

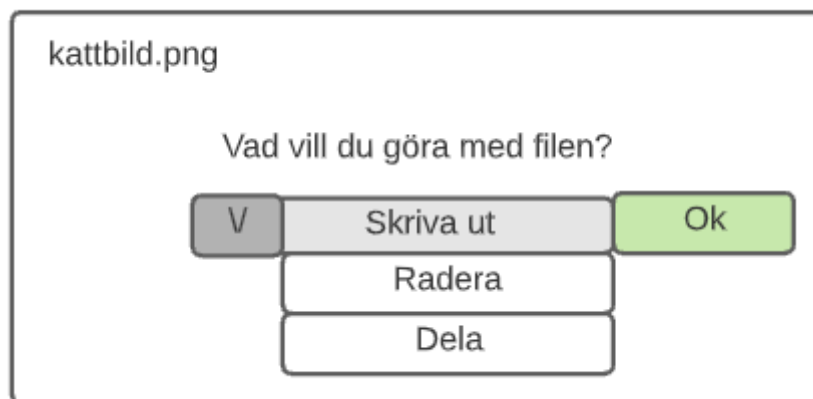
## Uppgift 1

**Antaganden uppgift 1:** Man skall bidra tydliga exempel på en situation när respektive belastning ökar och hur man kan minska den i detta exempel

### Motorisk Belastning

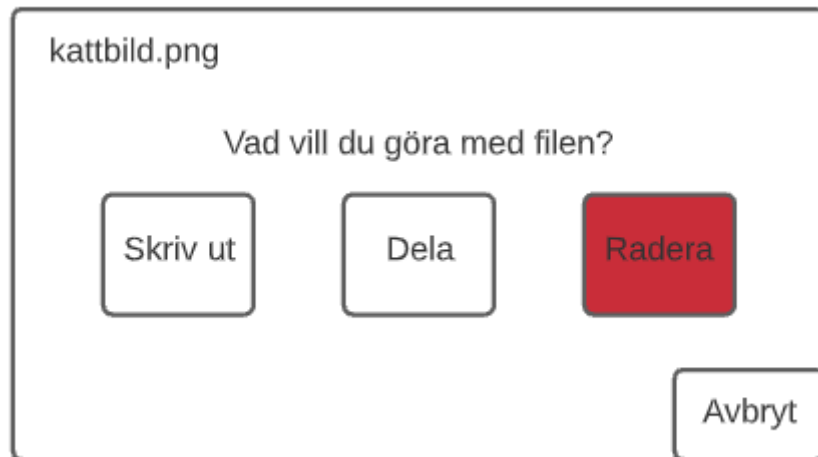
Innebär den kategori kognitiv belastning som uppstår som ett resultat av att en användare av ett program måste fysiskt (motoriskt) röra på sig för att utföra en funktion av ett program, exempelvis att denne måste röra på musen för att trycka på en knapp.

Ett konkret exempel på motorisk belastning skulle kunna vara att en användare måste klicka sig igenom en undermeny för att finna rätt alternativ för vad denne vill göra med en fil, exempelvis:



*Figur 1. Filhantering av kattbild.png*

I exemplet ovan måste användaren först klicka ned drop-down-menyn och sedan välja det alternativ denne vill ha för att sedan bekräfta med "Ok". Detta är 3 klick, om man istället skrev ut alla alternativ från början samt tog bort ok-knappen skulle man minska antalet klick till 1, se nedan:



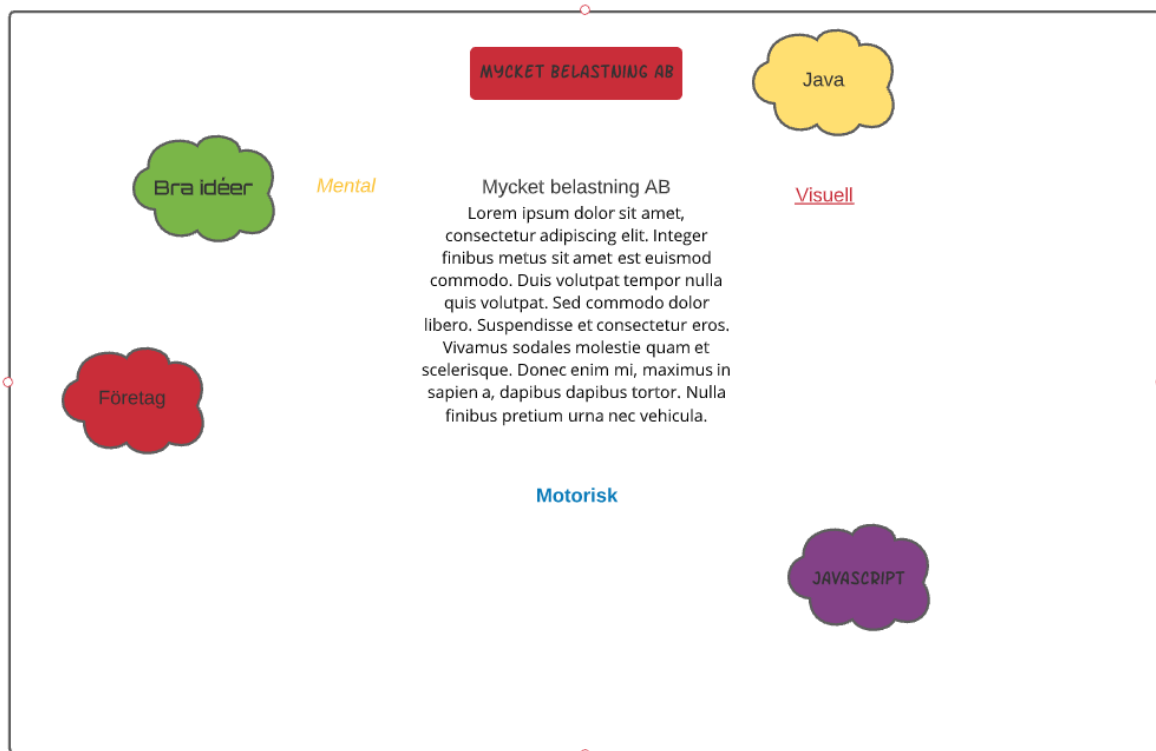
*Figur 2. Föreslagen ändring av design*

Detta innebär att den motoriska belastningen minskas avsevärt och användaren endast behöver klicka 1 gång för att hantera vad denne vill göra med filen, alternativt avsluta filhanteringen, vilket effektivt minskar antalet saker som användaren måste göra för att komma dit den vill vilket minskar den motoriska belastningen.

### **Visuell Belastning**

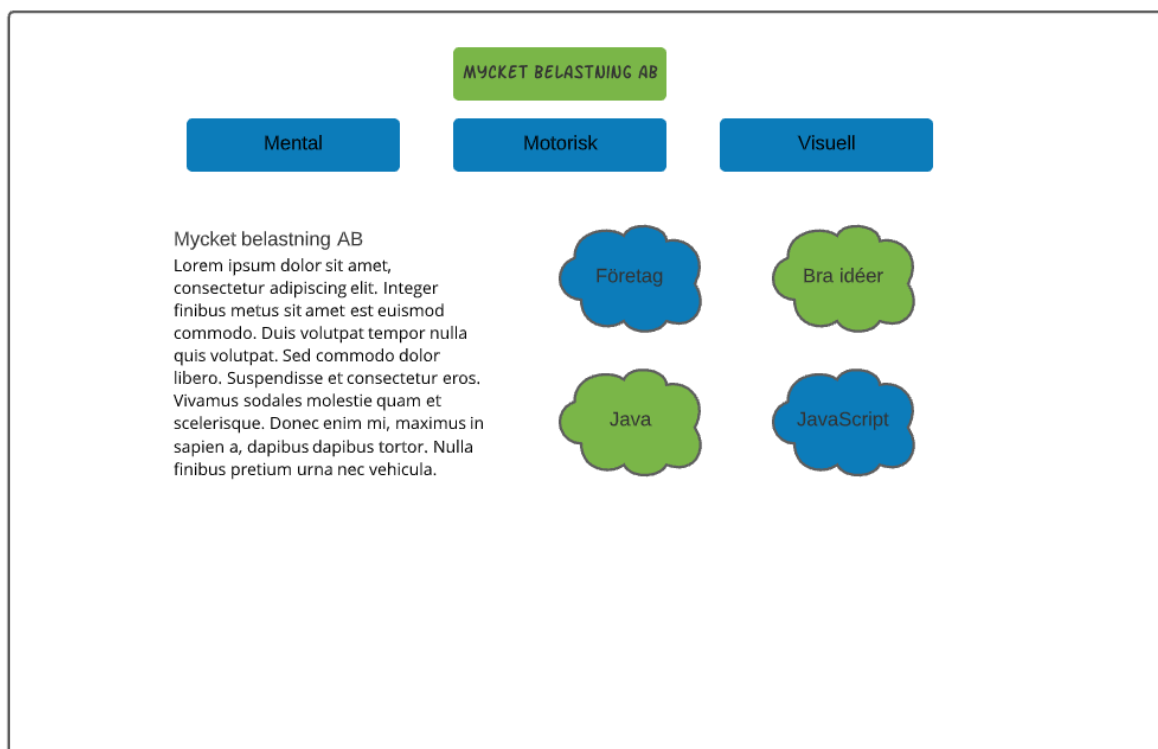
Innebär den kategori kognitiv belastning som uppstår som ett resultat av att en användare utsätts för olika visuella intryck när denne använder ett program, exempelvis många rörliga föremål, olika färger, typsnitt och dylikt.

Ett konkret exempel på visuell belastning är exempelvis en sida med många bilder, olika färger, olika typsnitt, mycket text och otydlig struktur samt hierarki. Detta kan exempelvis se ut:



Figur 3. Exempel på visuellt belastande sida.

För att minska den visuella belastningen kan man implementera en tydligare visuell hierarki, konsekvent typsnitt och färgschema samt omarrangera komponenterna, som nedan:



Här används samma typsnitt på hela sidan, det finns en tydlig visuell hierarki, allt klickbart är understruket och färgerna är konsekventa. Allt interaktivt är färgat i enlighet med färgschemat. Användningen av ett konsekvent färgschema och typsnitt samt att innehållet är tydligt uppdelat och strukturerat minskar den visuella belastningen (Piernik, 2017)

Designförslaget minskar även den visuella belastningen då ögonen dras åt den visuella hierarkin istället för över flera olika ställen på skärmen, vilket minskar den visuella belastningen och därmed låter användaren lätt skanna av sidan och dess innehåll, då användaren främst kommer att skanna igenom hemsidan när denne besöker den. (Krug, 2013, p. 20-27)

### **Mental Belastning**

Den kategori kognitiv belastning som uppstår som ett resultat av att en användare måste tänka eller på annat sätt mentalt bearbeta information för att uppfatta och förstå ett program eller för att utföra funktioner av programmet, exempelvis långa texter, otydlig navigation eller att användaren måste komma ihåg något för att gå vidare.

Ett konkret exempel på mental belastning kan vara:

Föremål 1..... 45kr  
Föremål 2.....40kr  
Föremål 3.....23kr  
Föremål 4.....90kr  
Föremål 5.....44kr

Order:

Summa:

*Figur 5. Exempel på mental belastning*

I detta exempel måste användaren ha koll på mycket i huvudet, nämligen vad som skall beställas samt vilken summa som det skall kosta, istället för att kunna checka i detta i systemet som sedan automatiskt beräknar och sammanställer ordern och summan för ordern. Om man istället reducerade textinmatningen till förmån för lägga till och ta bort knappar som nedan:

Föremål 1.....	45kr	<input data-bbox="981 320 1129 383" type="button" value="+"/>	0	<input data-bbox="1193 320 1342 383" type="button" value="-"/>
Föremål 2.....	40kr	<input data-bbox="981 427 1129 490" type="button" value="+"/>	0	<input data-bbox="1193 427 1342 490" type="button" value="-"/>
Föremål 3.....	23kr	<input data-bbox="981 535 1129 598" type="button" value="+"/>	0	<input data-bbox="1193 535 1342 598" type="button" value="-"/>
Föremål 4.....	90kr	<input data-bbox="981 642 1129 705" type="button" value="+"/>	0	<input data-bbox="1193 642 1342 705" type="button" value="-"/>
Föremål 5.....	44kr	<input data-bbox="981 750 1129 813" type="button" value="+"/>	0	<input data-bbox="1193 750 1342 813" type="button" value="-"/>

**Order:**  
x föremål 1  
y föremål 2  
**Summa:** x kr

Minskar den mentala belastningen då användaren inte behöver hålla koll på vad som är beställt och vad det kostar utan detta sköts automatiskt av systemet, det minimerar inmatningen samt även hur mycket användaren måste komma ihåg för att fortsätta.

## Uppgift 2.

### Uppgift 2a)

En konceptuell modell är den modell som utvecklaren för programmet skapar som en bild av hur programmet kommer att fungera samt se ut. (Brajdic, 2019)

Den konceptuella modellen skiljer sig från den mentala genom det att den konceptuella är den modell som utvecklaren av ett program tänker sig ett program skall se ut till skillnad från den mentala modellen där det är hur en användare tänker sig att programmet skall se ut och bete sig som är i fokus.

## Uppgift 2b)

Jag kan balansera min egna konceptuella modell för hur jag vill att programmet skall se ut med hur ett kalkylatorprogram typiskt ser ut och därmed föreställa mig hur den mentala modellen hos en tänkt användare kan se ut när denne vill använda mitt program.

Jag kanske vill sticka ut genom att göra mitt program unikt och använda mig av koncept som ofta inte används i kalkylatorprogram, men jag måste väga dessa mot den mentala modellen av den tänkta målgruppen för att inte göra programmet allt för främmande vilket eventuellt kan leda till ökad kognitiv belastning hos användaren då denne måste spendera tid och tankekraft till att navigera och lära sig programmet utan att kunna luta sig tillbaka på sin mentala modell och hur ett kalkylatorprogram vanligtvis ser ut.

## Uppgift 4)

### **Antaganden:**

Man skall bara kunna boka ett bord

Man skall inte hantera in/utloggning

Användaren kan se kartan utanför programmet

Programmet i lösningsförslaget är mycket minimalistiskt, men innehåller alla viktiga funktioner för ett bokningssystem för en yta.

Programmet använder sig av en lista med samtliga bord i salen där användaren får leta fram det bord denne vill boka, för att sedan klicka i sin bokning. Bokningen är aktiv så länge som den inte manuellt klickas ur och därmed finns det ingen tidspress på användaren.

Programmet använder ingen in/ut-loggning.

Programmet använder sig av färger för att förmedla ett budskap och allt i programmet som är klickbart, alltså knapparna för bokning respektive avbokning, är färgat i korresponderande färg för bordets bokningsstatus, antingen bokat (rött) eller inte bokat (grönt). Detta förmedlar tydligt vilket bord som användaren bokat vilket sedan låter denne komma tillbaka och avboka det bordet som den tidigare bokat.

Programmet har en tydlig numerisk hierarki där det går från botten till toppen genom i numerisk ordning (1-16) och användaren behöver inte klicka sig fram och tillbaka mellan menyer för att genomföra en bokning utan den motoriska belastningen är endast begränsad till ett klick för att boka samt ett för att avboka. Den mentala belastningen är även minimal då programmet inte kräver att användaren skall behöva komma ihåg vilket bord denne valt utan istället kommer programmet ihåg detta tills det blir avbokat. Den visuella belastningen är inte heller särskilt stor, eftersom programmet är mycket minimalistiskt och använder ett konsekvent färgtema och typsnitt för att dekorera sidan. Vidare finns inga bilder utan programmet är helt enkelt bara checka i.

## Bibliography

Brajdic, A. (2019, Jun 12). *Understanding mental and conceptual models in product design*.

Understanding mental and conceptual models in product design.

<https://uxdesign.cc/understanding-mental-and-conceptual-models-in-product-design-7d69de3cae26>

Krug, S. (2013). *Don't make me think* (3rd ed.). New Riders.

<https://www.amazon.com/Dont-Make-Think-Revisited-Usability-ebook/dp/B00HJUBRPG>

Piernik, M. (2017, Aug 17). *8 ways to reduce cognitive load: Part 2*. 8 ways to reduce cognitive load: Part 2.

<https://uxplanet.org/8-ways-to-reduce-cognitive-load-part-2-4b0f9d8ef5ad>