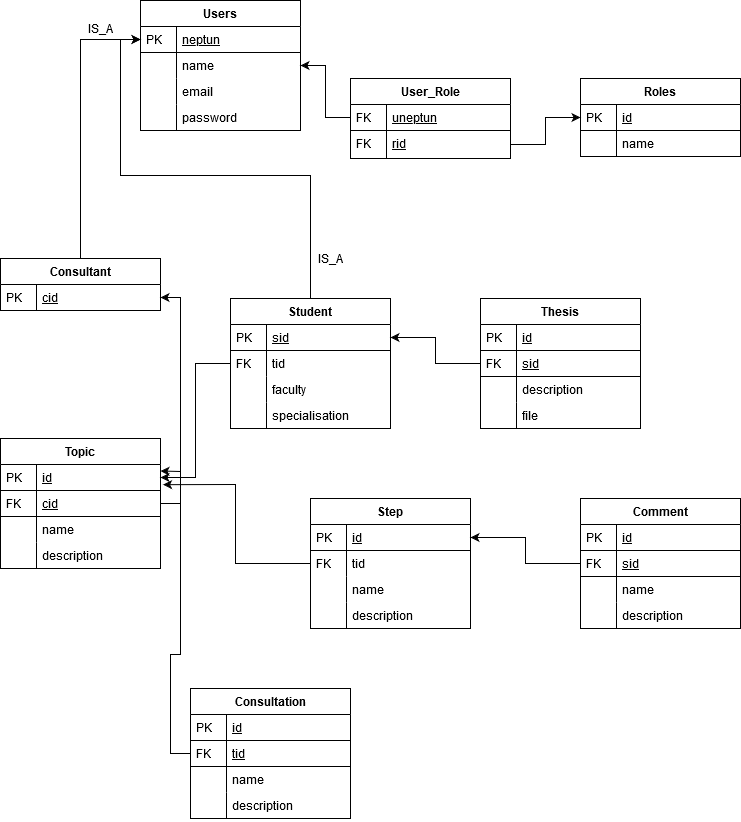
**RENDSZERTERV (kb 3 oldal)**

**ER model**

**Relációs adatmodel**



**Osztály diagram**

A képen képernyőkép látható

Automatikusan generált leírás

CONTROLLER diagram  
SERVICE diagram

**Osztályok és metódusaik:**

**Model:** Az úgynevezett model osztályok adják a program gerincét és sok esetben reprezentálják az adatbázis tábláit is.

* User: Az összes felhasználó közös ős osztálya, ami az alap adatokat és metódusokat tartalmazza egy felhasználóval kapcsolatban.
* Role: A felhasználói szerepkörök osztálya.
* Konzulens(User): A User osztályból származtatott osztály ami a konzulensekre specializálódott
* Student(User): A User osztályból származtatott osztály ami a hallgatókra specializálódott
* ThesisManagerObj: Egy olyan közös ős osztály mely sok más osztály számára kiinduló pont a programban.
* Topic(ThesisManagerObj): A szakdolgozati téma osztálya.
* Step(ThesisManagerObj): A témákhoz köthető lépés osztály.
* Consultation(ThesisManagerObj): A témákhoz köthető konzultáció osztály.
* Comment(ThesisManagerObj): A lépésekhez köthető komment osztály.
* Thesis(ThesisManagerObj): A szakdolgozat osztálya.

**Service:** A service vagyis szolgáltatás osztályok biztosítják a felhasználók számára a különböző szolgáltatásokat, rajtuk keresztül érhetőek el és ők valósítják meg lényegében a program funkcióit.

ConsultantService

* createTopic(): A konzulenshez tartozó téma kiírására, létrehozására szolgáló metódus.
* deleteTopic(): A konzulenshez tartozó téma törlésére szolgáló metódus.
* acceptStudent(): Elfogadja a hallgató jelentkezését az adott témára.
* declineStudent():Elutasítja a hallgató jelentkezését az adott témára.
* acceptStep(): A hallgató által elkészített és kérelmezett lépés elfogadása(kommenttel együtt).
* declineStep(): A hallgató által elkészített és kérelmezett lépés visszautasítása(kommenttel együtt).
* acceptThesis():Szakdolgozat elfogadása, ha a hallgató benyújtotta a kérelmet.
* declineThesis():Szakdolgozat visszautasítása, ha az nem megfelelő vagy javításra szorul valamiben.

StudentService:

* getters: Biztonságos hozáférhetőséget biztosít az osztáy adattagjaihoz.
* setters: Az osztály adattagjai állíthatóak be a setterseken keresztül.
* chooseTopic(topicId): Kérelem az adott téma kiválasztására a konzulens felé(komment a kérelemben ki,miért, mit tud stb)
* changeTopic(fromId, toId):Választott téma megváltoztatása. Egy másik választása.
* modifyStep(): A lépés elkészítésére vonatkozó tartalom módosítása.
* stepAcceptRequest(): A témához tartozó aktuális lépés beküldése elfogadásra.
* uploadThesis(): Szakdolgozat feltöltése adatbázisba.
* sendThesisAccept(): Szakdolgozat elküldése elfogadásra.

TopicService

**Controller:** A controller osztályok felelősek az irányításért.

**Repository:** A repository osztályok az adatbázis és a programkódunk között teremtik meg a hidat.

**Configuration:** A program konfigurációjára szolgáló osztályok