

Applikationens udvikling

Som en del af denne opgave, skulle der udvikles en mobilapp med tilhørende databaseadgang, som bl.a. kunne bruges i et byggemarked. App'en skulle udformes med en speciel funktion til kunder, der lige er trådt ind i butikken. Så der skal både udvikles en app og en database med et varesortiment, og hvor varerne befinder sig i butikken.

Applikation eller app, så er app bare en forkortelse af applikation, som i bund og grund er et program, så derfor vil man opleve, at jeg bruger de forskellige navne sådan lidt i flæng.

Idéen er, at når kunden er trådt ind i butikken og åbner appén på sin mobiltelefon, så skal vedkommende kunne søge efter en specifik vare i den pågældende afdeling, og samtidigt få information om, hvor varen befinder sig i butikken, og det kan gøres ved, at vise f.eks. lokation, reolnr og hyldenr. Derved kan kunden følge butikkens afmærkninger for, hvor lokationen befinder sig, og herefter er det bare at finde reolnummeret med det rette hyldenr på selve lokationen.

En lokation er en slags zone eller område i selve butikken, hvor man samler bestemte typer varer. F.eks. kan maling og værktøj befinde sig i hver deres område.

Det er derved vigtigt, at der er en god skiltning i butikken, så det ikke er for vanskeligt at finde frem til det område(lokation) i butikken, hvor varen befinder sig. Herefter er det bare at finde den rette reol med det rette nummer, og bagefter finde vare på den rette hylde med det korrekte hyldenr.

Det lyder jo nemt nok, men for at det skal kunne lade sig gøre, så skal appén jo lige udvikles først, og til det formål, har man forskellige udviklingsværktøjer til rådighed.

Normalt vil man udvikle sådan en app i mobiludviklingsværktøjer, men i vores tilfælde, der har vi ikke lige arbejdet med mobiludvikling, så i stedet var valgmuligheden i mellem at lave appén, enten som en windowsversion i enten C-Sharp eller Python, eller som en responsiv webside, lavet i html, css og javascript.

I vores tilfælde, valgte vi en C-Sharp version, som så også ville blive en windowsversion, og det havde noget at gøre med, det glimrende visuelle udviklingsværktøj, som er en del af Visual Studio Code. En Python version ville tage alt for langt tid, da der ikke findes noget visuelt windowsværktøj til Python. Man kan godt lave windowsprogrammer i Python med f.eks. tkinter, men der skal man selv definere form og controls manuelt, og det er en lidt større historie, så den løsning ville tage for langt tid, i forhold til den tid vi havde til rådighed.

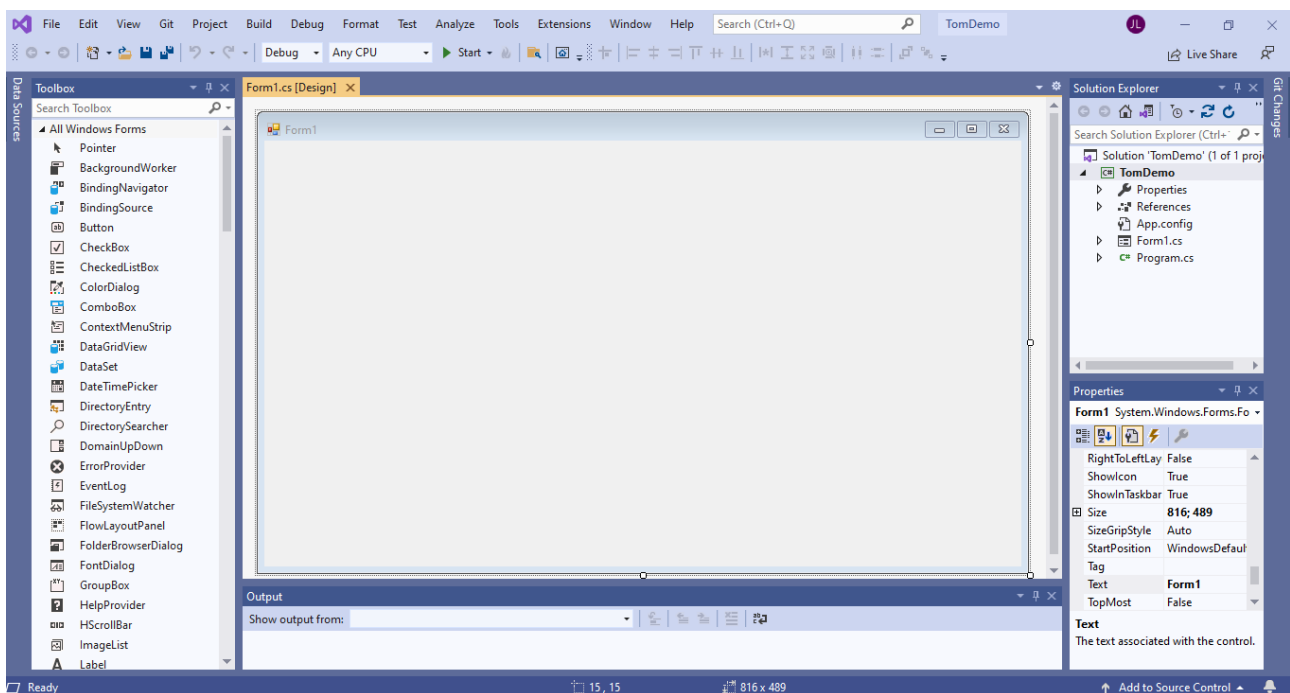
En html/css/javascript løsning med Python som backend til databaseadgang, kunne også have været en mulighed, da værktøjerne til webudvikling er langt mere dynamiske, og set i bakspejlet, så havde det måske været mest hensigtsmæssigt, at have lavet app'en ved hjælp af disse værktøjer, men nu blev det en C-Sharp løsning, og samtidigt kom man også rundt om nogle forskellige problemstillinger og løsninger, hvis man senere skal lave andre windowsprogrammer.

Som sagt, var hovedkravet til app'en, at den skulle kunne vise lokation, reolnr og hyldenr på en vare, som kunden havde søgt på, og naturligvis også et billede af varen, men for at denne løsning kunne realiseres, så skulle der først udvikles en database, som kunne indeholde et lokations/reol/hylde system, og hvor varen var bundet til et hyldenr, som igen var bundet til et reolnr, og som til sidst var bundet til en lokation.

Samtidigt skulle databasen være en fælles database, som indeholdt alle alle virksomhedens afdelinger på en gang, så man bruge den samme database til at få information om en vare, uanset hvilken butiksafdeling, man stod i.

Dvs app'en skulle have en dropdownbox, hvor man kunne vælge den butiksafdeling, man lige var trådt ind i, og samtidigt skulle app'en også have mulighed for at vise status på varen i samtlige butikker, så man kunne se, hvor mange der var på lager i en anden butik, hvis varen var udsolgt i den butik, man stod i.

En anden funktion app'en også skulle have var, at hvis den pågældende vare ikke var på lager i den butik man stod i, så skulle man have information om, hvilken dato den kunne forventes på lager igen. Yderligere skulle der også være en knap, så man kunne tilføje en vare til kurven, men igen, for at alt det her kunne lykkes, så skulle databasen udvikles først, og da den var færdigudviklet og testet, så var det tid til at gå i gang med selve app'en.



Ovenstående billede, viser udviklingsmiljøet i Visual Studio Code, når man starter udviklingen af en applikation helt fra bunden.

Udviklingsværktøjet er i øvrigt et produkt fra Microsoft, og findes i tre forskellige versioner. En Community version, en Enterprise version, og en Professional version.

I vores tilfælde her, gør vi brug af Community versionen, som er en gratis version, men fuldt ud tilstrækkelig til det behov vi har til udvikling af vores app.

I midten af skærmen har vi en form, og det er her man sætter knapper, combobokse osv. som man vælger i værktøjsboksen i venstre side af skærmen, og i højre side har vi bl.a. properties(egenskaber), hvor man kan definere mange forskellige egenskaber for en enkelt control, og det kan f.eks. være en knap, som også hedder en button eller en combobox, som er en dropdownboks. En dropdownboks er en slags udfoldende rullemenu.

Hvis man er forvirret over sammenblandingen af udtrykkene på dansk og engelsk, så er det fuldt forståeligt, for der findes ikke nødvendigvis en direkte dansk oversættelse af mange af disse engelske udtryk, som bruges i udviklingsmiljøet, og når man så alligevel oversætter det, så bliver

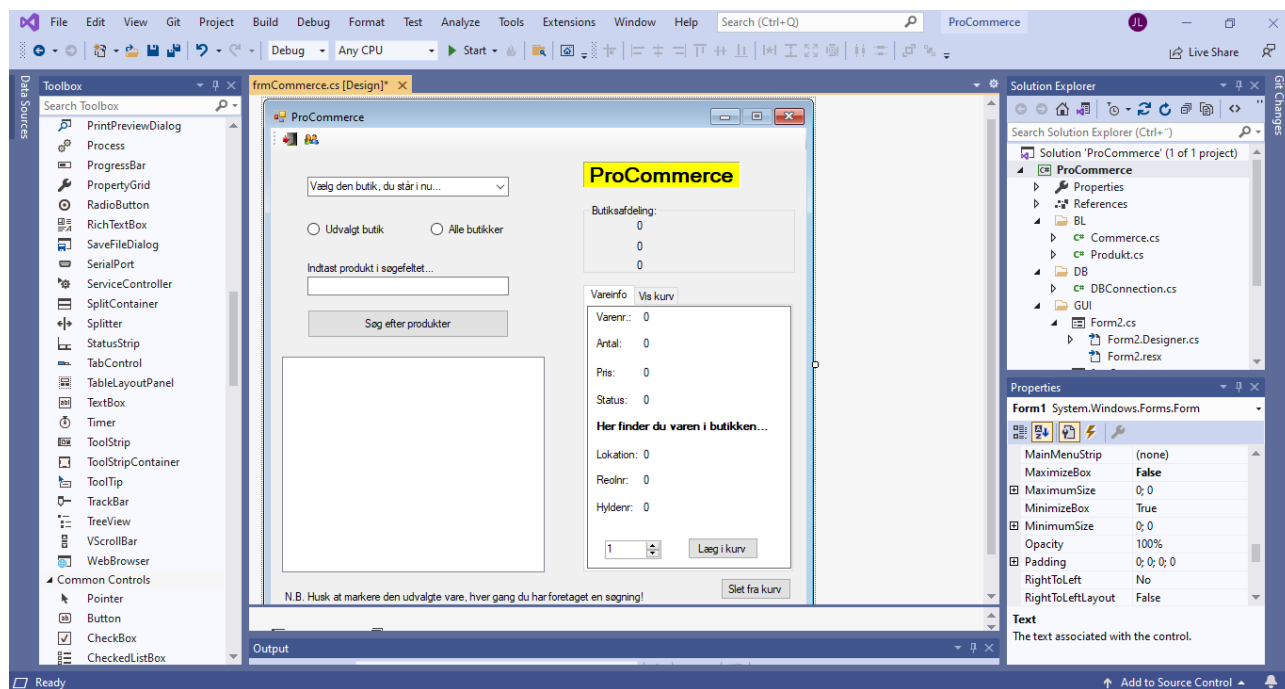
det sådan et mærkeligt sammenblandet ord, hvor halvdelen af ordet er dansk, og den anden halvdel engelsk, lige med undtagelse af f.eks. en button. Den kan oversættes direkte til en knap.

Formen, som er det store rektangel man ser i midten, kan ændres til den størrelse applikationen skal have, og det kan man se, når jeg om lidt viser, hvordan vores færdige applikation så ud, inden den blev oversat med compileren til den færdige prototype.

Når man går i gang med sådan en applikation, så kan det være en god idé, at lave en optegnet skitse på papir, så man nogenlunde har et udgangspunkt for, hvor og hvilke controls, der skal placeres på formen, i stedet for bare at sidde og kaste dem tilfældigt ind, med mindre man lige hurtigt skal teste et eller andet.

Når man begynder at sætte controls på formen, så skal der også skrives noget kode, for ellers virker de ikke. Hvis man f.eks. sætter en button(knap), så kan man dobbeltklikke på den og tilføje noget kode, så der sker en eller anden handling, når man trykker på den i det færdige program.

Når man har placeret alle sine controls de rette steder på formen, så får man en skærm, der i vores tilfælde, ser sådan ud:



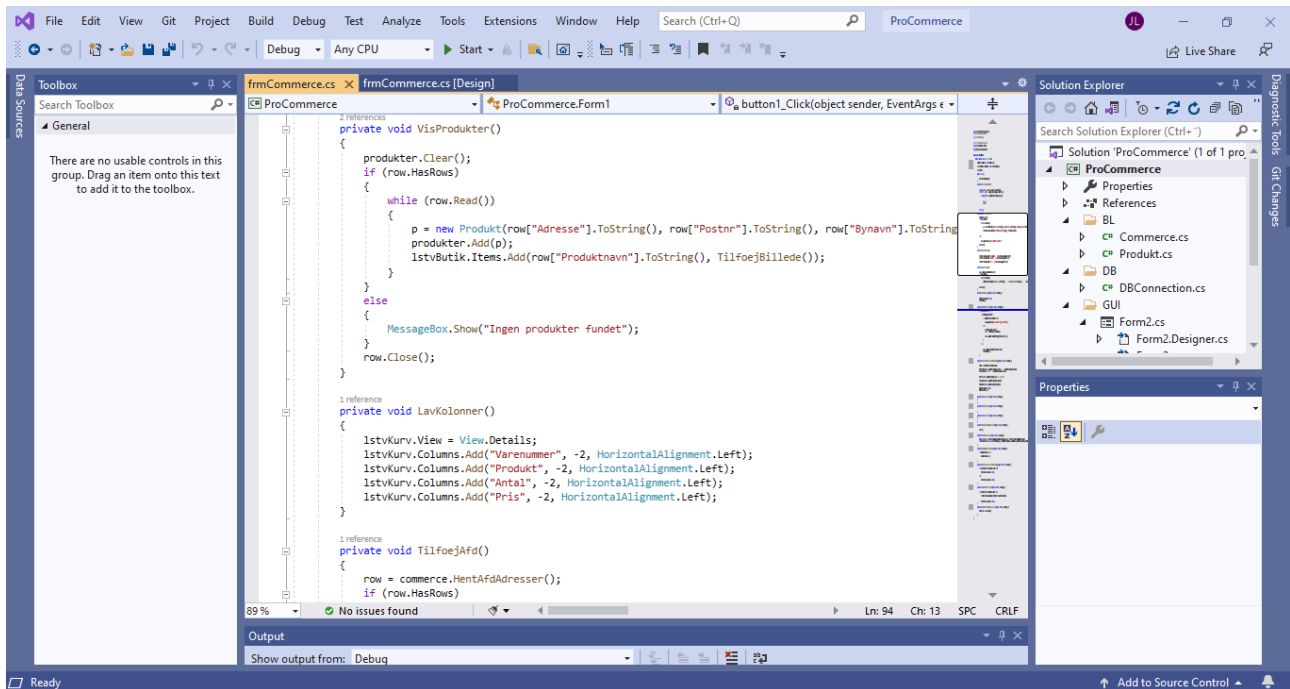
Som man kan se, så har vi kaldt vores applikation for "ProCommerce", og samtidigt kan man også se de forskellige controls, der er placeret på formen. Alt i alt, har vi brugt to forms. Hovedformen er den på billedet og form2 er en lille loginboks, som ikke er med på billedet. Normalt, så navngiver man de forms man bruger i en applikation med et passende navn, men form2 har bare standardnavnet, da den ikke spiller nogen særlig rolle i applikationen, men bare et eksempel. Man skal huske på, at det jo kun er en prototype, vi har lavet. Alt, hvad der har med betalingsdelen at gøre, såsom at gå til kassen, skulle ikke indgå i denne prototype.

Størrelsesmæssigt, så ville den faktisk passe fint til de nye mobiler med to skærme, som kan foldes ud til en dobbeltskærm, hvis den altså var lavet til en mobil.

Egentlig, så var det meningen, at den kun skulle have haft en størrelse, svarende til en almindelig mobil, men implementeringen af den måde, jeg gerne ville have lavet det på, ville tage for langt tid, da der var nogle ekstra ting, som jeg lige skulle sætte mig ind i først, bl.a. brugen af xaml-filer for at få det til at fungere, og da tiden var begrænset, så blev det denne dobbeltløsning i stedet.

Næste skridt er så, at det hele skal oversættes med compileren til et færdigt program, der kan afvikles, men for at applikationen overhovedet har nogen værdi, så skal der også indgå noget kode, som skal eksekveres, hver gang man foretager en handling, der har indflydelse på applikationens funktionalitet.

Her er et lille udsnit af noget af koden i applikationen:



Som man kan se, så har vi nogle erklærede metoder, der bliver kaldt andre steder i programmet. Nogle vil måske synes, at det er ufedt at bruge "MessageBox" i en færdig applikation, men eftersom det kun er en prototype og lavet som en windowsapplikation og ikke en egentlig mobilapp, så ser jeg ikke det store problem i det.

Nu mangler vi egentlig kun at få oversat vores færdigudviklede applikation til en eksekverbar fil, og det gøres ved, at klikke på start i værktøjslinjen. Ud over, at der bliver skabt en .exe fil, så bliver programmet også afviklet i samme ombæring.

Nedenstående billede viser, hvordan programmet ser ud i sin færdige form, når det bliver afviklet:

The screenshot shows the ProCommerce application window. The title bar reads "ProCommerce". The interface is divided into several sections:

- Search Section:** A dropdown menu labeled "Vælg den butik, du står i nu..." with a downward arrow. Below it are two radio buttons: "Udvalgt butik" (selected) and "Alle butikker". A text input field is labeled "Indtast produkt i søgefeltet...". Below the input field is a button labeled "Søg efter produkter".
- Product Details Section:** A large empty rectangular box on the left side of the window.
- Product Information Panel:** Located on the right side, it contains a yellow header "ProCommerce". Below it is a section "Butiksafdeling:" with three input fields, each containing the number "0". Below this is a section with two tabs: "Vareinfo" (selected) and "Vis kurv". Under "Vareinfo", there are fields for "Varenr.: 0", "Antal: 0", "Pris: 0", and "Status: 0". Below these is a section titled "Her finder du varen i butikken..." with fields for "Lokation: 0", "Reolnr: 0", and "Hyldenr: 0". At the bottom of this panel is a quantity spinner set to "1" and a button labeled "Læg i kurv".
- Footer:** A note at the bottom left says "N.B. Husk at markere den udvalgte vare, hver gang du har foretaget en søgning!". At the bottom right is a button labeled "Slet fra kurv".

Til at starte med, så vælger man den butiksafdeling, man står i, oppe i dropdownboksen. Herefter indtaster man et navn på en vare i søgefeltet og klikker på søgeknappen. Efterfølgende dukker der så en række varer op med billede i den nederste rude. Nu markerer man så en vare for at få oplysninger om, hvor den befinder sig i butikken. Ydermere får man også information om varenummer, antallet af enheder, der er på lager, samt pris og status.

Hvis status siger "OK", så er det fordi, der er varer på lager, og det kan man også se på antallet, men hvis antallet siger 0, så får man en status, der siger "Ikke på lager", inkl. dato for, hvor det pågældende produkt forventes på lager igen.

Hvis status siger, "Kontakt butiksmedarbejder", så er det fordi, at varen befinder på et sted, f.eks. på en hylde højt oppe(fra hylde nr. 5 og opefter), hvor kunden ikke selv kan nå varen, og derfor skal der en butiksmedarbejder, måske med en truck eller en stige, til at få varen ned til kunden.

Man kan nu vælge at lægge varen i kurven, og samtidigt kan man vælge, hvor mange enheder man vil købe på en gang.

I toppen af programmet er der en værktøjslinje, som indeholder to ikoner, som man kan klikke på. Den med døren lukker hele programmet, og den med de to personer, åbner en lille loginboks. Det skal dog siges, at den loginboks ikke har nogen funktion, da der ikke er oprettet kunder, så den er bare lidt til pynt, selv om den dog åbner et nyt lille vindue, når man klikker på ikonet.

Sådan ser det ud, når man har søgt efter og markeret en vare:

The screenshot shows the ProCommerce application window. At the top, there's a title bar with the ProCommerce logo and window controls. Below the title bar, there's a search bar with the address "Vonsildvej 2, 6000, Kolding" and a dropdown arrow. Below the search bar, there are two radio buttons: "Udvalgt butik" (selected) and "Alle butikker". Below the radio buttons, there's a search input field with the text "Indtast produkt i søgefeltet..." and the word "lim" entered. Below the search input field, there's a button labeled "Søg efter produkter".

On the right side, there's a yellow box with the ProCommerce logo. Below the logo, there's a section titled "Butiksafdeling:" with the address "Vonsildvej 2", "6000 Kolding", and "Tlf.: 33445566". Below this, there's a section titled "Vareinfo" with a "Vis kurv" button. The "Vareinfo" section displays the following information: "Varenr.: 3447448", "Antal: 14", "Pris: 24,95 pr. stk", and "Status: OK". Below this, there's a section titled "Her finder du varen i butikken..." with the following information: "Lokation: 20", "Reolnr: 2", and "Hyldenr: 3". At the bottom of the "Vareinfo" section, there's a quantity selector with the number "1" and a "Læg i kurv" button.

On the left side, there's a list of products. The first product is "MILLARCO SEKUNDLIM 30ML" with an image of a white bottle. The second product is "DANA TRÆLIM PU 421 LYS 750ML" with an image of a white bottle. The third product is "DANA KLAR KONTAKTLIM 284 40ML" with an image of a blue bottle. Below the product list, there's a note: "N.B. Husk at markere den udvalgte vare, hver gang du har foretaget en søgning!". At the bottom right, there's a button labeled "Slet fra kurv".

Hvis man vil se indholdet af kurven, så klikker man bare på fanebladet ved siden af "Vareinfo". Her kan man så også slette en eller flere varer fra kurven ved at markere dem enkeltvis.

Idéen er så nu, at man går til kassen og foretager en slags "Scan og betal", men den del af det er udeladt i programmet, og i øvrigt var det heller ikke en del af opgaven. Den del af det havde de allerede på plads, som Steen fra EG udtrykte det.

UML Class Diagram

Her har vi så generet og tilrettet et Class Diagram over applikationens klassehierarki, og til det formål, har vi brugt et program, der hedder Class Designer, som man kan installere i Visual Studio Code. Programmet er ikke installeret i Visual Studio Code som standard, så det skal man selv gøre, og det foregår via Visual Studio installeren.

Diagrammet man generer i Visual Studio Code kan i øvrigt eksporteres til et image, så man behøver ikke at bekymre sig om at skulle sidde og lege ”klippe klistre”.

