

# Oppgave 1: plan for filstruktur og klassebibliotek

«Lag en pdf-fil som viser en plan for filstrukturen i systemet som skal utvikles i prosjektet. Dette inkluderer en plan for hvilke include-filer du kan dra nytte av og en plan for ditt eget klassebibliotek. Denne oppgaven skal gjøres sammen med en medstudent, men leveres som en del av den individuelle innleveringen.»

## Introduksjon

Gruppen har som mål å lage et bookingsystem for veiledning som vi har valgt å kalle for MentorMate. Her skal studenter og læringsassistenter kunne legge til når de ønsker å ha veiledningstid, og koordinere dette på tvers av LA og studentens timeplaner.

## Includes

Til prosjektet kan vi gjenbruke kode som beskriver gjentakende elementer som «meny», «header» og «footer» for sider, samt gjenbrukbar funksjonalitet til «message» klassen, etc. En ting å være obs på ved bruk av «includes» er at «coupling» økes i systemet som følge av dette. Noen ganger kan dette være uønsket, ettersom endringer på en «includes» kode seksjon fører til endringer alle steder denne er blitt gjenbrukt. Siden det er to typer brukere i systemet kan det av den grunn være anstendig å unngå bruk av «includes» for kode som differensierer mellom brukertypene.

## Filstruktur

Implementeringen av en effektiv og organisert filstruktur er en viktig komponent i produktutviklingsprosessen. I mange tilfeller kan det forekomme som en utfordrende oppgave å etablere en optimal filstruktur på forhånd, grunnet de varierte og ofte uforutsette utfordringene som kan oppstå i løpet av prosjektets livssyklus. Dette kan potensielt medføre at den opprinnelig planlagte strukturen ikke lenger er egnet til å tilfredsstille prosjektets dynamiske krav og behov.

Til tross for dette, er det av vesentlig betydning å ikke undervurdere verdien av en foreløpig planlagt filstruktur. Slik en plan fungerer som en grunnleggende rettesnor som prosjektteamet kan følge, og tilbyr en strukturert tilnærming til organiseringen av filer og data, selv i lyset av potensielle revisjoner og justeringer som kan være nødvendige ettersom prosjektet utvikler seg.

I denne konteksten, har gruppen valgt å etablere en forhåndsbestemt filstruktur, vel vitende om at fleksibilitet og tilpasningsevne vil være avgjørende når uforutsette variabler presenterer seg. Det er med denne forståelsen at vi forventer og er villige til å revidere og tilpasse filstrukturen i takt med prosjektets utvikling og de uforutsigbare utfordringene og mulighetene som måtte oppstå.

I et forsøk på å være så forberedt som mulig på de uunngåelige endringene, prioriterer gruppen en proaktiv tilnærming. Det innebærer en kontinuerlig evaluering og, om nødvendig, revisjon av

filstrukturen for å sikre at den til enhver tid er optimalt tilpasset prosjektets krav og bidrar til en effektiv produktutviklingsprosess.

Gruppen velger å følge prinsippene til “(semi)-clean architecture” og “MVC” for selve filstrukturen. Merk at enkelte filnavn bare er foreløpig og kan endres.

Under ligger en plan over filstrukturen til systemet:

```
project/
├── core/                                // Core business logic and entities
│   ├── entities/
│   │   ├── user.php
│   │   └── booking.php
│   ├── use_cases/                       // Application-specific business rules
│   │   ├── user_manager.php
│   │   └── booking_manager.php
│   └── interfaces/                     // Interface adapters (input/output)
│       ├── user_repository_interface.php
│       └── booking_repository_interface.php
├── infrastructure/                     // Frameworks, drivers, and tools
│   ├── database/
│   │   ├── db_connector.php
│   │   ├── user_repository.php
│   │   └── booking_repository.php
│   ├── email/
│   │   └── email_service.php
│   └── utilities/
│       └── utilities.php
├── application/                       // User interface and presenters
│   ├── controllers/
│   │   ├── user_controller.php
│   │   └── booking_controller.php
│   ├── views/
│   │   ├── user/
│   │   │   ├── profile.php
│   │   │   └── login.php
│   │   ├── booking/
│   │   │   ├── create.php
│   │   │   └── index.php
│   │   └── messages/
│   │       ├── send.php
│   │       └── inbox.php
│   └── models/                         // Data representations
│       ├── user_model.php
│       └── booking_model.php
├── assets/                             // Front-end resources (CSS, JS, images)
│   ├── css/
│   ├── js/
│   └── images/
└── index.php                           // Entry point to the application
```

Formatert av OpenAI (2023).

## Klassebibliotek

Dette er en foreløpig versjon av klassebiblioteket, med forbehold for endringer.;

### User

- Properties:
  - userID
  - username (email)
  - password
  - role (student/teaching assistant)
- Methods:
  - login()
  - logout()
  - updateUser()

### Booking

- Properties:
  - bookingID
  - studentID
  - teachingAssistantID
  - timeSlot
  - status (confirmed/cancelled)
- Methods:
  - createBooking()
  - cancelBooking()
  - bookingConfirmation()
  - bookingReminder()

### Calendar/Bookings

- Properties:
  - teachingAssistantID
  - date
  - timeSlot
  - availability
- Methods:
  - viewAvailability()
  - updateAvailability()
  - addAvailability()
  - removeAvailability()

### Message

- Properties:
  - senderID
  - recipientID
  - subject
  - message
- Methods:
  - sendMessage()
  - viewMessage(\$id)

- `viewChat()`

## Referanseliste

OpenAI. (2023). ChatGPT-4 (16. oktober versjon) [Stor språkmodell].

<https://chat.openai.com/share/a5182e73-88b9-4a73-9738-e545171a5bf2>