

Rapport för individuell fördjupning DA336A VT16

Namn: Carl-Håkan Hovstadius

Födelseår: 79-01-28

Projektgrupp: 10 HitIt

Program: TGSPA15h (Spelutveckling)

Grupphandledare: Zahra Ghaffari

Vald uppgift: Användbarhetsanalys

Dokumentation:

Dokumentation som ligger till grund för användbarhetsanalysen.

Designdokument filnamn: da336a_designdokument_HitIt_v.1.3.pdf

Scenarion/Användningsfallsbeskrivningar och Användargränssnitt.

Kravdokument filnamn: da336a_kravdokument_grupp10_vt16_v1.3

Funktionella krav, kvalitativa krav och Icke funktionella krav.

Kod-fil: HitIt 0.99

I detta dokument beskrivs övergripande de delar som rör användarvänlighet.

Referenser

Shneiderman, B. and Plaisant, C., Designing the User Interface: Strategies for Effective Human-Computer Interaction: Fifth Edition, Addison-Wesley Publ. Co., Reading, MA (2010)

Metod

Vi har valt att använda Shneidermans åtta gyllene regler för Interface Design (Shneiderman, 2010). Detta dokument är uppbyggt så att först presenterar jag en av Schneidermans åtta regler och en förklaring av denna, därefter beskriver jag hur vi försökt att implementera regeln i vår design av webbapplikationen och till sist min analys och hur vi kan förbättra den. Dokumentet går igenom alla de åtta reglerna.

Var konsekvent: *Handlar om att sträva efter konsekvens i gränssnittets valmöjligheter, terminologi och grafiska design.*

HitIt applikationen är uppbyggd på ett sådant sätt att alla sidor och knappar har samma design vilket gör att man lätt känner sig hemma i designen och gör det smidigt att navigera. All text är tänkt att vara på Engelska.

Analys: på vissa ställen har applikationen automatiskt valt att översätta engelsk text som t.ex. "submit" till svenska "skicka fråga". För att göra applikationen så konsekvent som möjligt bör vi åtgärda problemet.

Tillgodose universell användbarhet: *Att sträva efter gränssnitt som kan användas av personer med olika erfarenhet, ålder, funktionshinder eller arbetsuppgifter.*

Denna regel är inte fullt implementerad då vi inte tagit hänsyn till personer med funktionshinder. Dock är den universell i den meningen att all information är på engelska. Vi har eftersträvat att göra en produkt som liknar många andra applikationer.

Analys: För att i framtiden göra det lättare för fler användare att använda applikationen så kan man se till att det går att välja att köra programmet på andra språk. Eftersom applikationen har en design som liknar andra applikationer så blir den lättare att använda eftersom användaren kan relatera till andra liknande applikationer.

Ge återkoppling: *Alla handlingar en användare utför ska ge respons. Responsen ska vara avvägd till hur vanlig eller viktig handlingen är.*

I HitIt applikationen ger alla länkar eller användarinput direkt respons. Trycker man på en länk till en annan sida så blir man direkt slussad dit. Trycker man i en textruta så kan man direkt börja skriva i den.

Analys: Vi har inte direkt gjort någon avvägning till hur vanlig eller viktig en handling är utan mest satt funktionalitet i fokus. Varje handling ska utföra en funktion med önskat resultat beroende på handlingen vilket jag anser att programmet gör.

"Början, mitten, slut": *Sekvenser av handlingar ska vara grupperade med början och slut. När en grupp handlingar avslutats ska systemet ge användaren återkoppling som bekräftar att sekvensen är avslutad.*

Vi har designat programmet att utföra saker i en viss ordning. Programmet är utformat att spara data i tabeller i en databas. Först navigerar man för att ändra eller lägga till information. Därefter hämtar applikationen denna data som blir synlig för användarna i olika situationer. Man kan sedan välja att radera information man lagt till.

Analys: Just nu så blir man dirigerad till olika sidor för att utföra olika handlingar. För att göra det ännu mer användarvänligt så skulle man kunna utforma sidorna så att fler saker gick att göra på samma sida. T.ex. att det gick att ändra användaruppgifter i samma vy som visar detaljer om användaren. Annars tycker jag att strukturen har en tydlig början och slut. När man trycker på "save" så ser man direkt vilka uppgifter som sparats.

Ha en bra felhantering: *Gränssnittet ska vara uppbyggt så att det förebygger fel. Exempelvis genom att inaktivera knappar som inte ska användas vid en viss tidpunkt eller bara tillåta formulär att ta data i rätt format.*

Vi har implementerat felmeddelanden i applikationen som gör det lättare att åtgärda fel (för användaren) som kan uppstå. Vi har även formulär som bara kan ta en viss typ av data och som "säger till" när det blir fel.

Analys: På denna punkten så anser jag att det bara finns små detaljer som man kan finslipa.

Det ska vara lätt att ångra saker: *Användaren ska enkelt kunna ångra sina misstag och återställa systemet till det tillstånd det var i innan misstaget begicks.*

Applikationen sparar data till tabeller i en databas. Vi har gjort så att det enkelt går att ta bort eller redigera datan/uppgifterna man matat in.

Analys: När man lägger till information så har man en bra överblick över vad som kommer att sparas när man trycker på spara. Det finns ingen ångra-knapp men det är lätt att gå tillbaka och ändra eftersom det går att navigera till edit på alla sidor som tillhandahåller information från databasen. Vi skulle kunna implementera en funktion som frågar om man är säker på att man vill göra ändringar innan informationen sparas.

Vem är det som bestämmer: *Utforma gränssnittet så att användaren känner att det är hon som styr vad som händer i systemet.*

Vi har strävat efter att sätta användaren i fokus. Inget händer utan användarens input. Det finns en navigationslist som gör det lätt att gå till önskad sida i applikationen.

Analys: Navigationslisten är ett smidigt sätt att förflytta sig mellan de olika sidorna i applikationen. Det är uppenbart vad som behövs göras för att utföra en handling och användaren har tillgång till funktioner och länkar för att utföra dem.

Belasta inte korttidsminnet: *Undvik att belasta användarens korttidsminne till exempel genom att undvika situationer där användaren måste komma ihåg information från en vy till en annan.*

I vår applikation behöver man aldrig komma ihåg information mellan två olika sidor/vyer.

Analys: Eftersom applikationen inte kräver att man behöver komma ihåg information från en sida till en annan så är denna regel inte relevant i analysen.