Flexilast beställningsguide

För att förbättra kommunikationen mellan kund och företag

LUNDS UNIVERSITET
Campus Helsingborg

LTH Ingenjörshögskolan vid Campus Helsingborg
Institutionen för datavetenskap

Examensarbete:
Daniel Hurtig
Tom Hansson

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Lunds universitet
Lund 2016

Sammanfattning

Flexilast är en leverantör av transport- och entreprenadtjänster med huvudkontor i Eslöv. Om man idag vill beställa eller få hjälp med varor och tjänster från företaget så måste man ringa dit. De anställda på företaget använder mycket av sin arbetstid i telefon för att hjälpa kunder med företagets olika tjänster och produkter.

För att lösa dessa problem så kontaktade Flexilast examensarbetarna Daniel Hurtig och Tom Hansson för att utveckla en prototyp som förbättrar kommunikationen mellan kund och företag. Det hölls möten med företaget där man diskuterade vad som önskades av prototypen. En kvantitativ undersökning genomfördes i form av en enkätundersökning för att utforska intresset av prototypen inom målgruppen.

Med hjälp av den iterativa projektmodellen Kanban, som anpassats efter examensarbetet, har det utvecklats en prototyp bestående av en databas, en webbapplikation och en mobilapplikation. Databasen som utvecklats i SQL består av tre tabeller som innehåller användare av webbapplikationen, de tjänster som finns med i mobilapplikationen samt inkomna beställningar. PHP och HTML har använts för att utveckla webbapplikationen som är till för att företaget ska kunna hantera informationen i databasen. Mobilapplikationen är utvecklad i Android studio och fungerar som en guide där företagets kunder kan få information och hjälp med produkter och tjänster.

Företagets kunder kan genom mobilapplikationen få den information de behöver om företagets tjänster. De kan även genom mobilapplikationen beställa dessa tjänster baserat på egna behov. Företaget har tillgång till alla beställningar genom webbapplikationen där de även kan hantera priser på de tjänster som mobilapplikationen innehåller. På så vis är kunderna inte tvungna att ringa företaget varje gång de behöver hjälp vilket i sin tur minskar tiden som de anställda sitter i telefon.

Nyckelord: Prototyp, Mobilapplikation, Webbapplikation, Android, PHP

Abstract

Flexilast is a company located in Eslöv that provides trucking and construction services. To order or get help with goods and services from the company today, you would have to call them. The employees at the company spend alot of their time on the phone to help customers with the company's various products and services.

To solve these problems Flexilast contacted Daniel Hurtig and Tom Hansson to develop a prototype that improves communication between customer and company. There were meetings between the thesis group and the company where they discussed what was provided of the prototype. A quantitative analysis was conducted in the form of a survey to examine the interest in the target group.

Using the iterative process model Kanban, tailored to fit the needs of the group, the thesis group has developed a prototype consisting of a database, a web application and a mobile application. The database developed in SQL consists of three tables containing users of the web application, the services available in the mobile application and incoming orders. PHP and HTML has been used to develop the web application which is for the company to handle the information in the database. The mobile application is developed in Android Studio and works as a guide in which the company's customers can obtain information and assistance with products and services.

The mobile application enables the customers to get the information they need about the company’s various services. They can also order these services per their own needs. The company controls all orders through the web application where they also can handle prices on the services that the mobile application contains. In this way, customers do not have to call the company every time they need help, which in turn reduces the time employees spend on the phone.

Keywords: Prototype, mobile application, web application, Android, PHP

Innehållsförteckning

1 Inledning

Detta är ett examensarbete inom datateknik på beställning av Flexilast. Innan arbetets start sattes det upp mål och syfte för arbetet samt togs fram frågeställningar som examensarbetarna genom examensarbetet ville få svar på.

1.1 Bakgrund

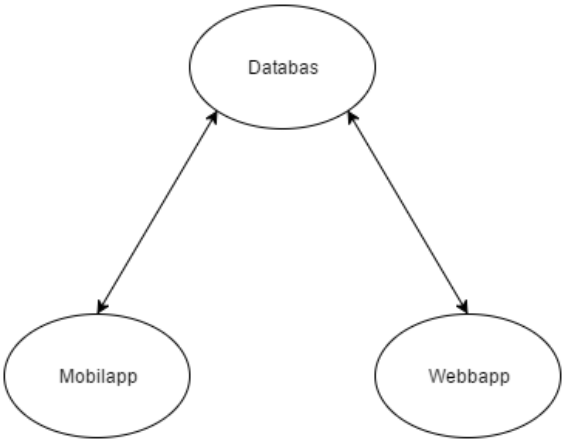
Flexilast är en leverantör av transport- och entreprenadtjänster med huvudkontor i Eslöv. En omfattande del av verksamheten är försäljningen av grus. Försäljningsprocessen sker i nuläget huvudsakligen via telefon där konsumenten ringer till företaget och frågar om eventuella tips och produkters tillgänglighet för att sen även beställa en viss mängd av en viss produkt. Denna process är omodern och onödigt tidskrävande för företaget. Detta examensarbete består av en prototypframställning som skall visa hur ett nytt system kan se ut.

1.2 Syfte

Syftet med examensarbetet är att utveckla en mobilapplikation med tillhörande backend som ska vara ett hjälpmedel för kunder och ett sätt att spara tid för de anställda i försäljningsprocessen av grus.

1.3 Målformulering

Målet med examensarbetet är att skapa en prototyp av ett system enligt figur 1.1. Prototypen ska visa hur kommunikationen mellan företag och kund kan se ut vid försäljning av företagets tjänster och produkter. Prototypen ska bestå av en mobilapplikation som används av kunden, en webbapplikation som används av företaget och en databas mellan applikationerna för datalagring. Mobilapplikationen ska tillhandahålla information om och tillgängligheten av olika typer av grus som företaget säljer. Webbapplikationen ska användas som ett medel för företaget att hantera innehållet i databasen.



Figur 1.1. *Diagram över prototypen.*

Mobilapplikationen kommer hämta information från databasen som visas för kunden. Kunden lägger en beställning som förändrar informationen i databasen. Webb-applikationen används av företaget för att uppdatera informationen som finns lagrad i databasen, till exempel priset på en viss tjänst.

1.4 Problemformulering

Följande frågeställningar besvaras under examensarbetet.

1. Hur skapar man ett tillfredsställande gränssnitt till en applikation som är till för ett ej digitaliserat företag?
2. Vad karakteriserar kundgruppen för en applikation för försäljning av transport- och entreprenadtjänster?
3. Hur ser en lämplig databas ut för prototypen?
4. Vilka tekniker ska användas för utvecklingen av prototypen?

1.5 Avgränsningar

Mobilapplikationen kommer utvecklas i Android Studio och kommer endast fungera för Android-enheter. Mobilapplikationen, webbapplikationen och databasen kommer att vara prototyper för hur ett nytt system kan se ut.

2 Teknisk bakgrund

I detta kapitel ges en översikt på verktyg och språk som använts under utvecklingen av prototypen. Mobilapplikationen har utvecklats för Android i Android studio. Programspråket Java har använts för applikationslogik och XML för uppbyggnad av användargränssnitt. Volley Java-biblioteket användes i nätverksdelen av applikationen. Webbapplikationen är uppbyggd med PHP och HTML. Data som skickas mellan dessa två applikationer är nerpackat i JSON-format.

2.1 Android

Android är ett operativsystem med öppen källkod som är designat för mobila enheter med pekskärm. Det utvecklas av Google och den första versionen kom ut september 2008. Operativsystemet är baserat på operativsystemet Linux Kernel. Användargränssnittet är baserat på direkt manipulation där användaren kan dra och peka för att manipulera objekt på skärmen. Applikationer till Android skrivs i Java tillsammans med XML (Extensible Markup Language). Där XML hanterar användargränssnittets uppbyggnad och Java står för applikationens logik. (Android, the world's most popular mobile u.å)

Androidapplikationer består av olika komponenter. Komponenter i en applikation kan anropa eller använda delar av andra program med hjälp intents (avsikter). En applikation kan till exempel starta en extern aktivitet för bildtagning. Med hjälp av intents går det återanvända andra Androidkomponenter som nödvändigtvis inte är en del av applikationen. Det finns fyra huvudsakliga komponenter. (Yaghmour, 2013)

Activities: Dikterar användargränssnittet och hanterar inmatning från användaren.

Services: Hanterar bakgrundsprocesser som är associerade med applikationen.

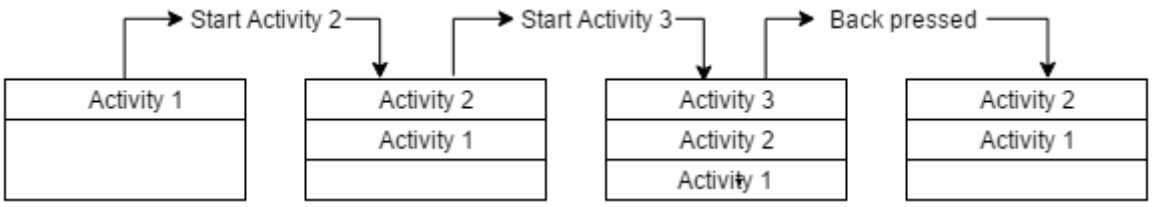
Broadcast Receivers: Hanterar kommunikation mellan operativsystem och applikation.

Content providers: Hanterar data och databasfrågor.

En aktivitet är en programkomponent som bygger upp vad som kommer att visas på mobilskärmen. Med vilket användare kan interagera för att göra något, till exempel ringa upp, ta ett foto, skicka ett e-post eller visa en karta. Varje aktivitet ges ett fönster där den bygger upp sitt användargränssnitt med hjälp av en korresponderande XML-fil. Fönstret fyller typiskt skärmen, men kan vara mindre än skärmen och flyta ovanpå andra fönster.

Applikationer kan bestå av flera aktiviteter som är knutna till varandra. Man definierar först en aktivitet som en huvudaktivitet, vilket är startpunkten för applikationen och den första aktiviteten som presenteras på skärmen. Varje aktivitet kan sedan starta upp en ny

aktivitet. När detta händer blir den förgående aktiviteten stoppad men systemet sparar den i en stack (”back-stack”). När den nya aktiviteten startar, läggs den överst i stacken och tar användarfokus. Stacken följer sist in, först ut mekanismen vilket innebär när användaren trycker på tillbakaknappen på enheten. Då förstörs den nuvarande aktiviteten och återupptar den förgående aktiviteten. Detta illustreras i figur 2.1 där Activity 3 förstörs i steget back pressed. (Yaghmour, 2013)



Figur 2.1. *Illustration på hur back-stacken används.*

Android erbjuder även ett ramverk (framework) för utvecklare. Där utvecklaren kan ta del av fördefinierad funktionalitet. Det innehåller användargränssnitts element t.ex. widgets som kan vara knappar, textrutor, dialogrutor, o.s.v. Ett UI-element definieras statiskt i XML och dynamiskt i Java. Det går även att modifiera elementen under applikationens körtid. Android framework erbjuder även olika sätt att spara data. Utvecklaren kan till exempel använda Androids SQLite funktionalitet för att lagra data i en privatdatabas. (Yaghmour, 2013)

2.2 Android Studio

Android studio är en integrerad utvecklingsmiljö (integrated development environment (IDE)) för utveckling av androidapplikationer. Den är baserad på utvecklingsmiljön InteliJ IDEA. Förutom IntelliJs utvecklarverktyg erbjuder den många funktioner som t.ex. mobilemulator, en utvecklingsmiljö för mobiler och surfplattor, testningsverktyg etc. (Android Studio, The Official IDE for Android u.å)

2.3 PHP

PHP (Hypertext Preprocessor) är ett skriptspråk med öppen källkod som är speciellt tillämpat för webbutveckling. Det skapades av Rasmus Lerdorf och släpptes 1995. Skriptet används på serversidan för att generera HTML-data dynamiskt. PHP är ansluten till en webbserver, vanligtvis används Apache eller Internet Information Server (IIS). Webbservern exekverar skriptet där resultatet skickas tillbaka till klienten. Skriptet tillåter logiska operatorer (se figur 2.2) t.ex. aritmetiska operatorer, tilldelningsoperatorer och jämförelseoperatorer. Vanligtvis används filnamnssuffix som ".php" , ".php3," eller ".phtml".(MacIntyre 2010)

En session är ett sätt att lagra information (i variabler) som skall användas över flera sidor. Till skillnad från en cookie är informationen inte lagrad på användarens dator, utan data är lagrad på en webbserver. PHP-sessioner består av två komponenter, ett klientsession-ID och serversessionsdata. Klienterna kan skicka ett sessions-ID till servern som en URL-parameter, cookie, eller till och med HTTP-huvuden. Servern använder sedan denna sessions ID för att hitta lämpliga sessionsdata som returneras till klienten. Man kan ändra sessionens beteende med en rad inbyggda sessionfunktioner. (MacIntyre, 2010)

```php
<?php
$x = 5;
$y = 4;
echo $x + $y;
?>
```

Figur 2.2. *Exempel: Variablerna x och y tilldelas 5 och 4. Variablerna adderas och resultatet skickas tillbaka från servern.*

2.4 JSON

JSON (JavaScript Object Notation) är ett textbaserat format för utbyte av data. Formatet kan härledas från Javascript och därför är syntaxen liknande. Det är specificerat av Douglas Crockford och presenterades först på webbsidan JSON.org. JSON är språk-oberoende, kod för att generera och packa upp formatet finns i många språk t.ex. Java, C/C++, JavaScript och PHP. Figur 2.3 illustrerar hur data packas ner i JSON-format med PHP-kod och figur 2.4 ger ett exempel på hur JSON-formatet ser ut. (Introducing JSON u.å.)

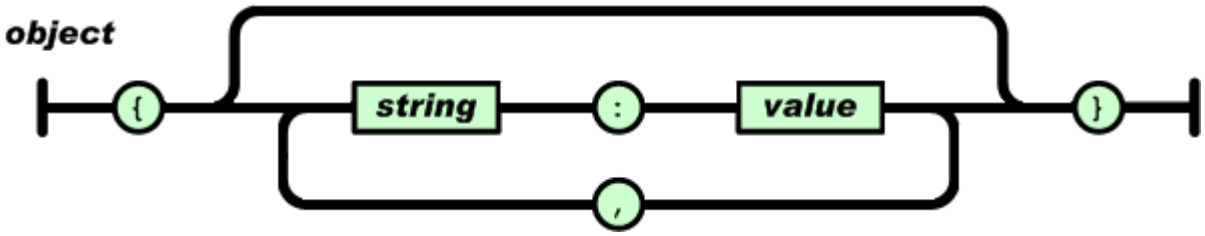
```php
$query = "SELECT  price, service FROM prices";

$result = mysqli_query($connect,$query);
$number_of_rows = mysqli_num_rows($result);

$temp_array = array();

if($number_of_rows > 0){
    while($row = mysqli_fetch_assoc($result)){
        $temp_array[] = $row;
    }
}

header('Content-Type: application/json');
echo json_encode(array("services"=>$temp_array));

mysqli_close($connect);
```

Figur 2.3. *Exempel: Kod i PHP som packar ner svaret från databas till JSON-format.*

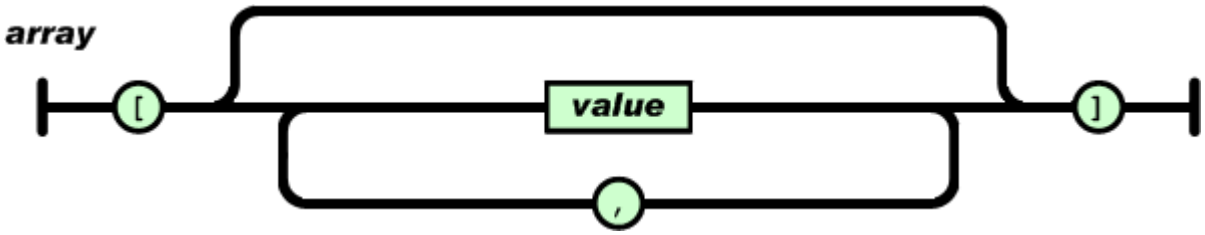
```
{"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
]}
```

Figur 2.4. *Detta exempel definierar ett Jason objekt för anställda. Med en array på 3 anställda.*

I JSON finns två olika strukturer som kan användas för att representera data. Den första är strukturerad som namn och värde det vill säga par enligt figur 2.5. Detta kan sedan i ett programspråk realiseras som ett objekt, hashtabell, ”record”, ”dictionary” eller ”associative array”. Den andra strukturen byggs som en ordnad lista. Där det beroende på vilket programspråk kallas lista, ”array” eller vektor enligt figur 2.6. (Introducing JSON u.å.)



Figur 2.5. *Ett objekt med namn/värde par.*



Figur 2.6. *En array med ordnade värde.*

2.5 Volley

Volley är ett Google HTTP-bibliotek för Java som gör arbetet med nätverkskommunikation i Android enklare och snabbare. Det släpptes av Google 2013. Volley används för att hantera nätverksförfrågningar och har en rad olika funktioner så som JSON APIs, ladda bilder och sträng-förfrågningar. (Transmitting Network Data Using Volley u.å.)

I exemplet som figur 2.7 visar skapas först en request-kö. Kön är tom när den initieras men sedan läggs det in en strängförfrågning till showUrl som är ett PHP-skript på en

lokal webbserver. I examensarbetet kommer det till användning genom att PHP-skriptet gör en förfrågning till databasen efter priset på de olika tjänsterna som Flexilast erbjuder. Sedan skickas svaret förpackat i JSON-format tillbaka och i onResponse packas data upp och läggs i listor. Till sist läggs förfrågningen in i request-kön där den utförs i bakgrunden av en arbetstråd.

```
public void getPrice()
{

    requestQueue = Volley.newRequestQueue(c.getApplicationContext());

    StringRequest stringRequest = new StringRequest(Request.Method.POST, showUrl,
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    JSONObject jsonObject = null;
                    try {

                        jsonObject = new JSONObject(response);
                        jsonArrayServices = jsonObject.getJSONArray(JSON_ARRAY);

                        for(int i=0;i<jsonArrayServices.length();i++){
                            JSONObject jo = jsonArrayServices.getJSONObject(i);
                            mPriceList.add(i,jo.getString("price"));
                            mServiceList.add(i, jo.getString("service"));
                        }

                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                    Log.d("JSON",response.toString());
                }
            },
            new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {

                }
            }) {
    };
    requestQueue.add(stringRequest);
}
```

Figur 2.7. *Exempel på användande av volley.*

2.6 Github och Git

Github är en webbserver-tjänst för mjukvaruutveckling som låter användaren publicera sina Git-repositories (lager). Detta innebär att man kan tillåta andra personer att lägga till kod till sitt projekt. På så vis kan flera personer arbeta med samma projekt samtidigt utan att nödvändigtvis behöva arbeta på samma dator. Det finns en betalnings- och gratisversion. I betalningsversionen kan man publicera sin källkod privat utan att andra kan ladda hem eller se koden. I gratisversionen kan vem som helst se och få tillgång till källkod som är publicerad. (Chacon, 2009)

Github använder versionshanteringssystemet Git som kom år 2005. Git är ett versionshanteringsprogram med öppen källkod. Det är skapat för mjukvaruprojekt där utvecklingen sker av flera personer. Med Git kan tidigare versioner av källkodsfiler återskapas och ändringar som har gjorts i tidigare versioner kan även spåras. (Chacon, 2009).

Den största skillnaden mellan Git och andra VCS (Version control systems) är sättet Git hanterar data. Varje gång det utförs en commit (registrerar förändringar) eller utvecklaren sparar tillståndet av ett projekt, då sparar Git det tillståndet på filerna och skapar en referens till tillståndet. Om till exempel filerna i projektet inte har modifierats sparar inte Git filerna igen utan länkar filerna till förgående identiska filer som redan är sparade.

Git arbetar lokalt och använder nästan bara filer som är lokala. Eftersom filerna sparas lokalt behövs ingen internetuppkoppling för att gå igenom versionshistoriken eller lägga till eller committa i ett projekt. Vid tillgång till Internet kan ändringarna laddas upp. (Chacon, 2009)

I ett projekt kopplat till Git har man en huvudversion där merparten av utvecklingen sker. Om en utvecklare skulle vilja implementera och testa en ny funktion, kan huvudversionen förgrenas till en temporär version. I den temporära versionen kan utvecklaren implementera och testa funktionen utan att ställa till det i huvudversionen. När funktionen är klar och testad kan den temporära versionen sedan sammanfogas och synkroniseras med huvudversionen. Dessa funktioner i Git kallas branching och mergin (Chacon, 2009).

2.7 XML

XML (Extensible Markup Language) är ett sidbeskrivningsspråk skapad av World Wide Web Consortium (W3C) och är baserad på SGML (Standard Generalized Markup Language). Sidbeskrivningsspråk är ett format för dokument som innehåller textkoder t.ex. taggar. Dessa syns inte i dokumentets slutliga form utan de ger instruktioner om hur t.ex. hur en mobilapplikation presenterar layout, text och bilder. Ett annat exempel på sidbeskrivningsspråk är HTML där HTML-koden beskriver hur text och webbsidan formateras. (IBM u.å.)

Android framework innehåller en XML-vokabulär som innehåller Androids förutbestämda taggar och element. Det korresponderar med layout- och widget-klasser i Java och följer oftast struktur, namn på klasser och metoder som finns i Android framework. Som exempel på detta finns det ett XML-attribut EditText som korresponderande med Java metoden EditText.setText(). När en applikation utvecklas skrivs användargränssnittet statiskt i XML. Det går även att ha flera XML-filer till samma aktivitet, t.ex. utvecklaren vet inte användares skärmstorlek och gör då flera XML-filer anpassade för olika enheter. Under körning av applikationen väljer Android den XML-filen som är anpassad för just den enheten. I den logiska delen av applikationen det vill säga i Java-koden, går det även att modifiera XML-element med inbyggda funktioner i Java. Om en utvecklare till exempel vill att en grå knapp ska ändra färg när den är nedtryckt så finns det definierat i XML-filen i figur 2.8. (Layouts u.å.)

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click here" />
</LinearLayout>
```

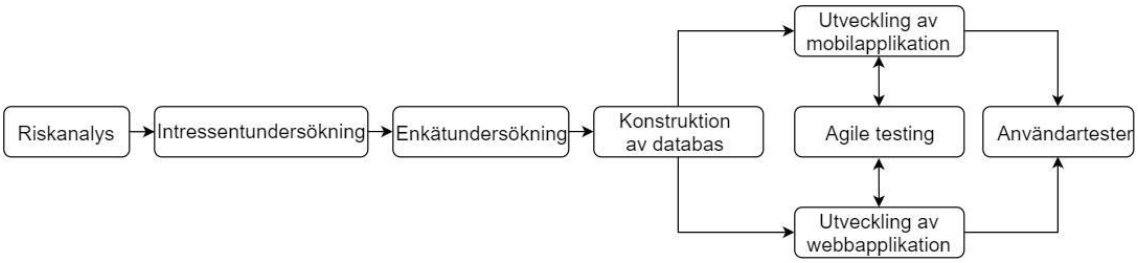
Figur 2.8. *Exempel på uppbyggnad av layout i XML.*

2.8 Trello

Trello är en webbapplikation för projektledning som utvecklats av Fog Creek Software (Pryor, 2014). Projekten är representerade som boards i vilka det finns listor som innehåller lappar med arbetsuppgifter. Lapparna kan tilldelas en användare. Det är tänkt att en lapp ska flyttas mellan flera listor innan den anses vara klar eftersom det ska gå att se i vilken fas uppgiften befinner sig i. Detta görs genom att dra en lapp från en lista till nästa. T.ex. skapas en ny lapp i en lista som kallas To do med arbetsuppgiften att implementera en ny funktion och tilldelas en person. När personen börjar på uppgiften flyttas den till listan Doing och när funktionen är klar flyttas den till Done.

3 Metod

För att genomföra examensarbetet har examensarbetarna tagit del av metoder som man lärt sig delvis inom utbildningen men också genom att studera och reflektera över innehåll i böcker, artiklar och annan litteratur. Detta kapitel beskriver vilka metoder och verktyg som använts i examensarbetet. Examensarbetets arbetsprocess illustreras i figur 3.1.



Figur 3.1. *Illustration av arbetsprocessen.*

3.1 Riskanalys

Med projekt medföljer risker och det är därför viktigt att tidigt identifiera dessa för att minimera konsekvenserna. Examensarbetarna valde att använda miniriskmetoden i detta examensarbetet (Eriksson & Lilliesköld, 2005). Miniriskmetoden innebär att först och främst komma fram till så många eventuella risker som möjligt. För varje risk uppskattas sannolikheten och hur allvarliga konsekvenserna blir om riskerna inträffar. När man sedan dokumenterat alla risker med respektive sannolikhet och konsekvens (ofta i form av en tabell eller matris) utvärderar man varje risk och, baserat på dess riskvärde, skapar åtgärder för att minimera eller eliminera riskerna och deras konsekvenser (Eriksson & Lilliesköld, 2005).

Valet av miniriskmetoden gjordes eftersom att den är en enkel metod som kan användas även om man har en begränsad erfarenhet av riskanalys. Analysen gjordes tidigt i examensarbetet för att ha riskerna i åtanke när examensarbetet planerades och utfördes. Resultatet av riskanalysen finnes i avsnitt 4.1.

3.2 Kartläggning av intressenter

Innan utvecklingsfasen genomfördes en intressentundersökning följt av en enkätundersökning för att kartlägga intressenterna för prototypen.

3.2.1 Intressenter

En målgrupp är en definierad grupp av individer som är intressanta som mottagare av en produkt (Target audience – Businessdictionary, u.å.). En intressent är en individ som är intresserad eller på något sätt påverkas av en produkt (Stakeholder - Businessdictionary, u.å.) En intressentundersökning görs för att kartlägga vilka som är intresserade av produkten som är tänkt att utvecklas. Det är viktigt att ha samtliga intressenter i åtanke när man utvecklar en produkt.

Examensarbetarna inledde examensarbetet med en intressentundersökning i samband med första mötet med företaget. Eftersom examensarbetarna redan börjat tänka på en målgruppsanalys av Flexilasts kunder så valdes det att tidigt fastställa vem man utvecklar de olika delarna av prototypen till.

Undersökningen var inte så omfattande utan bestod endast utav följande två frågor:
1. Vem utvecklas prototypen till?
2. Vem rådgör examensarbetarna med vid eventuella funktionsval?

Frågorna diskuterades på mötet och efteråt kunde examensarbetarna lista upp följande intressenter:

1. Flexilast
Flexilast är företaget som prototypen utvecklas till och de är därmed en nyckelintressent. De är användare av webbapplikationen och är även intresserade av att mobilapplikationen utvecklas på ett relevant och kundanpassat sätt.

2. Huvudsakliga användare av mobilapplikationen (Målgrupp)
Den slutliga användaren av mobilapplikationen – företagets nuvarande och framtida kunder. Nyckelintressent för mobilapplikationen. Beskrevs av företaget som ”Personer i åldern 20-35 som ska köpa bostad och är intresserade av att bygga lite själv”. Åldersintervallet användes när examensarbetarna senare valde deltagare i enkätundersökningen.

3. Sekundära användare av mobilapplikationen
Trots att utvecklingen av mobilapplikationen riktar sig främst mot målgruppen ovan så hindrar inte det att mobilapplikationen kan och förhoppningsvis kommer användas också av kunder som inte tillhör de primära användarna av mobilapplikationen.

3.2.2 Enkätundersökning

För att få en så generell undersökning som möjligt med konkreta och mätbara resultat så valde examensarbetarna att göra en kvantitativ målgruppsundersökning. I kvantitativa undersökningar samlar man in data från ett stickprov av intressenterna som man sedan använder för att dra slutsatser om hela intressentgruppen. (Park & Park, 2016).

Som kvantitativ metod valde examensarbetarna att göra en enkätundersökning med tanke på de begränsade resurserna (Tetteroo & Markopoulos, 2015).

Frågorna på enkäterna utformades utefter vad som diskuterades på första mötet med företaget i samband med att intressentundersökningen genomfördes. Undersökningarna gjordes i början av projektet då mobilapplikationen var tänkt att fungera som en webbshop som endast behandlade grusförsäljning. Men examensarbetarna försökte ändå ställa generella frågor istället för att fokusera på endast grushandel. På så vis skulle inte resultaten från undersökningarna bli irrelevanta vid eventuella förändringar av arbetet. Frågorna fokuserar istället på intresset kring att göra saker själv eftersom att det ingick i företagets beskrivning av den slutliga användaren.

Enkäterna bestod av följande frågor:

1. Vilken typ av mobiltelefon har du?
2. Brukar du handla via mobiltelefon/internet?
3. Har du ett intresse av att göra saker själv? (Lägga plattor, snickra etc.)
4. Hur letar du helst information/hjälp angående en produkt?
5. Vore instruktioner för användandet av en produkt du vill köpa intressant för dig?
6. Hade ditt intresse för att göra saker själv ökat ifall information/hjälp hade varit tillgängligt på ett smidigt sätt?
7. Hur gammal är du?

Urvalet av personer som deltog i undersökningen hämtades från den angivna målgruppen ”Personer i åldern 20-35”. Enkätundersökningen inleddes med personer ur examensarbetarnas bekantskapskrets som matchade målgruppen. Dessa personer hjälpte sedan till med att hitta fler deltagare (genom arbetsplats eller skola) som passade in på målgruppsbeskrivningen. Examensarbetarna delade även ut enkäter till personer i rätt åldersintervall på gatorna i Eslöv. Sammanlagt samlades det in resultat från 29 deltagare i undersökningen.

Prototypen beskrevs först kort för varje deltagare varpå de fick svara på enkäten antingen på ett papper (16 stycken), via mail (11 stycken) eller vid två tillfällen över telefon. Examensarbetarna ville ha svaren i samband med att deltagarna fick tillgång till frågorna för att minimera undersökningstiden. Eftersom enkäten endast tog en till två minuter att svara på så lyckades man med detta förutom några enstaka fall med deltagare via mail då svaren ibland inte kom tillbaka förrän dagen efter. Mer om samt resultatet av enkätundersökningen finnes i avsnitt 4.1.3.

3.3 Kravspecifikation

Under hela examensarbetet arbetades det med kraven vilket resulterade i en kravspecifikation (se Bilaga A). I kravspecifikationen dokumenterades de nya funktionerna som skulle implementeras samt att kravspecifikationen användes i arbetet med verifiering av prototypen. Innan examensarbetarna utvecklade en ny funktion skrevs det upp ett krav utefter företagets önskemål och behov. Kravarbetet blev därmed enkelt i och med att det inte skrevs upp något krav förrän examensarbetarna diskuterat eventuella lösningar (implementeringar) med företaget.

Kravspecifikationen användes även i den agila testningen som pågick parallellt med utvecklingsfasen. Varje gång examensarbetarna implementerat en ny funktion så testades den utefter vad som stod i kravspecifikationen.

3.4 Test

Under examensarbetets utvecklingsfas har det även varit en kontinuerligt pågående testning för att upprätthålla den kvalité och uppfylla de krav som examensarbetarna och företaget satt för systemet.

3.4.1 Agile Testing

Examensarbetarna har under hela utvecklingsfasen säkerställt att applikationerna som utvecklats uppnått de krav som är specificerade i Kravspecifikationen (se bilaga A). Detta har examensarbetarna gjort genom att anpassa en ”Agile testing”-metod (Watkins & Millis, 2010) som bland annat innebär att man har kontinuerliga leveranser av fungerande funktioner till kunden samt att man har ett testteam som testar nya funktioner så fort de utvecklats. I och med att examensarbetarna haft god kontakt med företaget genom hela examensarbetet så har man haft möjlighet att visa prototypen för företaget så fort man implementerat nya funktioner. Då har företaget även fått testa de nya funktionerna och gett omedelbar muntlig feedback som examensarbetarna använde för att göra eventuella ändringar.

Examensarbetarna har även agerat som testteam åt sig själva och regelbundet testat funktioner som man implementerat. I dessa tester så har man verifierat och validerat att det man utvecklat stämmer överens med det som står skrivit i kravspecifikationen samt det man kommit överens om med företaget.

3.4.2 Användartester

Mot utvecklingsfasens slutskede gjordes även ett användartest. Användartest kan även kallas testning av användbarheten och det är till för att undersöka hur den slutliga användaren uppfattar och använder den utvecklade produkten (Barnum, 2011). Man vill se

vad en utomstående person gör med produkten, vad som fungerar och inte fungerar för personen och sedan göra eventuella ändringar för att förbättra den slutliga användarens upplevelse av produkten.

Sammanlagt deltog tre personer från den angivna målgruppen som fick testa mobilapplikationens funktioner. Anledningen till att det endast deltog tre personer var på grund av tidsbrist i slutet av examensarbetet och det var vad examensarbetarna fick tag i med kort varsel. Eftersom fokuset låg på hur enkelt testpersonerna navigerade sig igenom applikationen så ansågs det även räcka med tre deltagare av olika ålder inom åldersintervallet. Enligt en forskning gjord av Jakob Nielsen och Thomas Landauer så är det mest kostnadseffektiva antalet testpersoner mellan tre till fem personer (Barnum, 2011). Forskningen säger att redan vid den tredje testpersonen så observerar testledaren saker för andra eller tredje gången och att den tredje testpersonen endast genererar lite ny information. Varje ny testperson genererar mindre och mindre information och efter den femte testpersonen så slösar man tid på att observera samma saker om och om igen utan att lära sig mycket nytt (Barnum, 2011).

I vanliga fall utförs dessa tester i ett rum med kameror och mikrofoner med en testledare i rummet och ett antal observatörer som antecknar utanför rummet (Barnum, 2011). Men eftersom examensarbetet är ett mindre projekt så modifierade examensarbetarna metoden. Testerna genomfördes på så vis att en av gruppmedlemmarna läste upp direktiv (till exempel ”Du behöver grus till plattläggning”) för testpersonen att följa medan den andra medlemmen studerade testpersonens navigation genom applikationen. Det sistnämnda är till för att undersöka hur enkel applikationen är och hur enkel den är att navigera igenom. Efter testerna fick testpersonerna lämna muntlig feedback på hela applikationen och det blev en öppen diskussion mellan testpersonerna och examensarbetarna. Om man ansåg att något speciellt väsentligt dök upp under dessa diskussioner så antecknades det för att diskuteras vidare mellan examensarbetarna.

Dessa användartester ledde inte till några större förändringar av applikationen. Större delen av den muntliga feedbacken handlade om prototypens utseende som examensarbetarna inte lagt någon större vikt vid eftersom prioriteten hela tiden var att skapa en fungerande applikation att visa för företaget. Men efter användartesterna så ändrade man färgerna i applikationen till mer neutrala eftersom det ljusröda temat, som man fortfarande kan se i menyknapparna i den färdiga versionen, inte verkade vara särskilt uppskattat av testpersonerna.

Webbapplikationen undergick aldrig något större användartest eftersom examensarbetarna var nöjd med den testning som beskrevs i avsnitt 3.4.1 som dessutom pågick från utvecklingens start till att man ansåg att prototypen var färdig. Eftersom funktionerna i webbapplikationen testades av både examensarbetarna och företaget (som är den slutliga användaren av webbapplikationen) så fort de blivit implementerade så var ett användartest överflödigt.

3.5 Utveckling av prototypen

Detta avsnitt beskriver projektmodellen som använts under utvecklingsfasen av examensarbetet.

3.5.1 Projektmodell

Examensarbetet har genomförts enligt en anpassad version av projektmodellen Kanban (Kniberg & Skarin, 2010). Kanban är en lättrörlig, iterativ projektmodell med få restriktioner. Modellen innebär att examensarbetarna delar upp arbetet i mindre uppgifter som skrivs ner på lappar och sätts upp på en vägg. Uppgifterna placeras sedan under namngivna kolumner för att visa var i processen uppgiften befinner sig (Kniberg & Skarin, 2010). Uppgifterna listas även i en prioriterad kö så att det viktigaste i arbetet genomförs först. Tiden mellan startad och avklarad uppgift kallas ledtid. Målet är att reglera storleken på varje uppgift för att arbetsprocessen ska vara så effektiv som möjligt (Kniberg & Skarin, 2010).

I Kanban är det viktigt att gruppen har sitt fokus på det pågående arbetet (WIP – Work in Progress). I Kanban försöker man uppnå detta genom att begränsa det pågående arbetet för varje arbetsprocess (Kniberg & Skarin, 2010). I en Kanban board uppdelad i till exempel tre kolumner: ”To do”, ”Doing” och ”Done” så bestäms antalet, som ej får överstigas, uppgifter som får ligga i de respektive kolumnerna.

Hela arbetsprocessen presenteras på en Kanban-board som samtliga medlemmar i en projektgrupp kan se.

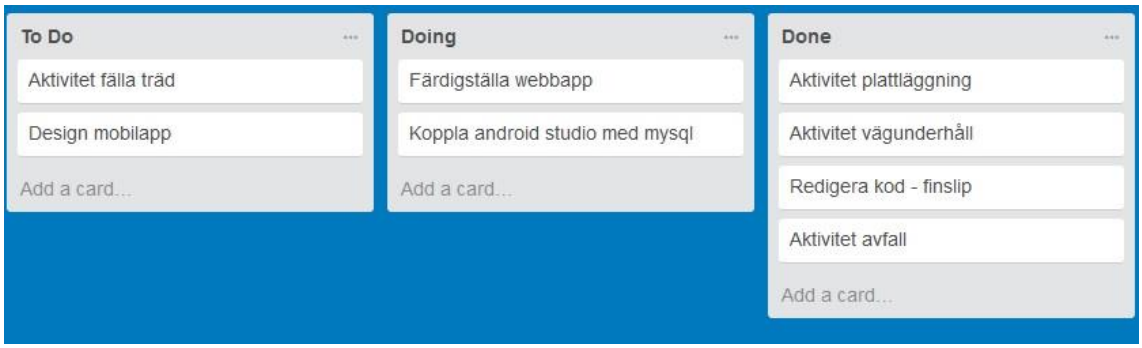
3.5.1.1 Examensarbetets anpassade Kanban

Projektmodellen Kanban innebär kontinuerliga och stegvisa förbättringar av en produkt. Examensarbetet har till stor del bestått av mjukvaruutveckling med frekventa leveranser till företaget. Därför ansåg examensarbetarna att en så pass iterativ projektmodell som Kanban passade bra för ändamålet eftersom att den kan anpassas efter plötsliga förändringar. Men eftersom detta arbete är litet jämfört med de större projekt som Kanban ofta används till så har examensarbetarna anpassat modellen till examensarbetet.

Det som uppskattats mest och gjort mest nytta är uppsättningen av Kanban-board på Trello. Hela utvecklingen har planerats och organiserats med hjälp av boarden enligt figur 3.1 och examensarbetarna har på så vis haft en överblick över vad som händer i arbetet. Arbetet har enligt Kanban delats upp i mindre uppgifter. En uppgift kunde till exempel vara en ny funktion i mobilapplikationen. När uppgiften sedan var avklarad så testades den av examensarbetarna för att kontrollera att den uppfyllde kraven specificerade i Kravspecifikationen (se bilaga A och avsnitt 3.3).

Det pågående arbetet sattes till en uppgift per person, det vill säga att varje examensarbetare endast hade en uppgift under ”Doing” åt gången, och detta har examensarbetarna reglerat med hjälp av boarden. Om man till exempel var ledig från arbetet ett par dagar och skulle börja arbeta igen så blev det naturligt att först kolla boarden för att undvika att påbörja en ny uppgift innan den förra var färdig.

Det bestämdes ingen fast ledtid i arbetet eftersom man tidigt märkte att tiden för uppgifter i olika arbetsmiljöer varierade väldigt mycket. Examensarbetarna försökte ändå anpassa storleken på uppgifterna och använda de bäst lämpade arbetsmetoderna för att effektivisera arbetet så mycket som möjligt.



Figur 3.1. *Kanban-board.*

3.6 Verktyg & arbetsmiljöer

Detta kapitel går igenom verktygen och arbetsmiljöerna som examensarbetarna använt och beskriver metoderna och lösningarna som anpassats för att nå ett önskvärt resultat.

3.6.1 Github

Examensarbetarna har arbetat tillsammans under större delen av examensarbetet, men har utvecklat prototypen på två olika datorer. Därför behövdes ett verktyg som möjliggör samtidigt arbete i samma källkodsfiler. Verktyget skulle även kunna fungera som backup på källkoden. Lösningen på detta var Github.

Github går att integrera med Android studio, där man kan uppdatera och överlämna incheckade (commit) ändringar i källkoden med ett par knapptryck. När en ny uppdatering av källkoden sker kan man se exakt var i koden det har skett förändringar. Med hjälp av detta har samarbetet mellan examensarbetarna stärkts då testning kan ske direkt efter att en funktion blivit implementerad och incheckad. Även felsökning förenklades, till exempel om en gruppmedlem gjorde en ny incheckning över en stabil version och applikationen började krascha så undersökte examensarbetarna de senaste ändringarna och felet hittades och åtgärdades snabbt.

3.6.2 Android studio

Mobilapplikationen har utvecklats i Android studio. Valet gjordes eftersom företaget tidigt i examensarbetet sa att de ville ha en mobilapplikation för operativsystemet Android.

Examensarbetarna åtgärda en del problem såsom korrupta filer från tidigare installationer, inställningar som fick programmet att krascha, sparade filer som försvann eller lades i dolda mappar. När programmerandet inleddes och examensarbetarna lärde sig mer om hur Android studio fungerade visade det sig vara ett bra val för utvecklingen av mobilapplikationen.

3.6.3 JSON och Volley

I början av examensarbetet var det tänkt att använda JDBC (Java Database Connectivity) till koppling mellan databas och mobilapplikation. JDBC är ett programmeringsgränssnitt som gör det möjligt att koppla en databas till ett Java-program. Detta visade sig under utvecklingen vara väldigt invecklat och tidskrävande, eftersom Android inte har något stöd för detta. Därför togs beslutet att hitta en annan lösning för att applikationen skulle kunna hämta information från databasen.

Lösningen blev att först och främst ha ett lager (JSON) mellan applikationen och databasen. Java packar ner data i JSON-format som skickas iväg till PHP som sedan packar upp data och gör en SQL-förfrågan till databasen. Svaret från databasen packas ner i JSON-format av PHP och skickar iväg data till Java som sedan kan använda informationen i sina funktioner.

För att skicka data mellan de olika plattformarna användes Javabiblioteket Volley. Här uppstod en del problem eftersom information och dokumentation om Volley var bristfällig vilket gjorde att inlärningen tog längre tid än tänkt. Implementeringen med hjälp av Volley ledde även till en del buggar, till exempel fanns det tomma listor under körning av programmet. Dessa listor skulle innehålla data hämtad från databasen. Anledningen till detta var att hämtning av data inte hann bli klar innan funktionen kördes. Detta löstes genom att dela upp processen i flera metoder som utförde datahämtning vid uppstart av applikationen.

3.6.4 SQL

Designen av databasen gjordes i början av examensarbetet och var därmed det första som utvecklades. Det blev en hel del ändringar i databasen under arbetets gång i och med att företaget kom med fler idéer för systemet som ledde till nya funktioner följt av nya tabeller i databasen. Den slutliga databasen är därför väldigt olik den som designades i arbetets tidiga stadie. Examensarbetarna hade goda erfarenheter av SQL sedan tidigare och det uppstod inga större problem.

3.6.5 HTML, CSS och PHP

Förutom mobilapplikationen så krävde systemet även en webbapplikation. Examensarbetarna beslutade att använda PHP och HTML/CSS för detta ändamål. HTML/CSS valdes på grund av examensarbetarna tidigare erfarenheter och PHP valdes för att man ville få mer kunskap inom området. Inlärningen av PHP var enkel eftersom det är bra dokumenterat och lätt att ta in för de som läst andra programmeringsspråk. PHP var även lämpat för ändamålet och medförde inga större problem.

4 Resultat

Examensarbetet har resulterat i en prototyp av ett system som är tänkt att förenkla kommunikationen mellan företag och kund vid försäljning av produkter och tjänster.

4.1 Analys

I detta avsnitt redogörs för hur de undersökningar som beskrivs i kapitel 3 användes under examensarbetets gång.

4.1.1 Riskanalys

Riskanalysen som beskrevs i avsnitt 3.1.1 resulterade i en lista med risker enligt tabell 4.1.
Riskerna bedömdes enligt följande riktlinjer:

Låg sannolikhet: Kommer med största sannolikhet inte inträffa.
Medel sannolikhet: Kommer förmodligen inträffa.
Hög sannolikhet: Kommer definitivt inträffa, förmodligen flertalet gånger.

Låg allvarlighet: Försumbart.
Medel allvarlighet: Kan leda till små stopp i arbetsprocessen.
Hög allvarlighet: Kan leda till stora stopp i arbetsprocessen eventuellt förstöra för examensarbetets chanser att lyckas.

ID	Risk	Sannolikhet	Allvarlighet
1	Dålig kommunikation inom examensarbetsgruppen	Låg	Låg
2	Oenighet i grupp	Låg	Låg
3	Dålig kommunikation med kund	Låg	Medel
4	Moment tar längre tid än förväntat	Medel	Medel
5	Gruppmedlem blir sjuk	Medel	Låg
6	Låg erfarenhet av Android- och webbutveckling	Hög	Medel
7	Otillräcklig tid	Låg	Medel
8	Tekniska motgångar*	Hög	Medel

Tabell 4.1 *Eventuella risker.*
**Installationssvårigheter, programkrascher, att verktyg inte fungerar som tänkt etc.*

Som tabell 4.1 tydligt visar var det två av punkterna som ansågs ha störst sannolikhet att inträffa: *6. Låg erfarenhet av Android- och webbutveckling* och *8. Tekniska motgångar*, två punkter som också hör ihop. Motiveringen till detta var att examensarbetarna arbetade i stort sett i helt nya miljöer som inte ingår i utbildningen. Redan tidigt kunde dessa risker identifieras. Därför var det möjligt att i planeringen ta hänsyn till dessa risker så

att konsekvenserna inte skulle bli så allvarliga om de inträffade. Därför uppskattades allvarligheten för dessa risker som medel och inte hög.

Det finns inte några direkta åtgärder som kan vidtas för att minska konsekvenserna av risker som 6 och 8. Därför försökte examensarbetarna redan under planeringen att minska sannolikheten att riskerna inträffar. Man planerade till exempel en lång utvecklingsfas eftersom man, på grund av bristen på tidigare kunskaper, visste att programmeringen i de nya miljöerna kunde bli tidskrävande. Examensarbetarna lade även stor vikt vid att ha god kommunikation med företaget genom hela arbetet för att försäkra sig om att man hela tiden arbetade i enlighet med företagets önskemål och behov. På så sätt minskar risken att prototypen måste genomgå stora ändringar i slutet av arbetet.

4.1.2 Intressentanalys

Beskrivningen av samtliga intressenter presenteras i avsnitt 3.2.1.

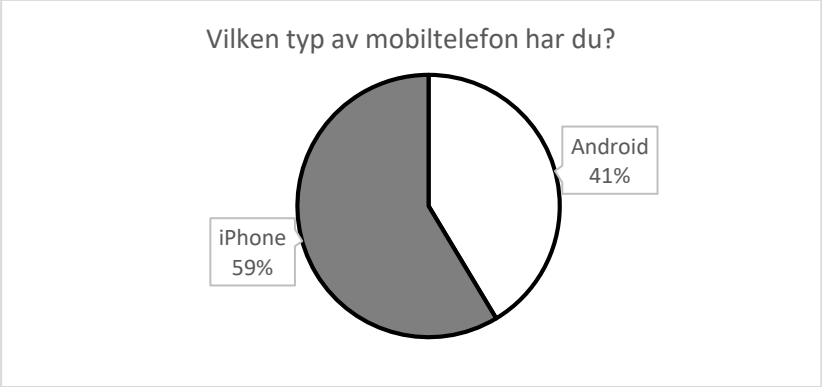
Flexilast agerade handledare under hela examensarbetet och bestämde tillsammans med examensarbetarna vad webbapplikationen och mobilapplikationen skulle innehålla. Varje beslut som togs diskuterades först av examensarbetarna och företaget. Flexilast deltog även i verifieringen och valideringen av prototypen som beskrivs i kapitel 3.4.

Den huvudsakliga användaren av mobilapplikationen användes som målgrupp. Examensarbetarna försökte välja deltagare i enkätundersökningen inom målgruppens åldersintervall och frågorna på enkäterna formades delvis utefter beskrivningen av målgruppen. Examensarbetarna gick även efter åldersintervallet när testpersoner för användartesterna valdes (se avsnitt 3.4.2).

4.1.3 Enkätundersökning

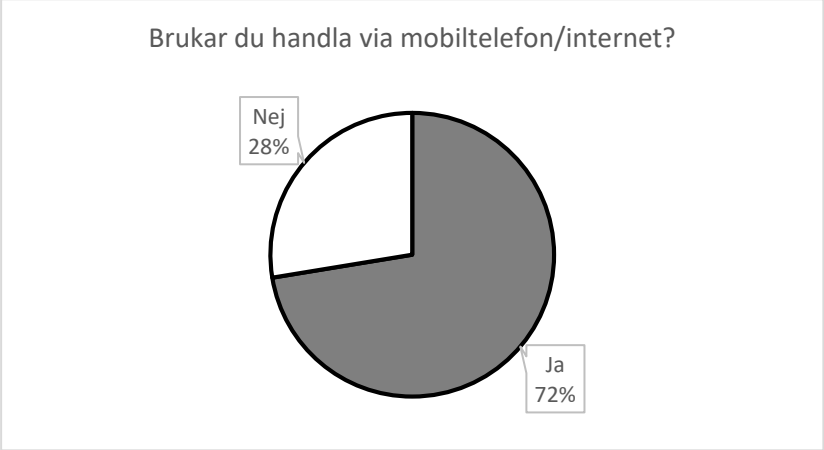
Enkätundersökningarna som beskrevs i avsnitt 3.2.2 gjordes för att
1. Undersöka det generella intresset för mobilapplikationen inom den angivna målgruppen.
2. Få information om vad som är intressant för den angivna målgruppen i en sådan mobilapplikation.

Nedan följer resultatet av enkätundersökningen med tillhörande kommentarer.



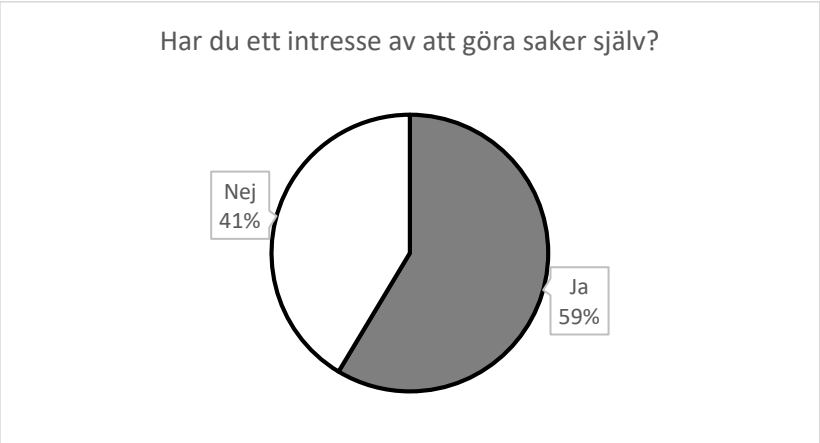
Figur 4.1. *Enkätdiagram 1.*

Det var redan innan enkätundersökningen bestämt att mobilapplikationen skulle utvecklas för Android och därför påverkar inte resultatet i figur 4.1 examensarbetet. Examensarbetarna valde ändå att ha med frågan i enkäten för att få en överblick över vilken typ av mobiltelefon som var vanligast och visa detta för företaget. Resultatet kan då finnas i åtanke vid eventuell vidareutveckling av prototypen.



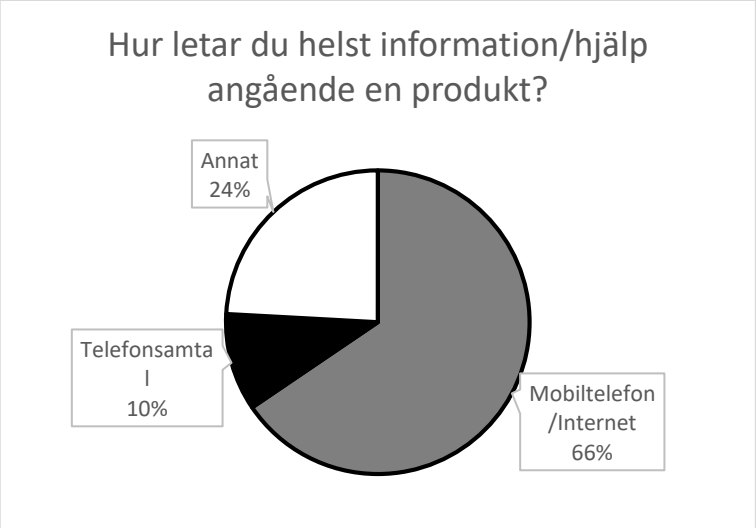
Figur 4.2. *Enkätdiagram 2.*

Resultatet i figur 4.2 visar att det idag är vanligt att kunder handlar via sina mobiltelefoner vilket examensarbetarna anser vara positivt i och med att prototypen behandlar en form av näthandel. Det finns ett tydligt behov av att kunna beställa saker genom mobiltelefoner.



Figur 4.3. *Enkätdiagram 3.*

För att ta hänsyn till företagets beskrivning ”Personer i åldern 20-35 som ska köpa bostad och är intresserade av att bygga lite själv” så valde examensarbetarna att ställa en intressefråga vars resultat visas i figur 4.3. Resultatet påverkar egentligen inte produkten på något vis, ett ointresse av att göra saker själv utesluter inte kunden från att använda mobilapplikationen.

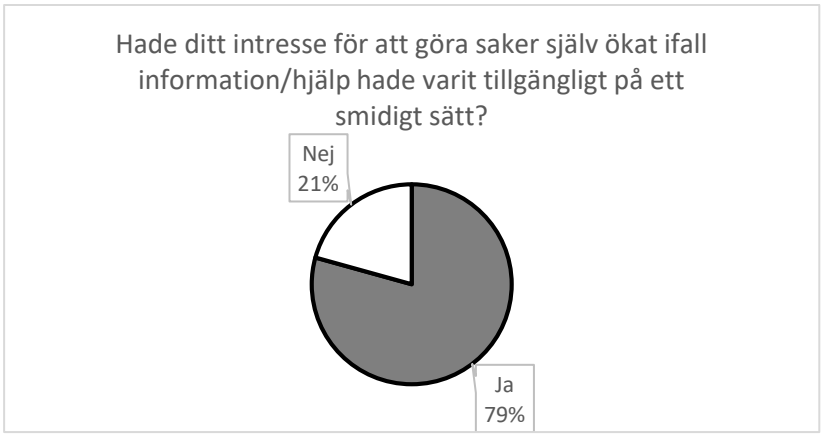


Figur 4.4. *Enkätdiagram 4.*

Nu i efterhand så borde denna frågan snarare varit ”Hur letar du helst information/hjälp angående en produkt eller tjänst”. Men det viktiga i frågan är att ta reda på hur kunder helst letar efter information eller hjälp. Svaren som presenteras i figur 4.4 visar återigen att en majoritet gärna använder telefonen när de letar efter information.

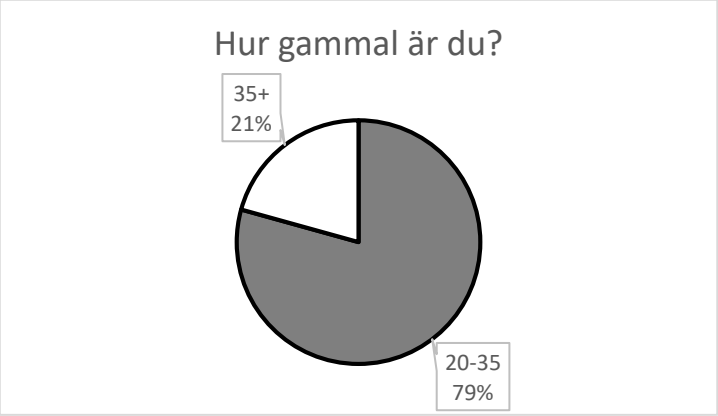
Figur 4.5. *Enkätdiagram 5.*

Redan på första mötet med företaget så diskuterades det om att mobilapplikationen skulle innehålla någon form av instruktioner. Efter att ha sett det stora intresset för instruktioner (visas i figur 4.5) så togs beslutet att implementera möjligheten för instruktionsfilmer på varje tjänsts sida i mobilapplikationen. Prototypen kommer att levereras med korta videoklipp som examensarbetarna har gjort för att illustrera möjligheten att ha instruktionsfilmer.



Figur 4.6. *Enkätdiagram 6.*

Ett alternativt sätt att undersöka intresset kunde ha varit att ställa frågan ”Hade applikationen som vi tänkt utveckla ökat ditt intresse för att göra saker själv?”. Examensarbetarna valde att ställa frågan i figur 4.6 för att kontrollera intresset för mobilapplikationen som ska utvecklas och tolkar resultatet som att intresset inte bara finns utan även är stort.



Figur 4.7. *Enkätdiagram 7.*

Som nämnt tidigare i rapporten så försökte examensarbetarna välja deltagare inom åldersintervallet 20-35. Svaren i figur 4.7 visar att man inte lyckades fullt ut med det. Vid sammanställningen av enkätresultaten så valde examensarbetarna att ändå räkna med resultaten från personer över 35 eftersom att dessa ingår i ”Sekundära användare av mobilapplikationen” som är en av intressenterna (se avsnitt 3.2.1).

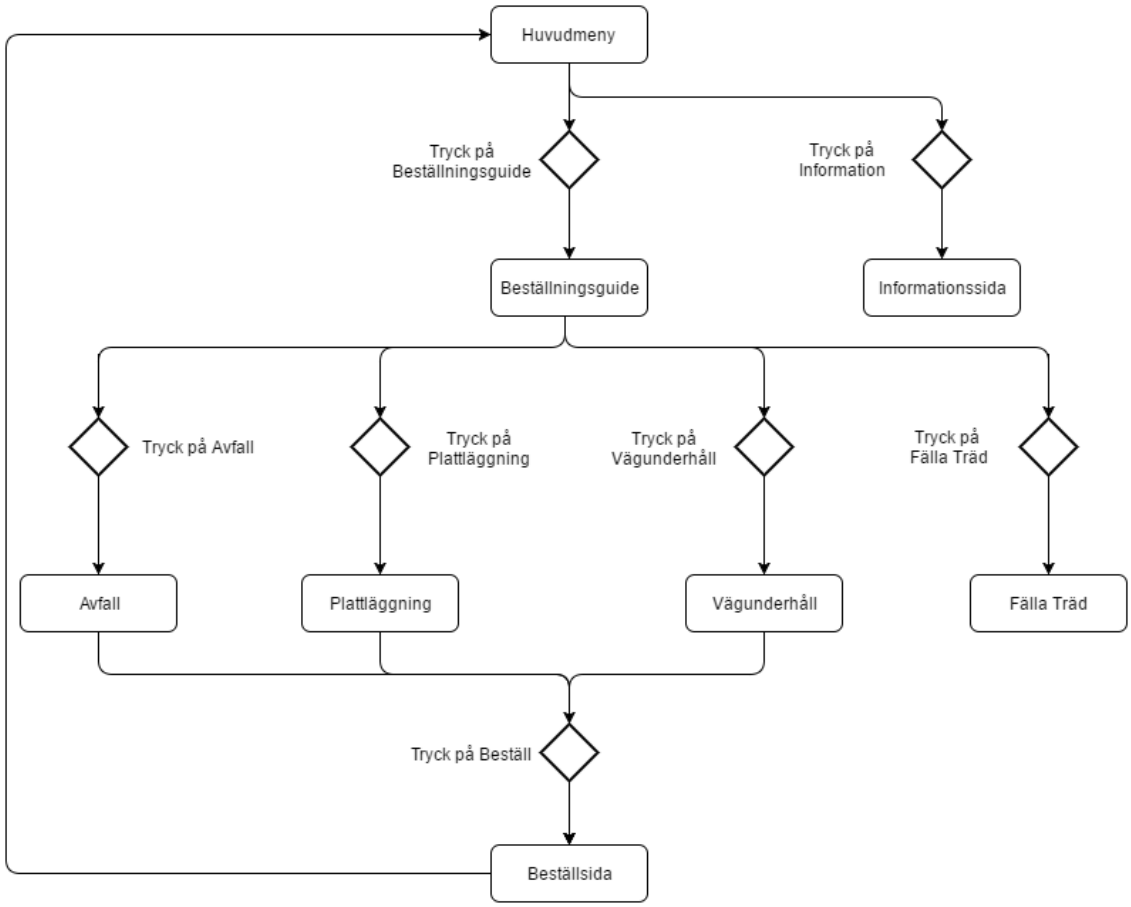
Under examensarbetets gång och efter diskussioner med företaget angående målgruppen blev det även allt mer klart att mobilapplikationen är till för alla Flexilasts kunder oavsett ålder. Åldersintervallet i den primära målgruppen sattes eftersom det är i den åldern som många köper ny bostad och det är även personer i den åldern som anses vara mest intresserade av ett nytt, modernt sätt att både få hjälp med och köpa produkter och tjänster inom branschen. Produkten är inte anpassad för en specifik åldersgrupp, men företaget tror att mobilapplikationen kommer användas mest inom åldersgruppen 20-35.

I och med att produkten är till för alla Flexilasts kunder så anses åldern på deltagarna i enkätundersökningen i efterhand vara irrelevant och resultatet i figur 4.7 har inte påverkat produkten på något vis. Det viktigaste med enkätundersökningen enligt examensarbetarna är att resultaten tydligt visar att det finns ett intresse för produkten som ska utvecklas.

4.2 Utveckling

Prototypen består av tre delar: mobilapplikation, webbapplikation och databas.
De presenteras i detta avsnitt.

4.2.1 Mobilapplikation



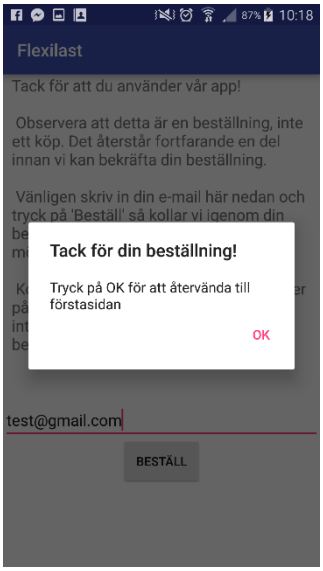
Figur 4.8. *Flödesdiagram över mobilapplikationen.*

Mobilapplikationen är tänkt att användas som en vägledning för Flexilasts kunder som är osäkra på vad som behöver beställas. Varje tjänst har en sida som innehåller en tillhörande instruktionsfilm. Under filmen kan kunden mata in egna mått, material etc. beroende på vilken tjänst som är vald. Figur 4.8 illustrerar flödet mellan de olika aktiviteterna i mobilapplikationen, där diamantsymbolerna betyder handling och rektangelsymbolerna betyder aktivitetssida (det vill säga vilken sida i mobilapplikationen som kunden befinner sig i).

Om kunden till exempel väljer ”Plattläggning” ber applikationen om mått på arean som kunden vill lägga plattor på enligt figur 4.9. Sedan beräknar applikationen hur mycket grus som behövs, hur mycket det kommer att kosta och lägger till andra kostnader

såsom frakt och visar det för kunden. Kunden kan sedan beställa denna tjänst genom att skriva sin e-postadress enligt figur 4.10. All viktig information (e-postadress, grustyp, mängd, beräknad kostnad, destination.) skickas sedan till databasen samt webbapplikationen varpå företaget kan se beställningen enligt figur 4.11 och kontakta kunden. Mer om mobilapplikationen och dess funktioner kan läsas om i Bilaga A: Kravspecifikation.

Figur 4.9. *Mobilapplikationen.*



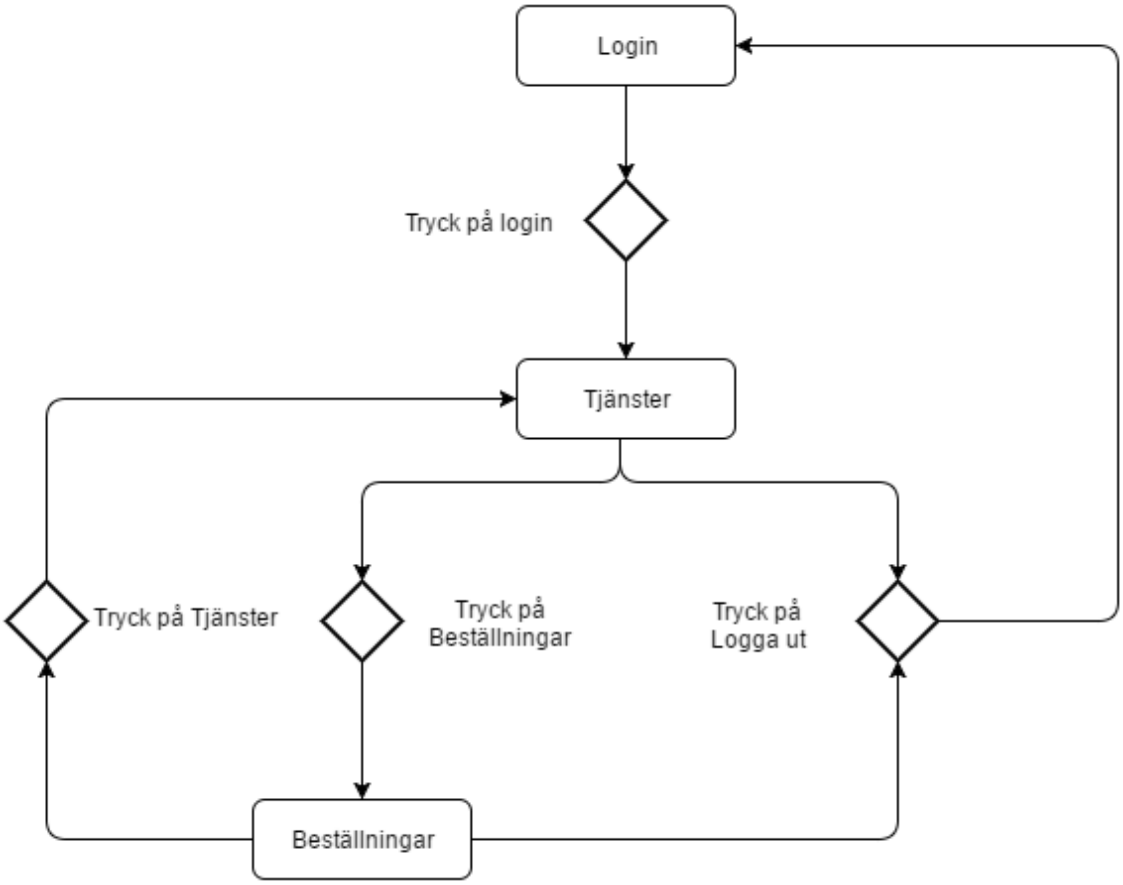
Figur 4.10. *Beställning kundvy.*

FLEXILAST

ID	Email	Service	Destination	Amount	Price	Done
9	test@gmail.com	Stenmjöl 0-8	Lund	3 ton	901	Done

Figur 4.11. *Beställning företagsvy.*

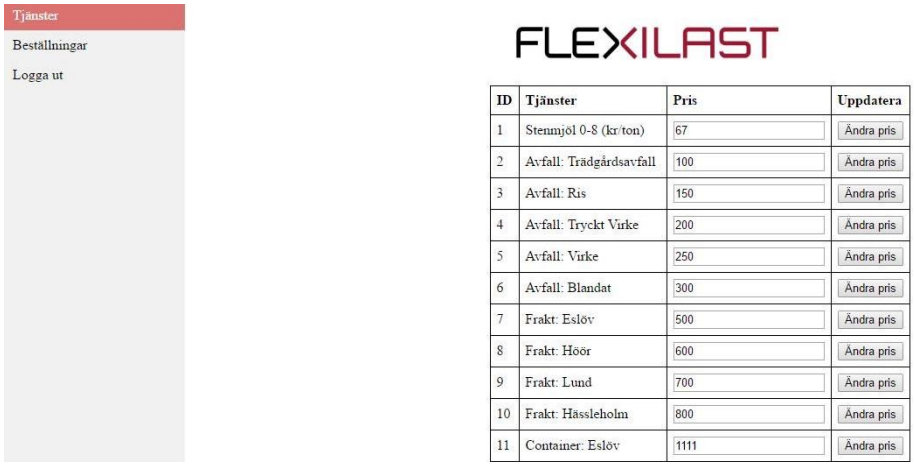
4.2.2 Webbapplikation



Figur 4.12. *Flödesdiagram över webbapplikationen.*

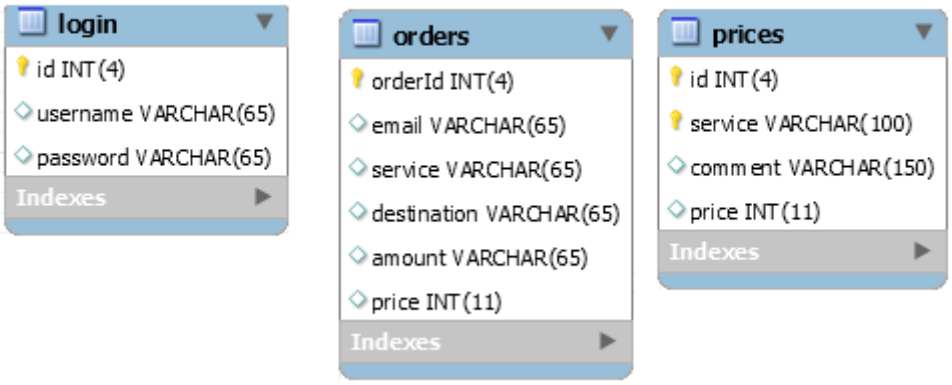
Webbapplikationen är till för att Flexilast ska kunna hantera databasen. Applikationen är strukturerad enligt figur 4.12, där diamantsymbolerna betyder handling och rektangelsymbolerna betyder aktivitetssida (det vill säga vilken sida i webbapplikationen som företaget befinner sig i). Den är utvecklad för att företaget på ett enkelt sätt ska kunna hantera tjänster och beställningar från mobilapplikationen.

Under ’Tjänster’ kan Flexilast uppdatera priserna som är aktuella i mobilapplikationen enligt figur 4.13 och under ’Beställningar’ kan de se alla beställningar som gjorts via mobilapplikationen. Varje beställning innehåller kundens e-postadress så att företaget kan kontakta kunden för att diskutera beställningen. Webbapplikationen tillåter även företaget att ta bort avklarad beställning.



Figur 4.13. *Webbapplikationen.*

4.2.3 Databas



Figur 4.14. *Databasens innehållande tabeller.*

Databasen består av tre tabeller enligt figur 4.14.

Tabellen ’login’ innehåller de användarnamn med respektive lösenord som har tillgång till webbapplikationen, tabellen innehåller ett administratörskonto när prototypen levereras till Flexilast. Tabellen ’prices’ innehåller det urval av tjänster (vid levererad prototyp följande: ’Avfall’, ’Plattläggning’, ’Vägunderhåll och ’Fälla träd’) som mobilapplikationen behandlar med tjänsternas respektive priser. Tabellen ’orders’ innehåller de beställningar som gjorts via mobilapplikationen och som presenteras i webbapplikationen.

5 Slutsats

5.1 Utvärdering

Den färdiga prototypen skiljer sig en hel del gentemot det som beskrevs i projektbeskrivningen innan examensarbetets start. Anledningen till detta är till stor del att kunden kom med nya idéer flera gånger under arbetets gång. Prototypen som från början var tänkt att vara en shop som Flexilasts kunder kunde köpa grus genom utvecklades istället till en beställningsguide som är till för att hjälpa kunderna med Flexilasts tjänster och kunna skicka beställningar som är beräknade för ändamålet.

Flexilast har under hela examensarbetet varit med och styrt utvecklingen genom att komma med nya idéer och att kontinuerligt få testa det som utvecklats. Den goda kommunikationen mellan Flexilast och examensarbetarna har lett till att företagets önskemål har kunnat uppfyllas under hela examensarbetets gång samt att risken för oenighet och missförstånd har minimerats.

Användartesterna visade att navigationen genom mobilapplikationen är enkel och intuitiv då ingen av testpersonerna hade några problem med att hitta rätt i applikationen. Det framkom också av testerna att mobilapplikationens utseende behövde förbättras vilket dock inte ligger inom ramen för examensarbetet utan lämnas till fortsatt utveckling.

Examensarbetets frågeställningar är:

1. Hur skapar man ett tillfredsställande gränssnitt till en applikation som är till för ett ej digitaliserat företag?

På första mötet med företaget diskuterades det kring gränssnitt och utseende på applikationerna. Man kom överens om att ha en enkel och lättnavigerad struktur med tydliga knappar/menyer och att det var viktigare med fungerande funktioner än ett tillfredsställande utseende. Examensarbetarna tog beslutet att lägga stort fokus på det funktionella och mindre fokus på utseendet. Därför formades inte frågorna i enkätundersökningen utefter utseende utan istället funktion.

För att upprätthålla samma fokus på funktionalitet under utvecklingsfasen så implementerades applikationerna utefter en kravspecifikation. Kravspecifikationen skrevs av examensarbetarna baserat på företagets krav och önskemål. Varje ny funktion som företaget ville ha i prototypen diskuterades och definierades i kravspecifikationen. Varje funktion testades även av examensarbetarna och företaget utifrån det som beskrivs i kravspecifikationen. Detta samarbete och den omedelbara feedbacken var viktig för att uppnå en kvalitet och funktionalitet som både examensarbetarna och företaget var nöjda med.

Utvecklingsfasen avslutades med ett större användartest av mobilapplikationen där eventuella användare fick testa applikationen efter examensarbetarnas direktiv. Examensarbetarna observerade då hur pass lätt testpersonen navigerade sig genom applikat-

ionen. Testpersonen lämnade efter testet kommentarer på applikationen i sin helhet. Resultatet av användartesterna var att man lyckats skapa en användbar och lättnavigerad struktur men att applikationen inte var estetiskt tillfredställande.

2. Vad karakteriserar kundgruppen för en applikation för försäljning av transport- och entreprenadtjänster?

Under intressentundersökningen så beskrevs målgruppen av företaget som ”Personer i åldern 20-35 som ska köpa bostad och är intresserade av att bygga lite själv”. Det framkom under examensarbetets gång att åldersintervallet inte har någon betydelse och att mobilapplikationen är till för alla Flexilasts kunder. Företaget tror att det är i åldersintervallet 20-35 som mobilapplikationen kommer användas mest. Detta styrks i viss mån av resultatet i figur 4.6 som, trots att den innehåller resultat från deltagare utanför åldersintervallet, visar att det finns ett intresse för mobilapplikationen. Resultaten från enkätundersökningen visar även att många använder sig av mobiltelefonen/internet när de behöver hjälp eller information. Man är redo att göra saker själv ifall det finns hjälp som är tydlig och lättillgänglig.

3. Hur ser en lämplig databas ut för prototypen?

Utvecklingen av databasen har skett i SQL eftersom examensarbetarna har tidigare erfarenhet av programmeringsspråket. Databasen består av tre tabeller. Den första tabellen innehåller ett administratörskonto. Den andra tabellen innehåller de tjänster som finns med i webb- och mobilapplikationen. Varje tjänst har ett enhetspris (per kubikmeter, per kilometer etc.) som också lagras i tabellen och som kan ändras via webbapplikationen. Varje tjänst har även en kommentar som beskriver tjänsten. Kommentarerna används i listan över tjänster i webbapplikationen. Den tredje tabellen innehåller beställningarna som kommer från mobilapplikationen och visas i webbapplikationen. Varje beställning består av beställarens e-postadress, tjänsten som beställts, mängden av en specifik enhet (beroende på tjänsten) och det beräknade priset.

Anledningen till att databasen är uppdelad i tre tabeller är för att det ska vara lätt att skilja på informationen som lagras. Databasen designades med de tre ovanstående tabellerna utan relationer. Det betyder att frågorna som ställs till databasen blir mindre komplicerade men medför också en risk för redundans. I nuläget får Flexilast in cirka 1500 beställningar från privatpersoner om året. Därmed löper det ingen risk för överbelastning av databasen.

4. Vilka tekniker ska användas för utvecklingen av prototypen?

För utvecklingen av webbapplikationen valde examensarbetarna att arbeta i PHP och HTML. Examensarbetarna hade tidigare erfarenheter av HTML och det föll därför naturligt att använda det. Valet av PHP gjordes efter som kopplingen mellan PHP och

SQL (databasen) är enkel att implementera och förstå. Examensarbetarna ville dessutom få ökade kunskaper inom PHP.

Mobilapplikationen, som är utvecklad för Android-enheter, har utvecklats i Android studio som använder Java som programspråk. Java passade bra eftersom Android har ett eget Java-ramverk samt att examensgruppen har mycket erfarenhet av språket. Det var i början tänkt att koppla mobilapplikationen med SQL-databasen med hjälp av JDBC (Java Database Connectivity). Tyvärr visade det sig att Android inte har något stöd för detta. Istället beslutades det, efter sökning av ett annat alternativ, att använda Java HTTP-biblioteket Volley, som skickar data i JSON-format. Valet att använda Volley gjordes eftersom att det är ett komplett bibliotek och därmed tidssparande. På så vis undvek examensarbetarna att behöva bygga nätverksdelen i applikationen från grunden.

5.2 Begränsningar och vidareutveckling

I och med att mobilapplikationen är en prototyp så innehåller den inte alla tjänster som företaget erbjuder utan gruppen fick ett urval av tjänster från företaget som de tyckte skulle vara med. Den väl kommenterade koden hos prototypen gör det enkelt att implementera fler tjänster under beställningsguiden eller till och med helt nya funktioner.

Prototypen är utvecklad för att fungera lokalt med en lokal databas. För åtkomst till Internet krävs vidare utveckling av mobilapplikationen och en uppsättning av webbserver. Eftersom att prototypen är utvecklad för att fungera lokalt så har heller inget fokus lagts på säkerhet. Mobilapplikationen är i levererat tillstånd känsligt för SQL-injektioner, det vill säga att på beställningssidan där man skriver in sin mail skulle användaren kunna skriva en SQL-fråga för att skada eller ändra databasen. En annan risk som kan uppkomma i mobilapplikationen är att ifall två användare gör vars en beställning samtidigt så kan det innebära att en av beställningarna försvinner. I webbapplikationen är det möjligt att hoppa mellan olika sessioner genom adressfältet.

Mobilapplikationen är utvecklad enbart för Androidenheter. Detta är på grund av att företaget ville att applikationen skulle vara utvecklad i Android. Enkätundersökningen visar en majoritet av iPhone-användare. Det kan vara bra att ha i åtanke vid vidareutveckling och utveckla en iPhone-version av mobilapplikationen för att nå fler kunder.

Utseendet på applikationerna var ingen prioritet under examensarbetet och resultatet av användartesterna bestod framförallt av kommentarer på det estetiska. Vid eventuell vidareutveckling kan företaget ändra utseendet i prototypen utefter deras och användarnas önskemål.

Genom hela utvecklingsfasen har mobilapplikationen testats på endast två enheter, en med 5.2” skärm och en med 5,1” skärm. Skulle man starta applikationen på en enhet med större eller mindre skärm så är det inte säkert att gränssnittet är optimerat för just den skärmen. Anledningen till detta är att applikationen har utvecklats efter och testats med examensarbetarnas egna mobiltelefoner.

Prototypen kan även utvecklas till en komplett webbshop genom att koppla ihop den med ett affärssystem.

6 Referenser

About – Git. (u.å.). Hämtad 2016-08-13 från
https://git-scm.com/about

Android Studio, The Official IDE for Android (u.å.). Hämtad 2016-08-22 från
https://developer.android.com/studio/index.html

Android, the world's most popular mobile platform (u.å.). Hämtad 2016-08-22 från
https://developer.android.com/about/android.html

Barnum, C. (2011). *Usability Testing Essentials.* Elsevier B.V.

Chacon, S (2009). *Pro Git.* CA: Apress.

Eriksson, M. & Lilliesköld, J. (2005). *Handbok för mindre projekt.* Liber

IBM (u.å.). Hämtad 2016-08-12 från
http://www.ibm.com/developerworks/xml/tutorials/xmlintro/xmlintro.html

Introducing JSON (u.å.). Hämtad 2016-08-14 från
http://www.json.org/

Kniberg, H. & Skarin, M. (2010). *Kanban and scrum making the most of both.* C4Media

Layouts (u.å.). Hämtad 2016-08-12 från
https://developer.android.com/guide/topics/ui/declaring-layout.html

MacIntyre, P (2010) *PHP: the good parts.* CA: O'Reilly

Park, J. & Park, M. (2016). *Qualitative versus Quantitative Research Methods: Discovery or Justification?.* Journal of Marketing Thoughts

Stakeholder – Businessdictionary (u.å.). Hämtad 2016-10-17 från
http://www.businessdictionary.com/definition/stakeholder.html

Target audience – Businessdictionary (u.å.). Hämtad 2016-10-17 från
http://www.businessdictionary.com/definition/target-audience.html

Tetteroo, D. & Markopoulos, P. (2015). *A Review of Research Methods in End User Development.* Springer International Publishing Switzerland

Transmitting Network Data Using Volley (u.å.). Hämtad 2016-08-14
https://developer.android.com/training/volley/index.html

Yaghmour, K (2013). *Embedded Android*. CA: O'Reilly Media.

Kravspecifikation

Flexilast beställningsguide

2016-08-09

Innehållsförteckning

1. Introduktion

Målet med detta dokument är att lista och specificera krav som skall förtydliga och vidare beskriva det som står i projektbeskrivningen.

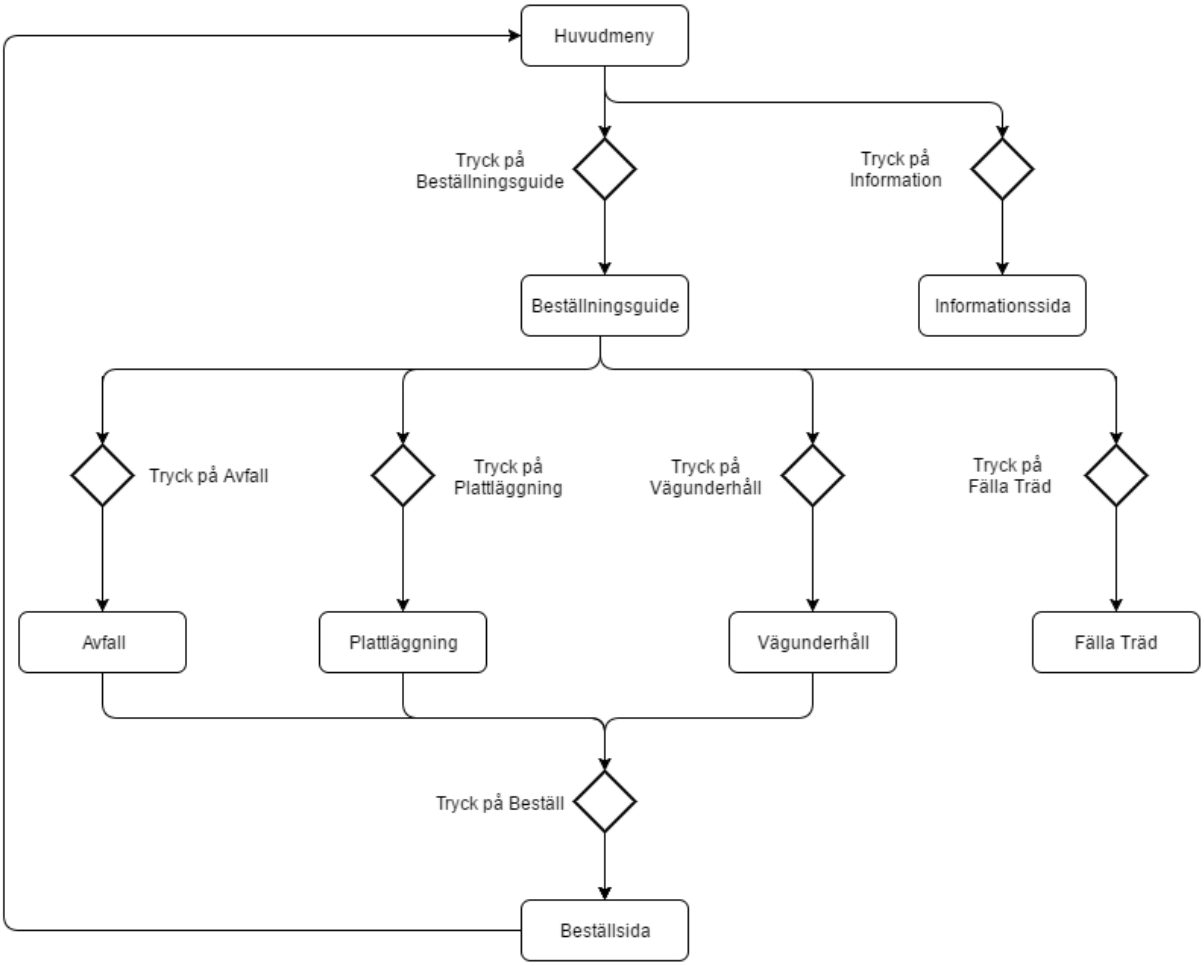
2. Referenser

1. Projektbeskrivningen
2. Mobilapplikationen
3. Webbapplikationen

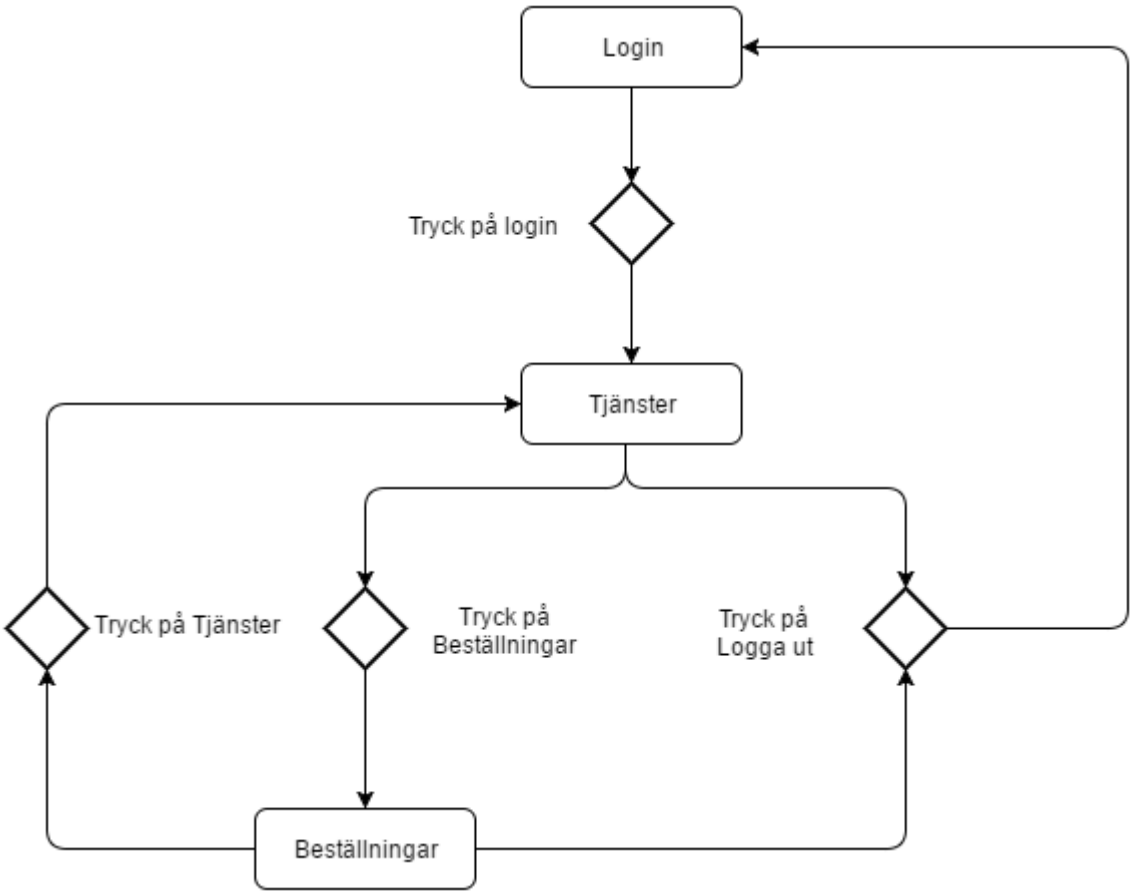
3. Terminologi

Admin – Användare av webbapplikationen
Användare – Användare av mobilapplikationen

4. Flödesdiagram



Figur 1. Flödesdiagram över mobilapplikationen



Figur 2. Flödesdiagram över webbapplikationen.

5. Krav

5.1 Webbapplikation

5.1.1. Vid start av applikationen ska inloggningssida enligt bild 1 visas
5.1.2. Admin ska få åtkomst till applikationens funktioner genom ett administrationskonto
5.1.3. När admin loggat in ska sida enligt bild 2 visas
5.1.4. På sidans vänstra sida ska en meny med följande alternativ visas:

1. Tjänster (Förstasidan)
2. Beställningar
3. Logga ut

5.1.5. På sidan ”Tjänster” ska en lista med följande information visas:

1. Beskrivning av en tjänst
2. Tjänstens pris

5.1.6. Admin ska kunna redigera priset på en tjänst genom att skriva ett nytt pris i prisfältet och sedan trycka på ”Ändra pris”-knappen
5.1.7. När admin trycker på ”Ändra pris”-knappen ska en dialog enligt bild 3 visas
5.1.8. Inmatning av ett pris får endast vara i form av heltal.
5.1.9. På sidan ”Beställningar” ska en lista med följande information visas:

1. Beställarens e-postadress
2. Beställd tjänst
3. Destination
4. Mängd
5. Uppskattat pris

5.1.10. Admin ska kunna ta bort en beställning genom att trycka på ”Avklarad”-knappen
5.1.11. När admin trycker på ”Avklarad”-knappen ska en dialog enligt bild 4 visas
5.1.12. Admin ska kunna logga ut genom att trycka på ”Logga ut” i menyn till vänster

Bild 1. Login-sida

Bild 2. Sida för tjänster / Förstasidan

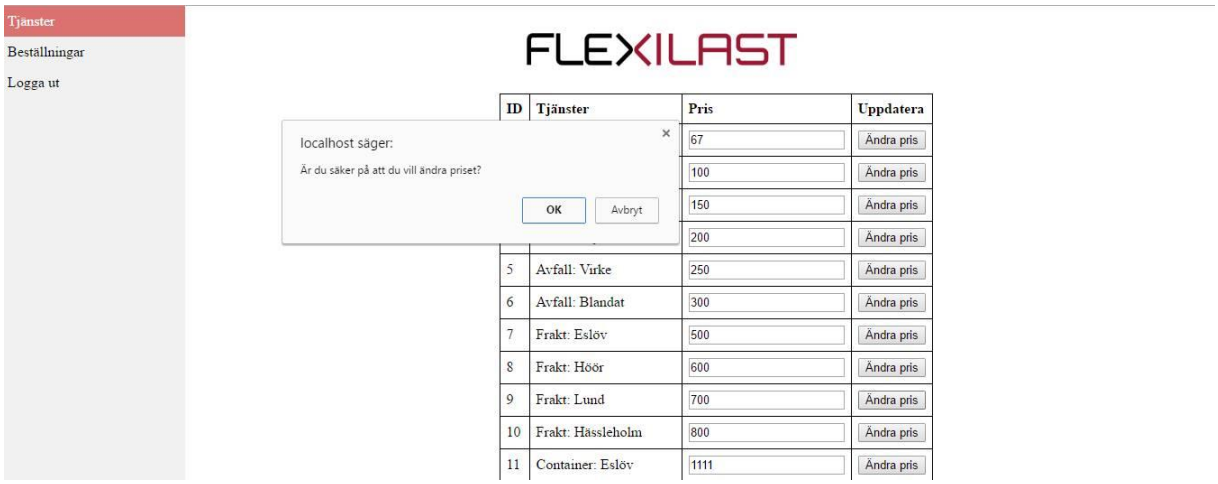


Bild 3. Dialogruta vid ändrat pris

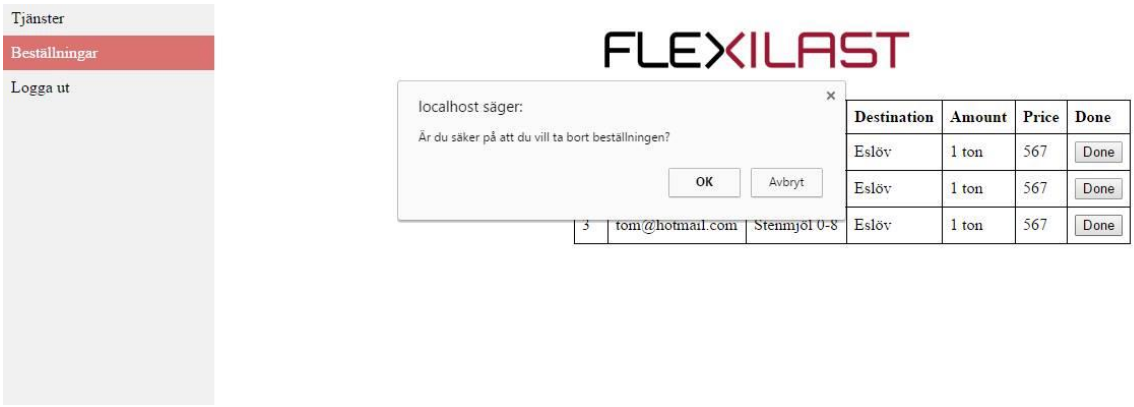


Bild 4. Dialogruta vid borttagen beställning.

5.2 Mobilapplikation

5.2.1 Huvudmeny

5.2.1.1. Vid start av applikationen ska huvudmeny enligt bild 5 visas

5.2.1.2. Huvudmenyn ska innehålla följande knappar:

1. Beställningsguide
2. Information

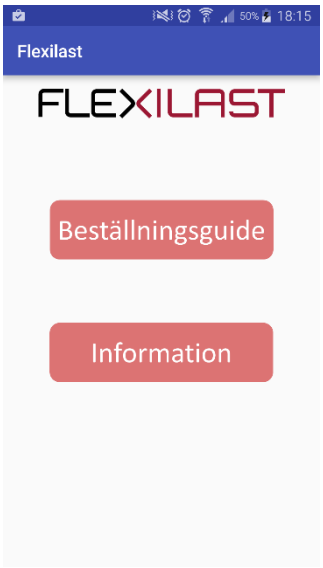


Bild 5. Huvudmeny

5.2.2 Beställningsguide

5.2.2.1. Beställningsguidens huvudsida ska innehålla följande knappar:

1. Avfall
2. Plattläggning
3. Vägunderhåll
4. Fälla träd

5.2.2.2. Varje knapp ska leda användaren till respektive aktivitets sida
5.2.2.3. Varje aktivitetssida ska innehålla en instruktionsvideo för respektive aktivitet
5.2.2.4. Sidan ”Avfall” ska innehålla följande dropmenyer:

1. Volym och plats
    a. Stor säck Eslöv
    b. Stor säck Höör
    c. Stor säck Lund
    d. Stor säck Hässleholm
    e. Container Eslöv
    f. Container Höör
    g. Container Lund
    h. Container Hässleholm
2. Avfallssort
    a. Trädgårdsavfall
    b. Ris
    c. Tryckt virke
    d. Virke
    e. Blandat

5.2.2.5. På sidan ”Plattläggning” ska användaren kunna mata in följande:

1. Längd i meter
2. Bredd i meter

5.2.2.6. Sidan ”Plattläggning” ska innehålla följande dropmenyer:

1. Fraktdestination
    a. Eslöv
    b. Höör
    c. Lund
    d. Hässleholm

5.2.2.7. Sidan ”Vägunderhåll” ska innehålla följande dropmenyer:

1. Tjänst inom vägunderhåll
    a. Hyvling
    b. Spridning
    c. Dammbindning
2. Fraktdestination
    a. Eslöv
    b. Höör
    c. Lund
    d. Hässleholm

5.2.2.8. På sidan ”Vägunderhåll” ska användaren kunna mata in följande:

1. Avstånd i kilometer

5.2.2.9. Varje aktivitetssida (undantag: Fälla träd) ska innehålla en ”Beräkna”-knapp
5.2.2.10. När användaren trycker på ”Beräkna”-knappen ska en dialogruta enligt bild 6 visas
5.2.2.11. När användaren trycker på ”Avbryt”-knappen i dialogrutan ska användaren skickas tillbaka till aktuell aktivitetssida

5.2.2.12. När användaren trycker på ”Beställ”-knappen i dialogrutan ska en ny sida enligt bild 7 visas
5.2.2.13. På sidan som bild 7 visar ska användaren kunna mata in sin e-postadress
5.2.2.14. Användaren ska kunna skicka information om sin beställning till databasen via en ”Beställ”-knapp
5.2.2.15. När användaren trycker på ”Beställ”-knappen ska följande information skickas till databasen:

1. Beställarens e-postadress
2. Beställd tjänst
3. Destination
4. Mängd
5. Uppskattat pris

5.2.2.16. När användaren trycker på ”Beställ”-knappen ska en dialogruta enligt bild 8 visas
5.2.2.17. När användaren trycker på ”OK”-knappen på dialogrutan ska användaren skickas tillbaka till applikationens huvudmeny

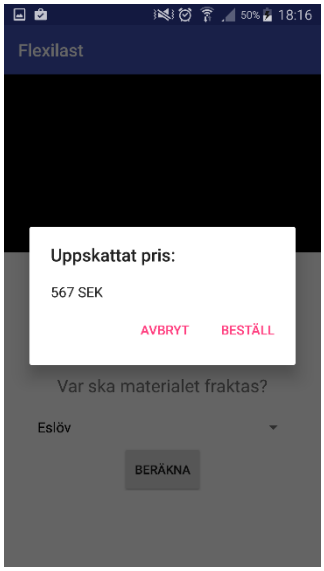


Bild 6. Beräknat pris

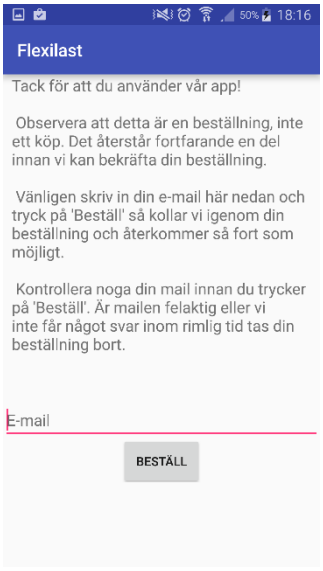


Bild 7. Beställsida

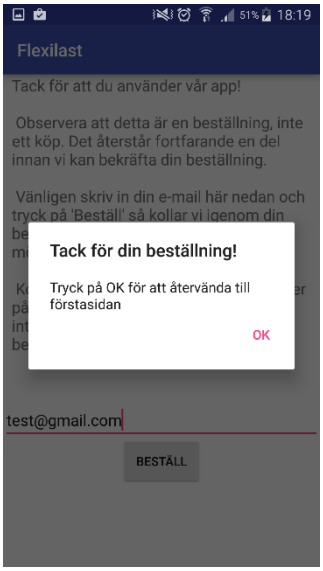


Bild 8. Dialogruta beställ

6. Leveranskrav

Systemet ska levereras i en mapp innehållande alla nödvändiga filer för att systemet ska fungera lokalt hos kunden.