

Mobile-first eller Desktop-first, en studie om utvecklingslösningar för den responsiva webben

Eduardo Castaneda

22 juli 2013

Sammanfattning

Abstract goes here... Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam sollicitudin varius libero ac consectetur. Nullam ornare, massa et sagittis consectetur, neque mi scelerisque arcu, in fringilla lectus risus non arcu. Suspendisse vestibulum tellus id mauris lacinia non hendrerit nibh tempor. Proin tempor interdum justo et elementum. Ut ultricies adipiscing ipsum et pharetra. Vestibulum pretium luctus est, quis egestas augue luctus et. Praesent volutpat pharetra lectus vitae elementum.

Integer fringilla ligula eu sem semper tincidunt. Nullam mi lacus, blandit non sollicitudin eget, tempor eu ante. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Morbi ornare sem et purus consequat ac adipiscing nunc tincidunt. Curabitur nisi ante, ornare vel adipiscing et, scelerisque vitae erat. Etiam blandit egestas magna, quis dapibus nulla euismod quis. Sed interdum interdum malesuada. Suspendisse lacinia imperdiet laoreet. Maecenas ullamcorper laoreet nunc ac egestas. Cras consequat elit eu lacus sollicitudin ut pharetra magna venenatis. Suspendisse scelerisque condimentum pulvinar. Mauris ut tellus sit amet nulla porttitor tristique. Suspendisse eleifend erat sed nisi lacinia eu lacinia metus porta. Nulla pretium, risus eget semper laoreet, dolor odio malesuada eros, at mattis enim turpis gravida felis. Aliquam adipiscing varius nibh, ac auctor eros bibendum non.

Innehåll

1	Inledning	5
1.1	Mobile-first eller Desktop-first	5
1.2	Syfte	6
1.3	Avgränsningar	6
1.4	Valtech AB	7
2	Bakgrund	8
2.1	Mobilanvändare	8
2.2	Desktopanvändare	9
3	Teori	10
3.1	Responsive web design	10
3.1.1	Fluid Grid	10
3.1.2	Fluid Images	11
3.1.3	Media Queries	11
3.2	Mobile-first	14
3.3	Desktop-first	14
4	Metod	16
4.1	Litteraturstudie	16
4.1.1	Miljö	16
4.1.2	Målgrupp	16
4.1.3	Kontext	17
4.2	Implementation av webbsida	17
4.2.1	Prototyp	18
4.2.2	Webbsidan i Desktop-first	18
4.2.3	Webbsidan i Mobile-first	18
4.3	Verktyg	19
4.3.1	HTML	19
4.3.2	CSS	19
4.3.3	JavaScript	21
5	Resultat	22
5.1	Litteraturstudie	22
5.1.1	Samlade synpunkter kring Mobile-first och Desktop-first	22
5.1.2	Användningsområde	26
5.2	Implementation	29
5.2.1	Prototyp	29
5.2.2	Responstid	32
5.2.3	Kodstorlek	33
5.2.4	Implementeringstid	34
5.2.5	Kommentarer från dagboken	34

6	Diskussion	35
6.1	Mobile- vs Desktop-first	35
6.1.1	Responstid	36
6.1.2	Kodanalys	38
6.1.3	Implementeringstid	39
6.1.4	Skärmstorlek	41
6.2	För- och Nackdelar	41
6.3	När ska Desktop-first respektive Mobile-first användas	43
7	Slutsats	45

1 Inledning

Användningen av mobilt internet ökar för varje dag och nya webblösningar har krävts för att nå ut till mobilanvändarna. En lösning har varit responsiva webbsidor, vilket innebär att samma webbsida rendereras olika beroende på enheten webbsidan ses ifrån. Tekniken för att skapa responsiva webbsidor existerar, med hjälp av utvecklingen av tekniker och verktyg så som CSS, HTML finns i nuläget möjligheten att skapa webbsidor som beroende på skärmstorlek har olika designlayouts med en och samma kodgrund. Det finns även kunskap om hur man skall gå tillväga för att skapa en responsiv webbsida, där utvecklare med formler och riktlinjer kan lära sig att använda grundprinciperna fluid grid, fluid images och media queries på bästa sätt. Men vägen till att skapa en responsiv webbsida är olika. I bloggar och forum på webben diskuteras flitigt valet av utvecklingsmetod när det kommer till responsiva webbplatser och innan implementeringen av en webbsida är detta en fråga som med stor sannolikhet dyker upp hos webbutvecklare. Utvecklingsmetoderna man diskuterar är *Mobile-first* och *Desktop-first*. Mobile-first och Desktop-first är två olika metoder där det responsiva angreppssättet används. Båda strävar efter samma mål men skiljer sig i prioriteringen av vilken enhet webbsidan skall vara anpassad för först.

1.1 Mobile-first eller Desktop-first

Det som kännetecknar metoderna är grunden på strukturen och designen, där storleken på webbläsaren avgör på vilket sätt den skall ändras för att anpassas för en annan enhet. Kunskapen om metoderna var för sig har med tiden blivit större i samband med utvecklingen av responsive web design. Däremot ställs webbutvecklare kring frågan om vilken metod som appliceras bäst till den typen av webbsida som skall skapas. I dagsläget finns ingen kunskap om hur bra metoderna appliceras i jämförelse med varandra. Det finns tankar och spekulationer, därefter väljs metod utifrån dessa, komplikationer hanteras men dokumenteras inte och till slut har man en fungerande responsiv webbsida. Eftersom webbsidan är färdig implementerad finns ingen anledning till att implementera om en fungerande lösning med en annan metod, därav finns ingen större kunskap om jämförelse av implementationen med respektive metod riktat mot ett och samma resultat.

Användning av mobilt internet ökar och användningen av internet via en dator kommer med stor sannolikhet bestå. Vilket gör att en grund för val av metod är högst passande, då det redan nu och i framtiden kommer att kräva mer kunskap än vad som idag kan erbjudas från bloggar. Båda metoder har sina fördelar samt nackdelar, vilket bör lyftas fram tydligare. Även frågan som bör ställas är när dessa går att utnyttjas på bästa sätt, och kan beslutet av val leda till positiva faktorer vilka motsatta metod inte hade kunnat uppnå inom en specifik situation.

1.2 Syfte

Syftet med arbetet är att kunna hitta riktlinjer till när en specifik metod appliceras bäst. Det vill säga beroende på faktorer kring webbsidan hitta den metod som tillför det mest optimala lösningen för implementationen av webbsidan. Att hitta en metod som fungerar bäst i alla lägen är inte nödvändigtvis målet med arbetet då det existera många olika vinklar, vilka gör det svårt att hitta en specifik metod som ger det optimala resultatet oberoende på situation.

Frågeställningar som besvaras i rapporten är:

1. I vilka lägen appliceras metoderna bäst?
2. Vad är för- och nackdelarna med Mobile-first?
3. Vad är för- och nackdelarna med Desktop-first?

Frågeställning 1 baseras på miljö, målgrupp och kontext. Syftet med frågeställningen är att kunna utifrån analys hitta situationer där metoderna visar sig vara fördelaktiga under utvecklingsfasen, för att hitta faktorer hos metoden vilka motsatta metod inte har och därmed inte kan tillföra samma lyckade resultat. Syftet med *Frågeställning 2 och 3* är att kunna lyfta fram de fördelar och nackdelar som finns vid implementation med de två metoderna. För att på så sätt hitta ramar för varje metod vilka en läsare kan relatera till med en egen situation och utifrån det använda det som grund vid val av metod.

1.3 Avgränsningar

Arbetet kommer att enbart fokusera på mobile-first och desktop-first, ett mellanläge som existerar för Ipads, tablets, osv. kommer inte att tas med i arbetet utan kommer att ses som ett desktopläge. I dagsläget finns även andra lösningar för mobilt webb t.ex Appar och Hybrida Appar, jämförelse mellan dessa och responsive web design kommer inte att göras, utan jämförelsen som görs är mellan mobilalösningar inom responsive web design.

Termen *Mobile-first* kan tolkas på olika sätt. Det har förekommit tillfällen för webbutvecklare där mobile-first är prioriterad designmässigt men där implementeringen ändå har skett för desktop innan mobilen, vilket en produktägare har tolkat som mobile-first. Det är inte fallet i arbetet, utan mobile-first beskrivs i arbetet som en tanke och en implementering avsedd för mobilen i första hand, och tvärtom vid desktop-first. Testfallen under implementeringsfasen i arbetet kommer att bygga på den teori för ovanstående metoder.

1.4 Valtech AB

Examensarbetet kommer att utföras på Valtech AB i Stockholm. Valtech Sverige fokuserar främst på utveckling av användaranpassade hemsidor, webbapplikationer och intranät. Hos Valtech finns erfarna gränssnittsutvecklare som har stött på problem under utvecklingsprocessen i form av ett beslutstagande av tillvägagångssätt för en responsiv webbsida. Det finns ett behov hos utvecklare på Valtech att ha en grund för vilka faktorer som är viktigast vid beslutstagande för metod, en metod som ger den bästa lösningen för en effektiv utveckling av en responsiv webbplats. Gränssnittsutvecklarna besitter på mycket erfarenhet och kunskap, vilket kan samlas i form av intervjuer för att analyseras och sammanfattas utifrån verkliga situationer som förekommer i företag, för framtida syfte.

2 Bakgrund

Mobilutvecklingen har under de senaste åren gått fort framåt, vilket har lett till att mobiler numera utvecklas till att fungera likt en handburen minidator. Huvudsyftet med mobiler är inte längre att bara kunna kommunicera med andra, utan även att bland annat ha möjligheten att få ut information genom webben. Att en websida inte längre endast ses i en skrivbordsskärm har gjort att nya webblösningar krävs. En utvecklare måste ha i åtanke att webbdesignen bör fungera för en användare som sitter på kontoret och läser sidan från en skrivbordsskärm, likaväl för en användare som sitter på tåget och läser sidan från mobilen. *Responsive Web Design* är en lösning vilken beroende på skärmstorlek renderar samma sida på olika sätt. I responsive web design har det fokuserats på två olika metoder, *Mobile-first* och *Desktop-first*. Metoderna grundar sig på att utveckling sker för en typ av skärm först för att sedan utifrån det utveckla så att den sidan även passar för andra skärmar. Det finns mycket kunskap om metoderna var för sig, men inget underlag för valet av metod i olika situationer och inom olika områden. Mobilanvändarna ökar för varje dag medan skrivbordsanvändare behåller sina användare med en viss minskning. Det största antalet är användare utav båda, vilket kräver kunskap om när metoderna appliceras bäst för att få effektivare lösningar inom webbutveckling.

2.1 Mobilanvändare

Sedan 2007 då Apple visade upp den första versionen av iPhones(Seitz 2009) har utvecklingen för *smartphones* eskalerat markant. Smartphones ger dig möjligheten att utföra datorliknande handlingar som att få ut information från webben och använda nätbaserade applikationer. Tidigare mobiler har delvis haft den funktionen men fokus mestadels har varit på musikspelare och kamera, då surfande gick trögt och var icke praktiskt användbara. Med smartphones, vilket innebar större pekskärmar, gjorde åtkomsten till internet på mobilen mer användaranpassade och har under de senaste åren utvecklats till en oundviklig funktion i dagens mobiler(Helmreich m. fl. 2009, s. 4). Även prisutvecklingen har varit en givande faktor till användningen av mobilt internet, då under de senaste åren telekomföretag har varit tvungna att sänka avgifterna samt erbjuda fast pris för att kunna passa in i marknaden. Sedan 2007 fram tills 2012 har lägsta priser för mobilt bredband minskat med 60 %, i dagens erbjudanden kan man finna abonnemang med fri surf för 89kr i månaden med 18 månaders bindningstid, vilket i jämförelse med 2007 hade legat på 169kr(Fransén m. fl. 2012, s. 36). Numera ser man satsningen på att även kunna erbjuda ett fast pris för att surfa utomlands, vilket tyder på att en breddning för användandet görs, och antal svenskar som använder mobilt internet kommer att fortsätta öka.(Gustaffson 2012)

Av svenskar mellan 12-79 år är det cirka 54 % som använder sig av mobilt internet, i jämförelse med 2008 har det ökat med ungefär 70 %. Under tidiga åren under 2000-talet låg ökningen per år på 1 %, vilket visar att stora genombrottet har skett under de två senaste åren(Findahl 2012, s. 24). Telefonoperatörer har anpassat sig till den nya marknaden och erbjuder abonnemanger vilka innehåller en fast kostnad för surf på ett 3G nät. Då

mobilsurf anses som en nödvändig funktion, sätts krav hos mobiloperatörer i form av att hög åtkomlighet och en hög hastighet för 3G nätet(Helmreich m.fl. 2009, s. 17). Effekten av att mobilanvändarnas åtkomst till internet ökar, leder till mobilt internet används mer och äldre tjänster ersätts. En tidning i tunnelbanan är inte alls lika vanlig nu som för fem år sen, tidtabell vid busshållplatsen är inte enda sättet att få ut information om busstider och att checka-in inför en flygresor behöver inte nödvändigtvis ske via en check-in disk(Meeker m.fl. 2009).

Konkurrensen som finns i dagens mobilmarknad har tvingat ledande företag att skapa mobiler med ny teknik till allt lägre priser. Pris, tillgänglighet och användbarhet har gjort att användning av internet via mobilen i världen närmar sig antalet användare av internet via en dator(Stanley 2009). Analytiker som förutspådde mobilen till att slå i marknaden har i dagsläget fått det bekräftat och förutspår att användare av mobilt internet i världen kommer att passera antalet användare av Desktop internet under år 2014(Stanley 2009). Detta kräver från webbplatser att följa målgruppen användare och anpassa webbsidor utifrån de enheter som webbsidor ses ifrån, vilket till stor del är från mobilen.

2.2 Desktopanvändare

Att mobilanvändare ökar för varje dag har till viss del ett samband med minskningen av antalet *desktop* användare. Men datorer har i dagsläget funktioner som gör det osannolikt för mobiler att helt kunna ersätta datorer, funktioner som kräver att man sitter vid en dator och på så sätt sker informationssökandet i webben via desktop.

Webbdesignen på en mobil är kompakt och uppfyller den nödvändigaste användbarheten. På desktop är designen mer informationsrikt, vilket ger en större inblick till webbsidans innehåll och en mer simpel navigering på webbsidan. Det gör att en användare beroende på komplexitet och säkerhet av handlingen väljer att utföra den via en desktop(Singhal 2009), än via mobilen. En sådan handling kan t.ex. vara bankärenden eller webshopping. Om bankärendet gäller en översikt av saldo i kontot eller överföring mellan egna konton, anses mobilen vara en smidig enhet att använda sig av. Däremot om handlingen innefattar att betala räkningar eller överföra stora summor pengar finns behovet av att utföra dessa handlingar via desktop, då det ger en större säkerhet och en mer simpel navigering på webbsidan. Även webshop faller i samma kategori, då användare väljer att söka information om produkten via mobilen, men väljer sedan att utföra köpet via desktop eller i affären(Singhal 2009). Detta tyder på att användare av mobilt- eller desktop internet inte är endera, utan är användare utav båda, beroende på situation, miljö och kontext på webbsidan.

Det finns ingen grundlig faktum som tyder på att mobilt internet kommer att ta över all desktop internet användande, endast att de kan bli fler. Därför krävs det att man vidgar vyerna kring webutveckling och lösningar som gynnar både mobil- och desktop användning av internet. För även om mobila användare blir fler, så går det inte att förbise desktop användare.

3 Teori

I nedanstående kapitel beskrivs den kunskap som krävs för att upprätthålla ett responsivt tillvägagångssätt för både *desktop-first* och *mobile-first*.

3.1 Responsive web design

Responsive web design är ett koncept som innebär att gränssnittet på en websida ändras beroende på skärmstorlek, vilket således ger möjligheten att ha olika layout på en och samma websida anpassat efter en enhet. Konceptet definierades av Ethan Marcotte (2010) i en artikel kallat *A List Apart*, som sedan blev en del av boken "*responsive web design*" där teorier och praktiska exempel används för att förklara begreppet.

Syftet med Responsive Web Design är att kunna rendera olika delar av sidan beroende på skärmstorlek för att ge en optimal vy för den enhet webbsidan ses ifrån. Om tre element renderas bredvid varandra på en desktop sida, så vore det optimala för en mobilskärm att kunna rendera elementen under varandra och även skala ner de till en rimlig storlek. På så sätt försvinner inte artiklarna från webbläsarfönstret eller tar för stor plats på skärmen.

Då endast layouten på sidan ändras behövs inga särskilda versioner för varje enhet, vilket gör det möjligt för en webbutvecklare att på ett enkelt sätt utföra en ändring i en fil, istället för antalet versioner. Däremot krävs en flexibel grundlayout för att responsive web design skall fungera, vilket enligt Ethan görs genom tre grundtekniker, *Fluid Grid*, *Fluid Images* och *Media Queries* (Marcotte 2011).

3.1.1 Fluid Grid

På en webbsida kan storleken på element definieras på olika sätt. Ett vanligt sätt att definiera element är med bredd och höjd i pixlar. När det definieras i pixlar betyder det att storleken på elementet är förutbestämd vare sig upplösning eller storlek på skärm. I tidigare skede informerade webbutvecklare användare vilken upplösning som renderade sidan på bästa sätt och sedan var upp till användaren att ändra upplösning på skärmen för att få den önskade layouten på sidan (Marcotte 2011, s. 6). I dagsläget finns många olika skärmar, och många olika alternativ till upplösning vilket gör en förutbestämd storlek inte lika optimalt.

Fluid Grid är en teknik vilken använder sig av procentsatser istället för pixlar vid definiering av ett elements storlek. När procentsatser används förstoras eller förminskas elementets storlek relativt till webbläsarfönstrets höjd och bredd, vilket gör att sidan istället anpassas efter användaren. Om ett element skall täcka hela bredden på skärmen, sätts bredden till 100 %, en fjärde del av skärmen, 25 % osv. För att kunna bestämma elementets storlek i procentform härleder Ethan Marcotte ett sätt vilket innebär att, i pixelstorlek, dividera

elementets storlek med behållarens storlek. Det ger ett resultat i procent, där elementet storlek ändras relativt till hållarens bredd och höjd (Marcotte 2011, s. 23).

3.1.2 Fluid Images

Fluid Images bygger på samma princip som Fluid Grid, däremot är lösningen mer komplex då det innefattar skalning av bilder. Om en bild skalas på fel sätt leder detta till att bilden ser för utsträckt eller för intryckt ut, men om skalning inte görs, kan en bild i princip ta upp all plats på webbsidan och även försvinna ut i kanterna. Inom responsive web design används Fluid Images, vilket innebär en flexibel hållare med en önskad storlek, med en bild anpassad till hållaren.

Lösningen som tas upp av Ethan Marcotte är användningen av egenskapen **"max-width: 100 %"** på bilden (Marcotte 2011, s. 45). Med egenskapen **"max-width: 100 %"** talar man om för webbläsaren att bilden skall skalas efter hållaren men endast till max av den originella storleken på bilden. Egenskapen **"width: 100 %"** innebär att bilden bara får anpassa sig helt efter hållaren, samtidigt som den får en normal skalning när storleken på webbläsaren minskar och hållaren krymper. Egenskapen **Max** i **"max-width"** innebär att bilden aldrig blir större än bildens verkliga storlek. Om *"max"* inte sätts, finns risk att bilden skalas ut, vilket gör pixelkanter synliga. Däremot krävs det att man har en bild som redan från början är av önskad storlek. En för liten bild kommer den inte att förstöras då den som störst kan vara 100 % av bildens ursprungliga storlek. Detta är ett sätt att ta använda sig av Fluid Images som fungerar för responsiva webbsidor, det kan uppstå andra komplikationer i form av att bilden inte skalas enligt önskemål där andra lösningar krävs, men grundtanken för att upprätthålla fluid images är att bilder skalas efter webbläsarfönstrets storlek.

3.1.3 Media Queries

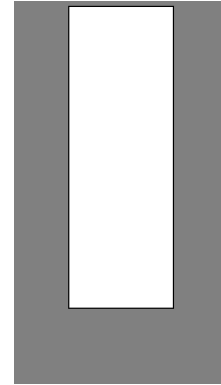
Fluid Grid och Fluid Images skapar tillsammans en del av en responsiv sida då storleken på element i sidan renderas utifrån storleken på skärmen. Minskningen eller förstoringen av elementen fungerar däremot endast till en viss gräns. Vid tillfällen då storleken på webbläsaren för liten, finns risk att element kommer för nära varandra och skapar konflikt i layouten, det leder till att element hamnar på felaktiga positioner och ger en vy som designmässigt ser förstörd ut. Blir storleken däremot för stor, finns risk att *max-width* på bilderna uppnås och stannar i storlek, medan andra element följer förstoringen hos webbläsaren, blir för stora och skapar en osymmetri i layouten. Media Queries är en lösning som tillåter element att ha olika värden beroende på skärmstorlek (Marcotte 2011, s. 65).

Inom responsive web design är bredden och höjden viktiga egenskaper då de avgör hur mycket av en sida skall synas på webbläsarfönstret. Media queries är anrop som görs i CSS filen, där man kollar antingen höjden och eller bredden för att utföra nödvändiga ändringar i layouten för att upprätthålla en bra design. Tillsammans med Fluid Grid och Fluid Image skapar Media Queries en responsiv webbdesign.

Vid användning av fluid grid och fluid image sätts egenskaper hos element i procentform, t.ex 50 %. Med en centrerung på elementet, blir resultatet en ruta i mitten av skärmen som täcker 50 % och har 25 % utrymme på var sida. När webbläsarfönstret minskas till storleken av en mobilskärm anpassas elementet relativt till förändringen, då fluid grid tekniken används för bredden. Vilket ger oss dessa två vyer, ena med webbläsarfönstret i storlek av en skrivbordsskärm och den andra i storlek av en mobilskärm:



Figur 1: Utifrån desktop



Figur 2: Utifrån mobilen

Denna design har endast ett element med egenskaperna:

```
div {  
    margin: 0 auto; //centrerad  
    width:50 %; //bredd  
    height:500px; //höjd  
}
```

Designen fungerar bra för skrivbordsskärmar (*Figur 1*), då rutan är centrerad och symmetrisk. 25 % av utrymmet på var sida av rutan gör designen läsvänlig och användbart då fokusering sker i mitten av skärmen där förmodligen information kommer att samlas. Däremot är den inte lika optimal när skärmen blir mindre (*Figur 2*). Rutan är fortfarande 50% av skärmen, vilket gör att designen blir kompakt i ett utrymme där förmedling av mycket information sker. 50% procent av skärmen går till spillo åt utrymme mellan rutan och kanter, vilket får rutan att se ihoptryckt ut. Därav en försämring på design och användbarhet när webbsidan ses ifrån en mobilskärm.

Med media anrop i CSSen, kallat för media queries, tillåter vi att värden hos "div" elementet skrivs över när bredden på webbläsaren underskrider 380px.

```
@media all and (max-width: 380px){  
  div {  
    margin:0;  
    width:100%;  
  }  
}
```

Med media anropet kommer "*width*" och "*margin*" skrivas över när skärmstorleken är 380px eller mindre, "*height*" förblir detsamma. Utrymmet mellan kanterna och rutan ändras till "0" och bredden till "100 %", vilket innebär hela skärmbredden. Detta leder till att sidan får andra designvärden när den ses från en mobil och andra enheter där skärmen har som störst 380px i bredd. Det ger en mer optimal lösning i både användbarhets och design perspektiv, då mer plats ges åt information och gör navigeringen lättare för en användare.



Figur 3: Resultatet vid användning av media queries

På så sätt kan samma CSS-fil låta element ha olika värden på egenskaper beroende på webbläsarstorlek. Responsive web design fyller sin funktion genom att tillföra en användbar design oavsett enhet som används för att se webbsidan.

3.2 Mobile-first

Mobile-first metoden bygger på att man skapar en webbsida anpassad för mobilskärmen först, för att sedan med hjälp av media queries och en flexibel layout renderera elementen på webbsidan desto större webbläsarfönstret blir. På så sätt är grund layouten designad för en mobil. Mobile-first anses som en rimlig utvecklingsmetod då man tar till hänsyn antalet mobilanvändare och det behov som finns vid navigering och informationsintagelse från mobilskärmar. Det betyder att innehåll prioriteras då bristen på plats är ett faktum och fokus läggs på de delar som informationsmässigt är de viktigaste (Lande 2009). Det behöver inte nödvändigtvis betyda att hänsyn till design för mobilen inte tas i desktop-first, utan snarare att implementationen för mobilskärm sker vid ett senare skede i utvecklingen än vid användning av mobile-first.

Kodmässigt är grunden anpassad för mobilen, layout och struktur är helt designat till mobil där förändringen sker vid ett media anrop som reagerar på när webbläsarfönstret blir större. Media querye anropet ser till att skriva om värden hos valda element så att de anpassas utifrån storleken på en skrivbordsskärm.

```
@media all and (min-width: 980px){  
    .main {  
        margin:0 auto;  
        width:50%;  
    }  
}
```

I fallet ovan sker ett anrop när bredden på webbsidan är som minst 480px, som gör att elementen med klassen main får "*margin*" och "*width*" skrivs över med nya värden. I fall då mobiler inte klarar av att läsa media queries (Clark 2009), vilket i dagsläget är få, är detta en optimal lösning då grunden redan är skrivet för mobilen och en läsning av media queries endast kommer att krävas från en skrivbordsskärm vilket de flesta webbläsare klarar av.

3.3 Desktop-first

Desktop-first metoden bygger på att man utvecklar för skrivbordsskärmen först och därefter rendererar sidan desto mindre skärmen blir. Det behöver inte nödvändigtvis betyda att en färdig sida för desktop i efterhand designas om till mobil, utan tanken för mobil finns redan från början men implementeringen sker i synnerhet för skrivbordsskärmen först. Webbssidor som i efterhand skapas till mobilen, brukar innebära en separat mobilsida då refaktoreringen av kod i samband med att förvandla sidan till responsiv kan innebära en del komplikationer (Clark 2009).

Kodmässigt är grunden anpassad för skrivbordsskärmen. Det vill säga att media anrop sker när skärmbredden når en minimum gräns, där media queries ser till att skriva om värden på valda element för anpassa de efter mobilen istället för skrivbordsskärmen.

```
@media all and (max-width: 380px){  
  .main {  
    margin:0 ;  
    width:100\%;  
  }  
}
```

I fallet ovan sker ett anrop när bredden är som max 380px. Anropet ger "*margin*" och "*width*" i klassen main nya värden, anpassade till mobilen. Desktop-first används oftast i samband med hemsidor där man vill skapa en upplevelse och vill att webbsidan skall nå ett design maximum när den ses utifrån en skrivbordsskärm, upplevelsen utifrån en mobilskärm är inte lika högt prioriterad, men bör innebära en webbsida med den nödvändigaste funktionaliteten.

4 Metod

För att kunna svara på frågeställningarna är tillvägagångssättet uppdelat i en praktisk och en teoretisk del. Syftet med uppdelning är att kunna sammanställa resultat genom en litteraturstudie, vilket består av en analys av den erfarenhet som finns hos gränssnittsutvecklare, interaktionsdesigners, användbarhetsexperter och profiler inom området och en analys av användningsområden för mobilt och desktopinternet. Tillsammans med påståenden som fås utav den praktiska delen, vilket är en implementering av båda metoder gentemot samma webbsida utgör detta tillvägagångssättet för att besvara frågeställningarna. Den praktiska delen vilket innebär gränssnittsutveckling, är den som tar mest tid och är den del som har vägt tyngst i arbetet. Litteraturstudien har använts för att besvara de områden som resultatet av implementationen inte har kunnat nå fram till, samt försvara de påståenden som tagits utav implementationsfasen.

4.1 Litteraturstudie

I litteraturstudien har bland annat 13 källor i form av bloggar samlats och analyserats, där skribenterna består utav gränssnittsutvecklare, interaktionsdesigners, användbarhets experter och profiler inom området. Dessa har kommit till användnings för att få en bild av de problem som kan uppstå, samt samla synpunkter och erfarenhet om för- och nackdelar gentemot vardera metod. Tekniken är så pass nytt vilket gör att fakta kring metoderna mestadels finns i bloggar då det är en vanlig källa där profiler inom området delar med sig erfarenhet kring ny teknik. Analysen utav skribenternas synpunkter har även använts för att stärka påståenden som har tagit utifrån implementationsfasen av metoderna. Litteraturstudien består även utav 3-5 artiklar och statistik som har analyserats för att få en klarhet till användningsområde för mobilt- och desktopinternet. Det visade sig i första delen av litteraturstudien att när ett responsivt tillvägagångssätt utförs är användningsområdet en viktig aspekt att analysera då det kan vara en avgörande faktor för prioritering utav enhet och därmed val av metod. Artiklarna och statistik har analyserats utifrån enheterna användningsområde beroende på webbsidans målgrupp, kontext och miljö.

4.1.1 Miljö

Med miljö menas den miljö som existerar hos användaren vid besök av en webbsida och det sammanhang för besöket. En miljö kan t.ex. vara kontor eller resande, vilket leder till att olika enheter används beroende på den plats användaren befinner sig på.

4.1.2 Målgrupp

I dagens läge är internetsurfandet utspritt i många olika åldrar, vilket även gör målgrupp till en viktig faktor då användningen av enheterna är olika mycket bland olika åldrar. Det

spelar en betydelsefull roll till det enhet vars webbsida kommer att ses ifrån och därmed kunna vara en prioriteringsfråga där olika implementeringsmetoder kan vara optimala för webbsidan beroende på målgruppen den är avsedd för.

4.1.3 Kontext

Kontext handlar om en webbsidas innehåll och den information som vill förmedlas till användare. Webbssidor har olika mycket innehåll och olika typer av information, vilka kräver olika stora ytor för att kunna förmedla informationen till användaren på bästa sätt. Därav är valet av prioriteringen för enhet beroende av webbsidans kontext och den information som är anses viktig att förmedla.

4.2 Implementation av webbsida

Den praktiska och tekniska delen av examensarbetet utgörs utav en implementation av webbsida som konstruerades i både Mobile-first och Desktop-first. Syftet med implementationen är att jämföra vardera av metoderna genom att applicera dem för en och samma webbsida m.a.o. nå samma mål och resultat med två olika lösningar. Innan implementeringen har en prototyp designats och ritats utifrån det ramverk som anses rimliga för att webbsidan skall spegla ett riktigt gränssnittsprojekt och även en webbsida vars besökare är både via mobilen och via desktop. Sedan har utifrån design för mobil och desktop, en webbsida i mobile-first och en webbsida i desktop-first skapats med hjälp av HTML-, CSS- och javascriptskod för att uppnå den gränssnittsfunktionalitet som eftersträvas, i det här fallet, en responsiv webbsida. Under implementationens gång har en daglig rapport skrivits för varje metod, för att på så sätt kunna jämföra data gentemot samma period av implementeringsfasen för vardera av metoderna. Under implementationsfasen har kodstorlek på css-filen, responstid för webbsidorna samt implementerings tid använts som data för att jämföras. Genom utförandet av en dagbok för respektive metod har även kommentarer antecknats och samlats in tillsammans med nedskrivna implementationsstider, kommentarerna beskriver svårigheter och speciella utmaningar för varje metod. Implementationstiden är mestadels för att jämföra totala tiden, samt tiden som har tagits för att göra grunden till webbsidan och webbsidan responsiv. Responstid och kodstorlek jämfördes efter implementeringen. Kodstorlek jämfördes i både filstorlek och antal css rader där antalet css rader har tagits med som data för att kunna jämföra hur mycket av elementens utseende har skrivits i grundkoden och skrivits om i media queries. Responstiden har mätts genom att utföra tre olika hämtningar av webbsidornas css-fil och html-fil i både desktop- och mobilvy, för att på så sätt få reda på vilken metod laddar sidan snabbast för varje vy.

4.2.1 Prototyp

Prototypen är det första momentet i den praktiska delen. Rent designmässigt handlar det om att hitta en bra struktur på webbsidan så att den kan spegla ett verkligt gränssnittsprojekt där webbsidan uppfyller sitt syfte. Designen och kontexten av webbsidan som implementerades bestämdes med hjälp utav en litteraturstudie där syftet var att hitta en typ av webbsida där besökarna är både mobilanvändare och desktopanvändare. Anledningen till att en webbsida med lika många besökare utav båda enheterna valdes är för att det då speglar en svår situation för en gränssnittsutvecklare som måste på förhand välja en implementeringsmetod. Därför krävdes det av ritningarna att de uppfyllde de nödvändigaste kraven i form av struktur och design. När sidtypen valdes ut, jämfördes den med andra webbsidor av samma typ och ritades ut med hjälp av photoshop. Ritningarna avser en hel webbsida, mer än vad skärmen kan visa för att få en helhetsbild. Vid fall där menyer öppnas och stängs har ritningarna gjorts för vardera av fallen. Ritningarna har granskats utav en gränssnittsutvecklare på arbetsplatsen, vilket har lett till att ritningarna, efter korrigering, har använts som designbas för kodningen.

4.2.2 Webbsidan i Desktop-first

Desktop-first är första implementeringsmetoden som utfördes utifrån ritningarna. Valet av att implementera i desktop-first var slumpmässigt. Under implementeringen utvecklades webbsidan responsivt från början i form av fluid grids och fluid images. När webbsidan var klar för desktop, påbörjades utvecklingen med media queries för att få en mobilvy utav de element som fanns hos desktopvyn. I slutändan jämfördes gränssnittsutvecklingens resultat med skisserna, då skisserna är de mål webbsidan förväntas att nå. Vid sidan av utvecklingen genomfördes en rapport som vid ett senare tillfälle jämfördes med rapporten för mobile-first.

4.2.3 Webbsidan i Mobile-first

Mobile-first är andra implementeringsmetoden och utvecklades efter desktop-first. Implementationen genomfördes direkt efter implementationen för desktop-first, då analysen av implementeringsfasen skulle göras efter att båda rapporterna för utförandena var slutförda. Under implementeringen utvecklades även här webbsidan responsivt från början i form av fluid grids och fluid images. När webbsidan var klar för mobilen, påbörjas utvecklingen med media queries för att få en desktopvy utav de element som har utvecklats hos mobilvyn.

4.3 Verktyg

Inom webdesignsutveckling används främst verktygen HTML, CSS och javascript, vilket är det verktyg som kommer att användas under implementeringsfasen i examensarbetet. I kommande sektion av rapporten kommer verktygen att förklaras för att få en bättre förståelse till kod och termer som används.

4.3.1 HTML

HTML står för *Hypertext Markup Language* och är ett format som definierar strukturen och logiken på en webbsida. HTML beskriver strukturen genom att märka upp olika delar av sidan med hjälp av *taggar* som beskriver typen av ett element. En webbläsare läser av HTML-koden och kan på så sätt rendera sidan med önskvärd layout.

Det finns olika sorters taggar inom HTML, stycke, rubrik, tabeller, länkar, listor, sektioner är en av dessa. Inom webbutveckling har användning utav taggen `<div>` blivit vanlig för att strukturera upp en sida i olika sektioner(Rana 2009), vartefter kan en sektion innehålla andra element.

```
<div>
    <h1> Mobile-first eller Desktop-first?</h1>
    <p>Frågan en webbutvecklare ställer sig inför skapandet av
        en responsiv webbsida
    </p>
</div>
```

I ovanstående HTML-kod har vi definierat en sektion på webbsidan med `<div>` taggen, inom den sektionen har vi en rubrik som definieras med taggen `<h1>` och ett stycke som definieras med taggen `<p>`. Taggarna i html-koden bildar tillsammans strukturen på sidan, däremot så har inte html-koden någon kontroll över utseendet för taggarna, det sköts utav CSS.

4.3.2 CSS

CSS står för *Cascading Style Sheets* och är ett språk som beskriver utseendet på en webbsida. Med hjälp av CSS-kod kan olika element i html-koden få ett speciellt utseende i form av storlek, färg och position. Det finns olika sätt att definiera css, man kan definiera det direkt i taggen t.ex `<p style="color:blue; width:50px">Frågan en webbutvecklare ställer sig inför skapandet av en responsiv webbsida</p>` vilket kallas för *inline-styling*. Man kan även i html filen, definiera utseendet i `<head>` taggen, vilket kallas för *internal styling*(w3schools 2009):

```
<head>
    <style>
    p {
    color:blue;
    width:50px;
    }
    </style>
</head>
```

På så sätt får alla `<p>` taggar i html-koden egenskaperna som sätts. Trejde sättet är att definiera det i en egen css-fil och länka till det från `<head>` taggen i html-koden, vilket är det vanligaste av de tre sätten och kallas för *external styling*.

```
<link rel="stylesheet" type="text/css" href="main.css">
```

I css-kod definieras utseendet av olika taggar. Det finns möjlighet att definiera samma tagg med olika utseenden, detta görs genom att sätta *Klasser* och *IDs* för varje element. Ett element kan ha olika klasser eller ids, vilket definieras med ett utseende i css-koden. Det gör att varje tagg som delar klassen, delar även utseendet:

I html filen:

```
<div id='main-container'>
    <p class='paragraf'>Mobile-first vs Desktop-first</p>
</div>
```

I css filen:

```
#main-container {
    width:50%;
    background-color:grey;
}

.paragraf{
    font-size: 125%;
    color:black;
}
```

I ovanstående kod kommer sektionen få utseendet som definieras som ID: *main* i css-koden och paragrafen utseendet som definieras som Klass: *paragraf*.

Dessa tre olika sätt att definiera utseendet visas enligt prioritering, inline-styling kod först, sedan internal och sist external. Anledningen till att external väljs främst är för att utseendet inte behövs definieras flera gånger, allt är samlat i en fil och det går snabbare att ladda sidans utseende(w3schools 2009). Därmed hålls det ur en webbutvecklarens synvinkel, en bra struktur. Vilket tillåter webbutvecklaren att på upprätthålla teknikerna fluid grid, fluid images och på ett enkelt sätt använda sig av media queries vid implementering av en webbsida.

4.3.3 JavaScript

JavaScript är ett scriptspråk som inom webbutveckling främst används för att hantera dynamiska funktioner för beteendet hos en websida, beteenden som skapas från klientsidan. Med JavaScript kan man t.ex. läsa en användarens klick i webbsidan och utifrån det kalla på funktioner som kan ändra webbsidans innehåll eller utseende. Eftersom JavaScript är ett skriptspråk behövs ingen kompilering och koden kan skrivas direkt i html-filen.

För att JavaScript skall fungera korrekt krävs det dock att webbläsaren stödjer det. I vissa fall har webbläsaren den funktionen men har den avstängd(Ingmar 2012, s.13). Tidigare ett problem varit att mobilwebbläsare inte har haft stöd för JavaScript, vilket har lett till att menyer och pop-up fönster inte har fungerat korrekt när man surfar till sidan via mobilen, men tekniken har utvecklats och numera klarar mobila webbläsare av JavaScript. Det finns fortfarande en del problem när det kommer till mer komplicerade JavaScript funktioner och bibliotek som mobilwebbläsare inte kan hantera(quirksmode.org 2009). Vid webbsidor som har helt separata mobilsidor kan det vara en fördel då man väljer att inte läsa in all JavaScript som behövs, med responsiva webbsidor använder man samma webbsida för varje enhet, vilket leder till att en mobilenhet laddar all JavaScript vilket försämrar webbsidans respons. En webbutvecklare måste ha i åtanke och redan från början utveckla en hemsida vars JavaScript inte kommer i vägen för den responsiva webbsidans funktionalitet.

5 Resultat

5.1 Litteraturstudie

I detta kapitel har resultat samlats från litteraturstudien som gjordes uppdelat i "synpunkter angående mobile-first och desktop-first" samt "användningsområden för mobilt internet".

5.1.1 Samlade synpunkter kring Mobile-first och Desktop-first

Utifrån de 13 olika källor som har samlats och analyserats säger 11 att mobile-first kommer att vara, om inte redan är, nästa stora teknik inom responsiv web design. Den främsta anledningen som poängteras är att mobilt internet börjar bli så oerhört stort att webb-utvecklingen måste finna ett anpassningssätt för det behov som finns hos mobilanvändare utan att gå miste om det som har funnits en längre tid, desktopanvändare. Anpassningen bör ske riktat mot att vid ett tidigt skede ta sig över barriärer så som mindre skärmar, sämre prestanda och begränsad internethastighet(Zee 2012), där statistik visar att om en användare finner mobilvyn slö, finns även risken att desktopvyn tappar användarens intresse(zurbword 2012). Mobilanvändare kräver i dagsläget snabb respons från mobilsidor, då många aktiviteter sker i tidspressade situationer(Wroblewski 2010). Med desktop-first finns risk att man skapar en stor sida med många element som även laddas i mobilvyn, vare sig dessa visas eller inte. Det försämrar mobilvyns respons och skapar även en kompakt, komplicerad design(zurbword 2012) eller en nerskalad version av desktopvyn.

Utav 13 källor påpekar 6 av dessa på funktioner hos mobilen som kräver en annan typ av fokus, då dessa kan ta webbsidan till en helt annan nivå funktionsmässigt. Funktionerna man syftar på är GPS, Touchscreen och Multitouch. Dessa tre funktioner kan anpassa en mobilsida till att fungera på användningsområden som i dagsläget inte fungerar lika bra från desktop, men som utvecklas åt att göra det(Dimestam 2012). Dessa funktioner är nya tekniker vilka fortfarande utvecklas för att nå optimal funktionalitet. Ett tidigt fokus på dessa ger en stabil grund för funktionaliteten hos mobilsidan, vilket gör mobile-first till ett bra alternativ(Wroblewski 2010, Chapman 2012).

Webbutveckling har sin grund hos desktopsidor, vilka har funnits i flera år. Flera källor påpekar detta i mån om att poängtera att ett säkert och välprövat alternativ, ibland kan kännas som den bästa startpunkten(Houx 2013, Chan 2013). Mobile-first är en ny teknik och som kräver ett annat tänk än den vid implementeringen av en desktopsida vilket då oftast kräver längre tid vid implementeringen än desktop-first och kan således kosta mer(Chan 2013, Pope 2012). Däremot anser alla förutom en källa(Houx 2013) att argumentet inte håller då detta leder till en ond cirkel där ny och bra teknik inte används eller utvecklas(Johnson 2013, Pope 2012). Webbutvecklare får även ha i åtanke att vanliga funktioner hos desktop som har existerat under en längre tid, i mobilt inte existerar(Foster 2012, Johnson 2013, Mellas 2011). Vid desktop-first finns risken att använda funktionaliteter, t.ex hover, vilka

kan leda till komplikationer när mobilvyn börjar implementeras(Chan 2013), med det menar källan att funktioner och innehåll kan av misstag försvinna utan att de har varit avsedda att tas bort(Chan 2013). *Exemplet med hover görs i samband med desktop-first, där en ruta med text i desktopvy blir större när man håller musen över, rutan visar då all text istället för en liten del av texten som visas när man inte håller musen över. I implementeringen av mobilvyn går inte samma kod att använda då "hover" inte fungerar, vilket leder till att elementet antingen är stor hela tiden(tar mycket plats) eller liten och endast visar en del av texten. Vid sen påkommenhet kräver detta att utvecklare får be interaktionsdesigners att finna en lösning så att texten får plats i mobilvyn eller atthaha fakti elementet prioriteras om.*

En nackdel med mobile-first är att Internet Explorer 6, 7 och 8 inte stödjer media queries(Pope 2012, Hansson 2012) vilket lösningen fås genom javascript som av prestanda skäl inte är bra för ett mobilgränssnitt(Foster 2012), fördelen med desktop-first blir då att media queries inte behövs läsas utifrån en desktopvy(Hansson 2012), utan det görs via mobilens webbläsare vilka de flesta klarar av tack vare den ständiga uppdateringen som görs för webbläsare i mobiler(Mellas 2011).

Utav 13 källor var det 2 som inte ansåg att webben och tekniken för responsive web design var redo för mobile-first(Grigsby 2011, Houx 2013). Enligt en studie hade flertal mobile-first sidor granskats och nått slutsatsen att mobile-first sidor blev större än desktop(Grigsby 2011). Enkelheten med desktop-first gentemot mobile-first tas åter upp med hänvisning till att desktop har funnits under en längre tid, vilket medför att det är simpelt för användbarhetsexperter att ta sig an, samt är en säker grund för gränssnittsutvecklare att utveckla responsivt ifrån(Houx 2013). Skapandet av en responsiv webbsida efterhand innebär mycket refaktorering med mobile-first(med antagandet att alla sidor som inte är responsiv är byggda för desktop). Vilket är något som resterande källor inte säger emot(Hansson 2012), men syftar på att en webbsida i desktop-first har tendensen att få mobilvyn förmedla en känsla av en nödlösning snarare än en genomtänkt produkt(Johnson 2013, Chapman 2012). I det fallet anses en bättre lösning vara att påbörja en ny design för mobilt, istället för att från desktop designa om till en responsiv webbsida.

7 utav 13 anser mobile-first vara en optimal lösning då den enkla principen säger att om element får plats i mobilvyn får de alltid plats i desktopvyn, gäller inte tvärtom. Med desktop-first kan element av misstag försvinna vid nedskallning till mindre skärmar då de anses överflödiga och inte får plats(Bull 2012(11) Foster 2012). Dessa hänvisar även till att webbdesign har tagit ett nytt steg mot användbarhet där webbsidan skall vara simpel för användaren att förstå och innehålla relevant information snarare än att den skall fänga användaren genom "överflödig" design(Bull 2012).

3 källor anser att användningsområde för webblösningen är en viktig aspekt att analysera inför implementeringen av den responsiva webbsidan(Hansson 2012, Pope 2012, Johnson 2013). Vilket fördelarna i de flesta källor redan bygger på att man har en aning om målgruppen i stort(zurbword 2012). En vanlig siffra som förekommer är att 25 procent av USA's befolkning använder sig endast av mobilt internet och där ökningen som sker för

användning av mobilt internet tyder på att mobile-first är ett bra tillvägångsätt till en responsiv webbplats. Men 25 procent av användare som enbart använder mobilt internet tyder även på att 75 procent antingen använder båda eller endast desktop, och är därför inte en tillräckligt övertygande anledning(Pope 2012). Källorna menar då att en bra riktlinje till val av metod är att analysera användarna som webbplatsen kommer att inrikta sig mot(Hansson 2012).

5.1.1.1 Nämda för- och nackdelar

Desktop-first

Fördelar	Nackdelar
Går fort att redesigna en existerande desktopsida till responsive, om elementen är definierade i procentform(fluid grid, fluid images).	Mycket kod som inte används läses in när sidan ses från mobilen, leder till sämre respons när sidan ses i mobilvy
Enkelt koncept att förstå sig på, finns mycket erfarenhet från både kund, designers och utvecklare.	Lätt att funktioner för desktop försvinner i mobilt, kan leda till att mobilvyn tvingas redesignas och element får prioriteras om.
Fokus på enheten den större delen av befolkningen använder. Även om antalet mobilanvändare växer består antalet som använder desktop.	Mycket fokus på desktop leder till nerskalad mobilvy design- och funktionsmässigt, eftersom mobilvyn har fler begränsningar.
Bra för webbsidor om internet explorer 6,7 och 8 används, dessa kan inte läsa media queries, desktop blir då kodgrunden och media queries läses endast utav mobilwebbläsaren.	Mindre fokus på mer speciella funktioner för mobilt t.ex hardware acceleration, touch, location awareness.
Säker punkt att utveckla responsivt ifrån, mer erfarenhet leder till mindre implementeringstid således blir kostnaden mindre.	All innehåll från desktopvyn får inte alltid plats i mobilvyn, samma element på en stor yta får inte alltid plats på en mindre yta.
	Problemen som finns hos mobilvy så som prestanda och sämre bandbredd åtgärdas i efterhand.

Tabell 1: Desktop-first, nackdelar och fördelar som samlades i litteraturestudien.

Mobile-first

Fördelar	Nackdelar
Användbarhet i fokus , att börja i en begränsad yta tvingar fram en design där det nödvändigaste prioriteras.	Ny teknik, nya utmaningar , kan ta längre tid att implementera
Fokus på innehåll , innehåll prioriteras först, sedan presentation och sist animation.	Internet explorer 6,7,8 klarar inte av media queries
Skapar anpassningsbara och framtidsvänliga hemsidor , varje enhet får ut det bästa från en och samma webbsida.	Liten skärm att påbörja designen för , svårt att få en helhetsbild för hur desktopvyn skall renderas.
Robust grund för arbetssätt och design , implementering sker där det viktigaste är i centrum och allt byggs utifrån det.	Komplicerat om man ska redesigna en existerande desktopsida
Robust grund för komplikationer , svårigheter tas i början (prestanda, design, prioritering).	Redan från start möter man det svåra , kan uppfattas som trist och jobbigt.
Läser in mindre kod i mobilvyn , behöver inte läsa in allt för desktop, utan bara det för mobilen.	Svår koncept att sälja , inte lika populärt bland kunder.
Anpassning efter dagens marknad , användning av mobiltinternet öka för varje dag.	Simpelt är svårt, prioritering är svårt , speciellt med olika investerare som tidigt vill få in varumärket i en produkt.
Fokuserar på kontext som fungerar för plats, tid och sociala egenskaper , situationer då snabb respons krävs.	
Simpelt att arbeta med nya funktionaliteter för mobilen , multi-touch, gps, hardware acceleration, location awareness.	
Allt från mobilvyn fås med till desktopvyn , samma element i en liten yta får plats i en stor yta.	
Känns naturligt att lägga till desto större fönstret blir , gentemot att ta bort desto mindre skärmen blir.	

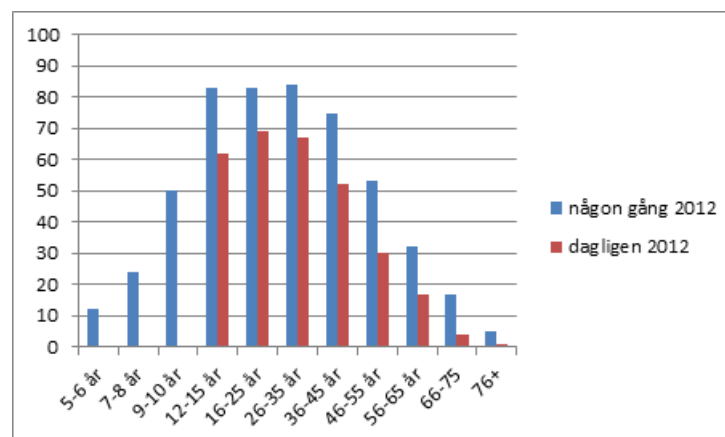
Tabell 2: Mobile-first, nackdelar och fördelar som samlades i litteraturstudien.

5.1.2 Användningsområde

Mobil- och desktopwebbsidor har olika besökare beroende på typ av webbsida och under vilka förhållanden den ses ifrån, även åldern spelar roll då det visar sig i statistik att ungdomar är de största användarna av mobilt internet (Findahl 2012, s.24). I Sverige utförs det varje år en statistiksredovisning på svenskar och deras internetanvändning (Findahl 2012). Trots distansen ser användningen av mobilt internet ungefär likadan ut i Sverige och USA, i detta kapitel visas resultat i en litteraturstudie där fokus läggs på användning av webbsidor beroende på användarnas åldersgrupp, kontext på webbsidan samt miljön webbsidan ses ifrån.

5.1.2.1 Målgrupp

I de vetenskapliga artiklar som togs med i litteraturstudien gällande användning av mobilt internet har dessa inriktat sig till användare där målgruppen är från 19 uppåt, med en medelålder mellan 20-29. Detta tyder på att en stor del av användarna för mobilt internet består utav nämnda målgrupp. I statistik som utförts för att ta reda på svenskarnas användning utav mobilt internet sträcker sig lägsta ålder ner till 12 år gällande mobilanvändning i en daglig basis. Trots detta stämmer målgruppen som har valts i de vetenskapliga artiklar väldigt bra då statistik visar att den stora målgruppen i Sverige av mobil internetanvändare är mellan 16-25 år där 69 % av målgruppen använder sig utav mobilt internet dagligen. Tätt efter är målgruppen från 26-35 med 68 % och 12-15 med 67 %. Den stora förändringen sker inte förens målgrupp där åldern avser 46-55 år då endast 30 % använder sig utav mobilt internet dagligen (Findahl 2012, s. 24). Lägst är målgruppen från 76 uppåt som ligger på 1 % men där internet användningen i överhuvudtaget inte är stor.



Figur 4: Andel procent av befolkningen som använder sig utav mobilt internet, statistik visar att den stora målgruppen är mellan 12-45 år gamla (Findahl 2012, s. 24).

5.1.2.2 Kontext

Med kontext menas sidans typiska innehåll, huvudsyftet med webbsidan. Beroende på webbsidans kontext kan användarna finna det simplare att använda antingen desktop, mobilen eller den enhet närmast till hands. Av de vetenskapliga artiklar som har tagits med i litteraturstudien har alla gemensamt att mobilt internet används flitigt för:

- Sköta e-post
- Sociala nätverk
- Kolla nyheter, väder
- Söka information

Andel procent av användarna skiljer sig beroende på artikel och statistik, då antalet användare för studierna har varit olika, men ändå visar vad fokus läggs på när mobilt internet används. I Sverige uppskattas 50 % använda sig utav mobilt internet för att sköta e-post, 43 % för socialt nätverk, 40 % för att kolla vädret, nyheter och 18 % för att söka fakta (Findahl 2012, s. 25). I en annan studie där användningen utav mobilt internet analyserades med hjälp utav 109 deltagare visade den att 70 % av deltagarna använder mobilt internet för att kolla nyheter, väder och söka fakta, 13 % utav deltagarna använder mobilt internet för sociala nätverk och 17 % för att kolla på film och lyssna på musik (Schmiedl m. fl. 2009). I en liknande studie där 18 aktiva mobilanvändare studerades varje dag under en 4 veckors period, bestod den dagliga användningen av mobilt internet 27 % för sociala nätverk, 24 % för koll på nyheter och vädret, 18.8 % för email, 14.9 för vanlig surf, 10 % för mobilsök och 5.1 % för kartor och navigation (Church m. fl. 2011).

För desktop anses dessa fyra områden vara minst lika populära. I en studie som gjorde hos The Harris Polls Research (Shannon-Missal 2013) samlades 2400 vuxna varav 991 mobilt internet användare för att se hur enheterna används i jämförelse beroende på kontext. I studien hamnade *Sköta e-post*, *sociala nätverk* och *söka information* längst upp som de typer av hemsida vars desktopanvändare är ungefär lika stora om mobilanvändare.

	Dator(desktop/laptop)	Mobil(Smartphone)
Sköta E-post	90 %	72 %
Sociala Nätverk	69 %	64 %
Söka information	81 %	45 %
*Kolla nyheter, väder	44.2 %	52.1 %

Tabell 3: Områden i webben där användarna utnyttjar både mobil och desktop.

**I studien ingick inte läsning utav nyheter, men i Q1 Research report för nyhetsläsning (Fidler 2013) visades det sig, med undantag från målgruppen 65 uppåt, att majoriteten utav läsare-*

na(59 % till 77 % i varje målgrupp) använder både mobil och desktop. Studien visar även att nyhetsläsare från mobilen numera är större än läsarna från desktop.

I The HarrisPole Research fanns även siffror för områden där användaren föredrar en enhet mer än den andra.

	Dator(desktop/laptop)	Mobil(Smartphone)
Fylla enkäter	86 %	24 %
Handla produkter	78 %	23 %
Navigation/Kartor	56 %	73 %

Tabell 4: Områden i webben där användarna föredrar en enhet mer än de andra.

5.1.2.3 Miljö

En användares miljö har förändras tack vare mobilens handburna storlek. Det har gjort att miljön inte är längre är den vanliga sederliga hemma eller på kontor, utan kan vara i vilket sammanhang som helst och även vart som helst. Ett vanligt antagande är att mobilen endast används vid rörliga situationer(Wroblewski 2011, s.12). Vilket vetenskapliga artiklar och statistik visar tvärtemot(Church m. fl. 2011, Wroblewski 2011).

Författaren utav mobile first hänvisar(Wroblewski 2011, s.12) till en enkätundersökning vilket visar användningen utav internet genom smartphones:

- 84 % använder det hemma.
- 80 % använder det under diversa dödtid genom dagen.
- 74 % använder det i kötid och väntan på möten.
- 69 % använder det under shopping.
- 64 % använder det på jobbet.
- 62 % använder det medan man kollar på TV.
- 47 % använder det under pendling.

Att en stor andel av smartphonebärare använder sig utav mobilt internet i hemmet styrks utav fler vetenskapliga artiklar(Church m. fl. 2011, Wroblewski 2011). Där orsakerna pekar på bekvämligheten av att använda mobilen istället för datorn. Mobilen tillåter 1 minuts användning utav internet, t.ex. vid brådskande tillfällen, vilket kan anses vara omständliga via en desktop(Church m. fl. 2011). I studien där 18 deltagare skulle föra dagböcker(Church m. fl. 2011) under en period av 4 veckor var över 70 % av inläggen om aktiviteter gjorda vid ett stationärt perspektiv, det vill säga, antingen på jobbet eller i hemmet. Bara 17 % av inläggen handlade om aktiviteter som gjordes vid resor, utomhus och vid pendling.

- 49,6 % av inläggen handlade om aktiviteter gjorda i hemmet.
- 21,8 % av inläggen handlade om aktiviteter på jobbet/universitetet.
- 11,4 % av inläggen handlade om aktiviteter gjorda inomhus.
- 9,6 % av inläggen handlade om aktiviteter gjorda under pendling eller transit.
- 7,2 % av inläggen handlade om aktiviteter gjorda utomhus.
- 0,5 % av inläggen handlade om aktiviteter gjorda under resa utomlands.

5.2 Implementation

5.2.1 Prototyp

Efter resultatet av litteraturstudien blev prototypen till webbsidan en nyhetssida. Grunden till valet gjordes med hänvisning till resultatet från litteraturstudien där statistik visar att andelen desktop och mobilbesökare hos en nyhetssida är jämnt fördelat, och även en av de största inom mobilområdet. För att skapa sidan gjordes en jämförelse mellan stora nyhetssidor så som aftenbladet.se, metro.co.uk, globalnews.ca och time.com, för att upprätthålla typiska element som finns hos nyhetssidor. Allt som finns i ena vyn finns även i den andra.

5.2.1.1 Banner

Banner visar tydligt vilken sidan man har kommit till. Beskriver namnet på sidan och en slogan som beskriver typen av webbsidan. Både i mobilt och i desktop går dessa inte att undgå, för igenkännande hos användaren, men är på mobilsidan mindre proportionerligt till artiklarna, tillskillnad från desktopvyn (*figur 5*). Detta för att undvika bannern ta för stor plats på mobilskrmen (*figur 6*).



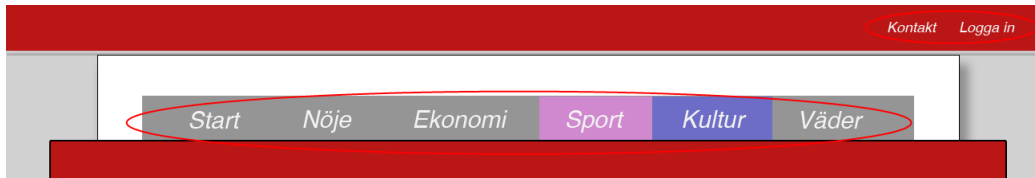
Figur 5: Banner för desktopvyn.



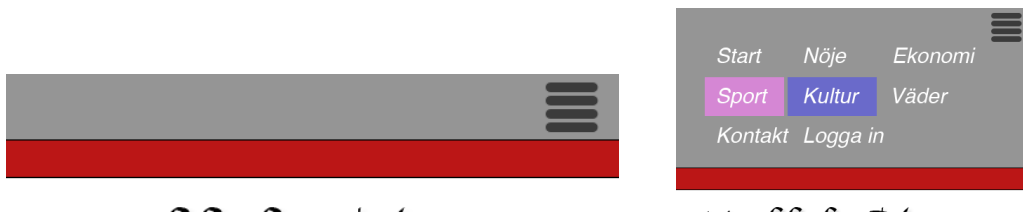
Figur 6: Banner för mobilvyn.

5.2.1.2 Menyer

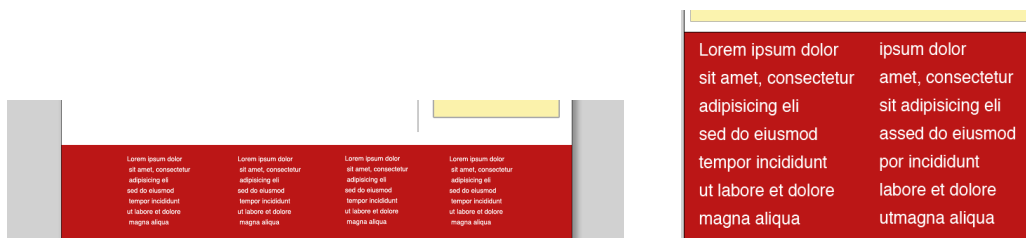
Nyhetsidorna som användes som referens för prototypen hade alla minst tre olika sorters menyer, en huvudmeny. En mindre meny bestående av typiska ”skapa konto, logga in” länkar(*figur 7*), samt en nedre meny längst ner på sidan med diverse länkar(*figur 9*). I mobilvyn, behölls nedersta menyn i en ihopsamlad version och de två översta sattes ihop och gömdes under en menyknapp för att spara plats på mobilskärmen(*figur 8*).



Figur 7: Huvudmenyn i desktopvy.



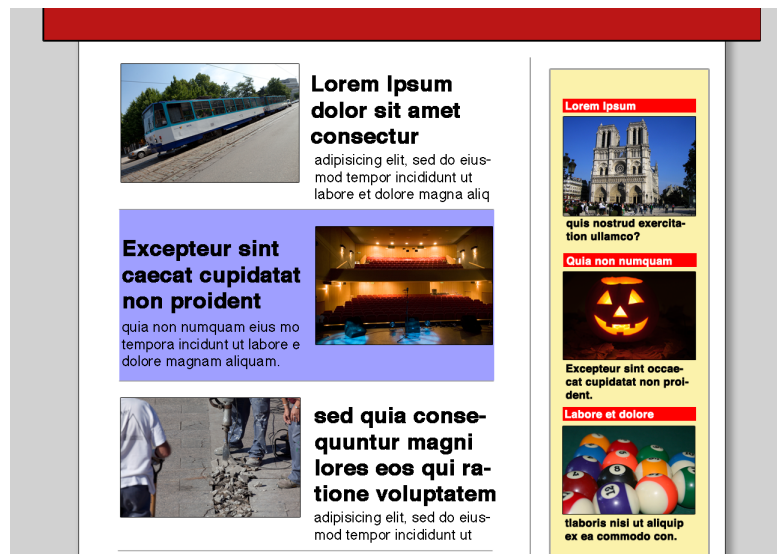
Figur 8: Huvudmenyn i mobilvyn.



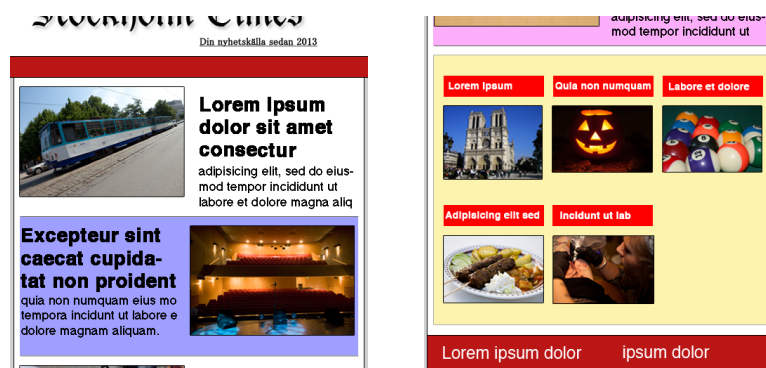
Figur 9: Nedremenyn i desktopvyn och mobilvyn.

5.2.1.3 Artiklar och Annonser

För artiklarna och annonserna, prioriterades dessa ungefär lika högt i desktopvyn (*figur 10*), mestadels för att annonser anser vara en viktig faktor till nyhetssidors intäkter. Dessvärre inte lika hög prioriterat i mobilläge då det finns mindre yta och artiklarna är huvudelementen i nyhetssidan, därför hamnar annonserna efter artiklarna (*figur 11*). Artiklarna har alla samma storlek men skiftar i plats på bild och text, för att skapa en mer levande känsla när man ser webbsidan. Annonsernas design härstammar från aftenbladet.se, där färgen på bakgrunden samt texten får läsaren att förstå att det är annonser.



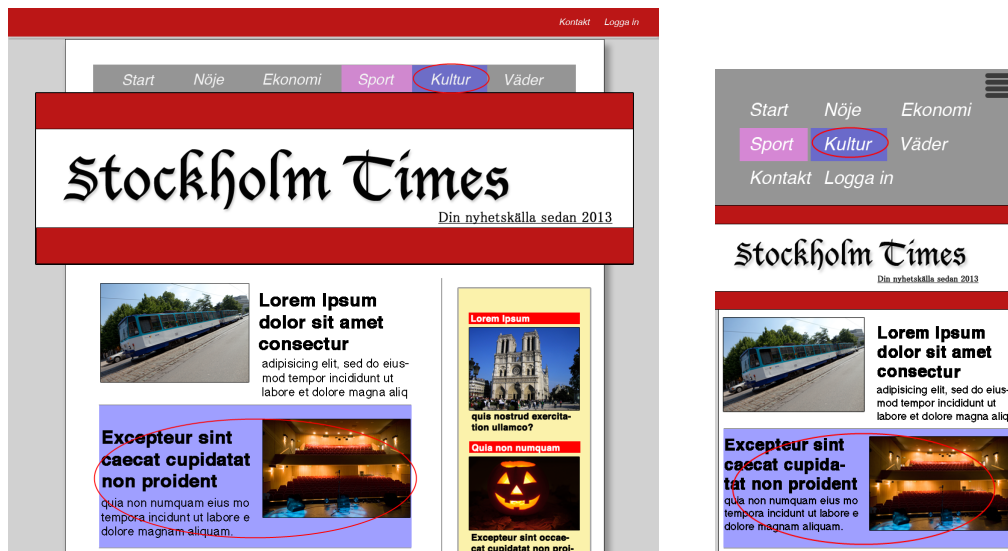
Figur 10: Artiklar och annonser i desktopvyn.



Figur 11: Artiklar och annonser i mobiltvyn.

5.2.1.4 Färgmarkeringar

Färgmarkeringen görs i webbsidan för att markera olika typer av artiklar, i nyhetssidor görs detta för att markera artiklar som tillhör en annan sektion av tidningen, t.ex sport, ekonomi, kultur osv. I prototypen markeras sport och kultur med en rosa och blå färg vilken även blir färgen för respektive länk (figur 12).



Figur 12: Blå färgmarkering för kultursektionen.

5.2.2 Responstid

Nedan visas tre olika hämtningar av bägge webbsidor, både i desktopmiljö samt mobilmiljö. Värdena som har antecknats är responstiden för hämtning av HTML-filen samt CSS-filen. Sambandet mellan de olika hämtningar visar att mobilvyn laddas snabbare i mobile-first och desktopvyn snabbare i desktop-first. I en av fallen tog hämtningen av CSS filen 19ms i mobilmiljön medan det tog desktop-first 36ms. I desktopmiljö gjordes hämtningen av CSS-filen 21ms för desktop-first och 23ms för mobile-first. Html filen varierar en del, men två av tre hämtningarna visar ungefär samma resultat som hämtningar för CSS.

Hämtning 1

	.html mobilvy	.css mobilvy	.html desktopvy	.css desktopvy
Desktop-first	73ms	36ms	36ms	21ms
Mobile-first	36ms	19ms	39ms	23ms

Tabell 5: Värden från första hämtningen av prototypsidan.

Hämtning 2

	.html mobilvy	.css mobilvy	.html desktopvy	.css desktopvy
Desktop-first	19ms	22ms	46ms	20ms
Mobile-first	33ms	20ms	22ms	22ms

Tabell 6: Värden från andra hämtningen av prototypsidan.

Hämtning 3

	.html mobilvy	.css mobilvy	.html desktopvy	.css desktopvy
Desktop-first	37ms	22ms	20ms	19ms
Mobile-first	16ms	18ms	32ms	20ms

Tabell 7: Värden för tredje hämtningen av prototypsidan.

5.2.3 Kodstorlek

Tabellen visar storleken på css-filen i både antal rader och filstorlek. Den visar att CSSen för mobil-first är större än CSSen för desktop-first i filstorlek. Däremot har den i desktop-first fler rader än den i mobile-first. I mobile-first CSSen är uppdelningen mellan rader i grundkod och rader i media queries 193 mot 180. I den för desktop-first består största delen utav rader i grundkoden, där är uppdelningen 221 mot 160.

main.css	Desktop-first	Mobile-first
Antal rader i grundkod	221st	193st
Antal rader i media queries	160st	180st
Totalt antal rader	381st	374st
Filstorlek	5.2kb	5.6kb

Tabell 8: Värden från main.css.

5.2.4 Implementeringstid

Implementeringstiden redovisas utifrån dagboken som skrevs under implementationsfasen. Resultatet visar skillnaden mellan implementeringstiden för grundsidan och tiden som krävdes för att göra webbsidan responsiv med hjälp av media queries. I mobile-first tog grundsidan mindre tid att implementera än att göra webbsidan responsiv, i desktop-first tog det däremot längre tid att implementera grundsidan. Desktop-first tog totalt längre tid att implementera än mobile-first. Ta i beaktande att implementeringen gjordes utav en junior inom området gränssnittsutveckling, samt att desktop-first implementerades före mobile-first.

	Desktop-first	Mobile-first
Tid för grundsidan	14.5h	6.5h
Tid för responsive	7.5h	10h
Totalt tid	22h	16.5h

Tabell 9: Värden på implementationstid, tagna från dagboken.

5.2.5 Kommentarer från dagboken

Desktop-first

"Mycket småpill i början, många element att finlipa, göra fina och sättas på plats."

"Oerhört lätt att göras till responsive så länge man har använt sig av fluid grid och fluid layout från början. Allt renderas av sig självt och så fort det ser konstigt ut, modifieras CSSen i media queries."

"Lång tid i början, i jämförelse med responsive delen, förvånad över att responsivedelen gick så smidigt. Endast lite småpill lite här och där för att. Det mesta hade fixats under desktoputseendet."

"Inte så många media queries, sidan ser bra ut mellan media queries. Mycket kod i grunden som är avsedd för desktop, inte lika mycket i media queries."

Mobile-first

"Enkelt sätt att påbörja webbsidan, följer skissen uppifrån neråt, går inte att tappat bort sig åt kanterna utan elementen är så pass små att det känns som att de finns i en smal led att följa."

"Behövs mycket småpill mellan media queries, media queries kommer i början hyfsat tät"

inpå varandra enbart för att fixa små fel som margins och padding så att elementen håller sig på ett och samma ställe. Samt för förstoring av text, som är väldigt liten i början och måste förstöras i media queries då det inte kan använda sig utav fluid grid.”

”Gäller att hitta en struktur redan från början mellan skisserna, blir några refaktoreringar för att få webbsidan, då bredden är 768px, att se perfekt ut, eftersom mycket ändras just under den vyn.”

”Från 768px till desktop-size gick väldigt smidigt.”

”Väldigt lätt i början, enkel struktur att lägga upp för mobilen, responsivt höll till 480px och till och med 768px var det mycket att fixa med hjälp av media queries, då många element går från att vara under varandra till att vara bredvid varandra.”

”Viktigt att planera och lägga en struktur på hur den ska övergå till desktop. Mycket kod i media queries sektionen, lite kod i grunden för mobilvyn. Ser bra ut mellan mobilvyn och desktopvyn.”

6 Diskussion

6.1 Mobile- vs Desktop-first

Mobile-first kontra desktop-first har varit den stora frågan i arbetet, efter litteraturstudien och implementationen har tydligheten i metodernas områden blivit allt klarare och har under arbetets gång även blivit mer populärt bland bloggar. I resultatet för litteraturstudien samlades synpunkter från 13 olika källor, vilka inläggsdatum för de flesta inte är mer än ett år gamla. Av dessa källor är det 11 som förespråkar för mobile-first, en som föredrar desktop-first och en som anser att tekniken är för ny för att förorda (inlägg från 2011). Implementationsfasen bekräftar mycket av det som har sagts i litteraturstudien, både till fördel för desktop-first och mobile-first. Men i slutändan har implementationen utav webbsidorna visat sig vara kodmässigt relativt lika, mobile-first använder sig utav min-width, medan desktop-first använder sig utav max-width, mobile-first ser till att rendera om elementen när skärmen blir större och desktop-first när skärmen blir mindre. Mycket fokus har lagts på metoderna för sig, när ett större problem visar sig vara skillnaden mellan mobilen och desktopens kapacitet. Mobiler har sämre prestanda, mindre skärm och bandbreddsbegränsningar, vilket inte existerar i samma utsträckning hos datorer. Dagens mobiler har kommit långt i utvecklingen men kan inte mäta sig med datorn. Det visade sig att arbetet kom att handla om vilken metod **passar** bäst i dagens teknik och inte vilken metod som **är** bäst. Dagens internet användning utgörs mestadels ifrån mobiler och datorer, där mobilen är den enhet som presterar sämre.

Mobile-first tar sig an problem som mobilen har och ser till att det utför det simplaste och läser in så lite som möjligt av webbsidan, med andra ord endast den kod som är avsedd för mobilen. Det finns ingen metod som tillför samma sorts fördel för båda enheter, utan mobile-first kommer att behöva läsa in mer när webbsidan ses ifrån skrivbordskärmen då den läser både allt för mobil och även det som skall ändras för desktop. Desktop-first läser in det nödvändigaste för skrivbordskärmen, men när webbsidan ses ifrån en mobil, kommer den läsa in allt för desktop och sedan allt som den behöver ändra för att webbsidan skall passa mobilen. Detta betyder att mobil-first kommer se till att mindre läses in för den enhet vars prestanda är låg och mer läses från den enhet vars prestanda är hög och har kapaciteten att hantera det.

I litteraturstudien nämndes att mobile-first bidrar till en robust grund för komplikationer som prestanda, design, och prioritering av element. Prestanda visas nedan i diskussionen genom responstid och kodanalys, men design och prioritering är lite svårare att visa. I arbetet designades både mobilvyn och desktopvyn samtidigt, sida vid sida och vid ett tidigt skede bestämdes att båda vyerna skulle innehålla samma funktioner. Med anledning att användare skall ha samma möjligheter vare sig enhet webbsidan ses ifrån. I litteraturstudien ses detta som ett tillvägagångssätt som existerar i mobile-first där meningen istället översätts till *"Får inte någonting plats i mobilvyn, så ska den inte heller få finnas i desktopvyn"*. Detta används för att undvika att funktionalitet endast finns på en av enheterna. Anledningen till att det är ett mobile-first princip är för att det är hos mobilen då element kan ha svårigheter att få plats, på grund av den begränsade skärmstorleken. Prioriteringen i mobilvyn sker då vid ett tidigt skede och allt som får plats i mobilskärmen kommer att få plats på en större skärm. I desktop-first gäller inte samma princip då element i en större skärm inte alltid får plats i en mindre yta. Det leder till att prioriteringen kommer i efterhand, och sker endast för mobilvyn, vilket kommer sakna funktionalitet som finns för desktopvyn och där mobilvyn istället blir en komplement till desktopsidan. En användare som tidigare har varit på desktopsidan kommer att sakna dessa funktioner i mobilvyn och då har webbsidan plötsligt tappat sitt syfte för en mobilanvändare. I arbetet dök aldrig problemet upp för att principen *öm inte ett element fick plats i mobil designen så fick den inte finnas i desktopdesignen* följdes från start.

Respons tid, kodanalys, implementerings tid och skärmstorlek är en av de större faktorer som stärker de påståenden taget från litteraturstudien, nedan beskrivs och analyseras dessa i synnerhet till de praktiska och teoretiska tillvägagångssätt.

6.1.1 Responstid

Responstid är den tid det tar mellan att en begäran har lagts, till dess att begäran har utförts, det vill säga i webbsammanhang den tid det tar att hämta och visa elementen på webbsidan. De tre hämtningar visar klart att när webbsidan ses utifrån ett mobilperspektiv hämtas css-värden för att rita ut elementen snabbare med mobile-first än med desktop-first (tabell 5, 6, 7). När webbsidan ses utifrån ett desktopperspektiv hämtas css värdena

snabbare med desktop-first än med mobile-first. Det vill säga, mobil-first ger bra siffror för hämtningar från mobilvyn och desktop-first bra siffror för hämtningar från desktopvyn, båda lika bra i sitt område om man skulle vilja jämföra metoderna strikt gentemot varandra. Att metoderna är bäst för var sin vy är förståeligt eftersom mobile-first i mobilvyn endast läser grundkoden utan att behöva ändra något värde i media queries, desktop fungerar bra i desktopvyn för att den endast läser grundkoden för desktopvyn. Men som tidigare poängterats i diskussionen handlar det inte om metoderna var för sig utan enheterna webbsidan ses från, där responstiderna för mobile-first gynnar den enhet med sämre prestanda vilket desktop-first inte gör. Båda responstiderna för mobil-first och desktop-first är hämtade från en dator där en emulator som minskar webbläsaren till en iPhones storlek har använts. Skillnaden mellan responstiderna är väldigt låga, ibland endast 2 millisekunder, för att prestandan och bandbredden som har använts för att hämta webbsidorna är ifrån en dator.

Ändå visar responstiden på en skillnad, men i verkligheten är inte omständigheterna likadana som i testfallen. En desktop-first sida som ses i mobilvyn tog enligt en av hämtningar 36ms och desktopvyn 21ms. 21ms är relativt snabbt, vilket även 36ms är. När desktop-first sidan ses på en skrivbordskärm kommer hämtningen av css-en ta ungefär 21ms, men när vi ser en desktop-first sida på mobilen kommer den inte nödvändigtvis att ta 36ms som det står i tabellen (*tabell 5*). Detta för att mobilen inte har samma prestanda och bredbandsmöjligheter som datorn som användes för hämtningar. Skillnaden mellan desktop-first och mobile-first blir inte längre 2ms, utan måste multipliceras med faktorerna, sämre prestanda och bredband. Vilket leder till att skillnaden blir mellan metoderna blir oerhört större än vad som har visats i tabellen för hämtningarna av css-filen. Mobile-first visar genom tabellerna (*tabell 5 ,6 ,7*) att enheter där webbsidan ses i mobilvy inte behöva ladda mer än vad som behövs för att visa webbsidan och låter de enheter med bättre omständigheter ta på sig att hämta mer än nödvändigt.

Förutom att hämtningarna är gjorda med en dator så måste prototypen ha i åtanke då den egentligen bara speglar skalet av en fulländad webbsida. Det vill säga att hämtningarna som görs är för en webbsida som är relativt liten i jämförelse med en normal webbsida. Återigen är dessa 2ms i skillnad mellan metoderna i en hämtning av mobilvyn små i jämförelse med vad skillnaden skulle vara om prototypen hade varit för ett större projekt.

Resultat av responstiden visar tydligt de punkter som diskuterats i litteraturstudien. Mobile-first ger en robust webbsida för mobilen som fokuserar i första hand på situationer hos användaren där snabb respons krävs. Tack vare prestandan hos dagens datorer utelämnar inte mobile-first desktop, utan låter den snarare ta hand om den större lasten, vilken den har kapacitet för.

6.1.2 Kodanalys

6.1.2.1 Storlek

Kodstorleken spelar en stor roll till resultatet av responstiden, beroende på metod läser webbsidorna olika mycket av css-koden för att renderera sidan beroende på webbläsarens bredd. Main.css består utav grundkoden, där elementen definieras och får de egenskaper för att ritas ut på webbläsaren och media queries koden, där dessa värden ändras för att anpassas efter webbläsarfönstret. Totalt antal rader är ungefär samma mellan desktop-first och mobile-first (*tabell 8*). Det som skiljer sig är antal rader av kod som metoderna har i grunden och i media queries. I mobile-first är grundkoden avsedd för mobilen, vilket betyder att den endast läser grundkoden när webbsidan ses i mobilvy, vilket är 193 rader. I desktop-first är grundkoden avsedd för desktop och läser endast in den vid desktopvy, vilket består av 221 rader. Om det kommer sig att webbsidan ses utifrån den vy vars metoden inte är avsedd för, betyder det att den måste läsa in all grundkod och sedan koden från media queries för att ändra egenskaperna för varje element. I desktop-first måste mobilwebbläsaren läsa totalt 381 rader för att kunna visa webbsidan, i mobile-first måste desktopwebbläsaren läsa totalt 374 rader för att kunna visa webbsidan. Båda metoder har sina fördelar och inriktar sig för att fungera bäst för var sin enhet. Men återigen måste enheterna ha i åtanke. Mobilen har inte samma prestanda som en dator, antingen läses det in få rader kod i mobilen och många rader kod i desktop, eller många rader i mobilen och få rader i desktop. Det visar fördelarna som diskuterades i litteraturstudien, vilka innebär att låta mobilen läsa in så lite som möjligt på grund av prestandaskäl och datorn mer än nödvändigt då den har kapaciteten att kunna hantera det.

6.1.2.2 komplexitet

Förutom olika mycket kod i media queries, skiljer sig metodernas main.css till att få mobile-first agera som ett mer naturligt sätt att implementera. I Mobile-first är grundkoden avsedd för mobilen där element ligger i en kolumn under varandra. Eftersom det är standard att element visas under varandra behövs ingen kod för detta, utan kod läggs istället till i media queries när webbsidan börjar övergå till desktop och element skall visas vid sidan om varandra. I desktop-first måste elementen redan från början definieras så att de skall ligga bredvid varandra då detta inte är standard. När webbsidan börjar övergå till mobilläge och webbläsaren blir mindre, måste i media queries se till att det blir gjort så att elementen hamnar under varandra, i en kolumn. Med andra ord kräver detta en rad kod i mobil-first som ändrar elementens position när den syns i desktop (*figur 13*), men två rader kod i desktop-first, en som ändrar elementens position från början och en som ser till att det återgår till standard när sidan syns i mobilen (*figur 14*). Det är endast fåtal element i prototypen där detta förekommer, men det tyder på att fallet existerar fler gånger på större webbsidor, speciellt nyhetssidor, där många element går från att vara vid sidan om varandra till att vara under varandra i en kolumn. Det medför att mobilsida blir ett naturligt sätt att påbörja en implementation av en responsiv webbplats, då det börjas från standard.

<pre> . articles { color: white; width: 200px; height: 200px; } @media (min-width: 768px) { float: left; } </pre>	<pre> . articles { float: left; color: white; width: 200px; height: 200px; } @media (max-width: 768px) { float: none; } </pre>
(a) Mobile-first	(b) Desktop-first

Figur 13: I Desktop-first(b) måste elementet först definieras så att de ligger vid sidan om varandra, sedan med media queries tas detta bort när det går över till mobilvyn. I Mobile-first(a) är det standard och inget behöver definieras förens webbsidan övergår till desktopvy, då sätts elementen vid sidan om varandra.

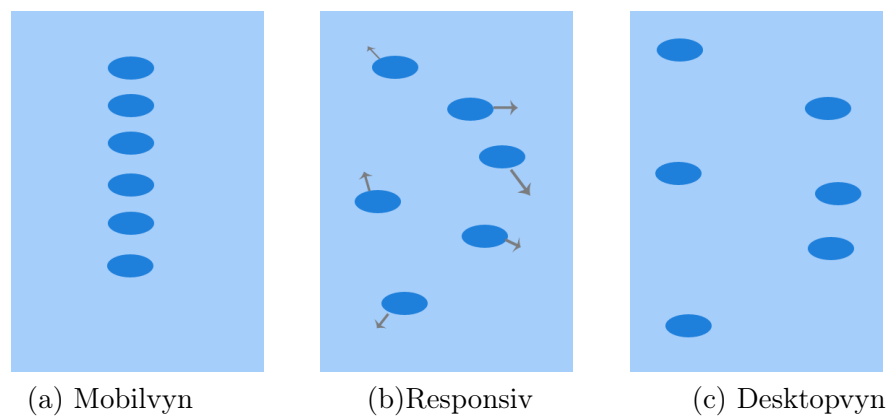
6.1.3 Implementeringstid

Implementeringstiden för webbsidorna skiljer sig med ungefär med 5 timmar, mobil-first tog totalt 16,5 timmar och desktop-first 22h. Det är svårt att dra slutsatsen att mobile-first tar mindre tid att implementera än desktop-first, på grund av att: 1. Desktop-first implementerades före mobile-first, för en junior gränssnittsutvecklare, kan desktop-first implementeringen varit en bra uppvärmning, 2. För att erfarenheten utav utvecklaren inte är tillräckligt stor, vilket kan medföra inläring inräknat i tiden. En teori kan ändå skapas utav kommentarerna ur dagboken och även tiden som redovisades under implementationsfasen

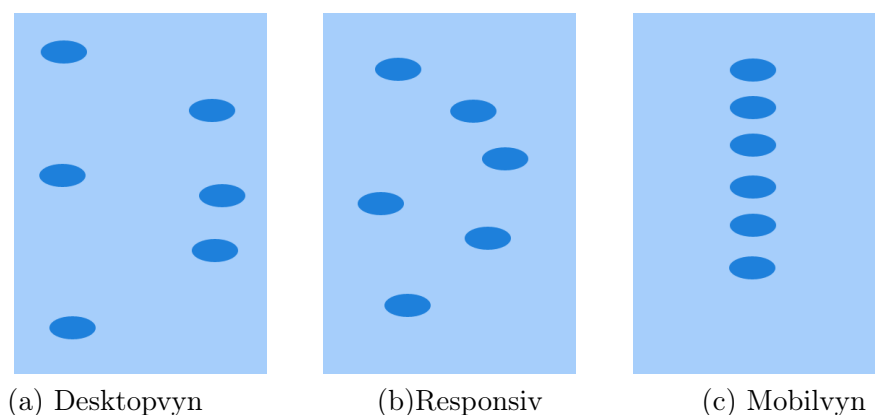
Grundsidan för desktop-first tog längst tid att implementera, vilket enligt dagboken krävde många element att dekoreras och positioneras utspritt på en stor yta. Saknas erfarenhet inom området, kan det vara svårt att finna en struktur på hur en webbsida skall formas. Grundsidan för mobile-first tog minst tid, vilket i dagboken beskrivs som ett enkelt sätt att forma en webbsida på. Innehållet är i en kolumn där element implementeras uppifrån neråt utan att kunna tappa bort sig åt sidorna eller ändra något som påverkar andra element. Att göra webbsidan responsiv tog mindre tid för desktop-first än mobile-first. I desktop-first har elementen dekorerats och lagts ut på rätt plats, det enda som behövs göras är att minska elementen och samla ihop dessa till en kolumn, vilket fluid grid, fluid images och media queries tar hand om utan att gränssnittsutvecklaren behöver justera för mycket. I mobile-first börjar elementen i en kolumn, det som skall göras är att låta fluid grid, fluid images göra så att dessa förstoras. I mobile-first krävs det däremot fler justeringar i media queries då elementen skall positioneras till sina utspridda platser på desktopvyn istället för en kolumn där elementen finns under varandra. Det kan vara anledningen till att responsivdelen för

mobile-first tog längre tid än responsivdelen i desktop-first.

Mobile-first har fördelen att grundsidan implementeras på ett simplare sätt än i desktop-first. Istället för att rita ut alla element utspridda på en sida, dekoreras detta i en kolumn och låter den responsiva delen ta hand så att elementen hamnar på rätt plats i desktopvyn, med ett antal justeringar med media queries. I desktop-first, sprids och dekoreras elementen för att sen låta den responsiv delen samla in dessa i en kolumn, men då har redan mycket tid lagts åt att rita och positionera ut elementen i desktopvyn. Teoring säger att mobile-first är ett mer effektivt tillvägagångssätt, men att om behovet istället är en snabb lösning till att göra en existerande webbsida responsivt, finns fördelen tidsmässigt hos desktop-first då grunden redan är lagd.



Figur 14: Responsiv i Mobile-first, elementen börjar designas och placeras i en kolumn på skärmen, responsive ser till att med en del justeringar placera dessa rätt i desktopvyn.



Figur 15: Responsiv i Desktop-first, elementen måste designas och placeras rätt på skärmen, responsiv ser till att de samlas en kolumn för mobilvyn.

6.1.4 Skärmstorlek

Storleken på mobilskärmen har i litteraturstudien setts som ett hinder, en barriär som kräver fokus där en design tas fram efter en prioritering av de nödvändigaste funktionerna. Under implementeringen var skärmstorleken en lättnad, en kompakt design som ledde till struktur i arbetssättet. Prototypen av webbsidan var en förenklad version utav riktiga nyhetssidor, därför är det inte lika lätt att förstå sig på prioriteringssvårigheten som kan uppkomma på större webbsidor.

Prototypen innehöll inte lika många element som en verklig nyhetssida, vilket gör det svårt att relatera till prioriteringssvårigheten som webbutvecklare i litteraturstudien har beskrivit. I desktop-first är skärmstorleken stor, där prioritering av funktionalitet inte har samma betydelse som i mobile-first, då det mesta får plats. Däremot visar detta återigen risken att för att mobilvyn istället blir en förenklad och nerskalad version utav desktop-sidan. Skärmstorleken bör ses som ett hjälpmedel istället för ett hinder. Ett hjälpmedel för utvecklaren och interaktionsdesignen som försäkrar att funktionalitet inte försvinner mellan vyer och samtidigt skapar en bra grund, för både arbetssätt och användbarhet. Användbarhet har eskalerat inom webbutveckling, vilket av egen erfarenhet anses vara högt prioriterande. I litteraturstudien delas denna åsikt med härledning till att webben börja ses i ett annat perspektiv, där funktionalitet och användbarhet uppskattas mer än mäktig design som kan kräva mycket prestanda, speciellt för en mobil. Desktop-first är inte nödvändigtvis ett hinder för att skapa en användarvänlig webbsida, men mobil-first är en försäkran att detta upprätthålls, tack vare mobilens kompakta skärmstorlek som kräver fokus på innehåll och funktionalitet.

6.2 För- och Nackdelar

För- och nackdelarna som diskuteras fram är dessa som både speglar det som har samlats i litteraturstudien och av implementeringsfasen, vilket betyder att fördelar så som att endast finns i litteraturstudien men som inte kan stärkas utav implementeringen av prototypen inte tas med i detta avsnitt, vare sig det är en fördel eller nackdel.

Fördelarna med mobile-first kan sammanfattas i tre punkter, **Fokus på innehåll och funktionalitet, anpassad till dagens teknik och ett naturligt arbetssätt.**

Innehåll och funktionalitet i fokus betyder att dessa prioriteras framför .t.ex. presentation och animation. Prioriteringen sker just för att grunden avser mobilsidan vilket innebär en begränsad yta att designa för. Detta tvingar fram bra användbarhet då betydelsen för hur lättsamt en användare kan förstå och navigera på webbsidan prioriteras högt. Med ett fokus på innehåll och funktionalitet undviks skapandet av webbsidor som blir subventionerade versioner beroende på vy, det vill säga, funktioner och innehåll som tas bort på grund av platsbrist. Eftersom som innehållet och funktionaliteten är i centrum skapar det även en robust grund där allt annat byggs utifrån den.

Anpassning för dagens teknik är en viktig punkt som visades tydligt utav resultatet från responstiden, kodanalysen och som även förstärks utav litteraturstudien. Tekniken utvecklas ständigt och möjligheten för mobiler att hinna ikapp datorerna i prestanda existerar, men tekniken är inte där än. Mobile-first har det i beaktande då metoden är anpassade för dagens teknik där mobilen har sämre prestanda än datorn. Mobile-first läser in mindre kod i mobilvyn, därav snabbare responstid för webbsidor som ses ifrån en mobilskärm. Användningen utav mobilt internet ökar för varje dag där användarna kräver snabb respons vid flexibla situationer där en tidspress existerar. Mobile-first är anpassad för de prestandasvårigheter som existerar hos mobiler, men som inte är lika kritiska för datorer.

Mobile-first tillför ett **naturligt arbetssätt**, här designas elementen på ett sätt som är standard först och som sedan ändras desto större skärmstorleken blir. Positionerna för elementen som formar webbsidan behöver inte definieras då dessa redan befinner sig en kolumn. Under implementationensfasen påpekades fördelen med att börja utveckla efter en design där alla element är under varandra. Det medför en struktur som är lätt att hålla sig efter och där fokus finns för att ta sig an de svåra problemen först, därav en robust grund att bygga på. Arbetssättet som mobile-first medför, påminner mycket om det agila och kan därför fungera väldigt bra tillsammans med agila metoder. Element prioriteras och det arbetas efter en grund där det viktigaste är i centrum och där resterande funktioner byggs utifrån den.

De flesta nackdelar som togs upp i litteraturstudien är inte relevanta till implementationsfasen, det är svårt att stärka ett påstående som bygger på känslor snarare än något tekniskt t.ex *"det kan uppfattas som tråkigt och jobbigt eftersom det utmaningarna möts i början"*. Däremot uppmärksammades det att media queries inte fungerar för internet explorer 6,7 och 8. Prototypen valdes därefter att inte vara avsedd för internet explorer 8, vilket tyder på att det är ett stort problem och en nackdel för mobile-first. Internet explorer 8 används fortfarande i en mindre utsträckning och så länge windows xp finns installerat som operativsystem kommer även möjligheten för internet explorer 8 att finnas. Lösningen är inte nödvändigtvis att övergå till desktop-first då detta medför nackdelarna desktop-first har för mobilen, men att istället använda en annan css för denna webbläsare. Det kommer att medföra en del omständigheter för webbutvecklaren som måste skriva en separat css-fil, men kan vara värt med tanke på de fördelar som mobile-first kan bidra med. Förutom media queries existerar andra tekniker som internet explorer 8 inte kan hantera och finns därför risken att webbsidan måste modifieras utifrån webbläsaren oavsett vilken metod.

Fördelen med desktop-first är att det är ett simpelt sätt att övergå till responsivt från en existerande webbsida. Det kan komma till användning om man snabbt vill visa en prototyp av en responsiv webbplats eller vill förbättra något litet när webbsidan ses i smalare vyer. Däremot är den inte lika optimal som mobile-first vid skapandet av större responsiva webbsidor. Fördelen beror på att grundsidan och elementen redan är designade och rätt positionerade, med hjälp av media queries går dessa element att enkelt anpassas till mindre vyer. Nackdelen blir däremot att mobilvyn har tendensen att bli en nerskalning av desktopsidan, allt innehåll från en desktopvy kan vara svårt att få in i en mobilvy och funktionalitet får istället prioriteras bort. Mycket kod som inte används läses även in när

webbsidan ses i mobilvy, vilket leder till att om prestandaproblem åtgärdas så sker det i efterhand. Mycket av fördelarna hos mobile-first blir strikt nackdelar hos desktop-first då desktop-first fokuserar på en optimal responsiv webbplats för desktop istället för mobilen. Vilket till skillnad från mobilen är en enhet med bra prestanda och en skärmyta som inte innebär samma begränsningar.

6.3 När ska Desktop-first respektive Mobile-first användas

I resultatet beskrivs användningsområdet vara väldigt stor och utspritt för mobilt internet, vilket två källor från litteraturstudien beskriver som en viktig aspekt att analysera när metod väljs. I analysen av användningsgruppen har målgruppen mellan 12-45 år visats sig ha mer än 50 % användare av mobilt internet dagligen. I målgruppen 45-55 år är användningen utav mobilt internet 30 % dagligen. I resultatet påpekas även det felaktiga antagandet att mobilt internet endast används vid rörliga situationer, vilket istället visar att större delen av användare brukar mobilt internet vid stationära situationer t.ex hemma eller på kontoret. Gällande kontext av webbsida har de fyra större områdena har visat sig vara, hantering utav e-post, sociala nätverk, söka information och kolla nyheter, vädret. Dessa fyra områden har växt markant inom mobilt internet och sannolikheten att dessa överkommer desktop visar sig vara stora. De områden däremot där mobilt internet inte har lyckats nå upp till desktop och där siffran är väldigt låg jämfört med desktop är "fylla enkäter" och "handla produkter". Det är väldigt svårt av resultatet att se varför det är så och ifall en förändring kommer att ske. Det som statistik visar däremot är att mobilen oftast används för att leta upp produkter innan köpet sker från desktop, vilket betyder att om webbsidan ändå ses ofta från en mobil, så skall även möjligheten att kunna köpa produkten finnas, och med en robust och säker mobilsida kanske detta till och med sker. Mobilt internet ökar så pass mycket i användning i alla områden för målgrupp, kontext på webbsida och miljö, vilket gör att det är svårt att hitta ett fall där mobilsidan har möjligheten att vara en nerskalad version utav desktopsidan, där prestandan inte är lika viktig när webbsidan ses från mobilen och där mobilsidan blir en bonus för desktopsidan.

Detta leder till att mobile-first bör användas i de flesta fallen, speciellt för webbsidor som avser stora och informationsrika webbplatser, vilket då kan vara en stor fördel för att främja mobilens prestanda. Det är även ett effektivt sätt att arbeta med CSS-filen, det ger en bra struktur för arbetet och försäkrar att webbsidornas vyer uppfyller samma kriterium. Användningen av mobilt internet är så pass stor och fördelat i så många områden att det är svårt att hitta ett scenario som tillåter en sämre mobilsida där information och funktionalitet inte nödvändigtvis behöver vara på samma nivå som den på desktopsidan. Mobile-first är motsatsen, det är en försäkran på att dessa nackdelar inte förekommer och att webbsidan uppfyller sitt syfte i både mobilvyn och desktopvyn, vilket visar sig i litteraturstudien för användningsområden att vara högst prioriterande.

Desktop-first är däremot en snabb lösning till en responsiv webbplats utifrån en existerande webbsida. Den existerande webbsidan har redan designats och med hjälp av media queries

modifieras elementen till att förminskas och passa in i en mobilvy. Områdena där detta skulle hjälpa är vid skapandet av en prototyp för en responsiv webbplats och hur denna skulle tänkas se ut utifrån ett responsivt perspektiv. Prototypen tillför funktionalitet vilket en vanlig designsketch inte kan tillföra. Däremot är det viktigt att förstå att prototypen inte kan tillföra den robusta grund att fortsätta implementera på, då grunden är avsedd för en desktopsida och inte en responsiv webbsida. Desktop-first kan även ses som ett bra alternativ när webbsidan skall uppfylla funktioner som helst görs i ett stationärt perspektiv och inte kräver samma användbarhet och funktionalitet från en mobilvy, men att hitta sådana området är i nuläget väldigt svårt.

Ett alternativ är att webbsidan in nödvändigtvis har samma funktionalitet på mobilen som på desktop, t.ex att utföra administrativa åtgärder från desktop, men endast kunna se ändringarna från mobilen. Med tanke på hur mobilt internet ökar i alla användningsområden tyder detta på att en mobilanvändare kommer att spendera mycket av tiden på internet via mobilen och därmed vilja ha samma möjligheter hos mobilen som på desktop, vilket därför inte är ett tillvägagångssätt som gynnar dagens webbgränssnitt. Istället blir mobile-first det tillvägagångssätt som i längden ger ett bättre resultat.

7 Slutsats

Referenser

- Bull, Matthew (2012): *Mobile comes first: responsive web design in a mobile world*. Senast besökt: 11 juli 2013. URL: <http://blogs.kent.ac.uk/webdev/2012/07/22/mobile-comes-first-responsive-web-design-in-a-mobile-world/>.
- Chan, Michael (2013): *The better approach to rwd: Mobile first design*. Senast besökt: 11 juli 2013. URL: <http://www.readyartwork.com/2013/05/15/mobile-first-design/>.
- Chapman, Katy (2012): *Interview by Edd Gold, Why design for mobile before desktop?* Senast besökt: 11 juli 2013. URL: <http://www.othermedia.com/blog/why-design-for-mobile-before-desktop,424,TA.html>.
- Church, Karen och Nuria Oliver (sept. 2011): "Understanding Mobile Web and Mobile Search Use in Today's Dynamic Mobile Landscape". I: *Telefonica Research*.
- Clark, Barry (2009): *Adaptive, Responsive or Mobile First*. Senast besökt: 14 maj 2013. URL: <http://barryclark.co/adaptive-responsive-or-mobile-first/>.
- Dimestam, Andreas (2012): *Acer släpper multi touch-skärmar för windows 8*. Senast besökt: 11 juli 2013. URL: <http://www.sweclockers.com/nyhet/16010-acer-slapper-multi-touch-skarmar-for-windows-8>.
- Fidler, Roger (2013): *2013 Q1 Research Report 1: News consumption on mobile media surpassing desktop computers and newspapers*. Senast besökt: 12 juli 2013. URL: <http://www.rjionline.org/research/rji-dpa-mobile-media-project/2013-q1-research-report-1>.
- Findahl, Olle (2012): *Svenskarna och Internet 2012*. .se. ISBN: 978-91-979411-7-4.
- Foster, Aidan (2012): *Mobile First: A key ingredient in great responsive design*. Senast besökt: 11 juli 2013. URL: <http://responsivedesign.ca/blog/mobile-first-a-key-ingredient-in-great-responsive-design>.
- Fransén, Karin och Oscar Holmström (2012): *Prisutevecklingen på mobiltelefoni och bredband*. PTS. ISBN: PTTS-ER-2012:26.
- Grigsby, Jason (2011): *Where are the mobile first responsive web designs?* Senast besökt: 11 juli 2013. URL: <http://blog.cloudfour.com/where-are-the-mobile-first-responsive-web-designs/>.
- Gustaffson, Marcus (2012): *Telia sänker surfpriser i USA*. Senast besökt: 18 juni 2013. URL: <http://www.telekomidag.se/nyheter/artikel.php?id=38201>.
- Hansson, Daniel (2012): *Responsive websites - mobile vs. desktop first*. Senast besökt: 11 juli 2013. URL: <http://www.neocreo.net/en/articles/responsive-websites-mobile-vs-desktop-first>.
- Helmreich, Doug och Philip Doriot (aug. 2009): "Smartphones, Providers and the Customers Who Love (and Loathe) Them". I: *CFI Group Smartphone Satisfaction Study 2009*.
- Houx, James (2013): *Choosing between Mobile-first, Desktop-first or Independent hemes and Sites*. Senast besökt: 11 juli 2013. URL: <http://www.armstrongtheme.com/appendix/choosing-between-mobile-first-or-desktop-first/>.

- Ingmar, Sara (aug. 2012): "Responsive web design: En studie i hur designkonceptet påverkar användbarheten hos en webbplats". I: *Uppsala Universitet*.
- Johnson, Joshua (2013): *Mobile First Design: Why It's great and why it sucks*. Senast besökt: 11 juli 2013. URL: <http://designshack.net/articles/css/mobilefirst/>.
- Lande, Simon (2009): *Mobile First, Desktop Later*. Senast besökt: 15 maj 2013. URL: <http://www.cmswire.com/cms/customer-experience/mobile-first-desktop-later-019716.php>.
- Marcotte, Ethan (2011): *Responsive web design*. Jeffrey Zeldman. ISBN: 978-0-09844425-7-7.
- Meeker, Mary och Liang Wu (2009): *2012 Internet Trends*. Senast besökt: 14 maj 2013. URL: <http://www.kpcb.com/insights/2012-internet-trends>.
- Mellas, Caleb (2011): *Why should I design for mobile-first?* Senast besökt: 11 juli 2013. URL: <http://www.webinsation.com/why-should-i-design-for-mobile-first/>.
- Pope, Marcus (2012): *Mobile-first, kind of like the secret*. Senast besökt: 1 juli 2013. URL: <http://www.marcuspope.com/blog/mobile-first,-kind-of-like-the-secret>.
- quirksmode.org (2009): *Mobile compatibility tables*. Senast besökt: 14 maj 2013. URL: <http://quirksmode.org/m/table.html>.
- Rana, Pranay (2009): *Div and table*. Senast besökt: 14 maj 2013. URL: <http://www.codeproject.com/Articles/47934/DIV-TABLE>.
- Schmiedl, Grischa, Markys Siedl och Klaus Temper (sept. 2009): "Mobile Phone Web Browsing - A Study on Usage and Usability Of The Mobile Web". I: *University of Applied Science St. Pölten, Institute of Media Informatics*.
- Seitz, Patrick (2009): *Inventors - Apple's Iphone Turns 5 'At The Top Of Its Game'*. Senast besökt: 14 maj 2013. URL: <http://news.investors.com/technology/062712-616302-apple-iphone-turns-5-faces-rivals-from-samsung-google.htm>.
- Shannon-Missal, Larry (jan. 2013): "Different Priorities in Smartphone vs Computer Use, But Some Common Ground". I: *The Harris Poll*.
- Singhal, Rohit (2009): *Analyzing mobile vs Desktop user behavior*. Senast besökt: 14 maj 2013. URL: <http://www.designer-daily.com/analyzing-mobile-vs-desktop-user-behavior-29380>.
- Stanley, Morgan (2009): *Internet trends 2010*. Senast besökt: 14 maj 2013. URL: <http://www.slideshare.net/fred.zimny/morgan-staneley-internet-trends-ri041210>.
- w3schools (2009): *CSS How To...* Senast besökt: 14 maj 2013. URL: http://www.w3schools.com/css/css_howto.asp.
- Wroblewski, Luke (2010): *Mobile web design tips: Mobile should come first*. Senast besökt: 11 juli 2013. URL: <http://www.techradar.com/news/internet/mobile-web-design-tips-mobile-should-come-first-719677/1#articleContent>.
- (2011): *Mobile First*. Jeffrey Zeldman. ISBN: 978-937557-02-7.
- Zee, Theo van der (2012): *Responsive web design and mobile first*. Senast besökt: 11 juli 2013. URL: <http://www.themepartner.com/blog/100/responsive-web-design-and-mobile-first/>.
- zurbword (2012): *Mobile first – An adaptive, future-friendly solution for website design*. Senast besökt: 11 juli 2013. URL: <http://zurb.com/word/mobile-first>.