

Webes Alkalmazások Fejlesztése

Dokumentáció az 1. Beadandó feladathoz

Huszár Ádám

X4YWL8

1. Kitűzött feladat:

Készítsünk kliens-szerver rendszert, amellyel egy magánkórház bejelentkezési naptárát, valamint a betegek kórlapjának kezelését tudjuk elősegíteni.

1. részfeladat: a páciensek időpontokat a webes felületen keresztül kérhetnek.

- A páciens regisztrálhat (név, cím, telefonszám, azonosító, jelszó megadásával), majd bejelentkezhets/kijelentkezhets. Bejelentkezést követően láthatóak a kórház orvosai, valamint azok szakterületei

(pl. sebészet, bőrgyógyászat, toxikológia). Az orvosok listája legyen az névsor szerint rendezve. Egy orvost kiválasztva látható a hozzá tartozó foglalási naptár (az aktuális dátumtól), ahol láthatjuk a szabad, illetve foglalt időpontokat.

- Egy páciens több időpontot is foglalhat, a naptárban látja a saját foglalásait, illetve más páciensek további foglalásait, de utóbbiak esetén csupán a „foglalt” állapotot, a másik páciens adatait nem.

- Az időpontfoglalás kétféle módon történhet: az orvos vagy a szakterület kiválasztásával. Ezt követően

ki kell választani egy szabad időpontot (minden nap 9 és 17 óra között, csak egész órára), a második

esetben az összes adott szakterülettel rendelkező orvos szabad időpontjai közül. Ilyenkor, ha a választott időpontban több orvos is elérhető, az aktuálisan legkevesebb előjegyzéssel rendelkező kerüljön

automatikusan kiválasztásra. Minden foglalás 1 óra időtartamú. A bejelentkezésekhez legyen lehetőség

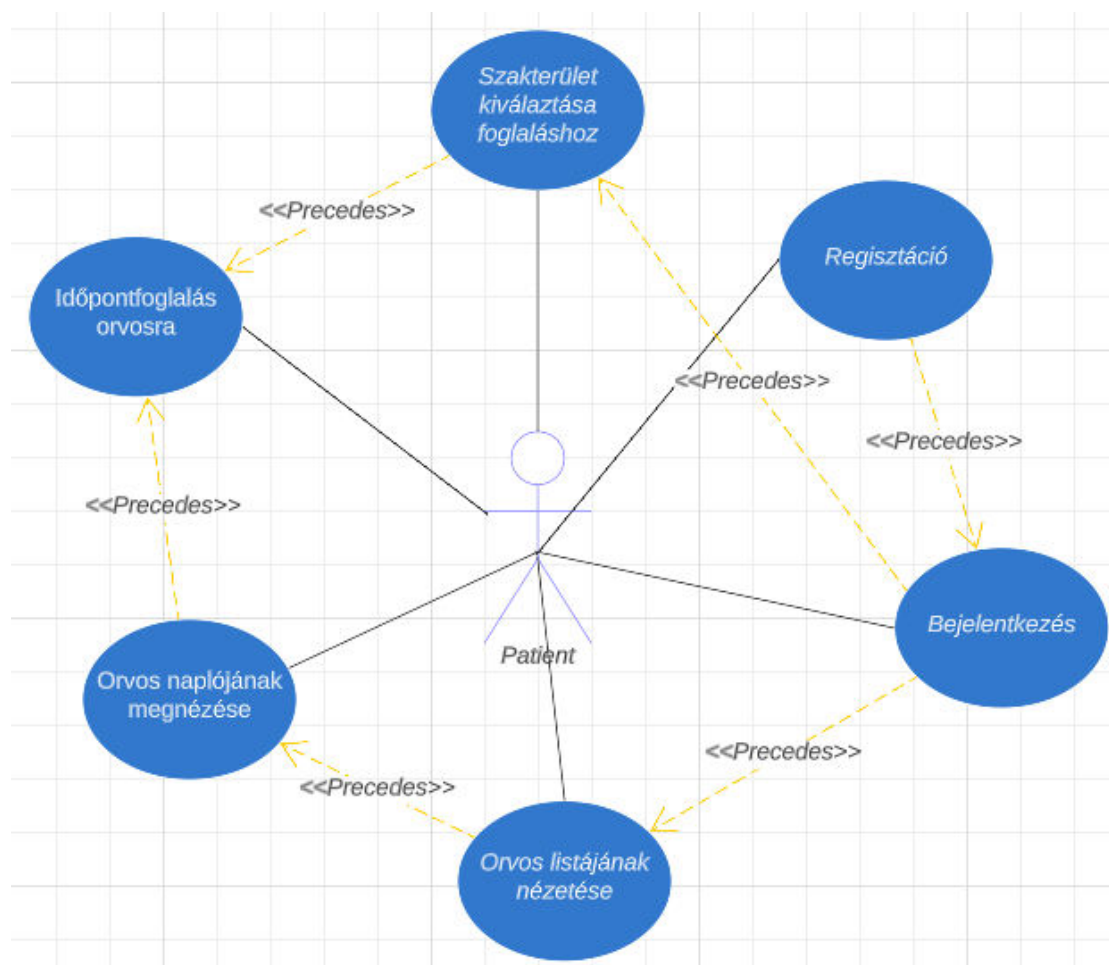
szöveges megjegyzést is írni. Természetesen csak jövőbeli időpontok foglalhatóak.

- A foglalásokat később (de még az időpontja előtt) lehet törölni is.

2. Elemzés:

A webes felület igénybevételéhez páciensként be kell jelentkezünk (illetve szükség esetén regisztrálnunk). Ezután a páciens illetve a felhasználó a doktorok listáját látja névleg rendezett sorrendben illetve a nevükre kattintva láthatják az adott doktorhoz tartozó foglalási naptárat. A naptárban adott hétre jelenik meg időpont (Bővített funkció lehet a lapozás jövőbeli heteken lévő időpontokra). A naptárban minden cella tartalmaz egy opcionális komment mezőt, illetve az adott időpontra tartalmazó lefoglaló gombot, ha üres az időpont. Ha nem üres és mi foglaltunk akkor egy törlés gombot. Múltbeli időpontok is megjelennek és jelzésre kerülnek, azokkal nem lehet interakcióba lépni. Specializáció avagy orvos szakterület szerinti foglalás is lehetséges. Ilyenkor a felhasználó kiválasztja a főoldalon lévő specializáció fajtát és annak az orvosnak a naptára kerül kiválasztásra aki rendelkezik az adott specializációval és azok között az orvosok között a legkevesebb előjegyzéssel rendelkezik.

3. Felhasználói esetdek diagrammja



4. Tervezés

4.1. Programszerkezet

A Webes felületet ASP.NET Core MVC keretrendszerben megvalósítjuk meg.

Tartalmaz osztálykönyvtárakat (Class), kontrollereket (Controller), valamint nézeteket (View). A nézetoldalak megosztoznak egy közös nézeten (_Layout.cshtml). Az adatokat

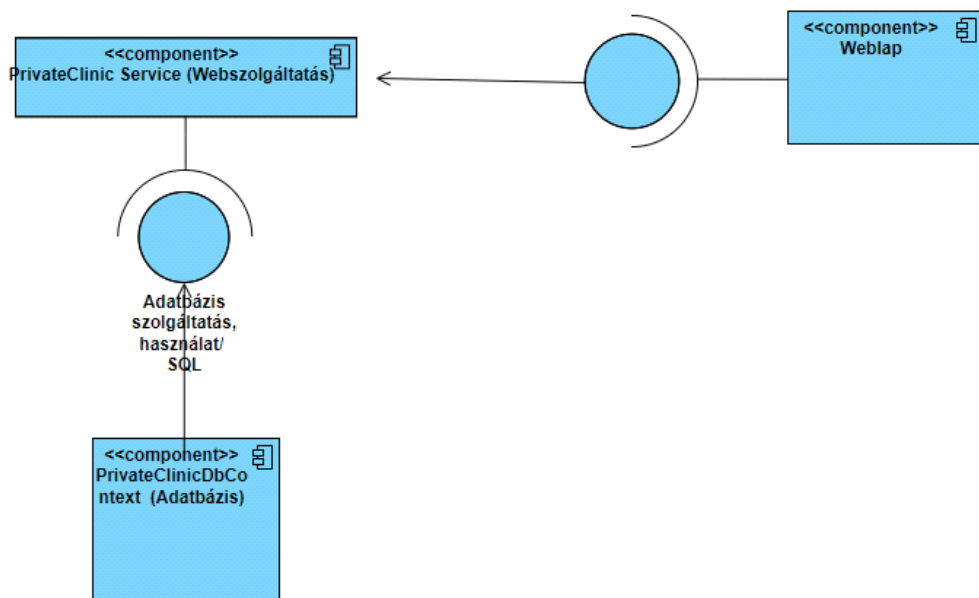
adatázisban tároljuk az Entity Framework Core segítségével

A programot MVC architektúrában valósítjuk meg, ennek megfelelően View, Model, Controller névttereket valósítunk meg az alkalmazáson belül.

A program környezetét az alkalmazás osztály Program.cs végzi, amely inicializálja az adat_bázist, és lehetővé teszi a különböző komponensek befecskendezését az IOC tárolóba és a pipeline-t kezeli.

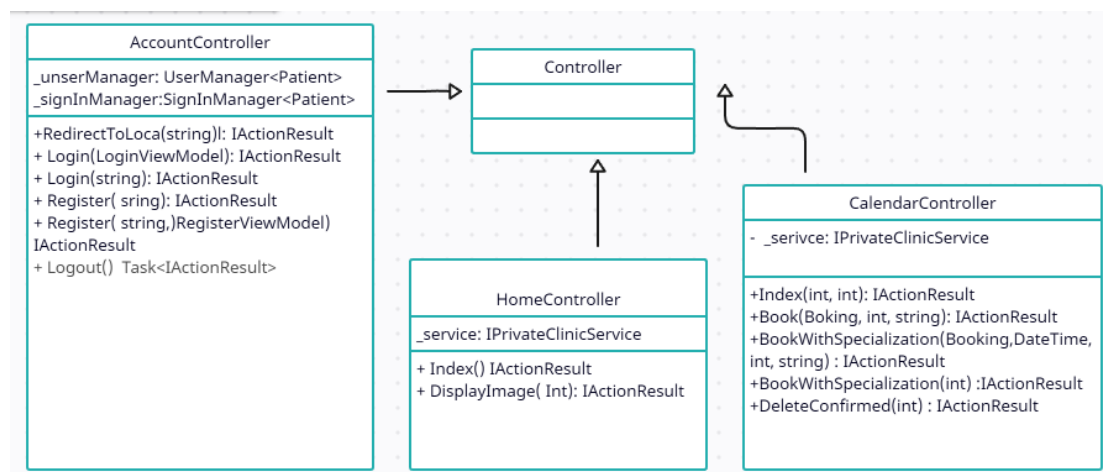
A kontrollereknek és a megfelelő akcióiknak nézetei vannak, egy-egy controllernek egy teljes mappa és az egyes akcióik alkotnak egy-egy nézetet. Három fő Controller osztályunk van. A Home, az Account illetve a Calendar Controller.

4.2 Az alkalmazás komponensdiagrammja:



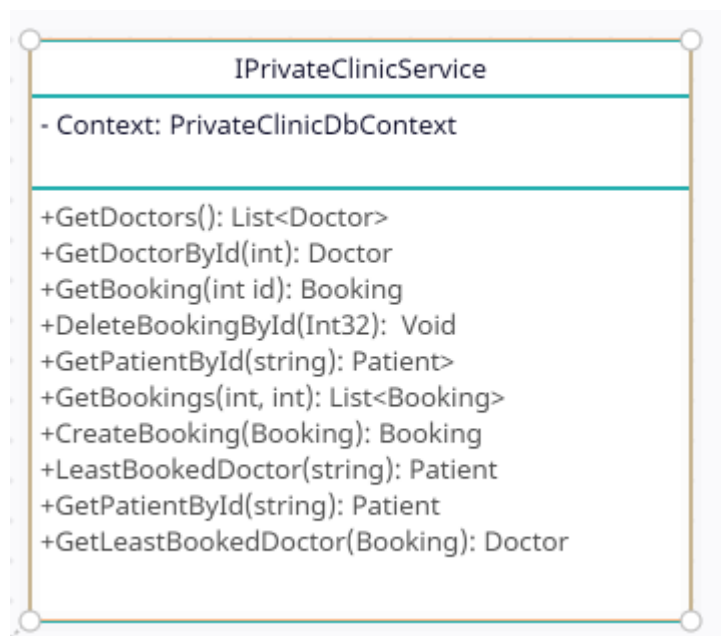
4.3 Vezérlők osztálydiagrammja:

Mind a három vezérlő egy közös őszvezérlőből származik



Az üzleti logikát a modellen belül a service osztály valósítja meg.

4.4 A Service osztálydiagrammja:



Az interface értelemszerűen ugyanezekkel az a függvényekkel és adattal rendelkezik.

4.5 Egyed-Kapcsolat Diagram

Az adatbázis feladata a modellobjektumoknak, regisztrált felhasználók adatainak, rendelőinek adatainak tárolása.

