



# **En webbapplikation för Pargas stad**

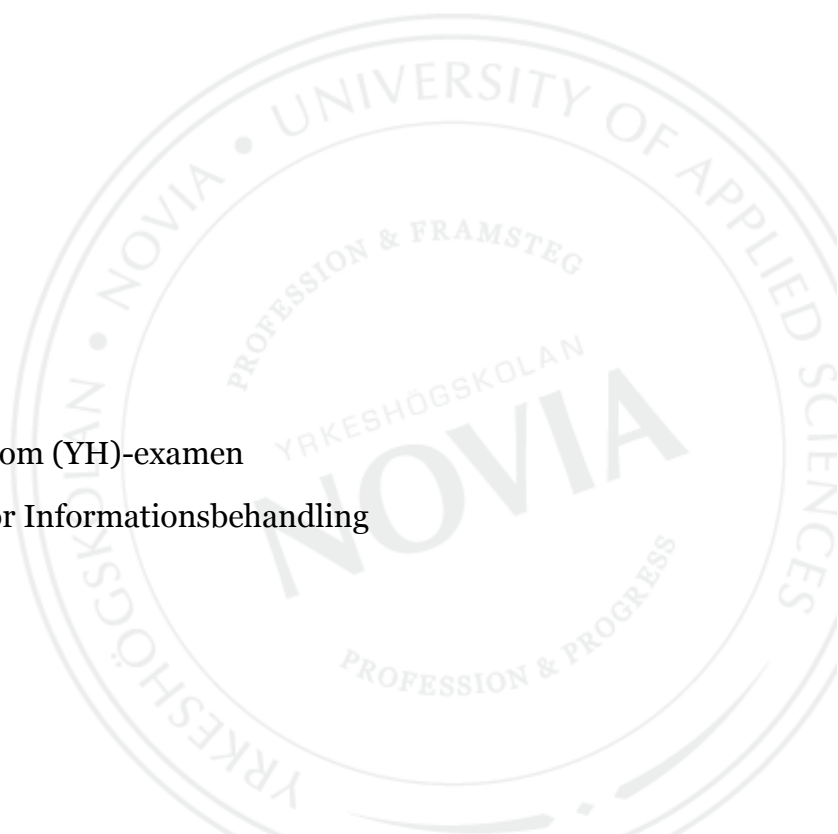
## **Utveckling och implementering av ett användargränssnitt**

Lönnberg Robert

Examensarbete för Tradenom (YH)-examen

Utbildningsprogrammet för Informationsbehandling

Raseborg 2016



## **EXAMENSARBETE**

Författare: Lönnberg Robert

Utbildningsprogram och ort: Informationsbehandling, Raseborg

Handledare: Gammals Rolf

Titel: En webbapplikation för Pargas stad – Utvecklingen och implementeringen av designen

---

Datum 13.4.2016

Sidantal 33

Bilagor 0

---

### **Abstrakt**

Examensarbetet beskriver utvecklandet och implementeringen av designen till en webbapplikation till Pargas stad, med hjälp av Drupal 7, Drupal-moduler och med kodspråket CSS. Applikationens uppgift är att förbättra informationsflödet mellan lokala företag, ortsbor och turister. Applikationen förser användaren med information om företagens geografiska läge, uppehållstider samt kontaktuppgifter. Även nyheter, arbetsplatsannonser, aktiviteter och tidtabeller för bussar, färjor samt tåg finns i applikationen. Applikationens uppgift är att få ortsborna att handla hos de lokala företag och inte gå till de stora butiks kedjorna.

Examensarbetet förklarar processen från designutkast till implementering till applikationen med hjälp av Drupal 7, Drupal-moduler och med kodspråket CSS. Examensarbetet förklarar också motiveringarna bakom de val jag gjort angående designen och navigeringen.

Slutprodukten av examensarbetet är designen och navigeringen till webbapplikationen för Pargas stad. Målet med designen och navigeringen var att göra en lätt använd webbapplikation, där användaren lätt kan lägga till material eller hitta information hen söker. Resultatet är en användarvänlig applikation som gör att man gärna fortsätter att använda applikationen.

---

Språk: Svenska

Nyckelord: Drupal, moduler, Pargas, webbapplikation, CSS

---

## **BACHELOR'S THESIS**

Author: Lönnberg Robert

Degree Programme: Business Information Technology, Raseborg

Supervisor: Gammals Rolf

Title: A Web Application for the Town of Pargas – Development and Implementation of the Design

---

Date 13 April 2016

Number of pages 33

Appendices 0

---

### **Summary**

This thesis describes the development, implementation and design of a web application for the town of Pargas, with the help of Drupal 7, Drupal 7-modules and with the code language CSS. The application provides information about the geological location, opening hours and contact information of local companies. News, job advertisements, activities, timetables for buses, ferries and trains are also provided. The mission of this application is to improve the communication and make the citizens of Pargas choose local companies instead of the big brands.

This thesis describes the process of starting with a design draft and how to implement it to the application with the help of Drupal 7, Drupal 7-modules and with the code language CSS. The thesis also explains the choices I have made with the design and the navigation.

The final product is an easy-to-use web application for the town of Pargas. The mission of the design and the navigation is to make the application user-friendly, so that the user easily can add material or find information. The result of this is a user-friendly application which will retain its users.

---

Language: Swedish

Key words: Drupal, modules, Pargas, web application, CSS

---

# Innehållsförteckning

1	Inledning .....	1
1.1	Syfte och målsättning.....	1
1.2	Bakgrund.....	1
1.3	Omfattning .....	2
1.4	Tillvägagångssätt .....	2
1.5	Pargas stad .....	3
2	Drupal .....	4
2.1	Installation av Drupal.....	5
2.2	Installation av moduler.....	6
2.3	Installation av tema .....	6
2.4	Val av tema .....	7
3	Definitioner .....	8
3.1	Olika versioner av HTML och CSS.....	8
3.2	CSS .....	8
3.3	CSS-kodoptimering.....	9
4	Designen .....	11
4.1	Regler för design.....	11
4.2	Användarvänlighet .....	12
4.3	Planeringen av designen.....	13
4.4	Designutkastet.....	14
5	Implementering av designen .....	16
5.1	Designen godkändes .....	16
5.2	Framsidan.....	17
5.3	Sida 2 – undermenyerna.....	19
5.4	Designen till underrubrikerna .....	20
5.5	Navigeringen mellan framsidan och undermenyerna .....	21
5.6	Bakåtknappen och hemknappen.....	22
5.7	Mobile sliding menu .....	23
6	Program som använts.....	25
6.1	Adobe Dreamweaver.....	25
6.2	Adobe Photoshop .....	25
6.3	WinSCP.....	26
6.4	Google Chrome Developer Tool.....	27

7	Sammanfattning.....	28
7.1	Resultat och reflektioner.....	30
7.2	Vidareutveckling.....	31

# 1 INLEDNING

Applikationer används i allt större utsträckning och det finns applikationer till många olika ändamål för att hjälpa och underhålla användaren. Examensarbetet presenterar en webbapplikation för Pargas stad. Jag kommer att beskriva processen från att få applikationens designutkast färdigt till att ha designutkastet implementerat till applikationen och vilka saker man bör tänka på då man gör en design till en applikation.

## 1.1 SYFTE OCH MÅLSÄTTNING

Jag deltog i ett team som fick som uppgift att framställa en applikation för Pargas stad och projektet var en del av ett större projekt vid namn ”Den levande skärgården i Pargas stad”. Förbättringen av informationsflödet skulle gynna de lokala företagen och de tjänster som erbjuds i Pargas. Min uppgift och även mitt syfte, är att skapa användargränssnittet till applikationen samt att implementera det. Målet med mitt examensarbete är att skapa användargränssnittet till applikationen som är up-to-date och att applikationen är användarvänlig.

Effektiva målet med examensarbetet är att skapa en applikation som är lätt att använda, informationen som söks hittas och detta skall leda till att användaren fortsätter att använda applikationen. Applikationen är till för de lokala företagen, Pargasborna samt turister och applikationen kommer att innehålla information om t.ex. restaurangers öppethållningstider, kontaktuppgifter, geografiska läge, aktuella aktiviteter, jobbbannonser och nyheter. Applikationen kommer också att innehålla tidtabeller för bussar, tåg och färjor. Avsikten är att den färdiga applikationen skall tas i bruk även på orter än Pargas.

## 1.2 BAKGRUND

Detta projekt fick sin start av Robin Elvström och Jonas Norrman. Elvström och Norrman är tradenomstuderande i Yrkeshögskolan Novia, Åbo. Elvström och Norrman kontaktade Novia Raseborg och utbildningsprogrammet för Informationsbehandling, eftersom de behövde hjälp med att få applikationen programmerad. De sökte tredje årets studerande som var i den

situationen att de skulle börja arbeta på sitt examensarbete. Jag och två studerandekollegor, Cedric Rantanen och Benjamin Duncker, valde att ta detta som vårt examensarbete.

I och med detta hade vi nu ett team som bestod av Elvström, Norrman, Rantanen, Duncker och jag. Vårt uppdrag var att göra en applikation till Pargas stad. Elvström och Norrman påpekade i ett tidigt skede av projektet, att Pargas stad var villiga att investera i applikationen, men inte förrän de såg några resultat.

### **1.3 OMFATTNING**

I detta examensarbete kommer jag att presentera gränssnittet till applikationen, mina motiveringar bakom valen jag gjort och hur jag implementerade gränssnittet till applikationen. Jag kommer att använda en del tekniska termer och en del kodsträngar. Kodsträngarna jag presenterar kommer jag att förklara hur de fungerar och vad de gör.

### **1.4 TILLVÄGAGÅNGSSÄTT**

I början av projektet delade vi in applikationen i tre delar och tanken var att Rantanen, Duncker och jag, alla skulle ha en egen del att arbeta på. Vi delade in applikationen i följande delar; första delen var applikationens funktioner, den andra var databas, säkerhet och sekretesser och den tredje var design och layout. Den första indelningen var att Rantanen skulle arbeta på funktionerna, Duncker på databas, säkerhet och databaser medan jag skulle arbeta på designen och layouten.

Efterhand märkte Rantanen och Duncker dock att funktionerna, databas, säkerhet och sekretesser behandlade ungefär samma saker och därmed beslöt de sig för att slå ihop sig och arbeta tillsammans. Rantanen och Duncker arbetade alltså tillsammans med funktionerna och sakerna kring det, medan design och layout var fortfarande min uppgift.

Första mötet vi hade med Elvström och Norrman var 30 maj 2015. Detta möte hade vi via Skype och för att komma ihåg allt av mötet, spelade vi in detta möte för säkerhets skull. Efter första mötet hade vi möten med jämna mellanrum, från varje vecka till cirka var tredje vecka var frekvensen vi använde oss av, beroende på i vilket skede projektet var. Mötena hade vi via Skype.

Under dessa möten diskuterade vi bl.a. hur det gick för var och en, vad alla hade gjort sedan förra mötet och vad alla skulle fortsätta att arbeta på, samt andra saker som hade med applikationen att göra. Tidpunkten för nästa möten diskuterade vi via vår Facebook chatgrupp. Till största delen var det Elvström som föreslog datumet för nästa möte. I och med att vi hade vår grupp-chat på Facebook kunde vi säkerställa att dagen och tiden passade alla teammedlemmar. Vissa gånger hade vi extra möten då Elvström hade kommit på en ny idé till applikationen.

Jag arbetade hemifrån under projektets gång. Eftersom Elvström fungerade som projektledare tog han ofta kontakt med mig personligen via Facebook. Han fråga hur det gick för mig, om jag behövde hjälp med något eller om han kunde göra något för att hjälpa till. Jag hade nu och då möten med Elvström via Skype om vissa saker som var aktuella. Under dessa möten diskuterade vi eventuella problem jag hade och hur jag kunde göra för att slippa vidare i implementeringen av min design eller om funktioner som tillhörde designen Vi bollade även med nya idéer som vi kunde lägga till applikationen. Idéerna som presenterades var vissa gånger utmanande att få implementerade.

Jag och Elvström hade samma åsikt om att vi ville göra en applikation som var inte som alla andra. Med andra ord, göra en applikation som inte var en i mängden. Vårt mål var att applikationen skulle ha något annorlunda och nytt i sig och på detta sätt bli populär.

## **1.5 PARGAS STAD**

Pargas stad ligger i sydvästra Finland, sydväst från Åbo. Pargas stad har ca 15 500 invånare. Pargas stad består av före detta kommunerna; Houtskär, Iniö, Korpo, Nagu samt Pargas. Dessa kommuner slog ihop sig 2009 till en ny kommun och bestämde sig 2012, att denna kommun skall börja heta Pargas stad. Staden är tvåspråkig men större delen av invånarna i Pargas talar svenska. Pargas stad har tusentals öar, holmar och skär. Pargas är en skärgårdsstad och sättet man förflyttar sig med är färjor och förbindelsefartyg. Färjorna och förbindelsefartygen går med jämna mellanrum och alla har sina egna rutten.

Pargas stad strävar till att erbjuda tjänster, arbete, boende och transport oberoende av var i Pargas man bor och på sommaren får staden många besök av sommargäster. (Pargas, u.å.).



## 2 DRUPAL

Vi försökte arbeta med DrupalGap i början av projektet, men vi märkte efter hand att allt vi försökte göra tog länge och vi gjorde inte så många framsteg som vi hade önskat, och därmed beslöt vi oss för att lämna DrupalGap. Vi hade ett möte där vi diskuterade hur vi ska fortsätta och det vi kom fram till var, att vi gör en webbapplikation som har Drupal som bas.

DrupalGap är ett open-source applikationsutvecklingsverktyg för Drupal webbsidor. Med DrupalGap kan man bygga skilda mobil applikationer för olika operativsystem och webbappar för kommunikation med Drupal webbsidor. (DrupalGap, u.å.)

Drupal är ett CMS (Content Management System) som grundar sig på öppen källkod. Med hjälp av Drupal, som är gratis, är det möjligt att göra webbsidor i ett webbgränssnitt och därmed behöver användaren inte ha kunskaper i kodning. I detta projekt använde vi oss av innehållshanteringssystemet Drupal 7.

Drupal fick sin start år 2000, då Drupals grundare Dries Buytaert byggde en plats där han och hans studiekollegor kunde prata och dela information om vad de håller på med. Driers fortsatte att utveckla vad som idag kallas Drupal och nästa funktion som tillades var en slags anslagstavla, där Dries och hans vänner kunde publicera information. Dries valde att använda domännamnet "Dorp.com". Dorp är ett holländskt ord och betyder "by" på engelska. Dries gjorde dock ett skrivfel och domännamnet blev "drop.com" istället för "dorp.com". Projektet som var ett personligt experiment, lockade till sig nya medlemmar p.g.a. skrivfelet. Medlemmarna höll diskussioner samt utvecklade idéer över hur de kan fortsätta att utveckla detta experiment. År 2001 släppte Dries mjukvaran bakom drop.com. Idén var att låta andra använda sig av mjukvaran och på samma gång ge möjligheten till att utveckla den. Dries kallade mjukvaran för "Drupal". (Drupal, u.å.).

Drupal används på miljontals webbsidor för tillfället. Själva Drupal utvecklas och uppdateras regelbundet av en stor mängd människor. Förutom att dessa människor gör uppdateringar till stommen i Drupal, gör dessa även arbete med att upprätthålla och utveckla moduler. Moduler möjliggör att användaren får en sådan webbsida hen vill ha och på detta sätt möta användarnas

eller kundernas krav. Modulerna hämtar flexibilitet och nya funktioner till webbsidan och allt fler teman i Drupal anpassar sig även till smarttelefoner och tabletter.

Kodningsspråket PHP är vad innehållshanteringssystemet Drupal grundar sig på och Drupal kan köras på olika webbservrar. Drupal kräver en databas, där den kan lagra data och inställningar samt behöver den tillgång till denna databas med hjälp av t.ex. MySQL. Innehållshanteringssystemet Drupal används av privatpersoner, små och stora organisationer och företag.

Det kommer regelbundet ut nya uppdateringar till Drupal Core. Det finns s.k. minor updates och major upgrades. Med en major upgrade menar man en övergång från t.ex. Drupal 6.x till Drupal 7.x. Minor update betyder att det finns en mindre uppdatering och det finns då en nyare version av den Drupal version som används, t.ex. Drupal 7.39 till Drupal 7.41. I Drupal Core finns det några grundläggande funktioner som så gott som alla innehållshanteringssystem har. Dessa grundläggande funktioner är bl.a. moduler, skapandet av webbsidor, registreringsmöjligheter, rollhantering, administrering av webbsidan och menyer, byte och konfigurering av tema för att nämna några funktioner. Med hjälp av dessa grundläggande funktionerna kan användaren göra en simplare webbsida eller en personlig blogg med hjälp av byte och konfigurering av temat. Ifall användaren behöver en mera avancerad sida med funktioner är mera avancerade moduler eller plugins nödvändiga.

## **2.1 INSTALLATION AV DRUPAL**

Att installera Drupal är relativt lätt. Det var Duncker samt Rantanen som utförde denna process och de installerade Drupal 7 till operativsystemet Linux. Installationsprocessen för att installera Drupal varierar lite beroende på till vilket operativsystem Drupal installeras. I detta fall installerades Drupal på en Linux server och eftersom Linux server saknar GUI (Graphical User Interface) eller grafiskt gränssnitt, använde Duncker och Rantanen olika kommandon för att installera Drupal. Installationsprocessen finns bättre beskriven i deras examensarbete (Duncker och Rantanen 2015, s. 6-7).

Efter att Drupal var installerat, var nästa steg att navigera till adressen <http://91.150.11.4/> med webbläsaren. [Http://91.150.11.4/](http://91.150.11.4/) var IP-adressen till servern som Drupal var installerad på.

## 2.2 INSTALLATION AV MODULER

För att få mera funktioner eller egenskaper till en webbsida är moduler det snabbaste och effektivaste sättet att åstadkomma detta. I januari var antalet moduler till alla Drupal versioner över 33 000. Till Drupal 7, versionen vi använde oss av, var antalet över 11 500. När användaren vill ha en ny modul skall hen gå till Drupals hemsida och ladda ner den därifrån. Det rekommenderas att välja den senaste stabila versionen för den version av Drupal som används.

Att installera moduler kan göras på två olika sätt, via Drupals webbgränssnitt eller manuellt via FTP. Gör man det via Drupals webbgränssnitt, behöver man modulens länk-adress och den får man genom att högerklicka på den modulversionen man vill ha och väljer ”Kopiera länkadress”. Efter att länkadressen är kopierad, navigerar man till sin Drupal-modulsida, <http://example.com/admin/modules/>. Där man antingen laddar upp en nedladdad ZIP-fil av modulen eller anger länkadressen till modulen. För att lägga den kopierade länkadressen man nyss kopierat, högerklickar man igen och väljer ”Klistra in”. Efter man gjort det, kommer en installationsprocess att börja. När installationen är klar, måste modulen ännu aktiveras.

Bestämmer man sig för att installera modulen manuellt, laddar man ner modulens ZIP-fil och därefter öppnar en anslutning till webbplatsen med hjälp av en FTP-klient. När man anslutit sig till webbplatsen, navigera till `sites/all/modules`. I denna mapp finns alla installerade moduler listade och det är till denna mapp man skall lägga sin uppackade ZIP-fil. När modul-filen är lagd in i `modules`-mappen är installationen färdig. Modulen måste ännu aktiveras via Drupals grafiska gränssnitt.

## 2.3 INSTALLATION AV TEMA

Att installera ett nytt tema till Drupal fungerar på samma sätt som att installera moduler manuellt. När man har hittat ett tema från Drupals hemsida, laddar man ner `tar.gz`-filen eller ZIP-filen. Nästa steg är att packa upp mappen, öppna en FTP-klient, logga in till hemsidan och navigera till `/sites/all/themes`. Efter att man lagt den uppackade filen till `themes` mappen är nästa steg att gå och aktivera temat, detta görs under ”<http://example.com/admin/appearance>”.

När man är i skedet att välja ett nytt tema finns det vissa saker man bör tänka på. Den första saken är att välja rätt version av temat. Många teman har olika versioner man kan välja. För att ge ett exempel, Zen temat har nio olika versioner. De två som rekommenderas att välja är 7.x-5.6 (Drupal 7) och 6.x-2.1 (Drupal 6). Sedan finns det tre stycken teman som går under rubriken "Other releases". De fyra sista hittas under rubriken "Development releases" och det rekommenderas att inte välja någon av dessa. Många teman stöder både Drupal 6 och Drupal 7. Om t.ex. Drupal 7 används, går det inte att använda ett tema som är kompatibelt till Drupal 6 och tvärtom.

Före man laddar ned ett tema bör man kolla dess egenskaper och funktioner. Förutom att temat ser visuellt bra ut och strukturen av temat passar in på det man är efter, bör man även kolla om temat anpassar sig för smarttelefoner och tabletter. I och med att antalet smarttelefoner samt tabletter växer, är det mer eller mindre ett måste att ha ett tema som anpassar sig till olika skärmstorlekar.

## 2.4 VAL AV TEMA

Efter att vi bestämde oss för att skippa DrupalGap och göra applikationen med Drupal, blev det aktuellt att hitta ett tema, eftersom Drupals standardtema inte är mobil- eller tablettvänligt. Temats utseende hade ingen större betydelse eftersom det skulle slopas och istället skulle min design implementeras. "Mobile responsive theme" var temat jag valde. Förutom att temat var mobil- och tablettvänligt, som var den största orsaken, hade temat också en annan bra egenskap, nämligen dess enkla/simpla design. Eftersom mitt designutkast skulle implementeras till temat, fanns det onödig kod som inte behövdes. När man tar bort onödig kod, finns det risk att man i misstag tar bort kod som inte skall tas bort. Detta kan leda till att temat inte fungerar helt korrekt. Risken att man tar bort något man inte skall ta bort, minskar ju mindre kod man behöver radera.

Jag måste dock säga att före jag valde att köra med ett färdigt tema, försökte jag laga ett eget tema helt från början. Min tanke var att istället göra ett eget för att jag inte skulle behöva ta bort element från det färdiga temat, utan istället göra det själv. På detta sätt skulle det enbart ha funnits element jag kodat. Efter att jag försökt göra ett eget tema och misslyckats, tog jag beslutet att söka efter ett färdigt tema och implementera min design till den.

### 3 DEFINITIONER

Stödet för stilmallar, eller style sheets på engelska, var en av de viktigaste nyheterna i HTML 4.0. Ursprungligen utvecklades HTML för vetenskapliga rapporter, dokumentation till program, där utseendet inte spelade någon roll utan det var innehållet av dessa dokument som var det viktigaste. Efter att webben började användas till andra saker, ökade kravet på att kontrollera hur informationen publicerades på skärmarna. Olika knep hittades för att ändra på layouten på tabeller och bilder. Microsoft och Netscape tävlade även om vem som fick dokumenten snyggare på sin webbläsare, genom att hitta på egna tillägg till HTML. HTML:s framtid var osäker i ett skede, eftersom ett ökande antal av dokument fungerade endast på en viss webbläsare och i vissa fall, endast på en viss version av en webbläsare. Lösningen till detta problem är formatmallar. Problemet löses eftersom man skiljer på dokumentets innehåll och hur innehållet presenteras, genom att införa ett nytt språk för presentationsinformation. Med hjälp av formatmallar kan man formatera dokument mycket noggrannare än tidigare. Det är möjligt att ha samma formatmall för en hel webbplats och flera olika mallar för samma dokument, om man vill ha olika utseende på skärmen. (Staflin 2013, s. 109).

#### 3.1 OLIKA VERSIONER AV HTML OCH CSS

Det kommer ut nya versioner av HTML och CSS eftersom webben utvecklas hela tiden. Men eftersom HTML och CSS är såpass utvecklade kommer förändringarna mera sällan. W3C eller *World Wide Web Consortium*, är en massiv sammanslutning som har över 300 medlemsorganisationer. Tim Berners-Lee, grundaren till webben 1989, arbetar där fortsättningsvis. W3C skapar standarder för Internet och när dessa standarder är klara, kallar man dem för rekommendationer. (Staflin 2013, s. 16-17).

#### 3.2 CSS

Dagens mest använda sätt att skriva formatmallar kallas Cascading Style Sheets eller bättre känt med namnet CSS. I slutet av 1996 fastställdes CSS nivå 1. CSS nivå 2 kom i 1998 och den nivån innehöll stöd för nedladdningsbara teckensnitt och utplacering av element och tabeller. 2011 kom den senaste helt färdiga versionen, CSS 2.1. Denna version är en liten uppdatering av

version CSS 2 som innehöll lite rättningar och några nya funktioner som stöds av de flesta webbläsarna. Nuförtiden används CSS 3. (Staflin 2013, s. 110).

### 3.3 CSS-KODOPTIMERING

Det finns vissa principer man bör tänka på då man skriver CSS som hjälper att hålla koden kort och optimerad. Att optimera kod handlar inte bara om att försöka minimera storleken på filen. Det handlar också om att vara organiserad och effektiv. Att veta principerna om att skriva optimerad CSS-kod, hjälper dig inte endast att få optimerad CSS- kod utan även kommer det att göra dig till en bättre webb-designer.

Första sättet att optimera kod är att skriva CSS-koden i s.k. "shorthand" stil (Kod 5, nedan). Istället för att ha alla relaterade egenskaper på skilda rader som i Kod 4.

Kod 4. Ej optimerad CSS-kod

```
p { margin-top: 10px;
      margin-right: 20px;
      margin-bottom: 30px;
      margin-left: 40px; }
```

Kod 5. Shorthand CSS-kod

```
p { margin: 10px 20px 30px 40px; }
```

Ett annat sätt att optimera CSS-koden är att använda tomma utrymmet i koden smart. Mellanslag, tabbar och extra radbyten är viktigt för läsbarheten i CSS. Även hur litet ett mellanslag än är, räknas det till filstorleken. Varje radbyte, tab och mellanslag man kan radera, är som att ha ett tecken mindre. Även om detta är ett sätt att få filstorleken mindre, är det inte smart att optimera för mycket. Det är viktigt att det du skriver är läsbart och tydligt, hoppeligen även för andra. (White, 2008).

Figur 8, nedan, visar två olika sätt att använda tomrummet. Första delen har extremt mycket tomrum och nedre delen använder tomtrummet på ett betydligt bättre sätt. Jag har använt mig av få mellanslag för att hålla koden komprimerad och för att hålla filstorleken så liten som möjligt också.

```

body {
  line-height: 170%;

  font-size:14px;

  font-family: 'Open Sans', sans-serif;

  background-image: url(../images/backgroundimage.png);

  background-size:cover;

  position:absolute;

  background-attachment: fixed;
}
body {
  line-height: 170%;
  font-size:14px;
  font-family: 'Open Sans', sans-serif;
  background-image: url(../images/backgroundimage.png);
  background-size:cover;
  position:absolute;
  background-attachment: fixed;
}

```

*Figur 8. Användning av mycket mellanslag och användningen av lite mellanslag.*

CSS kommentering är ett bra kodoptimeringssätt. CSS kommentering är ett bra sätt att dela in långa CSS filer, men kommentarer som förklarar vad man gjort ökar filstorleken. Att dokumentera CSS koden med hjälp av kommenteringar är definitivt bättre än att inte kommentera alls, men att hålla kommenteringarna i ett skilt dokument håller koden mera optimerad. (White, 2008).

Det finns även en hel del bra applikationer som komprimerar och optimerar CSS koden för dig. Applikationerna fixar läsbara men webbläsarvänliga filer vars filstorlek är en bråkdel av den ursprungliga filstorleken. Applikationer som CSSTidy och YUI Compressor komprimerar mellanslag, känner igen och korrigerar CSS egenskaper som skriver över varandra, samt kollar applikationerna efter möjligheter att använda sig av ”shorthand”. Det är viktigt att ha ren och optimerad kod, inte bara för filstorleken utan också för underhåll och läsbarheten. Principerna jag räknat upp är inte bara för kodspråket CSS utan även för HTML, JavaScript och andra kodningspråk. (White, 2008).

Av dessa kodoptimeringsprinciper eller tekniker har jag använt mig av CSS-kommenteringar och så jag har försökt ta bort så mycket tomrum som möjligt, utan att ta ner på läsbarheten. Med hjälp av CSS-kommenteringarna har jag gjort kommenteringar i koden och namngett element för att veta vilka element de är och vilka ändringar jag gjort på dem. Eftersom style.css filen

innehåller över 2700 rader med kod har egna kommentarer varit till stor hjälp under projektets gång då jag kunnat söka efter mina kommenteringar med Dreamweavers sökfunktion och hittat till rätt plats snabbt.

## **4 DESIGNEN**

Som jag konstaterade tidigare var jag ansvarig för designen och layouten för denna applikation. Här presenteras en del saker och regler man bör tänka på då man gör designarbete, mina mål med designen och till sist presenteras mitt första designutkast.

### **4.1 REGLER FÖR DESIGN**

Marissa Mayer, som har arbetat på Google men är nu anställd hos Yahoo, säger att det finns tre regler som alla borde följas för att göra en applikation bättre. I boken "Marissa Mayer and the Fight to Save Yahoo!" som är skriven av Nicholas Carlson, avslöjar Mayer sina tre regler av en bra app design. Den första regeln Mayers använder sig av är "The Two Tap Rule". Att testa denna regel är simpel säger Mayers, när applikationen är öppnad, kan du göra allting med två klick. Om inte, redesigna applikationen. Till exempel The Yahoo Flickr applikationen, som Mayers har designat, kan användaren ta en bild, navigera album, sätta alarm, kolla olika grupper och mera, endast med två klick.

Under tiden Mayers arbetade på Google, sade hon till designpersonalen: räkna ett poäng för varje färg, font och fontstorlek man ser på en sida. Om antalet blir mera än fem, är det dags för att redesigna. Detta var regel nummer två.

Sedan kommer vi till nummer tre som heter "The 98 % Rule". Mayers anser att varje applikation borde vara gjord på det sättet som den kommer att användas 98 % av tiden. Mayers brukar använda en printer som exempel för att förklara denna regel. Med printern syftar hon på att det är möjligt att göra en massa saker men största delen av användarna vill bara kopiera en bild. Därför är detta gjort så enkelt som möjligt, det är bara att lägga bilden på glaset och trycka på en stor knapp. Mayers anser att varje applikation borde ha en sådan knapp. Flickr är ett bra exempel igen. Flickr är ett stort socialt nätverk omkring fotografi. Därmed har Flickr en stor kameraknapp på varje sida. (Fastcodesign, 2015).



## 4.2 ANVÄNDARVÄNLIGHET

Användarvänligheten är ett kvalitetsmått på en produkt eller tjänst. I och med att det är ett sätt att mäta kvaliteten på en produkt eller tjänst, kan användarvänligheten var låg eller hög. Ifall användarvänligheten är låg kan det vara p.g.a. att användaren inte hittar den hen söker eller att användaren inte vet hur den skall använda dataprogrammet eller applikationen. Detta resulterar ofta i att användaren får en dålig eller negativ bild av produkten, tjänsten eller även företaget. Bra saker att tänka på när man talar om användarvänlighet:

1. Första steget är att det skall gå fort och smärtfritt att installera applikationen. I installationskedet får användaren första intrycket till applikationen och detta skede vill man inte göra användaren besviken med att ha en rörig installation. Ifall installationen är stökig kommer användaren troligen söka upp någon annan applikation efter att installationen är klar.

2. Samma sak gäller med uppdateringen av applikationen. Uppdateringen bör också göras simpel och smärtfri. På detta sätt säkrar du att användarna har den senaste versionen av applikationen och därmed även de nyaste funktionerna.

3. GUI (Graphical User Interface) – Grafiska användargränssnittet är en annan sak man bör tänka på. Grafiska användargränssnittet bör vara bra genomtänkt och bra utförd. Navigeringen skall göra det lätt för användaren att använda applikationen.

4. Effektiveten. Förutom att applikationen skall fungera som den förväntas fungera, skall den även vara kodad på ett bra sätt och ha alla buggar fixade. Applikationen skall hjälpa användaren med att slutföra uppgiften så smidigt som möjligt. Som ett exempel, Microsoft bytte utseende på Office programmen och det resulterade i att användarna spenderade dyrbar tid på att hitta var alla egenskapen var placerade. Detta är ett exempel på ett dåligt designbeslut. (Techrepublic, 2010).

I och med att applikation inte ännu är ute på marknaden är det endast nummer tre som är aktuellt, d.v.s. det grafiska användargränssnittet/navigeringen. Mitt mål med navigeringen i applikationen har varit att försöka göra den simpel. Målet har varit att användarna skall veta hur de skall använda applikationen och hitta informationen de söker och gärna med så få steg som möjligt. Varje klick eller alternativ användaren tar, för den närmare det som användaren söker.

Ett sätt äta mäta användarvänligheten är att kolla hur många tryck eller klick det tar innan användaren hittar informationen den söker. Denna regel är egentligen en oskriven regel inom webdesign som heter three-click rule (tre-klicks regeln). Jag trodde tidigare att detta var en oskriven regel i webbdesign världen, men jag undersökte saken och kom fram till att detta är endast en myt.

Det har bevisats att antalet klick eller tryck inte gör att användaren lämnar webbplatsen eller applikationen ifall det krävs mera än tre klick eller tryck. Användarna blir inte gladare med få tryck. Det som har betydelse är hur lätt navigationen på webbplatsen/applikationen är. Användarna räknar inte antalet klick det tar tills de har hittat det de söker. Därmed har det alltså ingen betydelse ifall det tar fyra eller fem klick, så länge som navigationen är uppbyggd på ett bra sätt. (Uxmyths, 2014).

Som det konstaterades i början av detta stycke, är det möjligt att mäta användarvänligheten. Är du på väg att publicera en applikation, webbsida eller produkt, kan det vara bra att testa användarvänligheten. Ett sätt att få in värdefull information om användarvänligheten och om hur applikationen/produkten fungerar, är att göra ett användarbaserat test. Fördelar med detta är att det kan ge klara resultat på de viktigaste problemen. Nackdelarna med detta sätt att testa, är att det är tidskrävande, kostsamt för större användargrupper och att det kräver en prototyp som kan testas. Ett annat sätt att mäta användarvänligheten är att använda sig av expertbaserade användare. Fördelen med detta är att det går snabbt och det är även billigare än att använda sig av ett användarbaserat test. Nackdelarna med detta är att antalet egentliga fel kan överskattas och att p.g.a. de är experter kan detta påverka resultatet. (Dillon, 2001).

### **4.3 PLANERINGEN AV DESIGNEN**

Jag börjar med att säga att det var svårare än jag trodde att det skulle vara. Att ha som uppgift att göra en design från scratch, att försöka få designen att se modern ut och att försöka få applikationen användarvänlig, var inte lätt. Att designen skulle vara modern och up-to-date samt användarvänlig var två viktiga egenskaper jag ville åstadkomma.

Användarna tröttnar snabbt på en applikation om det är svårt att använda den. Därför var användarvänligheten en viktig del av applikationen. Layouten försökte jag planera och göra så att

den såg modern ut eftersom designen är en stor del av helheten. Användarvänligheten och designen går lite hand i hand kan man säga. Med en bra design och bra planerad navigering får användaren ett bra intryck av applikationen och fortsätter därmed att använda den.

Jag hade så gott som ingen erfarenhet av att designa något från tidigare. Detta var min första gång jag producerade en design till en applikation. Detta projekt har med andra ord varit en utmaning för mig när det gällde att skapa något visuellt. Som jag konstaterade tidigare, var min uppgift att producera en design till applikationen och sedan implementera den. Vi gjorde inte någon kravspecifikation angående designen med min beställare Elvstöm. Jag hade fria händer med designen och detta hämtade med sig frihet men också utmaningar. Enligt mig skulle det ha varit lättare att skapa något om jag hade fått en del direktionser att följa, men som sagt hade jag helt fria händer. Därmed blev uppgiften att producera designen mera krävande.

Min idé och plan var att hålla designen simpel och användarvänligheten skulle också beaktas som jag skrev tidigare. I och med att applikationen skulle hjälpa användaren att hitta en viss information, ville jag att denna process var så smidig som möjligt. Det skulle vara enkelt att använda applikationen.

En annan sak som var viktigt att ta i beaktande var att den design jag kom fram till, den designen skulle också implementeras till applikationen med hjälp av Drupal, Drupal 7-moduler och med kodspråket CSS. Jag försökte göra en design som var möjlig att få implementerad till applikationen samt göra en design som både min beställare och teammedlemmarna var nöjda med.

#### **4.4 DESIGNUTKASTET**

Planen med att göra ett designutkast är att man först gör en prototyp på hur något kommer att se ut, före man börjar arbeta på det. På detta sätt sparar man eventuellt tid och pengar, då man inte behöver göra något på nytt ifall någon inte är nöjd med slutresultatet.

Jag gjorde ett designutkast på framsidan till applikationen med hjälp av Adobe Photoshop och designen som jag kom fram till ser ni nedan (*Figur 1*):



*Figur 1. Första designutkastet för framsidan.*

Bilden (*Figur 1*) är mitt första designutkast på designen till applikationens framsida. Tanken med framsidan är att användaren direkt får tillgång till innehållet och kan genast börja söka efter information. Ikonerna hämtar mera applikationskänsla och det ser visuellt bra ut också. Bakgrundsbilden anser jag att passar bra som bakgrundsbild, eftersom Pargas är en skärgårdsstad som är mycket vacker på sommaren.

Bilden (*Figur 2*) nedan, är också en del av mitt första designutkast. Bilden visar hur jag har tänkt att det skulle se ut efter att man trycker på en av rubrikerna på framsidan. Bilden nedanför visar underkategorierna för ”Resa”. I designutkastet finns det Taxi, Tåg och Färja. Tanken med designen och navigeringen är att man snabbt kan navigera till det man söker. I och med att alternativen är direkt framme på framsidan kan användaren med få steg navigera till informationen hen söker. Efter att användaren navigerat till underkategorisidan, väljer hen en av underrubrikerna och sedan syns informationen i den vita rutan. Den valda underrubriksikonen ändrar färg för att visa vilken underrubrik som är vald.



*Figur 2. Första designkastet på underkategori-sidan.*

## 5 IMPLEMENTERING AV DESIGNEN

Här presenteras hur framsidan är implementerad till applikationen med hjälp av Drupal efter att designkastet presenterades och godkändes av min beställare. Undermenyn, designen till undermenyn och hur undermeny sidan är konfigurerad med hjälp av kodspråket CSS behandlas också. Navigeringen mellan framsidan och undermenyn, bakåtknappen, hemknappen och till slut presenteras Mobile sliding menu modulen.

### 5.1 DESIGNEN GODKÄNDES

Efter att jag fick designkastet klart, presenterade jag den till vår beställare Elvström, som godkände den. Nästa steg var att ha ett möte med de andra teammedlemmarna och presentera designkastet till dem.

Att få designkastet färdigt var en stor milstolpe för projektet och för mig. Även om designkastet endast var två bilder, förklarade de hur navigeringen skulle vara i applikationen och även hurdant användargränssnitt applikationen skulle ha.

## 5.2 FRAMSIDAN

I och med att vi använde oss av CMS systemet (Content Management System) Drupal, skulle implementeringen göras med Drupal för att underlätta underhållningen och för att underlätta eventuella ändringar i framtiden. Mitt första mål var att få framsidan klar och sedan arbeta mig vidare därifrån.

Framsidan innehåller en meny och att göra en meny i Drupal får man gjord på en kort stund. Men att ha möjligheten att kunna konfigurera utseendet på menylänkarna skilt för sig, är en utmaning.

Första steget för att göra menyn på framsidan är att göra en meny i Drupal. Detta gör man under admin/structure/menu. Där är det möjligt att göra nya menyer samt konfigurera dem. Under länken "list links" ser man alla länkar som finns i menyn. Där har man möjlighet att lägga till/ta bort, aktivera, avaktivera, översätta eller konfigurera länkarna skilt. För att konfigurera varje länk skilt väljer man "edit". Under "edit" har man möjlighet att ändra på vissa grundläggande saker som t.ex. namnet på menylänken, pathen (sökvägen) till menylänken och till vilken meny denna länk skall kopplas ihop med. Eftersom "Menu Icon" modulen är installerad finns möjligheten att välja en ikon till länken.

Ändringar i utseendet görs i style.css(CSS-filen). För att göra ändringar på rätt element krävs det att man vet elementets id eller "class". Man kan säga att id och class är elementens namn och för att göra ändringar skall man ange detta namn i style.css, när man gör konfigureringar. Om "class" används finns det en möjlighet att andra element påverkas också, eftersom i många fall kan andra element vara kopplade till samma "class". Detta kan leda till att andra element påverkas också av konfigureringarna som görs. Detta hände mig några gånger när jag arbetare på att implementera designen. Jag gjorde några ändringar på vissa saker med hjälp av classes tills jag var nöjd med resultatet. Efter en stund märkte jag att andra element också hade påverkats av konfigureringarna även om jag inte hade gjort några ändringar på dem. Orsaken till att de andra elementen hade ändrats, var p.g.a. att de var kopplade till samma class. Det betydde att alla element med det class-namnet påverkades och det ville jag absolut inte.

För att undvika detta problem, lönar det sig att kolla noga hur HTML koden är uppbyggd och det valda elementets egenskaper med hjälp av Google Developertool:s inspektörverktyget. Kolla exakt i vilken div (del) elementet finns och använd sig av den classen. Använder man class från fel div (del) av applikationen, resulterar det i ändringar på andra saker också. Det säkraste sättet att konfigurera ett element är att använda id, eftersom id är alltid unikt eller att använda unika classes och därmed vara säker på att ändringarna endast påverkar det element som ändringarna görs på.

Före ”Menu icon” modulen var installerad, hade menylänkarna gemensamma classes. När en ändring gjordes på t.ex. bakgrundsfärgen, ändrades allas bakgrundsfärg. Efter att ”Menu icon” installerades och menylänkarnas ikoner var insatta, fick alla menylänkar egna unika ”classes”. I och med att menylänkarna hade egna unika classes, var det möjligt att konfigurera menylänkarnas utseende skilt.

Min tanke med framsidan är att användaren snabbt och enkelt hittar informationen och för varje tryck (steg) som användaren tar, går hen närmare informationen. Framsidan går rätt på saken kan man säga. Framsidan består av en meny som visar kategorierna. Menyn radar upp alla kategorier och varje kategori har en egen ikon. För att få ikonerna att se bättre ut och att få dem att stå ut mera, är de inne i en vit cirkel.

Orsaken till att jag har valt att lägga till dessa ikoner, är för att jag har märkt att det är populärt med ikoner. Jag har satt dem på webbsidor samt applikationer och jag anser att det ser visuellt bra ut också. Förutom att jag lagt märke till att ikoner är populärt, har jag också lagt märke till att färgen blå också är populär just nu, t.ex. Facebook och Instagram använder båda färgen blå. Måste dock säga att färgvalet blå-grå, på framsidans meny, är bara tillfälligt. Vår plan är att få designen implementerad och funktionerna klara. Efter det kollar vi närmare på färgerna vi använder och eventuella ändringar görs sedan.

Menyn på framsidan har en opacity (genomskinlighet) på 0.6, det vill säga 60 %. På detta sätt ser man genom menyn.

### 5.3 SIDA 2 – UNDERMENYERNA

Efter att användaren väljer en av kategorierna, flyttas användaren till en ny sida med underkategorierna. Jag har lärt mig att när man designar något, om det är en applikation eller något annat, skall man vara konsekvent. Därför finns dessa ikoner också med på undermeny-sidan. Vita cirkeln är ett måste när bakgrundsfärgen är grå. I och med att cirkeln är där gör det att man ser ikonerna betydligt bättre och så ser det visuellt bra ut också enligt min åsikt. Färgen på ikonerna är svart, som på ikonerna på framsidan. Förutom ikonerna, är det ett måste att ha en text till ikonerna. Även om bilderna ger en bild om vad rubriken handlar om, är det inte alltid klart och tydligt, därför är texten nödvändig. När texten finns där blir det lättare för användaren att välja rätt. Färgen på texten är också vit, jag har två orsaker till det. Orsak nummer ett, eftersom att bakgrundsfärgen är grå och ganska mörk, måste texten vara ljus för att synas bra. Orsak nummer två är ikonens vita bakgrundsfärg. Det skulle se aningen konstigt ut enligt mig att ha en annan färg på texten än vit. Att välja färgen vit är nästan en självklarhet.

Texterna är placerade under ikonerna. Den mest naturliga platsen enligt mig. Som man kan se på *Figur 3*, ser man igenom menyn. Menyns grå bakgrundsfärg har en genomskinlighet på 0.8 d.v.s. 80 %. På detta sätt ser man igenom den gråa färgen och man kan fortfarande se bakgrundsbilden. Jag är av den åsikten, att ha en liten genomskinlighet ser bättre ut än att ha ingen genomskinlighet alls även på denna meny.





Figur 3. Layouten på underkategorierna.

Uppställningen av ikonerna gjorde jag enligt bilden ovan (Figur 3), två ikoner per rad. Anser att två ikoner per rad är mest passande. Då blir det tillräckligt med utrymme mellan ikonerna. Anser också att det skulle bli lite trångt med tre ikoner per rad. Det skulle uppstå problem med utrymme med att få in tre ikoner på samma rad eftersom vissa underkategorier har långa namn. Jag vill undvika att ikonerna och texterna blir för nära varandra. Det skulle inte se visuellt bra ut eller vara praktisk för den delen. I och med att det finns mera utrymme mellan underrubrikerna minskar risken att välja fel underkategori. Gjorde dock ett undantag med rubriken ”Travel”. Där är alla underrubriker bredvid varandra eftersom antalet underrubriker endast är tre och alla har korta namn; ”Taxi”, ”Ferry”, ”Bus”.

#### 5.4 DESIGNEN TILL UNDERRUBRIKERNA

Underrubrikerna är placerade och konfigurerade med kodspråket CSS. Kodtexten nedan (Kod 1), visar inställningarna (koden) bakom hur underrubrikerna till ”Eat” (Figur 3) är kodad. Har nämnt tidigare att för att kunna göra konfigureringar på den sak vill konfigurera, måste man veta sakens id eller ha elementets class som måste vara unik. I exemplet nedan (Kod 1) har ”Eat” undermenyn id ”block-menu-block-5”. I css-kodspråket sätts #-tecknet före id. I detta fall blir det ”#block-menu-block-5”.

Förklarar de viktigaste delarna av Kod 1 nedan. Under ”#block-menu-block-5” visar egenskaperna för hela blocket. Position: fixed betyder att objektet är positionerat relativt till skärmen d.v.s. den hålls på samma plats även om man scrollar. Width och height anger yttre gränserna för blocket. Background-color är bakgrundsfärgen och 0.8 betyder att man kan se igenom bakgrundsfärgen (genomskinligheten). Top: 10 % betyder att blockets skjuts nedåt med 10 % från toppen.

#block-menu-block-5 ul är egenskaperna för punktlistningen medan #block-menu-block-5 ul li är egenskaperna för de enskilda punkterna på menyn. Width och height är bredden och höjden på ikonerna. I och med att bredden på ikonerna är 57px, ryms det inte mera än två på en rad. Minskar man på pixlarna, minskar bredden på ikonerna och detta leder till att det ryms tre på en rad, istället för två. Höjden som är 56px, bestämmer mellanrummet lodrätt. Display: flex är egenskap som möjliggör att man kan ändra mellanrummet på tabellens innehåll.

Kod 1. Koderna bakom underrubrikerna till Eat

```
#block-menu-block-5 {
  position: fixed;
  width: 100%;
  height: 62%;
  background-color: rgba(50, 57, 74, 0.8);
  top: 10%; }

#block-menu-block-5 ul {
  padding: 0;
  margin-top: 10%;
  list-style-type: none; }
#block-menu-block-5 ul li {
  width: 57px;
  height: 56px;
  float: left;
  display: flex; }
```

## 5.5 NAVIGERINGEN MELLAN FRAMSIDAN OCH UNDERMENYERNA

När användaren trycker på en av kategorierna på framsidan öppnas undermenyerna på en ny sida eller vy. Det kan vara att det finns andra sätt att göra detta på men ett sätt är att använda sig av en modul som heter ”Menu Block”.

För att få denna navigering att fungera är första steget att göra undermenyerna först. I Drupal görs detta under admin/structure/menu och add menu. Efter att undermenyn är skapad och sparad

har man möjlighet till att lägga till länkar. När länkarna skapas och konfigureras är de viktigaste inställningarna: namnge länken, sökvägen och sedan välja en ikon åt länken. När alla länkar är gjorda är nästa skede att navigera till admin/structure/block/ och därefter välja "add menu block".

När ett nytt menu block skapas, skall några saker konfigureras. Nummer ett, ange namnet på blocket som i detta fall blir <none> d.v.s. blankt eller inget, eftersom namnet på blocket skall inte synas på undermenysidan. Nummer två, är att välja till vilken undermeny blocket skall kopplas till. Under "Region Settings", bestäms hur blocket skall visas i olika teman och var på temat blocket syns. I detta fall valdes "Content" (innehåll). Sista konfigurationen är att säga när meny blocket skall synas. Detta görs under "Visibility settings". På fliken "Pages" kan man välja "All pages except those listed" eller "Only on listed pages". För att ge ett exempel, Eat undermeny blocket skall endast synas då man trycker på Eat på framsidan. Därmed är "Only on listed pages" vald och i fältet finns eat, som är vägen (pathen) till framsidans meny för kategori Eat. För att göra det enklare att förstå. Man kopplar det nya blocket till en meny och sedan bestämmer man på vilken sida menyn skall synas på. Sedan är det bara att göra samma procedur för resten av undermenyerna. Koppla ihop nya blocken till menyerna och sedan säga åt dem var de skall synas.

## 5.6 BAKÅTKNAPPEN OCH HEMKNAPPEN

Kod 2. Bakåtknappens kod.

```
<input id="back" type="button" value="" onclick="window.history.back()" />
```

Koden från Kod 2, är från html.tpl.php filen. Det är koden ovan som gör att man går ett steg tillbaka. Som man kan se har denna bakåtknapp id "back", typ av element är "button" och funktionen är "window.history.back()". För att få denna bakåt knapp modifierad visuellt och placerad på rätt plats, är ändringar i css.style-filen nödvändiga. För att få de ändringar man vill göra åt ett visst element, bör man ha ett id eller en unik klass som jag har konstaterat tidigare, i detta fall används id "back". Då man vill göra ändringar på ett objekt som har ett id, krävs det att man lägger märket # framför id. I detta fall har knappen id "back" och därmed blir det #back i style.css för att få detta att fungera. CSS-koden till bakåtknappen finns nedan (Kod 3.)

### Kod 3. CSS-koden till bakåtknappen

```
#back {
  position: fixed;
  top: 5px;
  left: 15px;
  background: url(../images/back.png) no-repeat;
  width: 39px;
  height: 39px;}
```

Position, som namnet säger handlar det om positioneringen av objektet. I detta fall är den inställd på fixed. Objektet är positionerad relativt till skärmen d.v.s. den hålls på samma plats även om man scrollar. Top: 5px, bestämmer positionen vertikalt, 5px från toppen nedåt. Left: 15px, bestämmer position horisontellt, 15px in från vänstra kanten. Background, som namnet säger handlar det om bakgrundsbilden och url:n (vägen) till bilden samt att den inte skall upprepas. Width och height bestämmer bredden och höjden på knappen.

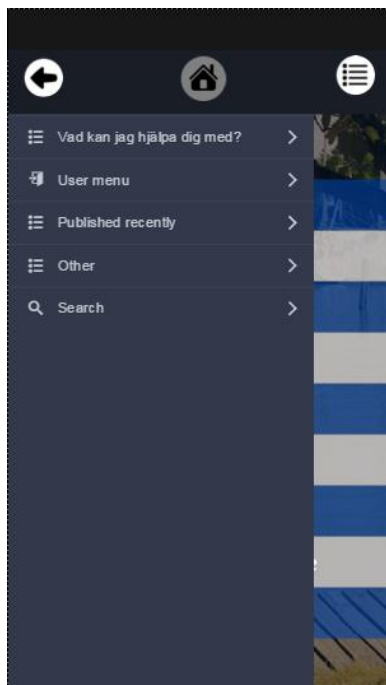
Drupal har färdigt en hemknapp. För att byta bort Drupals egna bild, navigera till `/admin/appearance/settings` och sedan väljer du temat som du använder. Där har man möjlighet att välja en egen bild som hemknapp.

## 5.7 MOBILE SLIDING MENU

I och med att vi använder Drupal som bas, blir vår applikation en webbapplikation. För att få mera känsla att de skulle vara en applikation och inte en webbapplikation installerades ”Mobile sliding menu” modulen. Med ”Mobile sliding menu” har man möjlighet att få app-liknande drag (glid) menyer. Förutom att vi får mera app-känsla, får vi även några problem lösta på samma gång. Saker som vi inte vet vart vi skall placera eller hur vi skulle få implementerade på ett bra och smidigt sätt får vi lösta med ”Mobile sliding menu”. Vi kom överens om vilka saker som skall finnas i slide-menyn eftersom antalet element man kan ha i menyn är begränsat till fem.

Första rubriken är menyn från framsidan och den har rubriken ”Vad kan jag hjälpa dig med?” eftersom applikationen är till för att hjälpa användaren att hitta information. Andra rubriken i menyn är ”User menu”, det är platsen man loggar in till applikationen. Vet inte vart eller hur vi skulle ha fått denna funktion placerad på ett bra sätt, om vi inte skulle ha stött på ”Mobile sliding menu” modulen. Under Published recently kan man kolla på nyligen publicerat material. Under Other menyn har man tillgång till lista på företag, evenemang, närliggande platser, aktiviteter

samt nyheter. Till sist finns sökfunktionen. Som man kan se på Figur 4, nedan, har ”Vad vill du göra?” från designutkastet (Figur 1) försvunnit. Utrymmet har behövts för de nya funktionerna.



Figur 4. Mobile sliding menu.

I och med att jag placerade bakåtknappen på den naturligaste platsen, d.v.s. till uppe till vänster, hemknappen uppe i mitten, fanns det endast högra hörnet kvar för sidomenyn. Utseendet för sidomenyknappen har som vanligt en vit cirkel som bakgrund och själva ikonerna skall föreställa en lista. Utseendet för mobile sliding menu var från första början bra, eller den passade åt oss. Därför har inga större visuella ändringar gjorts. Fontinställningarna för ”Vad kan jag hjälpa dig med?” konfigurerades för att få texten på en rad.

I inställningarna för denna modul har man möjligheten till att välja ett objekt som öppnar och stänger sidomenyn. En ny knapp gjordes för detta med hjälp av html och css. För att Clickopen skall fungera behöver den veta id för objektet. Knappen som gjordes har id ”#hide”. Nästa steg är att lägga in id i Clickopen modulens inställningar.

▼ **CLICKOPEN (ADD-ON)**

**open**

Yes ▼

Whether or not the menu should open when clicking on the selector.

**selector**

#hide

Clicks the selector to open the mmenu. e.g. #logo or a[id=logo]

Figur 5. Öppning och stängning av sidomenyn.

## 6 PROGRAM SOM ANVÄNTS

Under projektets gång har jag haft behov av en del program för att få uppgifterna gjorda. Det är en del steg mellan att börja designa något och att slutligen ha designen implementerad till applikationen.

### 6.1 ADOBE DREAMWEAVER

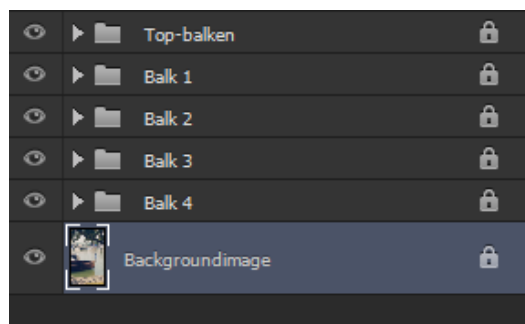
Med Adobe Dreamweaver programmet är det möjligt att designa och utveckla responsiva webbsidor med olika hjälpmedel. Med programmet kan man också förhandsgranska sina framsteg i realtid.

Jag har använt mig av Dreamweaver när jag arbetat på att implementera designen med hjälp av kodspråket CSS, när jag gjort ändringar på t.ex. bakgrundsfärger, fonter, fontstorlekar, ändringar på sakers positioner, höjder och bredder för att nämna några.

### 6.2 ADOBE PHOTOSHOP

Bildredigeringsprogrammet Photoshop anses vara ett av de bästa programmen när det kommer till att redigera bilder. Photoshop har miljoner användare varav största delen är designers, grafiker och fotografer. Photoshop har också en stor mängd användare som använder Photoshop i

hobbysyfte. En av Photoshops stora fördelar är möjligheten att arbeta med så kallade lager. Med hjälp av dessa lager är det möjligt att arbeta på ett visst lager, utan att det påverkar andra element. Jag gjorde designutkastet och resten av designarbetet i Photoshop eftersom jag har erfarenhet av Photoshop från tidigare och att arbeta med lagerfunktionen gjorde designarbetet enklare.



Figur 6. Mappstrukturen i min designutkastfil.

Tack vare lagerfunktionen är det möjligt att skilja åt alla element från varandra och editera dem enskilt. Förutom lagerfunktionen, går det att gruppera dessa lager hur man vill, t.ex. designutkastet grupperade jag bakgrundsbilden ”Vad vill du göra” texten och dess bakgrundsfärg, alla fyra menyknappar med alla deras element i skilda mappar. Att sätta dessa lager i olika mappar underlättar arbetet, speciellt då man har många lager att arbeta med. Efter man grupperat lagren i mappar, namnger man mapparna ännu för att underlätta arbetande ännu mera. När man har lagren i skilda mappar, går det även lätt att gömma, samt låsa de lager man inte vill att ska synas respektive ändras.

### 6.3 WINSCP

WinSCP är ett open-source filöverföringsverktyg för Windows. Eftersom Drupal installerades på en server på distans, var användningen av WinSCP ett måste för att få tillgång till olika filer som t.ex. style.css (CSS-filen). Jag använde WinSCP också då jag lade in nytt material som t.ex. bilderna för ikonerna. Hade också användning av WinSCP då jag arbetade på implementeringen av designen eftersom jag behövde tillgång till andra filer också.

## 6.4 GOOGLE CHROME DEVELOPER TOOL

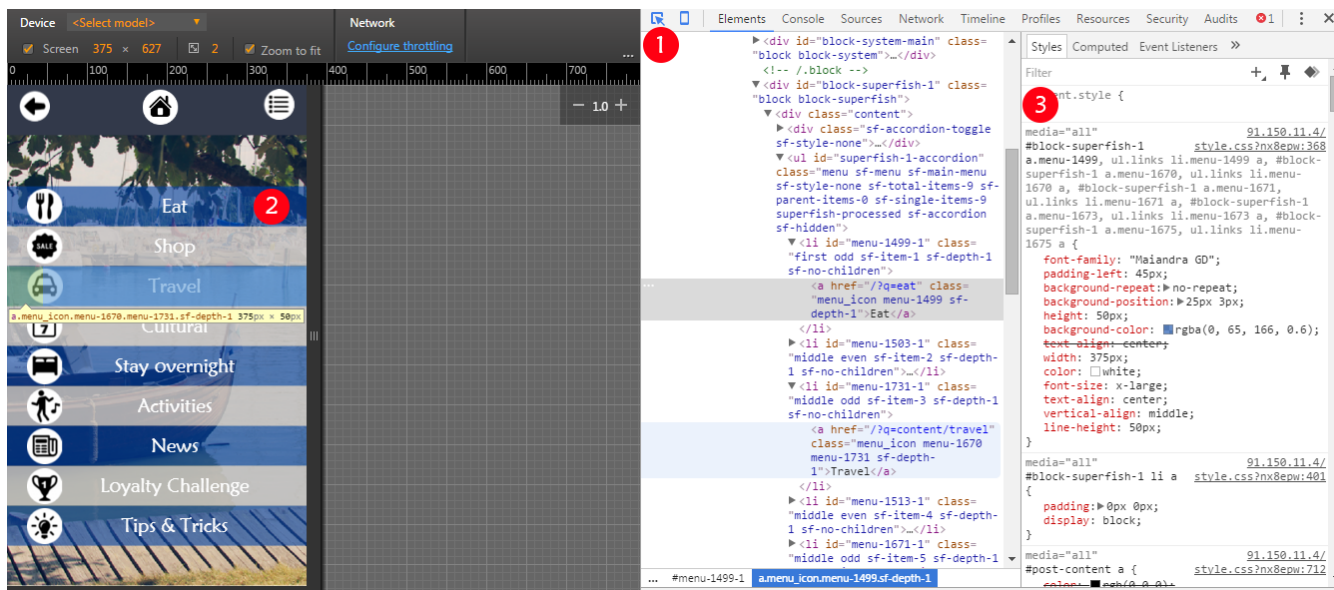
Googles Chrome Developer Tool (DevTool) är det mest använda programmet jag använt under projektets gång tillsammans med Adobes Dreamweaver. Trycker man på F12 när man öppnat Google Chrome, öppnas en ny vy, nämligen DevTool layouten. I detta läge kan man kolla hur t.ex. en hemsida eller webbapplikation ser ut i en tablett eller mobil. DevToolen är ett ypperligt verktyg för det arbete jag gjort. Gör visuella ändringar direkt i webbläsaren tills resultatet är bra, kopiera konfigureringarna från webbläsaren och sätt koden in i style.css filen.

Med DevToolen kan man också kolla strukturen på html koden och med hjälp av inspektör verktyget kan man kolla på vad enskilda element eller delar av webbsidan/applikationen har för class eller id. Om man inte har gjort ändringar på ett element från tidigare är det ibland så att elementet inte har ett eget id, utan endast en gemensam class. För att kunna göra ändringar på bara ett element krävs det alltså att man vet elementets id eller dess class som måste vara unik. Annars påverkar ändringarna även andra element. Har man gjort ändringar på ett element från tidigare visar inspektörverktyget all relevant kod, i vilket dokument koden finns och på vilken rad den relevanta koden finns på.

Med hjälp av DevTool och inspektörverktyget är det också möjligt att göra ändringar direkt i webbläsaren som sedan syns direkt i webbläsaren, som jag konstaterade. På detta sätt behöver man inte göra ändringar i ett kodediteringsprogram, utan man kan göra alla nödvändiga ändringar med DevTool i webbläsaren och när man är nöjd med resultatet, kopierar man koden från webbläsaren och klistrar in det i style.css-filen. På detta sätt slipper man processen hoppa mellan olika program.

Om man vill ändra på en sak, väljer man inspektörverktyget (Figur 7, nr 1) och klickar på elementet man vill ändra på (Figur 7, nr 2). Då visar verktyget koden och annan relevant information (Figur 7, nr 3). I detta fall har jag tryckt på Eat och då visas HTML och CSS-kod som hör till knapparna Eat, Travel, Stay overnight, News och Tips & Tricks eftersom de är sammankopplade tillsammans för att spara på kod.





Figur 7. Användningen av inspektörverkyget.

## 7 SAMMANFATTNING

I ett tidigt skede efter att jag tog emot uppdraget att göra en applikation åt Pargas stad, tillsammans med mina två studiekollegor, kom det fram att applikationen skulle bli krävande med all dess information och funktioner. Under utvecklingskedet har jag bekantat mig med DrupalGap, gjort en design till applikationen och med hjälp av Drupal, Drupal-moduler samt kodspråket CSS implementerat designen till webbapplikationen.

I början av sommaren 2015 satte vi igång med att börja arbeta ihop applikationen och då var planen att använda DrupalGap. Men vi beslöt dock att slopa DrupalGap tidigt p.g.a. att vi inte gjorde tillräckligt många framsteg för att kunna fortsätta med det. Efter det beslöt vi, med beställarens godkännande, att vi använder Drupal som bas för att göra en webbapplikation.

I slutet av sommaren blev applikationens första och även sista designutkast färdigt. Designutkastet presenterades för beställaren samt teammedlemmarna och efter mötet godkändes designen. Efter att designen godkändes, var nästa skede att börja fundera på hur designen fås implementerad till webbapplikationen med hjälp av Drupal och dess moduler. Under webbapplikationens utvecklande har nya idéer presenterats med jämna mellanrum. Detta har varit utmanande och vissa gånger orsakat problem om hur idéerna skall implementeras till

webbapplikationen. De problem och utmaningar jag stött på under projektets gång, har jag fått hjälp med av mina teammedlemmar och beställaren. Detta har varit trevligt att veta att jag har kunnat fråga hjälp samt råd av mina teammedlemmar vid behov.

Designutkastet gjordes i Photoshop och efter att det godkändes blev det klart vilken design webb-applikation skulle ha. Implementeringen av designen var utmanande, vissa gånger problematisk p.g.a. att designen skulle implementeras i Drupal och inte med egen kod. Moduler har använts för att hämta funktioner som t.ex. navigationen mellan framsidan och underkategorierna samt sidomenyn. En essentiell del av implementeringen av designen var möjligheten att kunna göra visuella ändringar på element individuellt. Detta löstes med hjälp av en modul. Det som användaren ser d.v.s. användargränssnittet, är gjort med kodspråket CSS. Från att göra designutkastet till att få designen till webb-applikationen har jag använt mig av ett antal olika program. De två mest använda program har varit Google Chromes DeveloperTools inspektörverktyg och Adobes Dreamweaver. Inspektörverktyget har använts för att få reda på hur koden var uppbyggd och för att få reda på olika elements id eller class för att kunna göra visuella ändringar. De visuella konfigureringarna i style.css filen samt andra relevanta filer har gjorts i Dreamweaver.

Applikationens uppgift är att förbättra informationsflödet mellan privatpersoner, företag och turister i Pargas. Resultatet av förbättrat informationsflöde skall leda till att människor gör köpen hos de lokala företagen och inte hos de stora butikskedjorna utanför Pargas. Applikationen hjälper användaren att hitta information och tjänster som Pargas har att erbjuda.

Slutprodukten av examensarbetet är en applikation som är lätt att använda och navigera i, så att användaren hittar informationen hen söker. Designmässigt har slutprodukten en design som är tilltalande och up-to-date. Resultatet av designen och navigeringen i applikationen skall göra att användaren fortsätter att använda applikationen.

## 7.1 RESULTAT OCH REFLEKTIONER

Syftet med mitt examensarbete var att göra designen och navigeringen till webbapplikationen. Designen är simpel men den fungerar bra tycker jag. Designen har jag försökt få up-to-date med att kolla vad som är populärt just nu. Därmed ikonerna samt användningen av färgen blå. Som jag har konstaterat tidigare har min plan varit att hålla designen och navigeringen simpel, så att användarna inte skall behöva fundera på hur de skall använda applikationen utan istället hitta informationen snabbt och smidigt. Framsidan innehåller en meny eller olika kategorier, och efter att användaren gjort ett val, visas underrubrikerna. Kan inte bli mycket simplare än det och som jag konstaterade tidigare, måste jag också tänka på att den design jag producerar, skall också vara möjlig att få implementerad med Drupal. Med andra ord är jag av den åsikten att navigeringen och designen fyller kraven på vad jag ville åstadkomma från första början.

Detta har varit ett roligt och spännande men samtidigt ett utmanande examensarbete. Speciellt uppgiften att producera designen själv från scratch var utmanande, även om designen är simpel. Har inte tidigare gjort denna typ av designarbete tidigare, så det var en intressant och lärorik uppgift. Men även att fundera ut lösningarna till att få designen implementerad till webbapplikationen med Drupal och inte med egen kod, var stundvis jobbigt men desto roligare när lösningen hittades.

Det är alltid lätt att vara efterklok, men som det kom fram tidigare, hade vi ingen kravspecifikation eller någon större plan på hur applikationen kommer bli. Vi kom på nya idéer, hade möten och bestämde hur vi skall göra saker efter hand. I ett skede kom vi på en funktion som var ett måste att ha med i applikationen, men som inte gick att verkställa med Drupal. Med detta problem, och beslutet om att använda Drupal som bas, kom vi fram till att vår applikation inte kommer att godkännas till App storen eller Google Play, som var vår avsikt. Vi borde ha gjort en plan på vilka funktioner och andra saker vi vill ha till applikationen och då kanske kommit fram till slutsatsen, att detta inte går att implementera med Drupal. Istället skulle vi ha kodat applikationen från scratch och därmed sett till att vi får alla funktioner gjorda samt att app-butikerna godkänner applikationen. Så med andra ord, borde vi ha satt mera tid på planeringen av applikationen och eventuellt kommit fram till resultatet att koda applikationen från scratch.

## 7.2 VIDAREUTVECKLING

Som konstaterades i föregående kapitel, hade vi en idé som inte gick att implementera med Drupal. För att få funktionen implementerad, skulle det kräva att koda applikationen från början. Om man inte använder något publiceringsverktyg som bas, som t.ex. Drupal, försvinner begräsningarna med vad man kan göra. Även om vi skulle ha fått alla funktioner implementerade till applikationen med Drupal, skulle Google Play och App store inte godkännt vår applikation eftersom vi använde Drupal som bas. För att få applikationen godkänd till app-butikerna krävs det att applikationen kodas om från scratch.

I och med att vi stötte på vårt problem mot slutet av projektet, har en stor del av arbetet gjorts redan och detta kunde detta utnyttjas på något vis anser jag. Ett sätt kunde vara att slutföra applikationen med funktioner som går att implementera med Drupal, slutföra designen, för att sedan prova applikationen i Pargas. Sedan samla in feedback av testningen på vad de har för åsikter om applikationen och fortsätta utvecklingsarbetet utgående från insamlad feedback. Ett annat sätt som vår version kunde utnyttjas till är att om någon annan gör en webbapplikation med liknande funktioner, navigering eller design som vår applikation innehåller, kan hen kolla hur sakerna är gjorda och därmed spara tid.

## Källförteckning

Dillon, A. (2001) Usability evaluation. In W. Karwowski (ed.) *Encyclopedia of Human Factors and Ergonomics*, London: Taylor and Francis.

<https://www.ischool.utexas.edu/~adillon/BookChapters/Dillon-evaluation%20of%20software%20usab.htm>

(hämtat: 15.1.2016).

Drupal (u.å.). *Our history*.

<https://www.drupal.org/about/history> (hämtat: 1.2.2016).

DrupalGap (u.å.). *DrupalGap - Mobile Apps and Web Apps for Drupal*.

<http://drupalgap.org/> (hämtat: 15.01.2016).

Duncker & Rantanen (2015). *En webbapplikation för Pargas stad – Utveckling av funktionerna*.

[http://theseus.fi/bitstream/handle/10024/100583/Benjamin\\_Duncker\\_o\\_Cedric\\_Rantanen\\_En\\_webbapplikation\\_for\\_Pargas\\_stad\\_Utveckling\\_av\\_funktionerna.pdf?sequence=1](http://theseus.fi/bitstream/handle/10024/100583/Benjamin_Duncker_o_Cedric_Rantanen_En_webbapplikation_for_Pargas_stad_Utveckling_av_funktionerna.pdf?sequence=1)

(hämtat: 15.2.2016).

Fastcodesign (2015). *3 Rules Of App Design, According To Yahoo's Marissa Mayer*.

[www.fastcodesign.com/3041374/3-rules-of-app-design-according-to-yahoos-marissa-mayer](http://www.fastcodesign.com/3041374/3-rules-of-app-design-according-to-yahoos-marissa-mayer)

(hämtat: 18.12.2015).

Pargas (u.å.). *Allmän info om staden*.

[http://www.pargas.fi/web/kommuninfo/info\\_om\\_staden/sv\\_SE/info/](http://www.pargas.fi/web/kommuninfo/info_om_staden/sv_SE/info/) (hämtat: 27.01.2016).

Staflin, R. (2013). *HTML- och CSS-boken*. (7. uppl.) Sundbyberg: Pagina Förlag AB.

Techrepublic (2010). *10 things that make software user-friendly*.

<http://www.techrepublic.com/blog/10-things/10-things-that-make-software-user-friendly/>

(hämtat: 19.12.2015).

Uxmyths (2014). *Myth #2: All pages should be accessible in 3 clicks*.

<http://uxmyths.com/post/654026581/myth-all-pages-should-be-accessible-in-3-clicks>

(hämtat: 18.1.2016).

White, T (2008). *7 Principles Of Clean And Optimized CSS Code*.

<https://www.smashingmagazine.com/2008/08/7-principles-of-clean-and-optimized-css-code/>

(Hämtat: 22.3.2016).