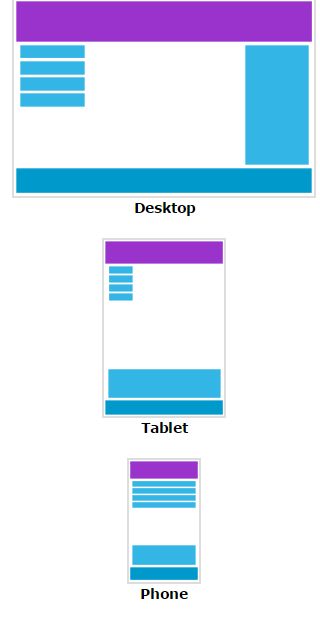
**Responderende utforming**

I dag har vi kommet til det punktet av vi ikke kan lage egne design for de forskjellige skjermstørrelsene på alle de ulike klientene. Det finnes rett og slett for mange klienter med forskjellig oppløsning. Responderende web-design er tilnærmingen som foreslår at design og utvikling skal respondere til brukerens behov og miljø basert på skjermstørrelse, plattform og orientering. Praksisen inneholder en miks av fleksible «grids» og «layouts», bilder og intelligent bruk av CCS.

Når en bruker bytter fra en bærbar datamaskin til ett nettbrett, så vil nettsiden automatisk bytte til den rette oppløsningen, bilde størrelse og «scripting abilities». Med andre ord skal websiden respondere til brukerens preferanser. Slik kan vi unngå å lage forskjellige design til forskjellige klienter.

Responderende design handler ikke bare om å rejustere skjerm-oppløsninger og automatisk rejustere bildestørrelser, men om en ny tenkemåte å tenke web-design på.

Nøkkelord:

CSS, media queries, HTML, Grids, layout,

**Universell utforming**

**Prinsipp 1: Mulig å oppfatte**

Informasjon og brukergrensesnittkomponenter må presenteres for brukere på måter som de kan oppfatte. Det vil si at informasjon som brukere skal kunne oppfatte, ikke kun lar seg oppfattes med en enkelt sans. WCAG krever at for tekst skal brukes som alternativ til andre medier som lyd, film og bilder. Slik at om en bruker for eksempel er blind eller døv skal kunne oppfatte innholdet.

**Prinsipp 2: Mulig å betjene**

Det må være mulig å betjene brukergrensesnittkomponenter og navigeringsfunksjoner. Web er interaktivt, og det er veldig viktig for oss at vårt web-baserte system kan navigeres i, ved å bruke det utstyret brukerne benytter. For eksempel skal det være mulig å navigere i web-systemet vårt ved å kun bruke tastaturet.

**Prinsipp 3: Forståelig**

All informasjon som er presentert i web-systemet skal være mulig å forstå, samt betjeningen av brukergrensesnittet. Når vi skal lage et web-basert system må vi sørge for at brukerne forstår både hvordan systemet brukes, samt forstå informasjonen de finner. Dette prinsippet handler om enkelt språk, god hjelpefunksjonalitet og forutsigbarhet. Dette prinsippet handler også om god kodeskikk, for eksempel ved at tekts blir lest opp på riktig måte for de brukerne som bruket talesyntese.

**Prinsipp 4: Robust**

Dette prinsippet handler om at innholdet i vårt web-system må være robust nok til at det kan tolkes på en pålitelig måte av en brukeragent (nettlesere, medieavspillere, tilleggsprogrammer og andre programmer inkludert kompenserende teknologi). Disse brukeragentene bidrar med henting av, presentasjon av og interaksjon med webinnhold. I praksis betyr dette at nettsider validerer og at koden er riktig. Dette blir som regel ivaretatt ved bruk av standardelementer i HTML.

5.6 Hoved aktiviteter i prosjektarbeidet

Aktivitetene presentert i tabellen nedenfor, vil ifølge planen starte 18.01.2017. Et gantt-diagram og prosjektnedbrytnings diagram blir inkludert i innleveringsmappen.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Aktivitet | Ansvar | Tidsperiode |
| A1 | **Prosjektplanlegging og oppsett av utviklingsmiljø** | **Prosjektgruppe** | **2 uke** |
| A1.1 | Forprosjekteringsrapport ferdig | Prosjektgruppe | N/A |
| A1.2 | Utarbeide en kravspesifikasjon | Prosjektgruppe | 2 dager |
| A1.3 | Design og skissering av database | Prosjektgruppe | 2 dager |
| A1.4 | Design og konfigurasjon av server | Prosjektgruppe | 2 dager |
| A1.5 | Valg og konfigurasjon av utviklingsmiljø | Prosjektgruppe | 1 dag |
| A1.6 | Innsamling av data til arkiv-systemet | Prosjektgruppe | 1 dag |
| A1.7 | Design og skissering av brukergrensesnitt til webapplikasjon | Prosjektgruppe | 2 dager |
| A2 | **Utvikling av databasesystem** | **Prosjektgruppe** | **5 uker** |
| A2.1 | Generere database med innhold | Prosjektgruppe | 1 uker |
| A2.2 | Initialisere og konfigurere server mot database | Prosjektgruppe | 2 uker |
| A2.3 | Lage funksjonalitet for innsetting/uthenting av data i database og server | Prosjektgruppe | 2 uker |
| A3 | **Utvikling av en fungerende prototype av webapplikasjon** | **Prosjektgruppe** | **5 uker** |
| A3.1 | Utvikle back-end for administrator | Prosjektgruppe | 1 uker |
| A3.2 | Utvikle brukergrensesnittet til applikasjonen | Prosjektgruppe | 2 uker |
| A3.3 | Utvikle funksjonalitet som håndterer samhandlingen mellom applikasjon og informasjonssystem | Prosjektgruppe | 2 uker |
| A4 | **Testing, oppgradering og debugging** | **Prosjektgruppe** | **2 uker** |
| A4.1 | Testing av webapplikasjon på flere plattformer | Prosjektgruppe | 1 uke |
| A4.2 | Analysering av kode | Prosjektgruppe | 1 uke |
| A5 | **Dokumentasjon: systemdokumentasjon og rapporter** | **Prosjektgruppe** | **1 uke** |

Dette bachelor-prosjektet vil ikke ha noen andre utgifter enn timene prosjektmedlemmene legger inn i prosjektet.

## Vurdering – analyse av risiko

Med risiko menes sannsynligheten for og konsekvensen av at noe uønsket skal skje eller utvikle seg. I dette avsnittet skal vi gjennomgå en risikoanalyse, før arbeidet på prosjektet starter for alvor. Dette gjør vi for å kunne finne eventuelle risikoelementer i prosjektet vårt, og iverksette passende tiltak for å øke kontrollen over forhold som må ivaretas under gjennomføringen av prosessen.

Etter en diskusjon i bachelorgruppen ser vi på det som sannsynlig at det er mulig å realisere prosjektet innenfor de rammene som er gitt, om kravene under blir møtt.

* Effektiv og konstruktiv kommunikasjon innad i prosjektgruppen og mellom prosjektgruppen og arbeidsgiver
* At forprosjektering og vurderinger underveis er gjort på en god måte
  + God planlegging gir god programvare
* Gode dokumentasjonsrutiner, ved å følge gitte dokumentasjonsstandarder.
* God programmeringsskikk og programmeringsstandarder.
* Grundige tester av applikasjonen, med påfølgende analyse av testene.
* At prosjektgruppen har valgt rett utviklingsmodell for programvare

Vi ser på det som særlig viktig at kommunikasjonen innad i gruppen er god, samt kommunikasjonen mellom prosjektgruppen og arbeidsgiver. Dette er fordi at vi tror at vi kan takle uforutsette hendelser og andre risikoelementer ved et godt samarbeid innad i prosjektgruppen. I tillegg er det særlig viktig med en tydelig kommunikasjon mellom prosjektgruppen og arbeidsgiver, siden vi antar at det er sannsynlig at arbeidsgiver kan endre på noen av sine ønsker for sluttproduktet.

Prosjektgruppen vurderer disse elementene som risikoelementer som kan utgjør en risiko for prosjektets-sluttprodukt. Disse elementene kan forlenge eller gjøre arbeidet med planlegging, gjennomføring og testing vanskeligere.

* For høy belastning på medlemmene i prosjektgruppen
  + Prosjektgruppen kan ta for seg for mange arbeidsoppgaver ved å utvide prosjektoppgaven, og dermed skape et større tidspress og en følelse av man ikke klarer å gjennomføre prosjektet på en tilstrekkelig måte.
* Støy og forstyrrelser i utviklingsmiljøet
  + Kan skape et dårlig utviklingsmiljø, og dermed en mindre effektiv prosjektgruppe.
* For lite ressurser og tid lagt ned i forberedelser og planlegging.
  + Kan skape problemer under utvikling, ved at det dukker opp for mange uventede faktorer.
* Manglende kommunikasjon mellom gruppemedlemmer
  + Kan føre til at arbeidsoppgaver blir glemt, eller gjort dobbelt opp. Dette kan bety at prosjektgruppen ikke klarer å holde tidsplan.
* Manglende eller utydelig kommunikasjon mellom prosjektgruppe og arbeidsgiver
  + Kan sette prosjektgruppen på feil vei i forhold til retningen arbeidsgiver ønsker, og dermed kan sluttproduktet ikke bli av god nok kvalitet.
* Ikke standardisert eller dårlig kodeskikk
  + Kan gjøre at prosjektet blir umulig å vedlikeholde
* Utilstrekkelig med testing og analysering
  + Kan gjøre at prosjektgruppen gir ut programvare som ikke fungerer godt nok, og dermed skape et dårlig sluttprodukt.

Vi i prosjektgruppen har laget en plan for å minimere risikoelementene. Denne planen innebærer:

1. Gjøre et grundig forarbeid av prosjektet, gjøre nødvendige analyser og lage en tidsplan.
2. Ha minst 3 gruppemøter hver uke, og bruke gode verktøy for å håndtere kommunikasjon og systemutviklingsmetoden.
3. Lage prosedyrer for dokumentasjon, og gi ansvar til prosjektmedlemmer utførelse av disse prosedyrene.
4. Vi skal gå igjennom hverandres kode, og etter beste evne bruke «beste praksis» innen programvareutvikling. Konstruktive tilbakemeldinger blir viktig gjennom hele prosjektet livsløp.
5. Prosedyrer for testing og analysering av utført arbeid.