

CHALMERS



Logga din miljöpåverkan

En androidapplikation för att logga ditt ekologiska fotavtryck

Kandidatarbete inom Data- och informationsteknik

HANNA MATÉRNE
PATRIK KOSKENNIEMI
JONATHAN THUNBERG
FREDRIK LUNDBERG

Chalmers tekniska högskola
Göteborgs universitet
Institutionen för Data- och Informationsteknik
Göteborg, Sverige, Juni 2015

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Logga din miljöpåverkan

En androidapplikation för att logga ditt ekologiska fotavtryck

HANNA MATÉRNE,
PATRIK KOSKENNIEMI,
JONATHAN THUNBERG,
FREDRIK LUNDBERG

© HANNA MATÉRNE, June 2015.

© PATRIK KOSKENNIEMI, June 2015.

© JONATHAN THUNBERG, June 2015.

© FREDRIK LUNDBERG, June 2015.

Examiner: JAN SKANSHOLM

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden June 2015

Förord

Denna rapport är ett kandidatarbete om 15 hp vid Institutionen för Data- och informationsteknik på Chalmers tekniska högskola, som genomfördes våren 2015. Projektgruppen vill rikta ett tack till Alex Gerdes som varit handledare för gruppen.

Abstract

This thesis describes the development of GreenMe, an application allowing users to log their ecological footprint within groceries and transport. The challenge is to know how big the impact on the environment an individual has. The goal of GreenMe is to make the user aware of this impact. To make the footprint accurate it is necessary to log extensive and detailed information. It is therefore mandatory that the application user-friendly and visually appealing. Also an incentive in terms of a competitive element is implemented for the user to actually feel motivated to do something for the environment.

The thesis also describes the implementation of the application, where the main functionality includes logging food that the user has bought and journeys made by the user. Furthermore, the thesis also includes usability tests that we performed and their results. The complexity within the implementation varied, which meant that certain functionality could not be completed, this is discussed in more detail in the chapter of discussion.

Sammanfattning

Denna rapport beskriver utvecklingen av GreenMe, en applikation där användaren kan logga sitt ekologiska fotavtryck inom livsmedel- och transportsektorn. Svårigheten är ofta att veta hur stor påverkan på miljön en individ har. Målet med GreenMe är därför att göra användaren medveten om denna påverkan. För att göra fotavtrycket exakt är det nödvändigt att logga omfattande och detaljerad information. Det är därför viktigt att applikationen är användarvänlig och utseendemässigt tilltalande. Även ett incitament i form av ett tävlingsmoment implementeras för att användaren ska känna motivation till att faktiskt göra något för miljön.

I rapporten beskrivs implementationen av applikationen, där huvudfunktionaliteten innefattar att logga livsmedel som användaren handlat och resor som användaren gjort. Den innefattar också hur vi utförde användbarhetstester och vad resultatet av dessa blev. Implementationsmässigt varierade svårighetsgraden vilket medförde att viss funktionalitet inte kunde slutföras, detta avhandlas mer ingående i diskussionskapitlet.

Nyckelord—ekologiskt fotavtryck, miljö, Android, klimat, hållbar utveckling, utsläpp.

Ordlista

Adapter	Brygga mellan User Interface-komponenter och datakällan som ska fylla User Interface-komponenten med data.
Android Studio	Utvecklingsmiljö med inriktning på Android.
API	<i>Application Programming Interface</i> , gränssnitt mellan ett bibliotek och en applikation.
Backlog	Lista av krav och önskemål för en produkt. Kan ha en sortering beroende på prioritet.
Cirklar i Google+	Google+ användarens kontakter är placerade i olika cirklar som används för att kategorisera kontakterna.
Ekologiskt fotavtryck	Mått på den mängd resurser som en människa förbrukar. Uttrycks i den area som krävs för att försörja denna människa. Mäts i globala hektar (gha).
Google Play	Googles digitala marknad där applikationer, filmer, spel kan köpas eller laddas ner.
GPS	<i>Global Positioning System</i> , ett amerikanskt satellitstyrt system för navigation.
JUnit	Ramverk för att skriva återupprepade tester genom att påstå vissa saker som testerna sedan kollar stämmer..
Latitud	Den vinkel en plats har söder eller norr om jordens ekvator. Skrivs oftast i grader.
Livscykelanalys	Metod för att få en helhetsbild av en produkts miljöpåverkan över hela dess livstid.

Longitud	Den vinkel en plats har öster eller väster om Greenwich, England. Skrivs oftast i grader.
NFC	<i>Near Field Communication</i> , ett sätt att koppla ihop två telefoner med radio, fungerar på väldigt nära håll.
OCR	<i>Optical Character Recognition</i> . Metod för att översätta analoga tecken till en digital motsvarighet.
Peer-to-Peer (P2P)	När två enheter skickar data mellan varandra istället för att gå via en server.
pkm	Personkilometer, mått på det sammanlagda antalet kilometer personerna i ett fordon transporteras.
Polyline	Ett objekt från Googles API som är en linje mellan två bestämda punkter utmålade på en Google map.
Sprint	Beskriver en fixt tidsram då uppgifter skall lösas och ett tydligt mål finns för aktuella sprinten.
SQL	<i>Structured Query Language</i> , ett programmeringsspråk som används för arbete med relationsdatabaser.
SQLite	Är en självbehållen databas som inte kräver konfigurering eller en server.
XML	<i>Extensible Markup Language</i> , markup språk som definierar regler för hur ett dokument ska beskrivas för att vara både människovänlig och maskinvänlig att läsa.

Innehåll

1	Inledning	1
1.1	Syfte	1
1.2	Utmaningar	1
1.3	Avgränsningar	2
2	Teori	3
2.1	Teknisk teori	3
2.1.1	Android	3
2.1.2	Designmönster	4
2.1.3	Activity	5
2.1.4	Fragment	6
2.1.5	Google API	6
2.1.6	Analytics	7
2.1.7	Versionshantering	8
2.1.8	Scrum	10
2.1.9	Användbarhet och användbarhetstester	10
2.2	Miljöaspekten	11
2.2.1	Ekologiskt fotavtryck	11
2.2.2	Privatpersoners klimatpåverkan	11
2.2.3	Rekyleffekten	12
2.2.4	Beteendeförändringar	12
2.3	Liknande koncept	13
2.3.1	Endomondo	13
2.3.2	100Koll	13
2.3.3	ICA hälsorabatt	13
3	Metod och verktyg	14
3.1	Utvecklingsmiljö samt versionshanteringssystem	14
3.2	Arbetsmetod	14
3.3	Programspråk och plattformsversion	15
3.4	Användbarhetstester	16

4	Resultat och implementation	17
4.1	Hem	18
4.2	Livsmedel	18
4.2.1	Inköpslistor	18
4.2.2	Varulista	19
4.2.3	Adaptrar	20
4.2.4	Streckkodsläsning	20
4.2.5	Varudata	20
4.3	Transport	21
4.3.1	Fordonsdata	23
4.4	Topplistor	23
4.4.1	Livsmedel	24
4.4.2	Transport	25
4.5	Statistik	25
4.6	Log in	25
4.7	Databasen	26
4.8	Testning	27
4.9	Resultat av användbarhetstester	27
5	Diskussion	30
5.1	Val av utsläppsområde	30
5.2	Beteendeförändringar och incitament	30
5.3	Livsmedel	31
5.3.1	Inköpslistor	31
5.3.2	Varulista	31
5.3.3	Streckkodskanning	32
5.4	Transport	32
5.4.1	Fordonsvyn	32
5.4.2	Kartvyn	33
5.4.3	Användarens integritet	34
5.5	Statistik	34
5.6	Inloggning	34
5.7	Databasen	35
5.8	Fusk	35
5.9	Problematik under implementationen	36
5.9.1	OCR - Optical Character Recognition	36
5.9.2	Tävlingsmoment	36
5.9.3	Analytics	37
5.9.4	Digitala kvitton	37
6	Slutsats	38
	Litteraturförteckning	39

Bilaga A	Användbarhetstester	I
Bilaga B	Resultat användbarhetstester	III
Bilaga C	Flödesdiagram för transportsektorn	VIII
Bilaga D	Använda bibliotek	IX
D.1	AndroidPlot	IX
D.2	AppCompat	IX
D.3	Espresso och Testbibliotek	IX
D.4	Google Play Services	IX
D.5	RecyclerView	IX
D.6	SQLiteAssetHelper	X
D.7	v4 Support Library	X
D.8	ZXing Barcode Scanner	X

Kapitel 1

Inledning

Klimatfrågan prioriteras allt högre i dagens samhälle. Det blir därför allt viktigare att vara klimatsmart. Genom att göra små förändringar i vardagen kan det ge stora resultat över tid. Det finns många som vill göra rätt för sig men, samtidigt många som har svårt att se sin egen vinning i att ändra sitt beteende. Det har även visat sig att många har kunskapen som krävs för att leva mer miljöanpassat men denna teoretiska kunskap omsätts vanligtvis inte i praktiken. I de allra flesta fallen handlar det om att förändringarna är för krävande för användaren. [5]

Idag används många applikationer för att till exempel se hur mycket och vad du äter eller att se hur mycket motion du får dagligen. Detta kandidatarbete resulterade i en liknande applikation där det istället är användarens ekologiska fotavtryck som är i fokus. I dagsläget är det genomsnittliga ekologiska fotavtrycket 6,4 globala hektar i Sverige [54]. För att det ska vara hållbart behöver det ekologiska fotavtrycket vara mindre än 1,7 globala hektar [24]. Det är tydligt att det krävs en stor minskning för att åstadkomma en sådan nivå. Resultatet av kandidatarbetet, i form av en androidapplikation, ska uppmuntra människor att kunna göra smarta val.

1.1 Syfte

Syftet med projektet är att bygga en Androidapplikation som loggar användarens ekologiska fotavtryck i vardagen. Prototypen ska skapa en ökad miljömedvetenhet genom användarvänliga funktioner, främst inom livsmedel- och transportsektorn. Den ska även innehålla ett bra incitament för att motivera användaren till att utnyttja applikationen där ett smidigt grafiskt användargränssnitt är en del i detta. Användaren ska även kunna följa sina framsteg under ett givet tidsspann.

1.2 Utmaningar

Människor påverkar miljön på många olika sätt. En stor del av denna miljöpåverkan har sin grund i människors beteenden. Problemet ligger ofta, när det handlar om miljö, i att oron för miljön inte är tillräckligt stark för att offra sin livsstil [12]. För att det ska bli möjligt att uppnå ett hållbart samhälle krävs beteendeförändringar vad gäller enskilda individers handlingar i vardagen. Svårighet är att kunna förändra dessa ohållbara beteenden [51]. Enligt Dobson,

som är professor i miljöpolitisk teori, måste förändringar ske genom att individen själv får motivation till att förändra sitt beteende [46]. En medvetenhet om konsekvenser av enskilda handlingar samt en känsla av skyldighet att minska sitt ekologiska fotavtryck måste finnas. Detta är något som måste tas hänsyn till under utvecklingen av applikationen.

En av de största utmaningarna med detta kandidatarbete är att få användarna av applikationen att faktiskt börja samt fortsätta använda applikationen i syfte att minska sitt ekologiska fotavtryck. Det krävs därför att applikationen har hög användbarhet och är enkelt utformad.

En teknisk utmaning är att få en komplett och användbar databas med varje livsmedel som finns på den svenska marknaden med dess motsvarande koldioxidvärde.

1.3 Avgränsningar

Kandidatarbetet är ett tidsbegränsat projekt. Detta medförde att i ett tidigt skede beslutades att utvecklingen enbart skulle ske för Androidenheter. Det ansågs rimligt att utveckla för enbart en plattform och eftersom alla medlemmar av projektgruppen tidigare läst kurser i Java, som är utvecklingsspråket för Androidapplikationer, blev det ett självklart val.

Applikationen ska "logga ditt ekologiskt fotavtryck" vilket är en mycket bred beskrivning och kan göras hur omfattande som helst. En begränsning inom området var därför tvunget att göras. Det togs ett beslut om att fokusera på två huvudområden, livsmedel och transport, för att hinna arbeta med dessa och inte behöva kompromissa med funktionaliteten.

Med anledning av att vissa bibliotek som används inte stödjer telefoner med äldre operativsystem än Android v4.0.3 Ice Cream Sandwich är detta en begränsning då det är krav för att kunna använda applikationen.

Applikationen är begränsad till att bara fungera för det svenska språket. Men för konventions skull och om någon i ett senare skede ska ta över applikationen för att fortsätta utvecklingen är alla kommentarer och kod skriven på engelska. Mycket av applikationen hänger på att användaren är ärlig och loggar sanningsenliga siffror, med den tidsbegränsning som fanns valdes det nämligen att inte programmera in några extra kontroller som hanterar "fusk".

Kapitel 2

Teori

I detta kapitel beskrivs den tekniska teorin som används i applikationen. Därefter kopplas centrala begrepp till miljöaspekten och hur beteendeförändringar kopplade till miljön sker. Även liknande koncept och applikationer som syftar till att förändra beteendet hos användaren presenteras.

2.1 Teknisk teori

För att utveckla en Androidapplikation krävs en viss teknisk bakgrund. Följande avsnitt ger en grundläggande fakta kring detta.

2.1.1 Android

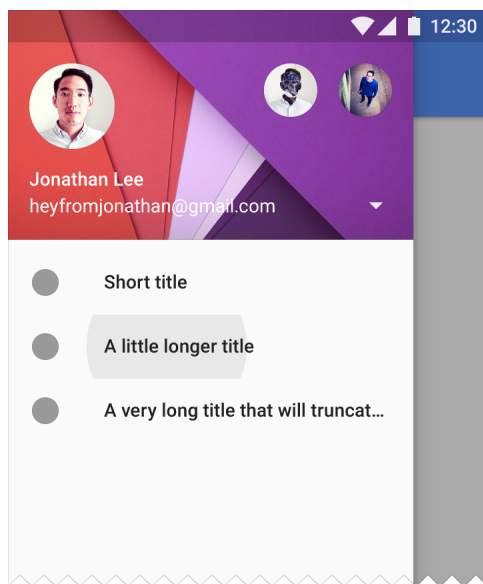
Android är ett mobilt operativsystem utvecklat av Google. Tabell 2.1 visar att Android har störst marknadsandel och samtidigt ökar [22]. I kommande avsnitt beskrivs ett designförslag från Google samt två grundstenar i Androidapplikationer.

Tabell 2.1: Marknadsandelar för smarttelefons plattformar

Smarttelefon Plattform	Andel Okt/2014 (%)	Andel Jan/2015 (%)	Förändring (%)
Android	52.3	53.2	0.9
Apple iOS	41.9	41.3	-0.6
Microsoft	3.5	3.6	0.1
BlackBerry	2.1	1.8	-0.3
Symbian	0.1	0.1	0.0

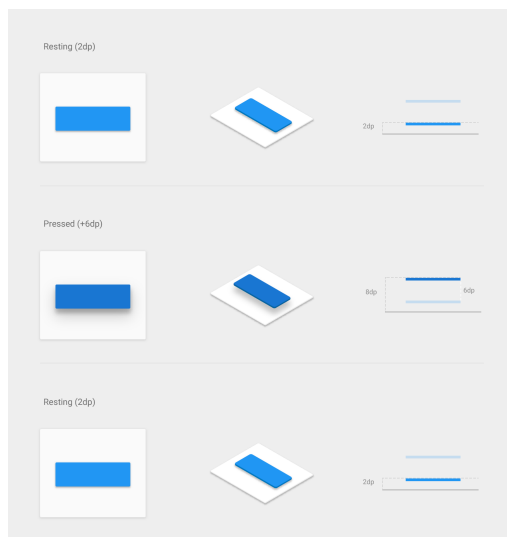
2.1.2 Designmönster

Google ger utvecklare chansen att skapa applikationer till deras produkter och för att få en grundstruktur hur en applikation ska se ut behövs designmönster och riktlinjer. Det är inget krav att följa dessa riktlinjer, men underlättar för användaren om denne kan känna igen sig när byten mellan olika applikationer sker. Google har tagit fram en design som kallas för Material Design [33]. Dessa riktlinjer föreslår bland annat hur en virtuell 3D-känsla kan skapas vid användning av applikationen samt hur och var element borde placeras samt utformas. Att ha en meny som dras fram från kanten, ska fokus ligga på innehållet i menyn och resten av den synliga applikationen dimmas ner fram tills menyn stängs, se figur 2.1.



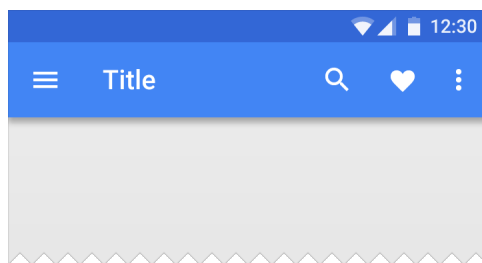
Figur 2.1: Designmönster för utdragbarmeny [1]

Som nämnt i föregående stycke är syftet med Material Design att få en 3D-känsla. Detta skapas genom skuggor på olika element där mängden skugga simulerar en distans som elementet höjs upp, det ger en illusion av att elementet sticker ut ur skärmen, se figur 2.2.



Figur 2.2: Illustration över hur skuggans mängd påverkar den visuella uppfattningen av distansen till bakgrunden [2]

Figur 2.3 visar hur den övre delen av applikation bör utformas för att följa Material Design. Detta designmönster kallas Action Bar Design Pattern och kan beskrivas som att applikationen hela tiden ska visa ett fält högst upp på skärmen vilket ska ge användaren information om vilken sida som visas just nu och tillgängliga alternativ genom knapparna på högersidan. [13]



Figur 2.3: Ett exempel på en vanlig framsida med tre streck som indikator för dold meny [3]

2.1.3 Activity

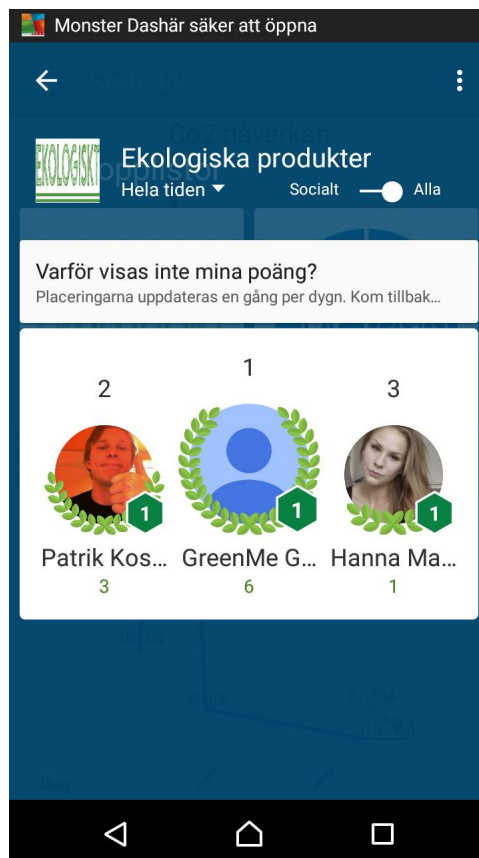
En Activity är ett objekt som används högt upp i applikationstrukturen. Det är detta objekt som operativsystemet kallar på och kör metoder som finns i den. När användaren växlar mellan olika applikationer växlas egentligen Activity. En applikation kan även ha flera Activities och kan ses som ett nytt fönster när det växlas inom applikationen [14]. När användaren växlar mellan olika Activity syns det tydligt när ett nytt fönster öppnas eftersom det nya läggs över den gamla. När detta sker startas en ny process [36].

2.1.4 Fragment

Fragment är objekt som måste placeras i en Activity. Vanligen används Fragments för att byta ut innehåll eller data inuti en Activity när användaren gör ett val i användargränssnittet. Ett exempel är menyn, där en huvud-Activity används och flera Fragments som visar olika sidor beroende på vad användaren väljer i menyn [27]. Fördelen att använda Fragments i en Activity för att växla mellan innehåll är att det ger en mer dynamisk känsla eftersom inget nytt fönster öppnas och informationen visas direkt för användaren. Då bytet mellan olika Fragments sker i en Activity används den aktuella Activity tråden för Fragments, då skapas inte några nya trådar vid bytet [36].

2.1.5 Google API

Google tillhandahåller en uppsättning av APIer som tillåter tredjepartsapplikationer att utnyttja deras tjänster så som inloggning genom Google+ [30] eller tillgång till Google Maps [31]. Med hjälp av APIerna kan utvecklarna undvika att hantera till exempel säkerhetsfrågor kring användaruppgifter eller behöva tänka på hur buffringen kring kartfönstret går till eftersom Google tar hand om det. Det blir en högre nivå på programmeringen eftersom utvecklaren använder många färdiga funktioner. Det finns APIer för allt från Google Drive till Google Analytics och mycket däremellan.



Figur 2.4: Topplistan för ekologiska produkter där topp tre syns under vald tidsperiod och även vilka som ska jämföras med, antingen alla eller dina vänner

Google Play Service är ett av Googles APIer som omfattar deras olika spelfunktioner som exempelvis turbaserat spelläge (eng. turn-based) eller topplistor. Just topplistor är utformade med två olika reglage som innehåller tre respektive två val. I figur 2.4 syns till vänster ett reglage där det går att ställa in vilken tidshorisont som kan användas, det går att välja mellan dagligen, veckovis eller total poäng. Det högra reglaget avser huruvida användaren enbart vill visa sina vänners resultat eller visa den globala topplistan för alla användare. Funktionalitetsmässigt uppdateras inte listan om resultatet som skickas till den är sämre än det som redan är registrerat.

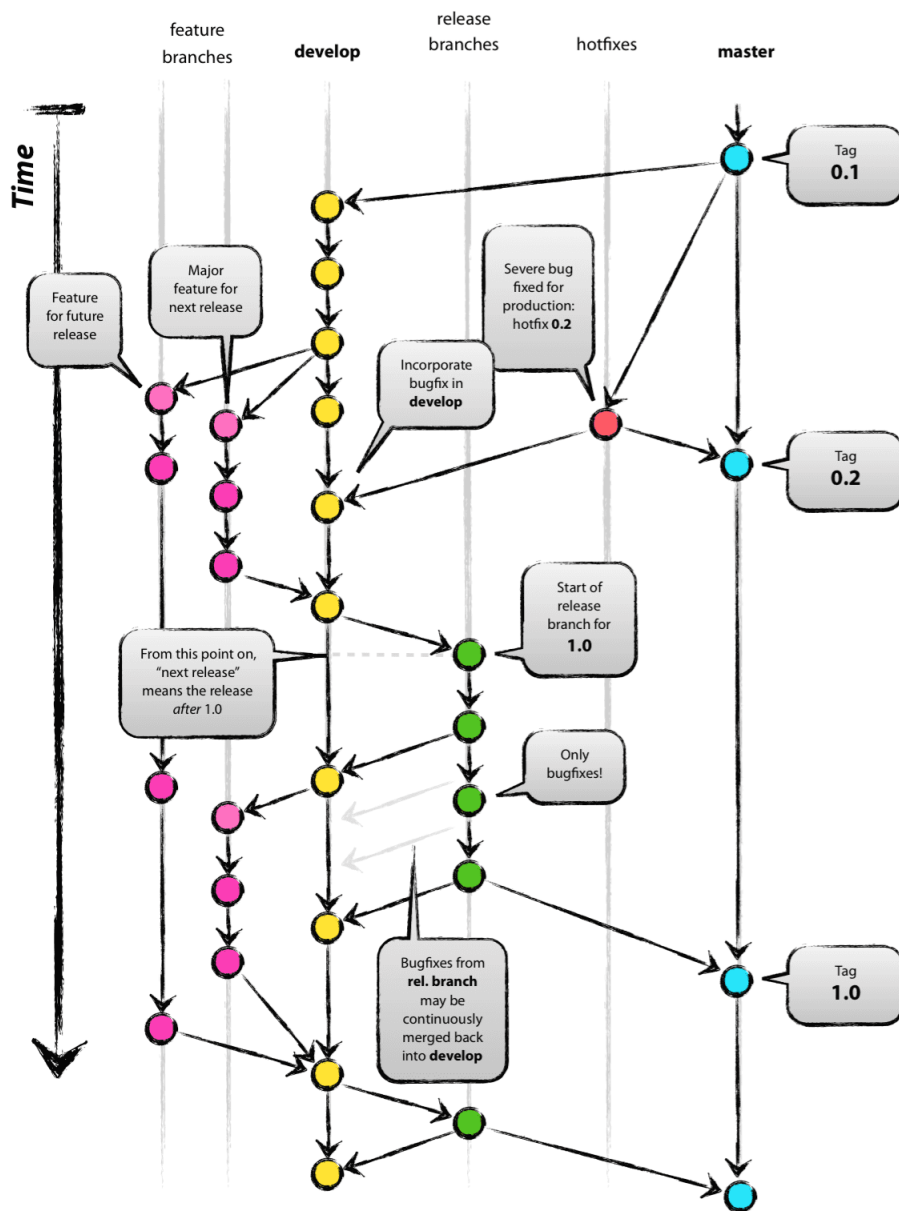
2.1.6 Analytics

Google Analytics är ett verktyg för att ge utvecklaren insikt i hur användaren faktiskt använder en webbsida eller applikation. Det går att logga när användaren gör vissa saker eller få ut värden och sedan sammanställa detta i grafer som utvecklaren sedan kan studera, i syfte att förbättra applikationen [29].

2.1.7 Versionshantering

När flera olika personer är delaktiga i samma projekt, likt kandidatarbetet, och arbetar med samma kod uppstår flera parallella versioner. Då behövs ett program som kan hantera detta, vilket kallas versionshantering. Fördelen med att använda exempelvis Git [28] är att delning av kod hanteras effektivt men även parallella utvecklingsspår är möjliga. Parallella utvecklingsspår gör att ny funktionalitet hela tiden kan implementeras men samtidigt ha en stabil och körbar version. Det gör inte heller något om ett problem uppstår under utvecklingen då det alltid går att återställa till senast körbara versionen.

För att kunna få en bra struktur och ett bra arbetssätt när Git används finns en metod som heter Git workflow [7]. Som illustreras i figur 2.5 kan arbetssättet Git workflow delas upp i olika stadier i applikationens cykel i grenar.



Figur 2.5: Arbetsflöde genom användning av Git workflow [47]

Git workflow är ett sätt att arbeta med grenar. Istället för att enbart använda en huvudgren är två grenar i fokus. En huvudgren och en utvecklingsgren som fungerar som en integrationsgren för funktioner. Varje ny funktionalitet skapas i en egen gren och när den anses färdig sammanfogas den med utvecklingsgrenen. När utvecklingsgrenen har tillräckligt med funktionalitet för att släppas skapas en tillfällig gren kallad leverans dit utvecklingsgrenen sammanfogas. Ingen ny funktionalitet läggs därefter till, enbart programkorrigeringar och dokumentation får göras. En sammanfogning av leveransgrenen till huvudgrenen sker och ett versionsnummer sätts, vilket medför att huvudgrenen blir en följd av leveransgrenar.

Därefter görs en sammanfogning från leveransgrenen till utvecklingsgrenen innan leveransgrenen tas bort, eftersom en ny iteration ska påbörjas och bygga på den äldre leveransen. Det skall således inte göras några stora ändringar direkt mot huvudgrenen. Finns det ett behov av en liten ändring, bör en ny gren av huvudgrenen göras och namnges till snabblösning. Därefter görs en sammanfogning till huvudgrenen och snabblösningsgrenen tas bort. [21]

2.1.8 Scrum

Inom mjukvaruutveckling finns ett agilt ramverk kallat Scrum [9]. Genom att använda Scrum kan en struktur byggas upp där alla enkelt kan få en överblick av projektets status samt uppdelning av arbetsflödet i sprintar och delmål. Det skall även finnas en tydlig produktägare som styr och bestämmer vad som skall göras vid varje sprint. Möten bör även hållas ofta för att hålla gruppen uppdaterad, där ska även alla berätta vad som har gjorts och vad som ska ske därefter, dessa möten brukar kallas för Daily Standup.

Genom att ha en backlog för hela produkten eller för varje sprint är detta ett sätt för produktägaren att lista allt som ska vara med i produkten och även sätta en prioritet på dessa. Listorna är aldrig klara utan kan uppdateras hela tiden beroende på projektets aktuella status. I en backlog ska det även finnas kravspecifikationer som sätts upp av produktägaren, dessa beskriver hur en viss funktionalitet ska vara utformad och används till hjälp för utvecklare.

2.1.9 Användbarhet och användbarhetstester

Användbarhet handlar om att säkerställa att applikationen är ändamålsenlig, effektiv samt tillfredsställande att använda. Frågor som är viktiga att besvara är: "Uppnår användaren det de behöver genom användning av applikationen?", "Hur lång tid tar det för användaren att utföra uppgifter?" samt "Hur upplever användaren interaktionen med applikationen?". Vidare är användarupplevelsen något som bygger vidare på hur tillfredsställande applikationen är att använda. Detta gör att användarens uppfattningar, känslor, prestationer samt psykologiska reaktioner som uppkommer före, under samt efter användning inkluderas. [20] Sammanfattningsvis är det viktigt att ta i beaktande vilka som använder applikationen, i vilken kontext och om applikationen stödjer vad användaren vill utföra. Men det är även viktigt att undersöka vilken inställning användaren har till applikationen.

Användbarhetstestning används i syfte att utvärdera och testa hur applikationen fungerar i praktiken genom tester utförda av personer inom den tänkta målgruppen. Detta ger underlag för att kunna utveckla samt förbättra applikationen vad gäller användbarhet. Användaren som också är testaren, får utföra vanligt inträffande uppgifter och utvecklarna har då möjlighet att observera vad som fungerar bra respektive dåligt. Uppgifter där sannolikheten att användbarhetsproblem uppstår är också viktiga att göra. Det är även viktigt att testa applikationen i den miljö som den faktiskt används i, det är där problemen uppstår. Det kan tilläggas att människor oftast inte tänker högt när de utför uppgifter men under testet kan testledaren instruera testarna att tänka högt [53]. Detta är särskilt användbart när testaren fastnar i olika

uppgifter. Det är då enklare att förstå var problemet ligger, eftersom testarens reaktioner är ett steg i detta.

2.2 Miljöaspekten

För att kunna förstå mer om hur människan påverkar miljön finns det en del information och uttryck som behövs kunskap inom. Dessa beskrivs kort i detta avsnitt.

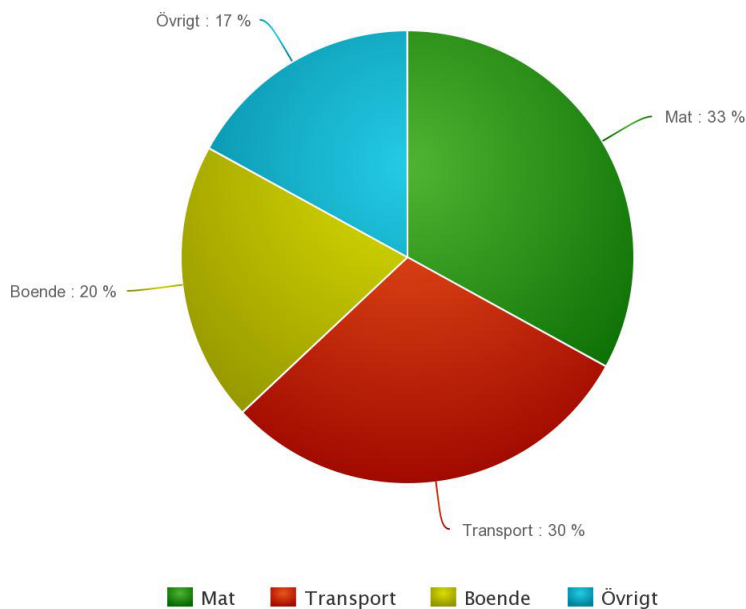
2.2.1 Ekologiskt fotavtryck

Allt människor konsumerar påverkar miljö i någon mån. All konsumtion och produktion kräver utrymme. Ett mått på hur stor biologiskt produktiv yta som behövs för att möta denna konsumtion och absorbera avfallet som tillkommer kallas ekologiskt fotavtryck. I detta mått ingår alla ytor på jorden som går till exempelvis byggnader samt att framställa mat, men också användning av fossila bränslen. För att räkna ut ytan för fossila bränslen estimeras hur stor yta nyplanterad skog som krävs för att kompensera för utsläppen [48]. Enheten för ekologiska fotavtryck är globala hektar (gha).

2.2.2 Privatpersoners klimatpåverkan

Klimatpåverkan för en privatperson kan delas in i fyra större sektorer. Dessa sektorer är boende, mat, transport och övrigt. Sektorn mat inkluderar klimatpåverkan för livsmedel, från åker till affär. Denna sektor inkluderar dock inte tillagningsprocessen eller resor som en privatperson gör för att frakta livsmedel. Sektorn transport inkluderar alla resor som en privatperson gör. Sektorn boende inkluderar ett stort spann, från värme till vatten i hemmet. Övrigt-sektorn inkluderar allt som inte ingår i någon av de andra sektorerna, det vill säga allt från kläder till elektronik. Nedanstående figur 2.6 beskriver fördelningen av koldioxidutsläpp mellan sektorerna procentuellt för genomsnittet av personerna som var med i Naturvårdsverkets undersökning år 2012.

Som det syns i figuren står mat för 33 procent, transport för 30 procent, boende för 20 procent och övrigt för 17 procent [41].



Figur 2.6: Uppdelningen av klimatpåverkan för en privatperson

2.2.3 Rekyleffekten

Rekyleffekten är ett begrepp som används för att beskriva hur en konsument genom effektivare teknik eller byte av vanor kan minska sin miljöpåverkan men i slutändan få en högre miljöpåverkan [6]. Ett klassiskt exempel är att när bilarna blir mer energisnåla blir det billigare för konsumenten att köra vilket medför att konsumenten kan köra mer för samma kostnad och i slutändan sannolikt släpper ut mer. Detta kallas direkt rekyleffekt. Det finns även något som kallas för sekundär rekyleffekt. Det innebär att en minskad kostnad inom ett område ger en ökad konsumtion inom ett annat område. Ett exempel kan vara att en familj börjar använda en mer energisnål bil. Detta leder till att de kan lägga undan pengar som sedan läggs på en resa till exempelvis Thailand, vilket medför att den totala koldioxidförbrukningen förblir oförändrad [44].

2.2.4 Beteendeförändringar

I avsnitt 2.3 beskrivs flera olika exempel på applikationer som alla försöker få användare att ändra vanor. Det blir problematiskt när det kommer till miljö eftersom alla människor prioriterar och värderar olika. Värden är en viktig beståndsdel för motivationen. Även attityder, som är en sammanfattning av positiva och negativa värderingar av något, påverkar motivationen. Således för att kunna ändra beteenden måste personen i fråga först bli medveten om sina vanor och bara då är det möjligt att bryta dessa. Enligt forskaren Cecilia Bergstad går det att påverka gynnsam miljö genom att göra gröna beteenden mer lönsamma eller lätta och roliga [43].

2.3 Liknande koncept

Detta avsnitt nämner några applikationer liknande GreenMe, där syftet med applikationerna är att förändra beteenden hos användaren.

2.3.1 Endomondo

Endomondo är en applikation för att motivera människor till att träna genom att göra träningen social och rolig. Motivationen kan till exempel få genom att utmana sina vänner [25]. Applikationen följer dina träningspass och tillhandahåller information om hur användaren ska nå sina mål [26].

2.3.2 100Koll

Applikationen 100Koll visar din elförbrukning i realtid. När användaren kan se mängden energi som faktiskt förbrukas blir det enklare att minska elförbrukningen. Applikationen gör det även möjligt att styra sitt hem genom en smartphone [34].

2.3.3 ICA hälsorabatt

ICA vill uppmuntra människor till mer vardagsmotion genom sin nya applikation som mäter hur mycket användaren rör på sig och ger rabatt baserat på antalet kilometer avverkade [37]. Den utnyttjar två olika tekniker beroende på vilken telefon användaren har. Om användaren har en Iphone 5s eller nyare utnyttjas telefonens inbyggda aktivitetsmätare för att mäta sträckan. Används en äldre telefon eller en Androidtelefon utnyttjar ICAs applikation en annan applikation som heter Runkeeper. Runkeeper är en träningsapp som främst används för att logga sina träningspass men kan även användas vid vardagsmotion. "ICA hälsorabatt" hämtar sedan datan som Runkeeper samlar in, vilket inte gäller manuellt inlagd data, och använder den i applikationen.

Kapitel 3

Metod och verktyg

Den övergripande planeringen för projektet tog form genom en planeringsrapport. Projektet delades därefter in i faser; förstudie, implementeringsfas samt testning. Kapitlet i sin helhet ämnar att ge en övergripande uppfattning av hur projektet genomfördes.

3.1 Utvecklingsmiljö samt versionshanteringssystem

Under utvecklingen av applikationen användes programvaran Android Studio. Den är designad och anpassad av Google specifikt för Androidutveckling [18]. Genom att använda versionshanteringssystemet Git som nämns i föregående kapitel blir det enklare för flera utvecklare att arbeta på samma kod. Det bestämdes att följa en modell som heter Git workflow vilket beskrivs mer i teoriavsnittet och att alla utvecklare skulle döpa sina grenar utefter en viss mall baserat på vad grenen har för kod i sig samt vem som har skrivit den.

3.2 Arbetsmetod

Projektet genomfördes med hjälp av Scrum över tre sprintar, pre-alpha, alpha samt beta. Varje sprint innebar en avgörande funktion för applikationen där pre-alpha var implementeringen av backend samt ett skelett för applikationen, alpha var implementationen av livsmedelsdelens funktioner och beta avsåg transportdelens funktioner. Sprintarnas längd varierade beroende på förväntad arbetsbelastning. Innan en ny sprint påbörjades genomfördes ett planeringsmöte för att bestämma vad som skulle utföras samt för att få en uppfattning om kommande sprint. För att kunna strukturera upp önskvärd funktionalitet utnyttjades ett verktyg som kallas Pivotal Tracker [35]. Detta för att snabbt kunna få en överblick över vad som behövde utföras och vad som var färdigställt. Varje vecka hölls möten, dock inte dagligen, för att stämma av vad som utförts den senaste veckan, vad som behövde göras kommande vecka samt vad som fungerat bra respektive dåligt. Efter en avslutad sprint demonstrerades implementerade användarberättelser.

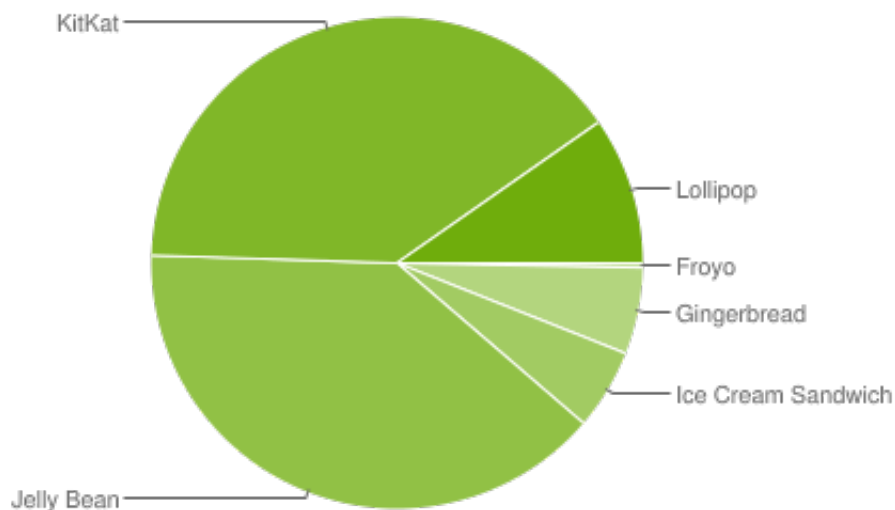
3.3 Programspråk och plattformsversion

Java är språket som Google rekommenderar vid utveckling av Androidapplikationer. För att implementera hela modellen, en del av vyn samt den del av applikationen som bearbetar händelser och framkallar modifieringar användes just Java. Även XML användes för att utveckla den största delen av vyn.

Databasen är skapad i SQLite, vilket är den databashanterare som Google rekommenderar för Androidutveckling [15]. I utvecklingen har en blandning av de hjälpmetoder som finns i SQLiteAssetHelper använts och något som heter rawQuery som är en metod i samma klass. RawQuery har använts i de fall som större precision över SQL-anropen har behövts.

För att ge användaren en grunddatabas att använda i applikationen utnyttjades SQLiteAssetHelper för att initiera databasen med koldioxidvärden för produkter. Detta innebär att databasen till en början inte är tom.

Applikationen är utvecklad mot en av de nyare versionerna av Android, 4.4 Kitkat med API nummer 19. Lägsta versionen som applikationen klarar av är 4.0 med hjälp av stödpaketet AppCompat, som beskrivs i Bilaga D. Detta stödpaket lägger till eventuella klasser som saknas i de lägre versionerna. Version 4.4 är starkt representerad på den splittrade versionskartan och har för tillfället en marknadsandel på 39.8% av det totala antalet Androidenheter, se figur 3.1 [23].



Figur 3.1: Marknadsandelar mellan Androidversioner [4]

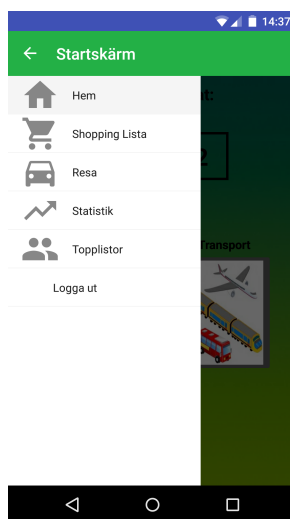
3.4 Användbarhetstester

Användbarhetstester av applikationen utfördes på 20 personer efter att implementeringsfasen var slutförd. Tester gjordes i syfte att upptäcka problem som användaren ställdes inför och då få möjligheten att kunna åtgärda dessa. Testerna gjordes i form av intervjuer, där testaren fick flera korta uppgifter samt följdfrågor som rörde funktionalitet och design. Testaren uppmanades att tänka högt, som beskrivs närmare i avsnitt 2.1.9. I bilaga A finns det testformulär som användes för att genomföra intervjuerna och avsnitt 4.9 redovisar resultatet.

Kapitel 4

Resultat och implementation

Följande kapitel beskriver hur applikationens funktioner implementerats där det överordnade skalet av applikationen består av två aktiviteter. Den första är inloggningsskärmen som användaren möter första gången applikationen startas, den andra aktiviteten används som en ram för resten av applikationen och fristående innehåller den enbart en navigationsmeny samt ett överskriftsfält, även kallat Action Bar, se figur 4.1. Navigationsmenyn används för att navigera sig runt i applikationen och finns med i alla vyer även om den per standard är minimerad. Varje vy byggs genom att inuti aktiviteten placera ett fragment innehållande just de önskade funktionerna och när användaren vill byta vy byts fragmentet ut mot ett nytt. Det medför att samma aktivitet hela tiden används men innehållet byts ut beroende på var i applikationen användaren befinner sig. Resten av kapitlet beskriver de olika fragmentens uppbyggnad mer i detalj, vilken Androidversion som använts, aktiviteten “Log in” och till sist finns det även ett avsnitt som beskriver testningen.



Figur 4.1: Menyn öppnas genom att dra från vänsterkanten inåt eller genom att trycka på de tre strecken som syns när menyn är stängd

4.1 Hem

Hemvyn är det första användaren möter efter att inloggningen har genomförts. Vyn består till huvuddel av två knappar; en som skickar användaren vidare till livsmedelsvyn och en som dirigerar vidare till transportvyn, samt ett textfält som visar användarens totala förbrukning sen dess att loggningen startade.

4.2 Livsmedel

En av de två huvudfunktionerna i applikationen är att kunna registrera sina produkter på ett enkelt och smidigt sätt. Det har då skapats en separat del i applikationen där användaren ska kunna registrera produkter som antingen läggs till manuellt eller med hjälp av enhetens kamera för att läsa av en produkts streckkod. De två övriga funktionerna är att användaren kan skapa olika inköpslistor för att kunna placera olika produkter var för sig samt att det totala antalet ekologiska produkter registreras i applikationen och används i topplistor.

4.2.1 Inköpslistor

Livsmedelsdelen består av två fragment. Det första fragmentet är en vy där en lista av alla sina tidigare inköpslistor kan betraktas, se figur 4.2. Listan blir skapad med hjälp av RecyclerView och visar medhavande namn, totala koldioxidpåverkan och vilket datum listan skapades. Genom att använda det editerbara textfältet högst upp i vyn kan även nya listor skapas. All denna information sparas i databasen och hämtas igen varje gång användaren går tillbaka till vyn. Om soptunnan på varje lista klickas på tas denna lista bort. I hörnet finns det alltid en rund cirkel med det sammanlagda koldioxidvärdet för alla listor. Listor kan bli redigerade genom att de blir klickade på.



Figur 4.2: Vy där användaren kan se alla sina inköpslistor och interagera med dem

4.2.2 Varulista

Efter att användaren skapat eller gått in i existerande inköpslista öppnas ett nytt fragment som påminner mycket om fragmentet för inköpslistor. Varor kan läggas till i inköpslistan genom att skriva in namnet på valfri matvara i ett textfält högst upp i vyn. När tre eller fler bokstäver skrivits in ger applikationen förslag för att matcha mot vad användaren försöker skriva. Detta sker genom att databasen hämtar förslag som sedan matchas mot vad som skrivs in i textfältet. Känner sig användaren klar kan antingen knappen på tangentbordet *Done* användas alternativt knappen med ett plustecken bredvid textfältet för att lägga till varan i varulistan. Applikationen kommer då göra en till genomsökning i databasen för att se om det finns fler förslag på önskad vara. Om så är fallet öppnas en dialogruta, i vilken alla förslag som hittas i databasen visas. Om inget av alternativen passar användaren finns även ett extra alternativ som innebär att användaren kan skapa ett nytt objekt som läggs till i databasen.

När en vara väl är tillagd kan vikten på denna ändras genom att användaren klickar på plus eller minus som lägger till och tar bort ett kilo av varan. Det går även att fylla i antalet kilo direkt för att kunna skriva in exakt hur mycket som ska loggas. Precis som i vyn med inköpslistor finns en soptunna men här tar den bort matvaran. Varan visar även sitt namn och hur mycket den släpper ut baserat på varan och antalet.

Så länge vyn för varulistan visas finns det en grön bubbla nere i högra hörnet som visar användarens totala koldioxidutsläpp för just denna listan. Högst upp brevid textfältet för att

lägga till varor finns även en knapp som ser ut som en pil som laddar upp resultatet till topp-listan. Längst upp i högra hörnet finns en knapp som rensar hela listan. Det finns även en knapp som ser ut som en kamera för att skanna streckkoder och lägga till varor i listan på detta vis.

4.2.3 Adaptrar

För att kunna lägga till egna objekt i RecyclerView-listorna behövde två stycken adaptrar implementeras. Dessa hanterar interna listor som håller reda på all information som vyn behöver. De hanterar också när nya objekt läggs till i listan och när objekt tas bort ur listan. Eftersom det är självskapade vyer behövdes de händelselyssnare som finns i listobjekten även implementeras i adaptrarna [10]. Detta inkluderade händelser som när användaren byter vikt på en vara eller klickar på en lista för att ta sig vidare in till den varulistan. Dessa adaptrar implementerar även hur listan målar om sig när objekt tas bort eller blir dolda av vyn genom bläddring.

4.2.4 Streckkodsläsning

Vid användning av kameraknappen i en inköpslista kommer först en kontroll att göras om en extern applikation med namnet ZXing Barcode Scanner existerar. Finns den inte, kommer användaren att få en förfrågan om att ladda ner och installera den. När det är utfört startas Barcode Scanner. Det är upp till användaren att rikta enhetens kamera mot en produkts streckkod. När en lyckad och skarp bild är tagen läser Barcode Scanner av streckkoden och får fram ett antal siffror samt skickar vidare detta till GreenMe.

GreenMe kommer då att ta denna siffersträng och göra ett anrop till en extern API-tjänst som tar emot siffersträngen och ger tillbaka en sträng med ett produktnamn. Det första som händer när produktnamnet har hamnat i enheten är att slå upp i den interna databasen för att se om det finns någon produkt som matchar, om inte, får användaren ett alternativ att lägga till denna produkt manuellt. Det kommer att visas en ny ruta där det uppskattade koldioxidutsläppet per kg fylls i. Efter detta finns produkten i den interna databasen för framtida behov. När allt är klart kommer den inskannade produkten att synas i inköpslistan och antalet produkter går då att välja med plus-och minusknapparna.

4.2.5 Varudata

Utsläpp från olika matvaror kan variera från produkt till produkt och från märke till märke, därför infogades först en del genomsnittsvärden som baserades på en rapport från Sveriges lantbruksuniversitet [52]. Värdena är uträknade med hjälp av livscykelanalys och då har de försökt att räkna med allt från producent till konsument, vilket kan inkludera besprutning, transport till matvarubutiken men däremot inte transporten från matvarubutiken till privatpersoners hem. Det är även uppmätt i CO_2e vilket innebär att andra miljögifter förutom koldioxid är medräknat och omräknat beroende på hur mycket skada de gör i jämförelse med koldioxid [39]. För att utöka databasen lades en del grönsaker vilket är baserade på en rapport från

Livsmedelsverket [50], även dessa är mätta i CO_2e och är beräknade från producent till konsument liksom rapporten från Sveriges lantbruksuniversitet.

4.3 Transport

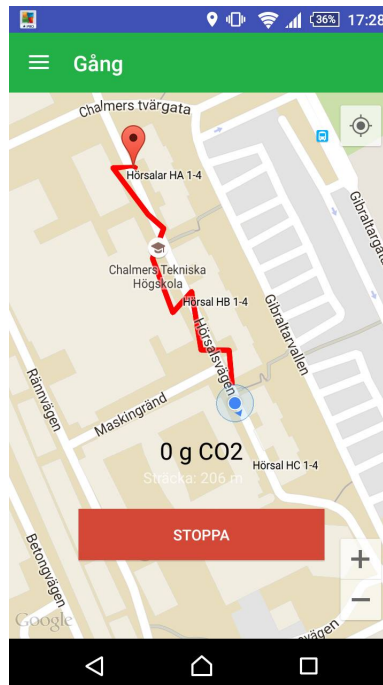
Transportdelen i applikationen består av två fragment; det första fragmentet, VehicleFragment, låter användaren välja mellan att antingen automatiskt logga en resa och på så sätt kunna få ut ett mer exakt koldioxidutsläpp eller att i efterhand manuellt kunna mata in antalet kilometer. Vidare låter detta fragment, i både fallet av manuell och automatisk loggning, användaren välja vilken typ av färdmedel som används alternativt användes under resan. Detta är presenterat genom olika knappar. Vid klick på valfritt färdmedel höjs transparensen på de övriga knapparna för att göra det tydligt för användaren vad som valts, se figur 4.3. I dagsläget finns det stöd för gång, cykel, buss, tåg och bil. När det kommer till bil finns det fyra olika typer av bilar att välja mellan, som i sin tur genererar olika koldioxidutsläpp. Dessa är liten bil, mellanstor bil, stor bil och slutligen stadsjeep. För mer detaljerad beskrivning kring uträkningarna av fordonsdatan, se avsnitt 4.3.1. När användaren valt ett transportmedel och vill följa resan automatiskt kommer en multipel, vilket motsvarar koldioxidutsläppet för valt transportmedel, skickas vidare till TravelFragment. När istället en resa läggs till manuellt fyller användaren i antalet kilometer, vilket multipliceras med det genomsnittliga utsläppet per kilometer som resulterar i det totala utsläppet för resan.



Figur 4.3: Manuell inmatning av resa där resan sker med buss

Det andra fragmentet behandlar den automatiska spårningen av hur långt användaren har förflyttat sig och självklart även loggningen av koldioxidutsläppet. Figur 4.4 som visar hur

vyn ser ut och består till största del av en karta som är en färdig komponent hämtad från Googles Maps API. Ovanpå den finns det en knapp samt ett textfält där knappen är till för start och stopp av loggning i applikationen. Resten av stycket beskriver händelseflödet i vyn och vad som händer i bakgrunden. För att få en överblick, se bilaga C.1.



Figur 4.4: Fragment över automatisk loggning med en användares förflyttning

Vid ett första tryck på startknappen zoomas användarens senaste inlästa position in med hjälp av “LocationServices”-paketet som skapar ett LatLng-objekt, kallat prevPosition, där longituden och latituden sparas. Därefter sätts en flagga ut på skärmen för att användaren ska veta var loggningen startar. Vidare utnyttjas ett paket som skickar ett anrop till en callbackmetod varje gång användaren har förflyttat sig en viss sträcka samt en bestämd tid har förflutit sedan senaste anropet. Sträckan och tiden är justerbar vilket medför att det går att anpassa exaktheten beroende på hur ofta utvecklaren anser att en ny position behöver hämtas. När en ny position avläses skapas ett nytt LatLng-objekt, kallat currPosition. Avståndet mellan den föregående positionen och den nya avlästa positionen beräknas samt ett sträck, en Polyline, mellan positionerna målas ut på skärmen. Sedan multipliceras sträckan med en multipel som kommer från föregående fragment och den totala koldioxidförbrukningen skrivs ut på skärmen.

Det sista som händer är att currPosition skrivs över till prevPosition vilket medför att nästa sträcka som ska mätas kommer vara mellan prevPosition (som tidigare var currPosition) och en ny currPosition (vilket är den senaste hämtade positionen). När användaren har nått sin destination och klickar på stoppknappen (som är samma knapp som för “starta”) avslutas

loggningen, resultatet skickas till topplistan och en markör för senaste positionen sätts ut för att visa användaren var loggningen slutade.

4.3.1 Fordonsdata

Gällande koldioxidutsläpp för bilar varierar det mycket från bil till bil och är även beroende av typen av bränsle [8]. Applikationen är i dagsläget begränsad till enbart bilar som drivs med bensen. För att skapa ett smidigt användargränssnitt får användaren välja mellan olika kategorier av bilar; liten bil, mellanstor bil, stor bil och stadsjeep. En liten bil motsvarar exempelvis en Volkswagen Polo och har ett koldioxidutsläpp på 0,15812 g/m. En mellanstor bil motsvarar exempelvis en Audi A4 som har ett koldioxidutsläpp på 0,19028 g/m. En stor bil motsvarar exempelvis en Volvo V70 och har ett koldioxidutsläpp på 0,23048 g/m. Slutligen motsvarar en stadsjeep exempelvis en BMW X5 och har ett koldioxidutsläpp på 0,27068 g/km. Alla dessa värden är hämtade från Utslappsrott.se.

För att bestämma utsläppsvärden för buss och tåg användes SJs miljökalkyl, som är baserad på en rapport från Kungliga Tekniska högskolan [49]. Koldioxidutsläppet för buss- och tågresor påverkas av antalet resenärer och därför har en belägningsgrad, antalet resenärer i fordonet, på cirka 50 % valts. Koldioxidutsläppet för buss anges som 0,053 g/m och tåg anges som 0,0007 g/m. I den senare ingår enbart framställningen av el som driver tågen.

Gång genererar inte i ett direkt skede något koldioxidutsläpp. Något som dock skulle kunna påverka indirekt är huruvida matintaget förändras. Men enligt tidigare forskning ändras inte matvanor i någon större utsträckning vid skifte från bilåkning till gång, givet att distansen är rimlig [45]. Snabbare andning ger ett ökat koldioxidutsläpp, men anses vara försumbart i detta fall och koldioxidutsläppet för gång anges därför som 0 g/m.

En av anledningarna till att cyklar genererar ett koldioxidutsläpp är främst på grund av den infrastruktur som cyklar behöver, exempelvis vägar [45]. På grund av cykelns vikt påverkas inte vägar i samma utsträckning som av bilar och det har därför valts att försumma denna påverkan för att på så sätt uppmuntra användaren till att cykla. Likt gång påverkas inte matvanor, men snabbare andning ger ett ökat koldioxidutsläpp, som även det ansetts vara försumbart och därför anges även koldioxidutsläppet för cykel som 0 g/m.

4.4 Topplistor

I likhet med inloggningen till applikationen är topplistorna kopplade till ett av Googles APIer; Google Play Services som i sin tur använder Google+ profiler för att lista användarens placering vilket medför att användaren kan se sina cirklars poäng. Topplistornas funktioner och utseende kommer från Google Play Services vilket medför att utvecklarens fokus har legat kring hur poängen ska registreras på bästa sätt. Nedan kan läsas om hur respektive topplista är implementerad och vad de mäter, se figur 2.4 för överblick av topplistan.

4.4.1 Livsmedel

Topplistan för livsmedel räknar antalet ekologiska produkter som användaren har köpt. Varje enhet av en ekologisk vara ger användaren ett poäng på topplistan. För att uppdatera topplistan med aktuell poäng lägger användaren till sina produkter i shoppinglistan och klickar sedan på ikonerna med en pil uppåt som kan ses uppe till höger i figur 4.5. Eftersom användarens poäng inte lagras lokalt på telefonen anropas topplistan med en förfrågan om att få nuvarande poängen som är registrerad i topplistan tillsammans med en callbackmetod dit resultatet returneras. I den metoden adderas sedan resultatet från anropet med användarens poäng från shoppinglistan och laddas upp i topplistan. Detta medför att topplistan tre olika tidshorisonter alltid kommer visa samma resultat eftersom poängskörden alltid kommer ökas på, åtminstone inte minska.



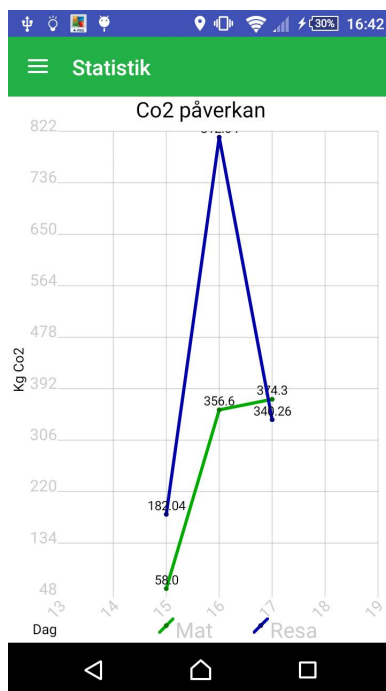
Figur 4.5: Varulista med tillagda varor och deras respektive inlagda koldioxidvärden och den totala mängden i den gröna rutan

4.4.2 Transport

Transportsektorns topplista räknar antalet meter som användaren förflyttar sig till fots eller med hjälp av cykel då dessa två inte har ett direkt utsläpp av koldioxid. Uppdateringen av listan är identisk med hur livsmedelssektorns uppdatering sker bortsett från den lilla skillnaden att anropet kan komma från två olika fragment beroende på om användaren väljer att logga sin resa manuellt eller automatiskt.

4.5 Statistik

Statistikvyn är en vy som tar information från databasen och presenterar denna på ett så informativt sätt som möjligt. För att uppnå detta valdes det att måla upp saker genom en graf, grafen ifråga blir genererad av AndroidPlot. Det finns två olika linjer på grafen som är blå respektive grön, se figur 4.6. Den gröna representerar det koldioxidutsläpp som livsmedelsdelen gett upphov till. Allt som loggas för livsmedelsdelen under en dag adderas ihop och representeras med en nod i grafen. Den blå linjen representerar det koldioxidutsläpp användaren har loggat för resedelen, detta hanteras på samma sätt som för livsmedelsdelen.

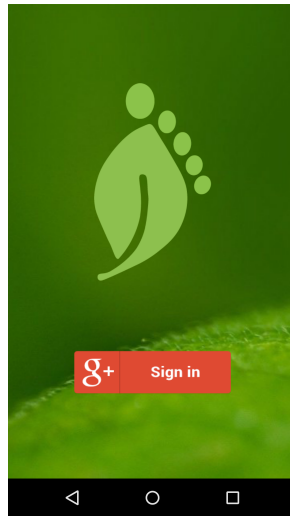


Figur 4.6: Statistik vyn för en användare som använt applikationen i tre dagar

4.6 Log in

Inloggningsskärmen till skillnad från de andra vyerna har inte ett grafiskt gränssnitt byggt med hjälp av en XML-fil bestående av olika komponenter utan det består av en Java-fil som

utökar en SurfaceView. En SurfaceView är en vy som målar om skärmen med en viss frekvens vilket gör det möjligt att få en skärm med rörliga delar. Detta utnyttjas för att få bakgrunden att rulla horisontellt men har även medfört att "Sign in"-knappen, se figur 4.7, inte tekniskt sett är en knapp utan en bild. Den har därför ingen egen lyssnare utan utnyttjar applikationens globala lyssnare som reagerar varje gång skärmen berörs. Vid beröring på skärmen där "knappen" ligger, skickas användaren vidare i inloggningsflödet. För att undvika problem vid användning av olika skärmstorlekar förändras "knappens" beröringsfält (och även knappens storlek) beroende på antalet pixlar i skärmen vilket gör vyn mer dynamisk.



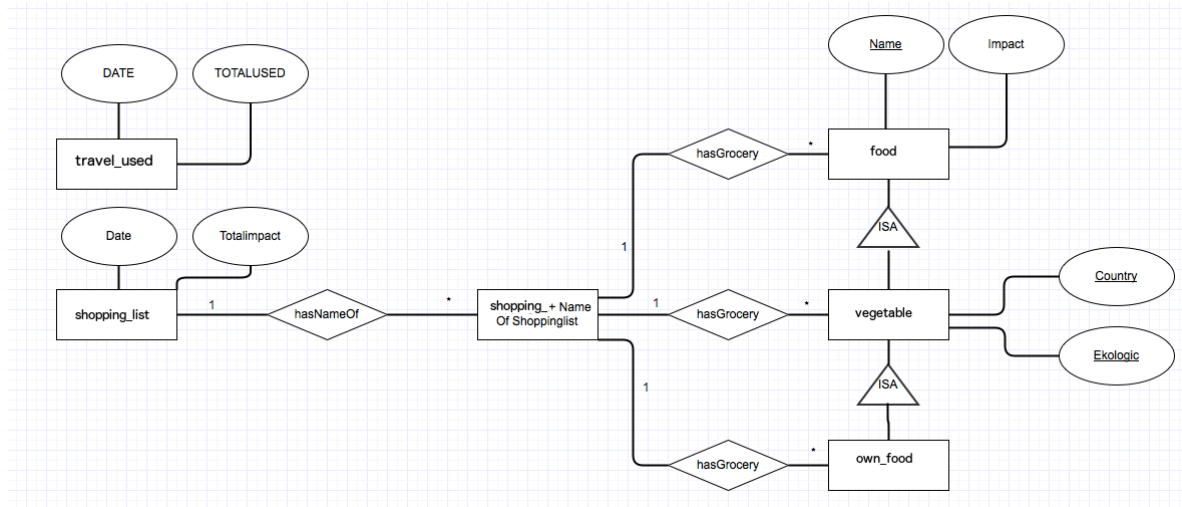
Figur 4.7: Startsidan som en icke-inloggad användare ser, rörlig bakgrund och en knapp för att logga in med sitt Google-konto

Inloggningsflödet för applikationen tillhandahålls av Googles API för Google+ och det enda utvecklaren behöver göra är att skapa en klient mot API:t för att sedan använda den klienten när uppkoppling mot Google ska ske. På varje telefon kan användaren välja att lagra sina olika Google-konton, dessa konton används sedan i inloggningsflödet vilket gör det enkelt för användaren att logga in med sitt konto. Allt som händer efter att användaren klickar på knappen hanteras alltså av Google, både gällande vyer och logik. Vanligtvis kommer dock inte användaren i kontakt med inloggningsvyn eftersom ett försök med uppkoppling mot Google sker i bakgrunden först och skulle det inte fungera visas inloggningsvyn. Användaren får då möjligheten att byta konto för att sedan testa på nytt.

4.7 Databasen

Efter applikationen har blivit installerad finns det fem tabeller i databasen, dessa är `travel_`, `used`, `shopping_list`, `vegetable`, `food` och `own_food`. Tabellerna `vegetable` och `food` är initierade med information från miljörapporter, se avsnitt 4.2.5. `food` innehåller enbart namn och hur stor påverkan respektive produkt har. `vegetable` innehåller även information om vilket land produkten kommer från och om den är ekologisk eller inte. `own_food` finns för att kunna

lägga till nya matvaror, och har samma uppbyggnad som vegetable, se figur 4.8. travel_used används för att logga hur mycket utsläpp som släpps ut genom transport och används sedan för att målas upp i statistikvyn. shopping_ är en samling av flera tabeller som innehåller matvaror för en inköpslista och där varje tabell heter shopping_ tillsammans med det namnet som listan är döpt till, dessa tabeller refererar bara vidare till food, vegetable eller own_food. shopping_list är en tabell med namn på de tabeller i shopping_, datum då de skapades och deras totala påverkan.



Figur 4.8: Entity-Relationship diagram över databasen

4.8 Testning

Testning genomfördes genom att skapa testklasser som testar att funktionalitet i applikationen inte ändras eller går sönder under utvecklingens gång. Dessa tester körs endast lokalt på utvecklarens system via Android Studio. Testerna består av enkla metoder för att säkerställa att objekt inte är odefinierade, innehåller fel data eller returnerar oönskat värde. Det utförs även ett test för att säkerställa att användaren är ansluten till tjänster som Google tillhandahåller.

4.9 Resultat av användbarhetstester

Användbarhetstesterna gav en bra resultatbild och för att göra det överskådligt har åtta diagram bifogats i bilaga B. Dessa är nerskalade till entydiga svar men många av frågorna uppmanade testerna till att utveckla respektive svar och det är de svaren som sammanfattas i detta stycket. Styckesuppdelningen är gjord utefter respektive sektor och för att se det kompletta frågeformuläret som utnyttjades vid användbarhetstesterna, se bilaga A.

Den första frågan kring hur användaren uppfattade processen att starta applikationen och komma igång innehöll inte särskilt många utvecklade svar utan figur B.1 representerar re-

sultatet sanningsenligt. Likaså går det att se svaret på frågan, “var menyn enkel att hitta” i figur B.2. På frågan om hur förstasidan upplevdes besvarades den överlag med att den var enkel och användarvänlig men att designen, framförallt knapparna, var ful. Även frågan om något steg var oklart ställdes men uppfattades olika av de som höll i användbarhetstestet och därför utgick frågan.

Det kommande stycket innehåller enbart frågor kring användarupplevelsen i livsmedelsvyn och responsen på när testaren skulle beskriva sitt första intryck var mycket spridd där de flesta var överens om att första intrycket inte var bra. En av de vanligare kommentarerna var att det var otydligt vad som visades samt hur testaren skulle slutföra uppgiften. Det fanns även en fråga om hur funktionen för streckodsläsning upplevdes och den var uppskattad men behovet av att vända på telefonen för att läsa av streckkoderna ansågs jobbigt och onödigt. Även kamerasymbolen upplevdes förvirrande och inte tillräckligt självbeskrivande. Frågan “Vad tycker du om hur listan är presenterad?” ställdes och likt ovan nämnt var detta en av de mer problematiska områdena. Utöver svårigheten med inmatningsfältet påpekades att överskriftsfältet borde ha ett bättre namn. Det fanns även åsikter om designen då den upplevdes svåränvänd. Överlag var det få utvecklade svar kring frågan om testaren trodde sig komma att utnyttja applikationens implementation av livsmedelsloggning men det vanligaste svaret var att inköpslistan hade varit mest intressant.

Den tredje och näst sista frågekategorin var mer inriktad på hur testaren såg på transportdelen. Första frågan var om huruvida det var enkelt att förstå knappen som avgör om det är manuell eller automatisk loggning. Även detta var en av de svårare momenten för testarna och de flesta svaren innehöll svar med förslag på namnbyte, några av dem var: “Starta nu/Rapportera” och “använd GPS/Registrera sträcka manuellt”. Positionen av denna knapp i vyn var också en återkommande motivering till svårigheterna. Däremot var de flesta testarna överens om att fordonsknapparna var tydliga men med kommentaren att valet spårvagn saknas, alternativ trodde tåg var spårvagn. Avslutningsvis för transportkategorin fick testarna svara på hur det uppfattade bilknapparna samt om något saknades med de. Svaret blev att färgvalen var ett problem samt definitionen på vad de olika bilstorlekarna innebar inte var tillräckligt tydlig. Något enstaka önskemål om annat fordon likt motorcykel och lastbil förekom också.

Som sista stycke fick testarna svara allmänt kring applikationen och ge sina förslag på förbättringar. Den första frågan formulerades just kring hur det generella intrycket var samt om testarna skulle använda applikationen i vardagen. I princip ingen av testarna skulle använda den i det rådande skicket när de testade den då många önskade en bättre design överlag och ett bättre incitament. Testarna fick även frågan om de skulle vilja ändra på något och det vanligaste önskemålet var att det skulle finnas fler saker att logga. Även önskemål om att höja helheten på applikationens design återkom. Frågan “Vad tyckte du särskilt om?” ställdes och i svaret fanns det 11 unika kommentarer där spannet var allt från inloggningsvyn till navigationsmenyn vidare till applikationens koncept. Även motsatsen om vad testarna inte tyckte om med applikationen besvarades och sammantaget tyckte inte testarna om hur livsmedelsvyn var utformad. Den var svårförstådd men konceptet uppskattades.

För att kunna sammanfatta vad testarna ansåg och se om applikationen hade någon spridningspotential ställdes frågan; "Skulle du rekommendera den för dina vänner?". Svaret var främst att de inte skulle rekommendera den och de som svarade ja tvekade ofta på om de skulle fortsätta använda applikationen under en längre period. Avslutningsvis ställdes frågan om de trodde att applikationen skulle göra dem mer medvetna om hur "gröna" de var och ingen av testarna avfärdar att applikationen skulle göra dem mer "gröna". Hälften trodde att den direkt skulle ge upphov till en förändring och det största motargumentet trodde testarna var svårigheten att locka användare.

Kapitel 5

Diskussion

Applikationens nuvarande status är att den är fullt fungerande men det finns fortfarande utrymme för förbättring av både design och hur de olika funktionerna är implementerade. Det hade också varit önskvärt med vidareutveckling av ny funktionalitet och på så sätt täcka fler områden. Under utvecklingen uppstod viss problematik som gjorde att önskad funktionalitet inte kunde realiseras. Dessa problem beskrivs närmare i avsnitt 5.9.

De övriga avsnitten är uppdelade efter applikationens olika sektorer samt att de första avsnitten belyser rena teoridelar.

5.1 Val av utsläppsområde

Under planeringsfasen behövde områden som skulle logga koldioxidutsläpp bestämmas. Det diskuterades en del om vad som skulle loggas. För att motverka den sekundära rekyleffekten valdes det att fokus skulle ligga på mer än ett område.

Mot bakgrunden att resor står för 30% av det totala koldioxidutsläppet upplevdes det som en naturlig del att logga. Eftersom utsläppen orsakade av boende exempelvis hör ihop med var du bor och hur bra isolerat ditt hus är, är en del svårt att ändra på och skiljer sig inte så mycket beroende på vad en person gör för val i sin vardag. Framförallt är förändringarna svåra att spåra. Av anledning av detta valdes boende bort. Det kändes därför naturligt att få med utsläpp som beror på livsmedel i applikationen, eftersom livsmedel trots allt står för 33% av koldioxidutsläppen.

5.2 Beteendeförändringar och incitament

För att faktiskt få användare att utnyttja applikationen var målet att skapa en applikation som inte kräver speciellt mycket av användaren, men också ett incitament till att faktiskt använda applikationen. Som nämnt i avsnitt 2.2.4 ligger problemet ofta i att viljan till att förändra ohållbara beteenden inte är lika stark som ängslan för miljön eftersom det ofta innebär att en beteendeförändring måste ske vilket kan vara krävande. Exempel på incitament i en potentiell nästkommande version av applikationen skulle kunna vara att användaren får hämta upp en gratis kaffe på vägen till jobbet om buss nyttjas istället för bil. Ett annat exempel skulle kunna

vara om användaren går eller cyklar en viss andel av tiden erbjuds en rabatt i valfri matbutik. Det finns dock nackdelar med detta, det finns risk för rekyleffekter, vilket gör att användaren ändå inte kommer minska sin miljöpåverkan.

Under arbetets gång noterade gruppen att respektive gruppmedlem har börjat ta ett större ansvar miljömässigt utan att aktivt tänka på det. Ett ständigt arbete med olika siffror kring vad som har stor inverkan på miljön har givit kunskap vilket medfört en beteendeförändring utan direkt uppsatta mål för det. Om denna beteendeförändring har skapats bland de som utvecklat applikationen finns det även en viss chans att användaren upplever samma sak. Även om inte användaren loggar sitt ekologiska fotavtryck under någon längre period kan det räcka för att skapa en medvetenhet och vana som leder till en beteendeförändring. En sådan förändring skulle medföra att syftet med projektet är uppnått utan att faktiskt behöva nå upp till vissa av målen eller utmaningarna eftersom användaren redan har skaffat sig en ökad miljömedvetenhet. Det ska dock tas hänsyn till att denna beteendeförändring troligtvis bara är temporär och kommer avta i takt med att kunskapen kring ämnet avtar.

5.3 Livsmedel

I kommande avsnitt diskuteras fördelar och nackdelar med en av huvudfunktionaliteterna; loggning av livsmedel.

5.3.1 Inköpslistor

Idéen med inköpslistor grundar sig i att användaren inte ska behöva en ytterligare applikation för att skapa inköpslistor. Med denna funktionalitet kan användaren skapa olika listor beroende på vad som brukar inhandlas, kanske baserat på vilken affär eller en lista för lördagens kalas. En vidareutveckling av applikationens funktionalitet som skulle kunna uppskattas är möjligheten att dela med sig av en befintlig lista till en vän eller sambo. Det skulle kunna ske via NFC-överföring genom att hålla upp två stycken mobiler mot varandra och överföra listan till den andra enheten. En ytterligare förbättring skulle kunna göras genom att inköpslistan blir smartare. Till exempel genom att applikationen känner av att enheten är nära en viss butik och därmed sorterar upp dessa listor användaren har skapat i just denna affären.

5.3.2 Varulista

En utopi hade varit en databas med ett koldioxidvärde för samtliga produkter och varor från olika affärer. Användaren hade som en följd inte behövt lägga in varor manuellt och dessutom inte blivit tvingade till att leta fram vad just den önskade produkten har för koldioxidpåverkan. Men fram till att en sådan databas existerar finns åtminstone möjligheten för användaren att lägga till varor manuellt med ett uppskattat koldioxidvärde.

En förbättring för att användaren enkelt ska kunna sortera vanligen förekommande produkter eller varor skulle kunna vara att sätta en stjärna på en produkt och på så sätt lägga till den i

sina favoriter. På detta vis skulle dessa favoriter kunna prioriteras högre vid en sökning.

Efter användbarhetstesterna noterades problem med designen och upplägget på de olika grafiska elementen. Inmatningsfältet behöver vara tydligare då det fanns uppenbara problem för användaren att lokalisera dess position och ibland även dess existens.

5.3.3 Streckkodskanning

I dagsläget görs uppslagningar av varor mot ICAs databas, men en möjlighet är att även göra uppslagningar mot andra kedjors databaser tills en träff uppnås. En sådan funktion skulle inte påverka enhetens batteritid märkbart, eftersom själva uppslagningen görs på tredjeparts hårdvara, då behöver bara enheten skicka en lite förfrågan och vänta på ett svar. En svårighet är också texten som returneras från databasen. Texten består ofta av förkortningar och halva ord, som det brukar vara på kvitton i affären, vilket gör det svårt att tyda vad det egentligen är för någon produkt.

En förbättring skulle kunna vara att användaren själv får uppskatta en vikt och föra in den i listan. Eftersom att hela applikationen är satt att visa innehållet i porträttiläge blir det förvirrande för användaren när den externa applikationen ZXing barcode scanner forcerar skärmen till ett landskapsläge och efter att lyckad skanning är gjord återvänder till GreenMe för att åter vara i porträttiläget igen. För att kunna lösa detta skulle hela ZXing barcode scanner biblioteket behöva implementeras i applikationen för att på så sätt kunna skraddasy inläsningen till applikationens behov. Detta skulle även ta bort kravet att ha ZXing Barcode Scanner som en separat installerad applikation på användarens enhet.

5.4 Transport

En stor fördel med resedelen i sin helhet är att det krävs få knapptryckningar för att logga en resa, i synnerhet om du vill logga din resa automatiskt. Detta i sin tur leder till att användarens snabbt når fram till sitt mål och en effektivitet uppnås med avseende på användbarhet. I de två nästföljande avsnitten följer fördelar och nackdelar med de två versionerna som bygger upp resedelen.

5.4.1 Fordonsvyn

En fördel med applikationen är att utöver automatisk inmatning är även manuell inmatning implementerad. Detta som en följd av att många människor är glömska vilket innebär att de troligen kommer glömma bort att logga sin resa i realtid, eller helt enkelt vill fylla i alla resor som gjorts under en dag på samma gång. Därför är det enkelt att lägga till en resa i efterhand. Det krävs dock av användaren att en ungefärlig uppfattning av hur lång sträcka som färdats skapas, vilket medför att vissa resor kan bli mycket generella. Ett problem som uppenbarades under användartesterna var att vissa användare hade svårt att uppfatta hur skiftet mellan automatisk och manuell inmatning går till. Detta behöver göras tydligare i en vidareutveckling eftersom det är en central funktion för användaren. Detta kan göras genom att exempelvis

skapa större avstånd mellan switchen, som avgör om det är manuell eller automatisk inmatning, och överskriftsfältet samt även byta färg på texterna för att dra till mer uppmärksamhet.

Det kan även diskuteras om huruvida knapparna för olika typer av bilar är rätt utformade. Fördelen är att bilägaren inte behöver ta reda på vad bilen har för koldioxidutsläpp. Användaren kan istället klassificera sin bil enkelt efter en kategori och får då en genomsnittssiffra på vad koldioxidutsläppet är för den bilstorleken. Som ett steg i en potentiell vidareutveckling hade ett alternativ till detta varit att användaren fick mata in koldioxidutsläppet för sin bil som sedan hade sparats till en databas kopplat till användarens konto. Resor som görs med både buss och tåg mäts i personkilometer och beror därför på beläggningsgraden. I rusningstrafik med många passagerare ger en resa ett mycket lägre koldioxidutsläpp, och tvärtom om en resa med kollektivtrafik har låg beläggningsgrad. Därför har ett medelvärde valts för respektive färdmedel.

En tanke från början var att även ha en knapp för flygresor, men detta valdes bort då telefonen i automatiskt läge bör ha tillgång till uppkoppling för Google Maps, vilket inte är möjligt när telefonen är satt i flygplansläge.

5.4.2 Kartvyn

Kartvyn är den vy som tveklöst kräver mest batteri av telefonen, med största sannolikhet beror det på att den är GPS-beroende. Vid en eventuell vidareutveckling av applikationen skulle detta behöva bekräftas med någon form av mätning och även åtgärdas. Frekvensen på hur ofta en ny position hämtas är satt högt men det behovet finns eftersom linjen (polyline) målas upp utefter varje ny position. En lägre frekvens för hämtningen hade medfört att linjen oftare passerat över(genom) byggnader om användaren till exempel, skulle gå runt ett hörn. Det är inte bara linjen som påverkas utan även den beräknade sträckan och koldioxidutsläppet. Om inte linjen hade målats ut borde det gå att kompromissa med en lägre frekvens, och ge en mindre korrekt siffra, för att spara på batteriet. Eftersom användaren då inte ser felet hade förtroende för att applikationen loggar korrekta siffror inte riskerats. Det är rimligt att genomföra en sådan ändring av den orsak att vyns inläsningsdata inte har den exaktheten att det skulle påverkas märkbart men det användaren ser måste inge en känsla av att allting är tillräckligt exakt, annars finns det risk att även användaren börjar kompromissa.

För att återknyta till bristen på exakthet i vyn har märklig data från positionsbestämaren genomgående avlästs. Ett fenomen där positionen ofta avläses till specifika platser om och om igen som att en mast eller liknande finns. Dock har inte fallet varit så och fortsatt felsökning kunde inte avgöra vad problemet var. Även ett problem med att GPS-signalen ofta varierar i exakthet har noterats. På grund av dess stundtals dåliga precision verkar det som att den hoppar runt mycket och skulle den bestämda sträckan för att skapa en ny position överstiga noteras en ny position vilket medföra en felaktig uppdatering av koldioxidförbrukningen.

5.4.3 Användarens integritet

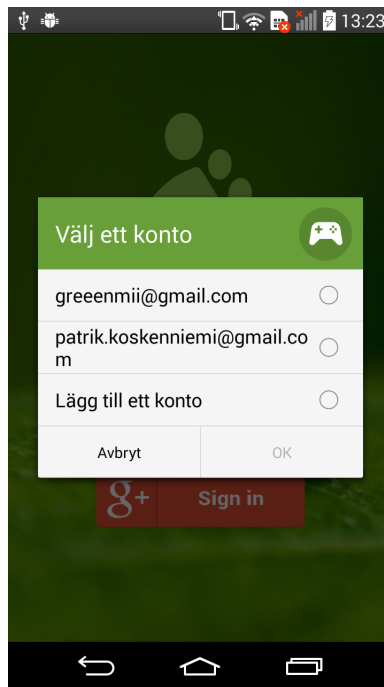
Ingen information som har med användarens position att göra sparas. Informationen som hämtas skickas direkt till Google Maps API och tillbaka ges en position som används vid beräkning av avstånd. Det går inte att garantera vad Google gör eller använder den datan till. Om information om användarens position hade sparats finns det en risk att andra applikationer missbrukar den informationen och användarens integritet äventyras.

5.5 Statistik

Färgvalet i grafen valdes till grön respektive blå för att ha i åtanke att den vanligaste färgblindhet är röd-grön färgblindhet [11]. Om mer tid hade funnits hade även en navigeringsfunktion för att titta närmare på specifika delar av grafen. Ytterligare funktioner hade också kunnat implementeras exempelvis fler linjer för att representera annat än bara livsmedel och transport. Ett exempel skulle kunna vara hur långt användaren har förflyttat sig utan att göra av med koldioxid. Något som även kunde implementerats är att filtrera ut vissa linjer i taget.

5.6 Inloggning

Eftersom inloggningen enbart sker med hjälp av ett Google+ konto begränsas användarbasen. Ur användarsynpunkt (och applikationens spridningspotential) hade det med största säkerhet varit bättre att utnyttja Facebooks motsvarande inloggning. Det hade dock inneburit ett nytt API att sätta sig in i och förstå vilket troligtvis hade tagit mer tid. Samma sak gäller om inloggningen hade implementerats på egen hand utan hjälpen av ett API. Däremot är en fördel att användarna ofta lagrar sina Google+ konton på telefonen och därför kan applikationen återanvända dessa vid inloggning, se figur 5.1, istället för att behöva slå in inloggningsuppgifterna efter varje gång en utloggning har skett.



Figur 5.1: Vid flera konton på telefonen får användaren välja vilket att utnyttja för inloggning

5.7 Databasen

Grundtanken med databasens struktur var att det skulle vara enkelt att dela användarskapade produkter mellan användarna och därför skapades till exempel `own_food` separerad från `vegetable`. Detta gjordes med tanken att det i framtiden skulle finnas en samlad databas för alla användare på en server. Till denna server skulle användare kunna skicka egna förslag och om tillräckligt många skickade liknande information skulle den produkten läggas till i databasen. Ett problem med att denna server inte finns för tillfället är att användaren manuellt måste lägga till många varor själv vilket inte är användarvänligt och kan bli mycket tidskrävande för användaren då denna antagligen inte vet hur mycket varje vara släpper ut. Detta hade även gått att åtgärda genom att ha en större databas från början. Som diskuterat i avsnitt 5.3.1 fanns det även tankar om att med hjälp av NFC låta användaren dela sina inköpslistor, därför gjordes valet att dela upp varje inköpslista i olika tabeller för att lättare kunna skicka dessa istället för att göra genomsökningar genom en stor tabell.

5.8 Fusk

När det förekommer tävlingsmoment finns det alltid en risk att användaren lägger in falsk data för att förbättra sitt resultat. Det skulle exempelvis kunna handla om att ange varor som är ekologiska även om de är icke-ekologiska, eller att låta bli att knappa in en viss vara. För att hantera ett sådant scenario skulle det krävas omfattande arbete med applikationen för att

implementera de kontroller som behövs och i nuvarande version finns ingen sådan kontroll.

I resedelen skulle en användare kunna säga att cykel eller gång nyttjats fast egentligen använt ett annat fordonslag som exempelvis bil eller buss. Ett möjligt sätt att hantera det skulle vara att införa tekniska kontroller som mäter hastigheten som användaren har vid tillfället och avgöra om hastigheten är rimlig för gång eller cykling.

5.9 Problematik under implementationen

Utvecklingen av denna typ av applikation har medfört en del svårigheter i utformandet och även implementationsmässigt, vilket beskrivs närmare i detta avsnitt.

5.9.1 OCR - Optical Character Recognition

För att kunna göra om ett foto med text till ett digitalt format av texten går det att använda Optical Character Recognition (OCR). I ett mobilsammanhang används mobilens kamera och tar ett foto som sedan ska analyseras av ett program. Beroende på programmet kan resultatet skilja sig med avseende på hur stora tabeller av inskannade bokstäver de har tillgång till. Tillvägagångssättet med att matcha mot en tabell har sina brister då programmet omöjligt kan spara ner alla olika varianter av hur ett tecken skrivs. Det finns dock smartare system som bygger på att hitta mönster, där varje bokstav har ett unikt mönster för att identifiera den.

En funktionalitet som hade varit avgörande för att göra applikationen mer lättanvänd är OCR. Idéen från start var att implementera en funktion där användaren med hjälp av OCR kunde skanna in kvitton och på så sätt få ut miljöpåverkan för dem produkter som inhandlats.

Tyvärr är det väldigt många parametrar som ska stämma för att detta ska vara möjligt. För det första ser kvitton olika ut beroende på matvarukedja. Kvitton kan även variera beroende på butik inom en viss kedja. För det andra var programvaran för att skanna in text helt enkelt inte tillräckligt bra för att hantera den typ av text som finns på kvitton. Bara att få det att fungera upplevdes som ett kandidatarbete i sig. Ytterligare en parameter var att enheten för mängden på en vara varierade. Hade slutligen inläsningen av en vara lyckats skulle det krävas att tolka det som läses in och koppla till Livsmedelsverkets databas för att få ut den faktiska påverkan på miljön. Mot den bakgrunden togs ett relativt snabbt beslut att inte fortskrida utvecklingen av den funktionen.

5.9.2 Tävlingsmoment

Tävlingsmomentet var under planeringen av projektet en av de viktigare punkterna just för att nå upp till ett av de mer kritiska målen; att få användaren att fortsätta utnyttja applikationen. Tävlingsmomentet skulle fungera som ett incitament för användaren att bruka applikationen över en längre period. Därför växte tanken fram att med hjälp av Google Play Game Services ge användaren möjligheten att skapa grupper där en tävling kunde genomföras mot kompisarna, till exempel "Vem kan förbruka minst koldioxid under en månad?". Incitamentet spelar på

vinnarinстинkten hos användaren och kräver således att detta är en drivkraft för användaren. Svårigheten med tävlingsmomentet var hur det skulle implementeras med den begränsade tid som fanns och valet att försöka utnyttja Googles API togs snabbt eftersom behovet av att konfigurera en server och kommunikation mellan användaren och servern försvann.

Det finns två färdiga spelmekanismer i Googles API; turordningsbaserat- samt realtidspelläge. Turordningsbaserat spelläge började implementeras med tanken att en förfrågan om att få göra ett drag (ladda upp användarens poäng i topplistan) skulle gå att göra och på så sätt komma runt problematiken med turordningen.

Tyvärr gick det inte göra ett sådant anrop via Google Play Game Services och alternativet blev då att implementera det själva vilket tog utvecklingen tillbaka till ruta ett. Arbetet fortskred likväl vidare med att implementera realtidspelläget vilket till en början fungerade bra. Först skapades ett väntrum där alla spelarna samlades som sedermera övergick till ett spelrum när alla hade anslutit. Allting gick att implementera men det slutliga och avgörande problemet var att kommunikationen mellan enheterna inte gick via en server utan skedde peer-to-peer vilket medförde att så fort en av användarna stängde ner applikationen (kopplade från Google Play Game Services) kopplades den enheten bort från spelet. Därför uteslöts tillslut även realtidspelläget.

5.9.3 Analytics

Det fanns en tanke att implementera Google Analytics för att i framtiden enkelt kunna utvärdera användarens beteende i applikationen. Andra funktioner som ansågs vara högre prioriterade blev viktigare och endast en början till implementering påbörjades.

Det som sattes upp var kontrollpanelen och ett loggande varje gång applikationen öppnades. Inget av detta togs dock vidare till utvecklingsgrenen.

5.9.4 Digitala kvitton

Efter misslyckandet med scanning av kvitton togs idén om att användaren skulle kunna koppla sitt konto till en av de större matvarukedjorna fram. Tanken var att utnyttja medlemskonton och kunna få tag i användarens kvitto den vägen. Det faktum att det fanns en liknande applikation för vissa klädesbutiker *sparakvittot* talade för idén. Tyvärr visade det sig att ingen av de större butikerna redovisade kvitton på respektive medlemsida, enbart den totala kostnaden per köp gick att se.

Kapitel 6

Slutsats

Syftet med kandidatprojektet var att skapa en applikation där användaren kan logga sitt ekologiska fotavtryck med avseende på livsmedel och transport. En första version av en sådan applikation är realiserad och fungerar till stor del som tänkt. Det kan dock konstateras efter användartester att en potentiell användare inte skulle använda applikationen i någon större utsträckning samt troligen inte heller rekommendera den för sina vänner. Detta beror med största sannolikhet på att användaren oftast inte har viljan till att förändra ohållbara beteenden när det i ett direkt skede inte gynnar en själv. I detta avseende har den resulterande applikationen misslyckats med att uppfylla en av utmaningarna, då ett av målen faktiskt var att få användare att börja samt fortsätta använda applikationen.

Sammantaget blev dock resultatet en fungerande prototyp och användartester genomfördes för att utvärdera de befintliga funktionerna samt att få upplysning om i vilken riktning en vidareutveckling av applikation bör ta.

Litteraturförteckning

- [1] Portions of this page are reproduced from work created and shared by the Android Open Source Project(<http://code.google.com/policies.html>) and used according to terms described in the Creative Commons 2.5 Attribution License at <http://creativecommons.org/licenses/by/2.5/>. [Online]. Available: <https://www.google.com/design/spec/patterns/navigation-drawer.html#navigation-drawer-content> [Hämtad: 2015-05-18]
- [2] Portions of this page are reproduced from work created and shared by the Android Open Source Project(<http://code.google.com/policies.html>) and used according to terms described in the Creative Commons 2.5 Attribution License at <http://creativecommons.org/licenses/by/2.5/>. [Online]. Available: <https://www.google.com/design/spec/what-is-material/objects-in-3d-space.html> [Hämtad: 2015-05-18]
- [3] Portions of this page are reproduced from work created and shared by the Android Open Source Project(<http://code.google.com/policies.html>) and used according to terms described in the Creative Commons 2.5 Attribution License at <http://creativecommons.org/licenses/by/2.5/>. [Online]. Available: <https://www.google.com/design/spec/layout/structure.html#structure-app-bar> [Hämtad: 2015-05-18]
- [4] Portions of this page are reproduced from work created and shared by the Android Open Source Project(<http://code.google.com/policies.html>) and used according to terms described in the Creative Commons 2.5 Attribution License at <http://creativecommons.org/licenses/by/2.5/>. [Online]. Available: <https://www.google.com/design/spec/patterns/navigation-drawer.html#navigation-drawer-content> [Hämtad: 2015-05-18]
- [5] “Ändra livsstil för miljöns skull!” 2002. [Online]. Available: <http://miljoforskning.formas.se/sv/Nummer/December-2002/Innehall/Notiser/Andra-livsstil-for-miljons-skull/> [Hämtad: 2015-05-18]

- [6] "Rekyleffekten och effektivitetsfällan," 2006. [Online]. Available: <http://www.naturvardsverket.se/Documents/publikationer/620-5623-9.pdf> [Hämtad: 2015-05-18]
- [7] "A successful git branching model," 2010. [Online]. Available: <http://nvie.com/posts/a-successful-git-branching-model/> [Hämtad: 2015-05-18]
- [8] "Beräkning av utsläpp från bilar," 2013. [Online]. Available: <http://www.utslappsrott.se/berakna-utslapp/berakning-av-utslapp-fran-bilar/> [Hämtad: 2015-05-31]
- [9] "The scrum guide," 2013. [Online]. Available: <http://www.scrumguides.org/scrums-guide.html> [Hämtad: 2015-05-18]
- [10] "Understanding adapters in android," 2013. [Online]. Available: <http://www.edureka.co/blog/what-are-adapters-in-android> [Hämtad: 2015-05-18]
- [11] "Färgblindhet hos barn," 2014. [Online]. Available: <http://www.1177.se/Vastra-Gotaland/Fakta-och-rad/Sjukdomar/Fargblindhet-hos-barn/> [Hämtad: 2015-05-18]
- [12] "Ökad oro för miljön - men beteendeförändringarna går långsamt," 2014. [Online]. Available: <http://yttra.se/okad-oro-miljon-men-beteendeforandringarna-gar-langsamt/> [Hämtad: 2015-05-18]
- [13] "Action bar," 2015. [Online]. Available: <https://developer.android.com/design/patterns/actionbar.html> [Hämtad: 2015-05-18]
- [14] "Android activity," 2015. [Online]. Available: <http://developer.android.com/reference/android/app/Activity.html> [Hämtad: 2015-05-18]
- [15] "Android database sqlite api," 2015. [Online]. Available: <http://developer.android.com/reference/android/database/sqlite/package-summary.html> [Hämtad: 2015-05-18]
- [16] "Android recyclerview api," 2015. [Online]. Available: <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html> [Hämtad: 2015-05-18]
- [17] "Android sqlitedatabasehelper," 2015. [Online]. Available: <https://github.com/jgilfelt/android-sqlite-asset-helper> [Hämtad: 2015-05-18]
- [18] "Android studio overview," 2015. [Online]. Available: <http://developer.android.com/tools/studio/index.html> [Hämtad: 2015-05-18]
- [19] "Androidplot," 2015. [Online]. Available: <http://androidplot.com> [Hämtad: 2015-05-18]

- [20] "Användbarhet & ux," 2015. [Online]. Available: <http://www.usabilitypartners.se/om-anvandbarhet/index.php> [Hämtad: 2015-05-18]
- [21] "Centralized workflow," 2015. [Online]. Available: <https://www.atlassian.com/git/tutorials/comparing-workflows/centralized-workflow> [Hämtad: 2015-05-18]
- [22] "comscore reports january 2015 u.s. smartphone subscriber market share," 2015. [Online]. Available: <http://www.comscore.com/Insights/Market-Rankings/comScore-Reports-January-2015-US-Smartphone-Subscriber-Market-Share> [Hämtad: 2015-05-18]
- [23] "Dashboards," 2015. [Online]. Available: <http://developer.android.com/about/dashboards/index.html> [Hämtad: 2015-05-18]
- [24] "Ekologiska fotavtryck," 2015. [Online]. Available: <http://www.wwf.se/vrt-arbete/hllbara-stder/ekologiska-fotavtryck/1514219-hllbara-stder-2-ekologiska-fotavtryck-2> [Hämtad: 2015-05-18]
- [25] "Endomondo features," 2015. [Online]. Available: <https://www.endomondo.com/features> [Hämtad: 2015-05-18]
- [26] "Endomondo-what we do," 2015. [Online]. Available: <https://www.endomondo.com/about> [Hämtad: 2015-05-18]
- [27] "Fragments," 2015. [Online]. Available: <http://developer.android.com/guide/components/fragments.html> [Hämtad: 2015-05-18]
- [28] "Git scm," 2015. [Online]. Available: <http://git-scm.com/> [Hämtad: 2015-05-18]
- [29] "Google analytics," 2015. [Online]. Available: <http://www.google.com/analytics/> [Hämtad: 2015-05-18]
- [30] "Google+ api," 2015. [Online]. Available: <https://developers.google.com/+api> [Hämtad: 2015-05-18]
- [31] "Google maps api," 2015. [Online]. Available: <https://developers.google.com/maps> [Hämtad: 2015-05-18]
- [32] "Google play services," 2015. [Online]. Available: <https://developer.android.com/google/play-services/index.html> [Hämtad: 2015-05-18]
- [33] "Material design," 2015. [Online]. Available: <http://www.google.com/design/spec> [Hämtad: 2015-05-18]

- [34] “Om 100koll,” 2015. [Online]. Available: <https://www.eon.se/privatkund/Produkter-och-priser/Elavtal/100Koll/Om-100koll/> [Hämtad: 2015-05-18]
- [35] “Pivotal tracker,” 2015. [Online]. Available: <http://www.pivotaltracker.com/why-tracker> [Hämtad: 2015-05-18]
- [36] “Processes and threads,” 2015. [Online]. Available: <http://developer.android.com/guide/topics/fundamentals/processes-and-threads.html> [Hämtad: 2015-05-29]
- [37] “Sundare liv med ica hälsorabatt,” 2015. [Online]. Available: <http://www.ica.se/appar/ica-halsorabatt/sa-funkar-appen/> [Hämtad: 2015-05-18]
- [38] “Support library features,” 2015. [Online]. Available: <https://developer.android.com/tools/support-library/features.html> [Hämtad: 2015-05-18]
- [39] “Sustainable business toolkit,” 2015. [Online]. Available: <http://www.sustainablebusiness toolkit.com/difference-between-co2-and-co2e/> [Hämtad: 2015-05-31]
- [40] “Testing support library,” 2015. [Online]. Available: <https://developer.android.com/tools/testing-support-library/index.html> [Hämtad: 2015-05-18]
- [41] “Utsläpp av växthusgaser från mat, transport och bostad från svensk privatkonsumtion,” 2015. [Online]. Available: <https://www.naturvardsverket.se/Samar-miljon/Statistik-A-O/vaxthusgaser-utslapp-av-svensk-privat-konsumtion/> [Hämtad: 2015-05-18]
- [42] “Zxing barcode scanner,” 2015. [Online]. Available: <https://github.com/zxing/zxing> [Hämtad: 2015-05-18]
- [43] C. J. Bergstad, “Miljöhandlingar ur ett miljöpsykologiskt perspektiv. betydelsen av attityder, värden och normer,” Föreläsning, 2014.
- [44] T. Broberg, “Rekyleffekten. Är energieffektivisering effektiv miljöpolitik eller långdistans i ett ekorhjul?” Konjunkturinstitutet, Tech. Rep., 2011.
- [45] S. Dave, “Life cycle assessment of transportation options for commuters,” Massachusetts Institute of Technology, Tech. Rep., 2010.
- [46] A. Dobson, “Environmental citizenship: Towards sustainable development,” *Wiley InterScience*, vol. 15, pp. 276–285, 2007.

- [47] V. Driessen, image used according to Creative Commons BY-SA found at <https://creativecommons.org/licenses/by-sa/2.0/>. [Online]. Available: <http://nvie.com/posts/a-successful-git-branching-model/> [Hämtad: 2015-05-18]
- [48] M. Eriksson, "Ekologiska fotavtryck... och hur många planeter skulle krävas om alla levde som du?" 2014. [Online]. Available: <http://www.wwf.se/vrt-arbete/ekologiska-fotavtryck/1127697-ekologiska-fotavtryck> [Hämtad: 2015-05-18]
- [49] P. L. Evert Andersson, "Energy consumption and related air pollution for scandinavian electric passenger trains," Department of Aeronautical and Vehicle Engineering, Royal Institute of Technology, KTH, Tech. Rep., 2006.
- [50] C. L. Fogelberg, "På väg mot miljöanpassade kostråd," Centrum för uthålligt lantbruk, Sveriges lantbruksuniversitet, Tech. Rep., 2008.
- [51] J. Lindberg, "Miljömedvetenhet, miljöansvar och miljöbeteende," Master's thesis, Göteborg Universitet, 2012.
- [52] E. Röös, "Mat-klimat-listan," Institutionen för energi och teknik, Sveriges lantbruksuniversitet, Tech. Rep., 2014.
- [53] K. S. . E. Söderberg, "Användargränssnitt för mobiltelefoner," 2000. [Online]. Available: <https://gupea.ub.gu.se/bitstream/2077/1391/1/ElisabethSoderberg.pdf> [Hämtad: 2015-05-18]
- [54] WWF, "Energi på hållbar väg," p. 2, 2007. [Online]. Available: <http://www.wwf.se/naturvaktarna/source.php/1154001/del6.pdf> [Hämtad: 2015-05-18]

Bilaga A

Användbarhetstester

Användbarhetstesterna kommer utföras på 20 personer . Användaren kommer få göra sig bekant med applikationen genom olika uppgifter. Om det är okej för användaren är det alltid ett plus att filma hela sessionen, annars utförs noga dokumentation av testet.

Tid: max 15 min

Vad mäts?: Tiden det tar att hitta en viss uppgift. Hur många back tracks som behövs för att uppnå ett mål.

Var?: Testerna ska helst göras i en miljö där användaren inte påverkas av diverse parametrar, dvs i en stängd miljö.

Övrigt: Ofta kan användaren köra fast, då finns det en metod som kallas think out load. Användaren säger då vad hen tänker är rimligt att utföra samt vad den tror kommer hända efter en viss tryckning. (Dokumentera detta.)

Uppgifter att utföra:

- Logga in via google+
- Lägg till en vara i din kombinerade inköpslista/livsmedelslogg.
- Lägg till en resa i efterhand som du åkt med en stadsjeep.
- (Lägg till en resa när du går)
- Titta på topplistan för transport

Uppföljningsfrågor:

1. Starta applikationen
 - (a) Var det enkelt att starta appen och komma till första sidan?
 - (b) Är menyn enkel att hitta?
 - (c) Vad tycker du om första sidan?
 - (d) Var något steg oklart?
2. Livsmedelsdelen
 - (a) Vad är ditt första intryck?

- (b) Hur upplever du att barcode-scanningen fungerar?
- (c) Vad tycker du om hur listan är presenterad?
- (d) Är det något du tror att du skulle använda?

3. Resedelen

- (a) Är det enkelt att förstå manuell och automatisk ?
- (b) Förstår man vad knapparna föreställer?
- (c) Vad tycker du mer specifikt om bilknapparna? Något du saknar?

4. Avslutande frågor

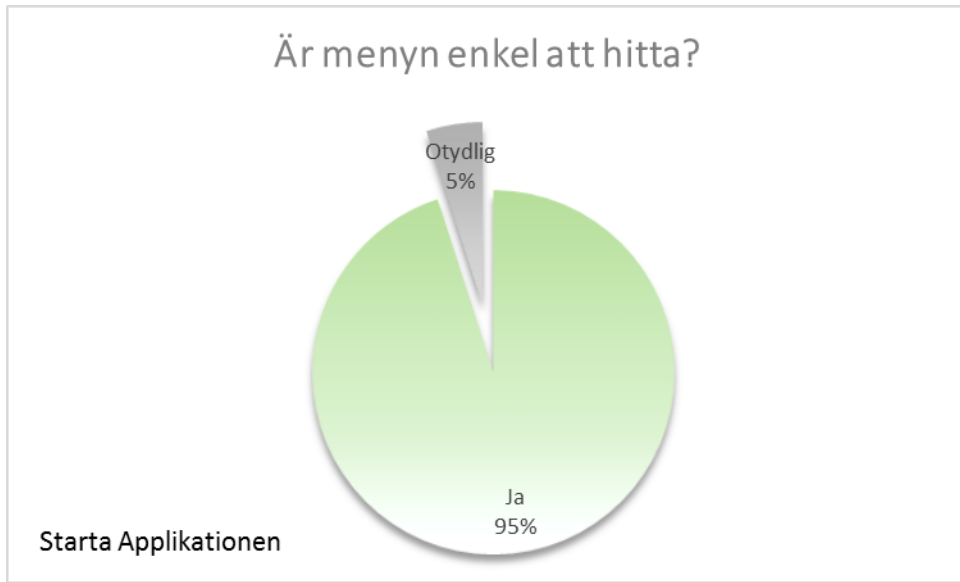
- (a) Vad är ditt generella intryck av applikationen? Skulle du använda den?
- (b) Något som du skulle vilja ändra?
- (c) Vad tyckte du särskilt om?
- (d) Vad tyckte du inte om?
- (e) Skulle du rekommendera den för dina vänner?
- (f) Tror du applikationen skulle göra dig mer medveten om hur "grön" du är?

Bilaga B

Resultat användbarhetstester



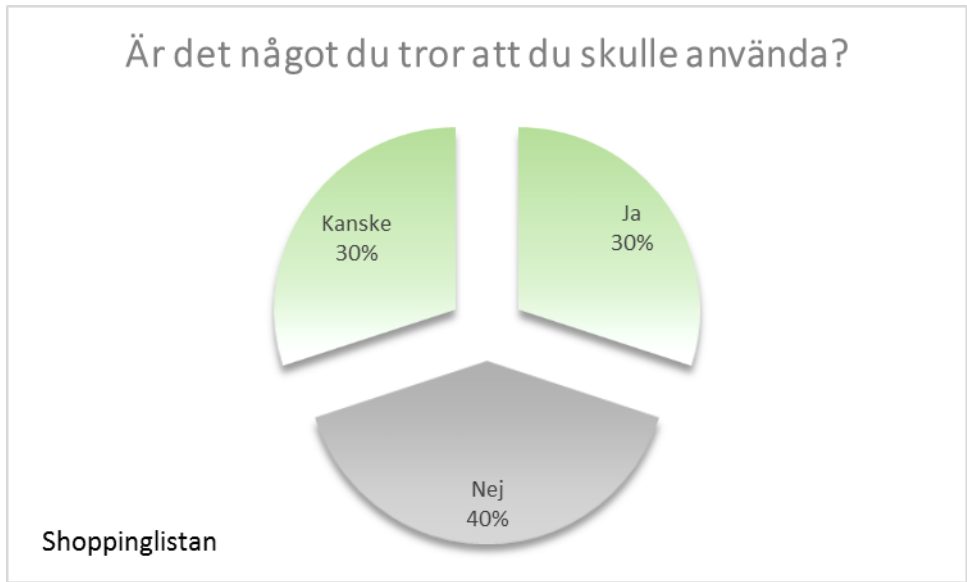
Figur B.1: Var det enkelt att starta appen och komma till första sidan?



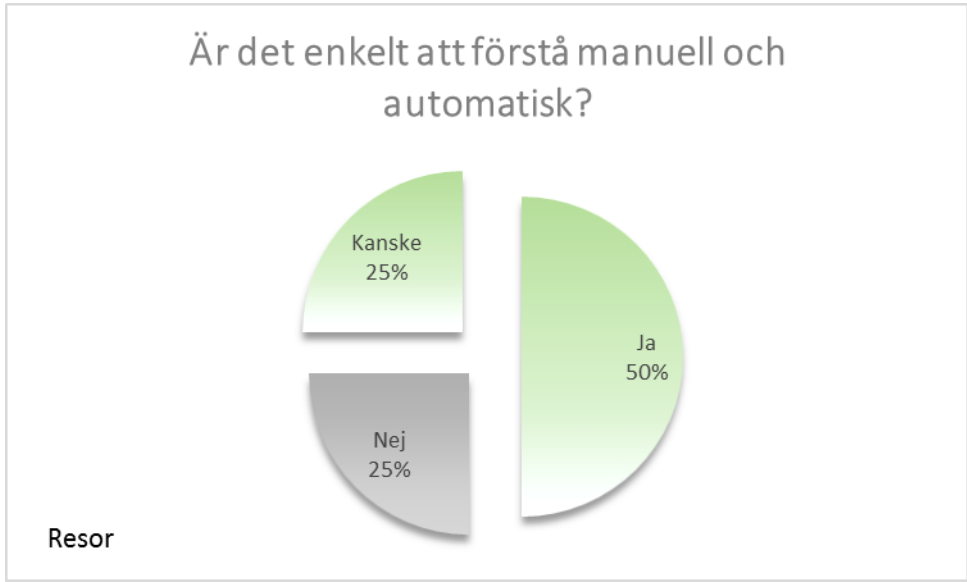
Figur B.2: Är menyn enkel att hitta?



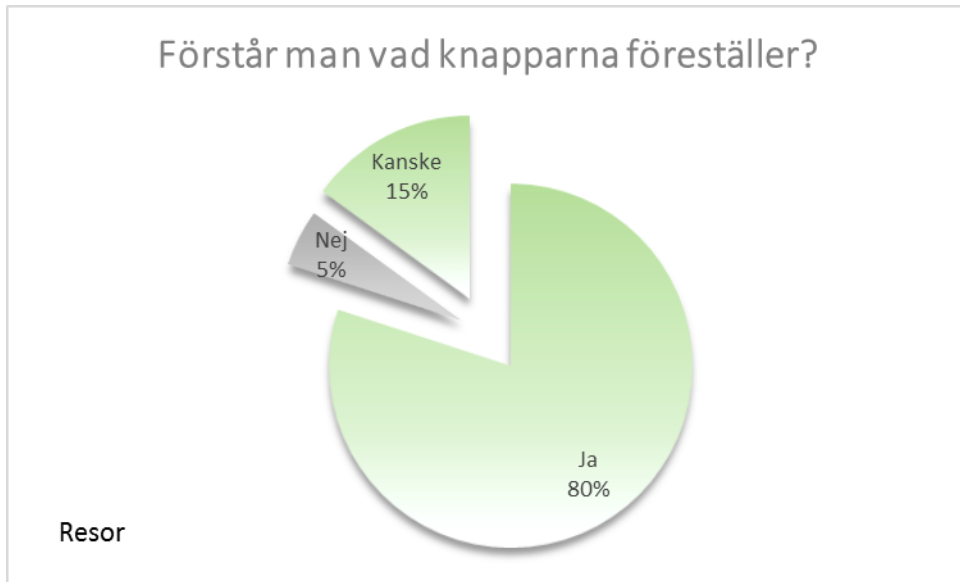
Figur B.3: Var något steg oklart?



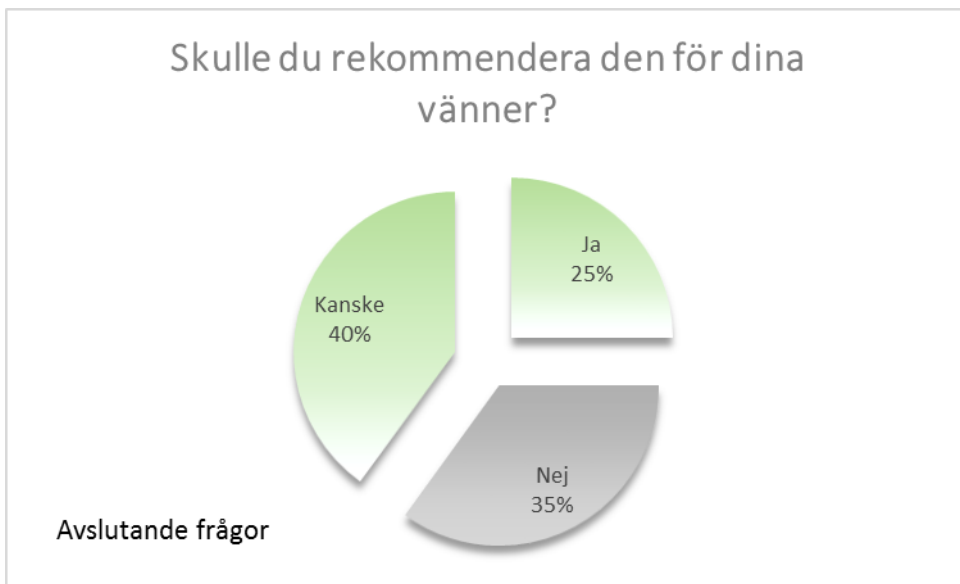
Figur B.4: Är det något du tror att du skulle använda?



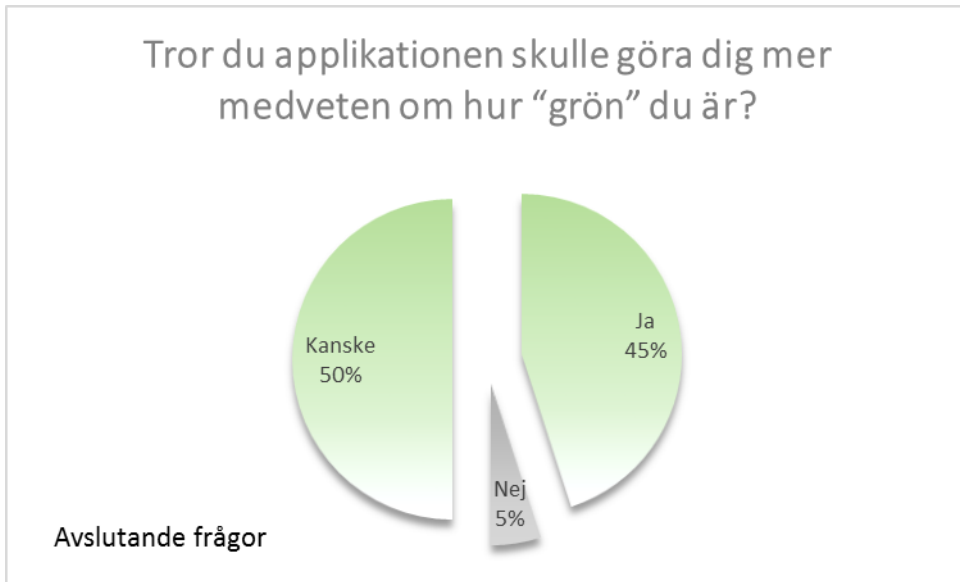
Figur B.5: Är det enkelt att förstå manuell och automatisk?



Figur B.6: Förstår man vad knapparna föreställer?



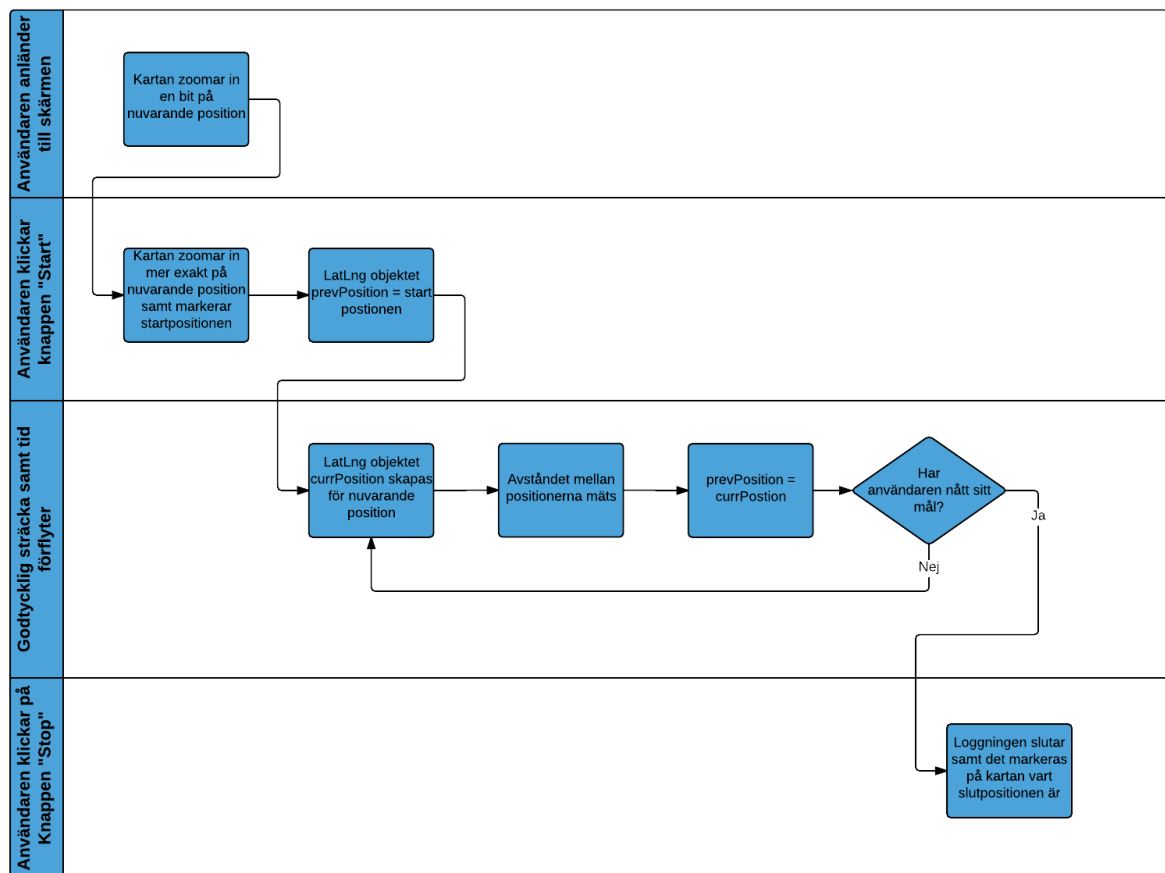
Figur B.7: Skulle du rekommendera den för dina vänner?



Figur B.8: Tror du att applikationen skulle göra dig mer medveten om hur "grön" du är?

Bilaga C

Flödesdiagram för transportsektorn



Figur C.1: Flödesdiagram för transportsektorn

Bilaga D

Använda bibliotek

I detta avsnitt beskrivs de bibliotek som användes under utvecklingen av GreenMe utöver Androids standardbibliotek.

D.1 AndroidPlot

AndroidPlot, ett öppet bibliotek som används för att måla ut flera olika varianter av grafer. Biblioteket har en del dokumentation och ger utrymme för mycket flexibilitet. [19]

D.2 AppCompatActivity

AppCompatActivity är ett bibliotek som tillåter äldre telefoner att använda Material design och ActionBar Design Pattern som ursprungligen bara stöds av Android 5.0 Lollipop och högre. [38]

D.3 Espresso och Testbibliotek

För att kunna göra effektiva funktionstester och enhetstester finns det hjälpbibliotek [40] att lägga till i sin utvecklingsmiljö. Dessa är dock inte inkluderade i själva applikationen utan kompileras lokalt vid behov. Espresso är ett bibliotek för att kunna testa flödet i användargränssnittet och för resterande tester användes JUnit anpassade bibliotek för Android.

D.4 Google Play Services

Detta bibliotek tillåter applikationen att använda Google Play tjänster via applikationen. Biblioteket behövs för att kunna utnyttja bl a Google Maps och Google+ [32].

D.5 RecyclerView

RecyclerView [16], en påbyggnad av ListView, användes i de fall som uppdateringsbara listor implementerades. Skillnaden mot ListView är att RecyclerView återanvänder de objekt som är i den och därav sparar på telefonens beräkningskraft.

D.6 SQLiteAssetHelper

SQLiteAssetHelper användes för att på ett enklare sätt kunna lägga in värden i databasen utan att behöva hårdkoda in dessa i någon Java-fil. Detta tillåter alltså förbestämda värden i databasen när applikationen blir installerad. [17]

D.7 v4 Support Library

v4 Support Library är ett bibliotek som tillåter alla Androidversioner över Android 1.6 Donut att stödja Fragments och Navigation Drawer som egentligen tillkom i Android 5.0 Lollipop. Biblioteket AppCompatActivity är även beroende av detta bibliotek. [38]

D.8 ZXing Barcode Scanner

ZXing Barcode Scanner är ett bibliotek för optisk läsning av streckkoder som finns tillgängligt för Androidplattformen. Genom att bara implementera ett bibliotek som är anpassad för Androidintegration behöver inte hela ZXing-biblioteket integreras i applikationen utan bara ett skal som drar nytta av en befintlig applikation som ZXing lagt upp på Google Play. Detta medför att plats kan sparas eftersom användaren inte behöver två kopior av ZXing. [42]