|  |
| --- |
|  |
| Eksamensopgave |
| Software Construction – S3 |

|  |  |
| --- | --- |
| Formalia: |  |
| Afdeling | AspIT Sønderjylland |
| Underviser | Jens Clausen |
| Censor | Nils-Asbjørn Frederiksen |
| Forløb | Modulperiode 2 forår 2023 |
| Opgaven udleveres | Tirsdag den 04. maj 2023 kl. 08:45 |
| Opgaven afleveres | Onsdag den 05. maj 2023 kl. 15:00 |
| Der arbejdes følgende dage i tidsrummet 08:45 – 15:00 | Tirsdag og onsdag |

Indholdsfortegnelse

[1 Opgave Gold Digger A/S 3](#_Toc133695278)

[1.1 Problemformulering 3](#_Toc133695279)

[1.2 Kravspecifikation 4](#_Toc133695280)

[1.2.1 Generelt 4](#_Toc133695281)

[1.2.2 Brugergrænseflade – Hovedformen 6](#_Toc133695282)

[1.2.3 Brugergrænseflade – Visning, oprettelse og vedligehold af kunder og metal priser 7](#_Toc133695283)

[1.3 UML-Class diagrammer for Gold Digger A/S 8](#_Toc133695284)

[1.3.1 WPF opgaven 8](#_Toc133695285)

[1.3.2 Konsolapplikationen 9](#_Toc133695286)

[1.3.3 ASP.NET Core WEB-API 9](#_Toc133695287)

[1.4 ER diagram for Databasen GoldDigger 10](#_Toc133695288)

[1.5 Validering af indtastninger via RegEx 11](#_Toc133695289)

[1.6 DataDrivenUnitTest 11](#_Toc133695290)

[2 Bedømmelseskriterier og evaluering 12](#_Toc133695291)

[2.1 Eksamen 12](#_Toc133695292)

[2.2 Karakterskala 13](#_Toc133695293)

# Opgave Gold Digger 2.0

## Problemformulering

Til firmaet Gold Digger A/S, har du tidligere udvikles et værktøj der hjalp ejeren Birger Guldtand med at holde priserne tilgængelige og opdateret for hans kunder.

Gold Digger A/S har været særdeles tilfreds med det produkt som AspIT Sønderjylland tidligere har leveret. Derfor har Birger Guldtand besluttet, at det opdaterings/vedligeholdelsesværktøj vi har leveret, skal udvides med ny funktionalitet til håndtering af handler og ansattes lønudregning.

Gold Digger A/S har gennem en længere periode, haft stor stigning i handel på det europæiske marked, hvor der afregnes i USD, EUR og DKK. Gold Digger A/S har derfor bedt os om at videreudvikle de tre produkter til firmaet.

1. Der skal udvikles en WEB API der kan levere både et JSON og et HTML svar på en forespørgsel, hvor man kan se de aktuelle priser i DKK, EUR og USD, på de typer metal Gold Digger A/S sælger:
   * Gold - Guld
   * Lead - Bly
   * Silver - Sølv
   * Copper - Kobber
   * Iron - Jern
   * Aluminum - Aluminium
2. Der skal udvikles en WPF-applikation til vedligeholdelse af kunder og priser på de metaller som Gold Digger A/S sælger, samt til visning af brugerens resultatløn for denne måned.
3. Der skal udvikles en konsol applikation der hvert minut skal kalde *openexchangerates.org*, for at hente de aktuelle kurser for valutaer fra hele verden. Valutakurserne skal gemmes i DB og vise hvad de forskellige valutaer koster i DKK.

## Kravspecifikation

### Generelt

Der er udarbejdet et UML Class-Diagram over, hvordan applikationen skal opbygges med angivelse af relationer og nedarvning imellem de enkelte projektelementer og klasser i programmet. Der må rent strukturmæssigt ikke foretages ændringer, men der må gerne tilføjes klasser og metoder.

Der skal i WPF-udviklingsprojektet implementeres et testprojekt, hvor der skal udføres datadriven unit-test på følgende elementer af koden:

1. Udregning af valutakurser
2. Udregning af samlet pris på en faktura, i alle tre valutaer
3. Kontrol af om indtastede mailadresser er valide

Alle priser skal indtastes i DKK og automatisk omregnes til USD og EUR.

Alle tre løsninger skal benytte samme database.

#### WEP API

Til håndtering af kommunikationen mellem Web API og databasen, skal der benyttes *ASP.NET Core* med *Microsoft Entity Framework Core SqlServer*. Der skal benyttes DB-First.

#### WPF

Der skal laves et system til login. Dette er et ønske der opstået i sidste øjeblik, og er derfor ikke implementeret noget sted, hverken i den medfølgende kode, UML-Class Diagram eller i Databasen.   
Der mangler ligeledes angivelse af løn og bonus aftaler for de enkelte medarbejder typer.   
Du skal implementere en løsning, der let og effektivt kan implementeres i WPF-Applikationen og i den database der stilles til rådighed.   
Der må IKKE laves ændringer i eksisterende tabeller i databasen, men der må gerne tilføjes tabeller og relationer.

Man må ikke kunne gøre noget, før der er gennemført et login ved brug af Brugernavn og Password.

Man må ikke kunne påbegynde en ordre, før der er valgt en kunde.

Der er i den medfølgende GUI ikke lavet nogen knapper eller funktion til at give adgang til redigering af priser på de metaller firmaet sælger. Der skal du komme med en god løsning, som dog skal godkendes af udviklingschefen Jens Clausen.

Så snart brugeren har indtastet en ny pris på et metal i DKK, skal programmet selv udregne hvad prisen svarer til i USD og EUR beregnet ud fra de aktuelle kurser hentet i databasen.

Brugeren skal kunne oprettet nye kunder og redigere eksisterende. Så snart brugeren har oprettet eller redigeret en kunde, skal data gemmes eller opdateres i databasen og på brugergrænsefladen.

Yderst til højre af brugergrænsefladen, er der blevet indsat en UserControl, der skal vise stamoplysninger for den der er logget på.  
Ud over stam oplysninger skal der også vises hvad brugeren kan forvente at få udbetalt ved næste månedsløn, beregnet ud fra grundløn + bonus ordninger.

WPF-løsningen skal overholde 3-lags arkitekturen med et sideordnet Repository.

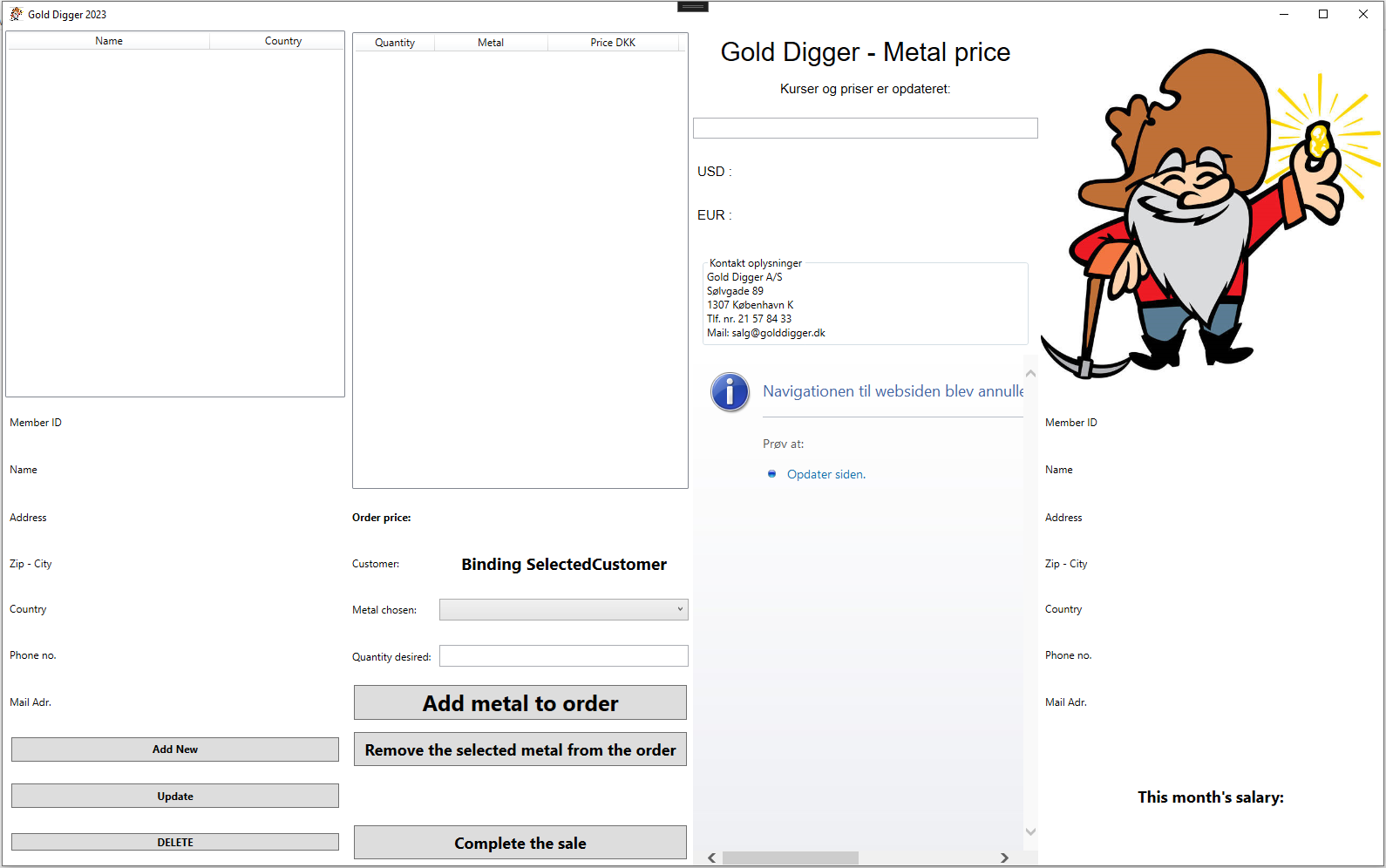
Browseren i UserControlMetalPriceInfo, skal opdateres hvert hele minut + 3 sec.

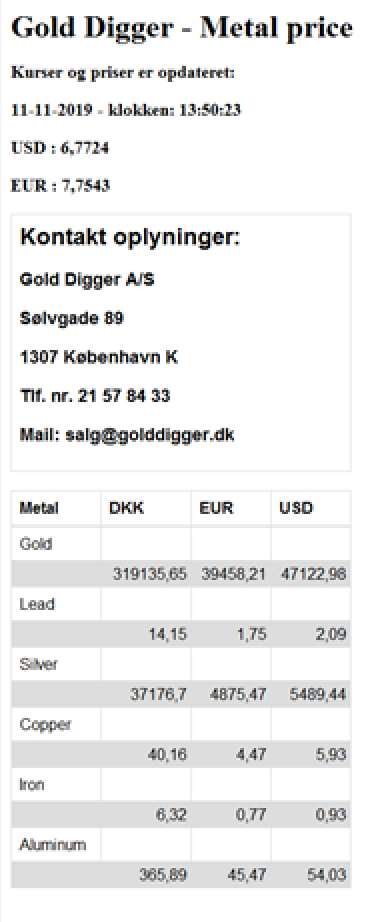
#### Konsol APP

Konsol applikationen skal for hvert hele minut, hente data fra *openexchangerates.org* og opdatere databasen med den nye kurs for alle valutaer, angivet med prisen i DKK.  
Den skal også opdatere USD- og EUR-priserne på metaller i databasen.   
Prisen i USD og EUR skal baseres på metalpriserne i DKK fra databasen.

### Brugergrænseflade – Hovedformen

Der må IKKE ændres i placeringen af elementerne på formen. Alle andre ændringer skal godkendes af udviklingschefen Jens Clausen og dokumenteres.



Brugergrænsefladen som ses her er vedlagt som et Visual Studio 2022 projekt, som du er velkommen til at videreudvikle på. Visningen i Browseren er kun et eksempel og IKKE et udtryk for hvordan data skal præsenteres.

Der kan i UserControlMetalPriceInfo, benyttes en ren WebBrowser løsning som vist her:

### Brugergrænseflade – Visning, oprettelse og vedligehold af kunder og metal priser

Til denne del af applikationen skal du benytte de UserControls der er oprettet i den applikation der medfølger opgaven.

UserControlCustomerEdit skal indsættes så den skjuler UserControlCustome, indtil brugeren er færdig med at redigere. Under redigering, skal alle andre UserControls være utilgængelige.

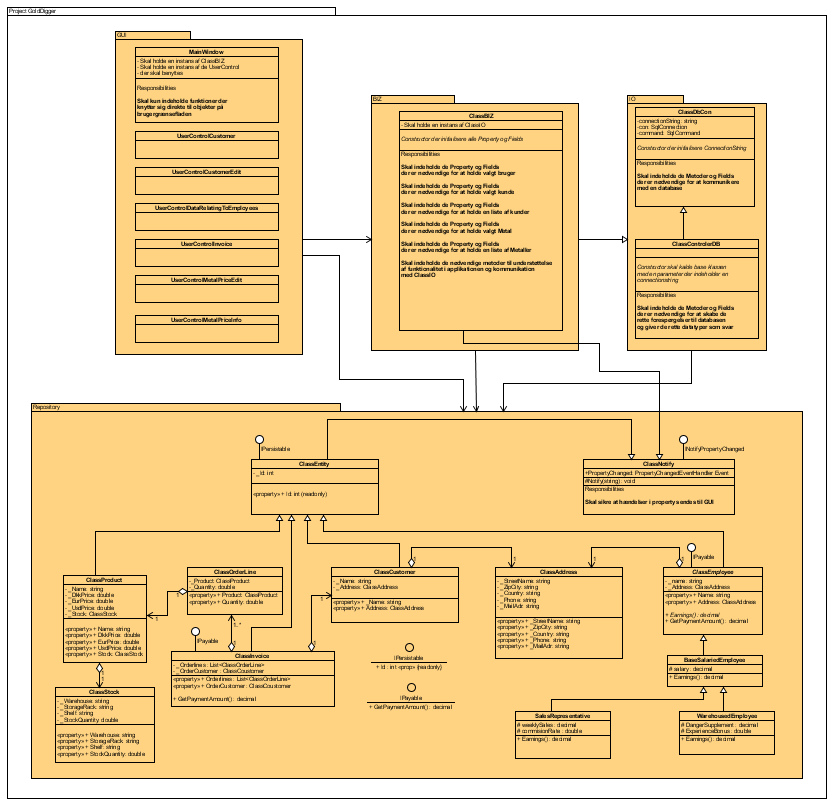
Det skal være muligt at fortryde en igangværende oprettelse eller redigering, uden at det har indflydelse på eksisterende data eller databasen.

Når man er færdig med redigering eller oprettelse, skal alle relaterede data automatisk opdateres på hovedformen.

## UML-Class diagrammer for Gold Digger A/S

### WPF opgaven

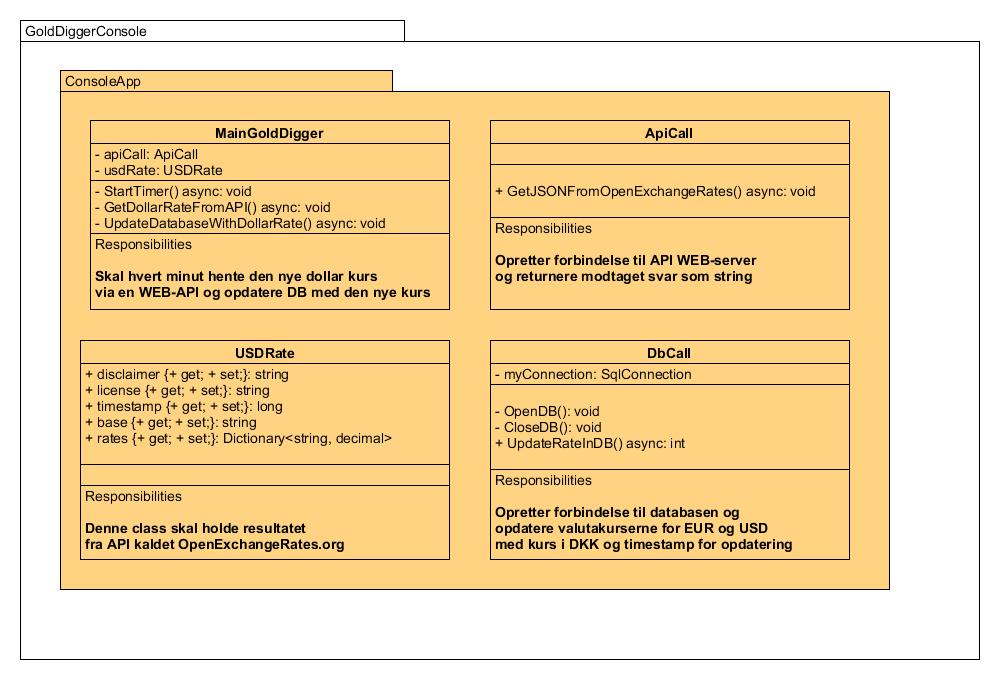
Dette ufuldstændige mug up UML-Class Diagram skal ses som vejledende i forhold til de class filer der benyttes i GUI, BIZ og IO. Du skal selv tilføje de nødvendige members i hver class og navngive dem, så den er selvforklarende og meningsgivende i forhold til hver members ansvarsområde. Der skal navngives på engelsk.  
Dette UML-Class Diagram skal ses som ufravigeligt i forhold til, hvordan løsningen skal struktureres.   
Dog kan der fraviges fra diagrammet, men kun efter godkendelse fra udviklingschef Jens Clausen.



Du skal ligeledes lægge vægt på, at din løsning lever op til reglerne omkring OOP.   
Dette diagram ligger som en .jpg fil og som en. uxf fil sammen med din eksamensopgave i opgavesættet. (*New\_GoldDiggerUML.jpg* og *New\_GoldDiggerUML.uxf*).

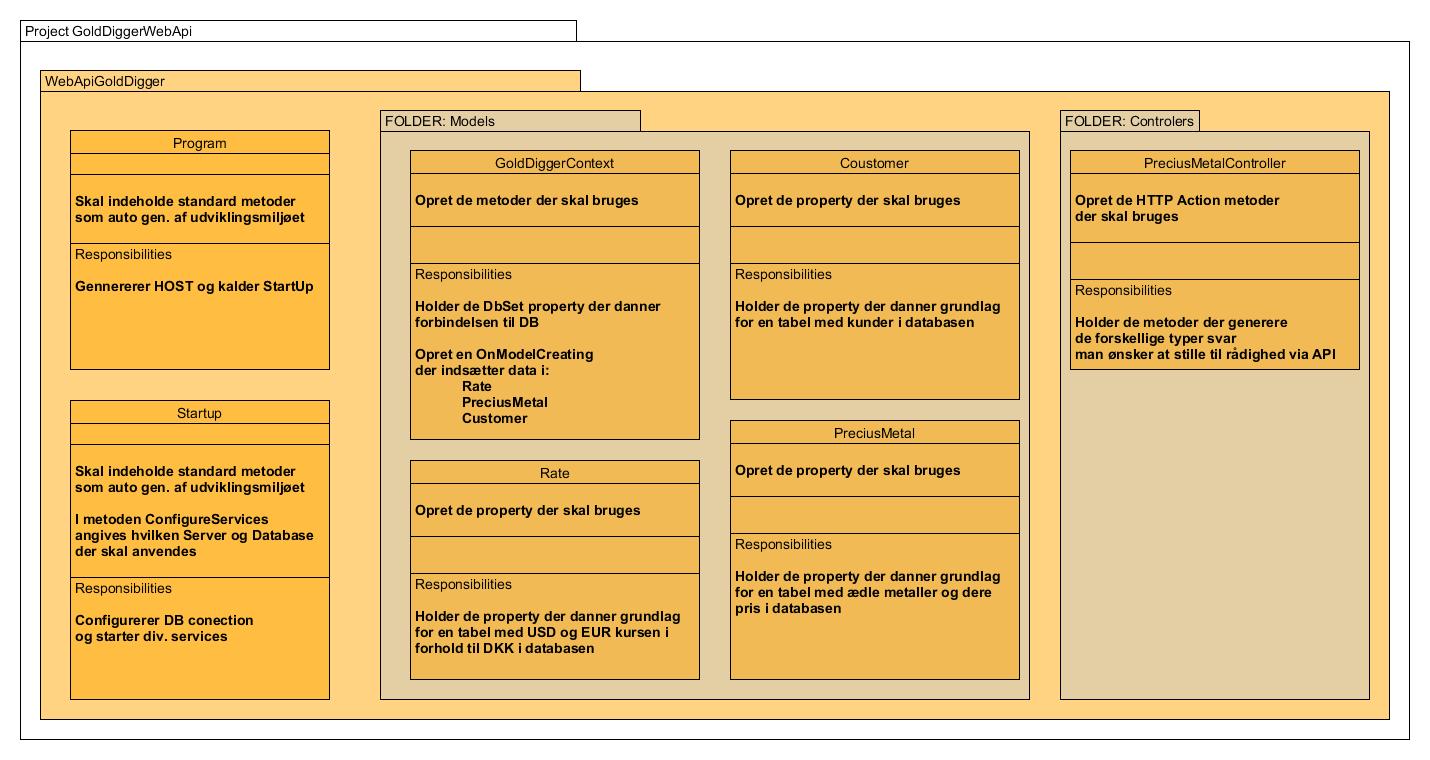
### Konsolapplikationen

Dette ufuldstændige mug up UML-Class Diagram skal ses som vejledende i forhold til, hvordan løsningen skal laves. Dog skal du overholde den overordnede struktur i dit projekt.  
Dette diagram ligger som en .jpg fil i opgavesættet. (GoldDiggerConsoleUML.jpg)



### ASP.NET Core WEB-API

Dette ufuldstændige mug up UML-Class Diagram skal ses som vejledende i forhold til, hvordan løsningen skal laves. Dog skal du overholde den overordnede struktur i dit projekt.  
Dette diagram ligger som en .jpg fil i opgavesættet. (GoldDiggerWebApiUML.jpg)

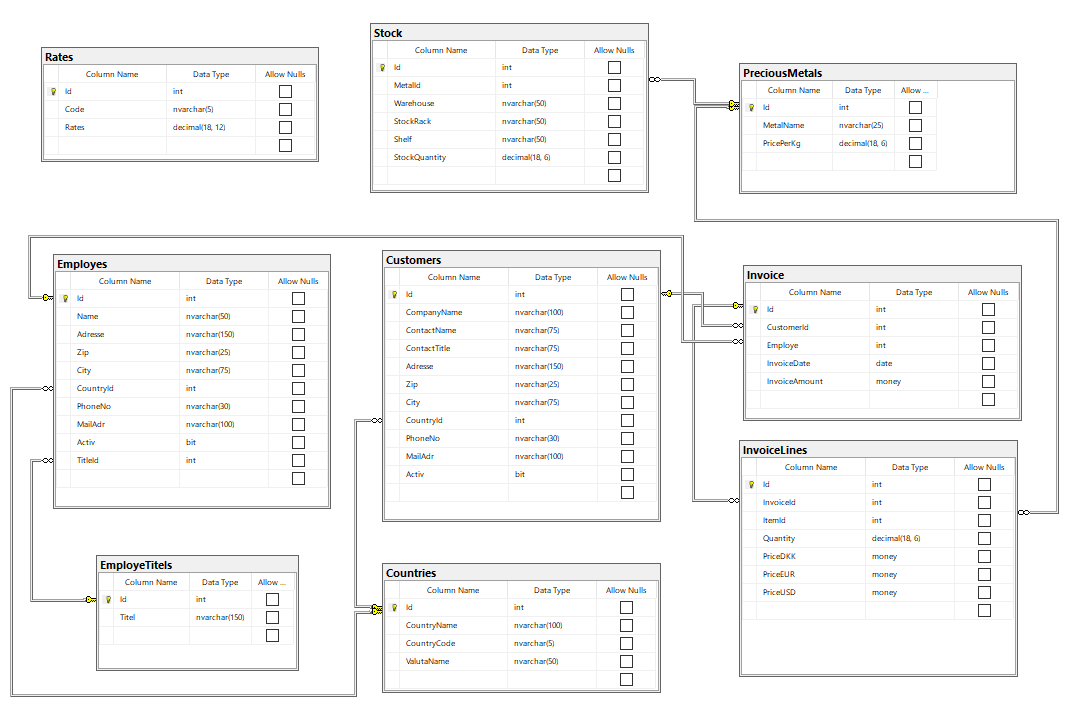


## ER diagram for Databasen GoldDigger

Databasen som skal benyttes, er vist her som et ER-diagram.

Der medfølger et script der danner en DB som vist herunder.  
DB indeholder et udvalg at testdata.

Script er navngivet: NewGoldDiggeDbrScript.sql



## Validering af indtastninger via RegEx

Du skal oprette en class med metode til validering af:

* En kundes indtastede mailadresse.
  + Valideringen skal foretages når brugeren vil opdatere/oprette en kunde.
  + Hvis brugeren har foretaget en indtastning, som ikke kan valideres positivt i RegEx, skal der vises en hjælpetekst på brugergrænsefladen, som skal angive reglen for de enkelte værdier der skal indtastes i dette felt.

## DataDrivenUnitTest

Der skal i WPF-udviklingsprojektet implementeres et testprojekt, hvor der skal udføres datadriven unit-test på følgende elementer af koden:

1. Udregning af valutakurser
   1. Der skal køreres mindst 100 forskellige test med en passende variation af værdier.
2. Udregning af samlet pris på en faktura, i alle tre valutaer
   1. Der skal køreres mindst 100 forskellige test med en passende variation af værdier.
3. Kontrol af om indtastede mailadresser er valide
   1. Der skal køreres mindst 20 forskellige test med en passende variation af forskellige mailadresser.  
      Der skal også forekomme ugyldige mailadresser i testdata, som skal registreres som ugyldige og derved bestå testen.

# Bedømmelseskriterier og evaluering

Der vil i bedømmelsen blive lagt vægt på følgende:

* Fuldstændighed – hvor meget er lavet færdigt?
* Har du hjulpet andre, eller har modtaget hjælp fra andre. Der ses positivt på begge situationer
* Struktur – anvendelse af relevante programmeringsstrukturer og teknikker, såsom anvendelse af 3-lags-model, OOP, variabler, properties og metoder
* Robusthed – giver programmet relevante fejlmeddelelser?
* Relevant dokumentation i koden.

## Eksamen

Det fremgår af modulbeskrivelsen for *Application Programming - S2* at:

Evalueringen foretages som løsning af en applikationsopgave med en tilhørende database. Opgaven løses over 2 skoledage, á 6 daglige lektioner. Opgaven giver dig mulighed for at arbejde med:

* Centrale programmeringselementer
* Databaseprogrammering
* Kobling af brugergrænseflade til resten af systemet.
* Client / Server
* Anvendelse af OOP.

Eksaminationen gennemføres uden foregående forberedelse som en præsentation af den fremkommende løsning. Eksaminationen gennemføres over 30 minutter, inkl. tid til bedømmelse. Opgaven bedømmes efter 7-trins karakterskalaen, og karakteren meddeles eleven.

## Karakterskala

|  |  |  |
| --- | --- | --- |
| Karakter | Betegnelse | Beskrivelse |
| 12 | Den fremragende præstation | Karakteren 12 gives for den fremragende præstation, der demonstrerer udtømmende opfyldelse af fagets mål, med ingen eller få uvæsentlige mangler. |
| 10 | Den fortrinlige præstation | Karakteren 10 gives for den fortrinlige præstation, der demonstrerer omfattende opfyldelse af fagets mål, med nogle mindre væsentlige mangler. |
| 7 | Den gode præstation | Karakteren 7 gives for den gode præstation, der demonstrerer opfyldelse af fagets mål, med en del mangler. |
| 4 | Den jævne præstation | Karakteren 4 gives for den jævne præstation, der demonstrerer en mindre grad af opfyldelse af fagets mål, med adskillige væsentlige mangler. |
| 02 | Den tilstrækkelige præstation | Karakteren 02 gives for den tilstrækkelige præstation, der demonstrerer den minimalt acceptable grad af opfyldelse af fagets mål. |
| 00 | Den utilstrækkelige præstation | Karakteren 00 gives for den utilstrækkelige præstation, der ikke demonstrerer en acceptabel grad af opfyldelse af fagets mål. |
| -3 | Den ringe præstation | Karakteren -3 gives for den helt uacceptable præstation. |

Tabel : Karakterskala for Application Programming - S2.