Økonomi

# Indledning

# Planlægning af programmet

## Krav / brugerhistorier

For at kunne lave et program, så kræver det at man sætter sig nogle krav for hvad der skal være med i programmet. Kravene skal også være med til at sætte nogle rammer for hvordan programmet skal fungere når man bruger det som bruger. Dette gør det nemmere for programmøren at programmerer programmet. Jeg har derfor lavet X krav til programmet som skal sikre at programmet fungere som planlagt, samtidigt med at programmet har de funktionaliteter som der vil være passende.

### Krav 1 - GUI

Programmet skal have en GUI. På programmets GUI skal brugeren blandt andet kunne indtaste hvor ofte brugeren får løn, hvor meget brugeren får i løn, hvor mange penge brugeren har brugt/fået siden sit sidste besøg. Brugeren skal til det sidste have mulighed for at kunne lave sine egne kategorier såsom ”Brændstof”, som kan opdele brugerens økonomi i kategorier som brugeren har lavet. De forskellige informationer skal være vist på GUIen og brugerens kontosaldo skal vises med en graf.

Programmets GUI skal starte med at bede brugeren om at logge ind, eller lave en konto. Derefter vil brugeren kunne komme ind og ændre sin økonomi.

### Krav 2 - Administrering af økonomi

Programmet skal kunne administrere brugernes økonomi igennem udregninger. Programmet skal kunne administrere økonomien, ved at udregne hvor mange penge der er tilbage på kontoen, hvis brugeren har brugt penge osv.

### Krav 3 - Visning af økonomi

Programmet skal via programmets GUI kunne vise grafer for hvordan det har gået en brugeres økonomi mens programmet har kørt. Dette skal være i form af en sumkurve eller et histogram. Dette kan hjælpe brugeren med at holdestyr på brugerens økonomi igennem tiderne.

Kravene er bestemte ting såsom tekniske krav med hvor den skal køres henne.

### Brugerhistorie 1 - Login

Denne brugerhistorie kan kun startes når programmet bliver åbnet af brugeren.

* Brugeren åbner programmet
* Programmet registrerer at brugeren ikke er logget ind og viser skærmen for login
* Hvis brugeren har en konto
  + Brugeren indtaster sit brugernavn og adgangskode i de tilhørende felter
  + Brugeren trykker på ”login”
  + Programmet logger brugeren ind og viser hovedsiden for programmet
* Hvis brugeren ikke har en konto
  + Brugeren trykker på ”Har du ikke en konto? Opret en her”
  + Programmet viser ”Opret konto” skærmen
  + Brugeren indtaster sit brugernavn, adgangskode, bekræftelse af adgangskode og e-mail ind i de tilhørende felter
  + Brugeren trykker på ”Opret konto”
  + Programmet tjekker om felterne ”adgangskode” og ”bekræft adgangskode” er ens
  + Hvis felterne er ens
    - Programmet opretter brugerens konto og viser login siden
  + Hvis felterne ikke er ens
    - Programmet laver en fejl og brugeren kan starte forfra

### Brugerhistorie 2 - Tilføjelse og ændring af kontosaldo

Denne brugerhistorie kan kun startes når brugeren er logget ind på programmet, og befinder sig på programmets hovedside.

* Brugeren vælger inputfeltet hvor der står ”Konto saldo” ved siden af
* Brugeren indtaster hvor mange penge brugeren har på sin konto
* Brugeren trykker på ”Gem”
* Programmet tjekker hvad der står i inputfeltet ”Konto saldo”
* Programmet gemmer informationen i databasen
* Programmet opdaterer programmets GUI, så brugerens nye Konto saldo står udenfor teksten ”Konto saldo:”

### Brugerhistorie 3 - Hvor mange penge har brugeren brugt eller fået

Denne brugerhistorie kan kun startes når brugeren er logget ind på programmet, og befinder sig på programmets hovedside.

* Hvis brugeren har fået penge
  + Brugeren vælger inputfeltet hvor der står ”Beløb modtaget” ved siden af
  + Brugeren indtaster det beløb brugeren har fået
  + Brugeren trykker på ”Gem”
  + Programmet tjekker hvad der står i inputfeltet ”Beløb modtaget”
  + Programmet gemmer informationen i databasen
  + Programmet opdaterer programmets GUI, så brugerens nye Konto saldo står udenfor teksten ”Konto saldo:”
* Hvis brugeren har brugt penge
  + Brugeren vælger inputfeltet hvor der står ”Beløb brugt” ved siden af
  + Brugeren indtaster det beløb brugeren har brugt
  + Brugeren vælger den kategori som er passende for brugerens udgift fra dropdown-menuen ”Kategori”
  + Brugeren trykker på ”Gem”
  + Programmet tjekker hvad der står i inputfeltet ”Beløb brugt”¨
  + Programmet tjekker hvilken kategori brugeren har valgt
  + Programmet gemmer informationen i databasen
  + Programmet opdaterer programmets GUI, så brugerens nye Konto saldo står udenfor teksten ”Konto saldo:”
* Hvis brugeren har brugt og fået penge
  + Brugeren vælger inputfeltet hvor der står ”Beløb modtaget” ved siden af
  + Brugeren indtaster det beløb brugeren har fået
  + Brugeren vælger inputfeltet hvor der står ”Beløb brugt” ved siden af
  + Brugeren indtaster det beløb brugeren har brugt
  + Brugeren trykker på ”Gem”
  + Programmet tjekker hvad der står i inputfelterne ”Beløb modtaget” og ”Beløb brugt”
  + Programmet gemmer informationen i databasen
  + Programmet opdaterer programmets GUI, så brugerens nye Konto saldo står udenfor teksten ”Konto saldo:”

### Brugerhistorie

## iterationer

# Udførelse af programmet

## Oprettelse af databasen

## Oprettelse af klasserne til behandling af data

## Brugerhistorie x - Something

Beskrivelse af hvordan brugerhistorien er blevet udført.

# Test af programmet

## Test 1

## Test 2

## Test 3

# Beskrivelse af arbejdsprocessen

## Trello

Jeg har i mit projekt haft brug for at have et overblik over de delopgaver som jeg skal have arbejdet med. Jeg har derfor brugt hjemmesiden trello.com til at hjælpe mig med at holde et overblik over mit projekt, hvilket har sikret mig en let tilgang til at holde styr på hvor meget der mangler osv. Man kan på de forskellige elementer angive hvornår de skal være færdige, så man ender med at bruge tiden fornuftigt.

## Tidsplan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Moduler  (12 moduler  i alt) | 2 | 4 | 2 | 1 | 3 |
| Arbejdsopgaver | Krav specifikation til programmet | Program opsætning | Database opsætning | Test af programmet | Synopsis skrivning |

# Litteraturliste

# Bilag