# Prototypen

Vores projekt bygger på at vi skal lave en applikation, som kan lave 3D prints beregninger. Prototypen er opbygget af en GUI applikation som gør brug af en database. For at skabe den grafiske brugerfalse har vi brugt Tkinter og CustomTkinter. Applikationen giver brugere mulighed for at logge ind, oprette brugere, lave 3D printer beregninger og se historikken for beregningerne.

## Moduler & biblioteker

For at kunne bygge vores applikation har brugt en række moduler og biblioteker i Python.

* **Tkinter og CustomTkinter as c**: Bruges til GUI
* **SQLite3**: Databasehåndtering
* **Pillow**: Håndtering af billeder
* **Matplotlib**: Grafer
* **Messagdebox**: Popup vindue med feedback

## Koden

**Opstart af programmet**

**Et billede, der indeholder tekst, Font/skrifttype, skærmbillede

Automatisk genereret beskrivelse**

For at starte programmet bruger vi denne kode.



Koden her fortæller os at programmet skal starte hvis dette er hoved scriptet. Hvis scriptet var blevet importeret ind i et andet, ville det ikke være Main. Dette har vi gjort for at sikre at det er hovedprogrammet som der bliver kørt.



Her oprettes der en instans til appklassen. manufacturing.db er vores database, som bliver taget med videre til App klassen som et argument db\_name.



Mainloop holder vores applikation kørende.

**App klassen**

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Klassen app fungerer som hovedvinduet og styrecenteret for programmet. Det er det første vindue som der bliver åbnet, når programmet køres. De andre klasser (LoginMenu, Mainmenu, Adminmenu, GUISetup og Historik) fungere som frames der bliver vist og styret i App.

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, linje/række

Automatisk genereret beskrivelse

Klassen arver fra en anden klasse c.CTK, som er en klasse fra bibloteket. Det betyder at app automatisk får alle funktioner og metoder der er i customtkinter. Dette bruges til at åbne vinduet for brugerfladen.

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede

Automatisk genereret beskrivelse

Følgende ligner kode bruger brugt til at sætte vinduet titel, størrelse og farvetema.



Her bliver der skabt en ny instans af Loginmenu klassen. Hvilket betyder at den indsætter Login framen på det nuværende vindue. Der bliver videregivet to argumenter som er self og db\_name. Self er instansen for klassen App, og db\_name er vores database.  
Linjen er placeret i klassen App, fordi det første man skal se når man åbner for applikationen, er login\_menu.

**Login menu**

Et billede, der indeholder tekst, skærmbillede

Automatisk genereret beskrivelse

Login menuen består af en masse widgets, som vi har brugt til at designe den. To af de widgets er entry til indtastning af brugernavn og adgangskode.



Når brugeren ønsker at logge ind, og trykker på loginknappen, kaldes funktionen log. For at log funktionen kan virke, tager vi brugernavnet og adgangskoden med over som parametre.

**Login funktion**

Et billede, der indeholder tekst, skærmbillede, software, Font/skrifttype

Automatisk genereret beskrivelse

Funktionen her bruges til at tjekke om de indtastet oplysninger fra login menuen er i vores database. Grunden til at den er oprettet som funktion var at vi oplevede nogle problemer med argumenter da vi skulle sætte vores kode sammen. Vi kører kun denne funktion når der bliver logget ind, og ikke andre tidspunkter i programmet, og ville derfor have været bedst placeret som en metode i menu klassen.

Koden starter med at hente de indtaste oplysninger fra menu siden, og tilkobler sig til databasen.



Sql forsørgelsen starter med at hente alle kolonner fra tabellen bruger. Tabellen bruger består af brugernavn, adgangskode og type id. Koden her bruger kun brugernavn og adgangskoden. Den tager brugernavnet og adgangskoden og erstatter ”?”, så den finder, kun rækker hvor begge matcher den værdi.  
Grunden til at den også skulle hente type id, var for at se om brugeren var administrator, eller en standard bruger. Meningen var at hvis brugeren var administrator, skulle man have adgang til admin siden, og hvis brugeren var oprettet som standard, skulle den ikke blive vist. Vi oplevede problemer med dette, og måtte fjerne det under sammensætning af vores kode.

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Hvis login lykkes, rydder den alle widgets i vinduet, for at gøre plads til de nye. Det gør ved hjælp af en for-lykke som går alle widgets igennem, og bruget destroy funktionen til at fjerne dem. Når alle widgets er blevet fjernet, starter den en ny instans af mainmenu.  
Hvis login fejler, viser den en pop-up besked med ”fejl i brugernavn eller adgangskode”. Efterfølgende fjerner den det indtastede brugernavn og adgangskode, så brugeren kan prøve igen.

**Mainmenu**

**Et billede, der indeholder skærmbillede, tekst, software, Multimediesoftware

Automatisk genereret beskrivelse**

Mainmenu består af 3 knapper, hvor alle bruges til at gå ind på følgende sider (frames), historik, admin og beregneren. Hver knap har sin egen funktion, som minder om hinanden.

**Navigation mellem frames.**

**Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, linje/række

Automatisk genereret beskrivelse**

Vi gør brug af 4 funktioner som gør det samme. Her fjerner de widgets i vinduet og laver en instans af en ny klasse.  
En bedre løsning var kun at have en funktion, og hver gang man kaldte den, bruge et argument for hvilken klasse (frame) der skal bruges.

**Adminmenu**

**Et billede, der indeholder tekst, skærmbillede

Automatisk genereret beskrivelse**

Admin menuen giver brugeren mulighed for at oprette nye brugere. Her bruger vi entrys for at lave et brugernavn og adgangskode. Man skal også vælge om brugeren skal være admin, eller normal bruger. Valget har ikke nogen betydning for vores prototype, men vil have når vi videreudvikler på programmet. Efter der er indtastet brugernavn, adgangskode og valgt rettigheder, kan brugeren bruge opret knappen for at få brugeren oprettet. Når der bliver trykket på opret knappen, bruger den funktionen opret\_bruger.

**Opret bruger**

**Et billede, der indeholder tekst, skærmbillede, software

Automatisk genereret beskrivelse**

****

Den henter brugernavn, adgangskode og brugerens rolle (admin eller normal bruger). Ved oprettelse af brugeren, gør vi brug af værdierne 1 for admin og 2 for standard bruger. Grunden til det, er fordi i vores database, indikere værdien 1 admin og 2 standard bruger. Det første den tjekker er om brugernavnet allerede eksisterer i databasen. Hvis det gør, vises en fejlbesked, og processen stopper. Hvis brugerens navn ikke allerede findes, indsætter den værdierne ind i databasen, og giver en besked om at brugeren er blevet oprettet, og inputfelterne ryddes.