

Kravspec:

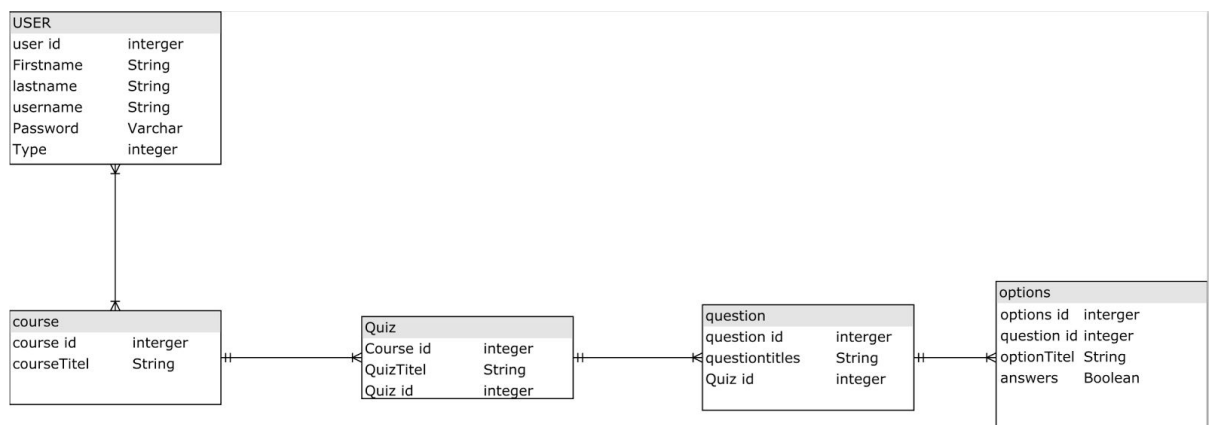
1. Server skal validere login på et hashet password
2. Server skal kunne oprette en bruger (User"student"/admin"teacher")
3. Admin skal kunne oprette en quiz
4. 2 typer af brugere; user og admin
5. Server skal udskille et API som gør det let at udarbejde klienter, som trækker på serverens funktionalitet. API'et skal have følgende funktionalitet:
 - En gæst uden en konto skal kunne oprette en ny konto.
 - En bruger kan logge ind og ud.
 - En bruger skal kunne tilgå samt besvare en quiz.
 - En bruger skal kunne slette sin konto.
 - En admin skal kunne oprette og slette quiz.
 - En admin skal kunne oprette en ny admin.
6. Server skal kunne udskille et interface som gør det muligt for admin at oprette og slette quizzer

Klient:

- Klienten skal kunne oprette en bruger (user) og herefter tilgå programmet med denne.
- Klienten skal udstille et login vindue for brugeren og give brugeren mulighed for at logge ud igen.
- Klienten skal kunne oprette og slette quizzer (admin)
- Klienten skal kunne tilgå quizzer og besvare dem. (user)
- Klienten skal ud fra brugertype kunne vise forskellige menuer for admin og bruger.

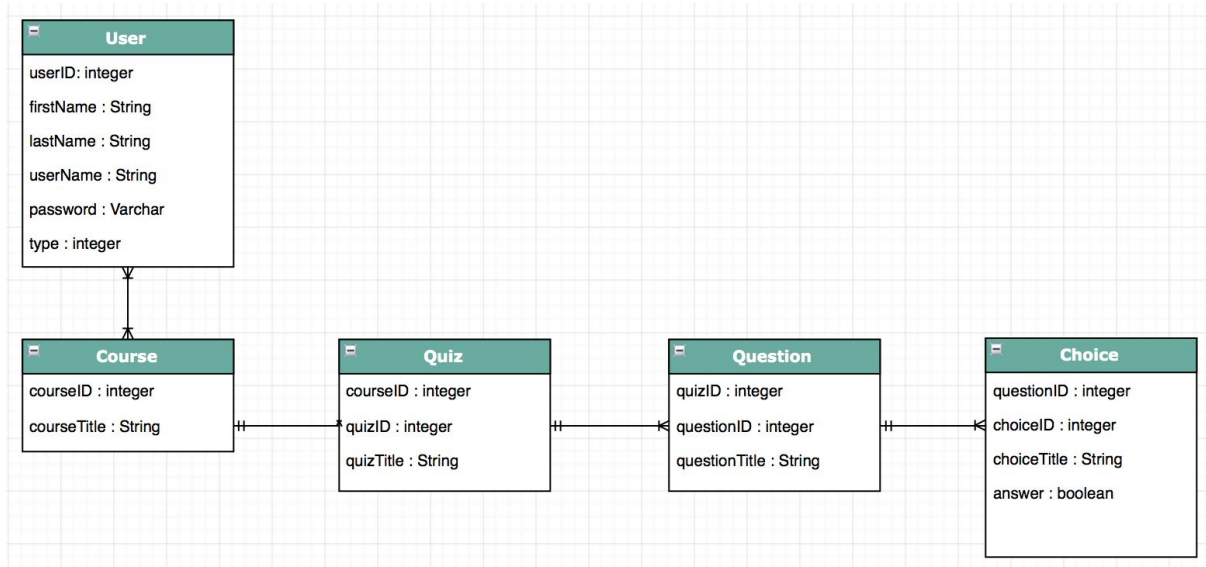
Datamodel

Har forsøgt at lave følgende ER-diagram: (ER-diagram er blevet godkendt)



Ikke nødvendig at koble til en student til et semester (men et felt på et fag)

Udkast til opdateret ER diagram (opdateret for konsistens)



I spg – hvad skiller svarmuligheder ad. Hvad er spgtekst? – lav relationstabel til svar/ lav svartabel.

Hvis multiple choice ??

Forelæser til fag og ikke quiz

Hvor langt er vi til tirsdag? – få mysql

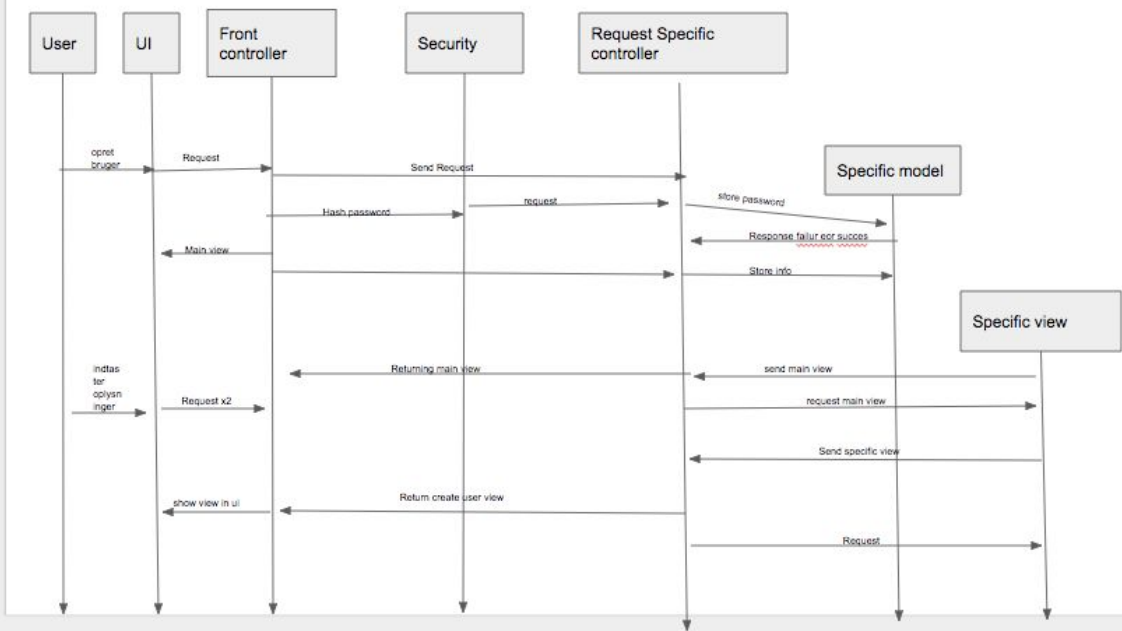
Grundlæggende modelklasser skal være lavet ’

Torsdag å hvor langt er vi 5 min præsentation

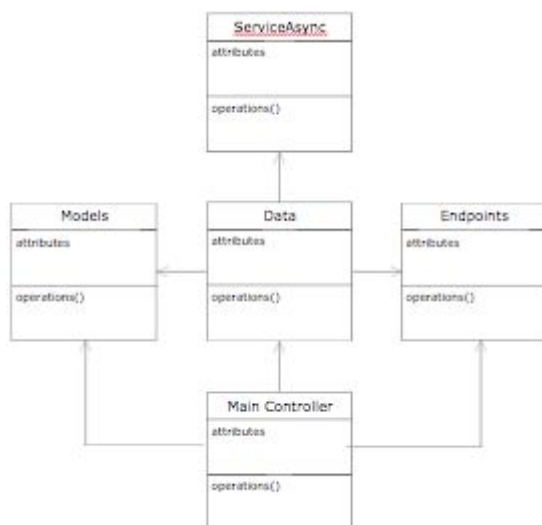
Arkitektur

Sekvens diagram

Gæst vil oprette en bruger



Klasse diagram



Main - Daniel, Rasmus, Olivia, Cecilie, Srdan

- ResponseHandler
- MainController
- Controller
- DB wrapper (muligt at skifte fra en til en anden database → den oversætter. eks. select*from.....)

Models - Koch (Github), Mads, Mads-emil, Ronnie, Jonathan

- User (type1, type2)
- Course
- Quiz
- Question
- Choice

Endpoints/Views - Saim, Imran, Mansur, Nikolaj

- Endpoint 1 (Log in-view)
- Endpoint 2 (Logged in-view)
- Endpoint 3 (MainView for type 2)
- Endpoint 4 (Fag)
- Endpoint 5 (Spørgsmål)

Database - Daniel

- UserTable
- CourseTable
- QuizTable
- QuestionTable
- ChoiceTable

ServiceAsync

- Håndterer alle db-kald

Variabler/Metoder

User :

Course:

Quiz:

Question:

Choice: