup>4</sup> kommunikation [24]. WebSockets använder sig av transportprotokollet Transmission Control Protocol (TCP) vid kommunikation mellan klient och server. Fördelen med WebSocket är enligt [25] att data skickas säkert via en öppen tunnel direkt från servern till klienten. Servern är då inte beroende av polling längre [25].

## 2.10 Cachelagring

Cachelagring är en teknik som används för att snabba på ett systems svarstid genom att hålla den data som gång på gång används nära användaren [26][27]. På så sätt undviker man att kostsamma beräkningar av data körs i onödan. I distribuerade system kan cachning [26] ske både hos klienten och servern. Det hanteras antingen lokalt på användarens dator alternativt på en delad cache som då kan nyttjas av flera processer [26].

<sup>2</sup> eXtensible Markup Language (XML)

XML är ett märkesspråk som används för att skicka och lagra data. [21]

<sup>3</sup> JavaScript Object Notation (JSON)

JSON är ett dataformat som används för att skicka och lagra data. [22]

<sup>4</sup> Duplex

Kommunikation i två riktningar. Med full duplex menas att kommunikation i två riktningar kan ske samtidigt. [29]

## 2.11 Tidigare arbeten

Här presenteras tidigare arbeten och forskning inom området datavisualisering och BI med fokus på ett företags framgångsfaktorer.

### 2.11.1 Stephen Few

Forskaren och författaren Stephen Few har publicerat en mängd artiklar inom ämnet datavisualisering och konsten att kommunicera kvantitativ data. Few menar i [28] på att man för att lyckas med virtuell kommunikation bör använda en modell som bygger på sju riktlinjer, se tabell 1.

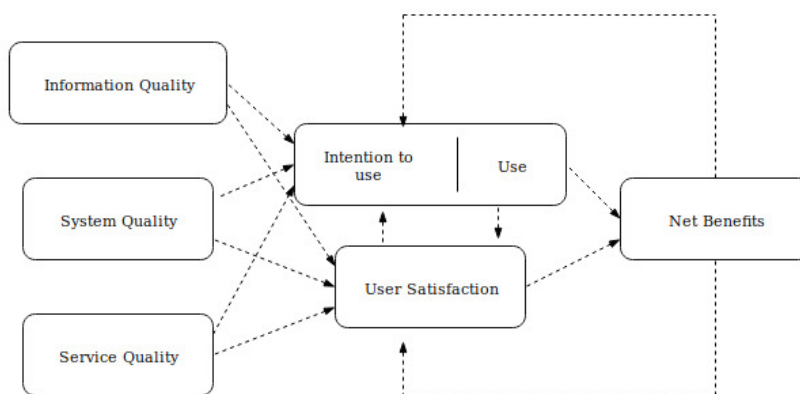
Few's riktlinjer i tabell 1 kommer inte översättas från engelska till svenska då betydelsen inte blir densamma.

<b>Riktlinjer enligt Stephen Few</b>	
<b>Informative</b>	
Usefulness	Presentera användbar information. Information som saknar värde är meningslös och ineffektiv.
Completeness	Vikten av att presentera rätt mängd information för att användaren ska förstå.
Perceptibility	Förstå hur data uppfattas och effektivt kunna presentera information så att användaren utan ansträngning begriper och kan avläsa resultatet.
Truthfulness	Säkerställa att informationen som presenteras är korrekt.
Intuitiveness	Kommunicera information med former som för användaren är bekanta.
<b>Emotive</b>	
Aesthetics	Estetik är viktigt vid datavisualisering. Ett tilltalade gränssnitt fångar fler användares intresse jämfört ett fult.
Engagement	Skapa ett engagemang hos användare som uppmuntrar till att undersöka innehållet.

Tabell 1: Few's sju riktlinjer [28] sammanställt i tabell.

### 2.11.2 Updated DeLone and McLean IS Success Model

Professorerna DeLone och McLean tog år 1992 fram ett ramverk som mäter framgång hos en verksamhet [30]. Den uppdaterade framgångsmodellen bygger på sex stycken riktlinjer, se tabell 2, och är tänkt att kunna appliceras även elektroniskt [30].



Figur 3: Framgångs modellen enligt [30].

<b>Riktlinjer enligt William DeLone och Ephraim McLean</b>	
Systemkvalitet	Här mäts till exempel ett systems tillförlitlighet, flexibilitet samt svarstider.
Informationskvalitet	Säkerställa att relevant information visas.
Servicekvalitet	Garantera kvalitet på systemstöd för användaren.
Användning	Systemets uppbyggnad, navigering och hur data hämtas.
Tillfredsställelse hos användaren	Ett mått på hur nöjd användaren är med upplevelsen.
Net benefits	Net benefits anses vara den mest betydelsefulla faktorn, och fungerar som ett mått på jämvikten mellan de positiva och negativa resultaten.

Tabell 2: Riktlinjer enligt [30], William DeLone och Ephraim McLean.

### 2.11.3 Evgeniy Gorodov och Vasiliy Gubarev

Gorodov och Gubarev beskriver i [31] svårigheterna med att fånga upp och visuellt presentera Big Data. Problemet uppstår när den volym information som ska bearbetas är för stor eller helt saknar struktur.

Ett av problemen är visuellt brus [31], vilket kan uppstå om en försöker presentera en array full med information via en enda graf. Risken är att data blir för koncentrerat och bara uppfattas som en stor klump i grafen. En lösning på problemet skulle kunna vara att använda sig av en större skärm. Men trots detta är mängden data som presenteras fortfarande för omfattande, människan har kognitiva begränsningar som gör det omöjligt att ta in och förstå detta.

Utifrån problemet ovan drar Gorodov och Gubarev slutsatsen att även kognitiva begränsningar i form av fysiska perceptionsgränser [31] bör ses över under arbetet med datavisualisering.

Utöver detta vill författarna i [31] även belysa två andra vanliga problem. Det första är informationsförlust vilket innebär att viktig data kan gå förlorad om en filtrerar resultatet på fel sätt. Det andra problemet är high performance requirements, här menar dom på att det ökade kravet på filtrering ihop med de dynamiska modellerna kräver mer datorresurser, vilket resulterar i höga prestandakostnader.

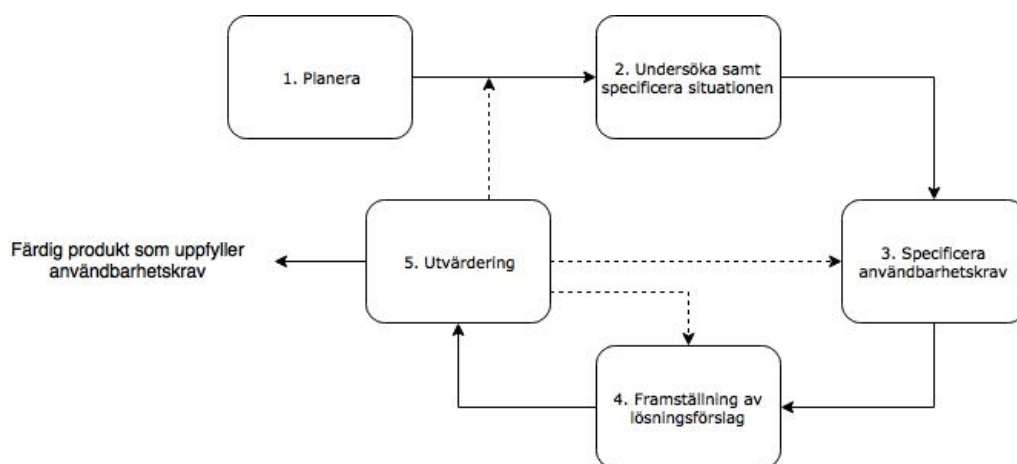
## 3 Metod

Under arbetet har det agila arbetssättet scrum använts med en sprintlängd på tre veckor. Vilket alltid avslutats med en sprint demo. Under ett sprintmöte har nuvarande status, eventuella problem samt projektets plan framåt redovisats.

Programvara som använts är Visual Studio 2017, LINQPad 5, Adobe Illustrator, Adobe Photoshop, Adobe XD, Visual Studio Code samt Tobii Studio. Diagram är framtagna i webbapplikationen Draw.io. Förutom nämnd programvara har en dator av märket Lenova Notepad med operativsystemet Windows 10 använts. Sogeti har beviljat tillgång till SCA's Virtual Private Network (VPN) samt en fjärrstyrd dator med operativsystemet Windows 7.

### 3.1 Designprocess

Parallellt med utvecklingen av systemet har en designprocess genomförts. Designprocessen har delats upp i tre etapper, vilka varit, konceptfasen, bearbetningsfasen samt detaljeringsfasen enligt [32]. Under konceptfasen, konceptualiseringen, sammanställs mål och krav i enkla skisser [32]. Enkla pappersskisser har sedan överförts till det vektorbaserade illustrationsprogrammet Adobe Illustrator för att realiseras.



Figur 4: Designprocess uppdelad i fem steg enligt [32].

Att skapa ett system för människor kräver kunskap om användaren och den tänkta målgruppen. Projektets målgruppsanalys illustreras med ett Use Case Diagram, se figur 1.

I den senare delen, bearbetningsfasen, utvecklas den första prototypen. En prototyp enbart till för test. Under detaljeringsfasen testas sedan den färdiga prototypen av användaren och utvärderas av uppdragsgivare [32]. Att arbeta iterativt, se figur 4, medför att slutresultatet blir bättre och mer genomtänkt.

### 3.2 Bedömningskriterier

Enligt delmål **P3** har undersökningen som mål att “utvärdera det färdiga programmet enligt användbarhetskriterier”. De designprinciper som avses granskas anges i tabell 3. Utöver den heuristiska utvärderingen har ett eyetracking-test genomförts.

Designprinciper	
Synlighet	Funktioner ska vara synliga för användaren.
Feedback (sv. Återkoppling)	Syftar till att ge användaren information om hennes handling. Exempel genom popup-fönster eller ljud.
Begränsningar	Förtydliga, alternativt eliminera låst funktionalitet.
Consistency (sv. Konsekvent )	Systemets funktioner ska vara konsekventa i sin betydelse.
Affordance	Hur lätt det är att förstå och lära sig systemet.

Tabell 3: Designprinciper. [33]

### 3.3 Eyetracking

Eyetracking (sv. ögonspårning) är en metod och system som avläser vart på en datorskärm användarens blick befinner sig samt när [34]. Detta ger bättre förståelse för hur väl ett användargränssnitt kommunicerar med användaren [34]. I det här projekt har programvara och utrustning från tillverkaren Tobii använts.

En eyetracker från Tobii är enligt [34] en sensor uppbyggd på ett antal belysare och kameror. Kameror fotograferar användarens ögon för att sedan matcha dessa mot algoritmer. Dessa algoritmer beräknar sedan blickens exakta läge samt vart på skärmen användarens fokus ligger. Resultatet sammanställs både grafiskt och numeriskt.

Under ett användbarhetstest i Tobii får testgruppen bekanta sig med gränssnittet samtidigt som de besvarar frågor rörande vart på skärmen en specifik information finns.

## 4 Konstruktion

I det här kapitlet presenteras arbetets kravspecifikation samt systemuppbyggnad. Under avsnitt 4.2 *Systemmodell* presenteras valt tillvägagångssätt under implementationen.

### 4.1 Kravspecifikation

#### 4.1.1 Förutsättningar

För att vidareutveckla det befintliga systemet har en lista på förutsättningar sammanställts, se nedan.

- Utveckling av system ska ske på Microsoftplattformen .NET med programvaran Visual Studio 2017 i språket C#.
- All programkod med tillhörande specifikationer och andra utvecklingsdokument ska versionshanteras med hjälp av Microsoft TFS.

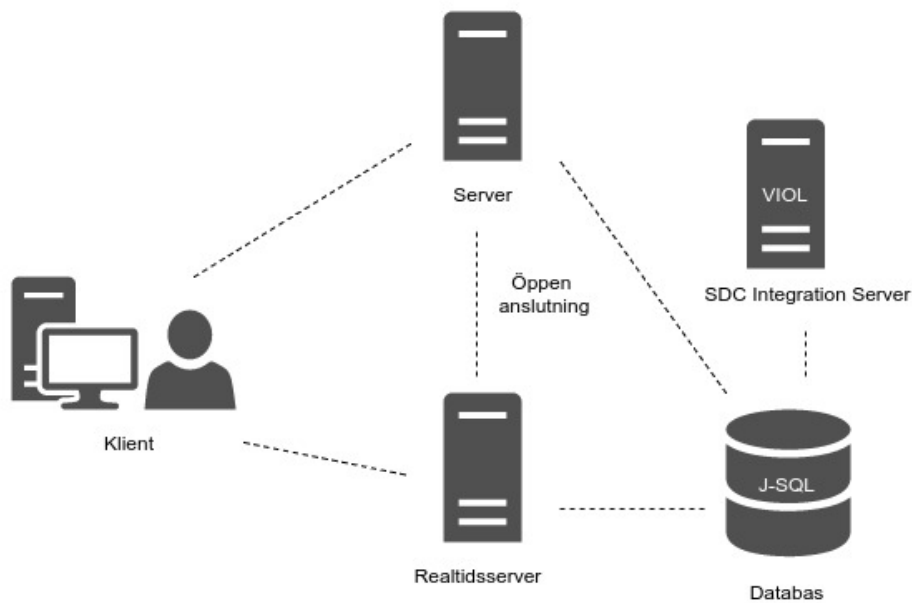
#### 4.1.2 Information som ska visas i systemet

Ett av projektets delmål, **P3**, var att undersöka vilken information som ska visas i systemet och hur den sedan ska presenteras. Efter samtal med representant från målgruppen har följande alternativ till krav fastställts:

- Volym virke som lossas
- Volym virke som fraktas
- Volym virke som finns i skog
- Volym virke hos lagerplats
- Antal avslut idag
- Aktiva åkerier
- Trucklossade / Självlossade
- Snittid vid mottagningsplats
- Antal fordon i arbete
- Antal skickade aviseringar idag
- Aktiva mottagningsplatser
- Systemets status
- Felmeddelanden

## 4.2 Systemmodell

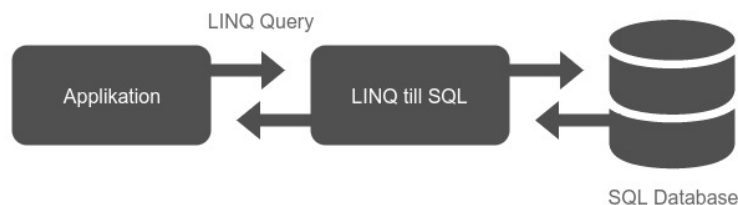
Det här systemet är en tillbyggnad på SCA's redan befintliga system GATA. Det här systemet innefattas av en klient, en realtidsserver och en databas, se figur 5. Klienten är den del av systemet som kommunicerar med användaren. Systemet hämtar och analyserar information från GATA's befintliga databas, hämtning sker genom en realtidsserver.



Figur 5: Grafisk representation av implementeringsschema.

### 4.2.1 Dataöverföring

Klienten hämtar data via en realtidsserver. Detta medför att responstiden kortas ned jämfört att hämta data via den vanliga servern. Frågorna som skickas är utformade i programspråket LINQ och skickas från klienten via en WebSocket.



Figur 6: Frågorna är utformade i LINQ och omvandlas till SQL.

För att hämta data har två stycken filer placerats i *SCA.Transport.Admin.Business*, filerna heter *Dashboard.cs* och *DashboardDTO*. *DashboardDTO* innehåller en mall för de objekt som ska hämtas. I *Dashboard.cs* hämtas sedan objekten samtidigt som den hämtade informationen tilldelas enhet, beskrivning samt ett unikt id-nummer. Det gör det möjligt att styra vart på dashborden den erhållna informationen ska placeras. Då antalet objekt på dashborden i dagsläget är relativt få och ska placeras förutbestämt är denna lösning bra. Om objekten hade haft automatiskt genererade identiteter skulle innehållet placerats ut slumpmässigt.

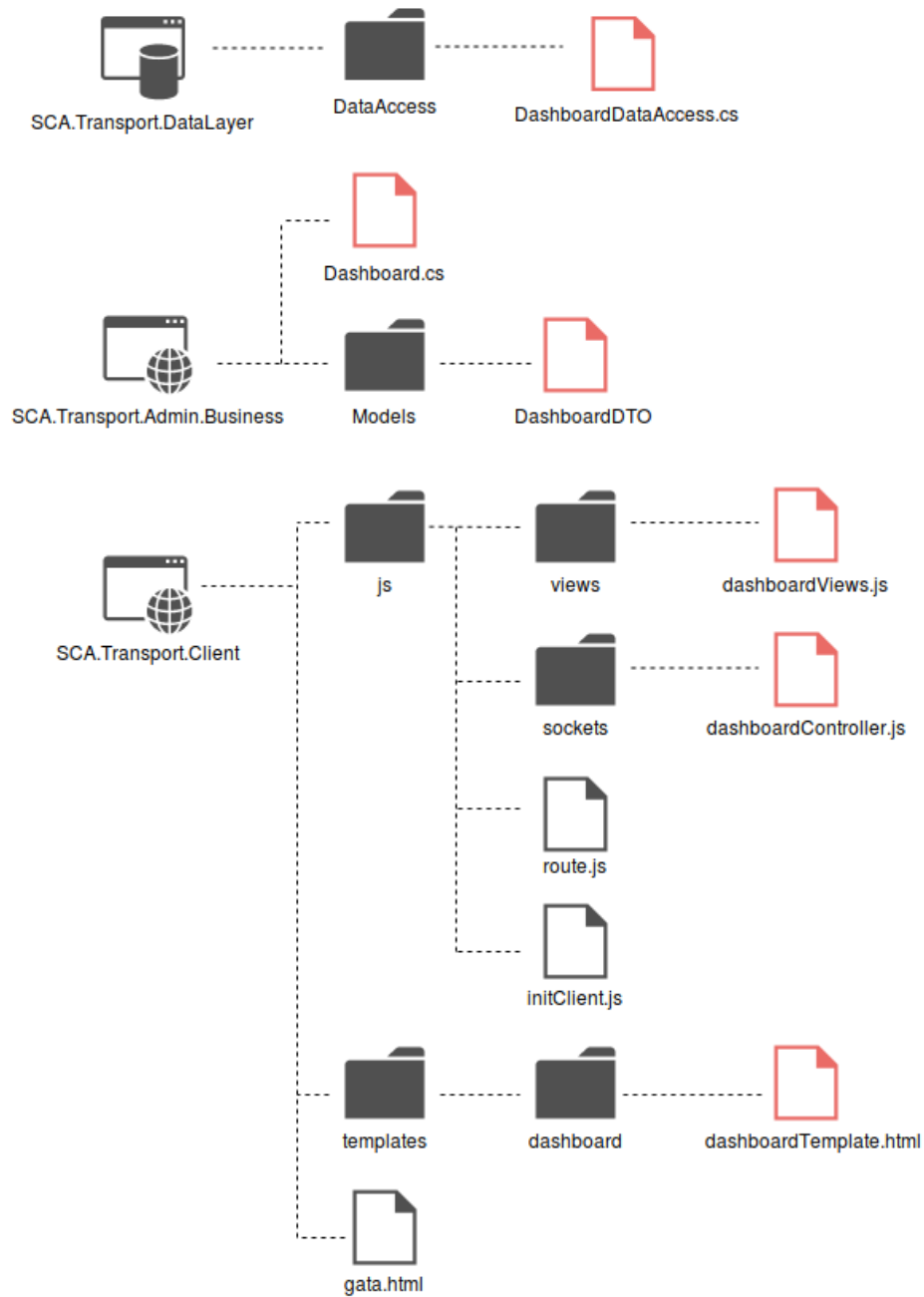
```
data.Add(new DashboardDTO
{
    Id = 7,
    Text = "Antal felmeddelanden",
    Value = DashboardDataAccess.FelMeddelanden(),
    Unit = "st"
});
```

Figur 7: Kodexempel från filen Dashboard.cs.

```
}
public string FelMeddelanden()
{
    var dt = DateTime.Today.AddDays(1);
    return Context.ErrorLog.Where(x => x.SenastAndrad > dt && x.Text != "INFO")
        .Select(x => x.MachineName).Count().ToString();
}
}
```

Figur 8: LINQ-fråga från DashboardDataAccess.cs som hämtar information om antalet felmeddelanden under dagen.

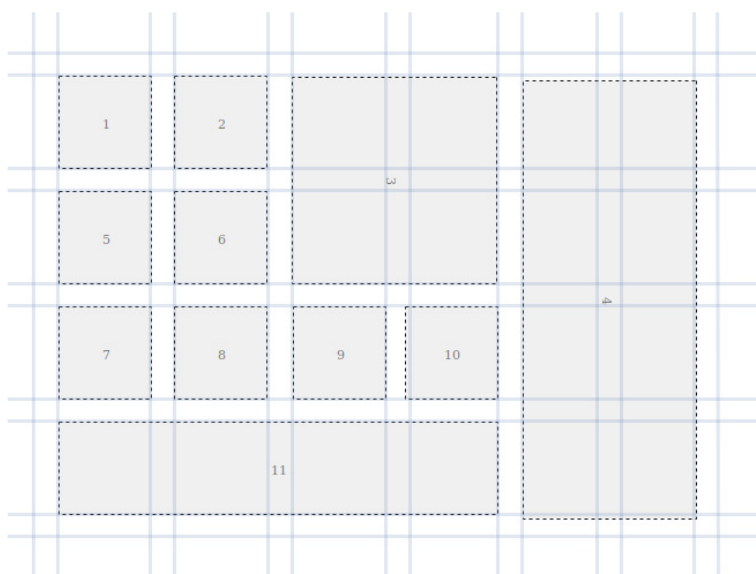
Utöver dessa två filer har en tredje fil skapats i *SCA.Transport.DataLayer* med namnet *DashboardDataAccess.cs*. I den filen ligger LINQ-frågorna, se figur 8 för ett exempel på en fråga.



Figur 9: Dashboard-systemets filstruktur. Alla filer som visas används i projektet. Filerna som är rödmarkerade skapades under projektet. Resterande filer är befintliga och har enbart ändrats.

## 4.2.2 Klienten

Klienten är den del av systemet där data presenteras för användaren. Klienten är uppbyggd kring ett antal delsystem, se figur 9 för fullständig filstruktur.



Figur 10: Dashboardens objekt uppdelat i rutor enligt CSS Grids.

I den redan befintliga filen *gata.html* deklareraras dashboardkoden. Det samma gäller den befintliga filen *route.js* som används för att dirigera om användaren till *dashboardTemplate.html*. I *initClient.js* sker initiering. *DashboardViews.js* innehåller klientens funktionalitet, bland annat funktionen *addOne*, se figur 11.

Språk som används i klienten är mestadels JavaScript och C#. Ramverket Backbone.js grupperar koden i *models*, *templates*, *views* och *sockets*.

```
addOne: function (model) {  
  
    var value = model.get("Value");  
    var text = model.get("Text");  
    var id = model.get("Id");  
    var unit = model.get("Unit");  
  
    spanValue = $("#dashboardBox"+id).html();  
    spanValue = spanValue + text + "<br />" + value + " " + unit;  
    $("#dashboardBox" + id).html(spanValue);  
  
},
```

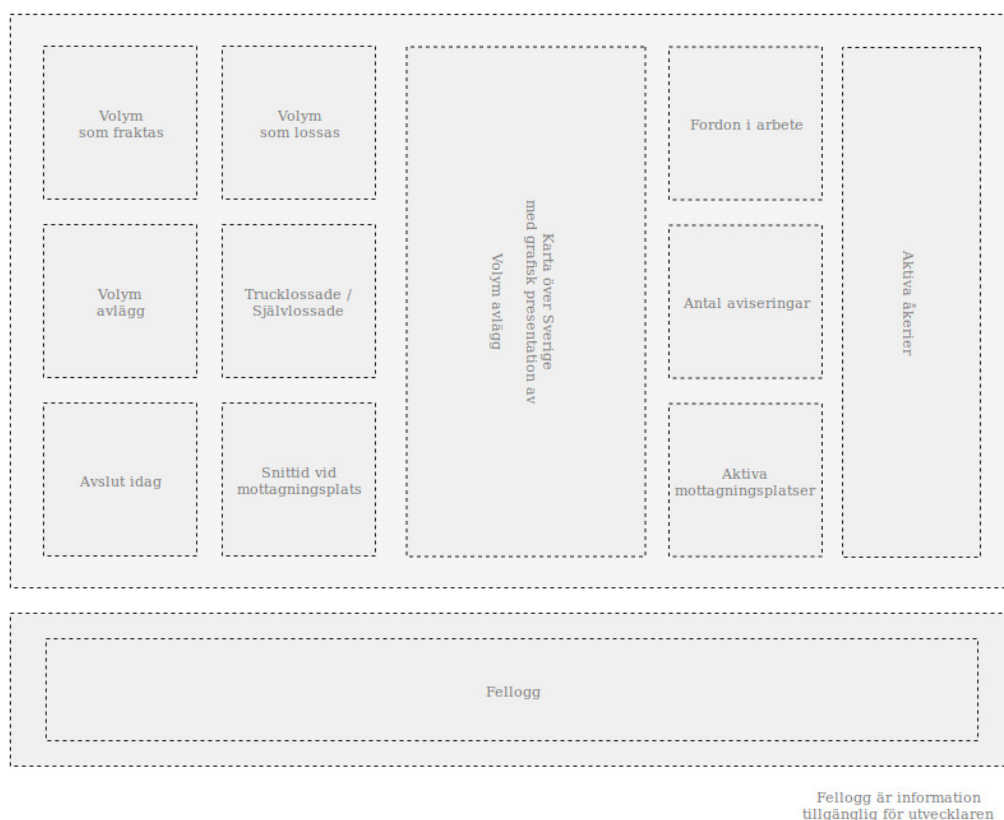
Figur 11: Funktionen *AddOne* lägger till elementen på dashboarden.

Användargränssnittets design är uppbyggt i en fil som går under namnet *dashboardTemplate.html* och består av kod skriven i HTML och CSS. CSS Grids gör det möjligt att dela in gränssnittet i ett rutnät, med ett förutbestämt antal kolum-

ner och rader. Detta medför även att objekten kan delas in i rutor, se figur 10. Det går manuellt att ange hur många rutor i x- samt y-led ett specifikt objekt ska bestå av. CSS Grids använder sig av två dimensioner jämfört Flexbox som bara har en dimension.

### 4.2.3 Användargränssnitt

Tidigt under projektet identifierades fyra olika målgrupper, se figur 1. Fokus under implementeringen har varit på en målgrupp bestående av utvecklare. Målgruppen hade önskemål om att kunna följa systemets status i realtid. Avsnitt 4.1.2 *Information som ska förmedlas via dashboarden*, innehåller uppgifter om vad som ska visas på dashboarden. Figur 12 visar en tidig gränssnittsskiss.



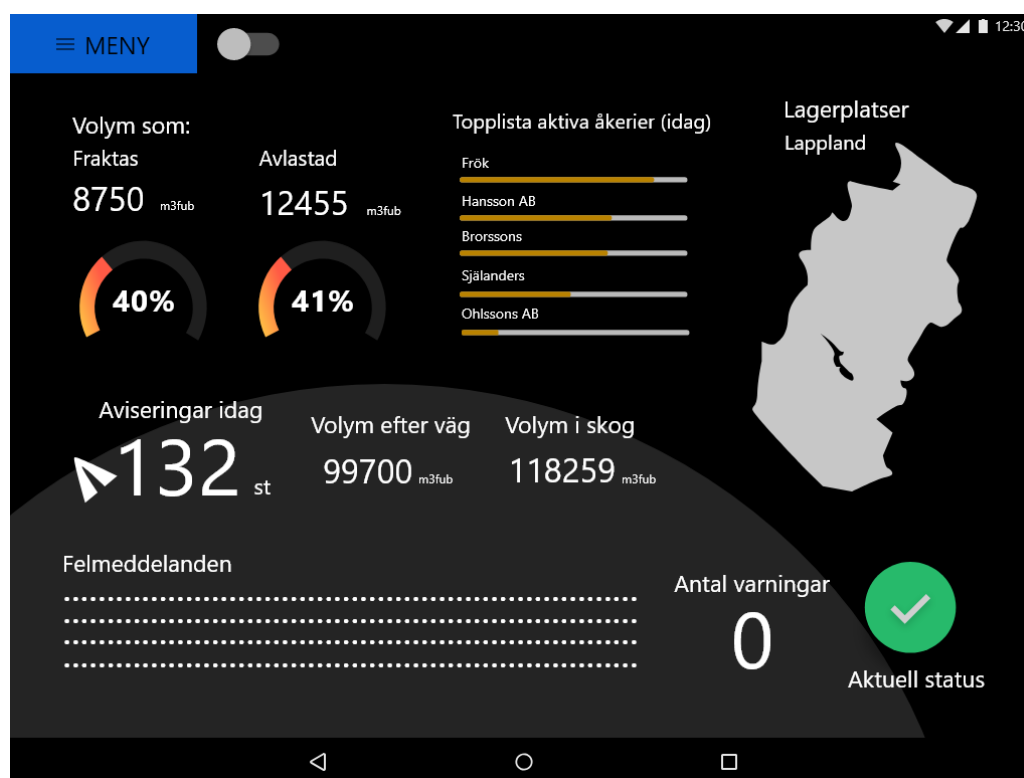
Figur 12: Tidig skiss på GUI.

I arbetet med att designa ett användargränssnitt användes Few, Delone och McLean's riktlinjer. I figur 13 visas det slutliga användargränssnitt. Som en kan se har information som hör ihop placerats tillsammans enligt närhetens lag.

Few beskriver i [28] vikten av att skapa ett tilltalande användargränssnitt. Genom att använda sig av datavisualisering bör materialet bli lätt att förstå och förhoppningsvis även intressera användaren. Exempel på en visualisering är barometern i figur 13 som visar den volym som fraktas. Barometern är en dynamisk modell som sammanställer och speglar tidigare resultat. Det ger användaren en indikation på hur arbetet längs med vägen går samt uppmärksammar avvikelser.

Felmeddelanden, antal varningar och aktuell status är objekt som enbart intresserar utvecklaren och bör plockas bort i ett mer säljande syfte exempelvis mot besökare på huvudkontoret.

Aktuell status är för vald målgrupp den mest relevanta informationen på dashboarden. En stor lampa lyser grönt alternativt rött beroende på hur systemet GATA mår. Under felmeddelanden ska en sedan kunna läsa vart felen påträffats och antalet varningar visas bredvid.



Figur 13: Slutligt GUI skapat i Adobe XD med hjälp av Marcelo Silva's UI kit.

När en timmerbil lämnar en avverkningsplats skickar hen en avisering till mottagande lagerplats. Dessa dagliga aviseringar sammanställs och presenteras numeriskt med en ikon föreställande ett pappersflygplan. Ikonens placering är tänkt att öka läsbarheten. Samt uppmärksamma användaren på skillnader mellan de olika värden som presteras.

Lagerplatser redovisas grafiskt via en karta uppe i höger hörn. Aktiva åkerier har sammanställts som en topplista där de åkerier som lastat av störst volym ligger placerad i topp.

## 5 Resultat

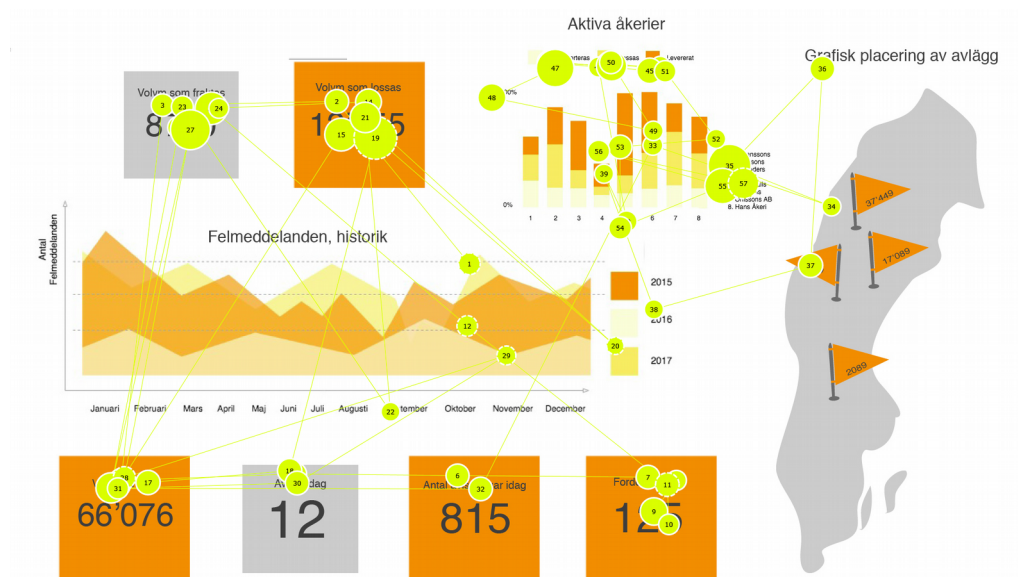
I det här kapitlet redovisas resultatet från användbarhetstester samt en utvärdering enligt användbarhetskriterier, se tabell 3.

### 5.1 Empirisk utvärdering

I det här avsnittet presenteras resultatet från användbarhetstest 1 och 2 vilka båda genomfördes med hjälp av eyetracking utrustning hos Sogeti. Resultatet är sammanställt i en gaze plot, en cluster plot samt ett enkelt diagram innehållande reaktionstider.

#### 5.1.1 Användbarhetstest 1

Användbarhetstest 1 visade att flera av testpersonerna läste dashboarden som en bok. Det vill säga började längst upp och arbetade sig nedåt. Den tid det tog för användaren att hitta den sökta informationen redovisas i figur 16. Medelvärdet för alla tre testpersoner var 1.08 sekunder. Figur 14 och 15 visar resultatet från eyetrackingtestet sammanställt som en gaze plot och en cluster plot. De gula prickarna i figur 14 visar vart på dashboarden testpersonens blick var fokuserade samt hur länge.

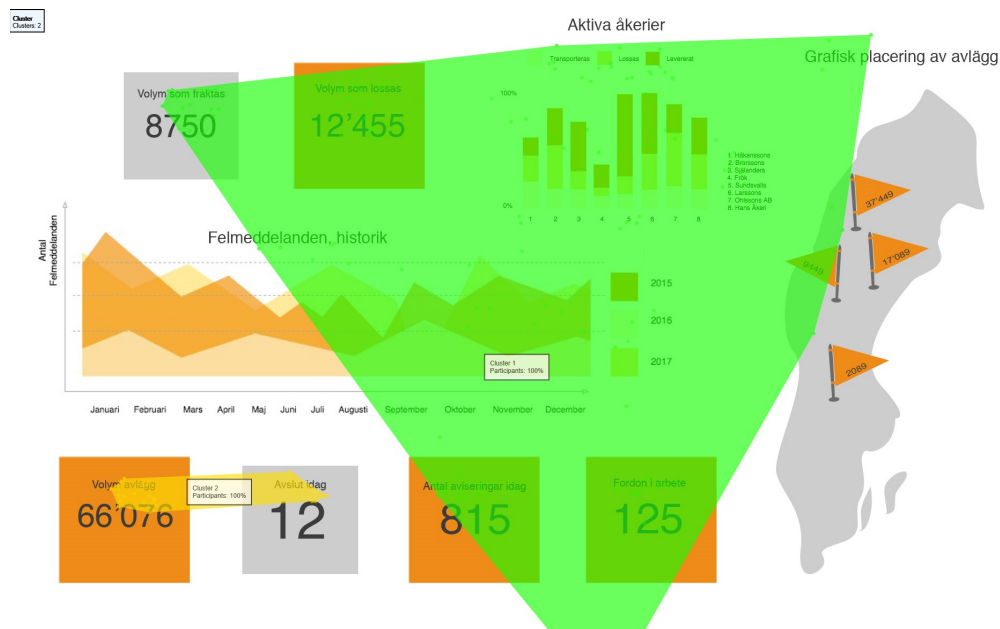


Figur 14: Eyetrackingtest sammanställt i Gaze plot.

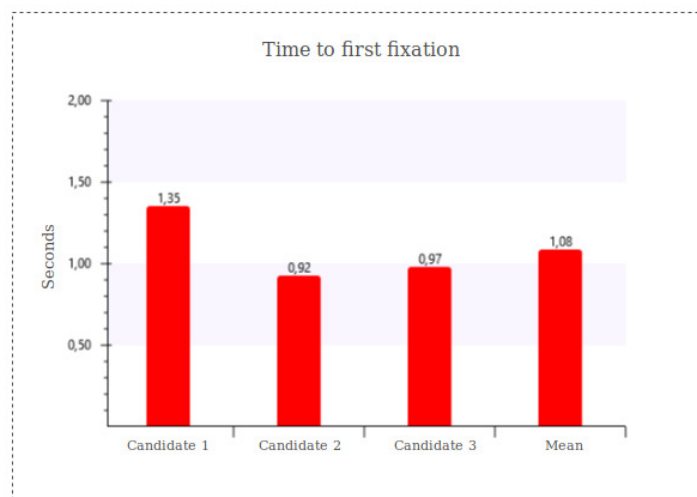
Figur 15 visar istället de område på dashboarden där störst fokus funnits. Utifrån den informationen kan man dra slutsatsen att Sverigekartan till höger var onödig.

Testet visade att detta kan förbättras:

- Lägga till enheter på de mätvärden som presenteras.
- Förenkla samt öka textstorlek på de stapeldiagram som visar aktiva åkerier.
- Placering av relevant information.



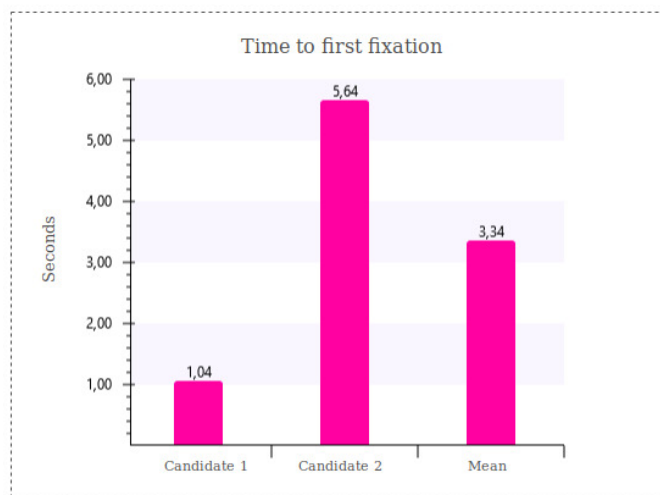
Figur 15: Eyetrackingtest sammanställt i Cluster plot.



Figur 16: Tid fram till det att testpersonen hittade den sökta informationen. Tid redovisad i sekunder.

### 5.1.2 Användbarhetstest 2

Användbarhetstest visade att tiden det tog för en av testpersonerna att hitta information var betydligt högre än tidigare, detta medförde en högre snittid, se figur 17. Snitttiden för användbarhetstest 2 var 3,34 sekunder. Testpersonerna kände trots den högre snittiden en positiv känsla av att använda systemet. För gaze plot och cluster plot se bilaga A.



Figur 17: Tid fram till det att testpersonen hittade den sökta informationen. Tid redovisad i sekunder.

Testet visade att detta kan förbättras:

- Tydligare rubriknamn.
- Placering av objekt innehållande information om volym som fraktas och avlastad volym.

## 5.2 Teoretisk utvärdering

I enlighet med mål **P5** har systemet utvärderats med fokus på den tid användaren behöver för att läsa information. Resultatet från det senare användbarhetstestet visade en snittid på 3,34 sekunder för att hitta samt förstå sökt information.

I tabell 4 redovisas resultatet från den heuristiska utvärderingen enligt designprinciper. Utvärderingen visade att systemet inte ger användaren någon feedback. Det finns inte några begränsningar att utvärdera i form av dold eller låst funktionalitet.

Bortsett från dessa två punkter visar utvärderingen på att systemet följer de övriga designprinciperna. Systemet är tydligt, konsekvent och lätt att använda. Det visar att systemet uppnår mål **P5**.

<b>Designprinciper</b>				
<b>Synlighet</b>	<b>Feedback (sv. Återkoppling)</b>	<b>Begränsningar</b>	<b>Consistency (sv. Konsekvent )</b>	<b>Affordance</b>
✓	×	×	✓	✓

Tabell 4: Heuristiska utvärdering enligt designprinciper sammanställt i diagram.

### 5.3 Sammanfattning

Projektets mål var att utveckla ett användbart system som ger användaren en positiv upplevelse. För att lyckas med det har resultatet från ett flertal undersökningar sammanställts. Det har undersöka hur och med vilken teknik systemet bör hämta data och vilken information som ska visas. Målgruppsanalysen har använts för att sätta ihop kravspecifikationen.

Enligt [30] DeLone och McLean's riktlinjer kan man använda balansen mellan ett systems positiva och negativa resultat som ett mått på framgång.

## 6 Slutsatser

I det här kapitlet diskuteras projektets resultat subjektivt av mig som författare. Vidare diskuteras arbetets samhällseliga och etiska aspekter. Slutligen presenteras förslag på fortsatt arbete.

### 6.1 Slutsatser

Målsättningen med detta projekt var att utveckla ett effektivt system som analyserar, filtrerar samt visar information via en dashboard. Projektet hade vidare som mål att verifiera att lösningen uppfyller användbarhetskriterier, samt eventuella krav på vald teknisk plattform.

På marknaden finns idag ett flertal färdiga APM-tjänster som sammanställer data i dashboards, ett exempel är Microsofts Application Insights. Det hade varit enklare att använda en färdig APM-tjänst men fördelen med att utveckla ett eget system är att det passar bättre ihop med Sogetis egna system. Därför bestämde jag mig tidigt för att fokusera på att utveckla ett helt eget system.

Systemet fungerar som jag planerat men det var tidskrävande att sätta sig in i Sogetis befintliga systemstruktur med tillhörande programmeringsspråk, kod och ramverk. Det medförde att jag fick mindre tid till det estetiska arbetet. Den gränssnittsdesign som presenteras under avsnitt 4 *Konstruktion* får därför representera det slutliga användargränssnittet även om det idag inte syns i programmet.

Sogeti använder sig av scrum som projektverktyg. Det innebär ett iterativt arbete där man arbetar mot delmål. Det agila arbetssättet scrum har varit ett arbetssätt som passat mig bra. Utöver scrum har jag använt mig av Arvola's designprocess för att färdigställa gränssnittet.

Utifrån artiklar och böcker jag läst har jag valt att arbeta enligt Few, DeLone och McLeans teorier. Stephen Few [1][28] fokuserar mer på det visuella jämfört DeLone och McLean [30] som istället betonar vikten av tekniken bakom. Jag har plockat de delar jag ansett bäst hos dem båda.

Med det här systemet tar det för en normal datoranvändare i snitt 3,4 sekunder att hitta den eftersökta information. Tidigare så krävdes det programmeringskunskaper och betydligt längre tid för att få fram motsvarande information. Jag drar slutsatsen att systemet bidrar med nytta till verksamheten då tiden för en sökning väsentligt kortats ned samt att det är mera användarvänligt.

## 6.2 Samhälleliga och etiska aspekter

Projektets mål har varit att utveckla ett system som analyserar stora mängder data. Detta för att effektivisera transportflödet. Både industriarbetarna och åkarna har idag en viktig roll i transportkedjan. Att synliggöra och spara data rörande medarbetarnas prestationer kan leda till att den enskilde individen känner sig övervakad. Det skrivs idag mycket om vikten i att värna om den anställdes integritet. Därför bör man i framtiden ha den enskilde individen i åtanke om man väljer att vidareutveckla systemet. Systemet sparar inte data om var en särskild medarbetare befunnit sig eller befinner sig. Detta är information som skulle kunna skada eller i värsta fall kränka en medarbetare. Systemet sparar heller inte information om skickade eller mottagna aviseringar.

Jag tror personligen inte på en arbetsplats där de anställda pressas till att prestera, då detta kan leda till stress eller i värsta fall utbrändhet. Jag vill istället uppmuntra genom att belysa organisationens resultat.

## 6.3 Framtida arbete

Jag ser gärna att Sogeti färdigställer systemet. Om jag själv hade fortsatt arbetet skulle jag ha undersökt förbättringsförslag enligt listan nedan.

- Undersöka möjligheten till cachelagring för att snabba på svarstider.
- Dashboarden saknar idag data om genomloppstider vid mottagningsplats. Är detta av intresse?
- Resultatet från användbarhetstesten är bara en indikation. Vid vidareutveckling samt innan driftsättning bör fler tester genomföras med en testgrupp som bättre motsvarar målgruppen.
- Idag hämtas data via WebSockets. Ur ett säkerhetsperspektiv kan det i ett framtida arbete vara av intresse att undersöka exakt hur säkert protokollet är.

## Källförteckning

- [1] S. Few, *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring* (2. Edition). Burlingame, California: Analytics Press, 2013.
- [2] "Executive Information System (EIS)", *Techopedia* [Online] Tillgänglig: <https://www.techopedia.com/definition/1016/executive-information-system-eis>. [Hämtad: 12 februari, 2018]
- [3] "Vilka är vi?" Sogeti, [Online] Tillgänglig: <https://www.sogeti.se/om-oss/vilka-ar-vi/>. [Hämtad: 24 maj, 2018]
- [4] SCA Skog. (2014) *Transportstyrningssystemet GATA* [PowerPoint-presentation]. Från SCA. [Hämtad 2 februari, 2018]
- [5] What is .NET? *Microsoft*, [Online] Tillgänglig: <https://www.microsoft.com/net/learn/what-is-dotnet> [Hämtad: 26 april, 2018]
- [6] M. Looock, Vad är .NET Framework? *Linnéuniversitetet*, [Online] Tillgänglig: <http://coursepress.lnu.se/kurs/inledande-programmering-med-csharp-ht12/files/2012/08/vad-ar-dotnet-framework.pdf> [Hämtad: 26 april, 2018]
- [7] Backbone.js, [Online] Tillgänglig: <http://backbonejs.org/> [Hämtad 22 mars, 2018]
- [8] "Introduction to the C# Language and the .NET Framework", *Microsoft*. [Online] Tillgänglig: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>. [Hämtad: 24 april, 2018]
- [9] R. W. Sebesta *Concepts of programming languages* (10th. Edition). University of Colorado at Colorado Springs, 2012.
- [10] "What is a KPI, Metric or Measure", *Klipfolio*. [Online] Tillgänglig: <https://www.klipfolio.com/blog/kpi-metric-measure> [Hämtad: 29 januari, 2018]
- [11] "Integrated development environment", *Computer Sweden: IT-Ord*. [Online] Tillgänglig: <https://it-ord.idg.se/ord/integrated-development-environment/> [Hämtad: 26 april, 2018]

- [12] Wagner, B. Wenzel, M. Hoag, S. Latham, L. “Language Integrated Query (LINQ)”, *Microsoft*. [Online] Tillgänglig: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/> [Hämtad 13 februari, 2018]
- [13] “What is Azure?”, *Microsoft*. [Online] Tillgänglig: <https://azure.microsoft.com/sv-se/overview/what-is-azure/> [Hämtad: 6 maj, 2018]
- [14] “Vad är Application Insights?”, *Microsoft*. [Online] Tillgänglig: <https://docs.microsoft.com/sv-se/azure/application-insights/app-insights-overview> [Hämtad: 6 maj, 2018]
- [15] “Inkrementell”, *Computer Sweden: IT-Ord*. [Online] Tillgänglig: <https://it-ord.idg.se/ord/inkrementell/> [Hämtad: 13 maj, 2018]
- [16] “Vad är scrum? Agil utveckling”, *Happiness*. [Online] Tillgänglig: <https://www.happiness.se/artiklar/vad-ar-scrum> [Hämtad 25 april, 2018]
- [17] “Datavisualisering”, *NE Nationalencyklopedin*. [Online] Tillgänglig: <https://www-ne-se.proxybib.miun.se/uppslagsverk/encyklopedi/l%C3%A5ng/datavisualisering> [Hämtad: 9 mars, 2018]
- [18] B. Bergström, *Effektiv visuell kommunikation* (Upplaga 10) Stockholm: Carlsson Bokförlag, 2016.
- [19] ”Om modeller och simulering av dem”, *Uppsala Universitet*. [Online] Tillgänglig: <http://www.it.uu.se/edu/course/homepage/oop/ht99/Projekt/html/node8.html> [Hämtad: 12 maj, 2018]
- [20] ”AJAX och PHP”, *HTMLHunden: En pragmatisk höghastighetsguide till webbutveckling*. [Online] Tillgänglig: <http://htmlhunden.se/dist/08-07-php-ajax.html> [Hämtad: 12 maj, 2018]
- [21] ”XML Tutorial” *w3schools.com*. [Online] Tillgänglig: <https://www.w3schools.com/xml/default.asp> [Hämtad: 12 maj, 2018]
- [22] ”JSON - Introduction” *w3schools.com*. [Online] Tillgänglig: [https://www.w3schools.com/js/js\\_json\\_intro.asp](https://www.w3schools.com/js/js_json_intro.asp) [Hämtad: 12 maj, 2018]
- [23] ”What i REST?”, *codecademy*. [Online] Tillgänglig: <https://www.codecademy.com/articles/what-is-rest> [Hämtad: 12 maj, 2018]
- [24] ”About HTML5 WebSocket”, *websocket.org*. [Online] Tillgänglig: <https://www.websocket.org/aboutwebsocket.html> [Hämtad: 21 maj, 2018]

- [25] A. Wessels, M. Purvis, J. Jackson och S. Rahman. "Remote Data Visualization through WebSockets," *2011 Eighth International Conference on Information Technology: New Generations*, s. 1050-1051, april 2011
- [26] "Cachelagring", *Microsoft*. [Online] Tillgänglig: <https://docs.microsoft.com/sv-se/azure/architecture/best-practices/caching#caching-in-distributed-applications> [Hämtad: 12 maj, 2018]
- [27] "Cacheminne", *Computer Sweden: IT-Ord*. [Online] Tillgänglig: <https://it-ord.idg.se/ord/cacheminne/> [Hämtad: 12 maj, 2018]
- [28] S. Few, "Data Visualization Effectiveness Profile" *Perceptual Edge*. [Online] Tillgänglig: [http://www.perceptualedge.com/articles/visual\\_business\\_intelligence/data\\_visualization\\_effectiveness\\_profile.pdf](http://www.perceptualedge.com/articles/visual_business_intelligence/data_visualization_effectiveness_profile.pdf) [Hämtad 16 maj, 2018]
- [29] "Duplex", *e-kommunicera.nu*. [Online] Tillgänglig: <http://www.e-kommunicera.nu/component/k2/item/212-duplex> [Hämtad: 17 maj, 2018]
- [30] W. H. DeLone, E. R. McLean, "Measuring e-Commerce Success: Applying the DeLone & McLean Information Systems Success Model," *International Journal of Electronic Commerce*, vol. 9, nr 1, s. 31-47, 8 december 2014.
- [31] E. Y. Gorodov and V. V. Gubarev, "Analytical Review of Data Visualization Methods in Application to Big Data," *Journal of Electrical and Computer Engineering*, vol. 2013, 7 sidor, oktober 2013.
- [32] M. Arvola. Interaktionsdesign och UX: om att skapa en god användarupplevelse (Upplaga 1:1). Lund: Studentlitteratur AB, 2014.
- [33] J. Åhlander. (2017). Konceptualisering, designkoncept och gränssnittsmetaforer. [PowerPoint-presentation] Tillgänglig: <http://apachepersonal.miu.se/~jimahl/DT168G/DT168G,%20IU127G%20Konceptualisering.pdf> [Hämtad 23 mars, 2018]
- [34] "Det här är eyetracking", *Tobii*. [Online] Tillgänglig: <https://www.tobii.com/sv/group/om-tobii/det-har-ar-eyetracking/> [Hämtad: 6 maj, 2018]

## Bilaga A: Resultat från användbarhetstest 2

