

# App för serveringspersonal

Grafiska användargränssnitt, Laboration 3

Simon Åkerblom (910330)

30 maj 2022

## Introduktion

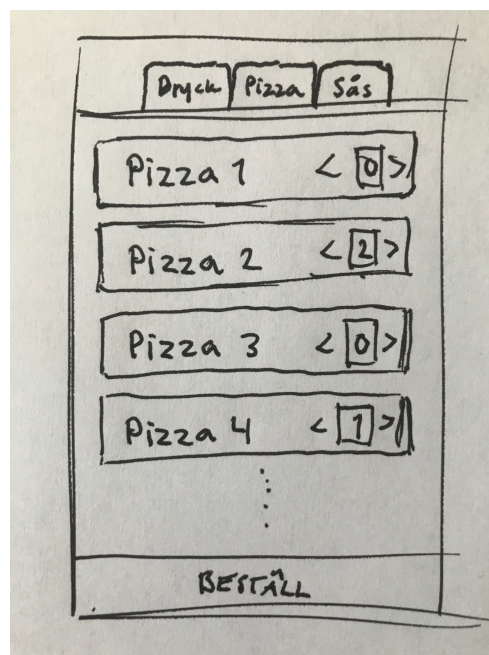
Det som presenteras i den här rapporten är en prototyp av en webb-app som är tänkt att ersätta penna och pappär för att ta emot beställningar på en restaurang. Webb-appen har en simpel layout med tre flikar, en för dryck, en för pizza, och en för sås. Appen är responsiv och fungerar på många olika skärmstorlekar, men är främst tänkt att användas på en surfplatta/telefon med touchskärm. För att ta emot en beställning behöver servitören först välja ett bordsnummer, sen lägga till det gästen beställer genom att trycka på de blå knapparna ("<" eller ">") bredvid varje menyobjekt. För att visa totalkostnaden och skicka vidare beställningen till köket trycker man på "Visa"-knappen. Om de vill göra några ändringar (lägga till/ta bort ingredienser) skrivs det i fritextrutan innan beställningen skickas vidare till köket. För att skicka beställningen klickar servitören på "Skicka"-knappen.

## Grafisk Design

Det som var viktigt att ha i åtanke när man gjorde den första skissen var att appen skulle vara responsiv och fungera väl med en touchskärm. Det var viktigt att alla knappar var stora nog att kunna användas på t.ex. en telefon. Appen ska vara avskalad för att inte distrahera stressig personal och tillräckligt självförklarande för att ny personal enkelt kan förstå gränssnittet utan någon direkt ansträngning. "Don't make me think" (Krug, 2022) var en av designprinciperna som användas för inspiration vid framtagning av den första skissen (figur 1).

Idéen var att få ner den kognitiva belastningen till ett minimum, och därigenom den mentala, visuella och motoriska belastningen. För att få ner den mentala belastningen är appen (skissen) konstruerat på ett sådant sätt att väldigt få steg krävs från att en beställning tas emot från kund till att

beställningen skickas vidare till köket. Servitören behöver inte heller memorera viktig information för att slutföra en beställning. Ingen komplexitet som kräver extra tankeverksamhet. Den visuella belastningen hålls till ett minimum genom att använda Bootstraps standard-komponenter med vedertagen design och färgkombination som både är tilltalande för ögat och med välkända former som är allmänt förekommande på webben. Den motoriska belastningen hålls nere genom användandet av marginaler och stora knappar/komponenter som gör det svårt att trycka fel, även på en touchskärm. Längst ned i appen ligger ett menyfält som alltid är tillgängligt så att servitören när som helst kan välja ett slutföra en beställning, eller ta reda på totalkostnaden. Inget onödigt skrollande.



*Figur 1. En första skiss.*

Dryck

Pizzor

Såser

1. Coca-Cola 33 cl (15kr)

< 0 >

2. Coca-Cola light 33 cl (15kr)

< 0 >

3. Fanta 33 cl (15kr)

< 0 >

4. Sprite 33 cl (15kr)

< 0 >

5. Mineralvatten 33 cl (15kr)

< 0 >

6. Lättöl 33 cl (15kr)

< 0 >

7. Coca-Cola 50 cl (20kr)

< 0 >

8. Fanta 50 cl (20kr)

< 0 >

Bord

#

Visa

Kvitto

Bord: 10

Dryck	Pris
Coca-Cola light 33 cl	15kr
Mineralvatten 33 cl	15kr
Lättöl 33 cl	15kr

Pizza	Pris
Margherita	65kr

Sås	Pris
Kebabsås blandad 10 cl	10kr

Notering till köket

Totalt: 120kr

Stäng

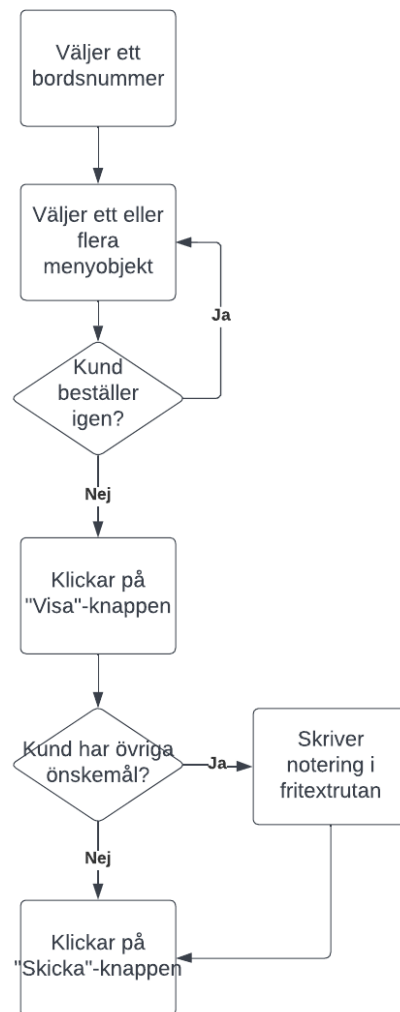
Skicka

Figur 2. Efter implementering.

## Programdesign

Appen är konstruerad med bootstrap-komponenter, lite css, och jQuery för funktionalitet och manipulation av DOM:en. Appen är händelsestyrd. Alla knappar är kopplade till click-händelser med tillhörande hanterare (funktioner) som sköter all funktionalitet. När appen öppnas och DOM:en är "färdigladdad" importeras menyn och nya element skapas och läggs in under respektive flik med jQuery. Datan om vilka menyobjekt som lagt till på kvittot sparas i DOM:en; mer specifikt i "data-"-attribut. När en beställning skickas vidare till köket loopas samtliga element med "data-"-attribut igenom och nollställs. Även informationen som visas på kvittot (modal-komponenten) nollställs inför nästa beställning.

### Servitör process



*Figur 3. Flödesschema som visar servitörens process.*

## Implementation

Inga klasser (förutom objektet innehållande menyn) användes för att implementera den planerade designen och funktionaliteten. I script-filen finns en `ready()`-metod som väntar på att DOM:en är redo och kan manipuleras. All funktionalitet är placerad här och styrs av användarinput. Alla knappar fångar upp klick-händelser och kör tillhörande kod deklarerad i `function()`-blocket. Menyn är sparad i JSON-format och är uppdelad i fem kategorier: Pizzor klass 1, Pizzor klass 2, Pizzor klass 3, Såser, Dryck.

jQuery-metoderna `append()`, `text()`, `empty()` används flitigt för att manipulera användargränssnittet. `each()`-metoden används för att loopa igenom samlingar av jQuery-objekt (eller DOM-element). När samtliga fält ska nollställas efter att en beställning skickats behöver alla element först hämtas ut med en selektor, sen loopas igenom och ta bort "data-" attributen.

## Slutsats

En repetition av jQuery och Bootstrap har varit väldigt nyttigt, och intressant att gå från designprocess till implementation. Jag har lärt mig hur viktigt det är att sätta sig in i slutanvändarens situation/arbetsmiljö när man designar ett nytt system. Jag har också lärt mig hur viktigt det är med en konsekvens i designen och att man inte undviker att ändra små detaljer. Marginaler och knappstorlekar kan t.ex. ha en stor betydelse för slutanvändaren, även om det kan kännas trivialt och onödigt att lägga tid på som utvecklare.

Eftersom det var denna laboration jag valde att göra användartestet på har jag även lärt mig att identifiera problem som är svåra att förutse innan ett användartest. Som utvecklare är det svårt att se helheten och att försöka imitera "vanliga" användare (d.v.s. personer som inte lägger någon tanke vid hur appen/programmet man använder fungerar bakom kulisserna) är hopplöst. Det är jätteviktigt att man utför användartester kontinuerligt genom hela projektet.

I användartestet beskriver jag ett antal problem med designen och hur dessa eventuella skulle kunna lösas. En ganska stor förändring som hade gjort appen mer användarvänlig hade varit om tillägg/borttagning av ingredienser sköts innan pizzan läggs till på kvittot, och att totalkostnaden kalkyleras direkt baserat på ändringarna. Om en kund t.ex. beställer extra ost så ska det framgå på kvittot och totalkostnaden ska öka med kostnaden för tillägget (+5 kr t.ex.). Fördelen med denna lösning är att man inte behöver skriva något i fritext-fältet – utom vid undantagstillfällen. Det hade snabbat på beställningsprocessen och

minskat missförstånd eftersom det med fritext alltid finns en risk att servitörerna skriver fel.

## Referenser

*Krug, S. (2022). Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability (3rd ed.). Pearson Education.*