# Dokumentasjon

Innholdsfortegnelse

## Metode

## Mål

Målet med prosjektet er at jeg skal lage et banksystem. Det vil si å lage en sikker måte for autentisering, gjennomføring av ulike operasjoner som sjekk av saldo, inntak av penger, uttak av penger og overføringer. All informasjon skal lagres i en database, og bli behandlet på en sikker måte.

Programmet skal være en fungerende prototype av et banksystem uten unødvendige funksjoner. Jeg har altså ikke tilpasse systemet til bare ett bruksområde. Det betyr at banksystemet skal kunne brukes til hva som helst. For eksempel kan det bli gjort om til et program som kobler seg til en chat-bot på nettet, og dette programmet kan for eksempel bruke flere funksjoner for autentisering som kan være tatt direkte fra koden min. Jeg har derfor delt opp koden min i flere dokumenter hvor hvert dokument inneholder en funksjon (eller flere dersom de fungerer sammen).

### Delmål

Det er viktig at man skal skjønne hva man skal gjøre når man koder. En bra strategi kan være å dele opp oppgavene i ulike deler for å kunne jobbe strukturert og skjønne når man har gjort noe feil. En ulempe med dette kan være at dersom man ombestemmer seg og vil gjøre om en del som var tidligere, så må man gjøre om alt eller det meste som kommer senere. Bankoppgaven min er et ganske lineært prosjekt, altså jeg vet akkurat hva jeg trenger fra starten. Derfor blir det ingen store endringer underveis, og derfor har jeg delt mine arbeidsoppgaver i ulike deler. Det kan likevel komme noen små endringer som jeg kan implementere på kort tid.

* Jeg skal opprette en database (skriver om tabeller og måten alt blir lagret på senere)
* Jeg skal lage følgende hovedfunksjoner (det blir flere funksjoner senere som blir hjelpefunksjoner for hovedfunksjonene)
  + Oppretting av
* Jeg skal opprette en database

## Krav

## Database

Jeg har valgt å bruke SQL som databasetypen. Grunnen til dette er at SQL er en relasjonsdatabase. Dersom jeg skal lage et bankprogram, så må jeg lagre flere ulike typer data. Eksempler på dette kan være informasjon om kundene, deres kontonummer, saldo, adresse, historikk/loggføring for ulike typer operasjoner som skjer i banken (overføringer, sjekking av saldo osv). Det kan være nyttig å lagre ulik type informasjon i ulike tabeller i SQL, og koble dem sammen med fremmednøkler. Samtidig kan jeg bruke andre funksjoner som primærnøkler for å lagre unike verdier.

Jeg bruker MAMP-applikasjonen for å kjøre en MySQL-server for å kunne lage SQL-databaser lokalt på PC-en.

Dette skal databasen min inneholde:

* Databasen skal ha mulighet for å lagre informasjon om kunden som fødselsnummer, navn, etternavn, postnummer, gatenavn, husnummer, telefonnummer, epostadresse og pinkode. Denne informasjonen skal lagres på en sikker måte, altså informasjonen skal være kryptert. Dette gjøres for at man ikke skal kunne lese av det som står direkte i databasen (altså hvis en fremmed får tilgang til databasen så kan han ikke lese noen ting). Det er back-enden i programmet som dekrypterer informasjonen.
* Kunden skal få mulighet til å opprette flere kontoer under profilen sin. Hver konto er koblet til en profil i den forrige tabellen som jeg nevnte. Hver konto skal ha saldo og kontonummer, og være relatert til fødselsnummeret i den første tabellen med kundeinformasjon. På denne måten kan kunden ha for eksempel en primær konto og en sparekonto.
* På grunn av sikkerhetsårsaker skal alle handlinger på kontoen lagres i historikktabeller. Dette gjøres fordi dersom en fremmed får tilgang til en konto og for eksempel overfører penger, så kan banken spore dette og mulig identifisere hvem som har brutt seg inn på en konto (for eksempel hvis en fremmed har fått tilgang til en konto og overført alle pengene til sin konto, så blir det lett å spore dette). Hver tabell inneholder ulik informasjon avhengig av hva som ble gjort. Dersom brukeren sjekker saldo blir det lagret kontonummer, saldo og tidspunkt. Dersom brukeren overfører penger er det lurt å lagre senderen og mottakerens kontonummer, sum, tidspunkt og saldo før/etter transaksjonen.
* All informasjon i databasen skal være kryptert (grunnene står lenger opp).

## Feilsøking og debugging

Jeg har hatt flere feil når jeg skulle sortere filene

## Versjonshåndtering

Koden jeg har brukt i prosjektet er en kode som jeg har skrevet på slutten av Vg1. Jeg har forbedret koden når jeg startet med innleveringsoppgaven på Vg2, men jeg får likevel ikke med hele prosessen. Jeg har ikke brukt GitHub når jeg skrev koden på Vg1. Derfor har jeg ikke håndtere ulike versjoner av prosjektet.

Dersom jeg hadde hatt med versjonshåndtering, så hadde jeg brukt GitHub og brukt commits for å beskrive hva jeg har endret på. Deretter hadde jeg pushet prosjektet til repositoryen.

## Informasjonssikkerhet

## Secure coding