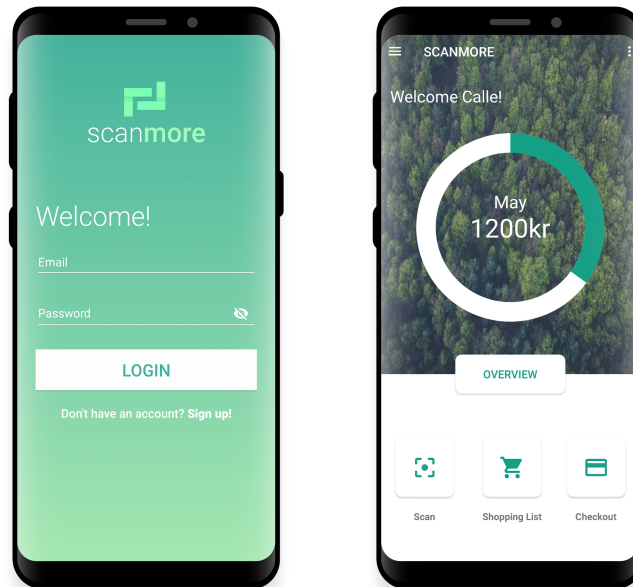




CHALMERS

UNIVERSITY OF TECHNOLOGY



Scanmore

Konceptutveckling av mobilbaserad skanningsapplikation

Examensarbete inom Data- och Informationsteknik

Carl Lidström Dreyer
Jenny Karlsson

EXAMENSARBETE 2019:06:12

Konceptutveckling av mobilbaserad skanningsapplikation

För att underlätta varuhusupplevelsen och främja teknisk
användning i samhället

Carl Lidström Dreyer & Jenny Karlsson



Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg 2019

Konceptutveckling av mobilbaserad skanningsapplikation

CARL LIDSTRÖM DREYER, JENNY KARLSSON

© CARL LIDSTRÖM DREYER, JENNY KARLSSON, 2019.

Handledare: Andreas Wieden

Examinator: Jonas Duregård

Institutionen för data- och informationsteknik

Chalmers Tekniska Högskola / Göteborgs Universitet

412 96 Göteborg

Telefon: +46 31 772 1000

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.

Omslag: Inloggnings- och startvy för den slutgiltiga applikationen.

Institutionen för Data- och Informationsteknik

Göteborg 2019

Förord

I denna rapport presenteras vårt examensarbete på Högscoleingenjörsprogrammet Datateknik, Chalmers Tekniska Högskola. Arbetet har genomförts av Carl Lidström Dreyer och Jenny Karlsson under vårterminen 2019. Arbetet motsvarar 15 högskolepoäng.

Vi skulle vilja tacka Conmore för att vi har kunnat utföra vårt examensarbete hos dem och vi vill även tacka alla på företaget som hjälpt oss på diverse sätt. Speciellt tack till Johan Svensson, vår handledare på Conmore och Andreas Wieden vår handledare från Chalmers.

- Carl Lidström Dreyer och Jenny Karlsson

Abstract

Grocery shopping nowadays may occasionally appear inconvenient, the lack of alternatives to the traditional costumer and cashier interaction remains inadequate. This report aims to investigate the demand for a mobile grocery scanner, by developing a concept application using the integrated development environment Android Studio. The finished application allows the user to create a customized profile, scan groceries and complete the transaction; all in the power of their hands. By allowing actual users to test the application, the user experience was greatly improved; yet the demand remains unproven. However, the test users expressed interest and satisfaction regarding the concept and developed application.

Keywords: proof-of-concept, Android Studio, barcode, groceries, application, user experience

Terminologi

- **Användargränssnitt** - Länk mellan användaren och hårdvaran/programvaran.
- **API** - Application Program Interface. Gränssnitt mellan applikation och mjukvarubibliotek.
- **App** - Applikationsprogram. Tillämpningsprogram som användaren själv kan ladda ner och installera.
- **Avlusare** - Ett programmeringsverktyg som används för att felsöka ett datorprogram.
- **Bugg** - En felaktighet i datorprogram som gör att programmet inte beter sig som förväntat.
- **Emulator** - Mjukvara avsedd för att efterlikna funktionen av annan hård- eller mjukvara, till exempel Android.
- **IDE** - Integrated Development Environment. Utvecklingsmiljö bestående av textredigerare, kompilator och avlusare.
- **iOS** - Operativsystem från Apple för användning i Apples mobila enheter iPhone, iPad och iPod touch.
- **Layout** - Användandet av typsnitt, logotyper och färger i digitala medier.
- **Molnbaserat** - IT-tjänster som tillhandahålls över internet.
- **Multiplattform** - Ett applikationsprogram som kan köras på flera plattformar.
- **MVP** - Minimum Viable Product. Minsta möjligt utvecklad produkt som används för att förklara idéer och göra dem mätbara.
- **Relationsdatabas** - Databas där informationen är organiserad i relationer, även kallade tabeller.
- **Simulera** - Återskapa en verklighet i en kontrollerad miljö.
- **SQL** - Structured Query Language. Standardiserat programspråk för att hämta och modifiera data i en relationsdatabas.
- **Syntax** - Regler som definierar huruvida en sammansättning av symboler anses vara korrekt blandade i ett programspråk.

Innehåll

1	Inledning	1
1.1	Bakgrund till problem	1
1.2	Syfte	1
1.3	Mål	2
1.4	Precisering av frågeställningen	2
1.5	Avgränsningar	2
2	Teknisk bakgrund	3
2.1	Android	3
2.2	Android Studio	3
2.3	SQLite	4
2.4	EAN-13	5
2.5	Git	5
2.6	Zebra Crossing	5
2.7	Scrum	6
2.8	Ad-hoc	6
2.9	Trello	6
3	Metodbeskrivning	7
3.1	Planering av arbetsmetodik	7
3.2	Val av teknologi för streckkodsskanning	7
3.3	Utvecklingsfasen	8
3.4	Testning och verifiering	8
3.5	Utformning av användartester	9
3.6	Användning av internetkällor	10
4	Konstruktion	11
4.1	Streckkodsskanning	11
4.2	Databashantering	12
4.3	Databasstruktur	13
4.4	User	14
4.5	CreditCard & Swish	14
4.6	Product	14
4.7	Visualisering av systemet	15
4.8	Systemkopplingar	16
4.9	Användargränssnitt	17

5	Resultat	19
5.1	Resultat från användartester	19
5.2	Resultat för precisering av frågeställningar	21
6	Slutsatser & Diskussion	23
6.1	Resultatsdiskussion	23
6.2	Mål	24
6.3	Lagring av personlig data	24
6.4	Diskussion av användartester	25
7	Fortsatt Arbete	27
7.1	Koncept till leverans	27
7.2	Ytterligare funktioner	29
A	Appendix 1 - Användartester	I
A.1	Förhandsformulär fråga 1	I
A.2	Följdfråga fråga 1	I
A.3	Förhandsformulär fråga 2	II
A.4	Förhandsformulär fråga 3	II
A.5	Följdfråga fråga 3	III
A.6	Förhandsformulär fråga 4	III
A.7	Följdfråga fråga 4	IV
A.8	Förhandsformulär fråga 5	IV
A.9	Följdfråga fråga 5	V
A.10	Testuppgift 1	V
A.11	Testuppgift 2	VI
A.12	Testuppgift 3	VI
A.13	Testuppgift 4	VII
A.14	Testuppgift 5	VII
A.15	Efterhandsformulär fråga 1	VIII
A.16	Efterhandsformulär fråga 2	VIII
A.17	Efterhandsformulär fråga 3	IX
A.18	Efterhandsformulär fråga 4	IX
A.19	Efterhandsformulär fråga 5	X
A.20	Efterhandsformulär fråga 6	X
A.21	Efterhandsformulär fråga 7	XI
A.22	Efterhandsformulär fråga 8	XI
A.23	Efterhandsformulär fråga 9	XI
A.24	Efterhandsformulär fråga 10	XII
A.25	Efterhandsformulär fråga 11	XII
B	Appendix B - Metoder	XIII
B.1	Gantt-schema som beskriver arbetsgången	XIII
B.2	Trellotavla som representerar en sprint	XIV
C	Appendix C - Applikationsvyer	XV
C.1	Inloggningsskärm	XV

C.2	Registreringsskärm	XV
C.3	Startskärm	XVI
C.4	Shoppinglista	XVI
C.5	Navigeringsmeny	XVII
C.6	Profilvy	XVII
C.7	Skanningsvy	XVIII
C.8	Skanningsvy med inskannad lista	XVIII
C.9	Borttagning av vara	XIX
C.10	Shopping Cart	XIX
C.11	Betalningsvy	XX
C.12	Tilläggning av nytt betalmedel	XX
C.13	Inmatning av kortnummer	XXI
C.14	Inmatning av kortnamn	XXI
C.15	Inmatning av utgångsdatum	XXII
C.16	Inmatning av CVV	XXII
C.17	Inlagt bankkort	XXIII
C.18	Inmatning av swishnummer	XXIII
C.19	Inlagt swishnummer	XXIV
C.20	Betala med bankkort	XXIV
C.21	Lyckad betalning med bankkort	XXV
C.22	Betala med Swish	XXV
C.23	Lyckad betalning med Swish	XXVI
C.24	Splash Screen	XXVI
C.25	Inmatning av felaktiga uppgifter	XXVII
C.26	Initial betalningsvy	XXVII
D	Appendix D - Databas källkod	XXIX
D.1	Product	XXIX
D.2	Databasehandler	XXX
D.3	Representation av tabeller i databasen med DB...	XXX

1

Inledning

I detta kapitel presenteras examensarbetets bakgrund samt uppsatta mål och syfte.

1.1 Bakgrund till problem

Butikskedjor erbjuder i dagsläget tjänsten att själv skanna varorna för att sedan slutföra köpet i en obemannad kassa. Medlemmar kan med hjälp av sitt medlemskort låna en fysisk skanner för att sedan registrera streckkoderna från de varor som ska införskaffas. När kunden har fullföljt sin inhandling lämnas skannern tillbaka i ett ställ och betalning kan slutföras via en terminal. I nuläget erbjuds inget alternativ att utföra skanningen på. Ett fåtal butiker tillåter kunderna att betala via mobilen, oftast med hjälp av tjänsten Swish.

Conmore AB är en ingenjörbyrå med inriktning mot fordonsindustrin [1], de har erbjudit oss en plats att utföra vårt examensarbete på. Conmore är intresserade av nya tekniska lösningar och idéer som inte endast rör deras huvudinriktning. Företaget har ingen tidigare erfarenhet av mobil varuskanning och projektidén presenterades av oss studenter. Varuskanning via mobilen används i delar av USA [2], fenomenet har inte kommit till Sverige ännu.

1.2 Syfte

Syftet med detta examensarbete är att utveckla en konceptapplikation, vars avsikt är att undersöka huruvida en mobil varuskanner är aktuell på den svenska marknaden.

1.3 Mål

Huvudmålen för detta examensarbete presenteras nedan. Målen ska bidra till att syftet uppnås.

- En konceptapplikation skall utvecklas
- Konceptapplikationen skall testas mot faktiska användare
- En undersökning kring behovet av applikationen skall utföras

1.4 Precisering av frågeställningen

Examensarbetet kommer att svara på följande frågor:

- Huruvida ett behov för mobil varuskanning existerar på den svenska marknaden
- Hur tillvägagångssättet för att skanna in en vara kan utformas
- Hur personliga uppgifter kan lagras säkert
- Vilka fördelar som finns med att utföra handlingsprocessen via en smartphone

1.5 Avgränsningar

Konceptapplikationens funktioner kommer begränsas baserat på given tidsram samt utvecklingspotential. De avgränsningar som valts är följande:

- Applikationen skall enbart stödjas av Android-baserade smartphones
- Begränsat antal avläsningsbara varor skall stödjas
- Ingen hårdvara skall utvecklas
- Köp i applikationen skall endast simuleras
- Skanningen av streckkoder skall ske med ett redan existerande mjukvarubibliotek

2

Teknisk bakgrund

I detta kapitel presenteras en teknisk bakgrund till de begrepp som är centrala för rapporten.

2.1 Android

Android är ett mobilt operativsystem främst utvecklat för smartphones och surfplattor. Operativsystemet bygger på en modifierad Linux-kärna med öppen källkodslicens. Detta innebär att koden bakom Android är öppen för allmänheten, därav har fristående operativsystem utvecklats med Android som grund. Androidbaserade enheter utgör i dagsläget 75% av världens mobila enheter [3]. Utvecklingen av en androidbaserad applikation är i regel snabbare, enklare samt mer kostnadseffektivt, gentemot att utveckla en mot operativsystemet iOS. De huvudsakliga programmeringsspråken för Android är Java och Kotlin och för att bygga användargränssnittet används XML [4].

2.2 Android Studio

Android Studio är en integrerad utvecklingsmiljö skapad för applikationsutveckling mot operativsystemet Android. Applikationer startar som projekt i Android Studio, dess funktioner representeras i form av Java-filer. Android Studio tillåter användaren att bygga programkoden till en färdig applikation, för att sedan installera den på antingen en emulator; eller kompatibel androidenhet som behöver vara kopplad till utvecklarens dator via USB [5].

2.2.1 Aktiviteter

Vid kodning av en applikation i Android Studio skapas aktiviteter, en aktivitet består av en Java-fil med tillhörande XML-fil. Java-filen består av den kod som hanterar aktivitetens logik, medan XML-filen behandlar aktivitetens visuella aspekter. Exempel på filbeteckningar kan vara MainActivity.java med tillhörande activity_main.xml.

2.2.2 Java

Java är ett objektorienterat programspråk med aktivt användande sedan 1995 då det skapades. Java brukas idag inom ett flertal olika användningsområden, bland annat androidbaserade mobilapplikationer, webbaserade applikationer och spel. Programspråket är väletablerat och är i nuläget ett av de största språken inom utvecklingssektorn. Java kan köras på diverse olika plattformar som Windows, Mac, Linux och Raspberry Pi [6].

2.2.3 Getters & setters

I objektorienterad programmering är ett utav syftena att olika objekt ska kunna samarbeta med varandra. Därav dela egenskaper som instansvariabler. Dessa variabler är endast tillgängliga inom en klass, såvida inte publika metoder skapas som gör det möjligt att hämta dem. Dessa metoder kallas för getters och setters. En get-metod returnerar instansvariabler medan en set-metod ändrar värdet på variablerna. Detta gör det möjligt att hämta redan skapade variabler istället för att definiera nya [7].

2.2.4 XML

XML står för eXtensible Markup Language och är ett universellt märkspråk. Språkets syfte är att utväxla data mellan informationssystem, vilket utförs genom att ren text skickas. Android Studio använder framförallt XML för att skapa grafiska komponenter [8].

2.3 SQLite

SQLite är ett programvarubibliotek utvecklat i programspråket C. Biblioteket används för att implementera en filbaserad SQL-databas. Begreppet filbaserad innebär att databasen lagras lokalt i form av en fil. Denna fil är flyttbar vilket betyder att man endast behöver flytta filen vid transport av databasen. SQLite stödjer de flesta standardiserade SQL syntaxerna, hursomhelst utelämnas vissa medan andra tillkommer specifika för SQLite [9].

2.4 EAN-13



Figur 1. Exempel på streckkoden EAN-13.

EAN-13 är en streckkodstyp som används för att märka konsumentförpackningar med ett artikel- eller viktvarunummer. Denna typ av streckkod är vardaglig och påträffas praktiskt taget på alla konsumentprodukter. EAN-13 är byggd av streck och mellanrum i fyra olika bredder. Streckkoden kan läsas av i flera olika vinklar eftersom EAN-13 består av två delar, en vänster- och högerdel. EAN-13 kräver även ett tomt område runt om koden, detta kallas för ljusmarginal och har som funktion att produktförpackningen inte skall störa avläsningen [10].

2.5 Git

Git är ett distribuerat versionshanteringssystem med öppen källkodslicens. Git erbjuder möjligheten att skapa en molnbaserad förvaringsplats där ett projekts filer kan lagras. Projektets medlemmar har därmed möjlighet att ladda ner filerna till en lokal version på sina egna datorer. Den lokala versionen kan därefter modifieras för att sedan skickas tillbaka till den molnbaserade versionen. Git tillåter även användarna att återställa projektet till en tidigare version ifall problematik uppstår [11].

2.6 Zebra Crossing

Zebra Crossing, ZXing förkortat, är ett bildbehandlingsbiblotek för hantering av diverse streckkoder. ZXing har öppen källkodslicens och är utvecklat för att integreras i Androidapplikationer byggda på programspråket Java. Biblioteket stödjer ett flertal olika koder för optiskt avläsning, bland annat endimensionskoder som EAN-13; men även tvådimensionskoder som QR [12].

2.7 Scrum

Scrum är ett agilt arbetssätt som delar upp projekts gång i tidsperioder kallade sprintar. Längden på dessa sprintar anpassas till projektets storlek, huvudsyftet med dessa är att i slutet besitta en produkt som skapat värde för produktägaren. Alla möjliga uppgifter för projektet antecknas och rangordnas i produktbackloggen, en samlingsplats för alla idéer och möjligheter som påtänks; oavsett huruvida de planeras att realiseras under projektets gång. I början av varje sprint hålls en planering där uppgifter från produktbackloggen selekteras, för att sedan utföras under nästkommande sprint. Varje dag håller medlemmarna ett möte för att diskutera vad som ska vara klart, är klart och eventuella problem som stötts på. I slutet av sprinten hålls slutgiltigen ett möte kallat retrospektiv där demonstration samt reflektion av den gångna sprinten infaller [13].

2.8 Ad-hoc

Ad-hoc är en testmetodik med avsaknaden utav plan samt dokumentation. Operationsflöden utförs på måfå med syfte att påträffa buggar i systemet. Testet är avsett att endast köra en gång om inte någon bug upptäcks. Ad-hoc är den minst formella testmetoden, på grund av detta har den blivit kritiserad för att vara alldeles för ostrukturerad [14].

2.9 Trello

Trello är en webbaserad projektorganiseringsappliktion som används för att organisera och strukturera projekt. Trello kan användas av flera personer för att samarbeta med varandra. I Trello kan anslagstavlor skapas med listor samt arbetskort, se bilaga B.2. Arbetskorten kan exempelvis innehålla uppgifter, händelser, möten och ärenden, korten går att placeras om för att visa framsteg. På dessa kort läggs medlemmar till som jobbar med tillhörande uppgifter. Detta underlättar synligheten för vem eller vilka som ansvarar för de olika korten. Anteckningar och checklistor kan även läggas till på korten, för att representera en kravspecifikation. Samtliga aktiviteter på Trello sker i nutid vilket underlättar vid gemensamt arbete [15].

3

Metodbeskrivning

I detta kapitel presenteras examensarbets metodik. Inledningsvis redogörs planeringen av arbetet. Följt av utvecklingsfasen samt hur testning och verifiering kommer att utföras. Till sist presenteras utformningen av användartesterna.

3.1 Planering av arbetsmetodik

Baserat på tidigare erfarenheter inom projektutförande kommer arbetsmetodiken Scrum att användas. Alla beståndsdelar av det agila arbetssättet kommer ej att brukas, detta på grund av den begränsade tidsramen. Inledningsvis kommer alla tänkbara funktioner antecknas i produktbackloggen, därefter kommer de rangordnas baserat på prioritet. Varje sprint kommer att pågå under två veckor, de uppgifter som skall utvecklas under varje sprint kommer att väljas enligt framtaget gantt-schema; se bilaga B.2. Produktbackloggen och tavlor för varje sprint kommer att organiseras i listtjänsten Trello. För att lättare arbeta gemensamt i projektets programfiler kommer versionshantering användas. En gemensam förvaringsplats för filerna kommer att skapas i webbhotellet GitHub, som bygger på versionhanteringsteknologin Git.

3.2 Val av teknologi för streckkodsskanning

För att kunna läsa av en konsumentprodukts EAN-13 kod krävs det ett mjukvarubibliotek. Biblioteket måste kunna behandla streckkoder i realtid och returnera dess värde. Innan utvecklingen av applikation påbörjar kommer en undersökning göras, med syfte att identifiera vilket bibliotek som passar projektet. Undersökningen kommer att utföras genom att möjliga redan utvecklade bibliotek skall upptäckas för att sedan jämföras med varandra, detta för att se huruvida kravet på EAN-13 skanning i realtid uppfylls.

3.3 Utvecklingsfasen

Utvecklingen av mjukvara kommer att ske i utvecklingsmiljön Android Studio, och utformas i programspråket Java. Valet av denna utvecklingsmiljö baserades på erfarenheter från tidigare kurser, där programvaran använts. Utvecklingsmiljön medför därav ingen inlärningsprocess, vilket resulterar i en kortare utvecklingsfas. Innan utvecklingen av applikation påbörjas kommer att tomt projekt skapas i Android Studio. Programutvecklingen kommer först fokusera på att uppnå en applikation som med framgång kan läsa av en produkt, presentera varans information i det grafiska användargränssnittet samt simulera en betalning av köpet.

Eftersom den mest centrala delen i projektet är skanningen av konsumentprodukter, kommer denna funktion utvecklas först. Första steget kommer vara en kameravy som framgångsrikt läser av streckkoden på en vara och returnerar värdet. En databas kommer sedan utvecklas med hjälp av Android Studios inbyggda databashanterare SQLite. Databasen kommer innehålla information om: produkterna, användarna och deras betalmedel. För att simulera ett helhetsperspektiv för applikationen kommer personliga profiler utvecklas. Möjligheten för en ny användare att registrera sig och logga in kommer att framgå.

3.4 Testning och verifiering

Applikationen kommer under utvecklingen kontinuerligt testas med hjälp av ad-hoc metoden. Detta kommer att utföras samtidigt somloggning i konsollen placeras ut i specifika delar av koden, detta för att notera huruvida delar utav koden exekveras eller ej. Testningen kommer att utföras på en androidbaserad telefon, och därmed ej med hjälp av en emulator. Användartester kommer att utföras i syfte att fastställa behovet av projektet, samt förse utvecklingen med återkoppling till det grafiska användargränssnittet, och behovet för utveckling av nya funktioner.

3.5 Utformning av användartester

Testerna kommer att utföras på Conmores kontor, deltagarna kommer främst vara anställda från företaget. Användartesterna kommer delas in i fyra delar: inledning, förhandsformulär, uppgifter att utföra i applikationen och efterhandsformulär. I samband med användartesterna kommer användarnas tankar och funderingar antecknas i ett Google Forms formulär.

3.5.1 Inledning

Under inledningen kommer arbetet presenteras och syftet med testet förklaras. Testpersonen kommer sedan informeras om den estimerade testtiden och hur testet kommer att gå till.

3.5.2 Förhandsformulär

Inledningsvis kommer nedanstående frågor att ställas för att skapa en övergripande bild om användarens handelsvanor.

1. Brukar du handla din mat online eller i butiker?
2. Hur ofta brukar du handla per vecka?
3. Många butikskedjor som t.ex. Coop erbjuder möjligheten att du själv kan skanna dina varor m.h.a. en handskanner. Är detta något du brukar använda dig av?
4. Våldigt få butiker tillåter kunderna att betala enkelt via mobilen. Är detta något du hade varit intresserad av?
5. Hade du varit intresserad av att sköta hela processen via din mobiltelefon? Dvs. att du själv scannar produkterna och sedan betalar i mobilen.

3.5.3 Testuppgifter

Under testets gång kommer användaren få utföra följande uppgifter för att förstå hur personen tänker och resonerar.

1. Börja med att skapa ett användarkonto och logga in i applikationen. Navigera sedan till din personliga profil.
2. Du ska nu inhandla följande: En snusdosa lundgrens och två stycken Conmore vatten. Hur gör du för att skanna in dessa produkter?
3. Testa nu att skanna in snusdosan en gång för mycket, för att sedan ta bort den från din inköpslista.
4. Det är nu dags att betala, navigera till betalningsvyn och slutför köpet m.h.a. ett kreditkort.
5. Du kom nu på att du behövde en till vattenflaska, skanna in produkten och slutför sedan köpet m.h.a av Swish och logga sedan ut från applikationen.

3.5.4 Efterhandsformulär

Efter att användaren utfört ovanstående uppgifter kommer följande frågor ställas i syfte att utvärdera användarens upplevelse.

Frågor med 1-5 svar:

1. Jag skulle kunna tänka mig att använda det här systemet i min vardag.
2. Jag tyckte att applikationen var svår att använda.
3. Jag tror att dom flesta personerna snabbt skulle lära sig hur man använder applikationen.
4. Jag tyckte att dom olika funktionerna i applikationen var välintegrerade.
5. Jag kände mig säker när jag använde systemet.
6. Användargränssnittet var lätt att förstå.

Frågor med öppna svar:

1. Var det någon funktion du skulle vilja ändra?
2. Är det någon funktion du skulle vilja lägga till?
3. Var det någon funktion du skulle vilja ta bort?
4. Hade du velat ändra på något i användargränssnittet?
5. Tycker du det hade behövts mer hjälp för att komma igång med applikationen?

3.6 Användning av internetkällor

Under projektets gång kommer utöver skriftliga källor även digitala källor användas, eftersom det oftast tar längre tid för fakta att etableras och publiceras i skriftlig form. Interkällor erbjuder en snabbtillgänglig och i många fall giltig informationskälla. Den specifika källans validitet testas genom att pröva metodiken som föreslås, om det fungerade används den angivna lösningen.

4

Konstruktion

För att realisera projektet och skapa något med potential för vidareutveckling, krävs ett system som tillfredsställer de uppsatta målen.

4.1 Streckkodsskanning

Tidigare i rapporten presenteras skillnaden mellan en traditionell streckkodsskanner och en kamerabaserad. Eftersom en gedigen mobilkamera saknar hårdvara för inläsning av streckkoder, behövs ett mjukvarubibliotek med bildbehandling. En undersökning gjordes där olika alternativ togs fram som tillfredsställde det satta kravet, att behandla EAN-13 koder i realtid. Två teknologier påvisade denna egenskap. ML Kit's barcode scanning API, ett API utvecklat av Google, och Zebra Crossing; ett privatutvecklat bibliotek. ML Kit's barcode scanning API saknade dokumentation för implementeringen av realtidsskanning. ML Kit är främst utvecklat för streckkodsläsning från redan tagna bilder, och stödjer ej inläsning från smartphonsens kamera; en egenskap som Zebra Crossing stödjer och främst är utvecklad för. Därav gjordes valet att implementera Zebra Crossing för skanningen av EAN-13 koder.

Det färdiga biblioteket kunde utan komplikationer implementeras och modifieras för att tillfredsställa våra krav. Biblioteket skannade ursprungligen kontinuerligt vilket innebär att en vara fortsätter att skannas in tills att streckkoden lämnar kamerans synvinkel. Genom att implementera en knapp som aktiverar skanningen kunde detta undvikas. För att en streckkod skall läsas av krävs det att den placeras i vyns dedikerade skanningsområde, se bilaga C.7. När detta sker läser bildbehandlingsbiblioteket av streckkodens värde och returnerar detta i form av ett tal, i projektets fall representerar detta EAN-13 koden. Denna kod jämförs därefter med produkternas koder i databasen för att fastställa huruvida produkten är en del av sortimentet eller ej.

4.2 Databashantering

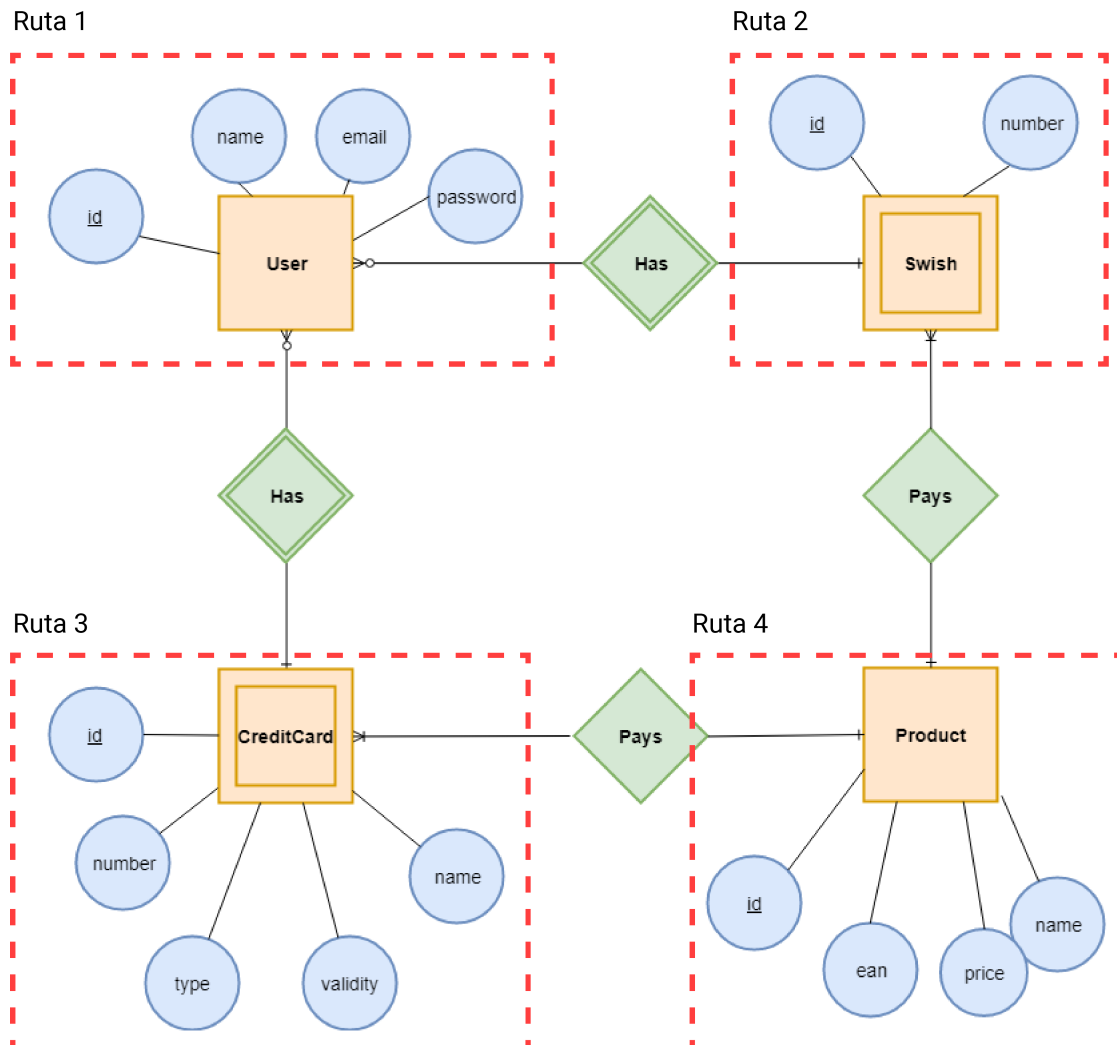
I projektet används SQLite som är baserat på SQL. För att göra det möjligt att skapa en databas krävs det att en översättning sker från Javas syntax till SQL. I Android Studio finns det redan klasser som åstadkommer detta, vilket är förklaringen till varför just SQLite är en fördel att använda vid arbete med Android Studio. De klasser som används vid databashantering kallas 'SQLiteOpenHelper' och 'SQLiteDatabase'. SQLiteOpenHelper hjälper till att skapa databasen och denna ärvs av 'DatabaseHandler.java', en klass som skapats i projektet och illustreras i bilaga D.1. SQLiteDatabase har bland annat metoder för att skapa eller ta bort tabeller samt exekvera SQL-syntax. I samband med implementationen genereras ett abstrakt lager mellan den underliggande SQLite-databasen och klassen [16].

Databashantering kan samläggas i flera klasser där varje del utför olika uppgifter. För att kunna definiera en databas skapades klassen DatabaseHandler.java (kodsnotat av denna kan observeras i bilaga D.2). Klasserna Product, CreditCard, Swish och User, se figur 2, gör det möjligt att på ett strukturerat sätt implementera interaktionen mellan databashanteraren och objekten, se bilaga D.1 & D.2. Bilaga D.1 illustrerar ett kodavsnitt på produktklassen. I denna klass har en sträng kallad CREATE_TABLE instansierats. Denna sträng innehåller ett par SQL-kommandon som skapar en tabell i databasen när databasklassen används. I bilaga D.2 presenteras ett kodavsnitt från databashanterarklassen. Denna har en metod som heter onCreate() vilket i sin tur skapar tabeller för varje objekt med metoden execSQL() som finns i klassen SQLiteDatabase.

Interaktionen mellan objekten och databashanteraren ger upphov till en fil i applikationen med .db format (db står för database). Filen namnges i databashanterarklassen med instanstilldelningen *'private static final String DATABASE_NAME = 'jenny' '*. Filen har sedan med hjälp av ett program som kallas 'DB Browser for SQLite' visuellt visat datan som finns i databasen. Här kan de tabellerna som skapas åskådas när databasen initieras. Denna fil går att hitta lokalt i Android Studio och påvisas i bilaga D.3.

4.3 Databasstruktur

I figur 2 visas en visuell representation av hur relationsdatabasen är uppbyggd. Kvadraterna representerar databasentiteter med sina attribut i blått. De gröna symbolerna representerar vad för typ av relation entiteterna har med varandra.



Figur 2. Övergripande bild av den använda databasens uppbyggnad.

4.4 User

Applikationen kräver ett personligt användarkonto för att kunna starta, detta innebär att användarens uppgifter vid registreringstillfället behöver lagras, eftersom de inmatade inloggningsuppgifter måste autentiseras gentemot databastabellen innehållande aktiva användare. Syftet med databasentiteten 'User', se figur 2, ruta 1, är att lagra den nya användarens namn, e-postadress och lösenord. Detta sker när användaren registrerar sitt konto och kontroll utförts att användarens valda e-post inte redan är registrerad.

4.5 CreditCard & Swish

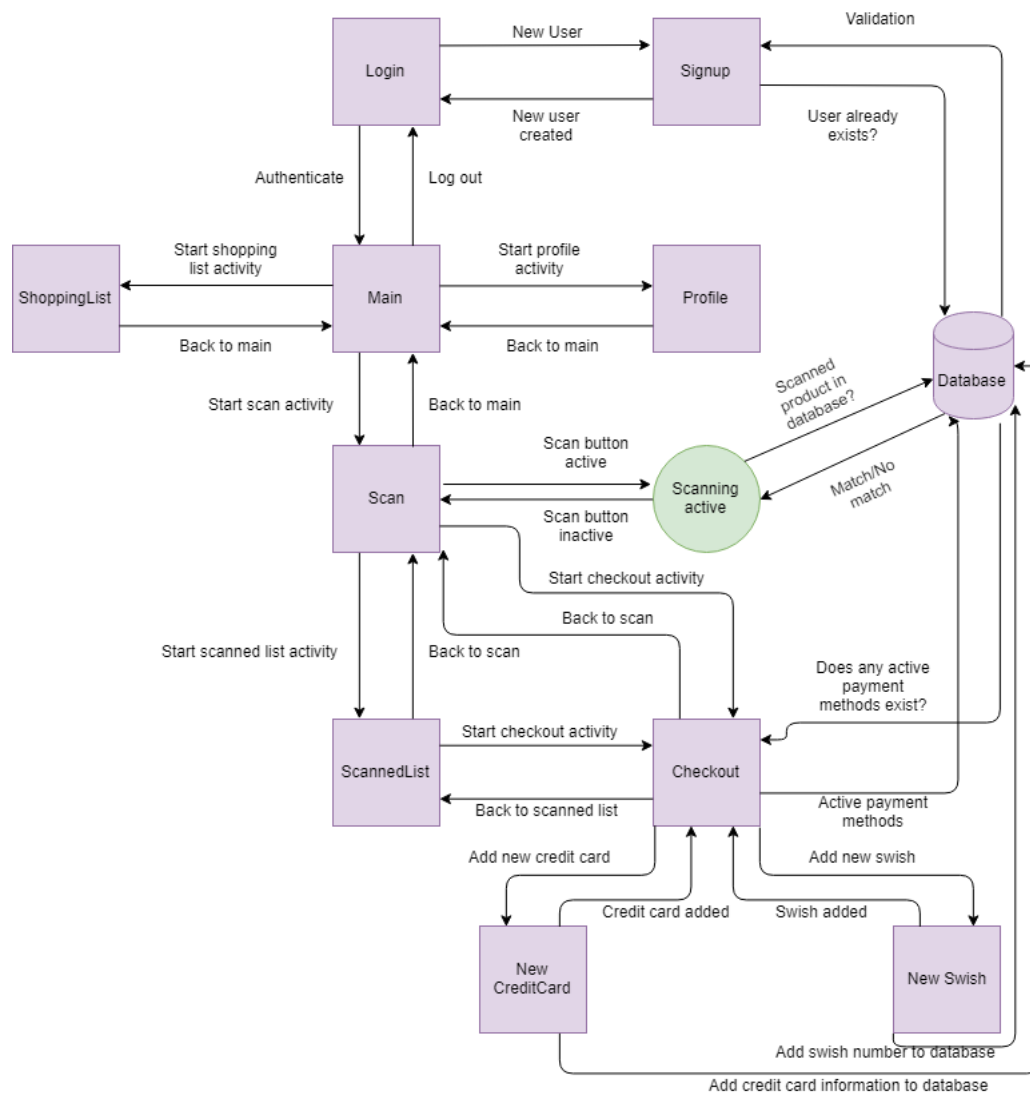
För att slutföra ett köp krävs ett giltigt betalmedel. Projektet är avgränsat till två typer, bankomatkort och Swish, se figur 2, ruta 2 & 3. Användaren presenteras med möjligheten att lägga till och spara dessa betalmetoder för att effektivisera framtida köp. När ett nytt bankomatkort registreras i applikationen sparas uppgifterna i databasentiteten 'CreditCard', innehållande kortnummer, namn, utgångsdatum och CVV-kod. När ett betalmedel av typen Swish läggs till sparas det angivna telefonnumret i databasentiteten 'Swish'.

4.6 Product

Databasentiteten 'Product', se figur 2, ruta 4, består av ett simulerat sortiment av konsumentvaror. Varorna införs manuellt vid instansieringen av databasen. En vara skapas med följande egenskaper: namn som beskriver varan, EAN-13 kod som återfinns på den verkliga varans streckkod och pris.

4.7 Visualisering av systemet

Applikationen innehåller ett stort antal klasser, nedanstående figur presenterar de mest centrala delarna. Utöver dessa finns ett antal hjälpklasser med syfte att underlätta huvudklassernas arbetslast. Klasserna representeras av lila kvadrater, pilarna förklarar kopplingarna mellan dem och den gröna cirkeln föreställer att skanningen är aktiv.



Figur 3. Övergripande visualisering av systemkomponenternas kommunikation.

4.8 Systemkopplingar

Vid start av applikationen presenteras användaren med en inloggningskärm, se bilaga C.1. Ifall användaren redan har ett konto kan hen logga in här. Inloggningsaktiviteten är kopplad till databasen innehållande registrerade användare. När inloggningsknappen aktiveras kontrollerar applikationen huruvida uppgifterna är korrekta eller ej. Ifall användaren ej har skapat en konto tidigare, kan detta göras genom att navigera till registreringsvyn; se bilaga C.2. Registreringsvyn kontrollerar så att den angivna e-postadressen inte är upptaget samt ifall ett godkänt lösenord angivits. När alla uppgifter stämmer sparas dem i databasens 'User' tabell. Vid en lyckad inloggning dirigeras användaren om till startskärmen, se bilaga C.3. Knapparna 'Scan', 'Shopping List' och 'Checkout' startar respektive aktivitet. Shoppinglistan är en fristående aktivitet, det vill säga icke kopplad till databasen eller utomstående klasser, se bilaga C.4. Via navigationsmenyn, se bilaga C.5, på startskärmen når användaren sin personliga profil. Profilvervyn använder sig av den inloggades uppgifter och presenterar dessa, se bilaga C.6. Via navigationsmenyn eller genvägen 'Scan' når användaren varuskanningsvyn, den centrala delen i detta projekt, se bilaga C.7.

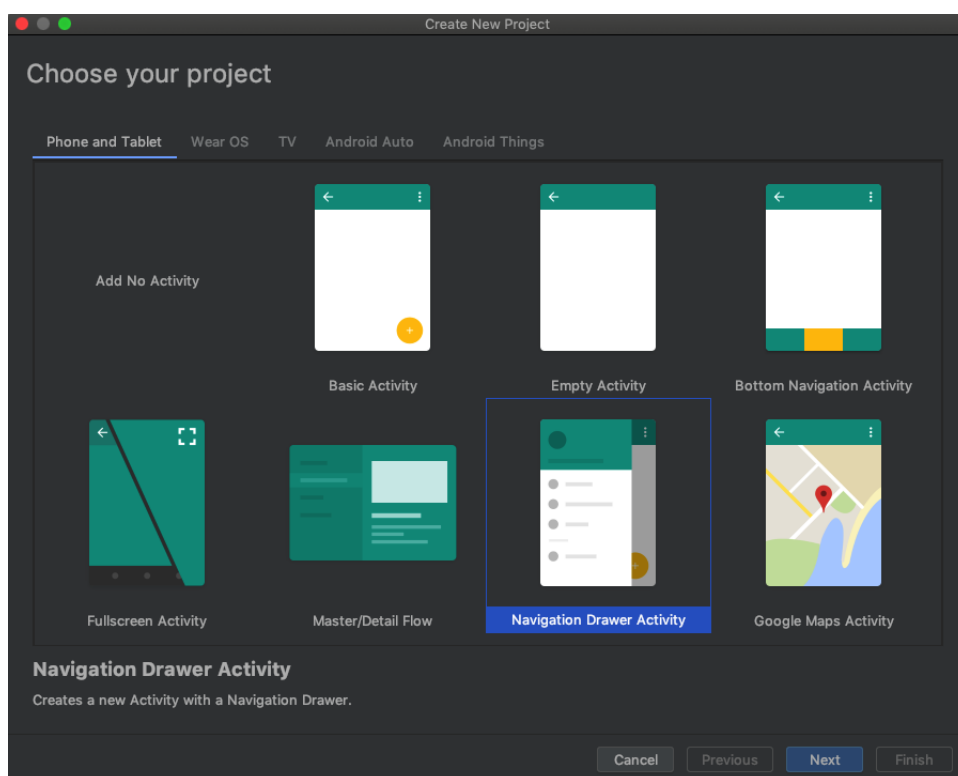
Skanningsvyn skapar en tom lista för de produkter som skall skannas in. När skanningsknappen är aktiverad söker bildhanteringsbiblioteket efter streckkoder inom den dedikerade avläsningsrutan. När biblioteket läst av ett värde hämtas en sortimentslista på de inlagda produkterna. Listan itereras genom och jämförs med det avlästa resultatet. Vid en lyckad matchning aktiveras telefonens vibrationsfunktion och den avlästa produkten förs in i listan över inskannade produkter, se bilaga C.8. Användaren presenteras med möjligheten att ta bort inskannade varor, se bilaga C.9. Listan över inskannade produkter återfinns i bilaga C.8, C.10 och C.11. I skanningsvyn visas de tre senaste inskannade produkter, med möjlighet att svepa upp eller ner för att se resterande varor. Genom 'visa mer'-knappen, representerad av tre horisontella punkter, når användaren vyn 'Shopping Cart', se bilaga C.10. Här ges användaren en större och mer lättläst version av listan för inskannade produkter, som hämtas direkt från skanningsvyn. Vyn innehåller även en knapp med titeln 'Checkout' som för användaren till utcheckningsvyn där betalningen sker, se bilaga C.11.

Utcheckningsvyn består av tre huvudkomponenter, en lista över inskannade produkter som hämtas från skanningsvyn, en lista över tillgängliga betalmedel som skapas lokalt samt en knapp som tillåter användaren att lägga till nya betalningsmetoder. I fallet vid återkommande användare kan tidigare använda betalmedel existera, i detta fall behöver användaren endast klicka på det valda betalmedlet för att slutföra köpet. Bilagor C.19 till C.22, visar hur slutförandet av köpet presenteras för bankkorts- respektive swishbetalningar. Ifall användaren inte redan lagt till ett betalmedel kan detta göras genom knappen med titeln 'Add a new payment method', denna knapp aktiverar dialogen som visas på bilaga C.12. Bankkortsikonen dirigerar användaren till vyn för att lägga till ett nytt bankkort, se bilagor C.13 till C.16. Swishikonen navigerar användaren till vyn för att lägga till ett nytt nummer för swishbetalningar, se bilagor C.18 till C.19. När den nya betalningsmetoden väl är

tillagd, sparas den i databasen under tabellen 'CreditCard' för bankkort och 'Swish' för swishnummer. Betalmedlet presenteras även i listvyn, se bilaga C.17, möjligheten att ta bort betalmedlet existerar här, där den både tas bort från listan samt databasen.

4.9 Användargränssnitt

En applikations utseende är ibland avgörande för om en användare faktiskt tar beslutet att använda appen. Därför är det värt att beskriva hur designval genomförs under utvecklingsfasen.



Figur 4. Val av nytt projekt

Till och börja med öppnades ett nytt tomt projekt i Android Studio. Vilket kan ses i figur 4 finns 7 typer av tomma projekt. I projektet valdes 'Navigation Drawer Activity'. Navigation Drawer Activity har en navigeringsmeny i applikationens vänstra hörn. Denna meny används för att komma till alla aktiviteter i applikationen vilket kan observeras i bilaga C.5.

4.9.1 Design

De designval som gjordes är baserade på simplicitet samt användartesterna som genomfördes. Scanmore har en enhetlig design, en faktor som generellt bidrar till en god användarupplevelse, se Appendix C. På samtliga vyer existerar ett verktygsfält placerat på samma plats i alla vyer för att kunna navigera tillbaka till föregående vy. Vilket betyder att koden (som kan studeras i figur 5) för detta verktygsfält förekommer i samtliga XML-filer.

```
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="top"
    android:background="@android:color/transparent"
    android:minHeight="?attr/actionBarSize"
    app:theme="@style/TransparentToolbar" />
```

Figur 5: XML kod för verktygsfältet

Applikationens logotyp samt huvudtema valdes till färgen grön i olika nyanser. Detta val baserades på psykologiska studier som påvisar att färgen grön utstrålar en generellt bra känsla [17]. Logotypen visas bland annat på laddningskärmen för att ge ett professionellt intryck.



Figur 6: Applikationens laddningsskärm.

5

Resultat

Utgångspunkten för detta examensarbete var att färdigställa en MVP tidigt i projekts gång, för att sedan vidareutveckla denna och skapa ett värde för den potentiella kunden. Möjligheten att läsa av ett antal utvalda varor utvecklades tidigt i processen. Hursomhelst, för att verkligen undersöka huruvida behovet finns av applikationen; krävs det en konceptapplikation med fokus på att leverera en genomtänkt användarupplevelse. Därav har mycket tid spenderats på att utveckla robusta funktioner och en design som tilltalar användarna. Detta kapitel presenterar resultaten kring ovannämnda punkter och skärmdumpar som visar applikationens samtliga vyer återfinns i Appendix C.

5.1 Resultat från användartester

Syftet med att användartesta applikationen var att fastställa huruvida ett behov av applikationen finns; samt bidra med förbättringsförslag angående användarupplevelsen av applikationen.

5.1.1 Förhandsformulär

Innan användartestet ställdes ett par frågor för att få en generell överblick över testpersonerna. De svar som antecknades kan läsas i appendix, se bilagor A.1 till A.9. 50% av användarna använde inte en handskanner med motivation att det fanns brist på erfarenhet och engagemang för att lära sig använda dem, se bilagor A.4 och A.5. Ytterligare var 62% intresserade av kunna betala sina varor via mobilen i butiken på grund av att det skulle vara smidigt.

5.1.2 Testuppgifter

Under utförandet av testuppgifterna iaktogs och antecknades användarnas upplevelse, detta genom att notera kommentarer som gjordes. De exakta svaren hittas i appendix, se bilagor A.10 till A.14. Sammanfattningsvis fann användarna det enkelt att skapa ett nytt användarkonto för att sedan logga in. 50% av användarna navigerade till sin personliga profil utan större komplikationer, resterande anmärkte att de önskade ett lättare alternativ.

Betalningsvyn påvisade ett antal designfel då många testanvändare fann det svårt att lägga till ett nytt betalmedel och slutföra köpet, se bilaga C.26 för den första versionen av betalningsvyn. Många uttryckte ett behov att en sista gång se de inskannade varorna innan köpet slutförs. Felhantering introducerades även vid inmatning av betalmedelsuppgifter, vilket visade användaren när felaktiga uppgifter matats in, se bilaga C.25.

5.1.3 Efterhandsformulär

Efter användartestet ställdes några frågor om applikationen, dessa rörde bland annat om användarupplevelse samt vilka förbättringar som kunde göras. Resultaten från detta formulär anges i bilagor A.15 till A.25. Nedan presenteras de förbättringsförslag som kunde utvinnas ur efterhandsformuläret:

1. Skapa en tydligare vy för inskannade produkter
2. En 'tutorial' för nya användare
3. Utöka shoppinglistan, ge användaren möjligheten att skanna in listor från anteckningar
4. Se enskilda köp under den ekonomiska översikten
5. Kunna se lösenord vid registrering och inloggning
6. Bekräftning vid borttagning av vara

De punkter som valdes att gå vidare till utvecklingsfasen var 1,5 och 6. Resterande ansågs ej hinna utvecklas inom projektets givna tidsram. I nästkommande kapitel angående fortsatt arbete presenteras de funktioner som hade prioriterats i ett scenario där applikationen skulle vidareutvecklas.

5.2 Resultat för precisering av frågeställningar

Inledningsvis i arbetet introduceras ett antal frågeställningar, dessa formulerades med syftet att besvaras under rapportens gång.

5.2.1 Konceptsbehov

Frågan kring huruvida ett behov av applikationen existerar på den svenska marknaden, relaterar till målet att en undersökning skall utföras. Denna fråga besvaras i 6.1.

5.2.2 Inskanningsprocess

Frågan hur tillvägagångssättet för att skanna in en vara kan utformas besvaras främst i systemkonstruktionen. Sammanfattningsvis användes bildhanteringsbiblioteket Zebra Crossing, detta läste av värdet av given streckkod som sedan jämfördes med databasen innehållande konsumentprodukter.

5.2.3 Lagring av personliga uppgifter

En annan fråga som skulle besvaras i rapporten var hur personliga uppgifter kan lagras på ett säkert sätt. I nuvarande läge sparas all data (personuppgifter och betalningsuppgifter) lokalt på en databasfil i programmet. Detta betyder att alla som har applikationen installerat på telefonen har tillgång till denna fil med databasen. Alltså kan olika användare se varandras data och logga in på deras profiler. Ur ett säkerhetsperspektiv är detta förödande och därmed inte ett bra sätt att lagra privata uppgifter på. I avsnitt 6.2 diskuteras hur man på ett bättre sätt kan lösa detta.

5.2.4 Fördelar med en mobil varuhandlingsprocess

Under förhandsformuläret påvisades ett antal fördelar att utföra hela skanningsprocessen i mobiltelefonen, se bilagor A.8 & A.9. Användarna frågades även varför/varför inte intresset fanns. Den största fördelen som påvisades var att det ansågs vara bekvämt vid köp av ett mindre antal produkter, därav nackdelen att det ansågs vara besvärligt vid storhandlingar.

6

Slutsatser & Diskussion

I detta kapitel redogörs de erfarenheter och slutsatser projektet medfört, detta innebär en diskussion kring resultatet, förbättringsområden samt en analys angående projektets arbetsmetodik.

6.1 Resultatsdiskussion

Den slutgiltiga konceptapplikationen har så väl som arbetets idé bemötts entusiastiskt av både när- och utomstående. Användartesterna visar att ett behov av applikationen finns från dem som testat den. Huruvida ett behov på den svenska marknaden existerar har projektet inte lyckats bevisa. För att kunna testa detta krävs det stöd från en större grupp människor. Definitionen av att ett behov finns behöver klargöras, eftersom de som testat applikationen påvisat ett behov kan resultatet tekniskt sätt bevisats. Men i verkligheten behöver behovet motsvara en tillräckligt stor del människor, som medför att projektet skulle vara en lönsam investering för butikskedjor. Relationen till testanvändarna kan även påverka resultatet. Personerna kan välja att verka positiva till applikationen trots att egentligen känner tvärtemot. Detta för att inte eventuellt framstå som sårande. I en optimal situation skulle testerna genomföras på personer utan koppling till utövarna.

6.2 Mål

De mål som sattes upp i början av rapporten presenteras nedan och vidare ska en förklaring genomföras hur dessa mål uppfylldes eller ej.

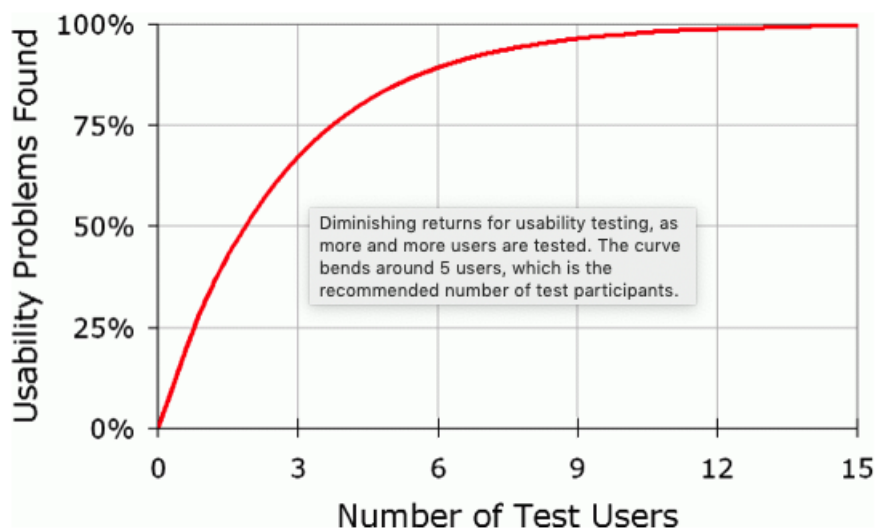
- En konceptapplikation skall utvecklas
- Konceptapplikationen skall testas mot faktiska användare
- En undersökning kring behovet av applikationen skall utföras

Applikation har testats mot faktiska användare, testgruppen bestod främst utav anställda på företaget Conmore, tillsammans med familj. Testgruppen bestod av 8 personer med olika bakgrunder och förutsättningar. En undersökning kring behovet av applikationen har utförts i samband med användartesterna, detta i form av fråga 5 i förhandsformuläret, se bilaga A.8. Samt fråga 1 i efterhandsformuläret, se bilaga A.15. 75% av testanvändarna svarade att de hade kunnat tänka sig att sköta hela handlingsprocessen via mobiltelefonen. 65.5% angav svaret 5, det vill säga att de starkt håller med om att de hade kunnat tänka sig att använda systemet igen, resterande 37.5% angav svaret 4. Innebärande att en undersökning kring behovet av applikation har utförts. Huruvida behovet motsvarar verkligheten kvarstår att undersöka. Trots att målet är uppnått är inte syftet tillfredsställt, undersökningen har utförts men behovet på den svenska marknaden är fortfarande oklart. Användartesterna genomfördes med hjälp av konceptapplikationen, figur 3 representerar de systemkopplingar som påvisas i appendix C, vilket är resultatet av utvecklingen samt kontinuerlig ad-hoc testning. Detta innebär att målet kring att utveckla en applikation har utförts.

6.3 Lagring av personlig data

Systemkonstruktionen presenterar hur lagringen av användardata hanterades, resultatkapitlet beskriver sedan varför denna typ av lagring medför problem. Att lösenorden ej krypterades kan anses vara dålig praxis, trots att projektet endast utvecklade en konceptapplikation. Hursomhelst, hade en kryptering ej bidragit med något i detta projekt. Applikationen installerades endast på utvecklarnas enheter i avseende att testas, både av utvecklarna samt mot användarna. En initial tanke angående det här problemet är att utveckla en server där uppgifterna lagras. Även detta medför hursomhelst ytterligare problem. Vi lever idag i ett samhälle där nästan all vår personlig data är digitaliserad, detta har resulterat i ett samfund av hackare med mål att komma över denna information. Trots att en egenutvecklad server kan upplevas som säker, finns det alltid människor runt om i världen med kunskap att knäcka den. Allt oavsett förbyggande teknik kan i slutändan komma att kompromissas. Därför är det i regel svårt för mindre företag, eller i detta fall programmerare utan vidare erfarenhet av IT-säkerhet, att utveckla en plattform säker nog för lagring av känslig information. För att kunna leverera applikationen till den nuvarande marknaden hade ytterligare åtgärder krävts, dessa kommer att presenteras i nästkommande kapitel angående fortsatt arbete.

6.4 Diskussion av användartester



Figur 7: Diagram som beskriver relationen mellan antalet testanvändare och antalet användbarhetsfel funna.

Syftena till användartesterna var följande: Att undersöka huruvida ett behov av applikationen existerar samt att förbättra konceptapplikationen. I metodkapitlet presenterades antalet testpersoner som utfört användartesterna. Anledningen till att det just blev 8 användartestar var dels på grund av tidsbegränsningen samt utbudet vid tidpunkten. Hursomhelst är antalet testpersoner tillräckligt för att upptäcka designfel enligt figur 7 som är hämtad från en undersökning av Jakob Nielsen. Han hävdar att det optimala antalet testpersoner har framgått till mellan 5 och 15 [18]. Enligt figur 7 innebär detta att fler testpersoner nödvändigtvis inte åstadkommer en mer användarvänlig applikation.

Däremot anses antalet vara lågt i avseende på huruvida det verkliga behovet av applikationen existerar. Även om att de 8 användarna uttryckte sig klart positiva till konceptidéen efter att ha testat applikationen, representerar dessa inte en majoritet av användarna på marknaden. För att fastställa detta med mer säkerhet, behöver applikationen lanseras, det vill säga göras tillgänglig för allmänheten; eftersom det innebär att en stor mängd återkoppling kan samlas in från användarna och därmed ge en mer tillförlitligt slutsats.

7

Fortsatt Arbete

Tidigare i projektet presenterades två avgränsande faktorer, tidsbegränsningen samt faktumet att en konceptapplikation utvecklades. I detta kapitel presenteras de åtgärder som hade krävts för att möjliggöra en leverans av applikationen, samt ytterligare funktioner som hade valts att utvecklas ifall mer tid förekom.

7.1 Koncept till leverans

För att ta steget från konceptnivå till marknadsredo krävs ett antal åtgärder. Först och främst behöver lagringen av känslig information ske på ett säkrare sätt, en redan existerande databas över konsumentprodukter behöver integreras; sådant att applikationen inte begränsas till ett mindre antal utvalda varor, sist men inte minst behöver applikationen utvecklas till att stödjas av fler plattformar än endast Android.

7.1.1 React Native

Som nämndes tidigare i rapporten användes Android Studio för att utveckla applikationen. Detta begränsar förstås till att appen endast kan användas och prövas på just ett operativ system. Men på grund av att målet inte var att skapa en applikation som skulle lanseras på marknaden, användes ett program som båda gruppmedlemmarna hade erfarenhet av för att snabbt kunna komma igång med arbetet.

En utvecklingsmiljö som skulle vara ett föredra istället är React Native, ett cross-plattform baserat ramverk. Om man istället skulle använda denna skulle man både kunna köra applikationen på iOS och Android. En annan fördel är att det går generellt mycket snabbare att utveckla i React Native, eftersom det är smidigare att utveckla samt testa.

7.1.2 Lagring av personliga uppgifter

Hade detta arbete utförts i syfte att lanseras, samt på ett företag med tillräcklig kaliber för att själva lagra informationen; hade detta naturligtvis utförts. Men i det givna fallet är det naturliga valet att överlämna ansvaret till ett externt företag, med de resurser och kompetens som krävs; säkert kan lagra informationen. I dagens marknad existerar ett flertal företag som erbjuder denna typ tjänst. Utöver personliga uppgifter som e-postadresser och lösenord, hade även betalmedel krävt ytterligare säker lagring samt integrering av betaltjänster. Även möjligheten att betala med bankkort och Swish hade behövts integreras med företagets nuvarande tjänster. ICA till exempel har redan ett system för betalningar med både kort och Swish, detta hade behövts kopplas så att det tillåter betalningar från applikationen.

7.1.3 Integration av existerande databas

Ifall applikationen hade utvecklats i syfte att användas i ett detaljhandelsföretag som till exempel ICA, behöver deras redan existerande databas över varorna de erbjuder göras åtkomliga i applikationen. Det vill säga att integrera databasen sådan att ICAs varor kan skannas in. Hur det skulle göras i nuläget är svårt att avgöra med tanke på bristen av kunskap angående deras databas, innebärande hur datan är åtkomlig från utomstående system. Hursomhelst, ICA erbjuder i dagsläget möjligheten att skanna produkter via en handskanner, detta innebär att en liknande funktion redan existerar; vilket medför att tillgången till databasen troligtvis varit möjlig för projektets applikation.

7.2 Ytterligare funktioner

Resultatkapitlet presenterade en lista över förbättringsförslag som kunde utvinnas ur efterhandsformuläret. Eftersom faktiska användare gav upphov till dessa punkter, anses de vara högt prioriterade angående ytterligare funktioner som hade utvecklats med mer tid tillgänglig. Först och främst hade den integrerade shoppinglistan vidareutvecklats. Användartesterna bidrog med många nya idéer kring denna, bland annat möjligheten att läsa av redan skapade shoppinglistor från till exempel telefonens anteckningar eller SMS. Detta för att sedan kunna bocka av produkter från shoppinglistan medan användaren utför sin handling utav varor.

Efter shoppinglistan hade vyn för ekonomisk översikt utvecklats (se bilaga C.3). Denna tjänst erbjuds till viss del redan av företag som Tink, men ej på liknande nivå som är möjlig när köpen sker direkt i applikationen. Självklart är teknologin begränsad till de köp som sker hos detaljhandelsföretaget. Hursomhelst, i nuläget tillåts användaren endast möjligheten att se tidpunkt samt beloppet för köpet. Med tanke på att detta arbete omfattar en lista över de inskannde produkterna, kan denna göras synlig i samband med tidigare köp. Detta innebär att användaren i praktiken skulle kunna se exakt vad, när och hur köpet är utfört. Exempelvis kan användare som handlar direkt efter jobbet, utan tillgång till vilka produkter som finns hemma; med hjälp av den ekonomiska översikten se att hen inhandlade tacos sås för mindre än en vecka sedan. Därav kan användaren dra slutsatsen att detta inte behöver inhandlas igen, vilket eliminerar problemet att användaren köper på sig dubletter av varor på grund av en vild chansning.

Till sist hade en tutorial utvecklats med syfte att guida förstagångsanvändare i applikationen. Denna hade visats första gången applikationen startas med möjlighet att avisas, ifall en återanvändande användare installerar om applikationen på nytt. Funktionen hade främst inriktad sig för användare som saknar vana av mobila applikationer, vilket huvudsakligen innebär användare utav äldre karaktär.

Litteraturförteckning

- [1] Conmore. (2019). Conmore Ingenjörbyrå - Conmore. [online] Available at: <https://conmore.se/en/om-oss/> [Accessed 27 May 2019].
- [2] Retail Innovation. (2015). Shoprite show barcode scanning using phones now a lot faster. [online] Available at: <http://retail-innovation.com/shoprite-show-barcode-scanning-using-phones-now-a-lot-faster> [Accessed 25 May 2019].
- [3] StatCounter Global Stats. (2019). Mobile Operating System Market Share Worldwide | StatCounter Global Stats. [online] Available at: <http://gs.statcounter.com/os-market-share/mobile/worldwide> [Accessed 8 Jun. 2019].
- [4] Budi Kurniawan. 2014. Java for Android. Vancouver, Canada: Brainly Software
- [5] Ted Hagos. 2018. Learn Android Studio 3: Efficient Android App Development. Apess.
- [6] Sandra E Eddy. 2000. XML på egen hand. Stockholm: IDG Sweden Books
- [7] W3schools.com. (2019). Java Encapsulation and Getters and Setters. [online] Available at: https://www.w3schools.com/java/java_encapsulation.asp [Accessed 21 May 2019].
- [8] Thomas W. Christoffer, George K. Thiruvathukal. 2000. High-performance Java platform computing. New Jersey: Prentice Hall PTR
- [9] Roos, M. (2018). Kom igång med databasen SQLite med DB Browser för SQLite | dbwebb. [online] Dbwebb.se. Available at: <https://dbwebb.se/kunskap/kom-igang-med-databasen-sqlite-med-db-browser-for-sqlite> [Accessed 24 May 2019].
- [10] Gs1.se. (2019). EAN-13 och EAN-8. [online] Available at: <https://www.gs1.se/vara-standarder/marka/EAN-13-och-EAN-8> [Accessed 18 May 2019].
- [11] Git-scm.com. (2019). About - Git. [online] Available at: <https://git-scm.com/about> [Accessed 14 May 2019].
- [12] Owen, S. (2018). zxing/zxing. [online] GitHub. Available at: <https://github.com/zxing/zxing> [Accessed 7 May 2019].
- [13] Mark Layton. 2018. Scrum For Dummies. City: For Dummies

- [14] Guru99.com. (2019). What is Adhoc Testing? Types with Example. [online] Available at: <https://www.guru99.com/adhoc-testing.html> [Accessed 27 May 2019].
- [15] Trello.com. (2019). Trello. [online] Available at: <https://www.trello.com> [Accessed 10 May 2019].
- [16] Android Developers. (2018). SQLiteDatabase | Android Developers. [online] Available at: <https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase> [Accessed 22 May 2019].
- [17] Elliot, A., Fairchild, M. and Franklin, A. 2015. Handbook of color psychology.
- [18] Nielsen, J. (2000). Why You Only Need to Test with 5 Users. [online] Nielsen Norman Group. Available at: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> [Accessed 28 May 2019]

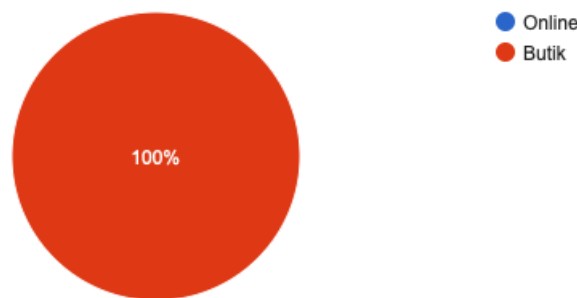
A

Appendix 1 - Användartester

A.1 Förhandsformulär fråga 1

Brukar du handla din mat online eller i butiker?

8 svar



A.2 Följdfråga fråga 1

Varför/Varför inte?

8 svar

jobbigt att hämta matkassar, själv kunna bestämma exakt när man handlar.

Småhandlar

Både online och butik, smidigt ibland att det kommer till dörren. Kul att se grejerna live, större utbud i butik.

Ibland på nätet, oftast i butik. Bättre priser i affären, miljötänk att inte få allt hemkört.

Inte erfarenhet, väntar tills beprövat, hört från andra att det funkar.

Ej vuxen

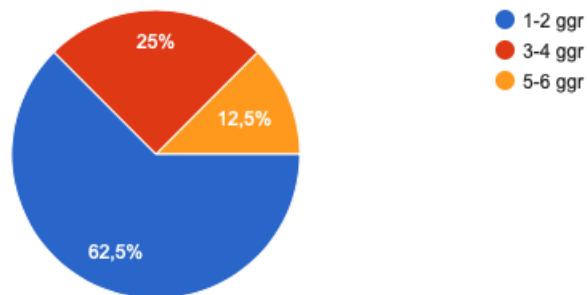
Kul att handla i butiker. Se alla varor. Handla inte online för att det är roligare i butik. Inspo.

Vill vara i fysisk miljö.

A.3 Förhandsformulär fråga 2

Hur ofta brukar du handla per vecka?

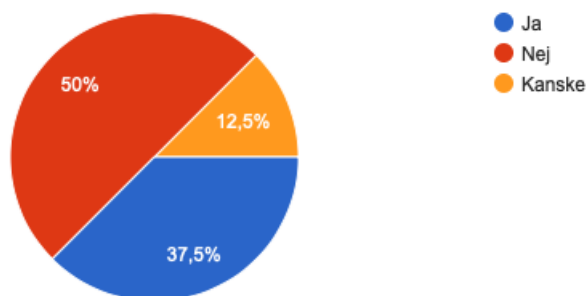
8 svar



A.4 Förhandsformulär fråga 3

Många butikskedjor som t.ex. Coop erbjuder möjligheten att du själv kan skanna dina varor m.h.a. en handskanner. Är detta något du brukar använda dig av?

8 svar



A.5 Följdfråga fråga 3

Varför/Varför inte?

8 svar

smidigt, betala fort och slippa stå i kö. Lägga ner direkt i kassen.

Använder självskanning, dom har ingen på willys.

Aldrig satt mig in i hur dom fungerar. Har ica-kort så tekniskt möjligt att använda. Aldrig försökt.

Har sagt att dom ska göra det, beror på lathet. Registrering.

Ibland, beror på ifall det är kö. Gentemot mobilen.

Går fortare att betala

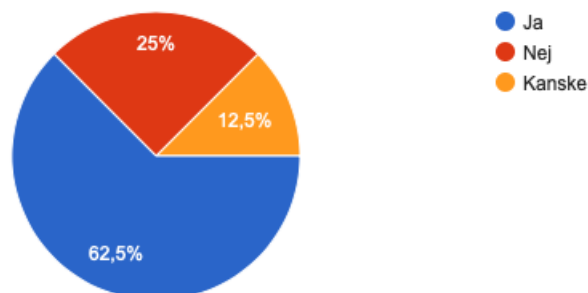
Det går snabbare, koll på hur mycket du handlar för.

Det frågar hon sig ofta. Ser det som en möjlighet.

A.6 Förhandsformulär fråga 4

Väldigt få butiker tillåter kunderna att betala enkelt via mobilen. Är detta något du hade varit intresserad av?

8 svar



A.7 Följdfråga fråga 4

Varför/Varför inte?

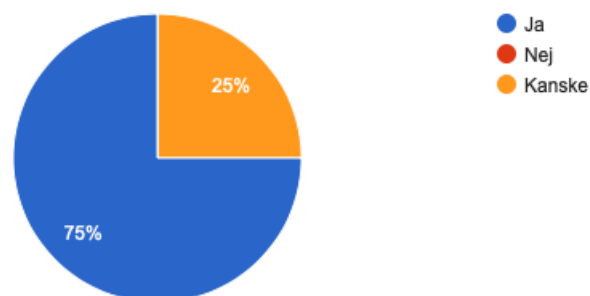
7 svar

Gillar inte att ha kort i mobilen, men har swish.
Om det är enkelt, om det är direktkopplat. Inte som swish med hitta nummer.
Smidigt att göra.
Absolut, jobbigt att det är olika appar för olika företag. Hade varit skönt att ha en app för alla matbutiker etc.
Man behöver inte ha kort med sig
Hellre swishar än kortet i appen.
Förstår inte hur man ska swisha numret.

A.8 Förhandsformulär fråga 5

Hade du varit intresserad av att sköta hela processen via din mobiltelefon? Dvs. att du själv scannar produkterna och sedan betalar i mobilen.

8 svar



A.9 Följdfråga fråga 5

Varför/Varför inte?

6 svar

Tänka sig att pröva.

Hade gärna prövat, hellre mobilen än handskanner.

Om det är ett tryggt system som funkar smidigt, standardiserat. Gemensamt gränssnitt.

Jättebekvämt

Inte ha med sig kort.

Tveksamt.

A.10 Testuppgift 1

Börja med att skapa ett användarkonto och logga in i applikationen. Navigera sedan till din personliga profil.

8 svar

Lätt att se sign up, inga problem att fylla i. Kanske beskriva att fylla i fullt namn. Hitta enkelt profilen. Tid: ??

Skapa konto lätt att hitta. "Var fan är profil" Hittade profilen lätt!

Hittade lätt registreringsknappen. Klickade direkt på översikt. Svårt att hitta profilen. Skulle behöva en tutorial. Läste texten under. Tänkte att det var alternativen på listan som fanns.

Hittade snabbt sign-up. Inga problem att signa up och logga in. Hitta direkt hur man hittade profilen.

Glömde att signa up, antagligen eftersom appen är i tidigt stadiet och inte antas funka. Sedan inga problem att skapa kontot. Klickar overview, byta namn? Svårt att hitta profilvy, hittade inte att man kunde klicka på profilden.

Kanske specificera för+efternamn, inga konstigheter med att hitta registrering.

Specificera att efternamn ska vara med. Hittar inte avatarklicken.

Otydligt var man skulle skapa konto. Behövs visa lösenord vid inloggning. Oklart hur man hittar hamburgaremeny/profil.

A.11 Testuppgift 2

Du ska nu inhandla följande: En snusdosa lundgren, en Conmore vatten och te. Hur gör du för att skanna in dessa produkter?

8 svar

Hittade enkelt skan, förstod snabbt hur man skulle skanna. Kanske lägga till ny vy för shopping lista. Lätt att skanna och ta bort.

Hittade lätt skanningsfunktionen, "helvete vad smidigt".

Hittade snabbt scan knappen. Försökte skanna upp och ner. Skannade produkterna utan problem.

Hittade direkt scan-vyn. Insåg snabbt att man behövde "trycka av". Inga problem att skanna.

Hittade lätt scanfunktionen och lätt att skanna in.

Inga problem att skanna in varor, lite för snabb när man håller in.

Lätt att skanna.

Behövs tutorial till hur man skannar.

A.12 Testuppgift 3

Testa nu att skanna in snusdosan en gång för mycket, för att sedan ta bort den från din inköpslista.

8 svar

Gjorde det utan instruktioner, lätt att förstå.

"antar att det finns ett kryss" hittade väldigt snabbt.

Förstod snabbt hur man skulle ta bort inskannade varor.

Lätt att skanna in, såg snabbt krysset för att ta bort varor.

enkelt att skanna igen och ta bort.

Inga problem att ta bort, lätt

Confirm för att ta bort varor.

Lätt att ta bort.

A.13 Testuppgift 4

Det är nu dags att betala, navigera till betalningsvyn och slutför köpet m.h.a. ett kreditkort.

8 svar

Såg inga active payment, hitta enkelt att lägga till och bank kort. Inga problem att lägga till kort. Hittade direkt att man ska klicka på kortet. BankID?

Hittade snabbt kreditkortet att klicka på, hittade snabbt add new payment method. Bra att det satte stopp efter 16 siffror. Antog att man trycker på kortet för att slutföra köpet.

Hittade snabbt betalikonen, "shoppinglistan var ju tom". Oklart vart man ska klicka för att checka ut. Hittade fortfarande snabbt var man ska klicka för att betala.

Hittade direkt betalningsvyn. Letar efter hur man betalar, insåg snabbt att man klickar på kortet.

Såg direkt payvyn, och att man ska lägga till kreditkort. Sempel process för att lägga till kort. Inga oklarheter. Prompt för validity. Såg snabbt att man skulle klicka på kortet för att beta.

lätt att hitta betalningsvyn. lite extra feedback för editText

Hittade lätt betalningsvyn. Lätt att välja kort.

Svårt att förstå att man skulle klicka på kortet.

A.14 Testuppgift 5

Du kom nu på att du behövde en till vattenflaska, skanna in produkten och slutför sedan köpet m.h.a av swish och logga sedan ut från applikationen.

7 svar

Hittade snabbt hur man lägger till payment method. Lätt att lägga till swish. Inga konstigheter.

Skannar utan problem. Tog bort föregående payment method. Kanske inte förstod ifall man kunde ha flera.

Snabb inlärningsprocess för att skanna och betala. Swishknappen buggar.

Snabb inlärningsprocess att skanna in. Inga problem att betala med swish.

Behövs prompt vid fel nummer. Förstod inte att man kan swipa. Lätt att använda vid första intryck.

Inga problem.

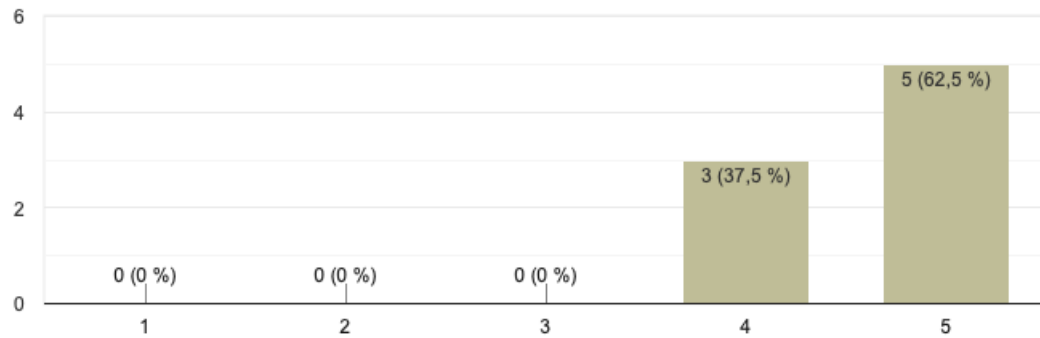
Prompt vid fel mobilnummer

A.15 Efterhandsformulär fråga 1

Jag skulle kunna tänka mig att använda det här systemet i min vardag.



8 svar

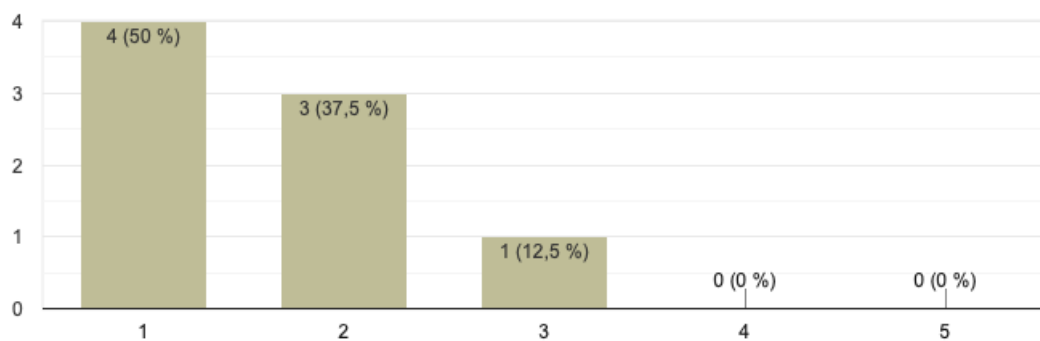


A.16 Efterhandsformulär fråga 2

Jag tyckte att applikationen var svår att använda.



8 svar

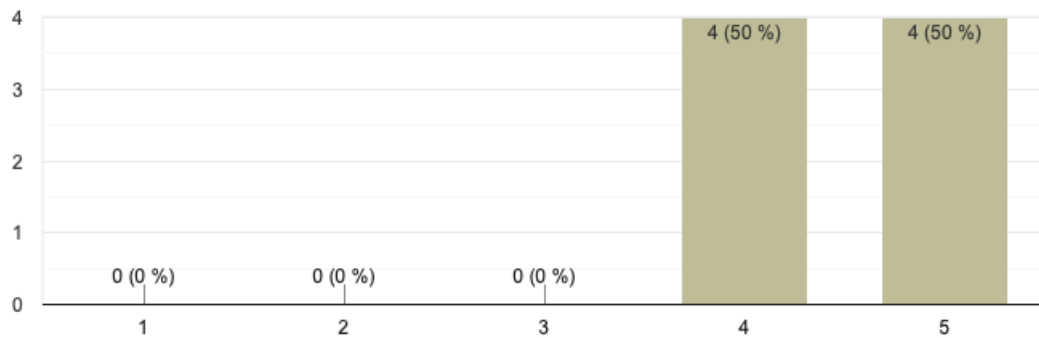


A.17 Efterhandsformulär fråga 3

Jag tror att dom flesta personerna snabbt skulle lära sig hur man använder applikationen.



8 svar

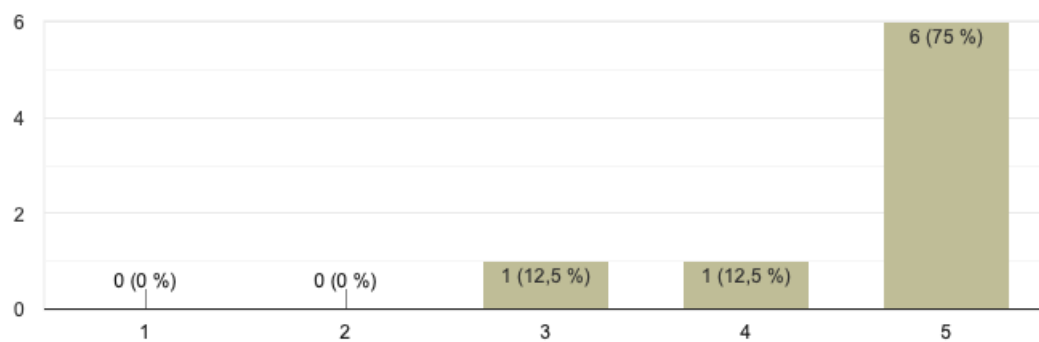


A.18 Efterhandsformulär fråga 4

Jag tyckte att dom olika funktionerna i applikationen var välintegrerade.



8 svar

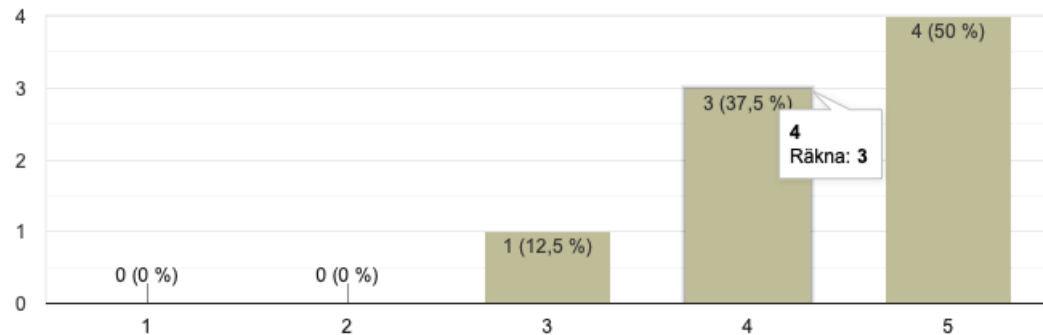


A.19 Efterhandsformulär fråga 5

Jag kände mig säker när jag använde systemet.



8 svar

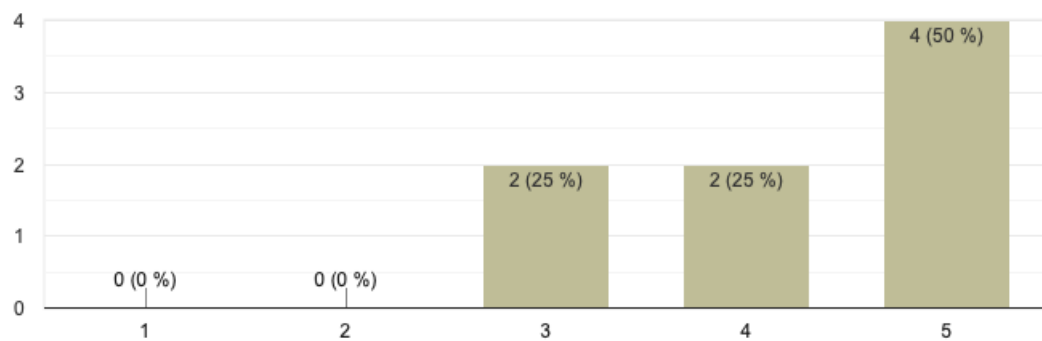


A.20 Efterhandsformulär fråga 6

Användargränssnittet var lätt att förstå.



8 svar



A.21 Efterhandsformulär fråga 7

Var det någon funktion du skulle vilja ändra?

6 svar

Knapp för allt man skannat, helskärn!

Nja, säkerhetsmässigt.

Skanfunktionen, kunna skanna utan att vända produkten. Annars inga konstigheter. Betalningsfunktion: Klicka på kortet för att ändra texten. Hint för expire date. Feedback på korttilläggnig. Röd plupp. En ny vy för inskannade produkter. Inskannade produkter i payment viewen.

Betalningsfunktion. Göra det lättare.

Lätt att navigera till olika delar. Inte för mycket knappar. Hamburgaremeny.

Nej.

A.22 Efterhandsformulär fråga 8

Är det någon funktion du skulle vilja lägga till?

7 svar

Helskärn shopping lista. En till vy för shopping lista och total, och sedan en egen för kort "CCardview"

Nej.

Importerar varulista. Bocka av från shopping lista. Ordna varor efter butik. Importera recept.

Nej.

Importerar shoppings lista från dator etc, återanvända tidigare listor.

nej, tillräckligt.

Bekräfta e-post,

A.23 Efterhandsformulär fråga 9

Var det någon funktion du skulle vilja ta bort?

4 svar

Nej.

Nej, tycker han inte.

A.24 Efterhandsformulär fråga 10

Hade du velat ändra på något i användargränssnittet?

6 svar

Nej.
Ändra payview.
Datum etc. Enkelt och klart att använda.
Estetiskt skitsnyggt. Lite lättare att navigera.
förtydliga inköpslista

A.25 Efterhandsformulär fråga 11

Tycker du det hade behövts mer hjälp för att komma igång med applikationen?

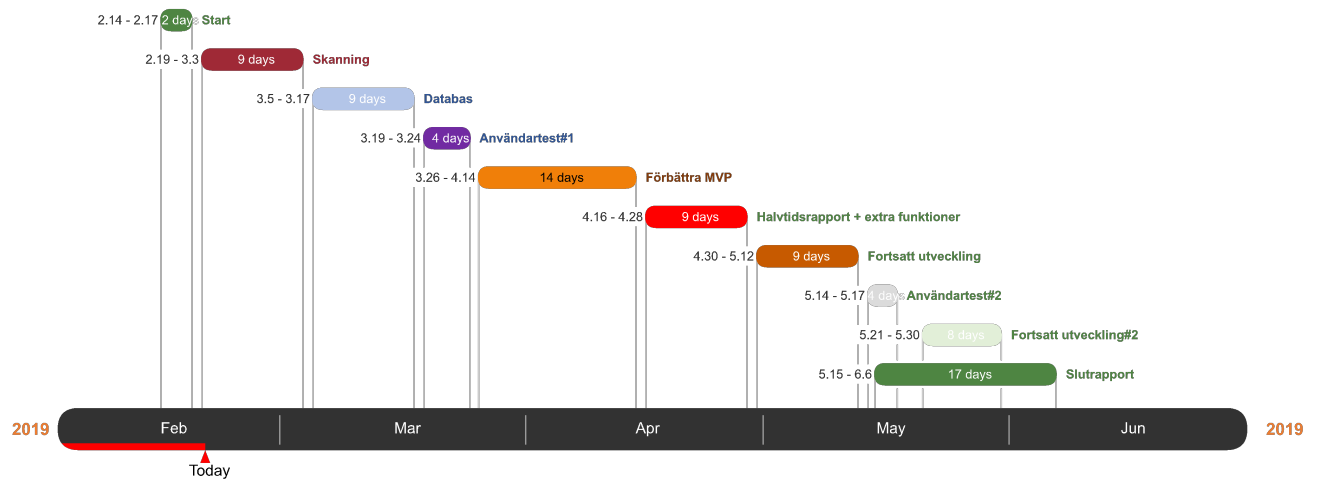
8 svar

Skulle kunna ha tutorial, pop ups första gången! Kan skippa.
Inte för hans del, men behövs alltid för andra personer.
Nej det tror han inte. Första gången tutorial.
Njaa tror inte det.
För någon som är helt ny hade det varit bra. Tutorial bra.
nej.
Bra med tutorial, välja att kolla eller inte.
Tutorial välbehövligt.

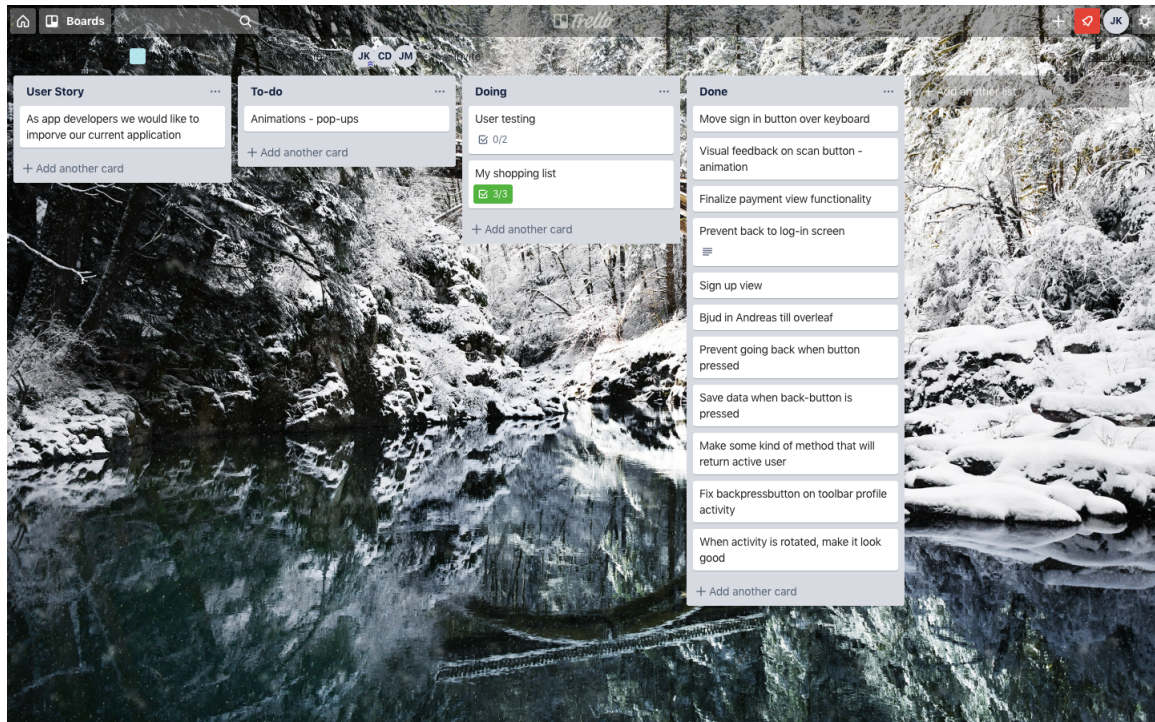
B

Appendix B - Metoder

B.1 Gantt-schema som beskriver arbetsgången



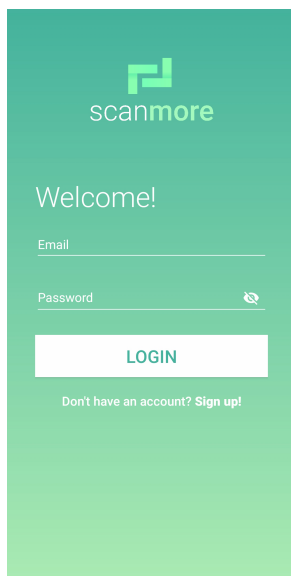
B.2 Trello-tavla som representerar en sprint



C

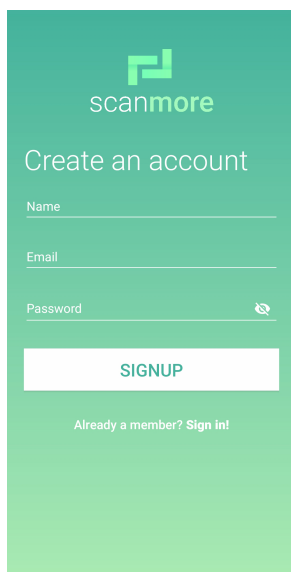
Appendix C - Applikationsvyer

C.1 Inloggningsskärm



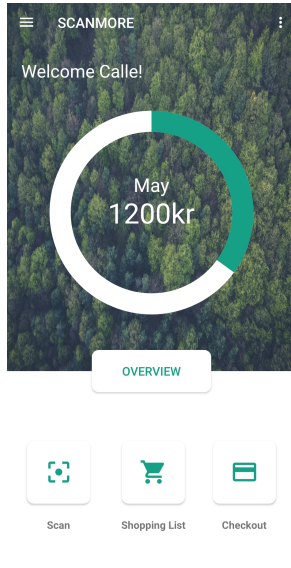
The screenshot shows the login interface for the 'scanmore' application. It features a teal header with the 'scanmore' logo. Below the header, the text 'Welcome!' is displayed. There are two input fields: 'Email' and 'Password'. The 'Password' field includes a toggle icon for visibility. A white 'LOGIN' button is positioned below the input fields. At the bottom, a link reads 'Don't have an account? Sign up!'.

C.2 Registreringsskärm



The screenshot shows the registration interface for the 'scanmore' application. It features a teal header with the 'scanmore' logo. Below the header, the text 'Create an account' is displayed. There are three input fields: 'Name', 'Email', and 'Password'. The 'Password' field includes a toggle icon for visibility. A white 'SIGNUP' button is positioned below the input fields. At the bottom, a link reads 'Already a member? Sign in!'.

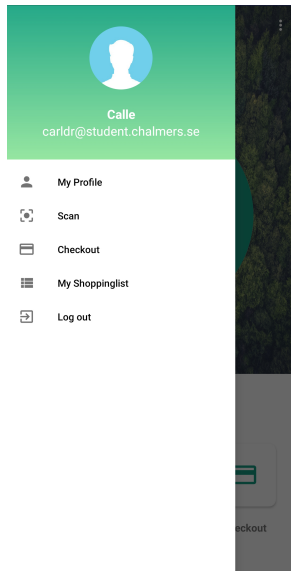
C.3 Startskärm



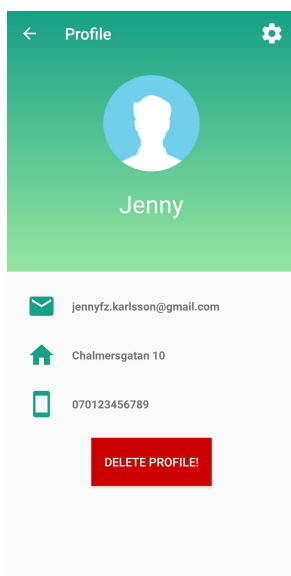
C.4 Shoppinglista



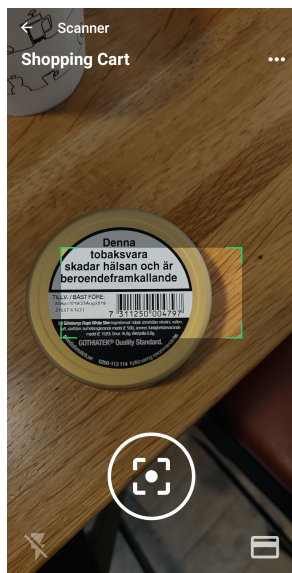
C.5 Navigeringsmeny



C.6 Profilvy



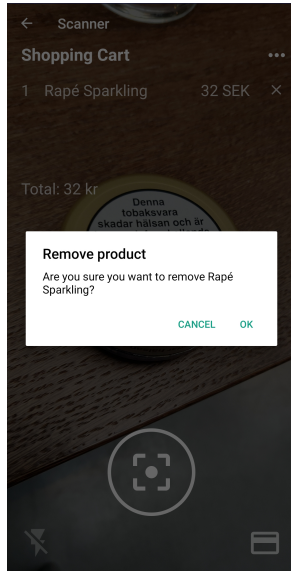
C.7 Skanningsvy



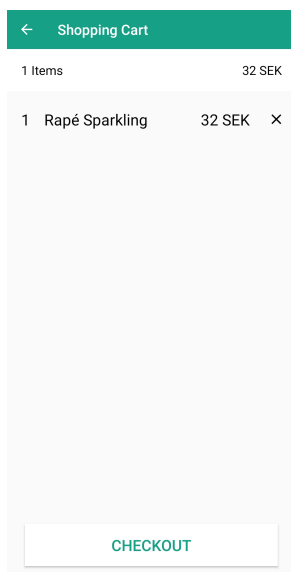
C.8 Skanningsvy med inskannad lista



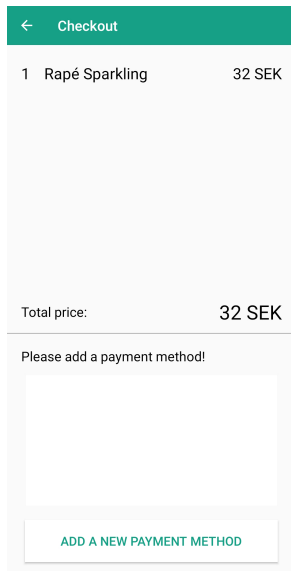
C.9 Borttagning av vara



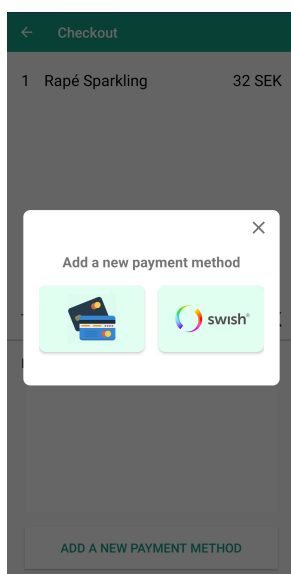
C.10 Shopping Cart



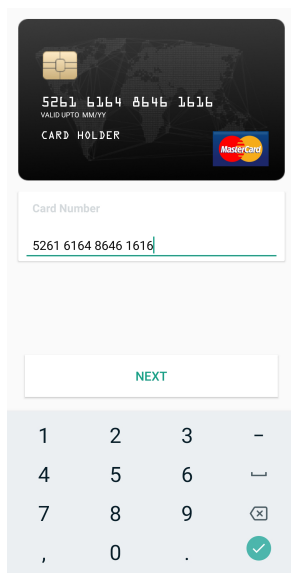
C.11 Betalningsvy



C.12 Tilläggning av nytt betalmedel



C.13 Inmatning av kortnummer



5261 6164 8646 1616
VALID UP TO MM/YY
CARD HOLDER
MasterCard

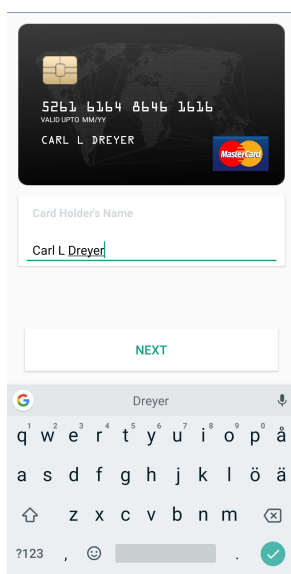
Card Number

5261 6164 8646 1616

NEXT

1 2 3 -
4 5 6 _
7 8 9 ✕
, 0 . ✓

C.14 Inmatning av kortnamn



5261 6164 8646 1616
VALID UP TO MM/YY
CARL L DREYER
MasterCard

Card Holder's Name

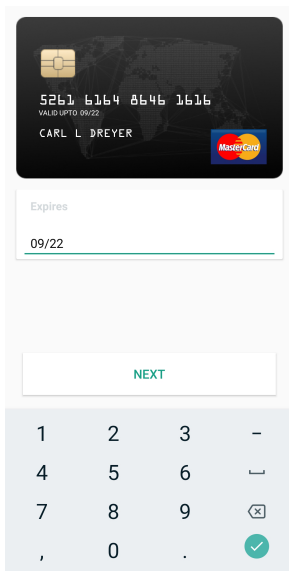
Carl L Dreyer

NEXT

Dreyer

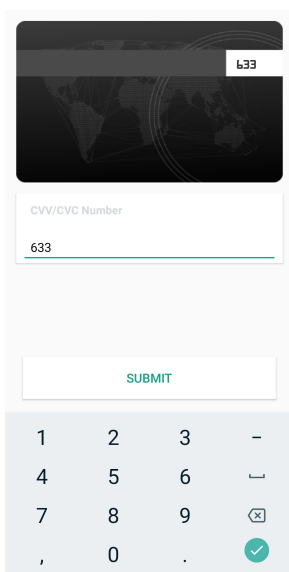
q w e r t y u i o p å
a s d f g h j k l ö ä
z x c v b n m
?123 , ☺ . ✓

C.15 Inmatning av utgångsdatum



A screenshot of a payment form. At the top is a credit card image for a MasterCard. The card number is 5261 6164 8646 1616, the name is CARL L DREYER, and the expiration date is 09/22. Below the card is a text input field labeled "Expires" with the value "09/22". Below the input field is a button labeled "NEXT". At the bottom is a numeric keypad with digits 1-9, 0, a decimal point, and a minus sign, along with a backspace icon and a green checkmark icon.

C.16 Inmatning av CVV



A screenshot of a payment form. At the top is a credit card image for a MasterCard. The card number is 5261 6164 8646 1616, the name is CARL L DREYER, and the expiration date is 09/22. Below the card is a text input field labeled "CVV/CVC Number" with the value "633". Below the input field is a button labeled "SUBMIT". At the bottom is a numeric keypad with digits 1-9, 0, a decimal point, and a minus sign, along with a backspace icon and a green checkmark icon.

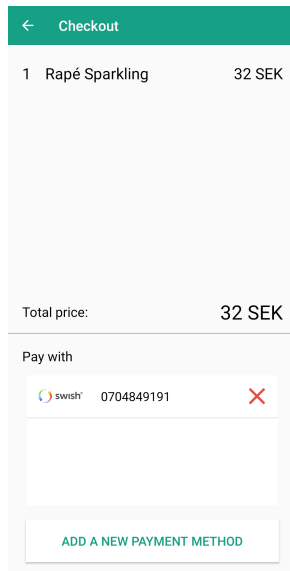
C.17 Inlagt bankkort

The screenshot shows a checkout interface. At the top, a green header bar contains a back arrow and the text "Checkout". Below this, a list item shows "1 Rapé Sparkling" with a price of "32 SEK". Further down, the "Total price:" is also listed as "32 SEK". Under the heading "Pay with", there is a failed payment attempt for a Mastercard with the number "5261 **** * 1616", indicated by a red "X" icon. Below the failed card, there is a large empty rectangular box. At the bottom, a button labeled "ADD A NEW PAYMENT METHOD" is visible.

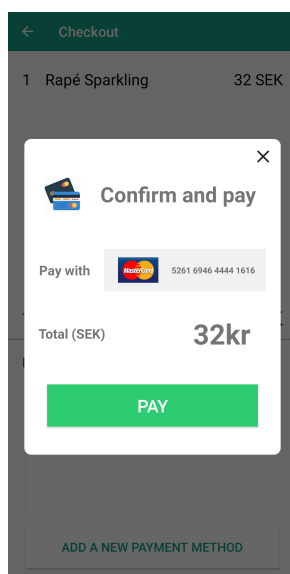
C.18 Inmatning av swishnummer

The screenshot shows a screen titled "Connect with Swish" with a green header bar containing a back arrow. The Swish logo is prominently displayed. Below the logo, the text "Enter your mobile number:" is followed by a two-part input field with a green border. Underneath the input field, a small instruction reads: "Tap Submit to connect the chosen Swish number to your account." At the bottom of the screen, there is a button labeled "SUBMIT".

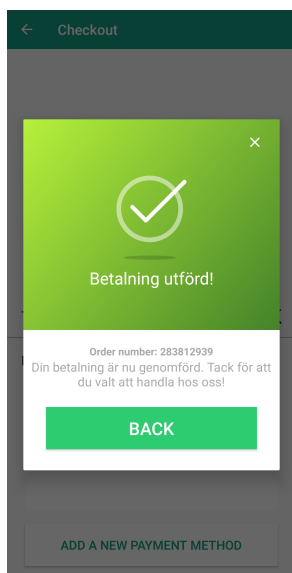
C.19 Inlagt swishnummer



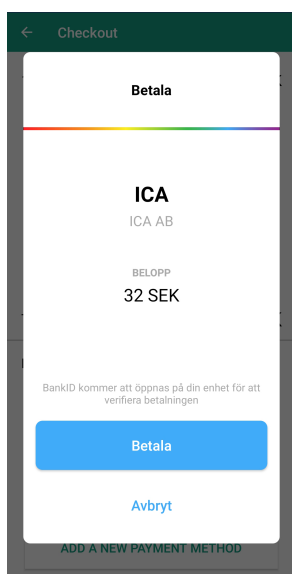
C.20 Betala med bankkort



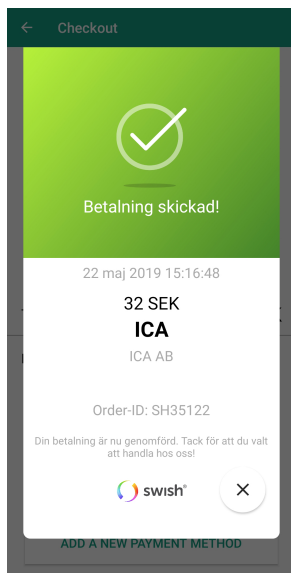
C.21 Lyckad betalning med bankkort



C.22 Betala med Swish



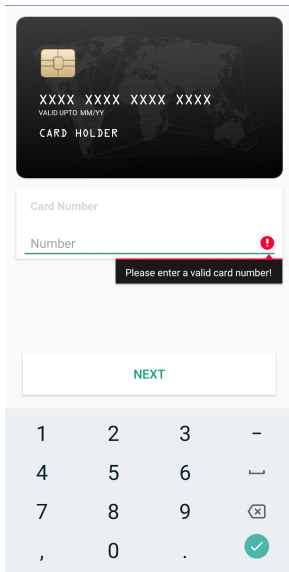
C.23 Lyckad betalning med Swish



C.24 Splash Screen

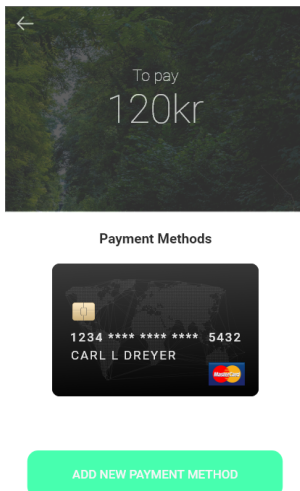


C.25 Inmatning av felaktiga uppgifter



The screenshot shows a payment form with a card image at the top. Below the card is a text input field labeled "Card Number" with the placeholder text "Number". A red error message "Please enter a valid card number!" is displayed below the input field. A green "NEXT" button is located below the error message. At the bottom of the form is a numeric keypad with digits 1-9, 0, a decimal point, a minus sign, and a backspace key. A green checkmark icon is also present in the bottom right corner of the keypad area.

C.26 Initial betalningsvy



The screenshot shows the initial payment screen. At the top, there is a back arrow and the text "To pay 120kr". Below this is a section titled "Payment Methods". Under this section, there is a card image showing the card number "1234 **** * 5432" and the name "CARL L DREYER". Below the card image is a green button labeled "ADD NEW PAYMENT METHOD".

D

Appendix D - Databas källkod

D.1 Product

```
public class Product {
    public static final String TABLE_NAME = "products";
    public static final String COLUMN_ID = "id";
    public static final String COLUMN_EAN = "ean";
    public static final String COLUMN_NAME = "name";
    public static final String COLUMN_PRICE = "price";

    private int id;
    private String ean, name;
    private int price;
    // Create table SQL query
    public static final String CREATE_TABLE =
        "CREATE TABLE " + TABLE_NAME + "("
        + COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
        + COLUMN_EAN + " TEXT UNIQUE,"
        + COLUMN_NAME + " TEXT,"
        + COLUMN_PRICE + " INTEGER"
        + ")";

    public Product() {
    }

    public Product(int id, String ean, String name, int price) {
        this.id = id;
        this.ean = ean;
        this.name = name;
        this.price = price;
    }

    public int getId() { return id; }
    public String getEan() { return ean; }
    public void setEan(String ean) { this.ean = ean; }
    public String getName() { return name; }
    public void setId(int id) { this.id = id; }
    public void setName(String name) { this.name = name; }
    public int getPrice() { return price; }
    public void setPrice(int price) { this.price = price; }
}
```

D.2 Databasehandler

```
public class DatabaseHandler extends SQLiteOpenHelper {
    // Database Version
    private static final int DATABASE_VERSION = 1;
    // Database Name
    private static final String DATABASE_NAME = "jenny";
    //Construction
    public DatabaseHandler(Context context) {
        super(context, DATABASE_NAME, factory: null, DATABASE_VERSION);
    }
    // Creating Tables
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(Product.CREATE_TABLE);
        db.execSQL(CreditCard.CREATE_TABLE);
        db.execSQL(Swish.CREATE_TABLE);
        db.execSQL(User.CREATE_TABLE);
    }
}
```

D.3 Representation av tabeller i databasen med DB...

Name	Type	Schema
▼ Tables (6)		
▼ android_metadata		CREATE TABLE android_metadata (locale TEXT)
locale	TEXT	"locale" TEXT
▼ creditcards		CREATE TABLE creditcards(id INTEGER PRIMARY KEY AUTOINCREMENT)
id	INTEGER	"id" INTEGER PRIMARY KEY AUTOINCREMENT
card_number	TEXT	"card_number" TEXT UNIQUE
card_name	TEXT	"card_name" TEXT
card_validity	TEXT	"card_validity" TEXT
card_type	INTEGER	"card_type" INTEGER
▼ products		CREATE TABLE products(id INTEGER PRIMARY KEY AUTOINCREMENT)
id	INTEGER	"id" INTEGER PRIMARY KEY AUTOINCREMENT
ean	TEXT	"ean" TEXT UNIQUE
name	TEXT	"name" TEXT
price	INTEGER	"price" INTEGER
▼ sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
name	TEXT	"name" TEXT
seq	TEXT	"seq" TEXT
▼ swish		CREATE TABLE swish(id INTEGER PRIMARY KEY AUTOINCREMENT)
id	INTEGER	"id" INTEGER PRIMARY KEY AUTOINCREMENT
phone	INT	"phone" INT
▼ users		CREATE TABLE users(id INTEGER PRIMARY KEY AUTOINCREMENT)
id	INTEGER	"id" INTEGER PRIMARY KEY AUTOINCREMENT
name	TEXT	"name" TEXT
email	TEXT	"email" TEXT NOT NULL
password	TEXT	"password" TEXT