

DIPLOMAMUNKA

KOVÁCS BALÁZS

DEBRECEN

2010

Debreceni Egyetem
Informatikai Kar

Háziorvosi alkalmazás fejlesztése

Témavezető:

Dr. Juhász István
egyetemi adjunktus

Készítette:

Kovács Balázs
programtervező informatikus (Msc.)

Debrecen
2010

Tartalomjegyzék

1. Bevezetés.....	4
2. Egészségügyi informatika	5
2.1. Egészségügyi szabványok.....	6
2.2. Alapellátás.....	8
2.3. Kódszisztemek.....	13
3. Rendszerfejlesztés	15
3.1. Folyamatmodell	16
3.2. Követelmények	17
3.3. Rendszermodellek	20
3.4. Tervezés	22
3.5. Megvalósítás	23
4. Összefoglalás.....	27
5. Irodalomjegyzék.....	28
6. Függelék.....	29

1 Bevezetés

Az informatika iránt tanúsított érdeklődésem nagyon régre nyúlik vissza. Amióta megkaptam első személyi számítógépet, ami egy 386-os 33 MHz órajellel rendelkező erőmű volt, évről-vére tudatosult bennem az informatika iránti fogékonyság.

Középiskola után felsőoktatásbeli tanulmányaimat a Debreceni Egyetem Informatikai Karának programtervező informatikus bsc. szakán kezdtem el. Az út koránt sem indult nehézségmentesen, de ezen akadályok, próbatételek leküzdése nem csupán fáradtságos munkát, de erőt, motivációt is jelentett. Az évek alatt számos új ismerettel gazdagodtam: az informatika, ma már nyugodtan kimondhatjuk, tudományának legfontosabb alappilléreit ismerhettem meg, a legalapvetőbb eszközöktől kezdve a már mai világban alkalmazott technológiákig. Úgy érzem, hogy a képzés elérte célját, vagyis olyan szilárd alapokat adott számomra, amellyel el lehetett indulni az úton. Ezért tartottam fontosnak, hogy folytassam tanulmányaimat mesterszakon. A választás számomra kétségtelenül ismét a Debreceni Egyetemre esett. A mesterképzésben fél év után dönteni kellett szakosodásunkról, és az egészségügyi informatikus szervező szakirányt választottam.

Az informatika fejlődésének, és a mindennapi életben betöltött elengedhetetlen szerepének köszönhetően azt látjuk, hogy nem csak önálló, hanem társtudomány is lett. Manapság informatikai nélkül nem tudjuk elképzelni világunkat: akár a televízió, vagy mobiltelefonunk bekapcsolásáról, akár a város villamosainak a megállóhelyről informáló berendezéseiről, akár az orvostudományban alkalmazott képalkotó eszközökről (MRI, CT, PET) is legyen szó, és még sorolhatnám végezetlenül, mindezek működését szoftverek végzik, vagyis az adott architektúra nyelvére lefordított program. Program, mely az informatika által elért eredményeket, kifejlődött eszközöket, technológiákat felhasználva igyekszik legjobban kiszolgálni feladatát.

Célom egy háziiorvosi alkalmazás fejlesztésével kapcsolatos folyamatok ismertetése: a rendszerfejlesztés egyes fázisainak egyrészt általános ismertetése, másrészt a háziiorvosi alkalmazáson, mint szakterület specifikus problémán keresztüli bemutatása. Szeretném ismertetni a megvalósítás egyik lehetséges útját, mely az üzleti világban alkalmazott technológiák közül az egyik leggyorsabban fejlődő és legszélesebb körben elterjedt JAVA EE-t takarja.

2 Egészségügyi informatika

Az **egészségügyi informatika** fogalmának számos megfogalmazása létezik:

- interdiszciplináris tudomány, amely több egymást kiegészítő tudományterület, például orvostudományok, informatikai műszaki tudományok, gazdaságtudományi, matematikai ismeretek tudásanyagából építkezik, amelyek folyamatos kölcsönhatásban vannak egymással. Az egészségügyi informatika az egészségügyi adatok megszerzésével, rendszerezésével, feldolgozásával, tárolásával és megjelenítésével foglalkozik: gondoljunk itt a kórházi informatikai rendszerekre, a betegnyilvántartó programokra, képdiagnosztikai eszközökre és még sorolhatnánk
- a számítógépek, a kommunikáció, az informatika és az információs rendszerek alkalmazása az egészségügy minden területén - a betegellátásban, az egészségügyi képzésében, valamint az orvosi kutatásokban (MF Collen, MEDINFO '80, Tokyo)
- a tudásnak és a technikának egy olyan fejlődő eleme, mely az információk szervezésével foglalkozik az orvosi kutatások, a képzés és a betegellátás támogatása érdekében. Magában egyesíti az orvostudomány, valamint az információ-elmélet és számítástudomány számtalan technikai és tudományos elemét és olyan módszertant ad, mely lehetővé teszi az orvosi tudás jobb alkalmazását és így hozzájárul a jobb, eredményesebb betegellátáshoz (Amerikai Orvosszövetség - AAMC)

Az egészségügyi informatikától, annak egy részét alkotó, manapság egyre elváló terület az **orvosi informatika**.



Korábban úgy tekintettek erre, mint az egészségügy komputerizálására. Mivel manapság a számítógépek egyre jobban mindennapi életünk részévé válnak, olyan tendencia mutatkozik, hogy nem helyeznek olyan hangsúlyt a számítógépre, technológiára, hanem inkább az

információ jelentése, feldolgozása az, ami központi szerepet tölt be az egészségügyi szakemberek napi munkájában, kommunikációjában, az ismeretek megosztásában, a döntéshozásban, valamint egészségügyi szervezetek és szolgáltatások működési szükségleteiben. Legfontosabb alkalmazási területei:

- jel (elektromos, akusztikai stb.) ill. képfeldolgozás
- adattárolás, keresés
- adatátvitel, kommunikáció
- adminisztráció, pénzügy, egészségügyi gazdaságtan
- biostatisztika
- teljesítménymérés, minőségbiztosítás

2.1 Egészségügyi szabványok

Az egészségügyi informatika **nagyfokú szabványosságot követel**, számtalan részrendszer kommunikációját feltételezi

- általános informatikai szabványok
- egészségügy specifikus szabványok

Az európai egészségügyi informatikai szabványosítás fóruma a **CEN** (Comité Européen Normalisation) TC251 bizottsága (Technical Committee 251, Health Care Informatics). Az európai egészségügyi informatikai szabványosítás az alábbi fő területeken folyik:

- architektúra, adatmodellek
- terminológia, kódrendszerek, fogalmi rendszerek
- üzenetek, kommunikáció
- adatvédelem, adatbiztonság

Általánosságban elmondható, hogy az itt kidolgozott szabványok keretjellegek. Ezért gyakorlati szakember túlságosan általánosnak értelmezheti őket. A szabványok típusairól beszélhetünk:

- meta-szabványok: a szabványfejlesztőknek készültek. Arról szólnak, hogyan kell egy-egy területen „szabványos” szabványt készíteni
- keretszabványok: közelebb van a gyakorlati problémákhoz, de még nem ad rájuk konkrét megoldást, hanem a standard megoldás módszerét rögzíti. Például az ENV1828 szabvány, mely műtéti kódrendszerek standard struktúráját írja le
- megoldás szabványok: a CEN szabványok között vannak olyanok is, amelyek egy-egy probléma konkrét megoldására vonatkoznak

Nézzünk egy konkrét európai szabványt: **HISA** (Healthcare Information System Architecture). Ez egy másik, általános jellegű informatikai világszabványra (ISO7498) épít. Ez az ISO szabvány az információrendszereket általánosságban szemléli, és kimondja, hogy bármely ilyen rendszerben három „réteg” különíthető el egymástól:



A szabvány szakmai tartalma:

- alap gondolata: sem a bit utak, sem pedig az alkalmazások rétege nem szabványosítható. A HISA az adatstruktúrák kereteinek (és még minimális adatkezelési funkciók) megadásával éri el az egészségügyi információrendszer architektúrájának szabványosítását
- A HISA főbb komponensei:

- minden egészségügy intézményben emberekkel foglalkoznak, így a HISA alapvető alkotója a Subject-HCS (Páciensadatok). Az S-HCS lehetővé teszi a betegek azonosítását, melyre minden intézményben szükség van
- a páciensek (betegek vagy egészségesek) vizsgálata és kezelése során létrejött egészségügyi adatokat (például laboratóriumi vizsgálati eredmények vagy röntgen kép) HealthCare data-HCS (HC-HCS) foglalja össze
- az Activity-HCS foglalja össze mindazokat a tevékenységeket, melyekre előbb a beteg vizsgálata, majd gyógyítása során kerül sor
- a Resources-HCS mindazokat az erőforrásokat jelenti, amelyek a fenti tevékenységek elvégzéséhez szükségesek (például műszaki erőforrás, fogyó- vagy állóeszköz)
- az eddig említett négy szolgáltatás együttesen teszi lehetővé az egészségügyi intézmény teljes tevékenységének nyomon követését

2.2 Alapellátás

Ahhoz, hogy egy háziiorvosi alkalmazás fejlesztéséről beszélni lehessen, elengedhetetlen a szakterület megismerése, a mögötte lévő folyamatok, fogalmak megértése. A következőkben a diplomamunkám címét képező ellátás szakma specifikus tartalmát, legfontosabb fogalmait próbálom tisztázni, melyek elengedhetetlenek lesznek az alkalmazás fejlesztési folyamatának szakaszaiban.

Az egészségügyi ellátás célja: a jó egészségi állapot biztosítása (prevenció és gyógyítás) Az egészségügyi ellátás szintjei:



Diplomamunkám témája egy háziorvosi szoftver fejlesztési folyamatának bemutatása. Így a háromszög legalsó területét érinteném a következőkben.

Az **alapellátás** (1997. évi CLIV. törvény az egészségügyről 88. §):

(1) a beteg lakóhelyén, illetve annak közelében biztosítani kell, hogy választása alapján igénybe vehető, hosszú távú, személyes kapcsolaton alapuló, nemétől, korától és betegsége természetétől függetlenül folyamatosan egészségügyi ellátásban részesüljön

(2) az (1) bekezdésben foglalt alapellátás célja

a. az ellátott lakosságra vonatkozó megelőző tevékenység

b. az egyén

i. egészségi állapotának figyelemmel kísérése, valamint egészségügyi felvilágosítása és nevelése,

ii. külön jogszabályban meghatározott kompetencia keretében történő gyógykezelése, gondozása és rehabilitációja az adott diagnosztikus és terápiás háttér mellett,

iii. szakorvoshoz történő irányítása a betegség megállapítása, kezelési terv készítése vagy terápiás ellátás céljából,

iv. gyógykezelése, házi ápolása és rehabilitációja a kezelőorvos által javasolt terápiás terv alapján

c. szükség esetén 2.b.ii és a 2.b.iv alpontban foglaltaknak a beteg otthonában történő ellátása, illetőleg a beteg otthonában végzendő szakorvosi konzílium kérése

Az alapellátás szakmai területei:

- **háziorvosi ellátás**, házi gyermekorvosi ellátás
- fogorvosi alapellátás
- az alapellátáshoz kapcsolódó ügyeleti ellátás
- védőnői ellátás
- iskola-egészségügyi ellátás
- család és nővédelmi gondozás

Háziorvosi ellátás

Annak érdekében, hogy a beteg ember számára biztosítható legyen, hogy lakóhelyén vagy annak közelében, választása alapján, nemétől, korától, betegsége természetétől függetlenül folyamatos egészségügyi ellátásban részesüljön, az alapellátás keretében szervezett egészségügyi szolgáltatás, a háziiorvosi ellátás működik.

A háziiorvosi szolgálat működését a regionális egészségbiztosítási pénztárakkal kötött szerződés alapján az Egészségbiztosítási Alapból finanszírozzák, a háziiorvoshoz bejelentkezett ellátásra jogosultak száma alapján.

Háziiorvosi tevékenységek:

- Diagnosztikus tevékenységek eljárások: azon diagnosztikus tevékenységek, eljárások köre, melyek elvégzése, illetve elrendelése és az eredmények megfelelő értelmezése a háziiorvos hatásköre és kötelessége a szakmai ajánlások szerinti indikációs körben:
 - anamnézis felvétele
 - fizikális vizsgálatok
 - diagnosztikai vizsgálatok elrendelése
- Önálló betegellátási tevékenység: betegségek és állapotok háziiorvos által irányított teljes körű ellátása (prevenció, kivizsgálás, kezelés és követés), melyet háziiorvosi szolgálat önállóan, vagy az ellátási helyzetnek megfelelően hozott saját döntése alapján más ellátók konzultánsi közreműködésével végez.
- Betegellátás, szakellátás irányítása: felismerendő betegségek és állapotok szakellátás irányításával, annak elsődleges felelősségével történő diagnosztikus, terápiás, gondozási jellegű ellátása, melynek folyamatában a háziiorvos esetenként meghatározott feladatkörben vesz részt a szakellátás felkérése alapján. Egyebekben a háziiorvos felelősségei körébe a beteg figyelemmel kísérése tartozik.
- Tájékoztatás: mindazon egyéb, nélkülözhetetlen szakmai ismeret, mely segíti a beteg optimális ellátását, tájékoztatását.

Háziiorvosi tevékenység feladata:

- Prevenációs (megelőzési) feladatok:
 - immunizáció (védőoltások)
 - egészségnevelés

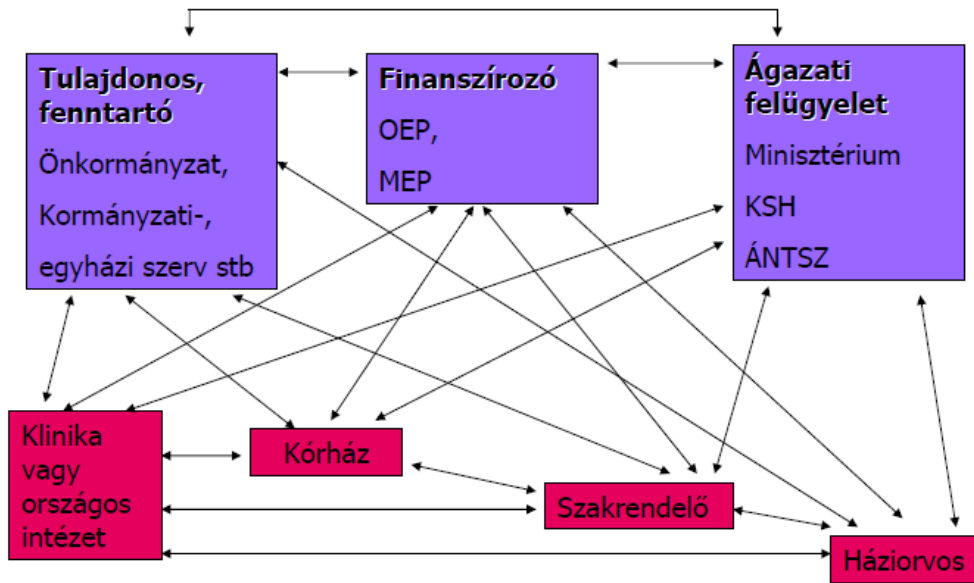
- szűrések
- egészségfejlesztés
- Kurációs feladatok: ez alatt diagnosztikus, terápiás, gondozási, rehabilitációs és ápolási tevékenységeket értünk.
 - saját hatáskörében elvégzett terápia, gondozás, ápolás
 - konziliárus által meghatározott terápia, gondozás
- Egészségügyi állapot menedzselése: a betegek egészségének nyomon követése
 - beutalás, további kezelések
 - prevenció programok szervezése

A házi orvos, házi gyermekorvos munkáját **ápoló(nő) vagy asszisztens** segíti. Az ápolónő feladata a házi orvos megbízása alapján:

- az orvos feladataihoz kapcsolt ápolási munka
- megelőzésben, szűrésben, gondozásban való részvétel
- a vizsgálathoz, gyógykezeléshez szükséges eszközök, anyagok előkészítése, fertőtlenítés, sterilizálás stb.

Mindezek ismertetését azért tartom fontosnak, mert az alkalmazás fejlesztésének folyamata során szükséges lesz a szakterületi folyamatok definiálására, meghatározására, melyek egyben az alkalmazás funkcionalitásának magját fogják képezni.

Másik fontos kérdés az orvosi tevékenység során keletkezett adatok kezelése, karbantartása és menedzselése, melyek lehetővé teszik egyrészt a betegre vonatkozó információk visszakereshetőségét, nyomon követhetőségét, másrészt aggregált adatok származtatását, illetve információáramlást az egyes intézmények, szervezetek között. A cél, hogy az egészségügy működésének minden pontján el kell számolni arról a betegről, aki az egészségüggyel kapcsolatba került és beszámolni arról, amit a betegség megelőzése, gyógyítása, rehabilitációja kapcsán anyagként, eszközként felhasználtunk. A következő ábra az információáramlást szemlélteti az egyes egészségügyi intézmények között:



Látható, hogy ebben a zárt rendszerben kétirányú információcsere valósul meg. Minden intézmény közvetlenül, vagy közvetett úton kapcsolatban áll az összes többi intézménnyel. Ennek következtében nagyszámú dokumentációs feladatokat kell ellátni az egyes szervezeteknek. Háziorvosi szempontból ez nagy feladatot jelent, hiszen a prevenció, betegápolás és egyéb tevékenységek mellett ez is feladatkörükhöz tartozik. Úgy gondolom, hogy a jelenlegi informatikai eredményeket, eszközöket kihasználva elengedhetetlen olyan szoftverek fejlesztése, amelyek egyrészt támogatást, segítséget nyújtanak ezekben a dokumentációs tevékenységekben, másrészt a modern kor követelményeinek megfelelő egyéb funkciókat is megvalósítanak, melyek elsősorban az orvos szakmai folyamatokban töltenek be szerepet (döntéstámogatás).

Az alapellátás dokumentációs feladatai határozzák meg, milyen típusú adatokat kell kezelni. Vagyis kicsit előre tekintünk a fejlesztési folyamatba, ahol majd részletezni fogom az egyes szakterületi fogalmakat.

- Szakmai dokumentáció

- | | |
|-----------------------------------|----------------------------------|
| • Törzskarton | • Betegkarton |
| • Keresőképtelenség dokumentálása | • Receptek |
| • Gondozási nyilvántartás | • Igazolások, javaslatok |
| • Beutalás | • Orvoshívások rögzítése |
| • Betegszállítási utalványok | • Bejelentendő betegségek nyilv. |
| • Fertőző betegségek nyilv. | • Utazási utalványok |

- Alkalmassági véleményezések
- Hatósági feladatok dokumentálása
- Társadalombiztosítással kapcsolatos dokumentáció
 - Páciens regiszter
 - Orvosi napló, táppénz utalvány
 - Ambuláns napló
 - Jelentések
 - be is kijelentkezett biztosítottak adatai
 - táppénz ellátási adatok
- Gazdasági ügyekhez tartozó dokumentáció
 - Gyógyszerigénylés
 - Kötszerigénylés
 - Vegyszerigénylés
 - Egyéb

2.3 Kódszisztemek

Az orvostudománynak sajátos nyelvi tulajdonságai vannak:

- sajátos szókészlet
- érzelmi, indulati tartalom kifejezésének hiánya
- explicit: nem használ szimbolikus, metaforikus kifejező eszközöket

Egy fogalom teljes leírása: $N|A_1, \dots, A_n$ típusú leírás, ahol N jelöli ki magát a fogalmat, A_i -k pedig a fogalmat leíró attribútumokat jelenti. Egy korszerű egészségügyi alkalmazásban elengedhetetlen, hogy az egészségügy specifikus fogalmakat kezelni tudjunk. A számítógépes fogalomreprezentáció, ismeretreprezentáció ezt a feladatot foglalja magába:

- kezelni kívánt fogalomkör kijelölése: az orvostudomány teljes fogalomkészlete behatárolhatatlan, jól körülhatárolt részt kezelhetünk. Ennek megoldása, hogy a fogalmainkat szakszerűen, célirányosan csoportosítjuk attribútumok segítségével
- fogalmak rendezése, csoportosítása: alapja fogalmak attribútum azonossága. A rendezési elv megválasztásánál a csoportosítás alapját képező attribútumokat jelöljük ki

- jelsorozatok hozzárendelése a fogalmakhoz a rendezés sorrendjében: a célkifejezés helyettesítése valamilyen nem nyelvi szimbólummal (kódolás)

Az egészségügyi informatika fontos területe a klasszifikáció (osztályokba sorolás). Ahhoz, hogy az orvosi, egészségügyi fogalomrendszerek számítógép által is feldolgozhatóvá váljanak, a fogalmakat egységesíteni, kódolni kell. Különböző kódrendszerek születtek meg, melyek a számítógépes adatfeldolgozást segítik elő:

- többdimenziós kódrendszerek: orvosi fogalmak több szempontú kódolását valósítják meg
 - SNOMED: a kódrendszer az orvostudomány teljes fogalomkészletét igyekszik lefedni. Bonyolultabb, komplexebb kifejezéseket is lehet reprezentálni benne.
- egydimenziós kódrendszerek: a kódrendszer kialakításánál egyetlen egy szempont szerint rendezik az orvosi fogalmakat
 - Reed: európai vonatkozási kódrendszer, hazánkban nem alkalmazzák. Rendszere felöleli a diagnózisokat és a beavatkozásokat is.
 - OENO (Orvosi Eljárások Nemzetközi Osztályozása): orvosi tevékenységek, beavatkozások kódolását valósítja meg.
 - BNO (Betegségek Nemzetközi Osztályozása): csak a diagnózisok fogalmkörét foglalja magába. Hazánkban jelenleg a 10-es verzióját alkalmazzák.

Ezek közül a BNO kódrendszerrel szeretnék pár szót ejteni, hiszen minden modern egészségügyi alkalmazásban használják, akár az alapellátásról, járóbeteg szakellátásról, vagy fekvőbeteg ellátásról legyen szó:

- 3 kötetes: az első kötet a megfelelő, korrekt kódokat tartalmazza, a második kötet útmutató, gyakorlati felhasználást segíti, a harmadik kötet tárgymutató
- elvben minden betegség kódolását lehetővé teszi. Tehát a betegségek számítógépes reprezentációját szolgálja
- adatok statisztikus feldolgozhatóságát biztosítja
- a betegségek, egészségügyi problémák diagnosztikai alfanumerikus kóddá alakítható, így könnyebb a tárolás, visszakeresés, elemzés

- 5 karakteres osztályozást használ: háromkarakteres kötelező szintű kódolás
- 21 főcsoport van, néhány ezek közül: fertőző és parazitás betegségek, daganatok, az idegrendszer betegségei, a szem és függelékeinek betegségei stb.
- példa: *Tonsillitis chronica* (idült garatmandula gyulladás)
 - BNO 10-es I. kötet (kódnövekvő listák - kódok sorrendjében)
 - J35-A garat és orrmandulák idült betegségei
 - J35.0-Idült garatmandula-gyulladás
 - kivéve: tonsillitis
 - BNO 10-es III. kötet (betűrendes tárgymutató – abc sorrend)
 - latinul: tonsillitis chronica J35.0
 - magyarul: Idült garatmandulla gyulladás J35.0

3 Rendszerfejlesztés

Mindennapi életünkben különböző szoftver rendszerekkel találkozhatunk, számítógép alapú világban élünk. Tulajdonképpen minden egyes elektronikus berendezés tartalmaz szoftvereket valamilyen formában. A szoftverek nagy komplexitást hordoznak magukban, fejlesztésük mérnöki megközelítést követel meg. Másrészt ezeknek a szoftvereknek a specifikációja, tervezése, menedzselése és evolúciója egymástól elválaszthatatlan fogalmak és a szoftverfejlesztés mint tudományágat alkotják. Így a szoftver egy speciális termék, mely nem anyagi természetű, viszont ugyanúgy van költsége, mint bármely más hétköznapi terméknek. Így a cél nem csak működő, szükségleteinket kielégítő alkalmazás előállítás, hanem a költséghatékonyság szem előtt tartása.

A szoftverfolyamat tevékenységek és kapcsolódó eredmények olyan sora, melyek egy szoftvertermék előállításához vezetnek. Számos szoftverfolyamat létezik, de vannak olyan alapvető tevékenységek, amelyek közösek:

- szoftverspecifikáció: a szoftver funkcióit, megszorításait definiáljuk
- szoftvertervezés és implementáció: a specifikációnak megfelelő szoftvert kell előállítani
- szoftvervalidáció: annak ellenőrzése, hogy azt fejlesztettük ki, amit az ügyfél kíván

- szoftverevolúció: szoftverrugalmasság biztosítása, mely lehetővé teszi az ügyfél igényeinek megfelelő változtatások véghezvitelét

3.1 Folyamatmodell

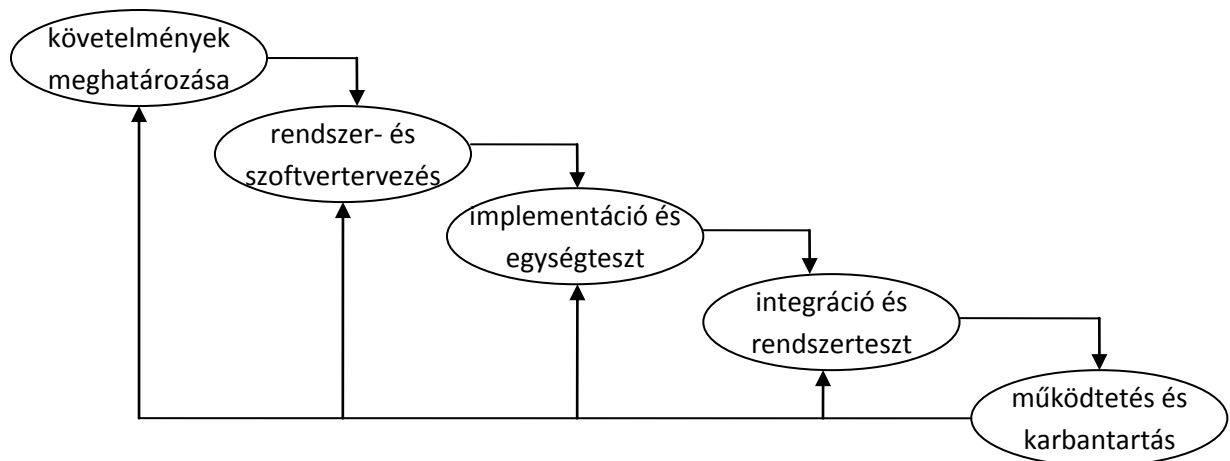
A szoftverfolyamat számos modelljét alkalmazzák. Az előző mondatban szereplő modell is arra utal, hogy ezek absztrakt reprezentációk. Egy általános szemléletmódot adnak, amelyeket különböző megközelítésből szemlélik az alkalmazás fejlesztési folyamatát:

- vízesés modell
- evolúciós modell
- formális rendszerfejlesztés
- újrafelhasználás alapú fejlesztés

A háziorvosi alkalmazás esetén én a vízesés modellt fogom alkalmazni. Egyrészt ez tekinthető a legklasszikusabb, legkönnyebben menedzselhető folyamatmodellnek. Másrészt kivételes a helyzet, mivel a projekt megvalósítását egyedül igyekszem véghezvinni, és ezt látom a legérthetőbb és ebben az esetben a legésszerűbb megközelítésnek. A vízesésmodell legfontosabb jellemzői:

- szekvenciális: minden fázisnak le kell zárulnia, mielőtt a következő elkezdődik (gyakorlatban lehetnek átfedések)
- nem szimpla lineáris modell: amint látjuk, fejlesztési tevékenységek eredményének visszacsatolása miatt iterációk sorozatáról beszélhetünk
- a fázisok lépcsősen kapcsolódnak egymáshoz
- hátránya, hogy a folyamat korai szakaszában szükséges a követelmények meghatározottsága

A vízesés modell lépési folyamata:



Diplomamunkám keretein belül a fent látható ábra első két fázisát szeretném részletesen bemutatni, és a példaalkalmazáson keresztül szemléltetni. Mivel szoftverfejlesztésnél nem beszélhetünk fizikai korlátokról, mint költségről, de az idő annál inkább meghatározó tényező, ezért az utolsó három fázis ennek tükrében kerül megvalósításra.

- Követelmények elemzése meghatározása: a rendszer szolgáltatásai, megszorításai és céljai a felhasználóival történő konzultációk alapján alakul ki. Ezek szolgáltatják a rendszerspecifikációt.
- Rendszer- és szoftvertervezés: ebben a fázisban kell kialakítani a rendszer átfogó architektúráját. Tulajdonképpen szoftverrendszer absztrakciók definiálása, köztük lévő kapcsolatok azonosítása és leírása történik.
- Implementáció és egységteszt: a szoftverterv programok, programegységek halmazaként valósul meg. Az egységteszt ellenőrzi, hogy az egyes egység megfelel-e a specifikációnak.
- Integráció és rendszerteszt: különálló programegységek, programok integrálása, és teljes rendszerként történő tesztelése történik meg.
- Működtetés és karbantartás: a rendszer karbantartását, valamint szolgáltatásainak továbbfejlesztését jelenti. Alaphelyzetben a szoftver életciklusának leghosszabb fázisa.

3.2 Követelmények

Az egész rendszerfejlesztés folyamata úgy indul, hogy valaki, legtöbbször a megrendelő, megfogalmaz egy víziót, elképzelést. A rendszer általános, informatikai kifejezésektől mentes, rövid leírása, amely tartalmazza a rendszer működésére és felhasználóira vonatkozó

alapvető kritériumokat: mire használjuk a rendszert, ki fogja használni, hogyan. **(Függelék 1. ábra)**). Ezután egy megvalósíthatósági tanulmányt készítenek el, amely a kifejlesztendő rendszerrel kapcsolatos kérdéseket válaszol meg:

- mennyiben fogja szolgálni az alkalmazás a megrendelő intézmény üzletpolitikáját?
- mennyire illeszkedik be a napi munkába?
- megvalósítható-e az adott idő, pénzügyi keretek között a felhasználható technológiával?

Ha a jelentés azt a választ adja, hogy megvalósítható a rendszer, akkor megkezdődik a követelmények, megszorítások meghatározása, amelyek körülhatárolják a szoftverrendszer működését. A követelmények több szempont alapján csoportosíthatóak:

1. Felhasználói-, rendszerkövetelmények

- a felhasználói követelmények leírják, hogy milyen szolgáltatásokat, milyen körülmények között várunk el a rendszertől. Vagyis azt definiálják, hogy a felhasználók hogyan használhatják az alkalmazást, milyen funkciókat érhetnek el
- a rendszerkövetelmények a rendszer egészére vonatkoznak. Megfogalmazzák, hogy a kifejlesztendő szoftver milyen gépen, milyen környezetben fogják használni (rendszer felépítése)

2. Funkcionális, nem funkcionális, valamint szakterületi követelmények

- funkcionális követelmények határozzák meg a rendszer által nyújtandó szolgáltatásokat
- a nem funkcionális követelmények a rendszer által kínált funkciókra tett megszorítások (metrikák, megbízhatóság, rendelkezésre állás). Magukba foglalhatnak időbeli korlátot, szabványokat. Azaz a rendszer egészére vonatkoznak, tehát eredendő követelmények, nem köthetők egy-egy modulhoz
- a szakterületi követelmények a szakterület területéről származnak és e szakterület jellegzetességeit tükrözik, szakterület specifikusak

A követelmények megértéséhez szükséges tisztáznunk fejlesztés során használt szakterületi fogalmakat, folyamatokat. Erre szolgál:

- a fogalomszótár, melynek célja, az alkalmazás szakterület specifikus fogalmainak általános, köznyelvi megfogalmazása. A fogalomszótárban nem használunk informatikai kifejezést, ez a dokumentum semmit sem mond a későbbi megvalósításról, csak a fejlesztő számára teszi érthetővé az adott szakterületen általánosan használt kifejezéseket **(Függelék 2. ábra)**
- szakterületi folyamatok, kapcsolatok leírása, melynek célja az alkalmazás szakterületi folyamatainak általános, köznyelvi megfogalmazása. A fejlesztő számára magyarázza el, hogy az adott területen hogyan zajlik le az adott folyamat **(Függelék 4. ábra)**

A követelmények feltárásának eredményét dokumentálni kell. Különböző reprezentációs módszerek vannak:

- természetes emberi nyelven leírom. Előnye, hogy a fejlesztő és felhasználó is megérti, de könnyű félreérteni (nem szabványos)
- követelményi űrlapok (strukturált természetes leírás)
- formalizált nyelven: kötött szintaktikájú. Hátránya, hogy a felhasználó nem érti meg. Például PDL – Program Description Language
- diagramok használata (modellező nyelv segítségével (UML))
 - használati eset diagram: a rendszer vagy annak egy részének viselkedését modellezi, vagyis a felhasználóknak az alkalmazással szemben támasztott elvárásait, követelményeit reprezentálja **(Függelék 5. ábra)**
 - tevékenység diagram: felhasználási körök nagyon széles, a legkülönbözőbb folyamatok szemléltetésére használjuk, például segítségükkel az alkalmazás dinamikus viselkedését modellezzük. A tevékenységek magukba hordozzák a rendszer lényeges jellemzőit, funkcióit **(Függelék 6. ábra)**

Gyakorlatban előszeretettel használják az elsődleges felhasználói kézikönyvet. A fejlesztés elején már egy közérthető, a rendszer működését körülhatároló dokumentumot állítanak elő: egyrészt a kommunikációt szolgálja a fejlesztő és megrendelő között, másrészt a fejlesztőknek

segít a probléma megértésében, illetve a felhasználónak is érthető formában adja vissza az alkalmazás által nyújtott szolgáltatásokat.

- általános leírás, amely egyben tekinthető egy funkcionális specifikációnak
- szakterületi fogalmak
- szakterületi folyamatok
- rendszerkövetelmények: néhány példát felsoroltam a MedHome rendszerre vonatkozóan **(Függelék 7. ábra)**

3.3 Rendszermodellek

A felhasználói követelményeket általában természetes nyelven fogalmazzák meg, sokszor a megrendelő számára felfogható diagramokkal egészítik ki, hogy egy nem informatikai szakember is megértse azokat. A részletesebb rendszerkövetelmények már szakmai fogalmakkal is kifejezhetőek. Leggyakrabban használt technika, amikor a rendszerspecifikációt rendszermodellek segítségével definiálják. Ezek olyan grafikai reprezentációk, melyek a tervezés felé képeznek átmenetet: leírják a megoldandó problémát és a kifejlesztendő rendszert. Segítségükkel a rendszert különböző szemszögből reprezentálhatjuk:

- külső szemszögből a rendszer kontextusát, környezetét modellezzük: korán alkalmazható modellek, melyek a rendszer lehatárolását szolgálják, viszont nem mond semmit a viselkedésről. Megmondják, hogy a kifejlesztendő alkalmazás hova integrálódik, milyen rendszerekkel áll kapcsolatban. A MedHome esetén az UML kontextus diagram finomításaként szolgáló montázsdiagram kitűnően használható ennek a feladatnak a modellezésére **(Függelék 8. ábra)**
- belső szemszögből a rendszer dinamikus viselkedését modellezzük. Az alkalmazás átfogó viselkedésének leírására használják. Két típust különböztetünk meg:
 - adatfolyam modell: az adatok feldolgozását mutatja be, az adatok rendszeren belüli vándorlását, feldolgozását reprezentálják. Jelentősége, hogy a rendszeren keresztül egyes folyamat-mozgásokhoz társított adatok dokumentálása és nyomon követése segíti az elemzőket megérteni, mi is történik. Ennek

bemutatására a MedHome betegtalálkozás rögzítésének folyamatát modellezem adatfolyam diagram segítségével (**Függelék 9. ábra**)

- állapotátmenet modell: strukturált viselkedést írja le. Az OO módszertanok kedvenc modellje. Központjában az események állnak, azokat a rendszerállapotokat és eseményeket mutatja be, melyek egy állapotból egy másikba történő átmenetet okoznak. Az adatok áramlását a rendszeren belül nem mutatja be. Olyan rendszerek modellezésére szolgál, amelyek úgynevezett valós idejűek, azaz a rendszert a rendszer környezetéből érkező ingerek vezérlik. A felhasználó regisztrálási folyamatában a jelszó megadásának lépését emeltem ki. Az "OK" gomb működésének állapotátmenet modelljét konstruáltam meg: adatok validálása (**Függelék 10. ábra**)
- adatmodellek: általános, hogy a nagy rendszerek mögött adatbázisok állnak. Nagytömegű adatokat kezelő alkalmazások modellezési eszközei, vagyis absztrakciós eszközök, melyek segítségével a rendszer mögött lévő adatbázist reprezentáljuk. Számos adatmodell született, ezek közül a legfontosabbak: relációs, szemantikus, OO, objektum relációs adatmodellek, melyekre konkrét adatbázis kezelő rendszerek épülnek (MySQL, Oracle, Java DB), illetve ER és az EER (kiterjesztett ER), amely kimondottan absztrakciós, grafikus modellező eszközök. A MedHome háziorvosi alkalmazás relációs adatmodellt megvalósító adatbázis kezelő rendszerrel kommunikál majd. A grafikus megközelítése miatt az ER modellt találtam a leghasznosabbnak az adatbázis tervezéséhez.
 - ER: objektumalapú, magas absztrakciós szintű sématervező eszköz. Az adatok absztrakt, koncepcionális leírására használják. Legfontosabb ER fogalmak:
 - egyedtípus
 - egyed
 - attribútum
 - kulcs
 - séma
 - kapcsolatok
 - ER sémákat le kell képezni konkrét adatbázis kezelő rendszerek mögött álló adatmodellt tükröző sémákra. Ez analóg módon történik, és a legtöbb modellező eszköz automatikus eszközöket nyújt relációs, illetve objektumrelációs leképezésekhez. A Visual Paradigm for UML modellező szoftvert használtam, amely kiválóan alkalmas erre a feladatra, és a modellező

diagramok széles skáláját adja kezünkbe. Igyekeztem az HISA szabvány adatstruktúráinak, komponenseinek megfelelően kialakítani az adatbázist **(Függelék 11. ábra)**

3.4 Tervezés

Az életciklus legfontosabb részét képezi az architektúrális tervezés, mely közvetlen hatással van az alkalmazás eredendő követelményeire: teljesítmény, biztonság, rendelkezésre állás. Ennek során meghatározzuk a kifejlesztendő rendszer főbb alrendszereit, komponenseit. Az alrendszerek olyan önálló elemek, melyek működése nem függ más alrendszertől. A komponensek nem önálló egységet képeznek, részben szolgáltatók vagy szolgáltatást vesznek igénybe. A rendszer architektúrájának leírására, az előzőkhez hasonlóan, UML diagramokat használtam. A MedHome háziiorvosi alkalmazás architektúrájának tervezését több lépésben hajtottam végre.

- első lépésben egy durva szemcsézetségű architektúra leírást adtam meg, ahol az MedHome alkalmazás alrendszereit, egymáshoz való viszonyukat reprezentáltam deployment diagram segítségével. **(Függelék 12. ábra)**
 - az ábrán látható modulok funkcionalitását a fogalomszótárban, és a szakterületi folyamatok és kapcsolatuk részben már definiáltam
- az alrendszerek kibontásával tovább granulálhatjuk architektúránkat. Az alrendszereken belüli komponensek, kapcsolatuk meghatározásával egyre tisztább képet kapunk.
 - mivel az alkalmazás többretegű architektúrát valósít meg (lásd rendszerkövetelmények), ezért minden alrendszer rendelkezik webes megjelenítésért felelős komponenssel, amely egy globális interfészt nyújt a felhasználónak, illetve perzisztenciakezelő, továbbá az adott modul funkcionalitását, üzleti logikáját megtestesítő komponenssel **(Függelék 13. ábra)**
 - komponens diagramokkal granuláltam tovább a Praxis modul felépítését. Analóg módon a többi alrendszer is hasonlóképpen modellezhető. Az ábrán látható üzleti logikát megvalósító komponensek működését szintén definiáltam már a fogalomszótárban, és a szakterületi folyamatokban. Az egyes

komponensek jól elkülöníthető funkciókat látnak el (modularitás) **(Függelék 14. ábra)**

- a rendszer egyes komponenseinek struktúrájának meghatározását osztálydiagramok segítségével reprezentálhatjuk. A praxis modul interfész definíciójával láthatóvá válnak az egyes komponensek műveletei, tevékenységei. **(Függelék 15. ábra)**
- a fenti lépések sorozata a rendszer struktúrájának kialakítását célozza. Szintén fontos kérdésnek tartom az alkalmazás mögött lévő adatbázis tárolási modelljének meghatározását. A MedHome alkalmazás mögött lévő adatok egy központi, megosztott adatbázisban helyezkednek el. Az egyes alrendszerek közösen használják ezt az adatbázist

3.5 Megvalósítás

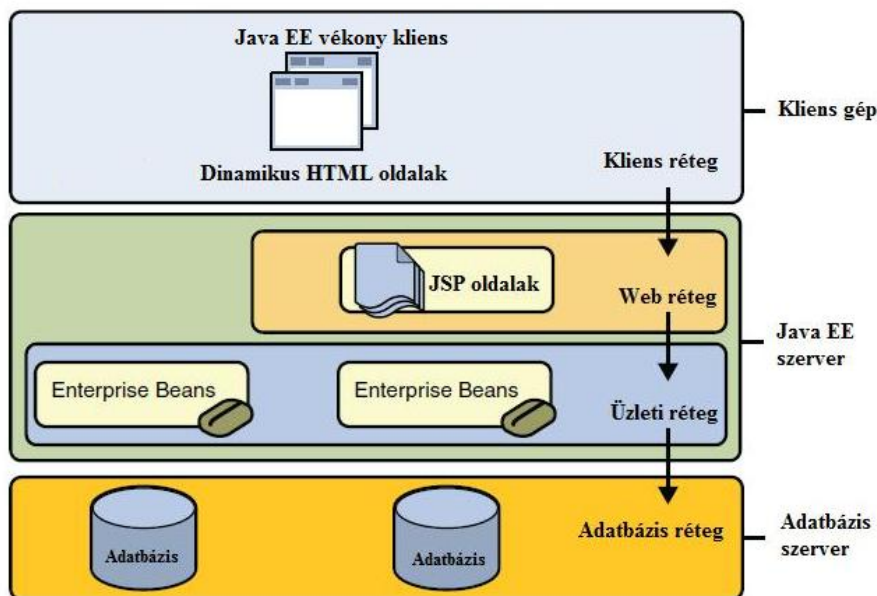
A rendszerfejlesztés folyamatának megvalósítási szakaszában az előző fázisokban előállt tervelemek transzformálása történik futtatható kóddá. Legfontosabb kérdés a felhasználható eszközök összegyűjtése, és az implementálás alapját adó technológia kiválasztása. Ezek nagymértékben befolyásolják a fejlesztési folyamatot.

Eszközök összegyűjtése:

- tervező eszközök: olyan, általában grafikus, modellező eszközök, melyek különböző szemléletű vizuális elemek választékát nyújtják a követelmények, tervelemek reprezentációjához (MagicDraw, Argo UML, UML). A MedHome háziiorvosi alkalmazás fejlesztése során én a Visual Paradigm szoftvert használtam, amely általam ismertek közül a legszélesebb körű szolgáltatásokat adja kezünkbe
- fejlesztői eszközök: olyan integrált fejlesztői környezetet nyújtó szoftverek, melyek az alkalmazásfejlesztés megkönnyítését, részben ezzel kapcsolatos automatizálható lépések támogatását valósítják meg. A fejlesztés során NetBeans IDE 6.7.1-t választottam, mert egyrészt ingyenes termék, másrészt rugalmassága, bővíthetősége, és általa nyújtott széles szolgáltatások segítséget adnak az implementálás folyamatában: hibakeresés (Debug), tesztelés (JUnit), profiling (optimalizáláshoz), Kenai integráltság (együttműködési és szoftverprojekt hosting szolgáltatását: dokumentáció, verziókezelés - SVN)

Technológia kiválasztása:

- a Java EE technológia vállalati méretű, komplex alkalmazások fejlesztését támogatja. Olyan programozási modellt valósít meg, melynek segítségével az alkalmazás fejlesztésének folyamata jelentősen lerövidül anélkül, hogy minőségi, biztonsági, egyéb funkcionális, és nem funkcionális kritérium megsérülne (perzisztencia, többszálúság, tranzakciókezelés, skálázhatóság, magas rendelkezésre állás stb.). A Java EE többrétegű architektúrát használ. Minden réteg egy jól definiált feladatot lát el, csakis az alatta lévő rétegből vesz igénybe szolgáltatásokat, és csak a fölötte lévő rétegnek nyújt szolgáltatásokat.



- egyes rétegekben használt eszközök
 - adatbázis réteg: Java DB adatbázis kezelő rendszert használtam. Persze használhatóak más gyártók rendszerei is (MySQL, Apache Derby, Postgresql)
 - JAVA EE szerver: Glassfish V2.1 alkalmazásszervert alkalmaztam
 - Üzleti Réteg: EJB (Session Bean, Entity Bean)
 - Perzisztencia réteg: JPA-t használtam, de más eszközök is igénybe vehetőek (Hibernate, EclipseLink)
 - Web réteg: JSF

- kliens réteg: az alkalmazás vékony klienst valósít meg, a felhasználónak csak egy böngészőre van szüksége, a nézetet a web réteg fogja megvalósítani. Számos előnye van a vékony kliens alkalmazásának:
 - az alkalmazás a központi szerveren fut, mialatt a felhasználó egység csak annyi számítógépes forrást használ, amennyi szükséges, hogy elérje a szervert
 - nincs szükség helyi támogatásra, használata sem igényel helyi telepítést. Bármely alkalmazást érintő változást szerver oldalon történik, így a felhasználónak nem kell konfigurálni, frissítenie stb.

A dolgozatom megkezdésekor azt tűztem ki célul, hogy minimális funkciójú alkalmazást készítek el. Általában ezt a szakma prototípuskészítésnek nevezi. Az implementálást a legelső és legfontosabb momentumánál kezdtem el. A MedHome alkalmazás használatához elengedhetetlen regisztrációs folyamatot valósítottam meg a fent felvonultatott eszközök segítségével: praxis, háziorvos, ápoló regisztrálása.

Továbbfejlesztés: A diplomamunkám témáját képező háziorvosi alkalmazás fejlesztésének során látható, hogy az architektúra kialakításánál (deployment) a háziorvos tevékenységek logikai csoportosítását vettem figyelembe, így egy olyan rendszert kaptunk, ahol az egyes alrendszerek teljesen függetlenek. Továbbá látható, hogy újabb alrendszerekkel is bővíthetjük, valamint egyes alrendszereken belül újabb komponenseket vehetünk föl. A MedHome a legfontosabb és legszükségesebb funkciókat testesíti meg önmagában, melyek a jelenlegi jogi szabályozásokat elégíti ki. Felmerülhet az igény újabb funkciók bevezetésére, amely az alkalmazás használhatóságának növelését szolgálja:

- kommunikáció más egészségügyi rendszerekkel: olyan kommunikációs alrendszer kialakítása, mely XML alapú üzenetek küldését, feldolgozását valósítja meg. Ennek haszna abban rejlik, hogy ha a háziorvos beutalja betegét járóbeteg, vagy fekvőbeteg ellátásra, az azonnal láthatóvá válik az érintett egészségügyi intézményben, valamint az ott kapott vizsgálati eredmények azonnal megjelennek a háziorvos rendszerében is, ami gyorsabb, hatékonyabb ellátást eredményez
- döntéstámogató alrendszer integrálása: a mesterséges intelligencia tudományterületére épülő olyan alrendszer kifejlesztése, mely segíti az orvosi döntéshozatalt. Például a betegtalálkozások során a beteg szubjektív panaszainak kiértékelését, diagnózis

felállítását, valamint terápiás, esetleg gyógyszerhasználati javaslatokat az orvosi döntést megerősítve, segítve egy egészségügyi szakértői alrendszer végezné

- a MedHome háziiorvosi alkalmazás felhasználója egyrészt lehet maga a házi orvos, és az ő munkáját segítő ápoló is. Ugyanazokkal a jogkörökkel használják a rendszert. Továbbfejlesztési lehetőségként a két szerepkör szétválasztását is fontosnak tartom, ami az ápoló jogosultságainak szűkítését jelentené, például nem adhat ki beutalót
- a páciens élhet azzal a joggal, hogy házi orvost vált. Így újabb funkcióval bővíteném az alkalmazást: a rendszerbe regisztrált praxisok közötti betegek átadás megvalósítása

4 Összefoglalás

Diplomamunkám célja a rendszerfejlesztés folyamatának bemutatása egy szakterület specifikus problémán keresztül. Eléggé nagy ez a terület, és komplex, így egyes részterületei önállóan is kitehethetnének egy egész szakdolgozatot. Inkább egy általános, átfogó kép ismertetését igyekeztem megvalósítani. Lépésről lépésre mutattam be, mik azok a legfontosabb, legalapvetőbb tevékenységek, melyek elengedhetetlenek egy alkalmazás fejlesztési folyamatában. Továbbá azokat a leíró, grafikus, modellező eszközöket is próbáltam felsorakoztatni, melyek az alkalmazással szemben támasztott elvárásaink, követelményeink reprezentálását, dokumentálását szolgálják, és melyek biztos alapot nyújtanak a fejlesztőknek az implementálás szakaszában. Az elkészült tervelemek nem tekinthetők validnak, és a rendszer csak egyes részeleleit ragadtam ki: a rendszer teljes feltérképezéséhez, minél pontosabb leírásához további számos tervelemre és azok tovább granulálására lenne szükség. Mivel az implementációt tekintve csak a regisztrációs folyamatot valósítottam meg, ezért a projekt megvalósítását a hiányzó tervelemek megkonstruálása, illetve azok kódolása jelentené. Ezzel le is zárulna a szoftver életciklusának első fontos állomása, a megvalósítás. Ezután a legfontosabb, és a szoftver életének legnagyobb részét kitevő működtetés, karbantartás szakaszok kezdődhetnek meg.

Remélem, hogy sikerült egy gyakorlati példán keresztül betekintést adnom a szoftver életciklusának folyamatába.

A dolgozat készítése közben rengeteg új ismerettel gazdagodtam. A mai piaci világban csak úgy lehetünk versenyképesek, ha ismereteink naprakészek, így nagy tudásra és tapasztalatra van szükség. Mivel még nem rendelkezem mély szoftverfejlesztési tapasztalattal, ezért hosszú út áll előttem még. Úgy érzem, hogy dolgozatom elkészítésével, és azzal, hogy a Debreceni Egyetem Informatikai Karán dolgozó nagyszerű oktatóitól tanulhattam, megtettem az első lépéseket a cél felé.

5 Irodalomjegyzék

1. Harald Störrle: UML2. Panem Könyvkiadó, Budapest, 2007
2. Ian Sommerville: Szoftverrendszerek fejlesztése. Panem Könyvkiadó, Budapest, 2002
3. Imre Gábor: Szoftverfejlesztés Java EE platformon. Szak Könyvkiadó, 2007
4. Java EE online tutorial: <http://java.sun.com/javaee/6/docs/tutorial/doc/>
5. Kékes Ede – Surján György – Balkányi László – Kozmann György: Egészségügyi informatika. Medicina Könyvkiadó, Budapest, 2000
6. Országos Egészségügyi Pénztár: <http://www.oep.hu/>

6 Függelék

Vízió, általános leírás (1. ábra)

A MedHome házi orvosok által használt többfelhasználós számítógépes rendszer, mely lehetővé teszi az alapellátás részét alkotó házi orvosi szolgálat beteg nyilvántartási, adatszolgáltatási kötelezettségeit (lásd dokumentációs feladatok). Továbbá az alkalmazás hozzáférést biztosít adott házi orvos munkáját segítő ápolóknak is, így MedHome szoftvert kétféle személy használhatja azonos jogkörökkel: házi orvos, ápoló. A rendszer webes felületen érhető el. A felhasználók által elérhető főbb funkciók:

1. Praxisba bejelentkezettek rendelésének levezetése

- Személyes adatok felvétele (törzskarton)
- Betegtalálkozások rögzítése (betegkarton)
 - i. anamnézis
 - ii. státusz
 - iii. diagnózis
 - iv. terápia
- Keresőképtelenség dokumentálása
- Receptírás
- Gondozási nyilvántartás
- Igazolások
- Táppénzjelés
- Beutalók
- Orvoshívások rögzítése
- Betegszállítási, utazási utalvány
- Krónikus betegségek
- Fertőző betegségek
- Gyógyszer allergiák nyilvántartása
- Állandó gyógyszerek
- Alkalmassági véleményezés

2. Praxisba nem bejelentkezettek (ambuláns) betegek rendelésének levezetése

3. Jelentések készítése OEP felé

- Orvosi napló
 - Ambuláns napló
 - Táppénz
4. Statisztikák készítése a praxis betegeinek állapotáról
 5. Gazdasági ügyek
 - gyógyszer-, vegyszer-, kötszernyilvántartás
 6. Törzsadatok karbantartása
 - Beteg adatait érintő változások kezelése
 - Praxisba kijelentkezés, bejelentkezés
 - BNO kód

Fogalomszótár (2. ábra)

praxis	Háziorvosi szolgálat. Meghatározott személyek körének elsődleges, személyes és folyamatos háziorvosi ellátására létrehozott, működési engedély alapján működtetett, a 4/2000. (II.25.) EüM rendeletben előírt személyi feltételekkel rendelkező szervezeti egység.
háziiorvosi (praxis) szolgálat azonosítója	Megyei, vagy fővárosi ANTSZ által kiadott kód. A kód a háziorvosi szolgálat adatainak változása során sem változhat.
működtető kódja	OEP által kiadott finanszírozáshoz használt kód.
háziiorvos	A rendszer egyik típusú felhasználója. A háziorvosi szolgáltatásban foglaltak ellátását végzi.
háziiorvos azonosítója	Az orvos országos nyilvántartási száma (pecsétszám).
ápoló	A rendszer másik típusú felhasználója. A háziorvos tevékenységét segíti.
páciens (beteg)	Háziorvosi szolgálatot igénybe vevő személy.
törzskarton	Praxisba bejelentkezésnél a törzskarton szolgál a személyes adatok, alap egészségügyi adatok nyilvántartására. (3. ábra)
betegkarton	A betegkartonon minden orvos- beteg találkozást, minden történést, vizsgálatot, esetleges kórházi kezeléseket, beállított gyógyszeres kezelést rögzíteni kell.
anamnézis	A beteg szubjektív panaszainak,

	állapotváltozásainak, felismert betegségeinek felsorolása.
státusz	A beteg jelentkezésekor elvégzett fizikális, vagyis természetes (megfigyelés, tapintás, hallgatóság) vizsgálatainak rögzítése.
diagnózis	A vizsgálatok végeredményeként megállapított betegség, illetve betegségek (BNO kód).
terápia	A betegség kezelésének leírása.
keresőképtelenség	Aki betegsége, vagy gyermeke betegsége miatt, járványügyi megfigyelés miatt kereső foglalkozását ellátni nem tudja.
recept	Az orvosnak megszabott formában írt utasítása a gyógyszerész számára a gyógyszer, gyógyászati segédeszköz kiszolgáltatásához!
gondozási nyilvántartás	A házi orvos az idült (állandó, hosszantartó) betegségben szenvedőket gondozhatja.
igazolás	Iskolai hiányzás, orvosnál megjelenés, valamilyen kérelemhez igazolás, ahova orvosi diagnózist kell igazolni
táppénz	Aki betegsége, vagy gyermeke betegsége miatt, járványügyi megfigyelés miatt kereső foglalkozását ellátni nem tudja az keresőképtelen és ezen idő alatt táppénzt kap (1997. évi LXXX. tv. szabályozza)
beutaló	A házi orvos tovább küldheti betegét. Szakrendelésre, fekvőbeteg gyógyintézetbe irányítás igazolása.
orvoshívások	A házi orvost házhoz is hívhatják a betegek, hozzátartozók.
betegszállítási, utazási utalvány	Ha a beteg saját lábán el tud utazni a szakrendelőbe, kórházba, amennyiben a

	szakrendelés más városban van, a TB fedezi az utazás költségét. 1997. évi LXXX. Törvény szabályozza.
krónikus betegségek	Hosszantartó, krónikus betegségek
fertőző betegségek	A 18/1998. (VI.3) NM rendelet szabályozza, hogy melyek azok a fertőző betegségek, amiket jelenteni kell.
gyógyszer allergiák	Gyógyszerérzékenységek.
állandó gyógyszerek	A páciens által folyamatosan használt gyógyszerek.
alkalmassági véleményezés	A 6/1992. NM rendelet alapján a házi orvos alkalmassági vizsgálatokat is végez: <ul style="list-style-type: none"> • gépjárművezetéshez • lőfegyver tartásához • munkaköri, és egyéb.
ambuláns beteg	A házi orvosnál nem bejelentkezett páciens.
BNO kód	Betegségek nemzetközi osztályozása. Az adatok statisztikus feldolgozását segíti. (kódok 5 jegyűek, itt Magyarországon ez az elfogadott)
orvosi napló	A házi orvos betegforgalmáról a betegforgalmi napló ad számot, ez az alapja az év végi elszámolásnak, amikor az eltelt év munkájáról kell elkészíteni különféle jelentéseket.
ambuláns napló	A házi orvosnál nem bejelentkezett eseti ellátásokról köteles vezetni. 2 példányban állítja ki (perforált). 1-et megkap a beteg és aláírja az ellátást, 1 marad az ellátó orvosnál.
B300 tételes jelentés	OEP felé tett jelentés, mely a finanszírozási tényezők meghatározásához szükséges.
Praxis modul	A rendszer praxisba bejelentkezettek ellátását végző modulja, mely globális webes interfész

	nyújt a felhasználók számára.
Ambuláns modul	A rendszer praxisba nem bejelentkezettek ellátását végző modulja, mely globális webes interfész nyújt a felhasználók számára.
OEP modul	Társadalombiztosítási jelentések kezelését végző modul, mely globális webes interfész nyújt a felhasználók számára.
Statisztikai modul	A páciensek állapotáról lekérdezéseket, statisztikákat kezelő modul, mely globális webes interfész nyújt a felhasználók számára.
Gazdasági modul	Gazdasági ügyek kezelését végző modul, mely globális webes interfész nyújt a felhasználók számára.
Karbantartási modul	Páciens adatok karbantartását, BNO kódtábla karbantartását, páciensek ki-, bejelentkezését kezelő modul, mely globális webes interfész nyújt a felhasználók számára.

Törzskarton (3. ábra)

TÖRZSKARTON										Szül. dátum		
Karton/Páciens reg. sorszám: ✓				Vezetéknév: ✓				Utónév: ✓		Személyi azonosító		✓
Leánykori név: ✓		Anyja neve: ✓		Családi állapot: ✓		Tb. kártya azonosító		Tb. fiz. hely		✓		
Lakcím:	Város	Utca	hsz.	e/a.	Telefonszám:							
Foglalkozás: ✓		Munkahely: ✓		Bejelentkezés: ✓		Kijelentkezés: ✓		Praxis azonos.:				
Cave / Allergia		Vércsoport		Rh		Terv. ell. sz. ✓		Tipus		✓		
Krónikus betegség: Diagnózis ✓			BNO			Év			Rizikószűrés _____ év			
Nőgyógyászati szűrés dátumai			1. _____ 2. _____			3. _____ 4. _____			Tudószűrés dátumai			
Tudószűrés dátumai			1. _____ 2. _____			3. _____ 4. _____			Ervényes			

Családi anamnézis								Egyéni anamnézis előző betegségek / műtétek		
	Él - éves	EXIT évesen	MALIGN.	KARDIOV.	ANYAGCS.	PSYCH.	EGYÉB	Év	BNO/WHO	
Apa										
Anyja										
Testvér										

ALAPSTATUSZ ✓ <i>Közelítő</i>	Szív:
Bőr:	Erek állapota:
Nyálkahártyák:	Vizenyő:
Szájüreg:	Vérnyomás:
Nyelv:	Mellkas:
Garat:	Tudó:
Fogazat:	Idegrendszer:
Emlő:	Érzékszervek:
Has:	Mozgásszervek:
Rect. dig:	
EKG:	

KÖTELEZŐ KITÖLTENI:

- Páciens regiszter sorszáma (Napló a finanszírozáshoz, 1-től folyamatos sorszám a bejelentkezés sorrendjében, ezt célszerű a betegkartonon is rögzíteni.)
- Név, leánykori név, anyja neve
- Születési dátum
- Társadalombiztosítási kártya szám – TAJ szám
- Családi állapot
- Foglalkozás
- Munkahely
- Lakcím
- Be és kijelentkezés dátuma, indoka (1-4)
- Praxisazonosító
- Tervezett ellenőrzések számát (1-5)
- Krónikus betegségek diagnózisait
- Status (Bejelentkezés időpontjában aktuális egészségi állapot, családi anamnézis)

Szakterületi kapcsolatok, folyamatok (4. ábra)

Regisztráció:

- **praxis:** a rendszer használata előtt a praxist regisztrálni kell
- **házi orvos:** a házi orvos a már rendszerben szereplő praxisba regisztrálja magát, így a rendszer használatára jogosulttá válik
- **ápoló regisztrációja:** a házi orvos tevékenységét segítő személy regisztrálása a rendszerben, ezáltal ugyanazokkal a jogkörökkel használhatja az alkalmazást

Belépés: a felhasználó (házi orvos, ápoló) megadja a belépési adatokat

Praxisba bejelentkezettek rendelésének levezetése: a rendszer azon modulja, mely adott praxisba tartozó páciensek rendelését kezeli

- **Személyes adatok felvétele:** ennek során a felhasználó a szükséges adatok megadásával regisztrálja a páciens (törzskarton)
- **Betegalákosások rögzítése:** a páciens házi orvosának felkeresése során alkalmazott eljárás, melynek során az alábbi adatkörök kerülnek tárolásra (betegkarton):
 - anamnézis
 - státusz
 - diagnózis
 - terápia
- **Keresőképtelenség dokumentálása:** a keresőképtelenséget a házi orvos bírálja el, annak érdekében, hogy a páciens táppénzt kapjon. Ezt dokumentálja, és igazolást ad
- **Receptírás:** annak érdekében, hogy a páciens a számára szükséges gyógyszereket megkaphassa, a házi orvos megszabott formájú utasítást ad ki (recept), mely a gyógyszerészt utasítja a gyógyszer, segédeszköz kiszolgáltatására
- **Gondozási nyilvántartás:** a házi orvos az idült betegségekben szenvedőket nyilvántartja a rendszerben
- **Igazolás kiadása:** különböző célú igazolásokat adhat ki a házi orvos: iskolai hiányzás, orvosnál megjelenés, valamilyen kérelemhez igazolás, ahol az orvosi diagnózist kell igazolni
- **Táppénzelés:** a páciens keresőképtelensége miatt táppénz kap. Ezt a házi orvosnak dokumentálnia, regisztrálnia kell

- **Beutalók kiadása:** a házi orvos páciensének további vizsgálatát elrendelő dokumentum kiadása: szakrendelésre, fekvőbeteg gyógyintézetbe
- **Orvoshívások rögzítése:** a házi orvos házhoz hívásainak nyilvántartása
- **Betegszállítási, utazási utalvány kiadás:** a TB fedezi az utazási költséget a páciens számára, amennyiben saját lábán nem tud elmenni a szakrendelőben (más városban van)
- **Krónikus betegségek nyilvántartása:** a páciensek hosszantartó, krónikus betegségeinek dokumentálása
- **Fertőző betegségek nyilvántartása:** a páciensek fertőző betegségeinek dokumentálása
- **Gyógyszer allergiák nyilvántartása:** ha a páciens valamilyen gyógyszerérzékenységgel rendelkezik, akkor ezt feltétlenül rögzíteni kell a rendszerben
- **Állandó gyógyszerek:** a beteg által folyamatosan használt gyógyszerek nyilvántartása
- **Alkalmassági véleményezés:** a házi orvos különböző alkalmassági vizsgálatokat végez, és ennek eredményét dokumentálja a rendszerben, és erről igazolást ad ki

Praxisba nem bejelentkezettek (ambuláns) betegek rendelésének levezetése: a rendszer azon modulja, mely a praxisba nem bejelentkezettek rendelését kezeli. A házi orvos a hozzá nem bejelentkezetteket is köteles ellátni, amennyiben annak ellátása sürgősségi eset, ezt dokumentálni kell a szükséges adatokkal

Jelentések készítése OEP felé: a rendszer azon modulja, mely különböző jelentések generálásáért felelős. Havonta a házi orvos betegforgalmáról számot ad, ez az alapja az elszámolásnak:

- B300 tételes betegforgalmi jelentés
- Orvosi napló
- Ambuláns napló
- Táppénz

Statisztikák készítése a praxis betegeinek állapotáról: a rendszer azon modulja, mely az egyéb házi orvosi tevékenységek (prevenció, egészségnevelés) menedzselését segítő statisztikákat készít

Gazdasági ügyek: a rendszer azon modulja, mely a háziiorvosi szolgálattal kapcsolatos gazdasági ügyeket kezeli

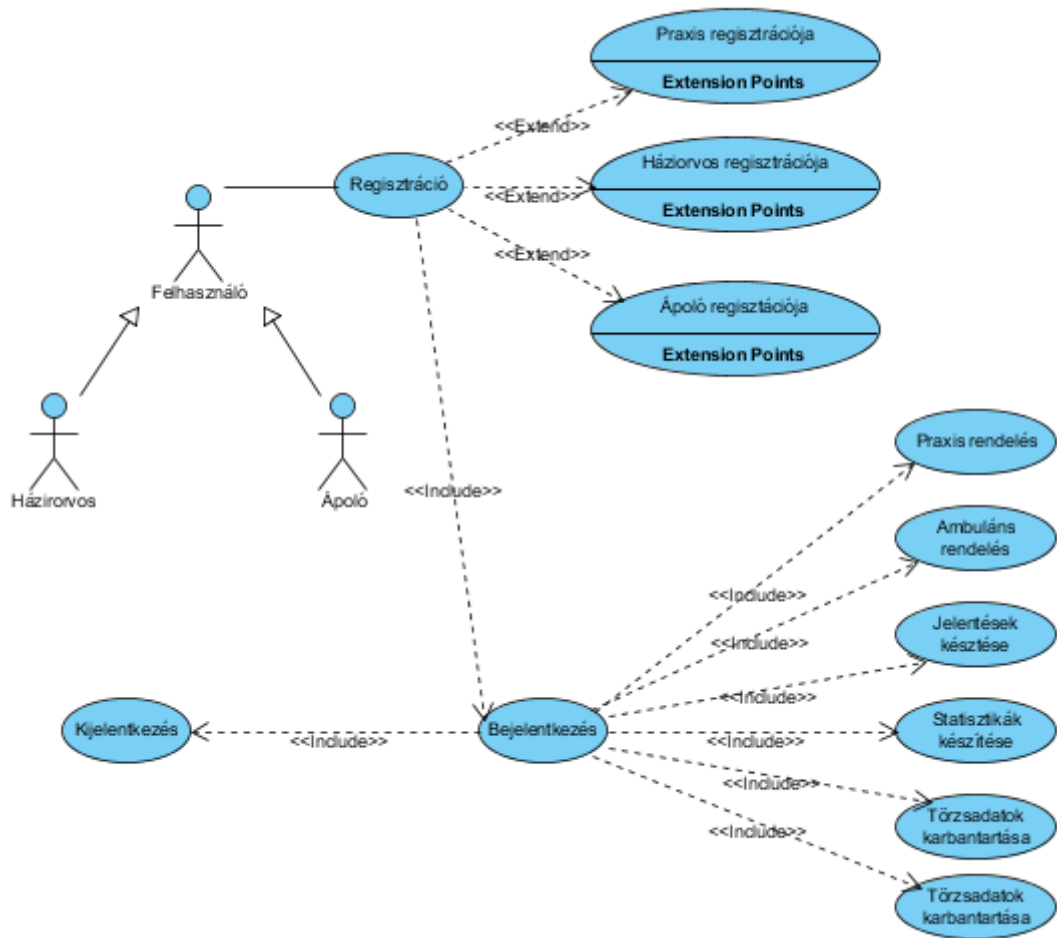
- gyógyszer-, vegyszer-, kötszernyilvántartás

Törzsadatok karbantartása: a rendszer mögött álló törzsállományok kezelését, karbantartását végző modul

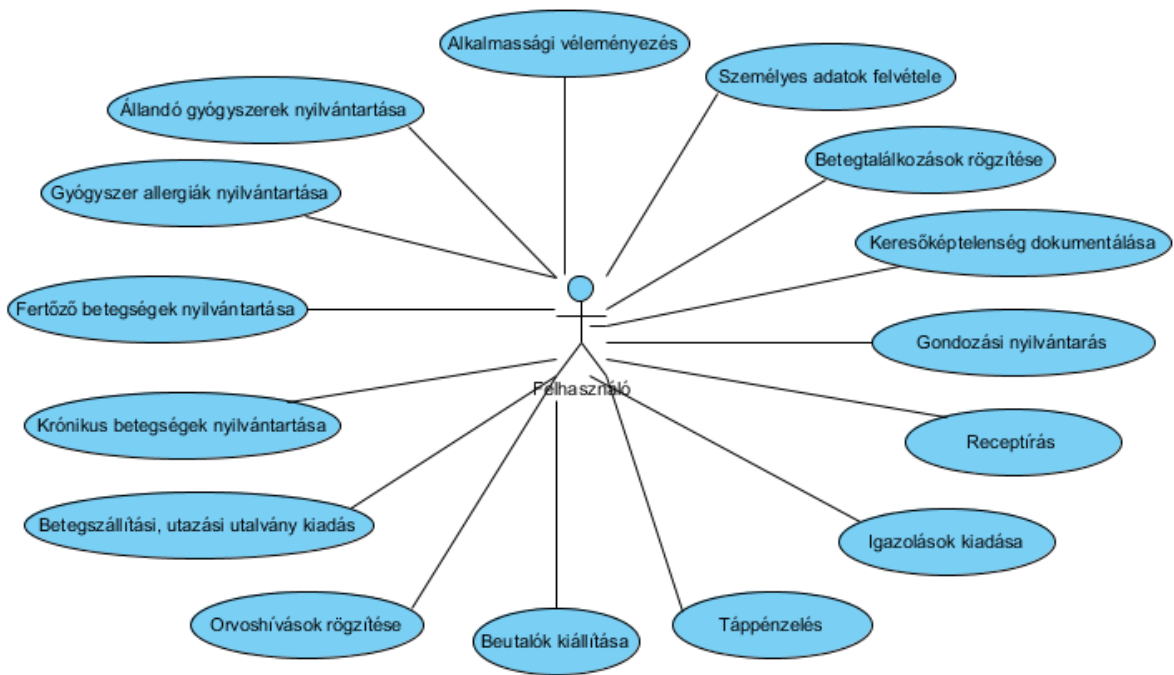
- Beteg adatait érintő változások kezelése: a páciens adatokban, egészségügyi adatokban történt változások regisztrálása
- Praxisba kijelentkezés, bejelentkezés: új páciens felvétele a praxisba, illetve háziiorvos változtatás esetén a páciens kijelentkeztetése a rendszerből
- BNO kód

Kijelentkezés: a felhasználó a rendszer használatának befejezésével kijelentkezik a rendszerből

Használati eset diagramok (5. ábra)

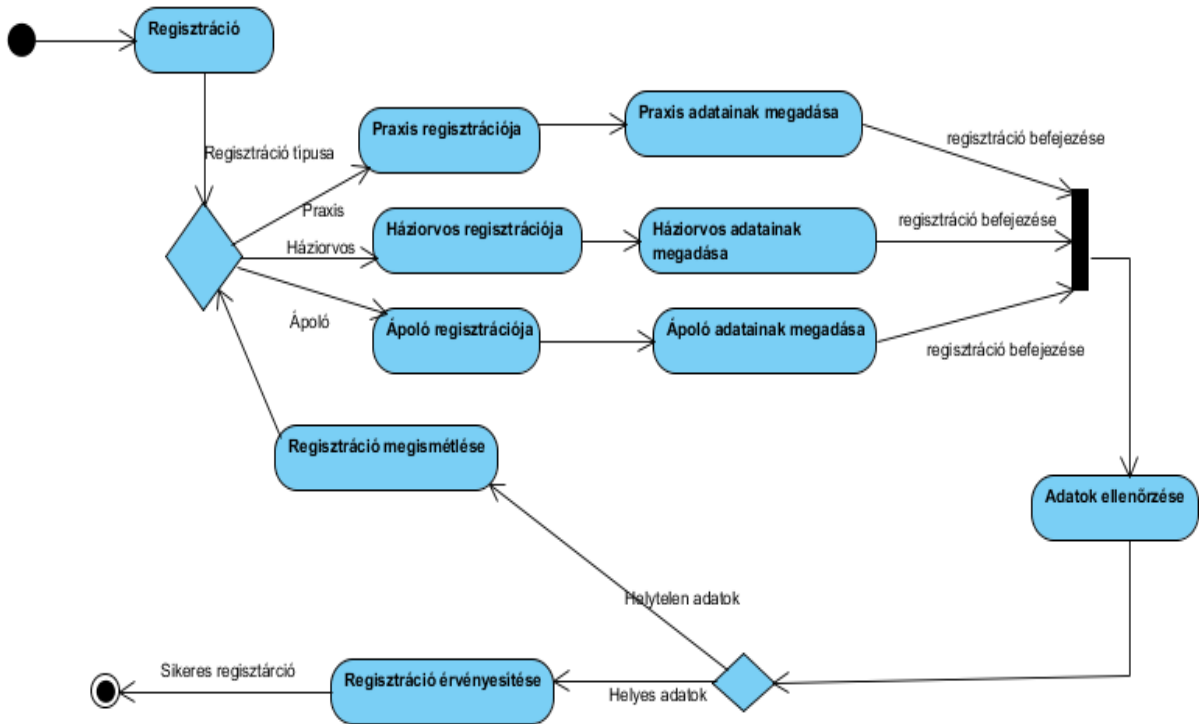


(Főbb funkciók modellezése)

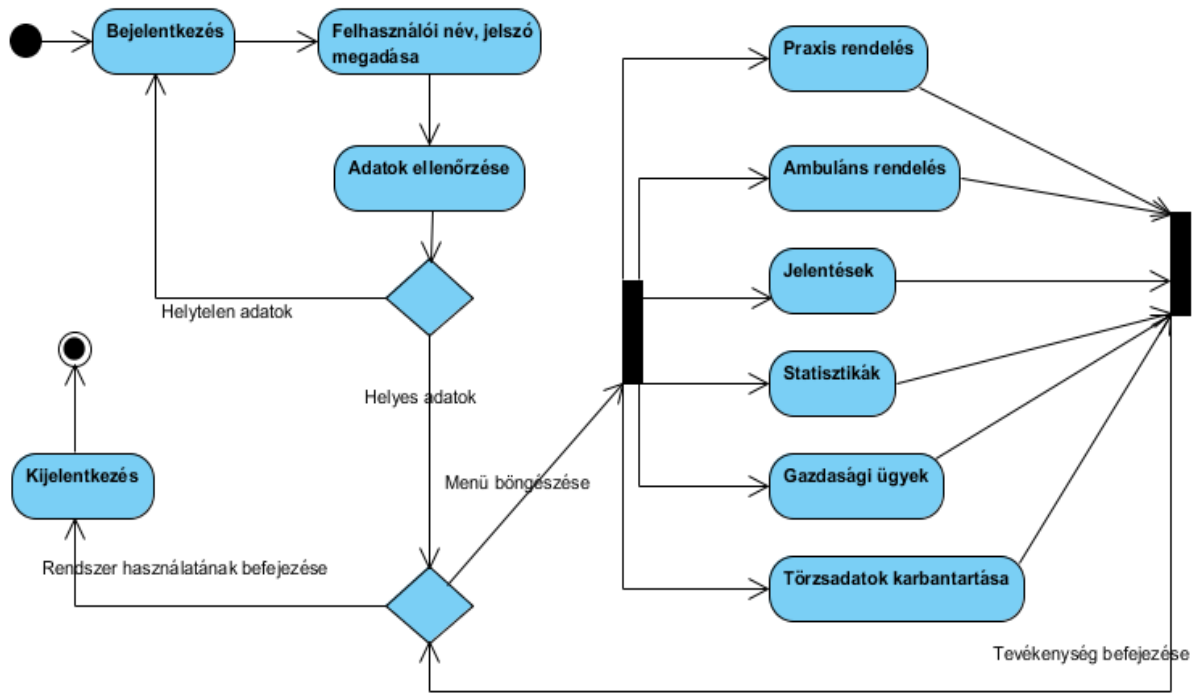


(Praxis funkció kibontása)

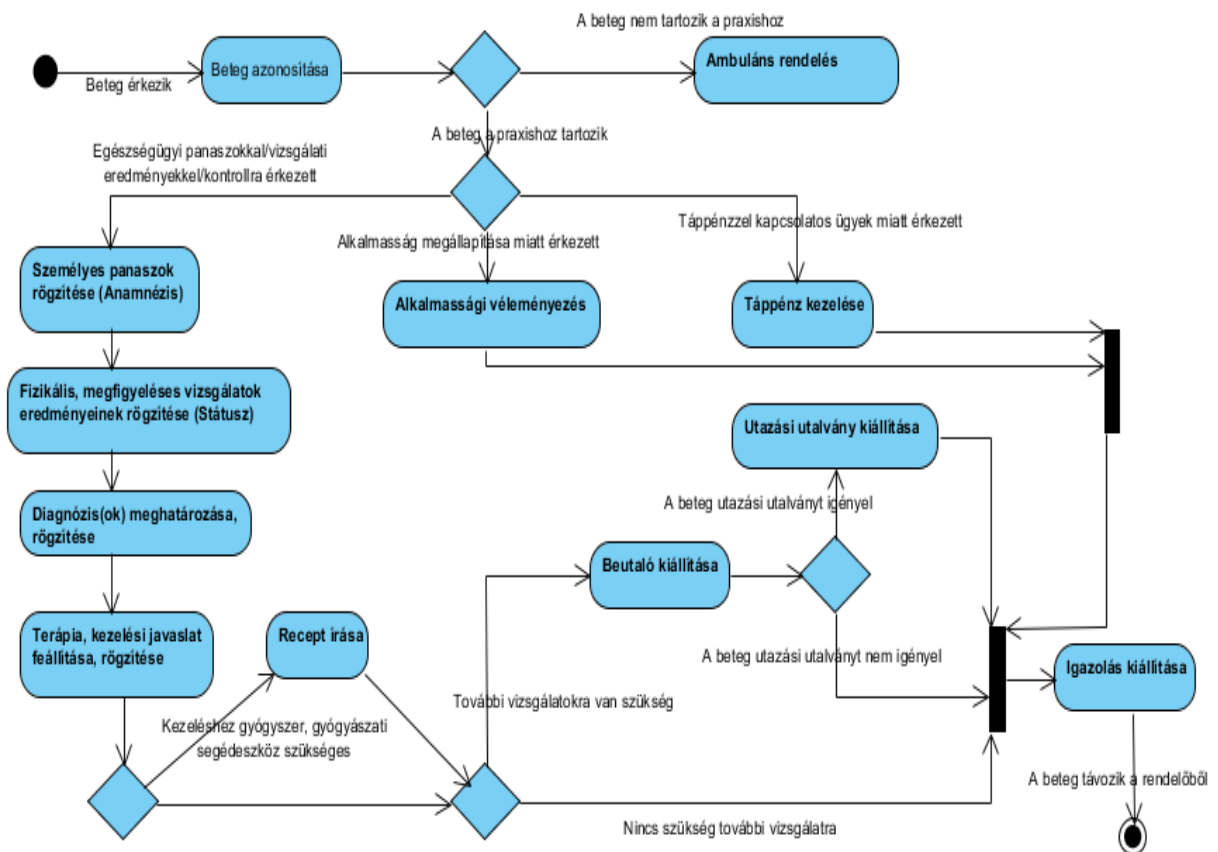
Tevékenység diagramok (6. ábra)



(Regisztráció tevékenység modellje, tovább granulálható)



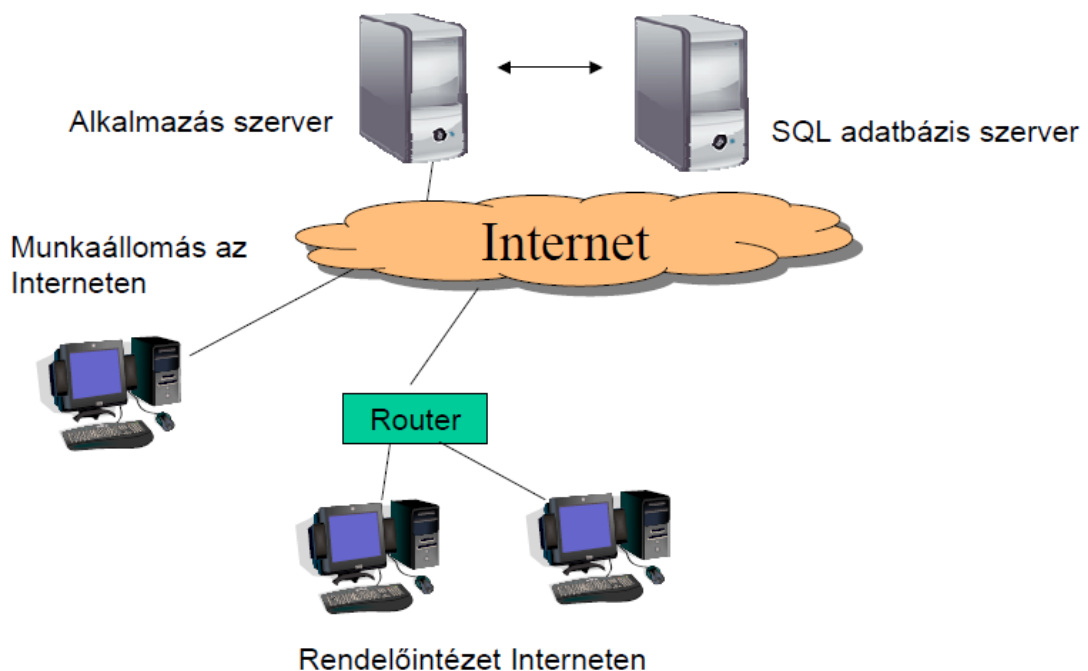
(A rendszer használatát modellezi: durva szemcsészettség)



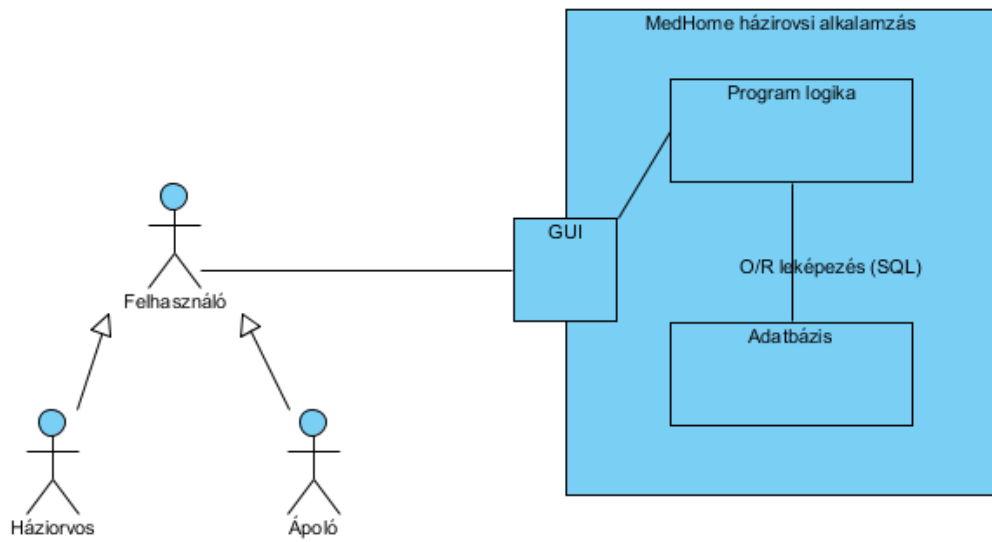
(Praxis rendelés folyamatának tevékenység diagramja, melyben azokat a tevékenységeket emeltem ki, amikor a beteg egészségügyi panasszal, vagy vizsgálati eredményekkel, vagy kontrollra, gondozásra érkezik a rendelőben)

Rendszer követelmények (7. ábra)

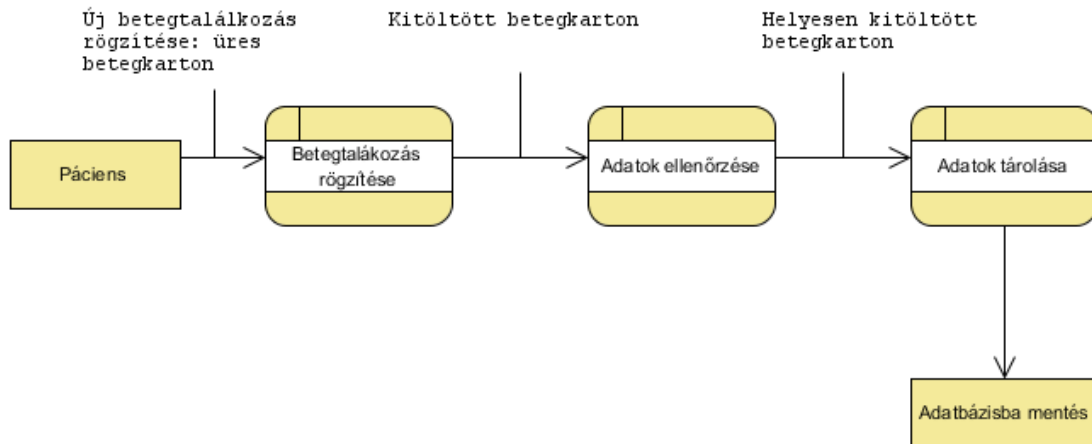
- Szoftver: A webes kialakításnak köszönhetően a szoftver platformfüggetlen. Bármely grafikus operációs rendszeren használható, mely képes az internetre kapcsolódni. Néhány gyakorta használt rendszer, melyeken működik: Windows 98, 2000, XP, Vista, Windows 7, Linux
- Célhardver: A szoftver igényeihez fognak igazodni, tervezéskor nem kell figyelembe venni
- Hálózati kapcsolat: A rendszer kialakításakor a minimális adatforgalom sarkalatos pontot jelentett, hogy minél kisebb Internetes sávszélességen is használható sebességgel fusson a rendszer
- Adatok biztonsága: az adatok tárolása perzisztens (tartós) módon központi rendszerben történik. Az adatok tárolása egy magas rendelkezésre állású szerveren, hibátűrő háttértáron történik. A felhasználó a rendszerbe való bejelentkezés alkalmával felhasználói nevet és jelszót ad meg, így azonosítja magát a rendszerben. Az egyes praxisok külön rendszerként jelennek meg a felhasználó oldalon, nem látnak bele egymás adataiba, és betegeik adataiba, dokumentumaiba
- Felhasználók száma várhatóan: 50 fő
- A rendszer felépítése:



Montázsvégi diagram (8. ábra)



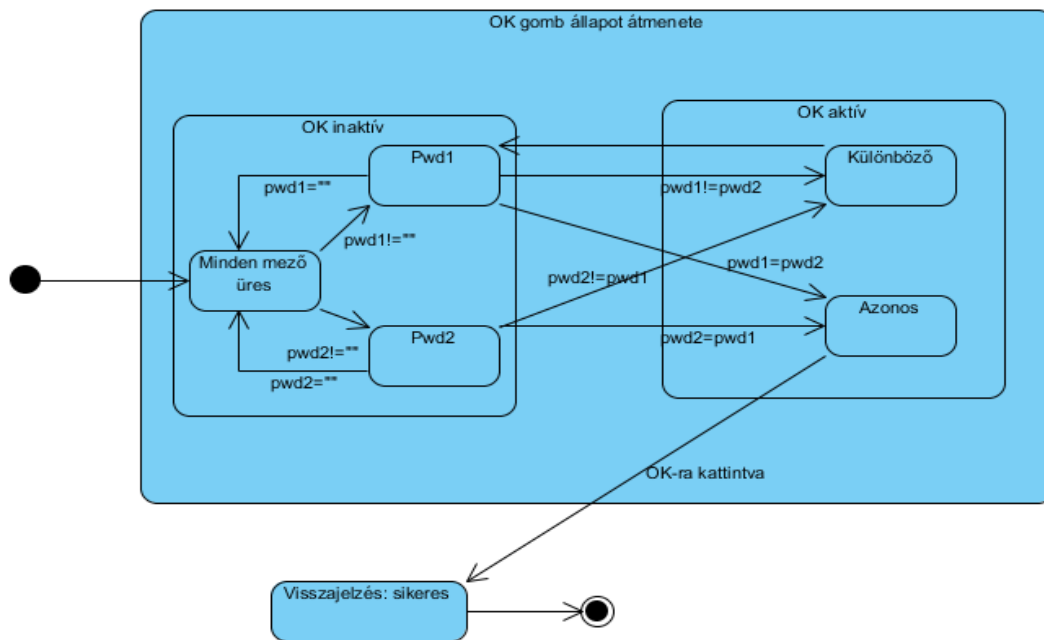
Adatfolyam diagram (9. ábra)



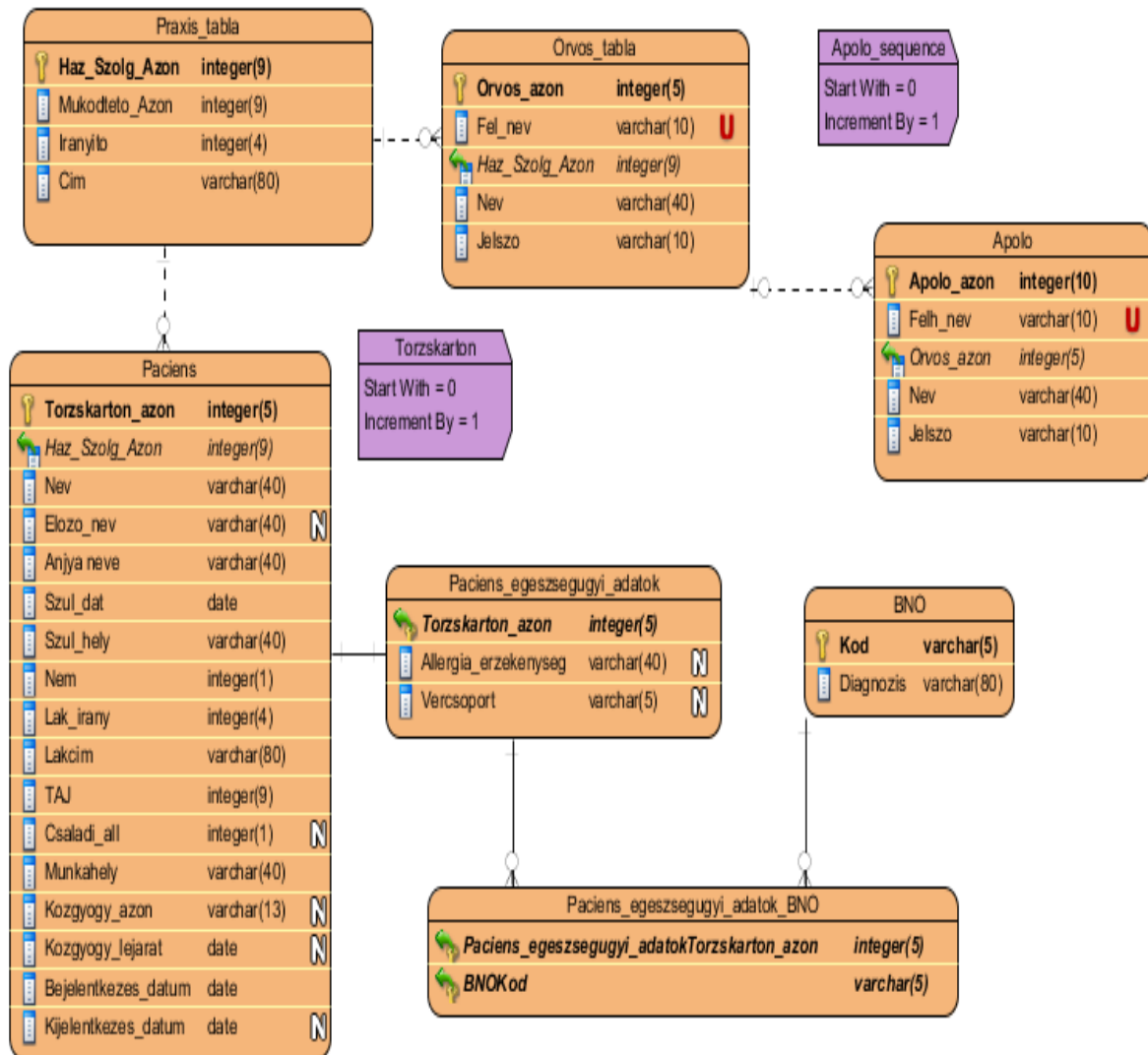
Betegkarton kitöltéséhez szükséges adatok:

- Praxis-páciens viszonya
- Betegforgalmi napló sorszáma
- Ellátás helye
- Ellátás oka
- Térítési kategória
- Diagnosztikai vizsgálatkérések
- Vizsgálat eredmények
- Anamnézis
- Státusz
- Diagnózis típusa
- Diagnózis
- Terápia

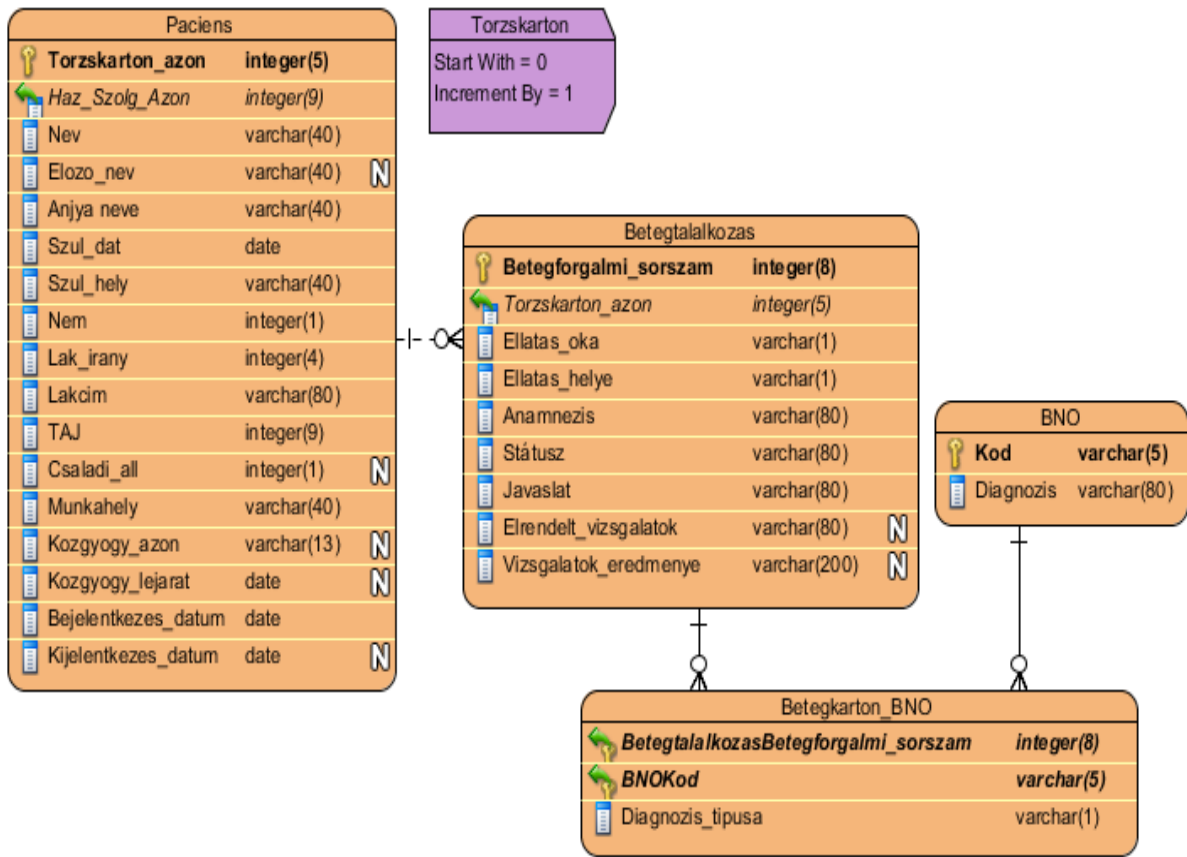
A regisztráció 'OK' gombjának állapotátmenet modellje (10. ábra)



ER adatbázis modell – a legalapvetőbb táblák (11. ábra)

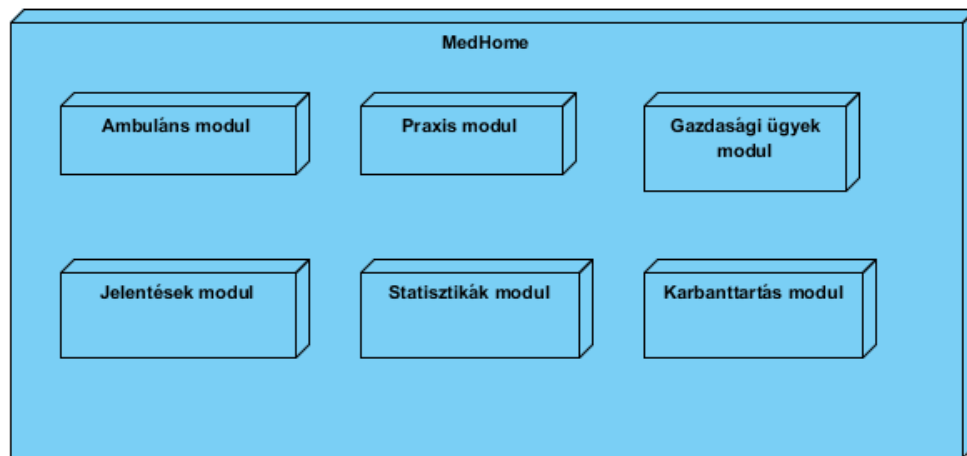


(Legfontosabb adatok tárolásához szükséges táblák ER modellje)

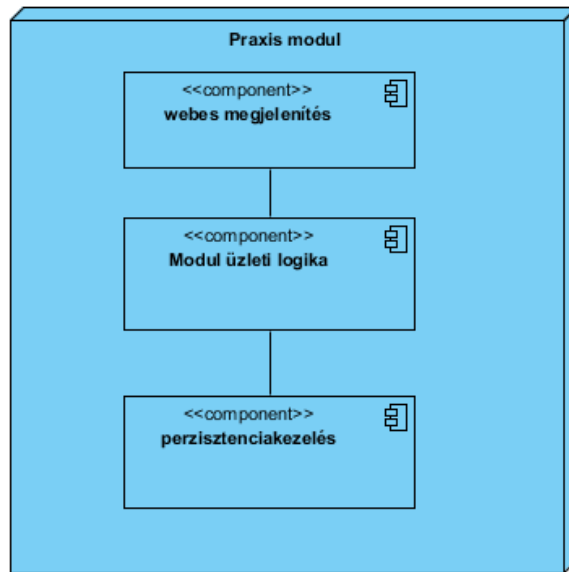


(Betegtalálkozások rögzítéséhez szükséges táblák ER modellje)

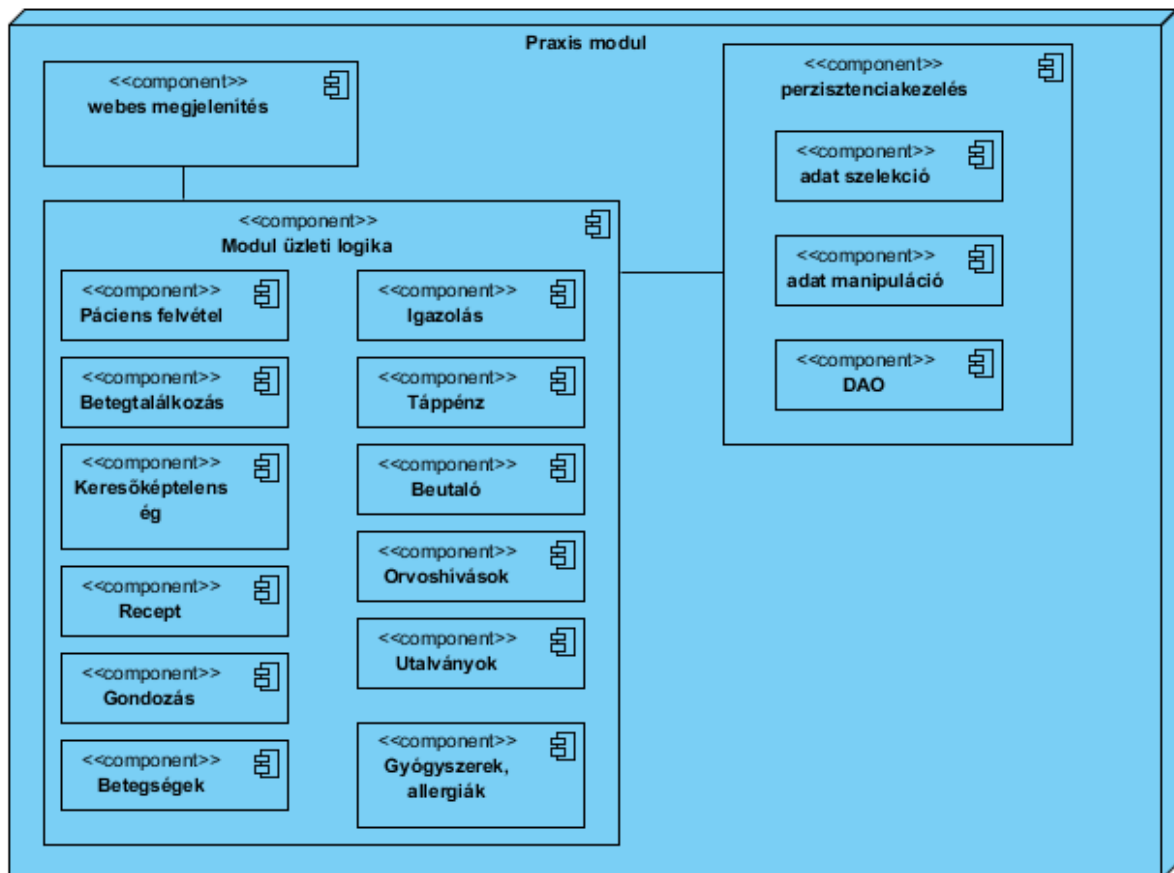
Deployment diagram (12. ábra)



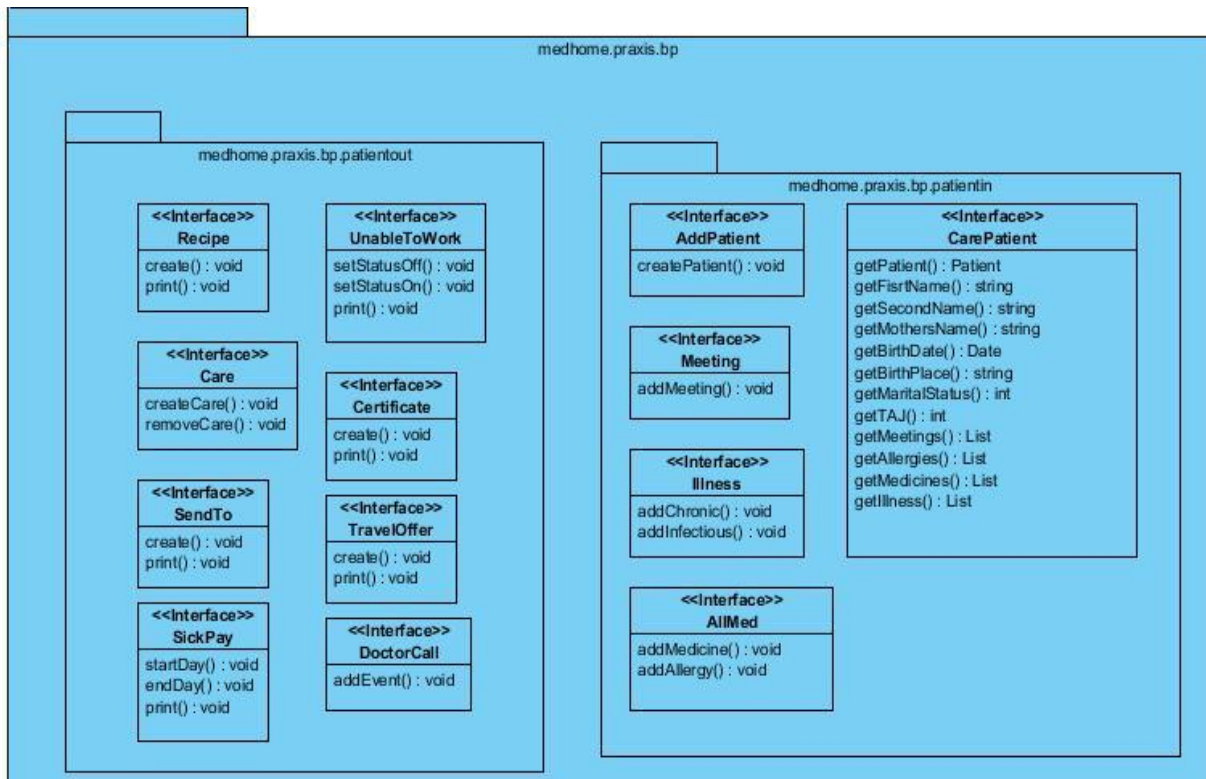
Komponens diagram – Praxis modul (13. ábra)



Komponens diagram – Praxis modul tovább kibontva (14. ábra)



Praxis alrendszer osztálydiagramja - interfészdefiniciók (15. ábra)



Köszönetnyilvánítás

Szeretnék köszönetet mondani mindazoknak, akik segítsége, munkája, pozitív hozzáállása, tanítása nélkül nem juthattam volna el idáig, és ez a dolgozat sem született volna meg.

Köszönettel tartozom Dr. Juhász István tanár úrnak a sok segítségért, továbbá Mócsánné Ambrus Katalinnak, és külső konzulensemnek, Hadházi Attilának segítő tanácsaikért. Szeretnék köszönetet mondani szüleimnek támogatásukért, és türelmükért.