

SZAKDOLGOZAT

Kittlinger Ádám

*Debrecen
2009.*

Debreceni Egyetem
INFORMATIKAI KAR

Webes alkalmazásfejlesztés FastFoodService

Témavezető:
Dr Kuki Attila
egyetemi adjunktus

Készítette:
Kittlinger Ádám
programtervező informatikus

Debrecen
2009.

Tartalomjegyzék

1. Bevezetés.....	4.
1.1 Témaválasztás.....	4.
1.2 A szakdolgozatban használt technológiák, szoftverek.....	4.
1.3 A rendszefejlésről általában.....	5.
2. Követelmények feltárása.....	7.
2.1 Követelmények csoportosítása.....	7.
2.2 Követelmény felderítési technikák.....	8.
2.3 Forgatókönyv.....	9.
2.4 Az áttekintés dokumentum.....	15.
2.5 Fogalomszótár.....	17.
2.6 Szakterületi kapcsolatok és folyamatok.....	18.
3. Tervezés.....	19.
3.1 A Deployment diagram.....	19.
3.2 A Use Case diagram.....	20.
3.3 Az aktivitás diagram.....	21.
3.4 A szekvencia diagram.....	23.
3.5 Az adatbázis tervezés.....	25.
4. Implementálás.....	29.
4.1 A PHP.....	29.
4.2 A FastFoodService webes alkalmazás implementálása.....	29.
5. Összefoglalás.....	33.
5.1 Köszönetnyilvánítás.....	33.
6. Irodalomjegyzék.....	34.
7. Függelék.....	35.
7.1 Az UML 2.0 alcsoportjainak hierarchikus szerkezete.....	35.
7.2 Néhány kép a FastFoodService alkalmazásról.....	35.

1. Bevezetés

Felgyorsult világunkban egyre nagyobb szerephez jut manapság a számítástechnika. Megbízhatóságának, gyorsaságának és újrafelhasználhatóságának köszönhetően életünket nagymértékben megkönnyítik az olyan új webes alkalmazások, amelyek felváltják a hagyományos, papír alapú szolgáltatásokat. Az internet egyre szélesebb körű elterjedése újabb és újabb igények kiszolgálását vonja maga után, ezért véleményem szerint, érdemes foglalkoznunk az informatika ezen régiójával.

1.1 Témaválasztás

Pontosan a fent elhangzottak miatt választottam a webes alkalmazásfejlesztés témáját, konkrétan egy olyan online szolgáltatást nyújtó weboldal létrehozását amely ételrendeléseket dolgoz fel. A honlapom neve FastFoodService, ezzel is utalva az oldal jellegére, amely hasonló egy webshophoz azzal a különbséggel, hogy itt a rendszer mögött egy étterem is áll.

1.2 A szakdolgozatban használt technológiák, szoftverek

Tanulmányaim során rengeteg számomra új technológiával ismerkedtem meg, mint például az Assembly, C és a Java programozási nyelvek, de felkeltették érdeklődésemet a HTML és a PHP szkriptnyelvek is sokoldalúságuk miatt. Ezért döntöttem úgy, hogy ezeket felhasználom a dolgozatíráshoz.

A rendszer tehát, webes mivoltjánál fogva, HTML-re épül melyben beágyazott PHP és Java szkriptek vannak. Ezek szerkesztése egyszerű jegyzetömbbel, vagy PSPad-dal. Továbbá szükséges egy webszerver, én az Apache-ot használtam. Valamint elengedhetetlen egy adatbázis-kezelő is, több szempontot figyelembe véve (nem túl nagy az adatbázis, ingyenesség), én a MySQL mellett döntöttem.

Mindezekből már össze lehet állítani egy működőképes webes alkalmazást, de nagyon fontos, hogy előtte megfelelően végezzük el a követelmények feltárását, amit UML diagramok segítségével gyönyörűen lehet reprezentálni. Ezekhez a diagramokhoz NetBeans 6.5-öt használtam.

Fontos továbbá, hogy jól megtervezzük az adatbázisunkat, ehhez a DBDesigner 4 adatbázis tervező programot használtam.

A kész alkalmazásomat Internet Explorer illetve Mozilla Firefox böngészőkön teszteltem localhost-on XAMPP segítségével de egy ingyenes webtárhelyre is feltöltöttem és kipróbáltam működés közben (<http://fastfoodservice.fw.hu/>).

A dolgozatom célja, egy teljes webes alkalmazás életciklusának bemutatása, ennek megfelelően az egész alkalmazást úgy hoztam létre, mint egy rendszerfejlesztési projektet, kiválasztottam a prototípus alapú inkrementális rendszerfejlesztési modellt, amely alapján a fejlesztés történt.

1.3 A rendszerefejlesztésről általában

Egy rendszer fejlesztése mindig egy komplex feladat, ezért több részre szokták bontani a folyamatot melyek a következők:

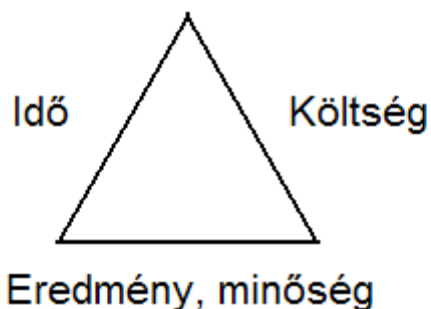
- ✓ Vízió
- ✓ Követelmények feltárása
- ✓ Elemzés
- ✓ Architektúrális elemzés
- ✓ Tervezés
- ✓ Implementálás
- ✓ Tesztelés
- ✓ Üzembe helyezés
- ✓ Üzemeltetés
- ✓ Karbantartás
- ✓ Evolúció
- ✓ Üzemen kívül helyezés

Ezek közül nem minddel foglalkozom a dolgozatban de megpróbálok minden fontosabb részt érinteni valamilyen szinten.

Egyre több tényező befojásolhatja a fejlesztést, minél nagyobb rendszerrel van dolgunk. Nagyobb rendszereket csapatokban fejlesztik, melyeknek tagjai változhatnak. Az eredeti követelmények is megváltozhatnak a fejlesztés során ezért fontos a szoftverfejlesztés folyamatán belüli fázisok menedzselése. Az informatikai projektmenedzsment ebben nyújt segítséget.

A szoftverfejlesztés heterogén, a fejlesztés lehet egyszerre több környezetben, különböző platformokon is folyhat, vannak elosztott rendszerek illetve különböző lehet a felhasznált technológia is, egy rendszeren belül.

Mivel egy projekt olyan összefüggő tevékenységek sorozata, amely valamilyen kitűzött eredmény elérésére irányul (most például egy webes alkalmazás megalkotása), meghatározott idő alatt végzendő el, és többnyire adott költségkerettel rendelkezik, ezért fontos hogy tisztában legyünk az úgynevezett projektháromszöggel ami a következő:



A három alapvető erőforrás minden projektben az idő, a költségkeret és az elkészült projektünk minősége. Ha ezek közül az egyik megváltozik akkor az biztos, hogy hatással lesz a másik két komponens közül legalább az egyikre.

Például csökkentünk a költségkeretet, akkor biztosan nem tudunk ugyanolyan

minőségű terméket (szoftvert) létrehozni csak ha növeljük a határidőket. Ma az idő és a költség az amire leginkább figyelnek a megrendelők és ez gyakran a minőség rovására megy.

Bizalmassági kérdések is felmerülhetnek minden projekttel kapcsolatban, de a megrendelőnek el kell hinni, hogy a szoftver úgy működik ahogy azt a gyártó állítja illetve fel kell tételezni, hogy a készítő nem él vissza az általa elérhető információkkal különben a megrendelő és a gyártó között etikai és szakmai konfliktusok léphetnek föl, amitől kudarcba fulladhat a projekt.

2. Követelmények feltárása

A követelmények határozzák meg, hogy mit várunk el a rendszertől. Ezeket mindenképpen még a tervezési fázis előtt egyeztetni kell a megrendelővel. Próbáljuk minimalizálni a félreértések számát és törekedjünk a teljességre. Fontos dolog, hogy tisztába legyünk azzal, hogy mit is akaraunk elkészíteni.

2.1 Követelmények csoportosítása

A követelményfeltárás során háromféle követelménnyel találkozhatunk, melyek a következők lehetnek:

- a) **Funkcionális követelmények:** a rendszer szolgáltatásaihoz kapcsolódnak, hogy milyen bemenethez milyen kimenetet kell produkálnia a rendszernek.
- b) **Nem funkcionális követelmények:** eredendő rendszertulajdonságok, nem tartoznak a rendszer részeihez általában. Könnyebben megfoghatóak és egyes esetekben sokkal lényegesebbek. Ez a csoport három alcsoportra bontható:
 - **Termék követelmények:** magára a rendszerre vonatkoznak: használhatósági, hatékonysági (teljesítmény, tárterület), megbízhatósági és hordozhatósági követelmények.
 - **Szervezeti követelmények:** nem magára a rendszerre vonatkoznak, hanem a környezet elvárásai: telepítési implementációs és szabvány követelmények.
 - **Külső követelmények:** a jogi, politikai, gazdasági, etikai, adatvédelmi és biztonsági követelmények tartoznak ide.
- c) **Szakterületi követelmények:** általában minden rendszerhez kötődik egy vagy több szakterület, amely beépül valamilyen szinten magába a rendszerbe. Például egy marketing rendszerben az üzleti szakzsargon.

Ezeket a követelményeket a megfogalmazásuk után verifikálni illetve validálni kell. A funkcionálisakat könnyebb, viszont a nem funkcionálisakat már nem egyszerű. Segítségünkre lehetnek a metrikák, mellyekkel mérhető lesz az adott követelmény, de sajnos a legtöbbhöz nem tudunk metrikát rendelni.

2.2 Követelmény felderítési technikák

A követelményeket többféle szempontból közelíthetjük meg. Meghatározásukra többféle módszert alkalmaznak.

Az egyik ilyen technika a **nézőpont-orientált követelmény feltárási módszer**. Ennek lényege, hogy a rendszert különböző, úgynevezett kulcsfigurák szemszögéből vizsgálják meg. Minden egyes kulcsfigura, más-más nézőpontot jelképez. Alapvetően 3 nézőpont csoport lehet:

- **Interaktor nézőpont:** azok az emberek és rendszerek, melyek majd közvetlenül érintkeznek a fejlesztendő rendszerrel. Speciális igényeik lehetnek melyeket figyelembe kell vennünk.
- **Közvetett nézőpont:** azok a kulcsfigurák, akik nem közvetlenül kapcsolódnak a rendszerhez, hanem csak a rendszer bizonyos funkcióinak eredményét használják. Tipikusan a döntéshozó menedzserek ilyenek például.
- **Szakterületi nézőpont:** különböző szabványok melyeket szintén nem hagyhatunk figyelmen kívül.

Miután felderítettük az egyes nézőpontokat, minden kulcsfigurához hozzárendeljük az adott személyeket és kikérdezzük őket. Nyilván ha nem személyről van szó, akkor van hozzá valamiféle dokumentáció amiből kiderül a számunkra fontos információ.

Egy másik lehetséges módszer a követelmények feltárására az **etnográfia** vagyis a megfigyelés. Az interaktor nézőpontokra vonatkozik, megfigyeljük, hogyan dolgoznak a végfelhasználók és begyűjtjük a szükséges dokumentumokat. Mindezekből leszűrjük a következtetéseket, hogy milyen követelményeket kell majd figyelembe venni a rendszer tervezésénél. Olyan helyeken lehet hasznos ez a technika, ahol másképp nem lehet „belelátni” a rendszerbe mert van egy formális kép a vállalat működéséről, de valójában az nem feltétlenül úgy működik.

Az etnográfiát gyakran ötvözik az **eldobható prototípusfejltséssel**. A prototípuskészítés akkor hasznos, amikor a megrendelő nem tudja pontosan behatárolni a követelményeket. Ekkor készítünk egy prototípust neki és azon pontosítva már világosabb képet adhat a fejlesztendő rendszerről. Sajnos ez elég költséges szokott lenni.

Gyakran használt módszer még az **interjúztatás** is. Személyes kikérdezésen alapszik. Van zárt és nyílt interjú:

- **Zárt interjú:** az adott kulcsfigura egy előre meghatározott kérdéssort kap. Hátránya, hogy csak azokra a kérdésekre kapunk választ amelyeket előre felteszünk.
- **Nyílt interjú:** nem használunk előre megírt kérdéssort, hanem beszélgetünk a kulcsfigurával ekkor az ott eszünkbe jutó kérdéseket tesszük fel.

Általában a zárt és a nyílt interjúk egy keverékét használják. Gyakran problémát jelenthet, hogy az adott szakterület kulcsfiguráinak sokminden triviális és ezért nem beszélnek bizonyos dolgokról, pedig lehet, hogy fontosak lennének a rendszer megfelelő működése érdekében. Az interjú eredményeképpen létrejön egy forgatókönyv.

2.3 A forgatókönyv

A forgatókönyv egy olyan dokumentum tehát, amellyel visszacsatolunk a megrendelőhöz és szembesítjük vele. Nem tartalmaz informatikai szakfogalmakat, teljesen köznyelven íródott, a rendszer követelményeinek pontosítására szolgál. Egy interakció sorozatot formálisan leírhatunk vele, struktúrált szöveg vagy diagram segítségével. A forgatókönyvnek a következőket kell tartalmaznia:

- mit várunk el a rendszertől,
- az interakció előtti feltételes állapot megadása,
- az interakció sorozat normál menetének leírása,
- a normál menethez képest mi romolhat el és ekkor a rendszer mit tesz (azaz a kivételkezelés),
- az adott interakció sorozattal egyidőben zajló párhuzamos tevékenységek leírása,
- a rendszer állapotának leírása az interakció sorozat végén.

A forgatókönyv tehát a rendszerhez kapcsolódó fontosabb eseményeket írja le, és mivel hétköznapi nyelven íródik ezért bármely megrendelő számára egyszerű és világos tényeket közöl.

Például az én webes alkalmazásom, a FastFoodService forgatókönyve a következőképp néz ki:

1. Regisztráció:

- **Feltétel:** a leendő felhasználó igénybe akarja venni a FastFoodService szolgáltatásait.
- **Menete:** a „Regisztráció” felíratra kattintva kitölti a felhasználó a regisztrációs űrlapot melyben megadja a felhasználónevét, jelszavát kétszer és e-mail címét. Majd a „Regisztráció” gombra kattintva a rendszer kiküld számára egy aktiváló levelet. Ha ezt megnyitja és a benne szereplő linkre kattint, akkor regisztrált felhasználóként felveszi a rendszer az adatbázisba.
- **Kivételkezelés:** ha nem tölti ki az összes mezőt a regisztrációs űrlapon akkor ezt a rendszer hibüzenettel jelzi és nem enged regisztrálni mindaddig míg ki nem tölti az összes mezőt az adott személy. Amennyiben a megadott két jelszó különbözik, vagy már a rendszerben létező felhasználónévvel vagy e-mail címmel próbálnak regisztrálni, abban az esetben szintén hibüzenetet kapunk és a regisztráció nem jön létre.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** az adott felhasználó most már bejelentkezhet, létrejött az adatbázisban.

2. Elfelejtett jelszó megújítása:

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már regisztrálva van a rendszerbe.
- **Menete:** a felhasználó az „Elfelejtettem a jelszavam” felíratra kattint és megadja a regisztrált e-mail címét, amelyre kiküld egy linket a rendszer, és erre kattintva megadhatja új jelszavát a felhasználó.
- **Kivételkezelés:** ha a rendszerbe nem regisztrált e-mail címet adunk meg akkor hibüzenettel jelzi a rendszer, hogy nem létezik tag ezzel az e-mail címmel.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** módosul az adott felhasználó jelszava az új, megadott jelszóra.

3. Bejelentkezés:

- **Feltétel:** csak az jelentkezhet be aki regisztrálva van a rendszerbe.
- **Menete:** a felhasználó beírja a „Név” mezőbe a felhasználónevét, a „Jelszó” mezőbe

a jelszavát és a „Login” gombra kattint.

- **Kivételkezelés:** ha a rendszerbe nem regisztrált felhasználó próbál belépni, vagy hibás jelszóval próbál belépni valaki akkor ezt hibaüzenettel jelzi a rendszer és nem engedi belépni őket.
- Rendszer állapota az interakciósorozat sikeres befejeződése esetén: A felhasználó be van jelentkezve és jogosultságainak megfelelően használhatja a rendszer szolgáltatásait.

4. Profilmódosítás:

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe.
- **Menete:** a felhasználó a „Profilmódosítás” felírra kattint és módosíthatja felhasználónevét, kötelezően meg kell adnia a jelszavát kétszer (új jelszó is lehet), megadhatja a nemét, weblapjának nevét és címét valamint születési dátumát majd ezeket a „Módosít” gombbal rögzítheti az adatbázisban.
- **Kivételkezelés:** ha a rendszerbe már foglalt felhasználónévre szeretnénk módosítani a sajátunkat akkor hibaüzenettel jelzi a rendszer és nem engedi. Ha nem adtuk meg jelszavunkat kétszer akkor nem engedélyezi a módosítások elmentését és hibaüzenetet ír ki.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** módosul az adott felhasználó profilja az adatbázisban.

5. Saját profil törlése:

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe és nem „admin” jogosultságokkal rendelkezik.
- **Menete:** a felhasználó a „Profilmódosítás” felírra kattint és ott a „Profil törlése” felírra kattintva törölheti a profilját.
- **Kivételkezelés:** ha nem sikerül valamiért a törlés azt hibaüzenettel jelzi ezt a rendszer.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** az adott profil törlődik az adatbázisból.

6. Szállítási adatok megadása:

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe és „user” jogosultságokkal rendelkezik.
- **Menete:** a felhasználó a „Rendelési adatok” felíratra kattint és megadhatja vagy módosíthatja a szállítási adatait (vezetéknév, keresztnév, város, cím, telefonszám) majd a „Mentés” gombra kattintva rögzítheti az adatbázisban.
- **Kivételkezelés:** ha a rendszerbe már léteznek pont ugyanezek a szállítási adatok akkor a módosítás nem történik meg és hibaüzenettel jelzi ezt a rendszer.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** módosulnak az adott felhasználó szállítási adatai az adatbázisban.

7. Ételfelrendelés:

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe és „user” jogosultságokkal rendelkezik és korábban már rakott legalább 1 ételt a kosarába.
- **Menete:** a felhasználó az „Kosár”-nál a „Megrendelés”-re kattint és ott megadja a szállítási dátumot majd a „Megrendel” gombra kattint.
- **Kivételkezelés:** ha nem sikerül valamiért a megrendelés felvétele azt a rendszer hibaüzenettel jelzi.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** az új megrendelés bekerül az adatbázisba.

8. Új étel kiírása:

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe és „chef” jogosultságokkal rendelkezik.
- **Menete:** a felhasználó az „Étel” menüpontra kattint és azon belül az „Új étel” ikonra ekkor megadja az új étel nevét, csoportját és árát majd az ikonra kattintva az új étel bekerül az adatbázisba.
- **Kivételkezelés:** ha a rendszerbe már létezik ilyen nevű étel, akkor a módosítás nem történik meg és hibaüzenettel jelzi ezt a rendszer.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** az új étel

bekerül az adatbázisba.

9. Új ételcsoport kiírása:

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe és „chef” jogosultságokkal rendelkezik.
- **Menete:** a felhasználó az „Étel” menüpontra kattint és azon belül az „Új csoport” ikonra ekkor megadhatja az új ételcsoport nevét, majd az ikonra kattintva az új ételcsoport bekerül az adatbázisba.
- **Kivételkezelés:** ha a rendszerbe már létezik ilyen nevű ételcsoport, akkor a módosítás nem történik meg és hibaüzenettel jelzi ezt a rendszer.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** az új ételcsoport bekerül az adatbázisba.

10. Étel adatainak módosítása:

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe és „chef” jogosultságokkal rendelkezik.
- **Menete:** a felhasználó az „Étel” menüpontra kattint és azon belül elnavigál az adott ételhez. Ha megtalálta akkor módosíthatja a nevét, csoportját és az árát is, majd a módosítás ikonra kattintva az étel adatai módosulnak az adatbázisba.
- **Kivételkezelés:** ha a rendszerbe már létezik ilyen nevű étel, akkor a módosítás nem történik meg és hibaüzenettel jelzi ezt a rendszer.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** az adott étel adatai módosulnak az adatbázisba.

11. Étel törlése:

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe és „chef” jogosultságokkal rendelkezik.
- **Menete:** a felhasználó az „Étel” menüpontra kattint és azon belül elnavigál az adott ételhez. Ha megtalálta akkor a törlés ikonra kattintva az étel törlődik.
- **Kivételkezelés:** ha nem sikerül valamiért a törlés azt hibaüzenettel jelzi ezt a rendszer.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** az adott étel törlődik az adatbázisból.

12. Rendelés törlése (chef):

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe és „chef” jogosultságokkal rendelkezik.
- **Menete:** a felhasználó az „Rendelések” menüpontra kattint és azon belül kiválasztja a törlendő rendelést és a törlés ikonra kattint.
- **Kivételkezelés:** ha nem sikerül valamiért a törlés azt hibaüzenettel jelzi ezt a rendszer.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** az adott megrendelés törlődik az adatbázisból.

13. Rendelés törlése (user):

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe és „user” jogosultságokkal rendelkezik.
- **Menete:** a felhasználó az „Rendelésem” menüpontra kattint és azon belül kiválasztja a törlendő rendelést és a „Megrendelés törlése” felíratra kattint.
- **Kivételkezelés:** ha nem sikerül valamiért a törlés azt hibaüzenettel jelzi ezt a rendszer.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** az adott megrendelés törlődik az adatbázisból.

14. Más profiljának módosítása:

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe és „admin” jogosultságokkal rendelkezik.
- **Menete:** a felhasználó a „Taglista” felíratra kattint és kiválasztja a profilt, rákattint a felhasználó nevére majd a „Profil szerkesztése” felíratra. Itt módosíthatja az adott felhasználó felhasználónevét, jelszavát, nemét, weblapjának nevét és címét valamint a születési dátumát. Miután beállította ezeket a „Módosít” gombra kattintva elmentheti a módosításokat az a adatbázisban.
- **Kivételkezelés:** ha a rendszerbe már foglalt felhasználónévre szeretnénk módosítani az adott profilhoz tartozó felhasználónevet akkor hibaüzenettel jelzi a rendszer és nem engedi.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** az adott profil adatai módosulnak az adatbázisban.

15. Más profiljának törlése:

- **Feltétel:** csak az veheti igénybe ezt a szolgáltatást, aki már bejelentkezett a rendszerbe és „admin” jogosultságokkal rendelkezik.
- **Menete:** a felhasználó a „Taglista” felírra kattint és kiválasztja a törlendő profilt, rákattint a törlendő felhasználó nevére majd a „Profil szerkesztése” felírra. Itt a „Profil törlése” felirattal törölheti az adott profilt.
- **Kivételkezelés:** ha nem sikerül valamiért a törlés azt hibaüzenettel jelzi ezt a rendszer.
- **Rendszer állapota az interakciósorozat sikeres befejeződése esetén:** az adott profil törlődik az adatbázisból.

2.4 Az áttekintés dokumentum

A forgatókönyv mellé hasznos lehet egy áttekintés dokumentum. Ez a köznyelven íródó dokumentum a fejlesztő és a megrendelő közti adategyeztetést biztosítja valamint pontosítja a követelmények megfogalmazását. A következő három részből áll:

1. **Általános leírás:** a rendszer egy általános, rövid köznyelvi leírása, amely tartalmazza működésének leírását, hogy mire való és milyen céllal készül.
2. **Általános követelmények:** ebben a dokumentumban szerepelnek, az olyan szabványok amelyeket az alkalmazásnak be kell tartania, és egyéb olyan általános követelmények, amelyeket a fejlesztőnek be kell tartania. Valamint tartalmazza azon dokumentumok listáját is, amelyet a rendszerrel együtt át kell adni majd a megrendelőnek.
3. **Rendszerkövetelmények:** azt a környezetet írja le, amely szükséges a szoftver működéséhez.

Az én alkalmazósom, a FastFoodService áttekintés dokumentumát mutatom be a következő pár oldalon.

1. Általános leírás

A FastFoodService egy étterem által használt számítógépes rendszer, amely nyilvántartja az ügyfelek adatait, valamint az ételrendeléssel kapcsolatos fontosabb információkat. A rendszer webes felületén, az ügyfelek rendelhetnek az adott étlapról vagy

leadhatják egy hétre, vagy egy hónapra előre a rendelésüket. A rendszert háromféle személy használja:

- ügyfél
- séf
- admin

Az ügyfelek rendeléseket adhatnak le, illetve módosíthatják az adatlapjukat. A séfek az adott étlapért felelősek, ők állítják össze az ételcsoportokat és viszik fel a rendszerbe az ételeket. Ki is törölhetnek ételeket vagy akár egész ételcsoportokat a rendszerből. Az árakat is a séfek kezelik. Az admin a rendszer adminisztrátora, hozzáférése van a felhasználók profiljához kitörölheti őket, vagy adatokat módosíthat, illetve ő osztja ki a jogosultságokat.

Az egész rendszer tulajdonképpen egy étterem és egy futárszolgálat egyben, ahol az ügyfelek online rendelései alapján futárok szállítják ki az étteremben elkészült ételeket.

2 . Általános követelmények

1. A kivitelező köteles minden üzleti logikához és felületi elemhez tartozó döntésükről kikérni cégünk egyetértését, hozzájárulását. A fejlesztés minden lényeges pontja, a fejlesztési eredmények, és a fejlesztés teljes dokumentációja elérhető kell hogy legyen a FastFoodService, vagyis a megrendelő számára. A fejlesztés végeztével a forráskód és a teljes dokumentáció tulajdonjoga a megrendelőt illeti meg.
2. A szoftverfejlesztés minden lépése az ISO IEC 90003 2004 Software Standard-nak megfelelően kell történnie, melynek következetes betartásából származó dokumentumok a FastFoodService számára hozzáférhetőek kell, hogy legyenek. A dokumentálás "OpenDocument Text" (ISO/IEC 26300:2006) formátumban kell történnie.
3. Alkalmazottaink egész napos munkájukat gépnél töltik, ezért lényeges a felületek szép kinézete, és a szemet nem zavaró színek, zavaró funkcióelrendezések kerülése.
4. A rendszer minden művelet eredményéről tájékoztatja a szoftver használóját. Hiba esetén értesítést ad a probléma okáról a felhasználó minőségétől függően
5. A szoftvernek a cég által használt szakterületi fogalmakat kell használnia melytől nem térhet el. Ennek részletes listázása a „Fogalomszótár” című dokumentumban található.
6. A rendszer minden kommunikációja biztonságos csatornán kell, hogy történjen.

3 . Rendszerkövetelmények

Operációs rendszer: Platform függetlennek kell lennie.

Célhardver: A szoftver igényeihez fognak igazodni. Szoftver tervezésekor nem kell figyelembe venni.

Várható felhasználók száma: 800 fő

2.5 A fogalomszótár

Az alkalmazásfejlesztés során rákényszerülünk különböző szakszavak használatára. Ezek akár többféle szakterületről is származhatnak és lehet, hogy más rendszerben mást jelölnek ezért a saját rendszerünknel konkretizálnunk kell őket. Ezt a fogalomszótár segítségével tehetjük meg.

Fogalomszótár – FastFoodService

- **Felhasználó:** olyan személy aki regisztrálva van FastFoodService adatbázisában. Háromféle jogosultsága lehet egy felhasználónak:
 - user
 - chef
 - admin
- **Profil:** a felhasználó felhasználóneve, jelszava, e-mail címe, neme, kora és weblapjának neve és címe.
- **Étel:** minden ételnek három attribútuma van:
 - neve
 - csoportja
 - ára
- **User (ügyfél):** olyan felhasználó, aki a rendszeren keresztül ételeket rendelhet.
- **Chef (séf):** olyan felhasználó, aki az ételekért felelős. Ő írja ki az ételek csoportját, nevét és árát.
- **Admin:** olyan felhasználó aki képes profilokat módosítani, törölni és jogosultságokat kiosztani.
- **Étlap:** az ételeket és azok árát tartalmazó lista.

2.6 Szakterületi kapcsolatok és folyamatok

A rendszerben zajló folyamatok egyszerű leírását hivatott bemutatni ez a köznyelven íródó dokumentum, mely igen hasznos lehet a rendszert fejlesztő programozók számára.

Szakterületi kapcsolatok és folyamatok – FastFoodService

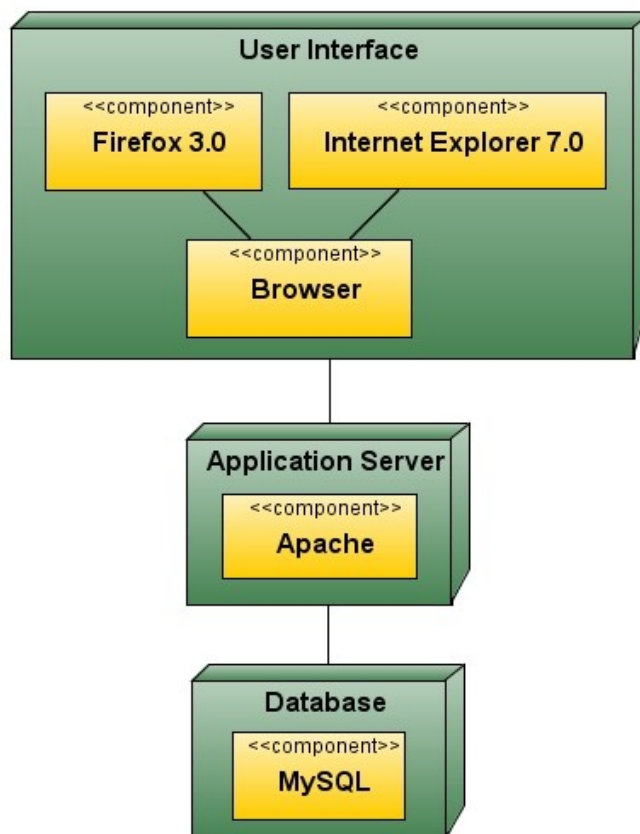
- Új étel kiírása: a séf feladata, az a folyamat amely során láthatóvá válik az új étel az ügyfelek számára az étlapon.
- Étél módosítása: a séf feladata, az a folyamat amely során megváltoznak az adott étel attribútumai.
- Étél törlése: a séf feladata, az a folyamat amely során az adott étel lekerül az étlapról.
- Megrendelés: az ügyfél feladata, az a folyamat amely során az általa kiválasztott ételeket megrendeli.
- Megrendelés törlése: az ügyfél vagy a séf feladata, az a folyamat amely során az adott megrendelés visszavonhatatlanul törlődik.

3. Tervezés

A tervezés, a követelményfeltárás után egy újabb mérföldkőnek számít az alkalmazásfejlesztési projektünkben. Az előző fejezethez hasonlóan ebben a fejezetben is megpróbálom kiemelni a fontosabb technikákat melyek segítenek, hogy minél jobban sikerüljön az alkalmazásunk. Az egyik ilyen technika az **UML** (Unified Modeling Language) használata. Az UML grafikus jelöléseket használ rendszerek absztrakt modelljének leírására. A szabvány UML jelölést használó diagramokat UML modelleknek nevezzük. Az UML 2.0 hierarchikus szerkezetű csoportokra bontható (lásd függelék 7.1-es ábra). Ezek közül pár hasznos modellt szeretnék bemutatni ebben a fejezetben.

3.1 A Deployment diagram

A telepítési diagram (Deployment diagram) egy olyan struktúrális diagram, amely a rendszerimplementációhoz használt hardvert, a hardverre telepített szoftverkomponenseket és azok viszonyát hivatott reprezentálni. A FastFoodService deployment diagramja a következőképp néz ki:

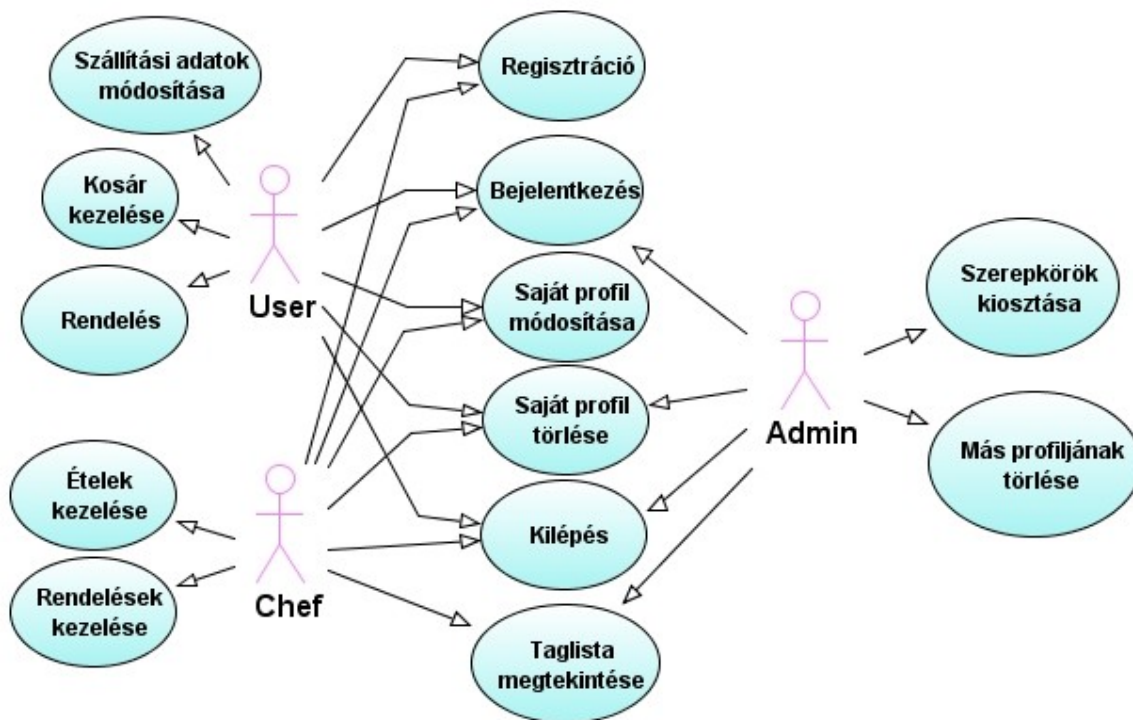


Az ábrán jól látható az alkalmazás 3 rétege azaz

- a felhasználói felület (user interface), melyet a felhasználók a böngésző (browser) segítségével érnek el,
- az alkalmazás szerver (application server), mely az egész rendszer motorja, összekapcsolja az adatbázist a felhasználóval,
- és maga az adatbázis (database), melyben a rendszerhez tartozó konkrét adatokat tároljuk.

3.2 A Use Case diagram

A használati eset diagram (Use Case diagram) már nem strukturális, hanem egy viselkedési diagram. A rendszerünk funkcionalitását lehet vele modellezni egyszerű és könnyen értelmezhető ábrák segítségével. A már bemutatott módszerek mellett a Use Case diagramot is a követelmények megfogalmazására valamint pontosítására használják. Leírja a rendszer, és az azt felhasználó külső szereplők közötti akciók, és reakciók sorozatát. A külső szereplőket aktoroknak nevezzük. Ők nem tartoznak bele a rendszerbe, csak a kapcsolatokat jelölik. Az általam készített rendszerhez a következő Use Case diagram tartozik:




A pálcika emberek jelölik az aktorokat, jelen esetben háromféle felhasználónk lehet (user, chef és admin). Azért kell elkülönítenünk őket mert a három csoport más-más funkciókat vehet igénybe a rendszerünkben.


Maguk a funkciók (használati esetek) az oválisokban vannak melyeket nyilak kötnek össze azzal az aktorral aki igénybe veheti az adott funkciót. Egyes funkciókat több aktor is igénybe vehet, például az ábrán jól látható, hogy a „Bejelentkezést” az összes felhasználó igénybe veheti jogosultságaitól függetlenül. Viszont vannak speciális szolgáltatások amit jogosultsághoz kötünk például „Más profiljának törlése” kizárólag az admin számára engedélyezett.


A Use Case diagramot előszeretettel használják a szoftverfejlesztés kezdeti fázisában egyszerűsége és gyorsan elkészíthetősége miatt.


3.3 Az aktivitás diagram


Az aktivitás diagram (Activity diagram) szintén egy viselkedési diagram. Segítségével modellezhetjük a rendszer dinamikáját. Leírja, hogy futás közben milyen folyamatok zajlanak a rendszerben és megadhatjuk vele időrendben, hogy milyen lehetséges „útvonalakon” mehet végbe egy-egy folyamat. A részletessége változó lehet, attól függően mennyire akarjuk bonyolítani az ábránkat. A következő jelöléseket használhatjuk benne:

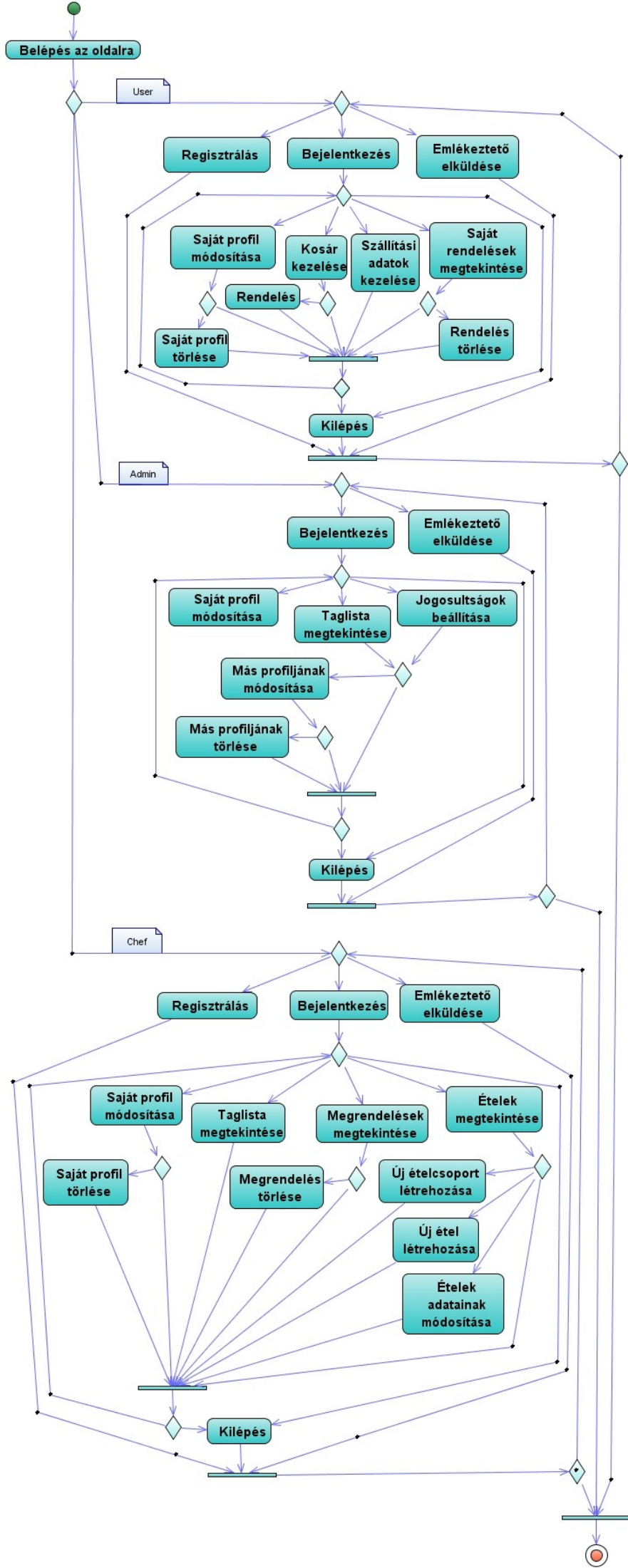
 - Az aktivitás diagram mindig egy belépési ponttal kezdődik, amelyet egy kitöltött zöld körlap jelképez.

 - Lekerekített téglalap jelöli az aktivitásokat, amelyeket irányított nyilakkal kötünk össze. Ezek a nyilak átmenetet képeznek az egyik aktivitásból a másikba. Így biztosítjuk a folyamatosságot, az egymásutániségot vagyis az időbeliséget.

 - A választási lehetőségeket vagyis az elágaztatásokat, egy rombusz segítségével adhatjuk meg. Ez az elágaztatás nyilván valamilyen feltétel alapján fog megtörténni. Az ágak páronként diszjunktak kell hogy legyenek.

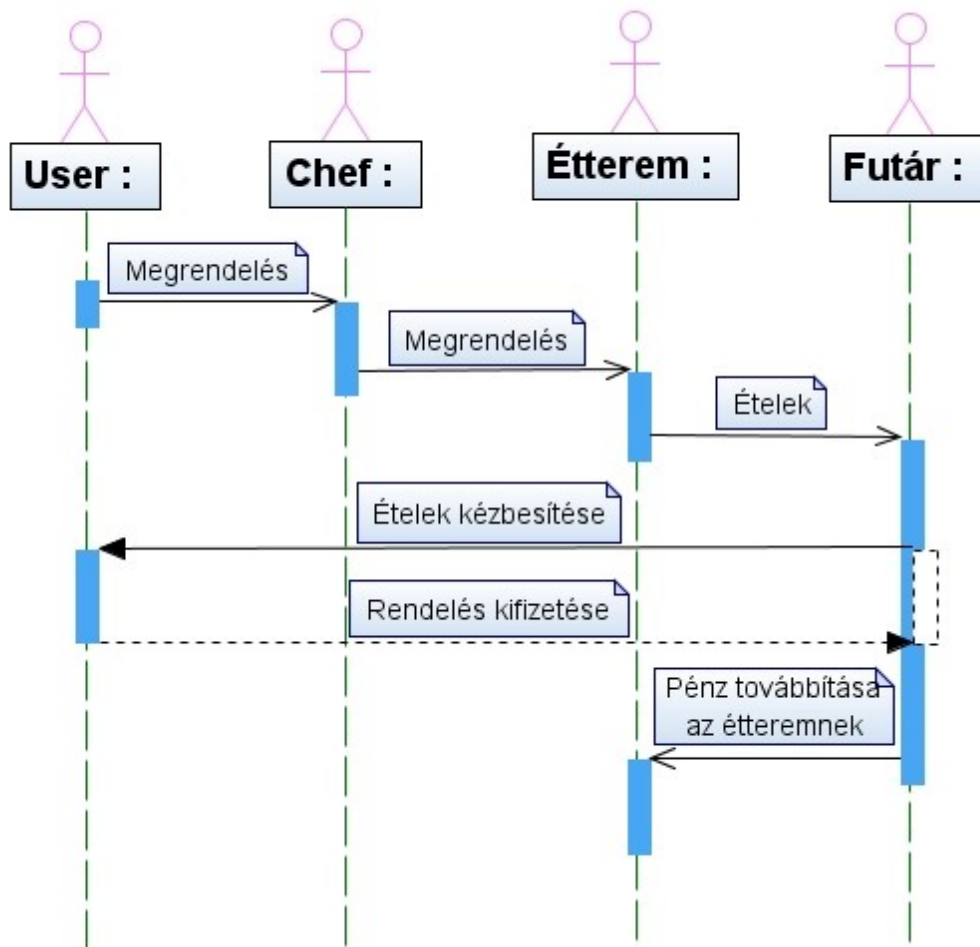
 - Használhatunk megjegyzéseket is, behajtott fülű téglalapok ezek, melyekbe beírhatjuk megjegyzéseinket.

 - A kilépési pontot egy ilyen piros körlap jelzi. A modellezés során ez a pont legyen minden folyamat utolsó lépése és ne legyenek zsákutcák a modellben.



3.4 A szekvencia diagram

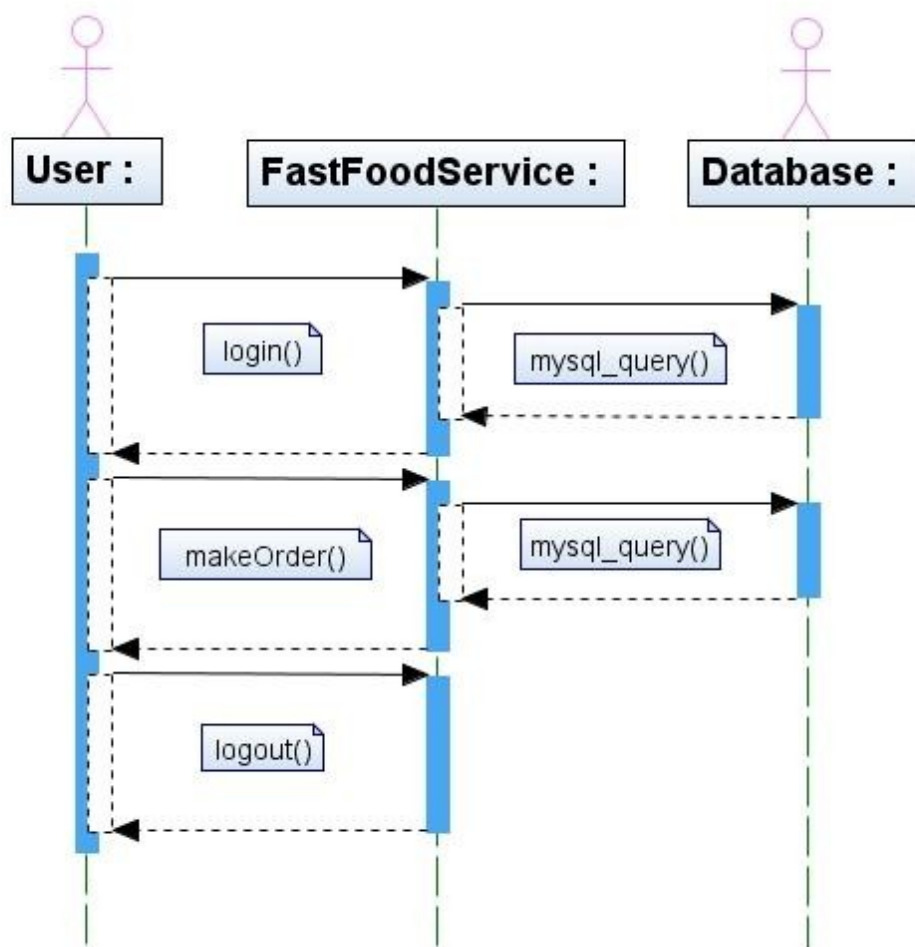
Szintén egy viselkedési diagramról van szó, azon belül is az interakciós diagram egyik fajtája a szekvencia diagram (Sequence diagram). Az interakciós diagramok mutatják meg a rendszerelemek közötti kommunikációt. A szekvencia diagramnál is erről van szó, tehát felvázolja, hogy a rendszerben résztvevő különböző objektumok közötti interakció hogyan valósul meg.



A FastFoodService esetében a fenti ábra egy rendelést mutat be egy szekvencia diagram segítségével. Fent láthatjuk a négy objektumot a User, Chef, Étterem és a Futár személyében. A fenti téglalapban mindig az objektum nevét és a típusát adjuk meg amit kettőspont választ el. Az alattuk levő függőleges szaggatott vonal az életvonaluk. Az életvonalak közötti nyilak a köztük levő kommunikációt jelképezi. Az üres nyíl az asszinkron kommunikációt jelképezi

ami azt takarja, hogy a küldő objektum nem foglalkozik a válasszal csak elküldi az üzenetet. A teli nyíl a szinkron kommunikációt takarja, vagyis a küldő objektum miután elküldte az üzenetet tétlenül várakozik amíg nem kap választ és a vezérlés vissza nem kerül hozzá. Ezt a válaszüzenetet szaggatott nyíllal jelöljük a szekvencia diagramban.

A fenti ábra egy kicsit sematikus, ezért szeretnék bemutatni egy kicsit konkrétabb példát is a szekvencia diagramra. A következő ábra szintén egy megrendelést mutat be, de nincsenek benne a külső objektumok (étterem, futár), kizárólag az alkalmazás által használt elemeket építettem bele.

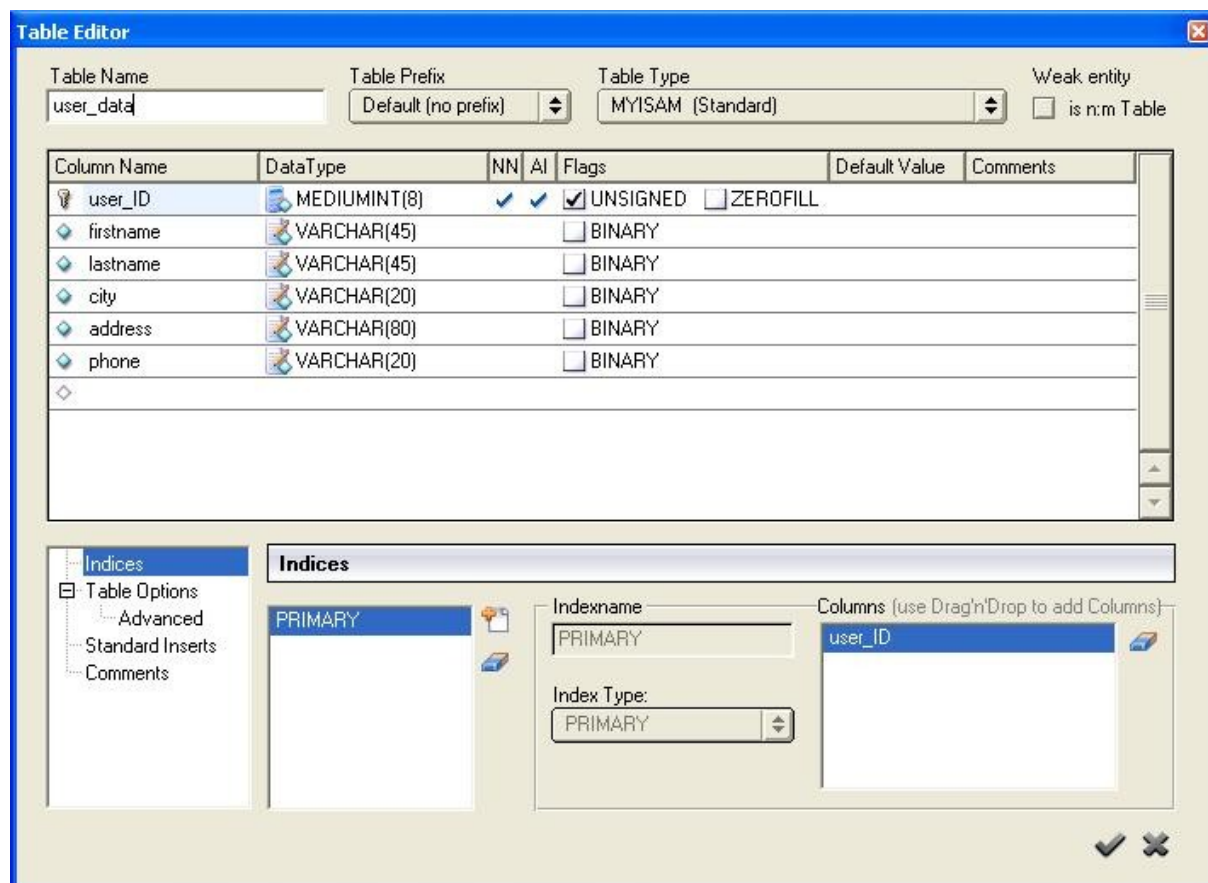


Az ábrán jól látszik, hogy a user jogosultságokkal rendelkező felhasználó először bejelentkezik a rendszerbe a `login()` függvény segítségével, ami kiolvassa az adatbázisból a bejelentkezéshez szükséges információkat majd visszaadja a vezérlést a felhasználónak aki ezek után egy rendelést ad fel melyet a rendszer rögzít az adabázisban. Az első `mysql_query()` egy `SELECT` utasítást takar amellyel kiolvassuk az adatbázisból hogy a felhasználónév és a

jelszó megfelelő-e. A második mysql_query() pedig egy INSERT utasítás, amellyel felvesszük a megrendelt ételeket az adatbázisba. Miután a vezérlés visszakérül, a felhasználó úgy dönt hogy kilép, a logout() függvény meghívódik, ami már nem veszi igénybe az adatbázist.

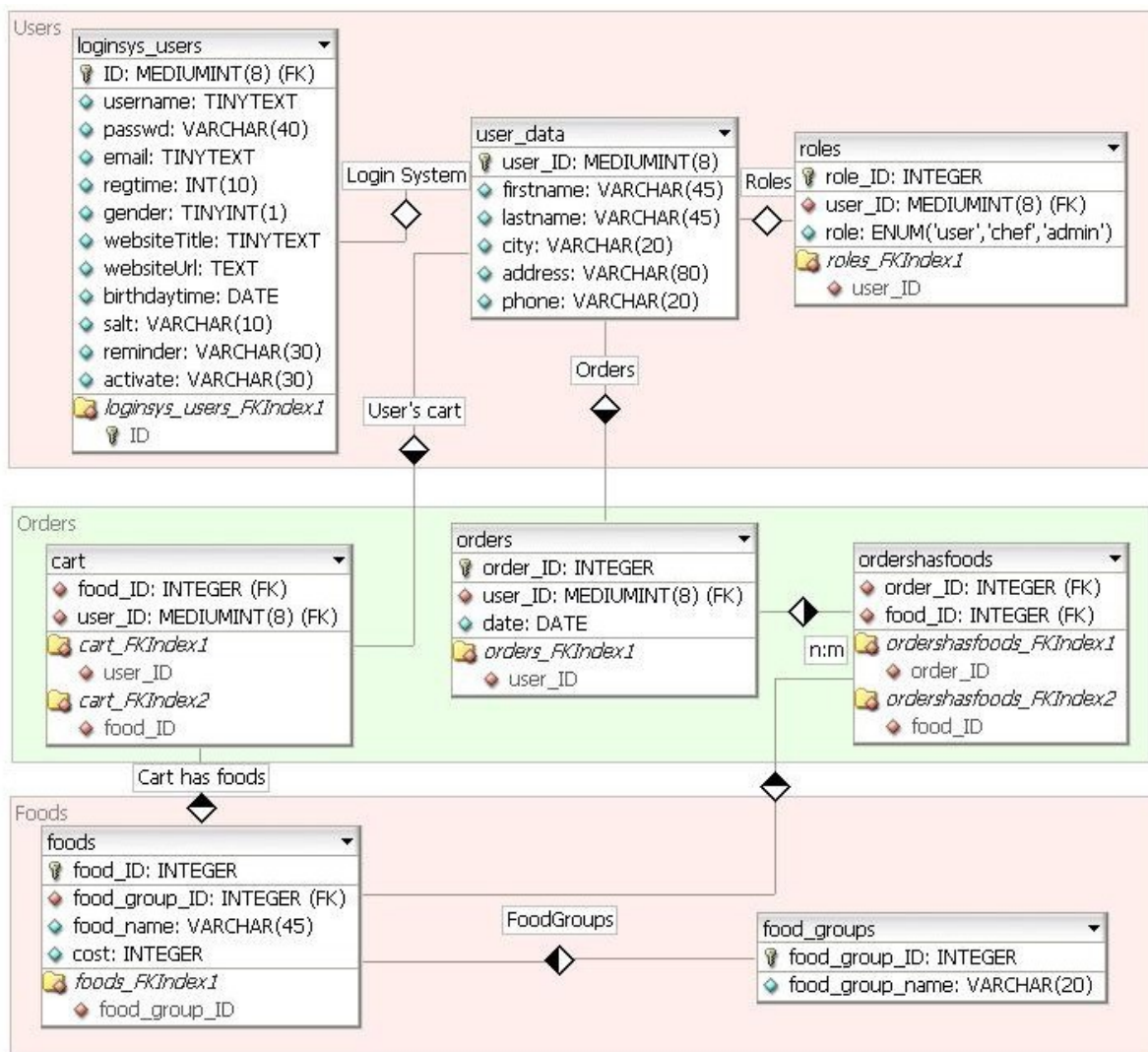
3.5 Az adatbázis tervezés

Miután a követelmények világossá válnak, elkezdhetjük megtervezni az adatbázisunkat. Én a DBDesigner 4 ingyenes adatbázistervező programot használtam a FastFoodService tábláinak megtervezésére. Kezelése egyszerű (drag and drop), közvetlenül tud kapcsolódni a MySQL szerverekhez valamint egy gombnyomásra képes az adatbázistervből .sql fájlt generálni. A rendszeremhez a következőképp hoztam létre a táblákat:



Jól látható, hogy az adatbázistábla nevét (Table Name), a mezőit (Column Name), azok típusát (Data Type) és egyéb fontos paramétereket milyen egyszerűen be lehet állítani itt.

Miután létrehoztuk a tábláinkat, be kell állítani a relációkat, függéseket. A FastFoodService táblái a következőképp kapcsolódnak egymáshoz:



Ezek után legeneráltathatjuk az .sql fájlt ami egy olyan szöveges állomány, amely SQL utasításokat tartalmaz és a PHPMyAdmin segítségével beimportálva létrejönnek a táblák a MySQL szerverünkön. Részlet ebből az állományból:

```
--
-- Tábla szerkezet: `loginsys_users`
--

CREATE TABLE `loginsys_users` (
  `ID` mediumint(8) unsigned NOT NULL auto_increment,
  `username` tinytext collate utf8_unicode_ci NOT NULL,
  `passwd` varchar(40) collate utf8_unicode_ci NOT NULL default '',
  `email` tinytext collate utf8_unicode_ci NOT NULL,
  `regtime` int(10) unsigned NOT NULL default '0',
  `gender` tinyint(1) NOT NULL default '0',
  `websiteTitle` tinytext collate utf8_unicode_ci NOT NULL,
```

```

`websiteUrl` text collate utf8_unicode_ci NOT NULL,
`birthdaytime` date NOT NULL default '0000-00-00',
`salt` varchar(10) collate utf8_unicode_ci default NULL,
`reminder` varchar(30) collate utf8_unicode_ci default NULL,
`activate` varchar(30) collate utf8_unicode_ci default '0',
PRIMARY KEY (`ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
  AUTO_INCREMENT=0 ;

-----
--
-- Tábla szerkezet: `user_data`
--

CREATE TABLE `user_data` (
  `user_ID` mediumint(8) unsigned NOT NULL auto_increment,
  `firstname` varchar(45) collate utf8_unicode_ci default NULL,
  `lastname` varchar(45) collate utf8_unicode_ci default NULL,
  `city` varchar(20) collate utf8_unicode_ci default NULL,
  `address` varchar(80) collate utf8_unicode_ci default NULL,
  `phone` varchar(20) collate utf8_unicode_ci default NULL,
  PRIMARY KEY (`user_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
  AUTO_INCREMENT=0 ;

-----
--
-- Tábla szerkezet: `roles`
--

CREATE TABLE `roles` (
  `role_ID` int(10) unsigned NOT NULL auto_increment,
  `user_ID` mediumint(8) unsigned NOT NULL default '0',
  `role` enum('user','chef','admin') collate utf8_unicode_ci NOT NULL
    default 'user',
  PRIMARY KEY (`role_ID`),
  KEY `roles_FKIndex1` (`user_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
  AUTO_INCREMENT=0 ;

```

Jól látható hogy tényleg SQL utasításokat tartalmaz az .sql fájlunk. Röviden felvázolnám még, hogy mire is fogjuk használni ezeket a táblákat.

Kezdjük az első táblánkkal a loginsys_user-rel. Az alkalmazosomhoz a PHP Stúdió egyik ingyenes loginrendszerét használtam fel amit jelentősen módosítottam. Jobbnak láttam külön táblába helyezni a beléptető rendszer adatait a rendszer többi adatától. Ez később még hasznos is lehet mert ha le akarjuk cserélni a loginrendszerünket akkor csak ezt az egy táblát kell majd módosítanunk.

A következő tábla a user_data. Ez a tábla tartalmazza a megrendelők szállítási adatait vagyis a nevüket, címüket és telefonszámukat.

A `roles` tábla tartalmazza a felhasználók jogosultságait ami háromféle lehet: `user`, `chef` vagy `admin`. Ezt a három táblát akár össze is lehetett volna vonni mivel a táblák közti relációk 1:1 számosságúak, ahogy azt az üres rombusz is jelzi a fenti ábrán.

A `cart` a kosár táblája. Nagyobb rendszerekben nem szokták tárolni a kosár tartalmát az adatbázisban, hanem `session` változók segítségével oldják meg, de ezek tartalma a böngésző bezárásával elvész, ezért én a saját adatbázisomban letárolom a kosár tartalmát amit bármikor megnézhet a megrendelő.

Az `orders` táblában az egyes felhasználókhöz tartozó megrendelések valamint a megrendelések időpontjai vannak.

Az `ordershasfoods` egy kapcsolótábla, összeköti a rendeléseket az ételek táblájával. Az ábrán is látszik, hogy ez egy n:m számosságú relációt valósít meg.

A `foods` tábla tartalmazza az ételek nevét, árát és külső kulcsként a csoportját, ezzel is csökkentve a redundanciát.

A `food_groups` pedig a már említett ételcsoportok nevét tartalmazza.

Ha mindezzel megvagyunk és jól sikerül a tervezési fázis, akkor nagy valószínűséggel működőképes alkalmazást hozhatunk létre. Attól hogy működik egy webes alkalmazás, még nem biztos, hogy sikeres is lesz ezért gondoljunk mindig az átlag felhasználókra amikor a felhasználói felületet tervezzük. Ne használjunk rikító színeket, ügyeljünk a betűméretre és törekedjünk az egyszerűségekre akkor biztosan sikeres lesz az oldalunk.

4. Implementálás

A következő nagy fázis az implementálás. A követelmények és a tervek alapján elkészítjük a konkrét alkalmazást. Az én rendszerem PHP 4.0-ban íródott. Szerettem volna PHP 5.0-ban megírni de a freeweb.hu csak a 4.0-át támogatja.

4.1 A PHP

A PHP (PHP: Hypertext Preprocessor) egy nyílt forráskódú, számítógépes szkriptnyelv, legfőbb felhasználási területe a dinamikus weboldalak készítése. Emiatt a PHP-t jórészt szerver-oldalon használják, bár létezik parancssori interfésze is, illetve önálló, grafikus felületű alkalmazások is létrehozhatóak vele.

A nyelvet eredetileg Rasmus Lerdorf alkotta meg 1994-ben, de a ma létező egyetlen (és hivatalos specifikáció híján *de facto* szabvánnyá vált). A PHP implementációt már a PHP Group tartja karban és fejleszti. A PHP a legtöbb webszerverre, operációs rendszerre és platformra ingyenesen telepíthető. Manapság több mint 20 millió weboldal és egymillió szerver futtat PHP-t, bár a nyelvet használó oldalak száma 2005 augusztusától kezdve folyamatosan csökken. A PHP emellett az Apache webszerver egyik legnépszerűbb beépülő modulja. A PHP legfrissebb változata az 5.2.9 verziószámú, amely 2009. február 26-án jelent meg.

4.2 A FastFoodService webes alkalmazás implementálása

Nagyobb rendszerekhez gyakran használnak keretrendszereket fejlesztésnél. Ilyen például a Zend keretrendszer amely 2004-ben (PHP 5.0) számos új objektumorientált lehetőséggel bővült. Mivel viszonylag kis rendszerről van szó én nem használtam semmilyen keretrendszert. Az alkalmazásnak az `index.php` a kezdőlapja ami egy keretes szerkezetű HTML oldal. Nem célszerű kereteket használni, elég sok baj van velük, de én mégis kihívásnak éreztem és megpróbálkoztam vele. A következőképp fest tehát az `index.php` lapom:

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>.::FastFoodService:.:</title>
</head>

<frameset rows="*" cols="250,*" framespacing="0" frameborder="no"
border="1">
  <frame src="contents.php" name="leftFrame" noresize="noresize"
  id="leftFrame" title="leftFrame"/>
  <frame src="main.php" name="mainFrame" id="mainFrame" title="mainFrame"/>
</frameset>

<noframes>
  <body>
    <h1> Sajnos a böngészője elavult, nem tudja megjeleníteni a keretes
    weblapot.
    </h1>
  </body>
</noframes>

</html>

```

Jól látható, hogy a keretek (framek) két részre osztják a weblapot egy baloldali `leftFrame` és egy jobb oldali nagyobb `mainFrame` keretre. Az is látszik, hogyha esetleg egy nagyon régi böngészőn próbálnánk megjeleníteni az oldalt, ami nem képes a keretek megjelenítésére, akkor csak az alábbi üzenetet jelenítené meg: Sajnos a böngészője elavult, nem tudja megjeleníteni a keretes weblapot.

A baloldali (`leftFrame`) keretben a menüt helyeztem el amivel navigálhatunk az oldalon. A navigáció során a megjelenített tartalom többnyire a jobboldali (`mainFrame`) keretben jelenik meg. A menüben az ételcsoportok kiírásánál elhelyeztem egy Javascript-et azért hogy „összecsukható” legyen a menü ezáltal is egyszerűbb és szebb lesz. A Javascript-eket a `<script type="text/javascript"> ... </script>` tag-ek közé rakva illeszthetjük be a HTML kódba. Rengeteg ilyen ingyenesen használható Javascript lelhető fel az interneten, ezek közül választottam ki én is ezt. Sajnos a Javascript-ek nem biztos, hogy egyformán jelennek meg minden böngészőn. A `contents.php` ami a bal keretbe töltődik be tulajdonképpen csak egy `<?php include_once("menu.php"); ?>` sort tartalmaz a Javascript-en kívül. Ezzel a sorral betöltődik a `menu.php`, ami az oldal „indulómotorjának” számít. Ez fogja betölteni login ablakot, ha még nincs bejelentkezve a felhasználó, valamint a jogosultságának megfelelően a számára elérhető tartalmakat.

Például figyeljük meg a következő kódrészletet a menu.php fájlból:

```
// Jogok
if($role == "admin")
{
print('<tr>'
.'<td>'
.'<a href="' . $mainurl . follow . 'roles=show"
target="mainFrame" style="text-decoration:none;">
&bull;Jogosults&aacute;gok</a><br>'
.'</td>'
.'</tr>');
}

if($role == "chef")
{
//Rendelések
print('<tr>'
.'<td>'
.'<a href="' . $mainurl . follow . 'orderlist=show"
target="mainFrame" style="text-decoration:none;">
&bull;Rendel&eacute;sek</a><br>'
.'</td>'
.'</tr>');
```

A `$role` változó értéke az aktuális felhasználó szerepköre (user, chef vagy admin). Látható, hogy a szerepkörtől függően lesznek kiírva a menüpontok. Például csak az admin számára jelenik meg a Jogosultságok menüpont valamint csak a séfek láthatják az összes rendelést.

Most vizsgáljuk meg a jobb oldali keretet. A baloldali keretről tehát tudjuk hogy linkeket tartalmaz, melynek célja (target) a jobb oldali keret. Így ha rákattintunk egy ilyen menüpontra akkor a vezérlés átadódik a mainFrame keretnek amibe a main.php töltődik be, melyben a következő fontos kódrészlet szerepel:

```
<?php
if(isset($_GET["memberslist"])) include_once("members.php");
elseif(isset($_GET["profileset"]) || isset($_GET["profile"]) ||
isset($_GET["delete_profile"])) include_once("modify.php");

elseif(isset($_GET["roles"])) include_once("special/roles.php");

elseif(isset($_GET["chef_submit"]) || isset($_GET["chef_check"]) ||
isset($_GET["chef_submit"])) include_once("special/foods.php");
elseif(isset($_GET["orderlist"]) || isset($_GET["delete_order"]))
include_once("special/orders.php");

elseif(isset($_GET["user_data"]) || isset($_POST["update_submit"]))
include_once("special/userdata.php");
elseif(isset($_GET["user_order"])) include("special/userorder.php");

elseif(isset($_GET["target"])) include_once("profile.php");
```

```
elseif(isset($_GET["reg"]) || isset($_GET["activate"]))
    include_once("registration.php");
elseif(isset($_GET["remind"])) include_once("reminder.php");
else ...
```

Gyakorlatilag a fent látható kódrészlet az egész alkalmazás vezérlője. Attól függően, hogy milyen paramétereket küldünk a main.php-nak úgy tölti be a megfelelő PHP oldalt a főkeretbe. Ha tehát a korábbi kódban rákattintunk a Rendelések menüpontra, akkor \$_GET paraméterben átküldi a leftFrame a mainFrame-nek az orderlist változót mégpedig show értékkel. Ezáltal a fenti kódban az aláhúzott részhez kerül a vezérlés ami befogja töltetni a főkeretbe az orders.php-t és megjelennek a rendelések:

Megrendelő neve	Rendelés száma	Szállítási dátum
Trab Antal	1	2009-05-31 19:15:00
Will Smith	4	2009-05-15 02:19:51
Will Smith	17	2009-05-13 18:15:00
Will Smith	18	2009-05-14 19:15:00
Will Smith	19	2009-05-13 19:20:00
Will Smith	20	2009-04-08 19:20:00
Will Smith	25	2009-05-01 20:05:00
Will Smith	28	2009-05-24 10:30:00
Will Smith	31	2009-05-14 17:05:00
Will Smith	34	2009-05-21 18:55:00

1 | 2 | >

<http://www.freeweb.hu/fastfoodservice/main.php?orderlist=show>

Tehát már csak ezeket a részeket kell megírunk a funkcióiknak megfelelően és kész vagyunk.

5. Összefoglalás

A dolgozat elején említettem, hogy igyekszem a legfontosabb lépéseket bemutatni, amely egy webes alkalmazásfejlesztés során előfordul és azt hiszem ez többé kevésbé sikerült is. Egy valódi webes alkalmazás fejlesztését nem egyetlen ember szokta végezni hanem általában team-ekben (csapatokban) dolgoznak. Ezért is nagyon sok minden kimaradt a dolgozathoz mint például az implementáció utáni tesztelés dokumentálása, vagy az üzembe helyezés. Nagyon sok tapasztalattal gazdagodtam a dolgozat megírása során, talákoztam számos nehézséggel amelyek leküzdése nem kis feladat volt, de végül sikerült.

Azt is említettem korábban, hogy a prototípus alapú inkrementális rendszerfejlesztési modell alapján fogok dolgozni, ezt úgy oldottam meg, hogy készítettem egy működő prototípust, ez a fejlesztésnek azon fázisában volt, ahol még csak az admin vehette igénybe a funkcionalitásokat. Később ezt a prototípust bővítettem inkremensekkel vagyis hozzáadtam a chef funkcionalitásait majd pedig a user funkcionalitásait.

Ha újra kellene írnom ezt az alkalmazást biztosan sok mindent máshogy csinálnék, például nem használnék frame-eket és megpróbálnám felhasználni a Zend keretrendszert, de alapjában véve úgy érzem, hogy jól sikerült a webes alkalmazásom és szeretnék a jövőben is az informatika ezen ágával foglalkozni, a programtervező informatikus szak elvégzése után is.

5.1 Köszönetnyilvánítás

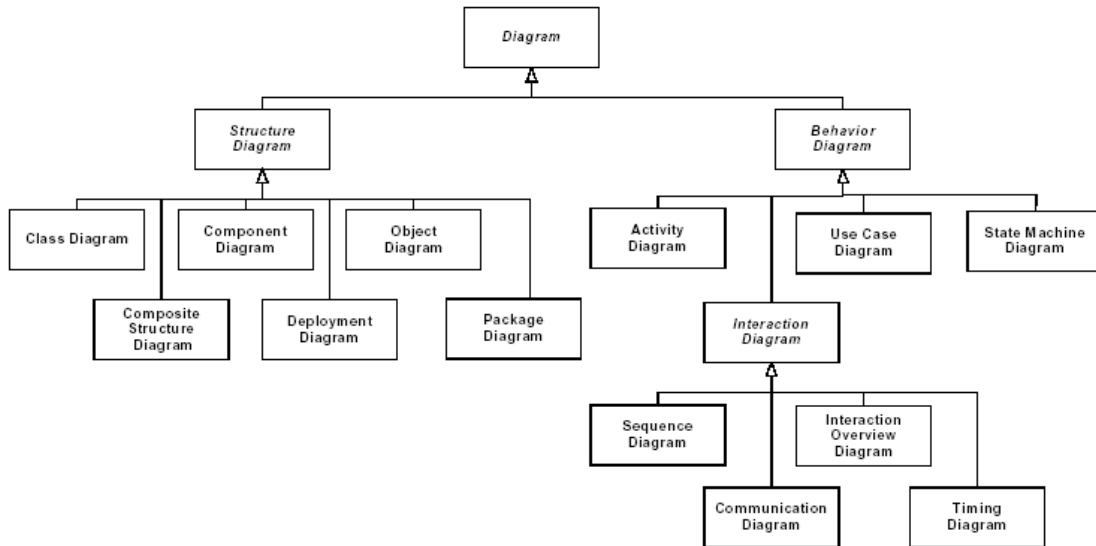
Ezúton szeretnék köszönetet mondani családomnak, barátaimnak a lelki támaszért, valamint tanárainknak odaadó munkájukért hiszen nélkülük ez a dolgozat nem készülhetett volna el.

6. Irodalomjegyzék

- Rendszerfejlesztés technológiája előadás jegyzet, 2008/2009.
- Dr. Rutkovszky Edéné: Projektmenedzsment, mobiDIÁK könyvtár, DE IK, 2005.
- Debolt, Virginia: HTML és CSS, Kiskapu kiadó, 2005.
- <http://phpstudio.hu/>
- <http://www.hotscripts.com/>
- <http://www.google.hu>
- http://hu.wikipedia.org/wiki/Unified_Modeling_Language
- <http://hu.wikipedia.org/wiki/PHP>
- <http://freeweb.hu/>
- <http://fastfoodservice.fw.hu/>

7. Függelék

7.1 Az UML 2.0 alcsoportjainak hierarchikus szerkezete



7.2 Néhány kép a FastFoodService alkalmazásról



FastFoodService

- Profilmódosítás
- Taglista
- Jogosultságok
- Kijelentkezés

Felhasználónév	Jogosultság	Módosítás
bandesz85	chef	chef
csaky	chef	chef
ffsadmin	admin	Fix
jocka	user	user
kaksi	user	user
kissepi	chef	chef
ktmboy	chef	chef
martinus	user	user
savika	user	user
user1	chef	chef
user2	user	user
user3	user	user
xeno	user	user
Yankee5	user	user
zava	user	user

FastFoodService

- Profilmódosítás
- Rendelési adatok
- Rendelésem
- Étlap
- Kijelentkezés

Rendelés száma: 28

Megrendelt ételek	Dátum
Madántej 1 adag: 450 Ft	2009-05-24 10:30:00
Mézes puszedli 1 adag: 250 Ft	
Palacsinta 1 adag: 400 Ft	
Piskóta tekercs 1 adag: 460 Ft	
Fizetendő: 1560 Ft	Megrendelés törlése

Rendelés száma: 31

Megrendelt ételek	Dátum
Gyümölcsleves 1 adag: 400 Ft	2009-05-14 17:05:00
Húsleves 1 adag: 400 Ft	
Fizetendő: 800 Ft	Megrendelés törlése

< | 1 | 2 | 3 | >

FastFoodService

- Profilmódosítás
- Rendelési adatok
- Rendelésem
- Étlap
- Kijelentkezés

Kosár

Burgonyapüré

Tavaszi tál

Megrendel

Megrendelt ételek	Dátum
Burgonyapüré 1 adag: 330 Ft	2009-06-05 13:15 <input type="button" value="Megrendel"/>
Tavaszi tál 1 adag: 800 Ft	
Fizetendő: 1130 Ft	

Június 2009

H	K	Sz	Cs	P	Sz	V
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

13 : 15

- Profilmódosítás
- Rendelési adatok
- Rendeléseim
- Étlap
- Kijelentkezés

Will Smith szállítási adatai

Vezetéknév:

Keresztnév:

Város:

Cím:

Tel. szám:

http://www.freeweb.hu/fastfoodservice/main.php?user_data=show

- Profilmódosítás
- Taglista
- Rendelések
- Ételek
- Kijelentkezés

Felhasználónév ▼	Regisztráció időpontja ▼	E-mail	Weblap
kissepi	2009. 05. 08. 22:29:30		
ktmboy	2009. 05. 06. 23:21:49		
martinus	2009. 05. 06. 20:29:54		
savika	2009. 05. 05. 17:27:40		
user1	2009. 04. 30. 13:55:55		

< | 1 | 2 | 3 | >

- Profilmódosítás
- Taglista
- Rendelések
- Ételek
 - Főételek
 - Levesek
 - Gyorsételek
 - Desszertek
 - Köreték
 - Előételek
 - K7mFoods™
 - Főzelékek
 - Új étel
 - Új csoport
- Kijelentkezés

Desszertek

Étel neve ▼	Csoportja ▼	Ára ▼		
<input type="text" value="Gofri"/>	Desszertek ▼	<input type="text" value="250"/>		
<input type="text" value="Gyümölcskenyér"/>	Desszertek ▼	<input type="text" value="400"/>		
<input type="text" value="Madártej"/>	Desszertek ▼	<input type="text" value="450"/>		
<input type="text" value="Mézes puszedli"/>	Desszertek ▼	<input type="text" value="250"/>		
<input type="text" value="Palacsinta"/>	Desszertek ▼	<input type="text" value="400"/>		
<input type="text" value="Piskota rolád"/>	Desszertek ▼	<input type="text" value="500"/>		
<input type="text" value="Piskota tekercs"/>	Desszertek ▼	<input type="text" value="460"/>		
<input type="text" value="Pozsonyi kifli"/>	Desszertek ▼	<input type="text" value="350"/>		
<input type="text" value="Somlói galuska"/>	Desszertek ▼	<input type="text" value="500"/>		
<input type="text" value="Tiramisu"/>	Desszertek ▼	<input type="text" value="450"/>		

Csoport törlése >>

1 | 2 | >