

# **Busify**

En reflektionsrapport för DAT255

Jonathan Fager  
Melinda Fülöp  
David Genelöv  
Sara Kinell  
Elias Svensson  
Annie Söderström



# Contents

<b>1</b>	<b>Introduktion</b>	<b>3</b>
1.1	Bakgrund . . . . .	3
1.1.1	Gruppens bakgrund . . . . .	3
1.1.2	Om ElectriCity . . . . .	3
1.1.3	Göteborg Energis problem med datatransparens . . . . .	4
1.2	Vår vision . . . . .	4
1.3	Vårt syfte och våra målsättningar . . . . .	4
<b>2</b>	<b>Metod</b>	<b>5</b>
2.1	Användning av Scrum . . . . .	5
2.1.1	Socialt kontrakt . . . . .	5
2.1.2	Sprints . . . . .	5
2.1.3	Roller . . . . .	5
2.1.4	Scrum Board . . . . .	6
2.1.5	Standup Meeting . . . . .	6
2.1.6	Sprint Planning . . . . .	6
2.1.7	Sprint Workflow . . . . .	7
2.1.8	Sprint Retrospectives . . . . .	7
2.1.9	Distribution av tid . . . . .	7
2.2	Value Proposition Canvas . . . . .	8
2.3	Git flow . . . . .	9
2.4	Firebase . . . . .	10
2.5	Test . . . . .	10
<b>3</b>	<b>Resultat</b>	<b>11</b>
3.1	Frontend . . . . .	11
3.2	Backend . . . . .	13
<b>4</b>	<b>Diskussion</b>	<b>15</b>
4.1	Sprint Planning . . . . .	15
4.2	Sprint Workflow . . . . .	15
4.3	Sprint Reviews . . . . .	16
4.4	Sprint Retrospectives . . . . .	16
4.5	Prototyp, Process och värde för intressenter . . . . .	17
4.5.1	Feedback från intressenter . . . . .	17
4.5.2	Interaktion med intressenter . . . . .	18

4.6	LEGO Scrum, Vision, Socialt kontrakt och halvtidsreflektion . . . . .	19
4.7	Vidareutveckling . . . . .	19
4.7.1	Businessperspektiv . . . . .	19
4.7.2	Tekniskt perspektiv . . . . .	20
<b>A</b>	<b>Appendix A - Bilder från produktutvecklingen</b>	<b>I</b>
A.1	Sprint 1 (18 April - 25 April) . . . . .	I
A.2	Sprint 2 (25 april - 2 maj) . . . . .	II
A.3	Sprint 3 (2 maj - 9 maj) . . . . .	III
A.4	Sprint 4 (9 maj - 16 maj) . . . . .	III
A.5	Sprint 5 (16 maj - 20 maj) . . . . .	IV
A.6	Sprint 6 (20 maj - 27 maj) . . . . .	V
A.7	Sprint 7 (27 maj - 01 Juni) . . . . .	VI
<b>B</b>	<b>Appendix B - Trelllokort</b>	<b>VIII</b>
<b>C</b>	<b>Appendix C - Design by Contract</b>	<b>XII</b>
<b>D</b>	<b>Appendix D - Findbugs</b>	<b>XIII</b>

## Contents

---

# 1

## Introduktion

Mobilapplikationen (hädanefter benämnd appen) Busify utvecklades av en tvärfunktionell grupp vid Chalmers tekniska högskola under våren 2016. Appen byggdes som en del i kursen Software Engineering Project (DAT255), för att ge oss som gruppmedlemmar en verklig erfarenhet av mjukvaruprojekt och den komplexitet de förknippas med. Projektet innefattade även att vi skulle reflektera över vad vi åstadkommit, vad vi önskat åstadkomma och hur skillnaden mellan dessa två områden hade kunnat överbryggas. Föreliggande rapport är resultatet av denna reflektion.

### 1.1 Bakgrund

För att förklara vad vi åstadkommit ges nedan en bakgrund till gruppen, projektet ElectriCity samt den problematik vi valde att bemöta.

#### 1.1.1 Gruppens bakgrund

Gruppen bestod av sex civilingenjörsstudenter vid Chalmers tekniska högskola. Fem av dessa läste sitt tredje år på programmet i industriell ekonomi med informationsteknisk inriktning, och en läste sitt fjärde år på det data tekniska programmet. De fem studenterna från Industriell Ekonomi kände varandra i varierande grad, medan datateknologstudenten kom ny till gruppen. Sammanlagt bestod gruppen av tre män och tre kvinnor utan tidigare erfarenhet av apputveckling.

#### 1.1.2 Om ElectriCity

Projektet ElectriCity är ett samarbete mellan bland andra Volvokoncernen, Göteborg Energi, Keolis och Chalmers tekniska högskola. Projektets syfte är att se hur framtidens kollektivtrafik kan utformas genom att testa och utvärdera eldriven busstrafik i Göteborg. Projektets aktörer tillhandahåller även en gemensam plattform för att kommunicera realtidsdata och annan information till utvecklare. År 2016 genomfördes kursen DAT255 i samarbete med projektet ElectriCity.

I april 2016 besökte studenterna i kursen DAT255 Volvo Bus Experience Center på Hisingen i Göteborg. Där förklarade de olika aktörerna bakom ElectriCity-projektet vilka problem de har. Gruppen blev särskilt intresserad av de problem som företaget Göteborg Energi upplevde med transparens och datahantering, och valde därför att arbeta med detta.

### 1.1.3 Göteborg Energis problem med datatransparens

Göteborg Energis roll i ElectriCity-samarbetet är bland annat att tillhandahålla den elinfrastruktur som bussarna behöver. Detta sker med hjälp av två så kallade ”laddningsstolpar”, som bussarna kopplar upp sig till vid ändhållplatserna på Johanneberg och Lindholmen. Göteborg Energi har även som uppgift att förse projektets andra aktörer med data från elinfrastrukturen. Företagets problem har hittills varit att de inte lyckas tillhandahålla data för hur mycket elektricitet som varje buss laddar vid de olika laddstationerna. Laddstolparna sparar information om hur mycket elektricitet som lämnar stolpen varje timme, men detta kunde inte kopplas till specifika bussar. Således saknade Volvo information om vilka bussar som är mest energieffektiva när de skulle bestämma vilka bussar som skulle gå i vilka omlopp, och de kunde därför inte optimera energianvändningen.

## 1.2 Vår vision

Eftersom den öppna ElectriCity-plattformen bland annat tillhandahåller data om vilken buss som befinner sig var, föddes snart vår idé. Genom att koppla ihop data om hur mycket energi som gick ut från respektive laddningsstation, med GPS-information om vilken buss som samtidigt befann sig vid laddstationen, kunde de olika bussarnas energieffektivitet beräknas. Detta sammanfattades under projektets första vecka till vår vision.

Projektets syfte: *Android appen Busify möjliggör för Göteborg Energi och Fredrik Persson att se samlad information per buss per timme. Det kommer innehåra att företaget kan ge bättre kundservice genom att besvara detaljerade frågor om energiförbrukningen för bussarna, samt att effektivare och mer faktabaserat planera, efter information om exakt hur mycket energi varje buss har och var de befinner sig. Vår app skiljer sig från andra på marknaden genom att data samlas och integreras på ett och samma ställe, med överskådliga vyer av bussar, laddningsstationer, linjer och dagar. All information är presenterad på samma ställe samt tillgängligt för energileverantör, verkstadsarbetare, bussförare, investerare och övriga intressenter.*

## 1.3 Vårt syfte och våra målsättningar

Tidigt i arbetet skrevs ett socialt kontrakt där vi som grupp bland annat funderade över vilka målsättningar vi skulle sträva efter under projektets gång. Mer om det sociala kontraktet kan läsas i denna rappers metodkapitel, men gruppens samlade målsättning med kursen definierades som att alla medlemmar skulle:

- ha roligt
- få nyttig erfarenhet av grupparbete
- få direkt och verklig erfarenhet av apputveckling
- få högsta betyg
- göra lika mycket arbete

Utifrån dessa målsättningar strukturerades sedan gruppens vidare arbete.

# 2

## Metod

Detta projekt byggde på användning av Scrum som ett teoretiskt ramverk, vilket skulle tillåta oss att bygga en produkt iterativt och inkrementellt i harmoni med kundinteraktioner. I denna rapport kommer vi att utgå ifrån att läsaren redan har en förståelse för ramverket, så vi kan fokusera på att beskriva och reflektera över vår användning av Scrum.

### 2.1 Användning av Scrum

Nedan kommer vi att beskriva vilket förhållningssätt, till Scrum som ramverk, vi som grupp hade under arbetets gång. Som poängterades under föreläsningen om agil mjukvaruutveckling inom fordonsindustrin är ”definitionen av agil agil”. Erafarna team kan kombinera olika metodologier men anpassar dem framför allt till sin situation, vilket vi också har gjort för att på bästa sätt uppnå projektets syfte.

#### 2.1.1 Socialt kontrakt

Det sociala kontraktet skapade först och kontrakt samlade punkter för gruppens kollektiva målsättningar för kursen och projektet, regler för kommunikation, möten och individuella uppgifter. Vi kunde återkomma till detta kontrakt och även uppdatera det för att ha en explicit källa för bästa praxis i hur vi ville arbeta tillsammans. Att diskutera målsättningar, och hur man vill jobba är viktiga aspekter när det gäller grupp dynamik och att få med sig andra människor<sup>1</sup>.

#### 2.1.2 Sprints

Vi valde att arbeta i sprints med sju dagars längd som mål, vanligtvis måndag till måndag. Vid tillfällen blev vi emellertid tvungna att lägga om start och slut för vår sprint för att anpassa oss till kursens deadlines och kundmöten.

#### 2.1.3 Roller

Vi ville ge alla medlemmar möjlighet att lära sig maximalt om Scrum som agilt ramverk under projektet. Därför valde vi att byta roller som Product Owner och

---

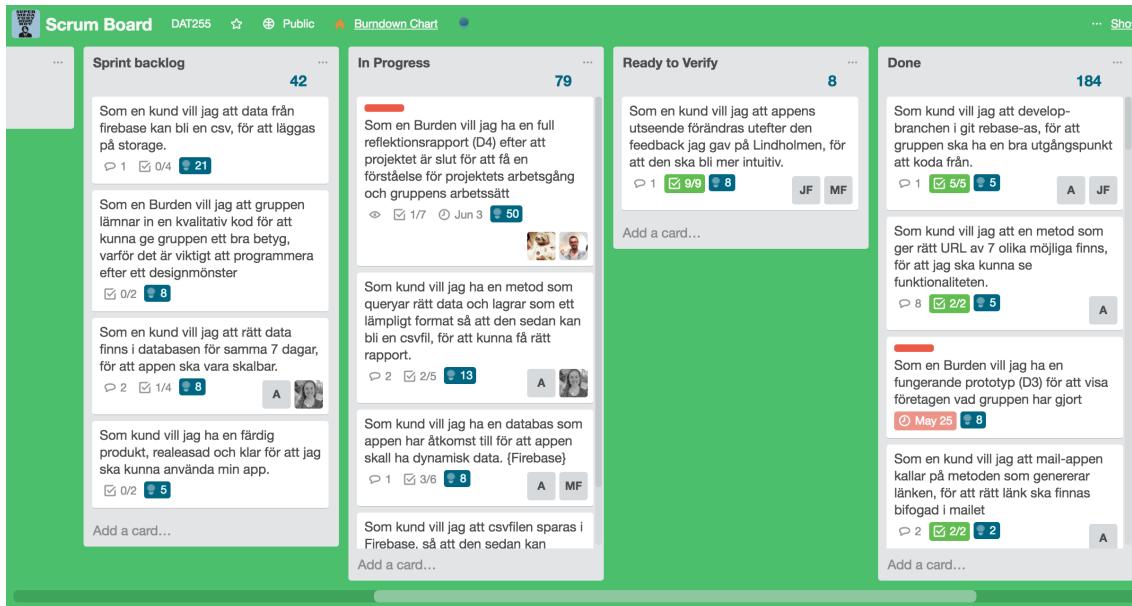
<sup>1</sup>Brunnegård, Viktor; Chalmers Ventures, Chalmers tekniska högskola. 2016. Entrepreneurship, föreläsning 2016-04-27.

## 2. Metod

Scrum Master vid varje ny sprint. I slutänden hade majoriteten av gruppen fått testa på båda dessa roller.

### 2.1.4 Scrum Board

Vi valde att använda tjänsten Trello (<https://trello.com/>) som en digital Scrum Board som kan ses i Figur 2.1.



**Figur 2.1:** En ögonblicksvy av gruppens digitala Scrum Board. Utanför bild (t.v.) finns även en “Product Backlog”.

### 2.1.5 Standup Meeting

Efter ett första försök att hålla Standup Meetings i person under första veckan valde vi att skapa en kanal på Slack (<https://slack.com/>) där vi varje dag skulle beskriva tre punkter för gruppen:

1. Vad gjorde jag igår?
2. Vad ska jag göra idag?
3. Vilka problem står jag inför som jag behöver hjälp med?

Detta system visade sig fungera väl för kommunikationssyften då vi ofta jobbade asynkront och på olika platser.

### 2.1.6 Sprint Planning

Under vår Sprint Planning satte vi först en gemensam vision för var vi ville att produkten skulle vara om en vecka. Denna vision skulle klart kunna beskriva en User Story som vi kunde leverera till kund inom sju dagar. Detta innebar ofta diskussioner och sketching på whiteboards. Efter det att hela gruppen kände sig bekväma med visionen började arbetet att lägga in kort i vår Product Backlog. Vi satte även DoDs (Definition of Done) under varje kort. Därefter skulle effort, vilket

används för att definiera den arbetsbelastning som gruppen bedömer en uppgift kräver, väljas. Först valdes ett kort ut med effort två. Detta gav oss möjlighet att enklare genomföra poker-processen där vi satte en estimerad effort på alla kort. Efter detta kunde Product Owner prioritera korten i Product Backlog och flytta över kort till Sprint Backlog tills det att vår estimerade kapacitet för sprinten fyllts, benämnd velocity. Efter vår första sprint kunde vår velocity också ligga till grund för nästkommande estimeringar av kapacitet. Planeringen avslutades då vår Sprint Backlog var fylld och prioriterad.

### 2.1.7 Sprint Workflow

Under varje sprint var korten på vår Scrum Board möjliga att arbeta på för alla medlemmar i gruppen. I början tenderade vi att ta på oss tasks och sedan arbeta ensamma, asynkront och på olika platser. Vi insåg emellertid snabbt att det var både svårt och frustrerande att, framför allt, programmera själv när programmeringsmiljön strulade eller när det var problematiskt att hitta dokumentation för hur man kunde bygga en speciell feature. Därför började vi att uppmuntra att man tog på sig kort två och två som pair programming. Detta ledde till ökad produktivitet och motivation för de involverade, men ledde till en minskad transparens över vem som hade gjort ett visst arbete på Github för den som skulle titta på vår statistik där, vilket är något vi vill poängtera.

### 2.1.8 Sprint Retrospectives

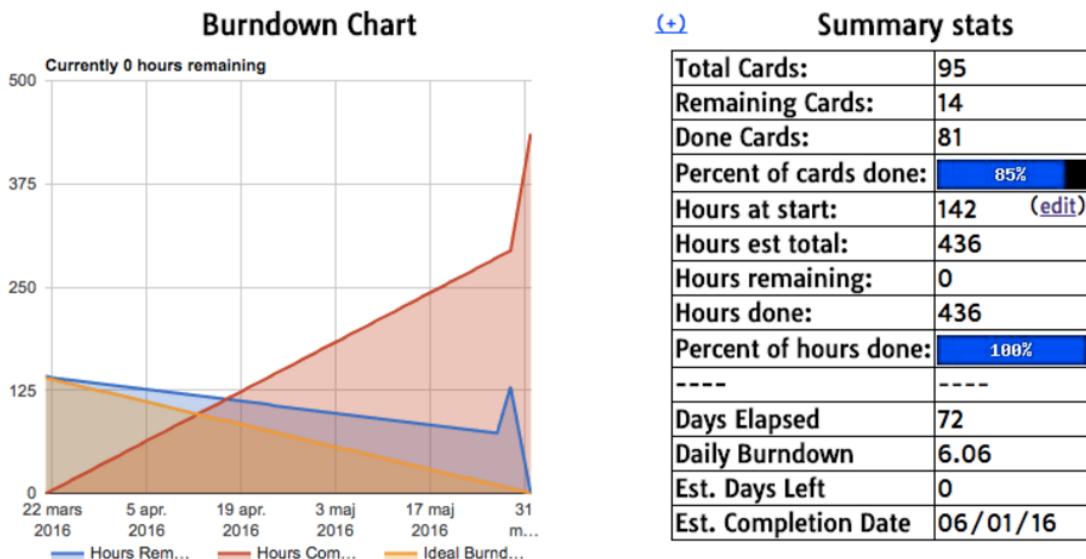
Under Sprint Retropsective gick vi runt i gruppen och ställde tre frågor till alla gruppmedlemmar:

1. Vad borde vi börja göra?
2. Vad borde vi sluta göra?
3. Vad borde vi fortsätta göra?

Då vi gjorde detta varje vecka hade vi möjlighet att utvärdera många beslut, särskilt gällande kommunikation, och vi kunde besluta om vi ville fortsätta på detta sätt eller inte.

### 2.1.9 Distribution av tid

DAT255 omfattar 7,5 högskolepoäng och pågick parallellt med gruppmedlemmarnas andra kurser. Disponerad tid enligt universitetet var alltså 20 timmar per vecka och person. Alla gruppmedlemmar förhöll sig till denna disponerade tid, men hade ofta perioder av mer eller mindre arbete beroende på relaterade deadlines till kandidatarbete eller övrigt arbete.



**Figur 2.2:** Burndown chart över projektets framfart mellan den 21 mars och den 1 juni 2016.

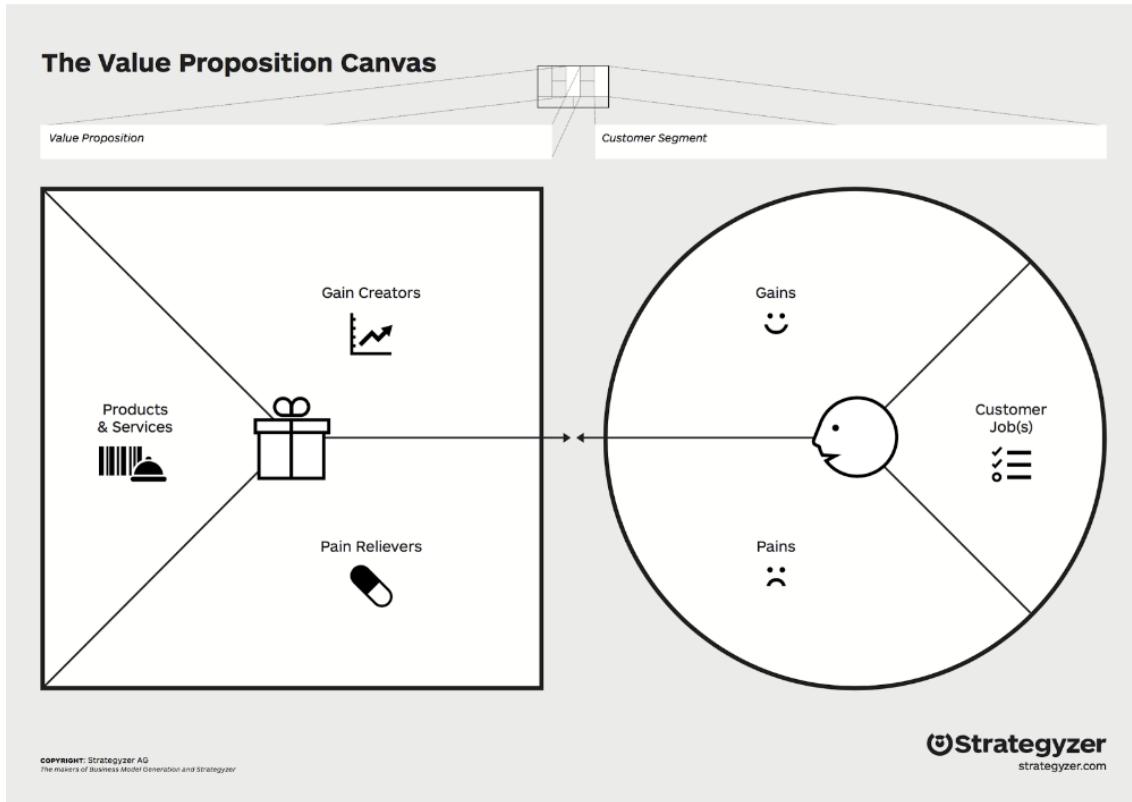
Projektets arbetsbelastning visas med en burndown chart, se Figur 2.2. Diagrammet är genererat från Trello, mer information om varje kort finns i Appendix B, och visar hur många kort som klassificeras som Done. Vissa kort som används i vyn för Trello är inte placerade i Done-kolumnen, t.ex. gamla kort som inte ska tas bort placerade i kolumnen Legacy, varför 85% av korten anges som klara i statistiksamtällningen ovan.

Efter input under presentationen på Lindholmen togs beslut att öka appens tekniska komplexitet, varför den blå (Hours Remaining) samt den röda (Hours Completed) kurvan visar en topp i arbetsprestation i slutskeden av projektet.

## 2.2 Value Proposition Canvas

Value Proposition Canvas, Figur 2.3, är ett verktyg för att konceptualisera och visualisera en kund och dess behov<sup>2</sup>, samt det värdeerbjudande man kan erbjuda för att möta detta behov. Viktor Brunnegård förespråkade detta verktyg under kursen och vi använde det för att sätta upp hypoteser om vår kund, samt vårt värdeerbjudande, för att sedan testa dessa mot kund genom kundintervjuer och kundinteraktioner.

<sup>2</sup>Brunnegård, Viktor; Chalmers Ventures, Chalmers tekniska högskola. 2016. Entrepreneurship, föreläsning 2016-04-27.



**Figur 2.3:** Value Proposition Canvas

## 2.3 Git flow

Valet av Git flow är baserat på rekommendationer från process supervision i kursen DAT255, rekommendation från andra erfarna team samt genom research från atlessians hemsida. Git flow användes som arbetsflöde enligt Vincent Driessens modell<sup>3</sup>. Två huvudsakliga branches, master och develop, levde under hela projektets tid. Från develop skapades nya branches för olika features, vilka med hjälp av pull requests i Github mergades med develop efter testning, kommentarer och peer review från minst en annan gruppmedlem.

Inför presentation för kund användes release-processer, vilka syftar till att testa den integrerade koden med alla dess existerande features. Dessa genomförs i git flows release branches. Git flows arbetssätt återkommande release-processer stämde väl överens med vår kontinuerliga kundkontakt och kundavstämning, varför denna branching strategi ansågs lämplig. Alla features som skapades mellan release-processer motsvarade en tydlig user story och genererade värde för kunden. Utöver release samt feature branches användes hotfix vid behov för att stödja master och develop. Hotfix utfördes när en bug eller liknande upptäcktes i master.

<sup>3</sup>Vincent Driessen, 2010. A successful Git branching model. <http://nvie.com/posts/a-successful-git-branching-model/> (Hämtad 2016-06-01).

## 2.4 Firebase

Firebase valdes för databashantering samt datalagring. Efter research av liknande lösningar hittade vi bra utbildningsmaterial via bland annat Udacity (<https://www.udacity.com/wiki/ud009>). Google, skapare av Android, hade dessutom nyligen ökat integration med Firebase, och länkat till denna lösning från sin primära sida för Androidutveckling (<https://developers.google.com/>), vilket motiverade valet ytterligare.

## 2.5 Test

Tester genomfördes av både kodförfattaren och andra gruppmedlemmar samt finns utförligt beskrivna i arbetsgruppens Git repository. Gruppen kom överens om Design by Contract (se Appendix C) vilket bland annat innebar att testning av varje enhet gjordes innan en pull request skapades i Github. De mindre enhetstester som varje kodförfattare utförde var exempelvis i form av test av en metod med inparametrar som förväntades fungera respektive med värdet som förväntades ge felmeddelande eller kasta undantag. Samtliga tester och felsökningar dokumenterades i gruppens gemensamma Google Drive för att säkerställa att fler inte letade efter samma buggar eller utförde tester som redan har gjorts på en viss feature branch. Detta underlättade även förståelse för någon som inte var lika bekant med koden som dess författare, exempelvis vid peer programming när paren uppdaterades.

Integrationstester<sup>4</sup> har genomförts under varje releaseprocess, i release branches numrerad med ökande primtal. Tidigare levande feature branches, borttagna efter merge med develop, testas då tillsammans. Även Android Studios inbyggda debug samt FindBugs har körts under releaseprocesserna. Såväl under integrationstester som acceptansteller har appen körts både på emulatorer, via Android Studio, och via gruppmedlemmars Android telefoner.

Acceptansteller<sup>5</sup> har gjorts både internt och externt. Genom att ha en gruppmedlem som agerade som Product Owner varje vecka har godkännande av utförda kort i Trello (se Appendix B) i majoriteten utav fallen gjorts av någon annan än den/de som utförde uppgiften. Kodmässigt har även samtliga funktioner acceptanstat genom att gruppen beslutade att pull requests i Github skulle testas och godkännas av någon annan än den som skapade den och koden. Externa acceptansteller har gjorts i de prioriterade och kontinuerliga mötena med kund, vilka förklaras vidare i avsnittet Prototyp, process och värde för intressenter.

---

<sup>4</sup>Software Testing Fundamentals. <http://softwaretestingfundamentals.com/integration-testing/> (Hämtad 2016-06-01).

<sup>5</sup>Software Testing Fundamentals. <http://softwaretestingfundamentals.com/acceptance-testing/> (Hämtad 2016-06-01).

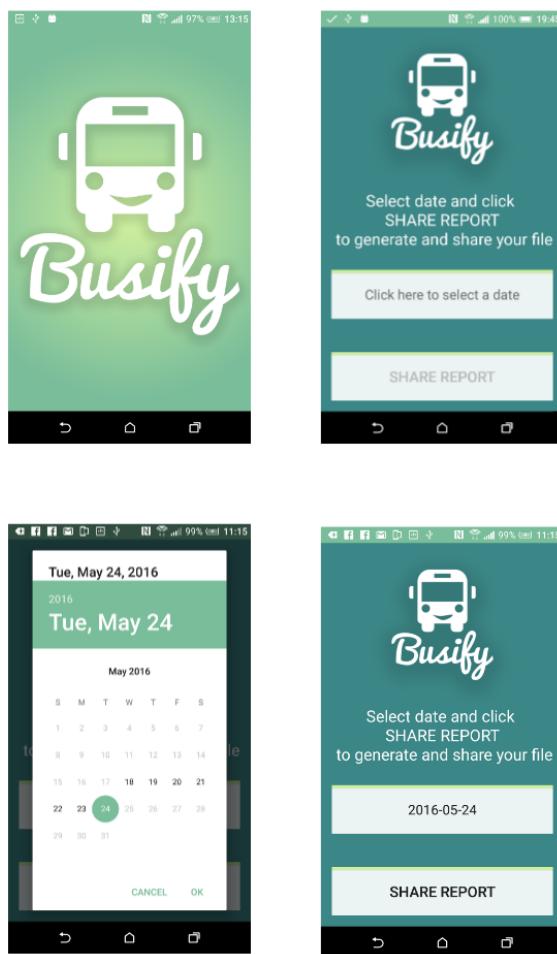
# 3

## Resultat

Under projektets gång genomförde vi sju sprints. Varje sprint ledde till ett ökat kundvärde, för att sedan byggas på iterativt och inkrementellt. Resultat av varje individuell sprint finns i Appendix A. Nedan presenteras vårt slutresultat efter Sprint 7. En diskussion om hela processen följer i nästa kapitel.

### 3.1 Frontend

Figur 3.1 och 3.2 visar den frontend som vår app använde efter vår sista sprint.

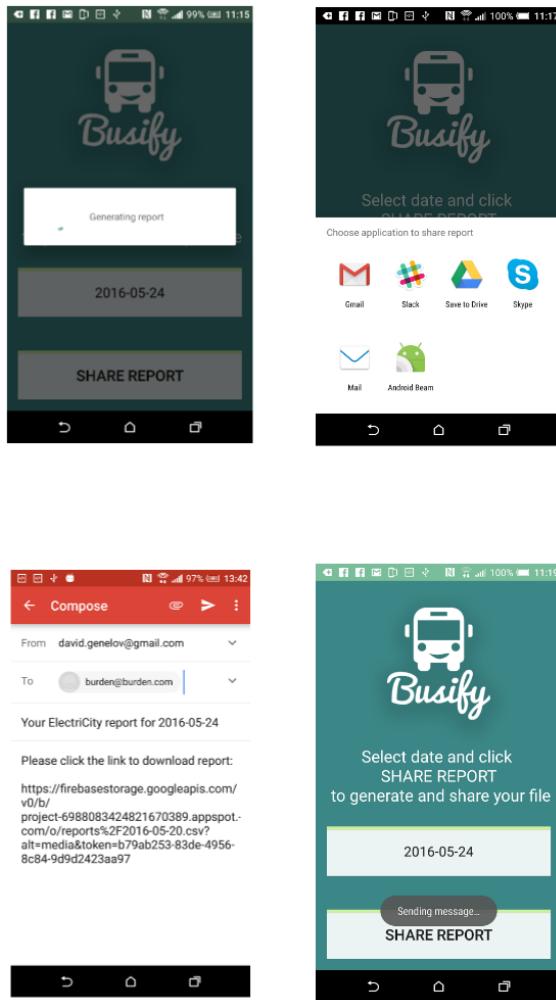


**Figur 3.1:** Appens utseende: Uppstartsvy; ej ifylld huvudvy; vy för att välja datum; ifylld huvudvy.

### 3. Resultat

---

Det första som händer när appen öppnas är att en splash-screen laddas. Därefter når användaren den primära vyn med en knapp för datumval samt en knapp för att dela rapporten, denna är dock inte först möjlig att trycka på. På vyn kan användaren alltså först välja ett datum genom en popup-ruta med en synlig kalender. Vi valde att endast tillåta användaren att välja datum mellan den 18-24 maj då vi skapat data för dessa datum i vår databas. Detta motiveras med att ElectriCitys API för dataåtkomst låg nere under hela projektets gång, vilket resulterade i att arbetsbelastningen för att lägga in data i databasen ökade då detta istället behövde göras manuellt. Vidare räckte sju dagars data väl för att visa appens funktionalitet. Efter att datum valts kan användaren trycka på ”SHARE REPORT” vilket resulterar i att tillgängliga appar för delning av en länk till denna data blir valbara. Detta kollar på en process som beskrivs i nästa avsnitt angående appens backend. Om Gmail-appen väljs, som i figuren, får användaren möjlighet att dela rapporten med ett godtyckligt antal mottagare. Efter att mailet har skickats iväg får användaren en bekräftelse i appen. Därefter kan användaren välja att upprepa processen eller stänga appen.



**Figur 3.2:** Appens utseende: Generating report vy; vy för att välja app att dela länk med; utseende på det mail som uppstår då användaren väljer att dela via Gmail; vy för att visa att filen skickas/delas.

## 3.2 Backend

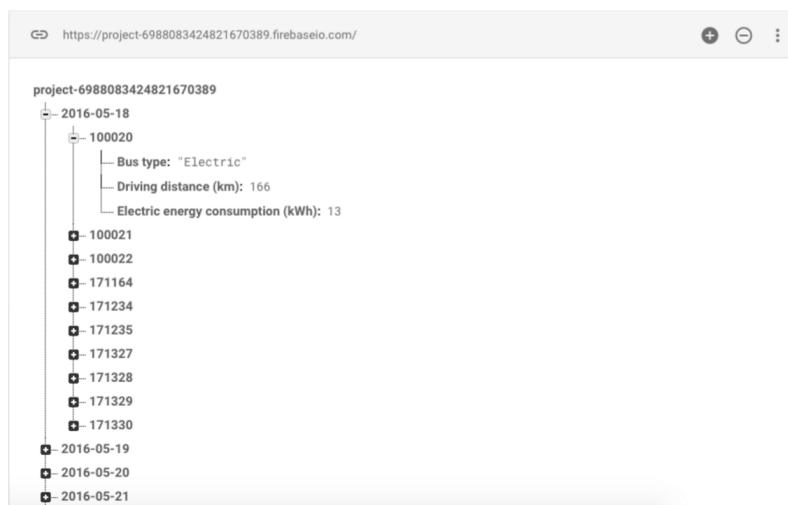
Som backend användes databas samt storage från Firebase, denna konstruerades så platt som möjligt eftersom det är lämpligt för NOSQL databaser. I databasen lagrades värden från den 18 maj 2016 till den 24 maj 2016, se Figur 3.3. Samtliga dagar har information om alla bussar, vilka har unika identifierare, enligt Electricity Innovation Platform<sup>1</sup>. Databasen innehåller vidare information om typ, körd sträcka samt elkonsumption för varje buss (se Figur 3.4).

<sup>1</sup>Electricity Innovation Platform 2016, Lindholmen Science Park,  
<http://platform.goteborgelectricity.se/api/bussar>. (Hämtad 2016-06-01).

### 3. Resultat



**Figur 3.3:** Information om bussarna lagras för en tidsperiod om en vecka.



**Figur 3.4:** Information om varje buss lagras under varje datum.

Informationen i databasen baseras på beräknad körsträcka mellan laddstationer på Lindholmen och Johanneberg. Elkonsumptionen har uppskattats med hjälp av information erhållen från Keolis, vilka idag ser hur mycket elektricitet som går ut från laddningsstolarna per timme.

De .csv-filer som genereras och skickas lagras i Firebase Storage, vilket visas i Figur 3.5. Om en fil redan finns används den bifogad i det mail som användaren skickar, annars genereras en ny fil, vilken även sparas i Firebase Storage. I Figur 3.5 visas hur filer för två datum har genererats.

gs://project-6988083424821670389.appspot.com > reports				
<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	2016-05-20.csv	395 B	text/csv	Jun 1, 2016
<input type="checkbox"/>	2016-05-24.csv	397 B	text/csv	Jun 1, 2016

**Figur 3.5:** Filer som sparas utifrån informationen i databasen finns tillgängliga i Firebase Storage. Ovan visas hur två filer, för den 20 maj 2016 samt 24 maj 2016, har genererats av användaren.

# 4

## Diskussion

I följande kapitel reflekterar vi över problem och lärdomar som uppstod under arbetets gång. Kapitlet är strukturerat efter moment i Scrum samt punkter vi vill elaborera kring.

### 4.1 Sprint Planning

Det visade sig vara svårt att definiera DoDs för uppgifter där vår kunskap och erfarenhet var bristande. Detta ledde till otydligt formulerade DoDs, särskilt i början för inlärningskorten, vilket ledde till att många av gruppmedlemmarna inte ansåg sig som färdiga med uppgifter. Det innebar att gruppen började skapa prototypen senare än egentligen möjligt. Genom att skapa mindre omfattande kort för user stories och tasks ökade sannolikheten för att gruppen kunde identifiera de viktigaste aspekterna för att kortet skulle anses vara klart. Dessutom minskade antalet kort som var ”halvfärdiga” vid sprintens slut. Precis som Jan-Philipp Steghöfer varnade för under första mötet på Lindholmen felbedömde vi, precis som han berättade att många mjukvaruutvecklingsteam gör, tidsåtgången för många uppgifter med ungefärlig faktor 2,5<sup>1</sup>.

Vi upplevde problem med att ta tillvara på samtliga gruppmedlemmars kunskap när vi arbetade enskilt i början eller när samtliga inte deltog under vissa sprint plannings. Detta resulterade exempelvis i att delar av gruppen började designa en databas utifrån relationsdatabas-principer och definierade DoDs, trots att andra i gruppen redan hade kunskap om att detta inte var rätt tillvägagångssätt för Firebase-databasen, som är av No-SQL typ, som sedan användes. Genom att prioritera att samtliga i gruppen medverkade under alla sprint planning möten skapades större möjlighet att den aggregerade kunskapen på bästa sätt togs tillvara och att tid och energi lades på rätt saker.

### 4.2 Sprint Workflow

Gruppmedlemmarna hade, trots stegvis ökat samarbete under projektet, kunskap inom begränsade områden. Efter de första veckorna insåg vi det stora värdet av att programmera i par, trots svårigheter att synkronisera mötestider. Efter att par

---

<sup>1</sup>Steghöfer, Jan-Philipp. 2016. Chalmers tekniska högskola. Handledning 2016-04-15.

bildades fortsatte de till stor del och samma duo programmerade tillsammans under flera veckor. Att istället säkerställa, genom ett rullande schema eller liknande, att alla programmerar med alla någon gång under projektet, precis på samma sätt som vi arbetade med Scrum Master- och Product Owner-rollerna, hade en större förståelse för appens alla delar kunnat skapas hos samtliga gruppmedlemmar. Detta hade dessutom varit väl i linje med den initiala vision som sattes.

Vid ett tillfälle fastnade två gruppmedlemmar som parprogrammerade på en uppgift i två veckor, vilket ledde till frustration och försening av produktutvecklingen. En potentiell lösning på detta problem hade varit att ta in en annan gruppmedlem på uppgiften för att få nya infallsvinklar på problemet snabbare. Två andra hade även kunnat arbeta på en reservplan, samtidigt som övriga gruppmedlemmar kunde ha fortsatt med det övriga arbetet för att föra produktutvecklingen framåt.

Gruppmedlemmarna fastnade även på grund av att gruppens Daily Standups, där avstämning för arbetsprocessen görs, innehöll en tredje punkt definierad som “Vilka problem står jag inför?”. När en gruppmedlem då poängterade ett problem skapades därför inte en förfrågan efter assistans. Genom att senare tillägga "... och vad behöver jag hjälp med?" skapades ett mer lösningsorienterat synsätt inom gruppen och olika medlemmar kunde hjälpa varandra direkt istället för att endast notera att problem hade uppstått. När någon hade uttryckt ett behov av hjälp i sin Daily Standup kunde den följas av att en annan gruppmedlem direkt erbjöd sig hjälpa till med problemet.

### 4.3 Sprint Reviews

Det visade sig vara svårt att uppskatta velocity för kommande veckor, särskilt med få historiska sprints att jämföra med. Då arbetsbelastningen var svår att uppskatta hade vi aldrig samtliga kort för veckan överfördas till kolumnen för “Ready to Verify” i tid till vår Sprint Review. Genom att kontrollera under sprintens gång hur vi låg till med antalet avverkade efforts i förhållande till utsatt velocity skulle det kunnat ha skapas möjligheter att ytterligare stämma av om någon satt fast eller om vissa kort var felpokrade, och sedan lära sig av det. På så sätt kunde en mer korrekt bedömning av kommande velocity ha gjorts. Vidare kunde dokumentation och diskussion ha gjorts av de felpokrade korten på sprint review för att på så vis skapa lärande och förhindra upprepning. Vidare hade detta också gjort att de gruppmedlemmar vilka inte hade arbetat med kortet hade förstått kortets komplexitet bättre. Det hade främjat gruppdynamiken och skapat förståelse för att alla bidragit lika mycket, trots att det visuellt inte synts på kortens effort.

### 4.4 Sprint Retrospectives

Det var fördelaktigt att lägga mycket tid på Sprint Planning, Sprint Retrospectives samt Sprint Review. Fyra timmar varje vecka, 20% av gruppens tid, var schemalagt för detta under arbetets gång. Detta skapade en tydlig riktning för hur arbetet

skulle fortlöpa den kommande tidsperioden. DoDs och prioriteringsordning kunde diskuteras noggrant så att samtliga var överens om arbetsordningen samt så att resurser kunde allokeras effektivt. Vidare var det fördelaktigt att gruppen alltid dokumenterade dessa möten så att de gruppmedlemmar som inte kunde närvara kunde ta del av informationen. Det innebar att samtliga skapade sig förståelse för målbilden och att vi arbetade mer som en helhet.

Det fanns svårigheter under sprint retrospectives. Till exempel var det som gruppmedlem svårt att ärligt kunna öppna upp sig och berätta om vad som grupp dynamiskt fungerat bra och vad som behövde ändras. Om mer öppna diskussioner kunde ha förts kunde gruppen inte bara ha presterat bättre, utan också trivts ännu bättre i varandras sällskap. Att fokusera på längre programmerings-/grupparbetessunder initialt<sup>2</sup>, precis som vår mentor Oskar Hagberg poängterade under första mötet på Lindholmen i mitten på april, istället för i slutet, samt att genomföra fler teambuilding-aktiviteter i början hade varit bra för att ett öppet klimat skulle kunnat ha skapas. Flera i gruppen var inte bekanta sedan tidigare, vilket var en ännu större anledning till att lära känna varandras styrkor och svagheter genom olika typer av teambuilding-övningar.

## 4.5 Prototyp, Process och värde för intressenter

Vi insåg tidigt att värde för intressenten var, för oss, den centrala delen i detta arbete. Värde för intressenten skapas genom produkter eller prototyper, som i sin tur skapas av våra arbetsprocesser. Vårt förbättringsarbete av processen blev därmed inte ett arbete för arbetets skull, utan sågs hela tiden som ett steg på vägen mot att producera bättre produkt som skulle leverera ökat värde för vald intressent. Kontinuerlig kommunikation skedde därför med Fredrik Persson för att säkerställa att Göteborgs Energis krav uppfylldes samt för att visa var i processen vi var. Det innebar att han som kund hade förståelse för vilken typ av slutprodukt vi hade möjlighet att leverera.

### 4.5.1 Feedback från intressenter

Vi upptäckte under arbetets gång vikten av kunddialog<sup>3</sup>, något som Jan-Philipp Steghöfer underströk under det första mötet på Lindholmen i mitten av april. Initialt var vår tanke att visualisera data och producera snygga grafer. Men det visade sig att kund endast ville ha rå data, siffervärdet i .csv-format, att skicka vidare via mail. Även under demodagen på Lindholmen upptäcktes flera möjliga användningsområden, vilka skulle kunna användas i en annan app eller vidareutveckling av denna, samt input om designmässiga delar. Genom att testa produkten fysiskt vid fler tillfällen hade en mer intuitiv design kunnat skapas ännu tidigare.

---

<sup>2</sup>Hagberg, Oskar. 2016. The Techno Creatives. Handledning 2016-04-15.

<sup>3</sup>Steghöfer, Jan-Philipp. 2016. Chalmers tekniska högskola. Handledning 2016-04-15.

### 4.5.2 Interaktion med intressenter

Vid visning av vår prototyp den 10 maj försökte vi att agera efter lärdomar från lego-övningen för att kunden dels skulle kunna ge feedback men att ändå inte ge denne för mycket ”frihet”. Vi upptäckte att vi trots detta inte ställde tillräckligt slutna frågor eftersom vi kände ett behov av att ändra vår riktning mitt under en sprint, vilket gick emot Scrum-metodiken och lärdomarna från lego-övningen. Vi tänkte att ett möte med kund mellan sprint retrospective/review och sprint planning, t.ex. som i vårt fall under måndag lunch, hade gjort att vi inte behövt uppleva detta dilemma. Vi begränsade oss därför till två alternativfrågor den 20 maj för att det var sent i processen och för att vi inte ville ge kunden utrymme för att uttrycka fler önskemål.

Vi upplevde tydligt under vårt projekt att kundnöjdheten inte ökar med antal features. Detta stämmer väl överens med vad föreläsaren från VCC berättade i sin föreläsning om mjukvaruutveckling i bilindustrin; nämligen hur Steve Jobs stryker funktioner, fokuserar på enkelhet och på så sätt ökar företagets utvecklingshastighet<sup>4</sup>. Med många features, som hos Ericsson och Nokia, ökar också antalet buggar, vilket var något vi naturligtvis avsåg undvika.

Daniel Sjölje presenterade under Human Computer Interaction-föreläsningen the McGurk effect, vilken illustrerar hur uppfattningen av något beror på hur det presenteras och sammanhanget. Eftersom vår app var tillägnad ett arbetsområde på Göteborg Energi och eftersom en presentation av funktioner kommer göras innan dess användning, precis som under demonstrationer på Lindholmen den 25 maj, skapades goda förutsättningar för att uppfattningen av funktionaliteten skulle bli korrekt.

Vi tog även tillvara på feedback från Daniel Sjölje under demonstrationen i slutet på maj och från föreläsningen där han poängterade att funktioner som knappar eller flikar kan göras gråa och visas som inaktiva för att underlätta användarens förståelse<sup>5</sup>. Vi lärde oss även andra koncept från föreläsningen av Daniel Sjölje. Vi jobbade mot ”rule-of-thumb” i appens design, vi försökte att ha så få vyer som möjligt för att göra appen direkt översiktig och begriplig samt inte skapa överbelastning och använde färger för att skapa ett uttryck som är kopplat till bussarna så användaren förstår appens syfte. Vi arbetade, precis som Daniel Sjölje poängterade under Human Computer Interaction-föreläsningen, med recognition framför recall<sup>5</sup>. Vi använde t.ex. en kalendervy som liknar Androids inbyggda kalender, samt inbyggda mailfunktioner istället för att skapa egna, för att användaren skulle känna igen sig. Vi arbetade med att skapa så intuitiv app som möjligt genom att arbeta med redan kända och etablerade funktioner som default-mail. Detta reflekterar vad Daniel Sjölje förespråkade angående affordances.

Vi strävade efter enkelhet för att inte förvirra användaren eller tillbringa tid på

---

<sup>4</sup>VCC. 2016. Föreläsning 2016-04-25.

<sup>5</sup>Sjölje, Daniel. 2016. Chalmers tekniska högskola. Föreläsning 2016-04-29.

icke värdeaddande funktioner. Vi använde t.ex. stora knappar, vilket är i linje med Fitt's law som, applicerad i detta projekt, betonar vikten av att ha en låg anstränning för att göra någonting som appanvändare<sup>5</sup>. Vi använde även ett kort avstånd för förflyttning av tummen mellan datumval och "share report"-knappen.

## 4.6 LEGO Scrum, Vision, Socialt kontrakt och halvtidsreflektion

Vi tyckte det var utmanande och lärorikt att, precis som Mikael Öhman från Spotify berättade om, få fungera som en squad i Spotify-terminer<sup>6</sup>. En liten grupp som gör vad de tycker är bäst för att lyckas med sitt mål. En utmaning i betygssättningssammanhang var att, som på Spotify, uppmuntra misslyckanden för att skapa innovation då projektet till syvende och sist var betygsgrundande. Dock var en viktig faktor i vårt sociala kontrakt att lära oss om agil mjukvaruutveckling, varför misslyckanden och omskrivningar av kod sågs som någonting nödvändigt och i slutänden som lärdomar. Notera prioriteringen i vårt sociala kontrakt, då denna var avsiktlig. Gruppen upplever att främst de första delarna i gruppens målsättning uppfylldes. Den sista punkten har dock varit svårast att veta om den har blivit uppfylld. Trellokorten samt daily-standups har använts för att försöka uppnå målet.

Halvtidsreflektionen gav oss som grupp en möjlighet att titta tillbaka på vår arbetsprocess samt ge en uppfattning av hur långt vår produkt hade kommit. Detta gav oss en insikt i var vi behövde lägga fokus framöver för att uppnå våra mål som vi beskrev i Kapitel 1. Vidare insåg vi tydligt att vårt beslut att involvera kund var ett bra beslut som vi ville fortsätta arbeta med.

## 4.7 Vidareutveckling

Nedan följer en reflektion om hur projektet skulle kunna tas vidare.

### 4.7.1 Businessperspektiv

Med ElectriCity Innovation Platforms API tillgänglig hade realtidsvärden kunnat användas. Då hade värden för varje buss GPS-koordinater kunnat användas. Dessa hade kunnat kombinerats med GPS-koordinaterna för laddstationerna fanns tillgängliga via vår kund Fredrik på Göteborg Energi. Genom att kontrollera varje buss position hade en mer korrekt uppskattning av elektriciteten intagen av varje buss kunnat beräknas, med antaganden att energin lämnar laddstationerna linjärt under dagen samt att energi ut från laddstationer motsvarar energi in i bussarna.

ElectriCity Innovation Platform innehåller också information om total körsträcka, vilken hade kunnat genereras för varje buss i databasen. Vidare hade API-tillgång

---

<sup>6</sup>Öhman, Mikael. 2016. Spotify. Föreläsning 2016-05-13.

möjliggjort funktioner för att uppdatera värden i realtid. Som databasen är implementerad idag tar den endast hänsyn till ”hela dagar”, med argumentet att fullständig information för dessa dagar har genererats. Därav visades inte den 25 maj som alternativ för nedladdning av rapport för besökarna under presentationsdagen den 25 maj.

Vidareutvecklingsmöjligheter, som t.ex. har identifierats genom tester mot potentiella kunder under presentationsdagen den 25 maj 2016, är att lägga till funktionalitet till appen i form av att inkludera koldioxidutsläpp i rapporten som genereras. Vidare skulle datumintervall kunna möjliggöras, dvs. ett alternativ för användaren att generera rapporter med information om bussar mellan den 18 maj 2016 och den 22 maj 2016 exempelvis. Denna funktionalitet skulle skapas med hjälp av en utbyggnad av informationen i Firebase. Detta hade enklare kunnat göras genom att utveckla en idé från projektets början, då vi ännu inte visste att ElectriCitys API låg nere. Den var att skapa ett skript via Firebase i t.ex Javascript som dagligen kunde hämta data från plattformens API och skriva detta till databasen. Genom denna typ av lösning kan skriptet enkelt byggas ut för att sedan spara mer data och på så sätt kunna ge mer information till databasen och i slutänden även användaren. Vidare kan även alternativet att generera halvårs- eller kvartalsrapporter utvecklas, vilket skulle kunna ligga i linje med företagens interna deadlines och rapporteringar.

Under presentationsdagen den 25 maj 2016 upptäcktes nämligen fler användningsområden. Volvo skulle se värde i applikationen genom att kunna se över hur mycket energi varje buss drar och på så vid få beslutsunderlag inför kommande investeringar i vissa busstyper exempelvis. Keolis skulle kunna koppla informationen om varje buss med dess förare, och exempelvis bedöma hur miljövärtigt busschauffören har kört. Att vidga applikationsområdet för databasens information skulle dock kräva olika interface för olika användare, och därmed olika applikationer. Liknande appar skulle exempelvis kunna skicka andra typer av filer eller inkludera detaljer om bussarnas inköpspris eller förare. Med hjälp av databasen kan alltså flera olika appar skapas, vilka alla då kan riktas mot olika intressenter och kunder. Vår tätta kundkontakt och vårt tydliga fokus mot Fredrik Persson på Göteborg Energi gör att sammanhang och applikationens användningsområde upplevs intuitivt för kunden. Att bredda kundgruppen har inte varit aktuellt under vårt projekt, men skulle kunna vara av intresse för vidareutveckling av vårt resultat.

### 4.7.2 Tekniskt perspektiv

Vidareutveckling av appens kod skulle kunna göras i form av att implementera ett designmönster och på ett ännu tydligare sätt separera användargränssnitt från logik, vilket exempelvis kan göras genom designmönstret Model View Controller eller, det inom Android ofta förekommande, Model View Presenter<sup>7</sup><sup>8</sup>. Fokus under projektet

<sup>7</sup>Gómez Sánchez, Pedro Vicente. 2016. Software Design Patterns on Android, <http://www.slideshare.net/PedroVicenteGmezSchn/software-design-patterns-on-android/>. (Hämtad 2016-06-02).

<sup>8</sup>Martin, Robert C. 2016. Clean Architecture, <https://vimeo.com/43612849>. (Hämtad 2016-06-02).

var större på att skapa fungerade kod, än att strukturera koden enligt teoretiska principer. Detta är ett fenomen som kallas architectural technical debt, vilket uppkommer när målsättning om snabb leverans skapar höga kostnader för det omarbete som måste göras för att långsiktigt underhålla och utveckla mjukvara<sup>9</sup>. Citatet och rådet "Write code for your mates, not for the machine"<sup>13</sup> hade kunnat tas tillvara genom att refaktorera kod vid varje pull request, och ännu grundligare vid varje release, för att säkerställa att ett helhetsperspektiv behölls och att en god, skalbar, kodmässig struktur skapades. Under de refaktoreringsprocesser som genomfördes noterades dock värdet av att diskutera och implementera kunskaper från tidigare kurser när det gäller goda designprinciper.

Ur säkerhetssynpunkt och för att skapa ett konsistent beteende för appen hade trådhantering kunnat integreras<sup>10 11</sup>. Genom att vidareutveckla vårt resultat med exempelvis thread scheduling och thread priorities säkerställs att rätt tråd exekveras i rätt ordning och att data alltid hämtas från databasen innan emailet skickas iväg, exempelvis. Med asynkrona anrop kan resultatet från en exekvering variera. Säkerhetsmässigt skulle även Firebase inställningar kunna modifieras för att inte erbjuda offentlig tillgång.

---

<sup>9</sup>Carnegie Mellon University. 2016. Architectural Technical Debt, [http://www.sei.cmu.edu/architecture/research/arch\\_tech\\_debt/](http://www.sei.cmu.edu/architecture/research/arch_tech_debt/). (Hämtad 2016-06-02).

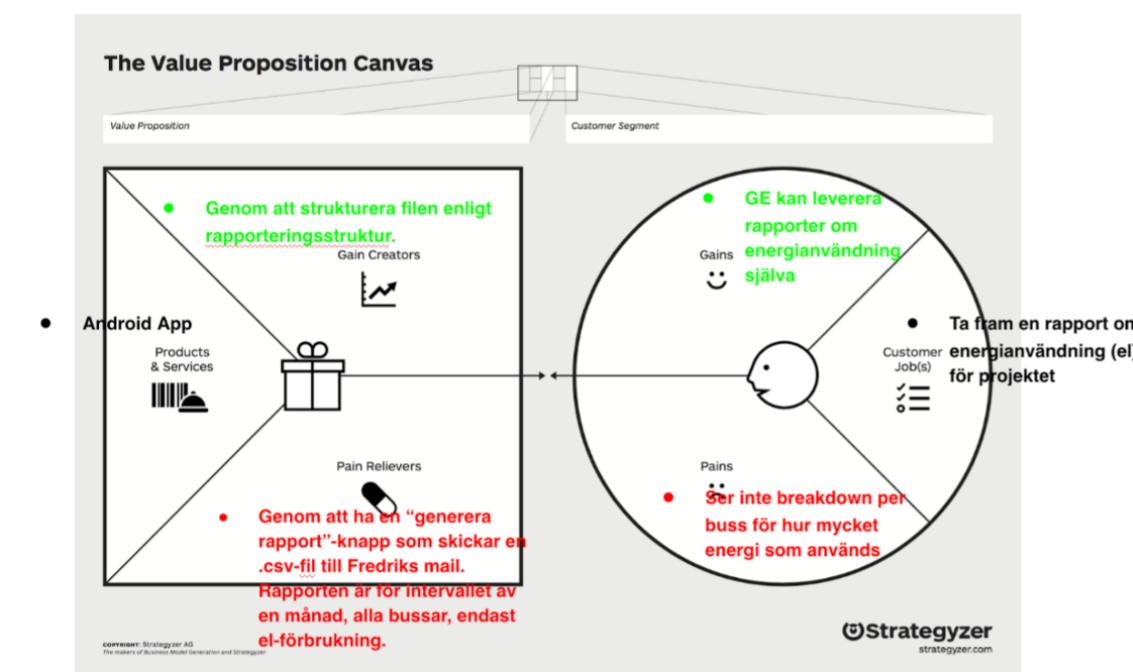
<sup>10</sup>Android Design Patterns. 2016. Thread scheduling in Android, <http://www.androiddesignpatterns.com/2014/01/thread-scheduling-in-android.html>. (Hämtad 2016-06-02)

<sup>11</sup>Android Developers. 2016. Processes and Threads, <https://developer.android.com/guide/components/processes-and-threads.html>. (Hämtad 2016-06-02).

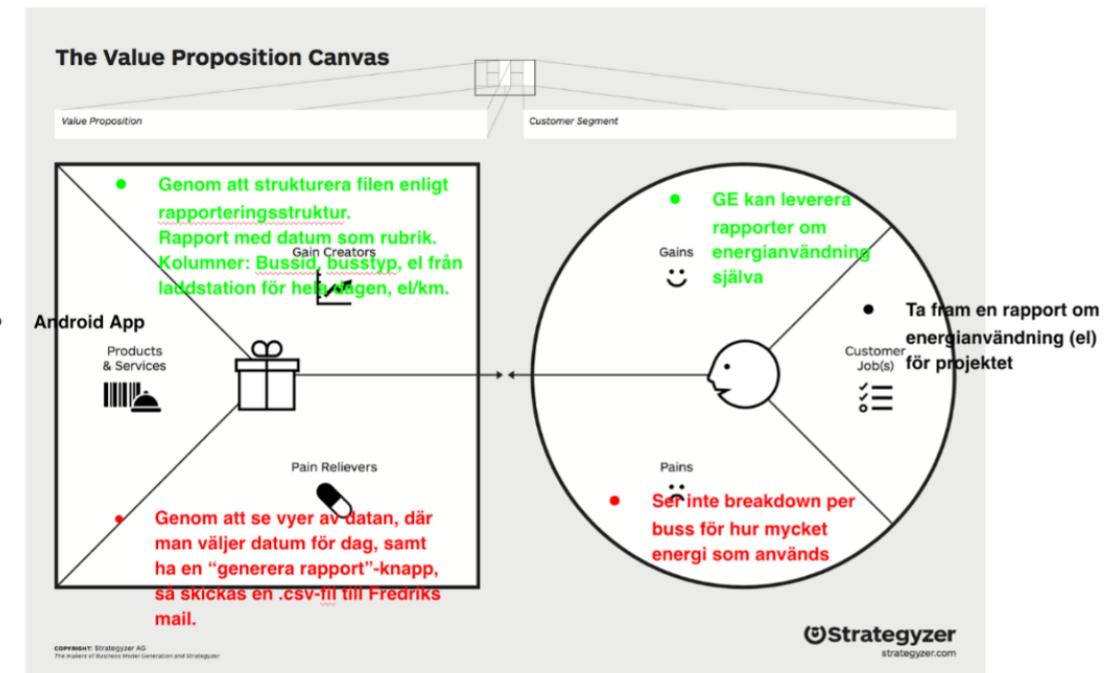
# A

## Appendix A - Bilder från produktutvecklingen

### A.1 Sprint 1 (18 April - 25 April)

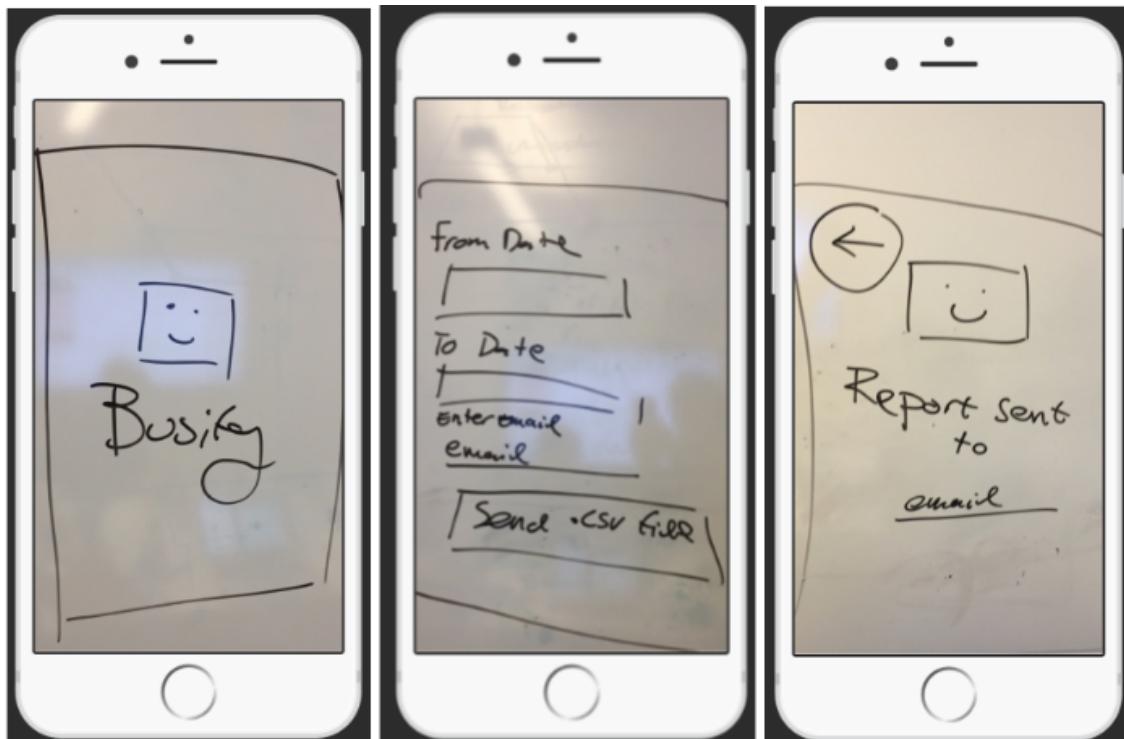


Figur A.1: Gruppens ursprungliga Value Proposition Canvas



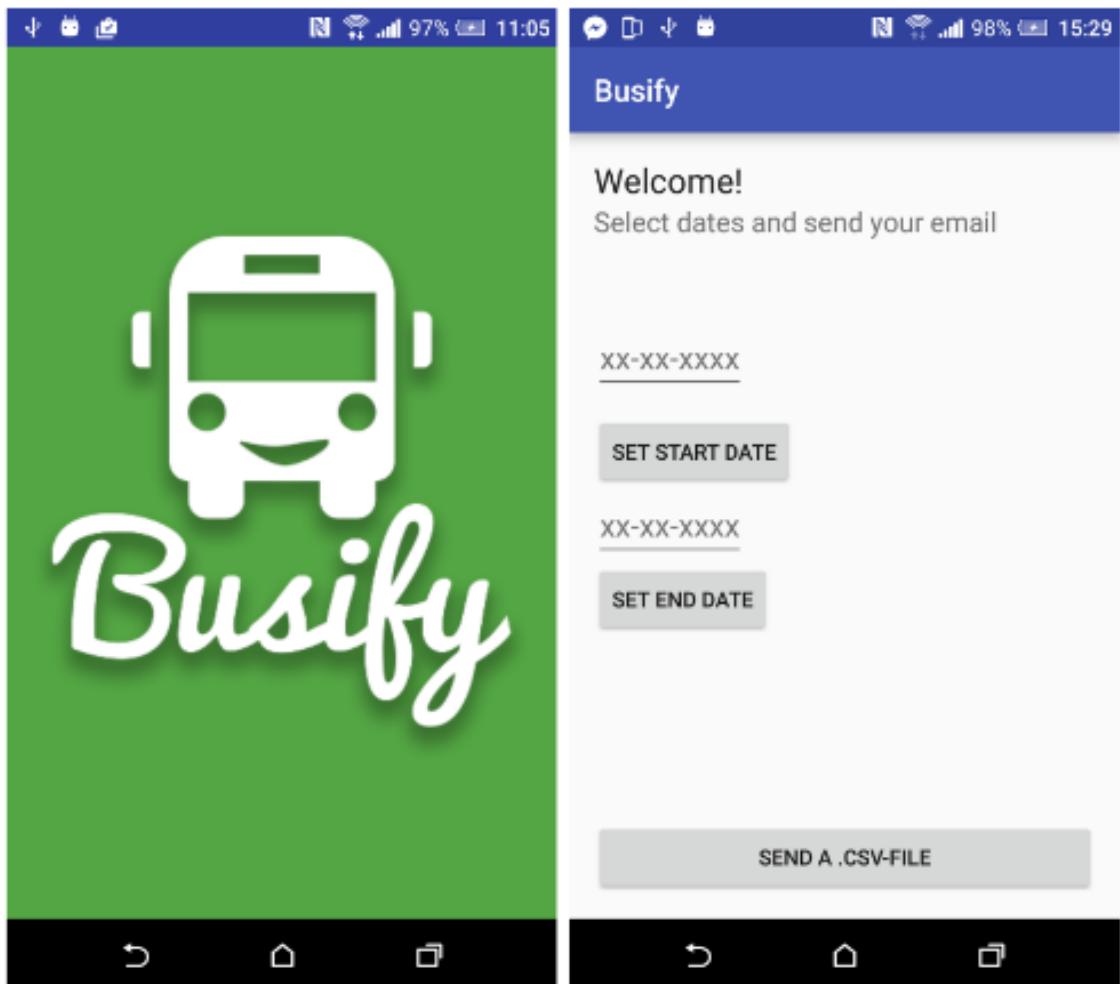
Figur A.2: Gruppens Value Proposition Canvas efter första kundintervjun

## A.2 Sprint 2 (25 april - 2 maj)



Figur A.3: Prototyp för appen. Fr.v: Appens uppstartsvy, huvudvy, och "skickat-email"-vy.

### A.3 Sprint 3 (2 maj - 9 maj)

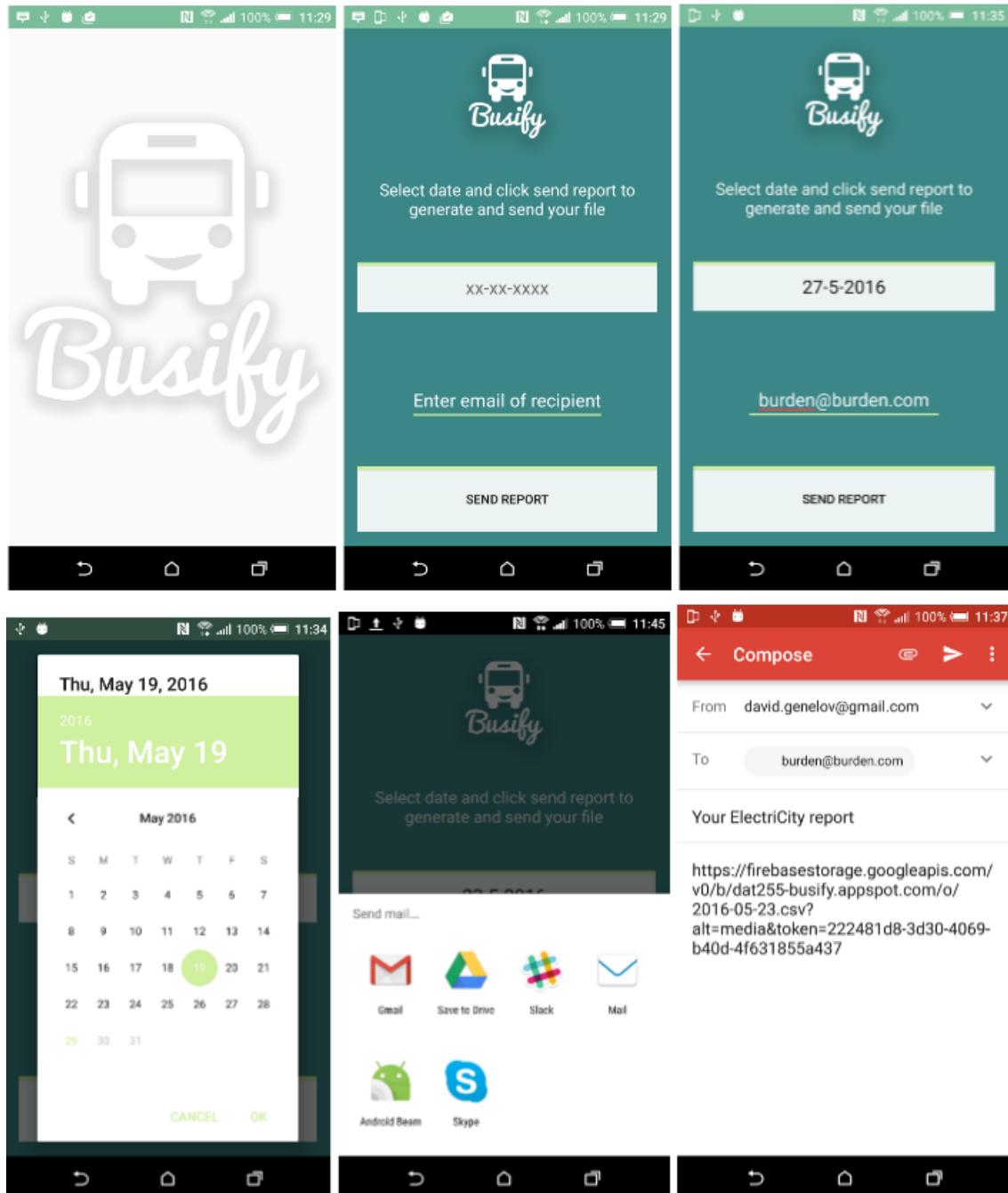


**Figur A.4:** Prototyp för appen. Fr.v: Appens uppstartsvy, huvudvy, och “skickat-email”-vy.

### A.4 Sprint 4 (9 maj - 16 maj)

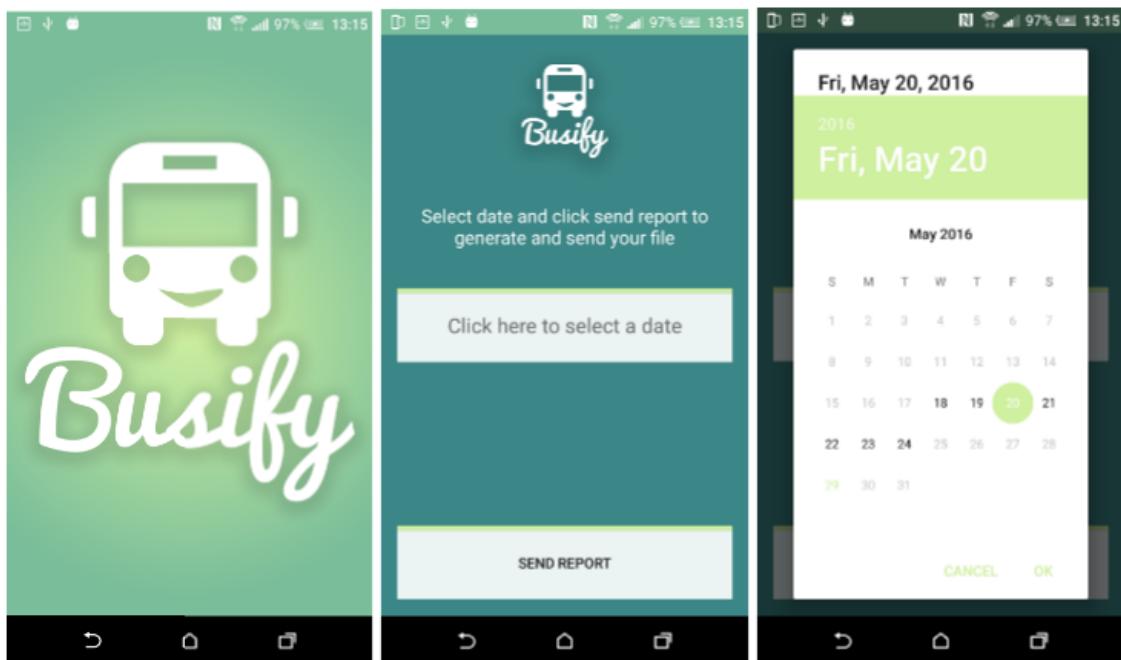
Inga uppdateringar av produktutvecklingen.

## A.5 Sprint 5 (16 maj - 20 maj)



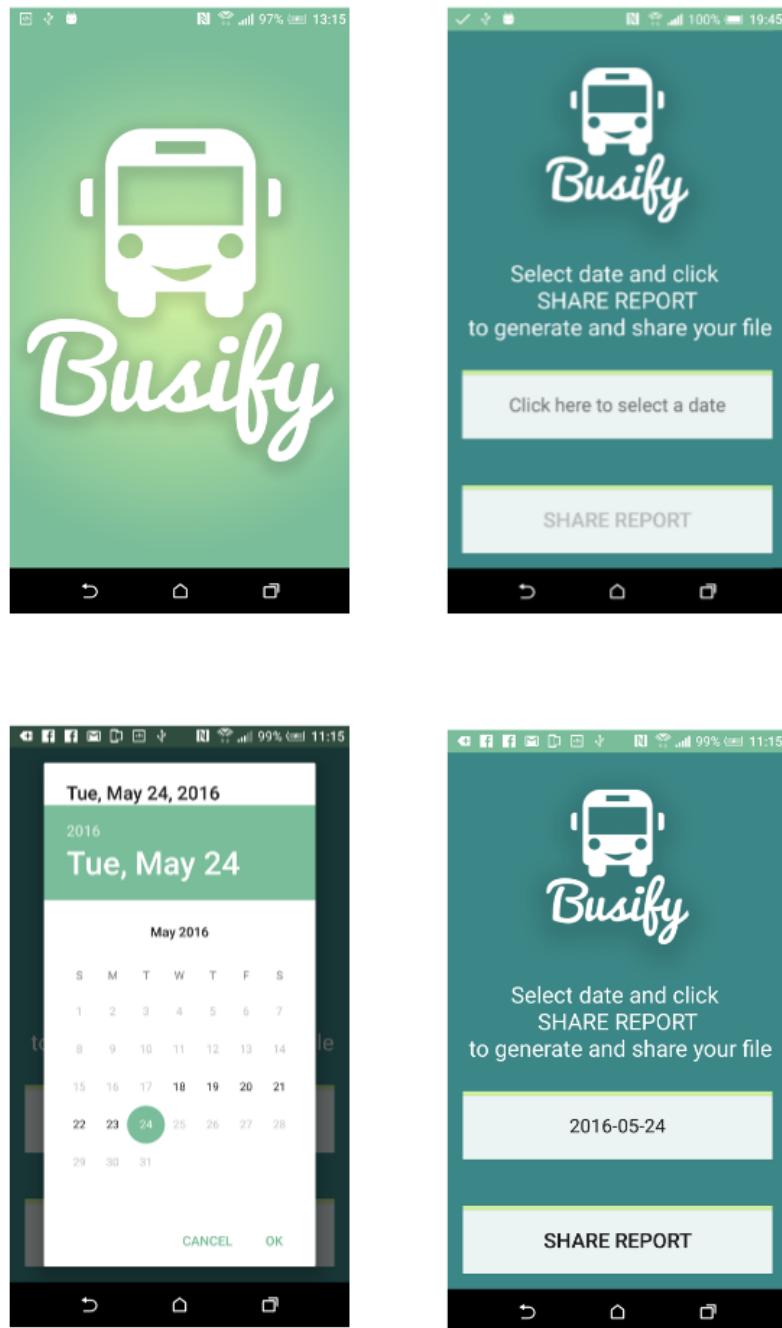
**Figur A.5:** Appens utseende efter sprint 5. Fr.v: Uppstartsvy, ej ifylld huvudvy, ifylld huvudvy; vy för att välja datum; vy för att välja app att dela länk med; utseende på det mail som uppstår då användaren väljer att dela via Gmail.

## A.6 Sprint 6 (20 maj - 27 maj)



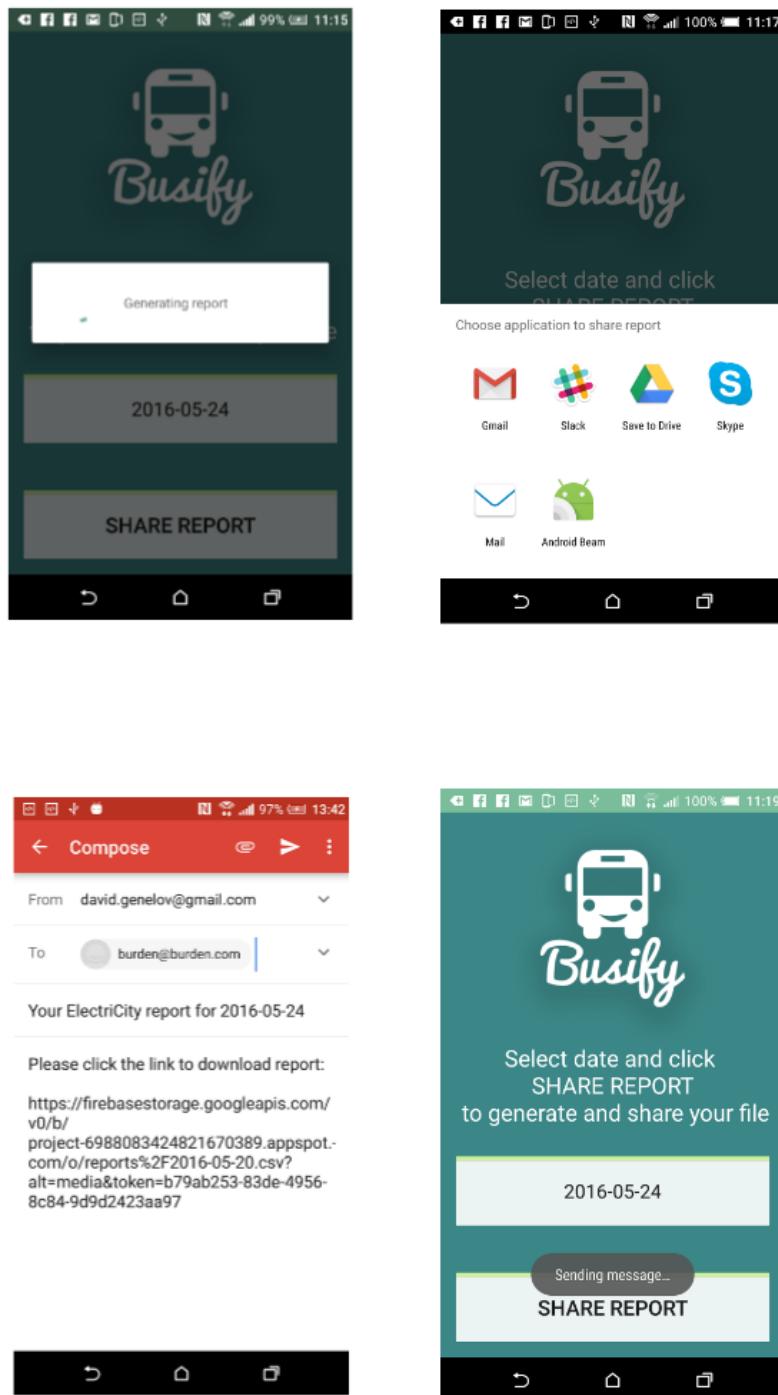
**Figur A.6:** Appens utseende efter sprint 6. Uppstartsvy, huvudvy, och vy för att välja datum. I övrigt ser appen ut som efter Sprint 5.

## A.7 Sprint 7 (27 maj - 01 Juni)



**Figur A.7:** Appens utseende: Uppstartsvy; ej ifylld huvudvy; vy för att välja datum; ifylld huvudvy.

## A. Appendix A - Bilder från produktutvecklingen



**Figur A.8:** Appens utseende: Generating report vy; vy för att välja app att dela länk med; utseende på det mail som uppstår då användaren väljer att dela via Gmail; vy för att visa att filen skickas/delas.



B

# Appendix B - Trellokort

**Scrum Board** DAT255 Public

[Burndown Chart](#)

**LEGEND**

- AKADEMIA
- UTVECKLING

**Chocolate Icebox**

**Product backlog**

**Sprint backlog**

**In Progress**

**Ready to Verify**

**Done** 326

Som en Burden vill jag ha en full reflektionsrapport (D4) efter att projektet är slut för att få en förståelse för projektets arbetsgång och gruppens arbetssätt

1/7 Jun 3 50

Som en Burden vill jag att gruppen lämnar in en kvalitativ kod för att kunna ge gruppen ett bra betyg, varför det är viktigt att programmera efter ett designmönster

0/2 8

A JF MF

Som en kund vill jag att rätt data finns i databasen för samma 7 dagar, för att appen ska vara skalbar.

4/4 8

A JF MF

Som en kund vill jag att data från firebase kan bli en csv, för att läggas på storage.

6/6 21

A JF MF

Som en kund vill jag att det först kontrolleras om det finns en fil i storage innan jag börja quarya data, för annars vill jag att en länk till den befintliga filen skickas med i mejlet.

2/2 5

A JF MF IX

Som kund vill jag ha en färdig produkt, realeasad och klar för att jag ska kunna använda min app.

0/2 5

Som kund vill jag att csvfilen sparas i Firebase, så att den sedan kan

## B. Appendix B - Trello-kort

- Som kund vill jag att all data som ska användas lokaliseras, så jag vet att den går att få tag på och hur det görs

💬 1    3/3   8      A   
- Som en kund vill jag att gruppmedlemmarna ska enas om Android apputveckling, därför att de skall kunna bygga en app

👁️    6/6   8      A         JF   MF   
- Som en kund vill jag att gruppmedlemmarna enar sig om arbetsflöde i Git, därför att de skall kunna bygga en app

👁️   💬 1    7/7   5      A         JF   MF   
- Som kund vill jag att gruppen enas om en back end för att de ska kunna arbeta mot mitt önskade interface enligt value proposition

👁️    1/1   8      A         JF   MF   
- Som kund vill jag att gruppen enas om ett interface för att de ska kunna arbeta mot mitt önskade interface enligt value proposition

👁️    2/2   8      A         JF   MF   
- Som kund vill jag ha ett value proposition med funktionalitet presenterat för mig, därför att jag ska kunna validera eller ge feedback

💬 1    4/4   5
- Som en kund vill jag att alla gruppmedlemmar skall lära sig om arbetsflöde för Scrum, för att de skall kunna jobba effektivt

💬 1    12/12   8
- Som en kund vill jag att arbetsgruppen har gjort en releaseprocess V1 för veckan, så att de olika ingående metoderna och vyerna fungerar ihop och är tillgängliga för mig.

👁️    2/2   5            
- Som grupp vill jag att vi kommer överens om en databas

👁️    3/3    May 11   5      
- Som en kund vill jag att de två fake csv-filerna finns tillgängliga på telefonen för att kunna använda vid testning.

💬 2    2/2   2      A
- Som kund vill jag ha att en enkel uppstartsvy med Busifys logga visas när jag öppnar appen, så att jag vet vad jag öppnat för app.

💬 2    5/5   5      JF
- Som en kund vill jag ha en "hello world" app byggd av gruppen, därför att jag vill se en fungerande app från gruppen

👁️   💬 4    8/8   8         
- Som kund vill jag att alla gruppmedlemmar skall lära sig om arbetsflöde av Android Apputveckling, därför att de skall kunna bygga en app

👁️   💬 1    9/9   5      A         JF   MF   
- Som kund vill jag ha en interface-vy där jag kan välja datum och klicka på en skicka-knapp, för att jag ska kunna interagera med appen.

👁️    6/6   5         



Figur B.3: Våra trello-kort

# C

## Appendix C - Design by Contract

Gruppens Design by Contract skapades med inspiration från definition of done författad av All about Agile <sup>1</sup>.

1. Code produced (all ‘TODO’ items in code completed)
2. Code commented, checked in and run against current version in source control
3. Peer reviewed (or produced with pair programming) and meeting development standards (only when merging)
4. Builds without errors (debug i Android Studio ger inga fel)
5. Unit tests written and passing (when testable)
  - (a) Run something that should work and something that should not
6. Any build, deployment, configuration changes implemented, documented, communicated through Git - push & commit.
7. Relevant documentation/diagrams produced and/or updated (Javadoc or Wiki)
8. Relevant cards on Trello are updated and/or moved

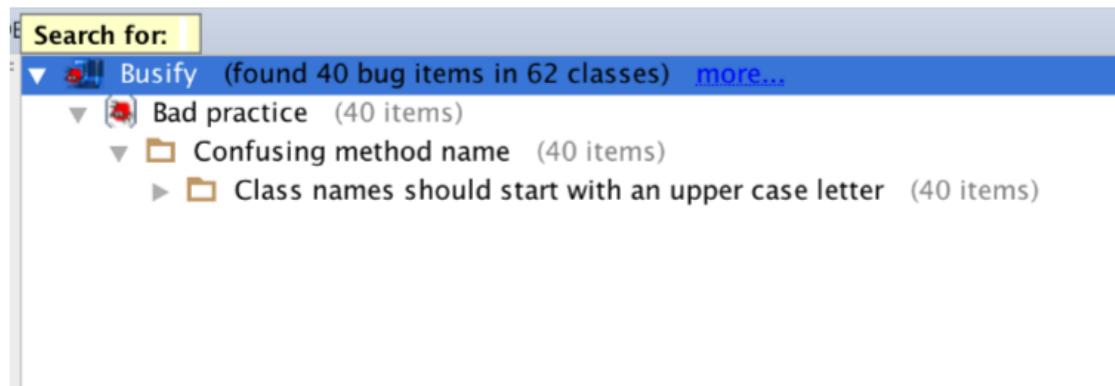
---

<sup>1</sup>Kelly Waters, 2007. All about Agile, <http://www.allaboutagile.com/definition-of-done-10-point-checklist/>. (Hämtad 2016-06-02)

# D

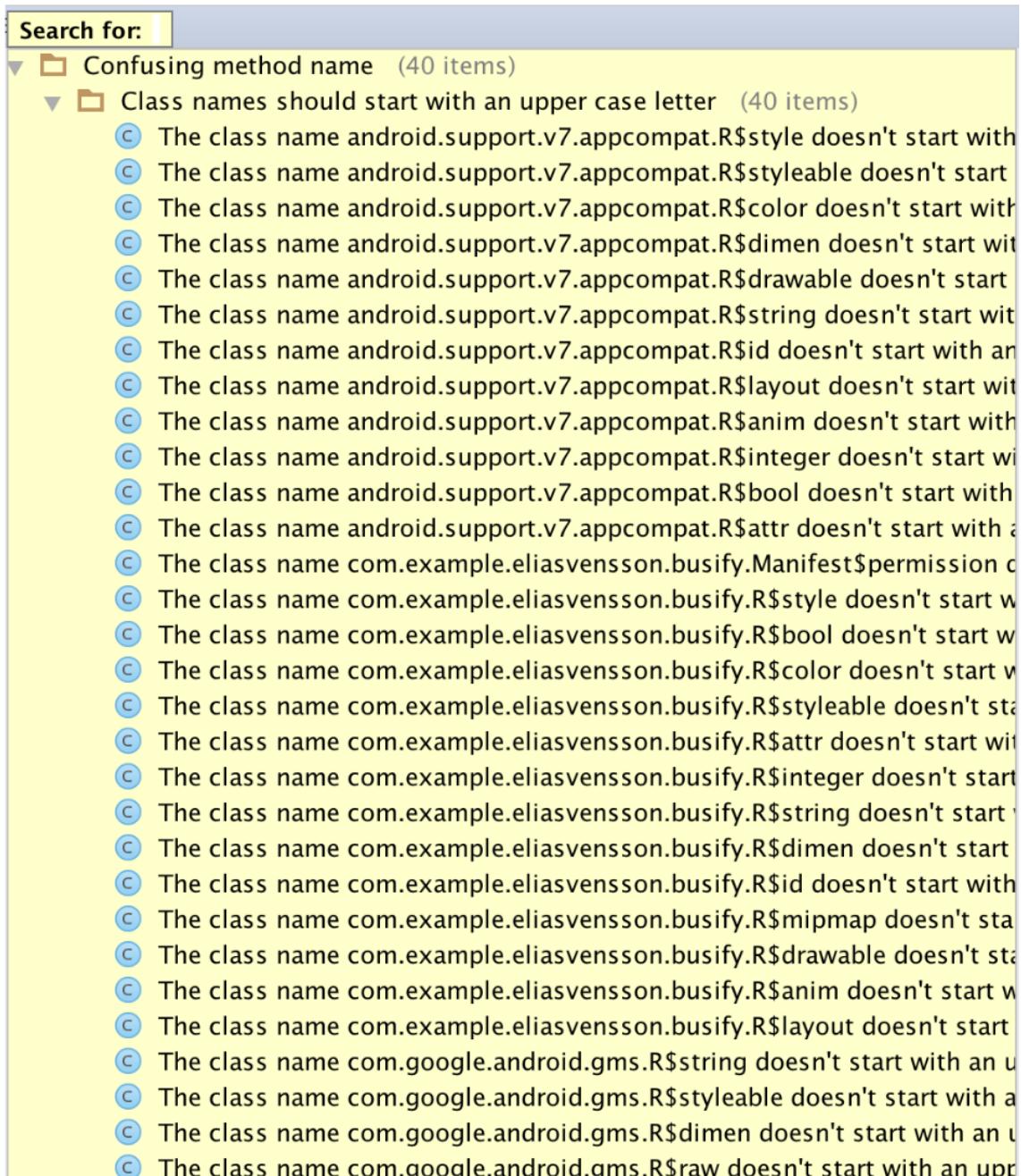
## Appendix D - Findbugs

Ett krav för projektet var att använda FindBugs (<http://findbugs.sourceforge.net/>). Resultatet av detta visas i figuren nedan.



**Figur D.1:** Resultat från FindBugs för den slutgiltiga produkten.

Värt att notera är att samtliga buggar som upptäckts är från Androids standardbibliotek och därmed ingenting som vi hade kunnat undvika. De klasser som vi kodat innehåller alltså inga buggar, vilket styrks av figurerna nedan.



Figur D.2: Första delen av den expanderade FindBugs-rapporten.

- ↳ The class name com.google.android.gms.R\$string doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$styleable doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$dimen doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$raw doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$style doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$layout doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$anim doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$integer doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$bool doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$color doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$drawable doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$attr doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$interpolator doesn't start with an uppercase letter.
- ↳ The class name com.google.android.gms.R\$id doesn't start with an uppercase letter.

**Figur D.3:** Andra delen av den expanderade FindBugs-rapporten.