

Innledning

Problemstilling

Design et system som skal oppføre seg som et utlånssystem for biblioteket. Det skal være mulighet for å legge inn bøker og filmer i listen av ting som kan lånes ut.

Det skal være mulig å markere noe som lånt ut eller levert inn. Koden skal være lesbar og ryddig, ikke glem kommentarer

All bruk av KI skal være tydelig beskrevet, og logger må bli lagt ved som vedlegg. KI skal ikke bruker for å løse oppgavene, men kan brukes som diskusjonspartner. Ingenting skal kopieres fra KI inn i rapporten. Generell forståelse vil bli sjekket under siste samtale.

1. Implementer funksjoner for å legge inn i biblioteket, og å låne ut bøker. Implementer funksjonene som gjør det mulig å legge til bøker og filmer fra kommandolinjen (10%)
2. Ta en avgjørelse av hva som skjer i tilfelle duplikater blir lagt til. Implementer en funksjon for å finne objektet fra biblioteket ved å bruke det fulle navnet. Legg til mulighet for å låne og å levere bøker basert på det fulle navnet. (10%)
3. Lag to nye funksjoner for å liste alle bøker og filmer som er tilgjengelig. De skal være i Library klassen. (10%)
4. Legg til støtte for å ha flere av en bok eller film i systemet til utlån av gangen (10%)
5. Legg til brukere, for å kunne vite hvem som har lånt hva fra biblioteket (20%)

Bruk av KI i oppgaven

For utvikling av programvaren vil KI bli brukt rent som samtalepartner for å ideløse funksjoner, legge til enkelte funksjoner. KI vil bli brukt som læringsgrunnlag for programfunksjoner som jeg enda ikke har lært, men koden vil bli skrevet selv basert på forståelsen jeg danner fra KI sine svar.

Hvis jeg ikke forstår funksjonen til slutt vil funksjoner forenkles eller fjernes fra oppgavebesvarelsen.

Plan for utvikling og programmering

Jeg starter med å lese meg opp på oppgaveteksten og den leverte biblioteksfilen "library-code.py" for å få en forståelse for grunnlaget som er gitt for oppgaven. Deretter vil jeg starte med å bestemme hvilke funksjoner som skal implementeres først.

Her går jeg igjennom tankene og prosessen rundt oppgaveforståelsen og prøver å redegjøre for det jeg skal gjøre videre i oppgaven.

Ferdig definerte klasser

Vi har 4 klasser som ikke er ferdig definert. LibraryItem er en klasse for å holde styr på det vi har i biblioteket vårt og satusen på det vi har (utlånt og innleveringsdato). Det skiller på bøker og filmer, bøker og filmer er egne klasser som arver egenskapene fra LibraryItem, her må det tilpasses ut ifra det vi trenger for å skille mellom bøker og klasser.

Klassen for Library er for å holde styr på hva som er i biblioteket vårt. Her ser jeg for meg at vi trenger å lage en database som lagres og hentes tilbake når programmet stenges ned.

Med første øyekast vil jeg se for meg at vi trenger noen funksjoner for å liste opp objekter som ikke er utlånt, objekter som er utlånt, status på hvilke som skal leveres innen en uke som påminnelse og hvilke som ikke er levert tilbake når fristen er utgått.

Det mangler definisjoner for hvem det er som har lånt ut objektet, her må det lages en database med ID, navn og kontaktinfo. Vi holder informasjonen kort for å forenkle implementeringen.

Tanker rundt oppgaven

Det er lagt opp til at hele programmet skal fungere i kommandolinjen, dette kan være vanskelig å holde styr på så ett enkelt grafisk grensesnitt ville vært fint, men det får komme hvis en får tid.

Databasen som bygges er ikke lagret noe sted, som vil si at for hver kjøring så må en legge til alt på nytt. Her må det lages en enkel database for å kunne hente inn igjen det som lages. Som en enkel start så kommer jeg til å lage enten en CSV fil eller en JSON fil som database, men først fokuseres det på å få alle grunnfunksjonene på plass.

Som første mål så ønsker jeg å implementere følgende:

- Ferdigstille alle klassene med den informasjonen jeg ser for meg jeg trenger.
- Legge inn bok og film manuelt basert på å hente info direkte fra en nettside som <https://depotbiblioteket.no/> for bøker og <https://www.imdb.com/> for film. Identifikasjon brukes fra de sidene og så skal det være mulig å hente fra API senere.
 - ISBN brukes som identifikasjon for bøker
 - IMDB Id som tt0120737 brukes for film
 - Ønske om å hente informasjon automatisk fra API senere
 - film: [OMDb API - The Open Movie Database](#)
 - bøker:
- Liste opp objekter som er utlånt
- Liste opp objekter som ikke er utlånt
- Liste opp objekter som skal leveres om en uke

Veien videre

Jeg lager ett repository på Github for å holde styr på kildefilene underveis og endringene som gjøres. Målet er å lage kode for en funksjon, teste den og deretter pushe til Github før jeg fortsetter på neste funksjon.

Før jeg starter implementasjon av noe kode så skal jeg lage ett flyt diagram for hver enkelt klasse som fargelegges med flyten for hver funksjon. Disse lages med canvas i ett program som heter Obsidian hvor jeg noterer ned alt som skjer underveis. Dette skal i ettertid eksporteres til ett eget dokument som ferdig dokumentasjon på oppgaven.

Planlegging av å legge til brukere

Planlagte egenskaper:

- Lånekort ID
- Fornavn
- Etternavn
- Epost

Egen klasse for håndtering brukere og egen for databasen.

Først blir implementering av å legge til og fjerne brukere fra systemet. Brukere må bli en del av eksporteringen og innlasting av bibliotek. Implementering av hva som er lånt ut av hvem må forberedes for at en kan ha flere av samme bok. Plan for implementering blir å lage en dictionary av borrowed by hvor hver bok har en UUID som linkes mot lånekortet til brukeren.

due_date attributt på LibraryItem må endres til å også linke mot UUID på aktuell bok.

På sikt må vi ha følgende funksjoner:

- List opp alle brukere
- List opp hva brukeren har på utlån med innleveringsdato

Hoveddel

Jeg har laget ett bibliotek hvor en kan låne bøker og filmer. Biblioteket har mulighet for brukere å registrere seg inn og får da ett utlånskort ID som skal kunne brukes på sikt for å enkelt scanne og låne i stedet for å søke etter navn.

Jeg startet med grunnfunksjonaliteten i biblioteket ved å oppdatere en og en klasse som var nødvendig for å starte testing av resterende funksjoner. Jeg traff relativt greit på egenskapene, utenom GUID, den ideen skrapte jeg, da jeg følte at det kompliserte arbeidet en del og det hadde jeg ikke tid til og har ingen erfaring med å sette det opp.

Ved start av testing så satte jeg opp mindre egenskaper å legge inn for å raskere legge til bok / film for å teste, det gjorde ting litt lettere.

Som start så begynte jeg med å tenke igjennom prosessene som måtte lages, hvilke egenskaper jeg trengte og begynte med flytdiagram. Dette var veldig bra hjelp til å få en oversikt over hvordan koden kom til å henge sammen når alt var ferdig. Det fungerte bra i noen tilfeller og andre ganger så overtenkte jeg litt og gjorde arbeidet litt for vanskelig for meg selv.

Som for eksempel med implementering av brukerdatabasen, der hadde jeg ett ønske om å

legge inn utlån av flere bøker/filmer samtidig av forskjellige personer, det skulle jeg planlagt for tidligere da dette betydde å skrive om store deler av koden og det hadde jeg ikke tid til. Derfor ble det implementert med kun en utlåner om gangen slik at jeg hadde ett ferdig produkt å levere inn.

Avslutning

Hva har jeg lært?

Å defaulte til false verdi når f.eks listene er tomme er kanskje ikke det beste designvalget. En kunne kjørt sammenligning mot lengden på listen i stedet slik at forventet type alltid er riktig.

Å planlegge full funksjonalitet i koden er veldig nytt, da slipper en å skrive om masse funksjoner i etterkant for å få det til å fungere. Dette traff jeg på når jeg skulle utvide funksjonaliteten for antall bøker, det viste seg å være vanskelig å holde styr på når det kunne være flere lånetagere pr. oppføring i biblioteket. Derfor ble denne funksjonaliteten bare ren utlåning av en person.

Lært masse i forhold til bruk av klasser og dictionaries, men det er fremdeles en del jeg tror jeg mangler tror jeg, er noen funksjoner som f.eks max funksjon som jeg ville trodd kunne være enklere, men fikk det ikke helt til. Endte til slutt opp med å lage en loop i stedet, det er sikkert ikke så effektivt hvis biblioteket blir veldig stort(?).

Det er veldig viktig også å skrive kommentarer underveis på hva koden gjør og hva en har tenkt for å lette feilsøking senere.

Jeg skulle gjerne hatt litt mer introduksjon på funksjonene som:

- Hva gjør denne funksjonen
- Hva er forventet input
- Hva er output

Ønskefunksjoner

- Lokasjon på boken i biblioteket, område, hyllenummer f.eks
- Liste opp objekter som skal leveres om en uke
- Kan søkefunksjonen ikke være case sensitive?
- Utvide lånefunksjonalitet til å spørre etter lånekort ID
- Utvid søkefunksjon til å spørre om du vil låne en av de funnede objektene
- Lage ett faktisk GUI på biblioteket og ikke bare ren kommandolinje
 - I stedet for å utvide søkefunksjon kan en da lage en GUI hvor en trykker på bok som er funnet og velger lån ut til
- Liste opp hvilke bøker/filmer en bruker har på utlån.
- Implementere API for henting av bøker og filmer som en kan legge til i biblioteket

Log fra programmering

2025.12.04

- Oppdatert klassene LibraryItem, Book og Movie med planlagte attributer

2025.12.05

- Oppdatert klassen Library med planlagt funksjonalitet
- Lå merke til at klassen movie og book kun inneholder de nye attributene og ikke attributene for LibraryItem klassen. Dette ble også lagt inn.
- Gått igjennom og testet de forskjellige funksjonene for funksjon 1 til 6 i menyvalgene.
- Jeg ser at det er kanskje noen funksjoner som kunne vært mer generelle ved å sjekke hvilken klasse en jobber med. Mulig en kan se på det på sikt for å forenkle programmet noe.

Plan for videre arbeid er å opprette søkefunksjon for å finne ting i biblioteket og vise statusen på den samt legge til hvem som låner boken.

For øyeblikket er funksjon for å legge til bok og film strippe ned for å spare tid ved testing.

2025.12.07

- Lagd funksjon for å søke i biblioteket.
- Syntes oppsettet for å liste ut objektene ble litt rotete etter søk, så ønsket å sette det opp i tabellform for å vise mer informasjon på en gang. Før jeg begynte å lage noe selv så søkte jeg etter ett eksisterende bibliotek og fant Texttable. Nå blir det å implementere Texttable i allerede ferdige definisjoner og rydde litt i koden.
- Jeg har lagt merke til at søkefunksjonen er case sensitiv. Det ønsker jeg å endre på, skriver det ned som ett aksjonspunkt.
- While loop på funksjoner for å sjekke om valgt alternativ er riktig er endret til å bruke in range funksjonalitet.
- Lagt inn ønske om å utvide søkefunksjon sånn at en kan låne bok eller film direkte basert på resultat
- Endret originalt menyoppsett til å bruke Match funksjon i stedet for if elif.
- Oppdatert list tilgjengelige filmer og bøker til å gi alle bøkene og ikke kun de som er tilgjengelig for utlån. Utvidet funksjonen med en trigger for å liste alt eller kun tilgjengelig for utlån for å kunne bruke den til begge formål. Kunne også flyttet funksjon for å liste ting som skal leveres tilbake i samme funksjon, men det får bli senere.
- Lag til funksjon for å lagre bibliotek til fil og hente den inn igjen.
- Startet implementasjon av brukere. Sliter med iterasjon av databasen. Dette må ses på i morgen.

2025.12.08

- Fant feilen med brukerdatabasen. Jeg hadde kopiert funksjonen for å liste opp brukere i databasen fra biblioteksklassen. Der var det en funksjon som lagde en liste, så i brukerdatabasen hadde jeg lagd en liste på en liste som: [users.values()] noe som ble

feil, da user.values() er en liste fra før. Så ved å fjerne [] så fungerte itereringen av listen som tiltenkt.

- Ferdigstilt add user og delete user funksjon. Slet litt med å iterere igjennom dict_values, dette må jeg se på senere om det er mulig å bruke en max() funksjon for å finne høyeste verdi på bibliotekskort. For nå så bare loopet jeg igjennom for å finne høyeste kortnummer for å lage ett unikt nummer. Alternativt, endre funksjonen til at den tar første unike verdi. Dette er gjerne bedre i tilfelle en bruker blir slettet.
- Startet på programmering av linking av hvem som låner hvilket objekt i bilioteket. I første omgang så jeg for meg at jeg kunne brukt en liste som endre seg for å klargjøre for flere av samme bok. Dette så jeg fort gjorde at jeg måtte skriv om store deler av eksisterende kode og bestemte meg for å først implementere enkel linking av kun en bok. Dette vil si at å implementere antall av samme bok eller film senere kan bli vanskelig, eller så må dette implementeres på en annen måte.
- Utvidet userDatabase til å søke igjennom brukere for å finne hvem som skal låne.
- Implementasjon av hvem som låner er fullført, la til søkefunksjon for å finne bruker som skal låne bok som så sendes videre til borrow funksjonen.
- Borrow funksjonen ble utvidet til å sende videre hvem som skal låne til "Find books/movies borrowed", for å ikke ødelegge denne funksjonen andre steder i koden så gjorde jeg argumentet userFullscreen optional ved å gi den en standardverdi. Da vil funksjonen fortsatt fungere andre steder i koden.

2025.12.09

- Laget en funksjon for å liste opp menyvalg. Dette ble implementert i hovedmenyen, men ikke de andre. Burde bli gjenbrukt flere plasser.
- Rettet litt på load library funksjonen sånn at koden kan kjøre som en executable fil også.

Funksjoner som må lages

- Ferdigstille klasse som er predefinert
 - LibraryItem
 - ID i bibliotek GUID
 - Tittel → Predefinert
 - Beskrivelse
 - Antall (legges til, vent med funksjonalitet)
 - Sjanger
 - Publiseringsår
 - Språk
 - Utlånt av (legges til, vent med funksjonalitet)
 - Utlånsdato
 - Innleveringsfrist → Predefinert
 - Lokasjon (legges til, vent med funksjonalitet)
 - Funksjon for om den er lånt ut
 - Funksjon for å låne

- Funksjon for å returnere
- Book
 - Author → Predefinert
 - isbn → Predefinert
 - Antall sider
- Film
 - Direktør → Predefinert
 - IMDB ID → Predefinert
 - Lengde spilletid
 - IMDB Rating
 - Tekstet språk (legges til, vent med funksjonalitet)
- Library
 - Lage lister for hele biblioteket med egen ID fra LibraryItem → Predefinert
 - Lage funksjon for å liste ut kun bøker
 - Lage funksjon for å liste ut kun filmer
 - Lage funksjon for å liste ut utlånede bøker
 - Lage funksjon for å liste ut utlånede filmer
 - Funksjon for å teller antall i biblioteket, antall bøker, antall filmer (legges til, vent med funksjonalitet)
 - Skal inneholde statistikk for: antall tilgjengelig for utlån, antall som er utlånt i hver kategori.
- Finn en gjenstand i biblioteket – Søkefunksjon
- Utvide utlåningsfunksjon til å ta med seg person som låner
- Utvide opplistningsfunksjon til å gi mer en bare tittel.
- Lage klasse for håndtering av brukere i systemet
 - Lage flytdiagram og ønskede attributer til bruker
- Liste opp objekter som er utlånt
- Liste opp objekter som ikke er utlånt
- Lagre informasjon til database i JSON eller CSV – Brukte Pickle i stedet