

Tartalomjegyzék

1. Funkcionális terv	2
1.1. Feladat leírása	2
1.2. Rendszer célja, motivációja	3
1.3. Szereplők és igényeik	3
1.3.1. Valódi felhasználók	3
1.3.2. Hirdetők	4
1.3.3. Rendszer adminisztrátor	4
1.4. Komponensek leírása	5
1.4.1. Kliens oldali komponensek listája	5
1.4.2. Szerver oldali komponensek listája	6
1.5. Környezet	8
1.6. Rendszert leíró blokkvázlatok, interfészek	9
1.6.1. Teljes rendszer	9
1.6.2. Kliens oldal	12
1.6.3. Szerver oldal	14
2. Tesztelési terv	18
2.1. Komponens tesztek	18
2.1.1. Kliens oldal	18
2.1.2. Szerver oldal	18
2.2. Globális tesztelés	18
3. Üzemeltetési terv	19
3.1. Monitorozás	19
3.2. Karbantartás	21
4. Adatszerkezeti terv	23
4.1. Adatbázis terv	23
4.1.1. Entitások táblái	25
4.1.2. Kódtábla	26
4.1.3. Kapcsolatok táblái	26
4.2. Naplófájlok	27
5. Megvalósítási terv	28
5.1. Objektumkatalógus	28
5.1.1. Kliens oldal	28
5.1.2. Szerver oldal	29
5.2. Statikus osztálydiagramok	30
5.3. Kliens állapotgép	31
5.4. Szerver szekvenciadiagramok	33

1. Funkcionális terv

1.1. Feladat leírása

A feladat egy navigációs alkalmazás fejlesztése GPS-képes eszközökre. A program a szerveren tárolt adatbázisból a felhasználó által küldött kérés alapján nyújt térképi és egyéb információt a felhasználók felé.

A felhasználó GPS vevővel és vezeték nélküli internetkapcsolattal rendelkező eszközzel kapcsolódik a szerverhez, ez lehet mobiltelefon, PDA, PNA, laptop, stb. A felhasználó elküldi GPS-en keresztül megszerzett pozícióját a szervernek, mely egy a pozíciónak megfelelő és kért felbontású / nagyítású térképrészletet küld vissza.

A rendszer többfelhasználós, vagyis nem csak egy felhasználó kommunikál a szerverrel, hanem egyszerre akár több is. A felhasználók csoportokat alakíthatnak, és az egy csoporton belüli felhasználók látják egymás pozícióját a kapott térképen. A térképeken lehetőség van hirdetések / célok elhelyezésére, mint például éttermek vagy hivatalok, a hirdetéseket hirdető adhatja fel, míg célokat a térkép fejlesztői adhatnak a térképhez.

Későbbi fejlesztői lehetőség, hogy nem csak a térképen elfoglalt pozíciót adja vissza az alkalmazás, hanem navigációs szolgáltatást is nyújt, vagyis legrövidebb / leggyorsabb útvonalat kalkulál a kijelölt célpont felé. Ez egyelőre nem része a projektnek.

A rendszer létjogosultságát az adja, hogy ebben az elrendezésben nem szükséges a mobil eszközben nagy kapacitású adattároló eszköz, mert a térképet és minden hozzá kapcsolódó információt a központi szerver tárol. Egyszerűsödik a térkép frissítése is, mivel nem szükséges minden egyes mobil eszközön ezt elvégezni, elegendő csak a központi szerveren frissíteni.

További lehetséges előny egy fizetős rendszert feltételezve, hogy a felhasználóknak nem szükséges például egy komplett Kelet-Európa-térképpel rendelkeznie, ha ő csak Magyarországon szeretné használni az eszközét. Ebben az esetben könnyedén megoldható, hogy ez a felhasználó csak a magyarországi adatokhoz férhessen hozzá, természetesen egy alacsonyabb árért. Ezek az előnyök egyelőre csak lehetséges fejlesztési irányok, egyelőre szintén nem részei a projektnek.

Természetesen hátrány is származik abból, hogy a térkép nem a mobil eszközön tárolódik, mivel szükség van vezeték nélküli internet kapcsolatra a szerver felé. Ennek a kapcsolatnak elég nagy sávszélességűnek kell lennie ahhoz, hogy a szerver által küldött adatokat megfelelő gyorsasággal tudja a mobil eszköz megjeleníteni. Természetesen a technológiai fejlődésnek köszönhetően a mai szélessávú vezeték nélküli kapcsolatok már alkalmasak ezen feladat ellátására. Ehhez szükséges az is, hogy az alkalmazás ne pazarolja ezt az erőforrást, praktikust megoldást kell találni az adatok átvitelének formájára. Egy alternatív megoldásként felmerül, hogy a mobil eszköz működhessen offline módban is, vagyis működő internetkapcsolat esetén folyamatosan frissül a térkép az eszközön, annak hiányában viszont az utolsó frissítést használja. Ez az opció egyelőre nem része a projektnek.

Megszorításokat jelent a mobil eszközön való használat is, mivel korlátozottak az eszköz lehetőségei, mint például a számítási sebesség, megjelenítési képesség. Éppen ezért a szoftvert úgy kell megalkotni, hogy igazodjon ezekhez a korlátokhoz, ne támasszon túlzottan nagy minimális futtatási igényeket, ne pazarolja a szűkös erőforrásokat.

1.2. Rendszer célja, motivációja

A feladat egy valós alkalmazás fejlesztése, de egyben önálló labor feladat és a diplomamunka alapját képezi. Ennek megfelelően nem csak azok a követelmények vonatkoznak rá, mint egy a való életnek készülő projekt esetében, hanem figyelembe kell venni, hogy egy diák több féléves munkáját is értékelni kell a projekten keresztül. Emiatt a feladat nem lehet olyan komplikált és mindenre kiterjedő, mint egy valódi fejlesztés. Fontos, hogy ki tudjam emelni azokat a sarokpontjait a projektnek, aminek a kivitelezése nem haladja meg a lehetőségeimet, de mégis komplett alkalmazás áll össze belőlük.

A projekt fejlesztése során kiemelt figyelmet fordítok majd a rendszer alapjainak minél részletesebb specifikálására, az adatmodellre – különösen a térinformatikai vonatkozásban – illetve a fejlesztéshez különösen fontos viselkedési modell és tesztelési terv megalkotására. A fent felsoroltak mellett kisebb súllyal foglalkozok majd például a rendszer integrálhatóságával, az emberi felhasználásból fakadó problémákkal, vagy az üzemeltetés kérdéseivel.

1.3. Szereplők és igényeik

A rendszer működése során háromféle szereplő fog megjelenni. A legfontosabb az a felhasználó, aki a mobil eszközén keresztül futtatja az alkalmazást. A legtöbb igény az ő esetében merül fel. Ezeket a felhasználókat nevezem *valódi felhasználóknak*.

Szintén lényeges a *hirdető*, aki nem használja a szoftver kliens oldali változatát, csak a szerverhez fér hozzá, és ide tölti fel a hirdetéseit a megfelelő pozícióadatokkal. Feléje kevesebb megkötéssel és igénnyel lehet számolni.

A harmadik szereplő a rendszer *adminisztrátora*, aki karbantartási és egyéb munkákat végez.

1.3.1. Valódi felhasználók

A valódi felhasználó a mobil eszközén keresztül használja az alkalmazást. Előfizet a szolgáltatásra, mely számára az alábbiakat nyújtja:

- regisztráció, biztonságos bejelentkezés
- tartózkodási helytől és az előre beállított méretaránytól függő térkép a mobil eszközre
- navigálás a térképen – elforgatás, eltolás
- a térképen megjelennek POI-k (Point Of Interest), mint például hirdetésben szereplő hely illetve közérdekű intézmény
- lehetőség a felhasználónak, hogy csoportot alakítson, csoportba lépjen, és a csoport tagjainak helyzetét a térképen megjelenítse

Egy felhasználó számára a legfontosabb, hogy a szoftver használata számára hasznos és élvezhető legyen. Mivel mobil eszközön történik a használat, ezért lényeges, hogy használata

ne térjen el jelentősen az egyéb, mobil eszközökre fejlesztett alkalmazásoktól. Ennek megfelelően az alapvető igények:

- egyszerű használatbavétel – beszerzés, telepítés, bejelentkezés
- kis méret
- gyorsaság – főleg használat közben, indításnál kevésbé
- könnyen elsajátítható, egyszerű és logikus kezelőfelület – ide tartozik a menürendszer, billentyűkombinációk, színek, formák
- áttekinthető képernyőkép – képek és szöveg aránya, színek
- biztonság – jogosultságkezelés, személyes adatok védelme

A mobil eszközön való használat további problémákat is felvet:

- kis sáv szélesség a világháló felé – vezetékes kapcsolathoz képest, tehát az adatforgalmat minimalizálni kell
- korlátozott számítási kapacitás – ezért minél kevesebb számítást kell a felhasználó eszközén elvégezni
- korlátozott megjelenítési képesség – információ, menürendszer megfelelő elosztása

1.3.2. Hirdetők

A hirdetők számára a szoftver használata, mint befektetési lehetőség jelenik meg. Ennek megfelelően számukra nem lényeges a megjelenés, viszont fontos, hogy egyszerűen és gyorsan lehessen használni. Ebben az esetben is fontos, hogy a kezelőfelület áttekinthető legyen, könnyű legyen a navigáció. A hirdetők számára az az ideális, ha nem kell telepíteni és megvásárolni semmit, így a kézenfekvő megoldás egy publikus weblap, amin keresztül a hirdetés feladható, mivel ehhez mindössze egy web böngésző szükséges.

Tehát a hirdetői igények listája:

- regisztráció, biztonságos belépés
- hirdetés szövegének és pozíciójának megadása – lehetőleg minél egyszerűbb menürendszerben
- adatvédelem – hirdetéseket más ne módosíthassa
- korábbi hirdetések listázása, módosítása, törlése
- hirdetések látogatási statisztikájának megtekintése (opcionális)

1.3.3. Rendszer adminisztrátor

A rendszer adminisztrációjának elvégzéséhez szintén nincsen szükség komoly felhasználói felületre, sokkal fontosabb a funkcionalitás. A legfontosabb igények:

- biztonságos azonosítás – illetéktelen személy ne tudjon adminisztrátorként hozzányúlni a rendszerhez

- egyszerű kezelőfelület, logikus felépítés
- naplófájlok gyors lekérdezése
- lekérdezések összeállításának megkönnyítése – statisztikák készítéséhez
- előre elkészített lekérdezési sémák
- lekérdezések eredményének jól áttekinthető megjelenítése
- széleskörű módosítási lehetőségek (törlés, létrehozás manuálisan)

A rendszer adminisztrátoránál foglalkozni kell azzal a kérdéssel is, hogy bizonyos követelményeknek meg kell felelnie annak, aki ezt a posztot betölti. Míg más felhasználók esetén semmilyen követelmény nincs, a rendszer adminisztrátorának teljesítenie kell az alábbiakat:

- adatbázis szerkezetének ismerete – lekérdezések manuális összeállításához
- adatbázis-kezelő működésének ismerete – statisztikák, riportok készítéséhez
- SQL lekérdezőnyelv ismerete – lekérdezések manuális összeállításához

1.4. *Komponensek leírása*

1.4.1. **Kliens oldali komponensek listája**

1.4.1.a. Kontroller

A komponens feladata a kliens oldali működés összefogása. Kezeli a felhasználói interakciókat, és azoknak megfelelően kapcsolatba lép a szerverrel és irányítja a megjelenítő és a GPS modult.

Feladatok:

- kliens oldali program fő szála, többi modul irányítása
- felhasználói interakciók (egérmozgatás, egérekattintás, billentyűleütés, érintőképernyő-érintés) kezelése és a megfelelő reakció (képernyőfrissítés, feladat indítása) végrehajtása
- kapcsolattartás a szerverrel
 - kérés a kapcsolatfelvételre
 - elküldendő adatok összeállítása és továbbítása
 - válaszként érkező adatok feldolgozása
 - lehetőség több kérés-válasz periódusra egy kapcsolatfelvételen belül (vagyis a kapcsolat állapotát is tárolni kell)
 - kapcsolat korrekt lezárása (sávszélesség-pazarlás elkerülésére)
- GPS modul kezelése
 - koordináták lekérése
 - kapott adatok feldolgozása
- megjelenítő modul kezelése
 - képernyőrajzolás kérés küldése

1.4.1.b. Megjelenítő

A komponens feladata a szervertől kapott adatok és a kliens által generált információk megjelenítése a mobil eszköz képernyőjén.

Fontos, hogy a modul jól paraméterezhető legyen az eszközönként nagyon eltérő megjelenítő képességek miatt.

Feladatok:

- kontroller modultól kapott adatok (térkép és egyéb információk) alapján a képernyőkép rajzolása
- képernyőkép frissítés
- alkalmazkodás (vagy paraméterezési lehetőség) a különböző megjelenítőkhöz

1.4.1.c. GPS

A komponens feladata a GPS vevővel való kapcsolattartás és a koordinátaadatok lekérése, majd továbbítása a kontroller felé.

Feladatok:

- kontrollertől érkezett kérés hatására GPS koordináták kérése a megfelelő interfészen keresztül
- beérkező adatok ellenőrzése, majd megfelelő formátumban továbbküldése a kontrollerhez

1.4.2. Szerver oldali komponensek listája

1.4.2.a. Kontroller

A komponens feladata a szerver oldali működés összefogása. Kezeli a kliensek kapcsolatait, a kliensektől illetve a hirdetői és adminisztrátori oldaltól érkező kéréseket, és azoknak megfelelően irányítja a lekérdező és módosító modult. A visszatérési értékeket továbbítja az azt kérő modul felé.

Feladatok:

- szerver oldali program fő szála, többi modul irányítása
- kapcsolatok kezelése a kliensekkel
 - bejövő port figyelése
 - kliens kapcsolatkérese esetén egy kapcsolat létrehozása, függetlenül a többi kapcsolattól
 - lehetőség több kérés-válasz periódusra egy kapcsolatfelvételen belül (vagyis a kapcsolat állapotát is tárolni kell)
 - kapcsolat korrekt lezárása (sávszélesség-pazarlás elkerülésére), függetlenül a többi kapcsolattól

- kapcsolat kezelése a hirdetői és adminisztrátori oldallal
- beérkező kérések feldolgozása, továbbítás a megfelelő modul felé
- válaszok fogadása, továbbítás a kérő modul felé
- lekérdező modul kezelése
 - paraméterek összeállítása, elküldése
 - lekérdezés eredményének továbbítása
- módosító modul kezelése
 - paraméterek összeállítása, elküldése
 - sikeresség vagy hibaok továbbítása

1.4.2.b. Lekérdező

A modul feladata az adatbázisban történő lekérdezések megvalósítása.

Feladatok:

- kontrollertől érkező paraméterek alapján lekérdezés összeállítása, majd az adatbázishoz fordulás
- eredmények visszaküldése a controllerbe
- esetleges meghiúsuló keresés esetén a hibaok visszaküldése

1.4.2.c. Módosító

A modul feladata a módosító adatbázis elérések megvalósítása.

Feladatok:

- kontrollertől érkező paraméterek alapján a módosító utasítás összeállítása, majd az adatbázishoz fordulás
- módosítás eredményének (sikeresség vagy hibaok) visszaküldése a controllerbe

1.4.2.d. Hirdetői oldal

Ez a komponens egy publikus weblap lesz, ami arra alkalmas, hogy a hirdetők tudjanak regisztrálni, megadni a saját pozíciójukat illetve a hirdetett hely egyéb adatait (pl. cím, elérhetőség, weblap, stb.). Nem terjedne azonban ki különböző, egyébként egy valós alkalmazásban fontos elemekre, mint például a fizetési paraméterek és a számlázás lehetőségei.

Feladatok:

- regisztráció
- bejelentkezés
- új hirdetés létrehozása
- keresés korábbi hirdetések között
- korábbi hirdetések listázása, törlése, módosítása

- minden hirdetői adat és tevékenység az adatbázisban fog tárolódni, így a hirdetői oldal kapcsolatban kell, hogy álljon a kontrollerrel, melyen keresztül elérhető az adatbázis

1.4.2.e. Adminisztrátori oldal

Ez a komponens arra szolgál, hogy a rendszer karbantartója figyelemmel tudja kísérni a működést. Itt lehetőség lesz a naplófájlok, a felhasználók, hirdetések lekérésére, utóbbiak törlésére is, amennyiben szükséges.

Feladatok:

- bejelentkezés
- lekérdezések összeállítása naplófájlokról
- statisztikák készítése – pl. felhasználók, hirdetések
- módosítási lehetőség – pl. felhasználó, hirdetés törlése, maximális felhasználószám változtatása

1.5. Környezet

A programozási és implementációs környezet nagyrészt a feladat kiírásából fakadóan adott. Így például az alkalmazás által használt adatbázisokat Oracle 11g adatbázis-kezelőben fogom tárolni, kihasználva a Spatial adta lehetőségeket.

Az alkalmazás forráskódját Java-ban készítem el, mivel a legtöbb mobil eszköz támogatja ezt a platformot, így az alkalmazás használata nem jelent majd nehézséget. További előny, hogy az Oracle beépítetten kínál Java fordítót (JDeveloper), mely így még kényelmesebbé teszi a munkát. Ezen kívül az Oracle Spatial szerves kiegészítője a MapViewer alkalmazás, mely a térképek megjelenítését teszi lehetővé. Ezt is Java-ban írták, ráadásul a JDeveloper is támogatja a használatát. A Java alkalmazása még azzal az előnnyel is jár, hogy a kliens-szerver architektúra megvalósítása sem jelent nagy nehézséget, mivel ez funkció a Java-ban egyszerűen megvalósítható, így a kommunikációs háttér megvalósítása sem fog túl nagy feladatot jelenteni. Java-ban továbbá van lehetőség webes felületek implementálására (Java Servlet) is, így egy felületen meg tudom oldani a különböző felmerülő problémákat. A JDeveloper fejlesztőkörnyezet minden fent felsorolt követelménynek eleget tesz.

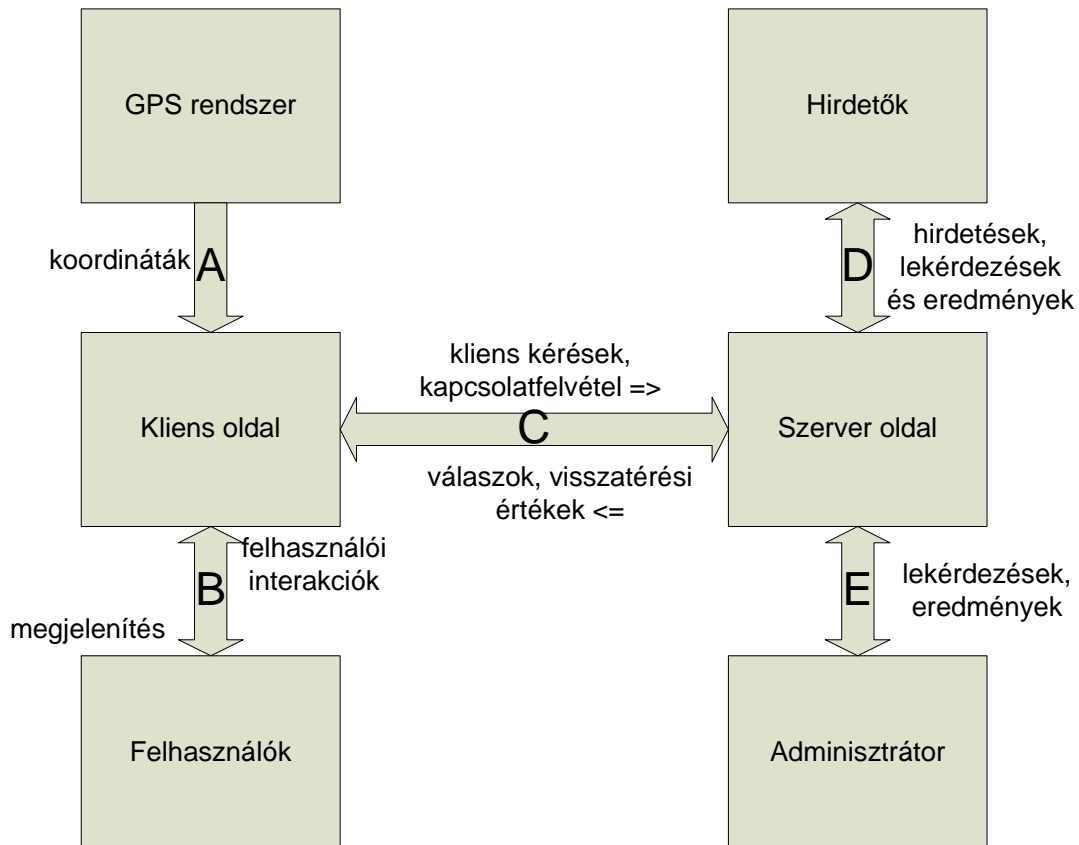
Mivel a Java alapvetően platformfüggetlen programozási nyelv, így a szoftvereket futtató operációs rendszer típusa csaknem lényegtelen. A GPS vevővel is rendelkező mai mobil eszközök (okostelefonok, PDA-k) nagy része Windows vagy Symbian alapú operációs rendszert használnak. A fejlesztés szintén Windows operációs rendszer alatt fog történni, sőt a szerver oldali alkalmazás is a fejlesztői gépen fog futni. A későbbiekben lehet szó nyilvános webszerveren való üzemelésről is, ez azonban még a jövő tárgya egyelőre.

Mivel nem rendelkezem egyelőre olyan eszközzel, ami GPS vevőt tartalmaz, ezért valószínűleg szoftveres emulátort fogok majd használni a teszteléshez.

1.6. Rendszert leíró blokkvázlatok, interfészek

1.6.1. Teljes rendszer

Teljes alkalmazás



1. ábra Rendszerszintű blokkvázlat

Rendszerszintű interfészek (1. ábra):

A interfész

A kliens oldali alkalmazás és a GPS rendszer közötti szabványos interfész.

Üzenetformátum: szabványos.

B interfész

A kliens oldali alkalmazás és a felhasználó közötti interfész. A felhasználó felől interakciók és azok paraméterei küldődnek a kliensnek, ezek az alábbiak lehetnek:

- gombnyomás
- egérekattintás
- érintőképernyő-érintés

- kurzormozgatás

Paraméterek lehetnek például az űrlapokba beírt szövegek vagy a kurzorpozíció. A felhasználó felé a megjelenítés zajlik ezen az interfészen, esetleg hangjelzés.

Üzenetformátum: nem definiálható, a kezelői felületbe építve.

C interfész

A kliens és a szerver oldali komponens közötti kommunikáció helye. A szerver egy adott porton figyel a kliensek kapcsolatkéreseit. Beérkező kapcsolatkéres esetén kiépül a csatorna a két oldal között. A kliens kéréseket küld, és ezekre várja a választ a szerver felől. A kérések lehetnek:

- belépés és regisztráció
- kilépés
- térkép lekérés
- csoportjelentkezés
- csoportból kijelentkezés
- csoport keresése
- hirdetés keresése
- hirdetés olvasása

Szerver oldali válasz lehet nyugta (például kilépés esetén), vagy lehet adatcsomag is (például térkép lekérés esetén).

Üzenetformátum: 3-féle üzenet-válasz típust lehet megkülönböztetni.

1. Kapcsolatfelépítés üzenetei

Mező	Lehetséges értékek	Leírás
irány	kliens => szerver	kérés
	szerver => kliens	válasz
IP cím (csak kérés)	x.x.x.x	kliens IP címe
port szám (csak kérés)	y	kliens port száma
nyugta (csak válasz)	pozitív	kérés elfogadva
	negatív	kérés elutasítva

2. Egyéb kérés, válaszban csak nyugta lehet

Mező	Lehetséges értékek	Leírás
irány	kliens => szerver	kérés
	szerver => kliens	válasz
kérés típusa (csak kérés)	belépés / csoport létrehozás / stb.	megadja a paraméterek értelmezését
paraméter(ek) (csak kérés)	azonosítók / csoportnév / stb.	kérés paramétere
nyugta (csak válasz)	pozitív	kérés elfogadva
	negatív	kérés elutasítva

3. Egyéb kérés, válaszban adatcsomag érkezik

Mező	Lehetséges értékek	Leírás
irány	kliens => szerver	kérés
	szerver => kliens	válasz
kérés típusa (csak kérés)	térkép lekérdezés / hirdetések keresése / stb.	megadja a paraméterek értelmezését
paraméter(ek) (csak kérés)	koordináták / keresési feltételek / stb.	kérés paraméterei
adatcsomag (csak válasz)	valódi adat	lekérdezés eredménye
	NULL érték	lekérdezés sikertelen

D interfész

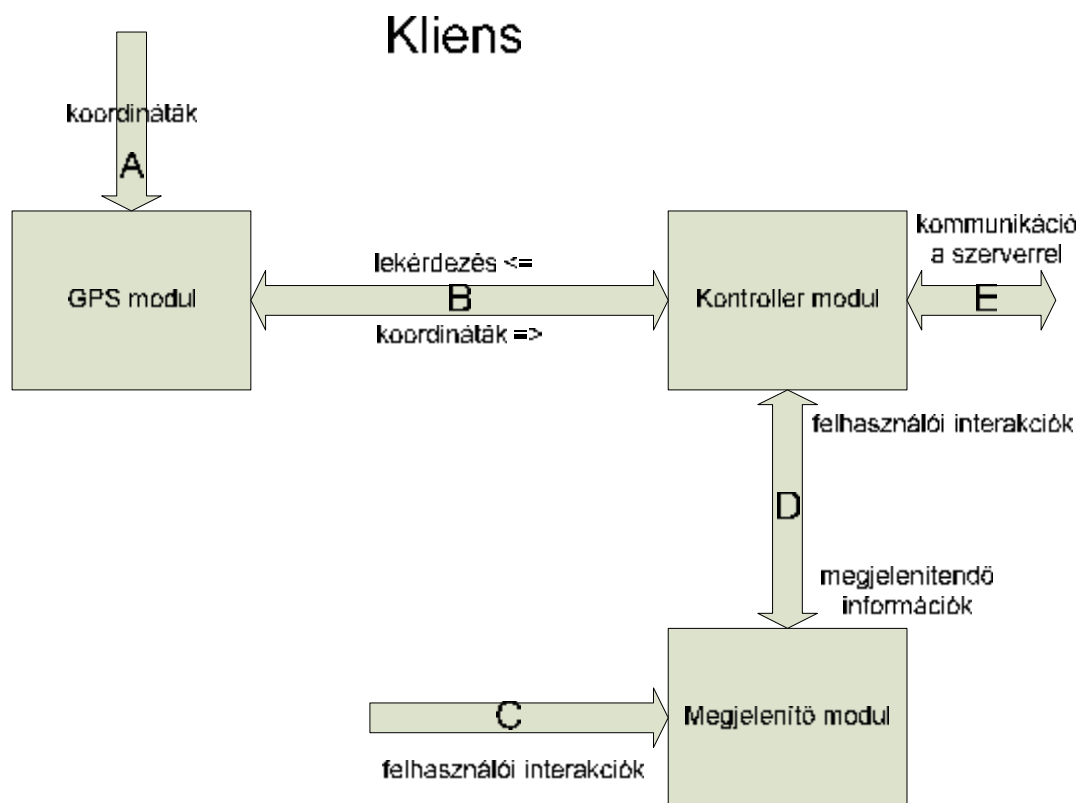
Hirdetők és szerver oldali alkalmazás közötti interfész. Ezt a szerepet a hirdető oldal felülete tölti be.

Üzenetformátum: nem definiálható.

E interfész: az adminisztrátor és a szerver oldali alkalmazás közötti interfész. Ezt a szerepet az adminisztrátori oldal felülete tölti be.

Üzenetformátum: nem definiálható.

1.6.2. Kliens oldal



2. ábra Kliens oldali alkalmazás blokkvázlata

Kliens oldali interfészek (2. ábra):

A interfész

A GPS-kezelő modul és a GPS rendszer közötti szabványos interfész, lásd 1.6.1. *A interfész*.

B interfész:

GPS modul és a kontroller modul közötti interfész. A kontroller kérést küld a GPS modulnak, melyre válaszképpen az aktuális GPS koordináta értékek érkeznek vissza.

Üzenetformátum:

Mező	Lehetséges értékek	Leírás
irány	kontroller => GPS	kérés
	GPS => kontroller	válasz
koordináták (csak válasz)	x = ...; y = ...; z = ...	a GPS rendszertől kapott értékek

C interfész

Megjelenítő modul és a felhasználó közötti interfész, lásd 1.6.1. *B interfész*.

D interfész

Kontroller és megjelenítő modulok közötti interfész. A kontroller a megjelenítendő adatot küldi a megjelenítő felé, míg a megjelenítő a feldolgozott felhasználói utasításokat küldi a kontrollernek.

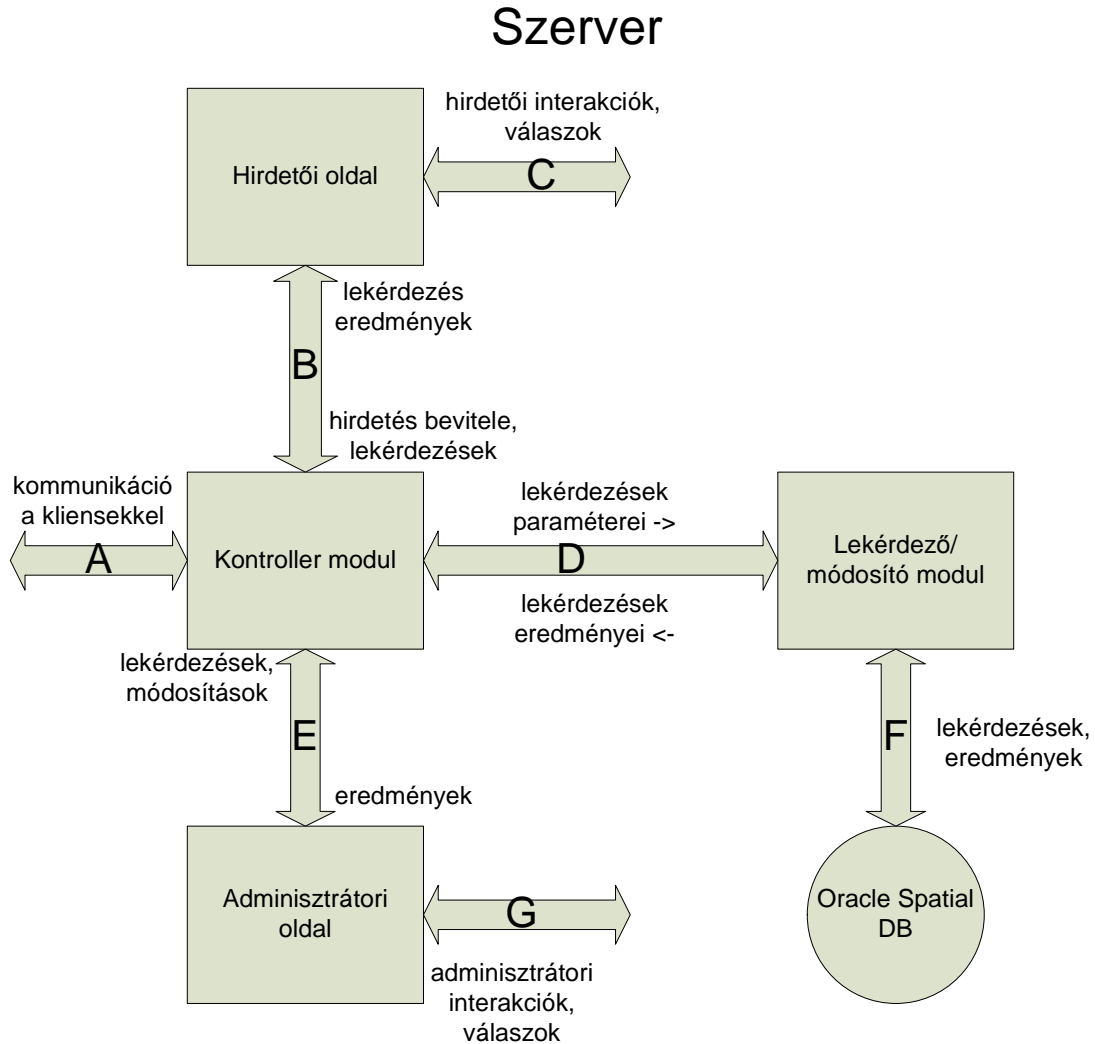
Üzenetformátum:

Mező	Lehetséges értékek	Leírás
irány	kontroller => megjelenítő	megjelenítendő adat
	megjelenítő => kontroller	felhasználói utasítás
adat típus (1. irány)	térkép	térkép megjelenítése
	kurzorpozíció	kurzor megjelenítése
	menü	menü megjelenítése
adat (1. irány)	...	típus szerint értelmezendő adat
utasítás típus (2. irány)	egérekattintás	
	egérmozgatás	
	kijelölés	
utasítás adat (2. irány)	...	típus szerint értelmezendő adat

E interfész

Kontroller modul és a szerver közötti interfész. Lásd 1.6.1. *C interfész*.

1.6.3. Szerver oldal



3. ábra Szerver oldali alkalmazás blokkvázlata

Szerver oldali interfészek (2. ábra):

A interfész

A kliensek és a kontroller modul közötti interfész, lásd 1.6.1. *C interfész*.

B interfész

Kontroller modul és a hirdetői oldal közötti interfész. A hirdetői oldalról kérések érkeznek, ezek lehetnek:

- lekérdező jellegű kérések (például hirdetések listázása)
- módosító jellegű kérések (például hirdetés feladása)

Válasz lehet nyugta és adatcsomag is.

Üzenetformátum: 2-féle üzenet-válasz típust lehet megkülönböztetni.

1. Kérés, válaszban csak nyugta lehet

Mező	Lehetséges értékek	Leírás
irány	hirdetői oldal => kontroller	kérés
	kontroller => hirdetői oldal	válasz
kérés típusa (csak kérés)	belépés / hirdetés létrehozása / stb.	megadja a paraméterek értelmezését
paraméter(ek) (csak kérés)	azonosítók / hirdetés szövege / stb.	kérés paraméterei
nyugta (csak válasz)	pozitív	kérés elfogadva
	negatív	kérés elutasítva

2. Kérés, válaszban adatcsomag érkezik

Mező	Lehetséges értékek	Leírás
irány	hirdetői oldal => kontroller	kérés
	kontroller => hirdetői oldal	válasz
kérés típusa (csak kérés)	hirdetések keresése	megadja a paraméterek értelmezését
paraméter(ek) (csak kérés)	keresési feltételek	kérés paraméterei
adatcsomag (csak válasz)	valódi adat	lekérdezés eredménye
	NULL érték	lekérdezés sikertelen

C interfész

Hirdetők és a hirdetői oldal közötti interfész, lásd 1.6.1. *D interfész*.

D interfész

Kontroller és a lekérdező / módosító modul közötti interfész. A kontroller kéréseket küld a lekérdező / módosító modul felé, kérés lehet:

- lekérdező jellegű kérés
- módosító jellegű kérés

A kérés paramétereiben az adatbázishoz forduláshoz szükséges paramétereket kell megadni. Válasz lehet nyugta (módosítás) vagy adatcsomag (lekérdezés) is.

Üzenetformátum: 2-féle üzenet-válasz típust lehet megkülönböztetni.

1. Kérés, válaszban csak nyugta lehet

Mező	Lehetséges értékek	Leírás
irány	kontroller => lekérdező / módosító	kérés
	lekérdező / módosító => kontroller	válasz
paraméter(ek) (csak kérés)	táblanév / törlendő objektum azonosítója / stb.	kérés paramétere
nyugta (csak válasz)	pozitív	kérés elfogadva
	negatív	kérés elutasítva

2. Kérés, válaszban adatcsomag érkezik

Mező	Lehetséges értékek	Leírás
irány	kontroller => lekérdező / módosító	kérés
	lekérdező / módosító => kontroller	válasz
paraméter(ek) (csak kérés)	keresési feltételek	kérés paramétere
adatcsomag (csak válasz)	valódi adat	lekérdezés eredménye
	NULL érték	lekérdezés sikertelen

E interfész

Kontroller modul és az adminisztrátori oldal közötti interfész. Az adminisztrátori oldalról kérések érkeznek, ezek lehetnek:

- lekérdező jellegű kérések (például naplófájlok listázása)
- módosító jellegű kérések (például hirdetés törlése)

Válasz lehet nyugta és adatcsomag is.

Üzenetformátum: 2-féle üzenet-válasz típust lehet megkülönböztetni.

1. Kérés, válaszban csak nyugta lehet

Mező	Lehetséges értékek	Leírás
irány	admin. oldal => kontroller	kérés
	kontroller => admin. oldal	válasz
kérés típusa (csak kérés)	belépés / hirdető törlése / stb.	megadja a paraméterek értelmezését
paraméter(ek) (csak kérés)	azonosítók / hirdető neve / stb.	kérés paramétere
nyugta (csak válasz)	pozitív	kérés elfogadva
	negatív	kérés elutasítva

2. Kérés, válaszban adatcsomag érkezik

Mező	Lehetséges értékek	Leírás
irány	adminisztrátori oldal => kontroller	kérés
	kontroller => adminisztrátori oldal	válasz
kérés típusa (csak kérés)	hirdetők keresése / naplófájlok keresése / stb.	megadja a paraméterek értelmezését
paraméter(ek) (csak kérés)	keresési feltételek	kérés paraméterei
adatcsomag (csak válasz)	valódi adat	lekérdezés eredménye
	NULL érték	lekérdezés sikertelen

F interfész

Lekérdező / módosító modul és az adatbázis közötti interfész. Az adatbázis felé SQL utasítások formájában érkeznek a kérések. Az adatbázis válaszolhat nyugtával és lekérdezés eredményét tartalmazó adathalmazzal is.

Üzenetformátum:

2-féle üzenet-válasz típust lehet megkülönböztetni.

1. Módosító utasítás

Mező	Lehetséges értékek	Leírás
irány	módosító => adatbázis	kérés
	adatbázis => módosító	válasz
SQL utasítás (csak kérés)	update ... / alter table ...	SQL utasítás
nyugta (csak válasz)	pozitív	kérés elfogadva
	negatív	kérés elutasítva

2. Lekérdező utasítás

Mező	Lehetséges értékek	Leírás
irány	lekérdező => adatbázis	kérés
	adatbázis => lekérdező	válasz
SQL utasítás (csak kérés)	select ... from ... where ...	SQL utasítás
adatcsomag (csak válasz)	valódi adat	lekérdezés eredménye
	NULL érték	lekérdezés sikertelen

G interfész

Az adminisztrátor és az adminisztrátori oldal közötti interfész, lásd 1.6.1. E interfész.

2. Tesztelési terv

A program tesztelését két fő fázisra bontottam.

Az első a komponensek önálló tesztelése. Ekkor a feladat az, hogy egy komponens mindenben megfelel-e az elvárt működésnek. Tesztelni kell a komponens funkcionálisan, vagyis ellenőrizni, hogy minden tipikus jó bemenet esetén az elvárt kimenetet produkálja-e.

Emellett szükséges egy hibatűrési teszt is, melynek során tipikus rossz bemenetekre vizsgáljuk a komponens működését. Ilyen rossz bemenettel modellezhető egy képzetlen vagy rosszindulatú felhasználó, kívülről érkező adat esetén így jellemezhető az adatszolgáltató hibája. Alapvető elvárás, hogy az alkalmazás ne omoljon össze ilyen esetben, legalább tájékoztassa a felhasználót a hibáról. Természetesen optimális esetben a hiba korrekt kezelése megy végbe, és ennek tesztelését is szolgálja a hibatűrési teszt.

Bizonyos modulok esetén szükséges lehet olyan teszt végrehajtása is, mellyel a komponens teljesítményét, válaszidejét teszteljük különböző terhelések mellett.

Ezekhez a tesztelésekhez egy egyszerű kis alkalmazás segítségét fogom igénybe venni. Az alkalmazás bemenete a modul neve és a bemeneti tesztfájl vagy tesztvektor lesz, míg a kimenet egy fájlba kerül. Ez a kimeneti fájl tartalmaz minden függvényhívást, illetve a modul által generált kimenetet. Az alkalmazás feladata mindössze annyi lesz, hogy elindítja a modult a megfelelő bemeneti értékeket paraméterként átadva, majd a kimenetet fájlba menti. Ehhez természetesen a modulokat módosítani szükséges a teszteléshez, de azok funkcionalitását nem érinti. Ezek a módosítások az alábbiak: minden függvényhíváskor kimenet generálása, minden kimenet átirányítása a kimeneti fájlba, bemeneti adatok kezelése.

A teszt második része az alkalmazás globális tesztelése. Ennek során már minden modul a végleges formájában, az alkalmazásba ágyazva van jelen. A globális tesztelés esetén élhetünk azzal a feltevéssel, hogy a modulok önmagukban megfelelően működnek, így a globális tesztelés során az egymással való kooperációjukat kell vizsgálni.

Természetesen ebben az esetben is az elsődleges a funkcionális teszt, vagyis hogy a rendszer az elvártaknak megfelelően működik-e. Itt is szükség lehet hibatűrési és teljesítménytesztekre, mivel a komponensek hibatűrő tulajdonsága és megfelelő teljesítménye nem garantálja a teljes rendszer hasonló tulajdonságát.

A globális tesztelés felhasználók általi tesztelés lesz.

2.1. *Komponens tesztek*

2.1.1. **Kliens oldal**

2.1.2. **Szerver oldal**

2.2. *Globális tesztelés*

3. Üzemeltetési terv

Az üzemeltetés során kétféle alapvető feladattal kell számolni. Az első ezek közül a rendszer monitorozása, vagyis különböző paraméterek figyelése, riportok készítése. Ezek lehetnek 1 vagy több órára, egy napra vagy akár hosszabb időtartamra vonatkozó statisztikák. Az ilyen jellegű adatgyűjtés alapját a naplófájlok képezik.

A második, az előzőből részben következő feladat a rendszer karbantartása, mivel karbantartásra lehet szükség akkor, ha rendellenes viselkedést tapasztalunk a rendszernél. Emellett karbantartási feladatnak minősül minden olyan tevékenység, mely a rendszer működtetésével kapcsolatos. Ilyen karbantartási feladat például a jelen feladatban a térképek frissítése, már nem létező felhasználók rendszerből való törlése.

Mindezen feladatok végrehajtásához szükség van egy adminisztrátori felületre. Itt lehetőség van mind különböző riportok létrehozására, mentésére; illetve módosítások végrehajtására.

3.1. Monitorozás

A monitorozás alapja tehát a napló lesz. A naplóba események kerülnek, minden eseményhez a pontos időpont, az eseményhez kapcsolódó felhasználó és esetleges egyéb paraméterek kerülnek letárolásra.

Eseménynek nevezhetjük a felhasználók minden olyan cselekedetét, ami hatással van a szerver oldali működésre. Tehát a hirdetőik és az adminisztrátorok esetében minden cselekedet bekerül a naplóba, a valódi felhasználóknál viszont például egy térkép lekérése bekerül, míg ugyanennek a térképnek az elforgatása a mobil eszközön már nem. Mivel így minden, a rendszer működésének szempontjából lényeges eseményről készül a naplóba bejegyzés, ezért ezek alapján rekonstruálható a működés.

Az alábbi, a rendszert jellemező paramétereket lehet monitorozni:

- felhasználók
 - felhasználók száma
 - új felhasználók száma
 - belépések száma (bejelentkezés)
 - térképlekérések száma
- hirdetőik / hirdetések
 - hirdetőik száma
 - hirdetések száma
 - új hirdetőik száma
 - új hirdetések száma
 - belépések száma (bejelentkezés)
- adatbázis
 - térkép utolsó frissítésének ideje
 - szabad hely mérete
 - adatbázis terheltség

Ezek mellett lehetőség lesz minden az adatbázisban található objektum listázására és keresésére is.

A fenti paraméterekről 24 óránként összesítést kell készítenie az adminisztrátornak, és ezt az elmúlt 24 óra minden naplóbejegyzésével együtt az adatbázison kívül menteni kell, majd a naplót az adatbázisból törölni kell. Így az adatbázisban mindig csak legfeljebb az utolsó 24 óra eseményeit kell tárolni. Ezen feladat elvégzése kezdetben manuális lesz, de lesz lehetőség arra is, hogy ezt a későbbiekben egy automatizmus váltsa fel.

Riportkészítésnél az adminisztrátornak lehetősége lesz egy időintervallumot megadni, melyre vonatkozóan a fenti adatokat le szeretné kérdezni (*lásd 1. riport*). Természetesen válogathat a paraméterek között, nem kell feltétlenül mindent kiválasztani. Emellett a rendszerben lévő objektumok listázása és keresése is lehetséges (*lásd 2. riport*).

Példa az időintervallumra keresésnél, csak a felhasználók paramétereit kiválasztva:

2008. 09. 25. 16:00 – 2008. 09. 27. 16:00	
<u>Felhasználói statisztika</u>	
- Rendszerben lévő felhasználók az intervallum elején:	
	32
- Rendszerben lévő felhasználók az intervallum végén:	
	35
- Új felhasználó:	
	4
- Törölt felhasználó:	
	1
- Összes bejelentkezések száma:	
	77
- Összes térképlekérés száma:	
	657
	Készült:
	2008. 09. 30. 21:09

1. riport: Statisztika

Példa listázásra, hirdető neveinek listázása, szűrés azokra a nevekre, amelyekben szerepel a „Bt.” kifejezés:

```
Hirdetők
-----
- Megjelenített attribútum(ok):      név
- Szűrési feltétel(ek):              név => %Bt%
- Rendezés:                          csökkenő

Szűrési feltételnek megfelelő elemek:

- Zebra Bt.
- Tehénke Bt.

...

- EBtanoda Kutyaiskola

Készült:
2008. 10. 01. 02:37
```

2. riport: Szűrés, keresés

A fentihez hasonló riportok készítésére két lehetőség lesz:

- előre megírt, paraméterezhető lekérdezések, ahol az adminisztrátornak elegendő csak a paraméterek értékeit megadni
- bonyolultabb lekérdezés esetén SQL parancsszerkesztő

3.2. Karbantartás

A következő alapvető karbantartási feladatokat kell tudnia ellátnia az adminisztrátornak:

- felhasználó törlése
- hirdető törlése
- hirdetés törlése
- térkép frissítése (régiről törlés mellett)
- térkép frissítése (régiről bővítés)
- napló mentése adatbázison kívül
- maximális felhasználó- / hirdető- / hirdetésszám állítása

Rendszerbeli objektumok törlésére akkor lehet szükség, ha az adott objektum már inaktív, nincs használatban. Ilyen például egy felhasználó, aki egy bizonyos időkorláton belül nem jelentkezett be a rendszerbe. Természetesen lehetőség van akár felhasználók, akár hirdetők vagy hirdetések moderálására is a törléssel.

Térképfrissítés kétféleképpen történhet. Az első esetben a régi térképet lecseréljük egy frissebbre, míg a második esetben a térképek számát növeljük, ezáltal frissítjük a rendszert.

Amennyiben a monitorozás során a rendelkezésre álló terület túl kevésnek bizonyul vagy az adatbázis túlterhelt, akkor lehetőséget kell biztosítani a maximális felhasználószámok módosítására, vagy a naplófájlok mentésével tárterület felszabadítására.

A karbantartási feladatok esetén is két lehetősége lesz az adminisztrátornak:

- egyrészt a fenti feladatokra előre megírt, paraméterezzhető utasítások állnak majd rendelkezésre, melyekhez csak a paraméterek értékeit kell megadni
- másrészt egy SQL parancsszerkesztő minden ezektől eltérő feladat megvalósítására

4. Adatszerkezeti terv

Az adatszerkezeti terv a passzív erőforrások leírását tartalmazza. Passzív erőforrások közé szokás sorolni mind az adatbázisokat, mind pedig a naplófájlokat. Jelen esetben e két kategória nem különíthető el ilyen élesen.

Terveim szerint a naplózás részben az adatbázisban, részben pedig önálló, az adatbázistól független fájlokban valósul meg. Ennek segítségével csökkenthető az adatbázis mérete, viszont a régebbi naplóbejegyzések keresési ideje megnövekszik.

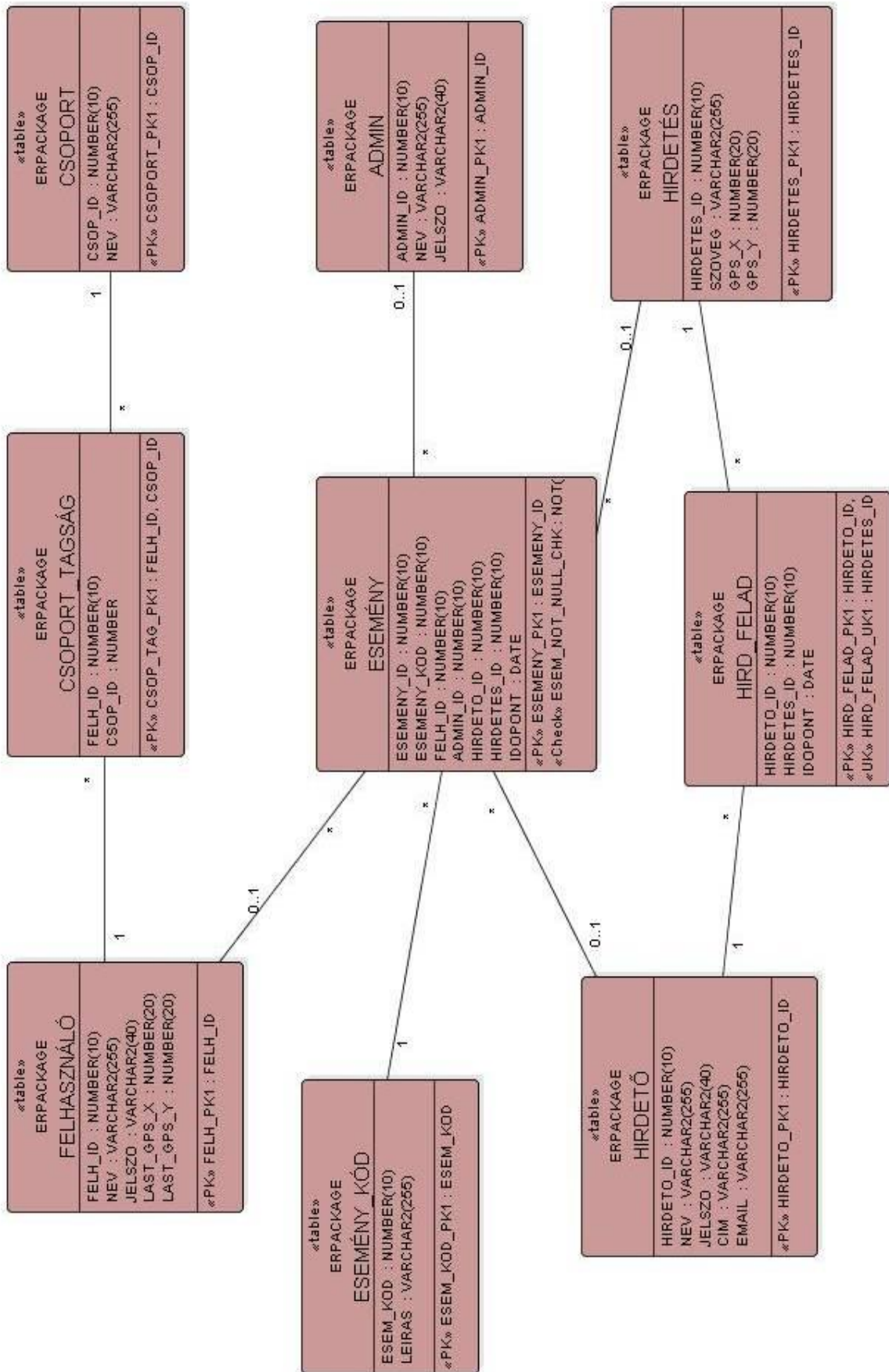
4.1. Adatbázis terv

Az adatbázis tulajdonképpen két nagy részre bontható.

Az egyik rész egy térkép, amit készen fogok beszerezni, kezdetben nagy valószínűség szerint a Navteq cég által kiadott, ingyenesen elérhető Magyarország térkép lesz. Később természetesen a térképek száma bővíthet. Az adatbázis ezen részének tervezésével ebben a rendszertervben nem foglalkozom.

A másik rész az, amit én tervezek meg és implementálok. Ezekben a táblákban fognak tárolódni a felhasználók adatai, a hirdetőik adatai és hirdetések, valamint a friss naplóbejegyzések.

Adatbázis megtervezéséhez az általunk tanult eszköz az entitás-relációs diagram készítése, a 4. ábrán látható az elkészült diagram. A diagram az Oracle 10g-hez kapcsolódó JDeveloper alkalmazással készült. Ez az alkalmazás képes a diagramból SQL kódot generálni, mellyel az adatbázisban az adatmodell létrehozható, meggyorsítva ezzel a fejlesztést.



4. ábra Az adatbázis entitás-relációs diagramja

4.1.1. Entitások táblái

Felhasználó: a mobil eszközén az alkalmazást futtató felhasználót reprezentálja.

Attribútumok:

- felh_id – azonosító
- név
- jelszó
- utolsó ismert GPS koordináták

Elsődleges kulcs:

- felh_id

Csoport: a felhasználók egy csoportját reprezentáló tábla.

Attribútumok:

- csop_id – azonosító
- név

Elsődleges kulcs:

- csop_id

Admin: adminisztrátorokat reprezentáló tábla.

Attribútumok:

- admin_id – azonosító
- név
- jelszó

Elsődleges kulcs:

- admin_id

Hirdető: a hirdetőket reprezentáló tábla.

Attribútumok:

- hirdeto_id – azonosító
- név
- jelszó
- cím
- email

Elsődleges kulcs:

- hirdeto_id

Hirdetés: a hirdetések reprezentáló tábla.

Attribútumok:

- hirdetes_id – azonosító
- szöveg
- GPS koordináták

Elsődleges kulcs:

- hirdetes_id

4.1.2. Kódtábla

Esemény_kód: különböző események kódolását tartalmazó tábla.

Attribútumok:

- esem_kod – esemény kódja
- leírás – szöveges leírása az esemény típusának

Elsődleges kulcs:

- esem_kod

4.1.3. Kapcsolatok táblái

Csoport_tagság: a felhasználók csoportba tartozását reprezentálja.

Attribútumok:

- felh_id – felhasználó azonosítója, idegen kulcs – több-egy kapcsolat
- csop_id – csoport azonosítója, idegen kulcs – több-egy kapcsolat

Elsődleges kulcs:

- felh_id, csop_id

Hird_felad: hirdető és az általuk feladott hirdetések közötti megfeleltetés.

Attribútumok:

- hirdeto_id – hirdető azonosítója, idegen kulcs – több-egy kapcsolat
- hirdetes_id – hirdetés azonosítója, idegen kulcs – egy-egy kapcsolat
- időpont – hirdetés feladásának ideje

Elsődleges kulcs:

- hirdeto_id, hirdetes_id

Megszorítás:

- hirdetes_id-nak egyedinek kell lennie

Esemény: az üzemeltetési tervben definiált eseményeket (naplóbejegyzéseket) reprezentáló tábla.

Attribútumok:

- esemeny_id – egyedi azonosító
- esemeny_kod – esemény típusának kódja, idegen kulcs – több-egy kapcsolat
- felh_id – felhasználó azonosítója, idegen kulcs – több-egy kapcsolat
- admin_id – adminisztrátor azonosítója, idegen kulcs – több-egy kapcsolat
- hirdeto_id – hirdető azonosítója, idegen kulcs – több-egy kapcsolat
- hirdetes_id – hirdetés azonosítója, idegen kulcs – több-egy kapcsolat
- időpont

Elsődleges kulcs:

- esemeny_id

Megszorítás:

- felh_id, admin_id, hirdeto_id, hirdetes_id egyszerre nem lehet mind null érték – ebben az esetben nem lehetne az eseményhez objektumot kötni

4.2. Naplófájlok

A naplófájlok szerkezetét meghatározza a fent vázolt adatbázis-szerkezet. Mivel az Esemény tábla bejegyzéseit kell a naplóba kiírni, egy naplóbejegyzés az alábbi formátumú lesz majd:

<időpont><esemény kód><esemény leírás><felhasználó név><felhasználó típus><...>

Jelen esetben a felhasználó lehet valódi felhasználó, adminisztrátor és hirdető is.

Példa egy naplórészletre:

```
...
2008. 10. 02. 16:58
5 - Belépés - Gipsz Jakab - Felhasználó

2008. 10. 02. 17:01
7 - Térkép lekérés - Nagy Alajos - Felhasználó

2008. 10. 02. 17:02
2 - Hirdetés törlés - Adminisztrátor1 - Adminisztrátor - hirdetes_id = 4576457152
...
```

5. Megvalósítási terv

5.1. Objektumkatalógus

5.1.1. Kliens oldal

A kliens oldal osztályairól készült UML diagram az 5. ábrán látható.

5.1.1.a. Controller

Példányok száma: 1

Komponensek:

- View
- GPS
- ConnectionSetup

Változók: -

Függvények:

- LoginRegistration - belépés és regisztráció
- Logout - kilépés
- CoordinateRequest - koordinátakérés a GPS modul felől
- MapRequest - térképkérés a szerverből
- Navigate - navigálás a térképen
- GroupLogin - csoportjelentkezés
- GroupLogout - csoportból kijelentkezés
- GroupSearch - csoportok keresése
- AdSearch - hirdetések keresése
- AdLook - hirdetés olvasása

5.1.1.b. View

Példányok száma: 1

Komponensek: -

Változók: -

Függvények:

- Draw - a képernyőt újrajzoló függvény

5.1.1.c. GPS

Ez a komponens egy előre elkészített, átvett modul lesz.

5.1.2. Szerver oldal

A szerver oldal osztályairól készült UML diagram az 6. ábrán látható.

5.1.2.a. Controller

Példányok száma: 1

Komponensek: ConnectionSetup, Select, Modify

Változók: -

Függvények:

- LoginRegistration - belépés és regisztráció
- Logout - kilépés
- MapSearch - térkép lekérés
- GroupLogin - csoportjelentkezés
- GroupLogout - csoportból kijelentkezés
- GroupSearch - csoport keresése
- AdSearch - hirdetés keresése

5.1.2.b. Select

Példányok száma: 1

Komponensek: -

Változók: -

Függvények:

- MapSelect - térkép lekérdezése
- GroupSelect - csoport lekérdezése
- PasswordSelect - jelszó lekérdezése
- AdSelect - hirdetés lekérdezése
- LogSelect - naplóbejegyzés lekérdezése

5.1.2.c. Modify

Példányok száma: 1

Komponensek: -

Változók: -

Függvények:

- NewUser - felhasználó módosítása
- DeleteUser
- UpdateUser
- NewAd - hirdetés módosítása
- DeleteAd
- UpdateAd
- NewGroup - csoport módosítása
- DeleteGroup
- UpdateGroup
- NewLog - naplóbejegyzés létrehozása

5.1.2.d. AdPage

Példányok száma: 1

Komponensek: Select, Modify

Változók: -

Függvények: ez egy weblap lesz, ami majd a komponens osztály függvényeit hívja.

5.1.2.e. AdminPage

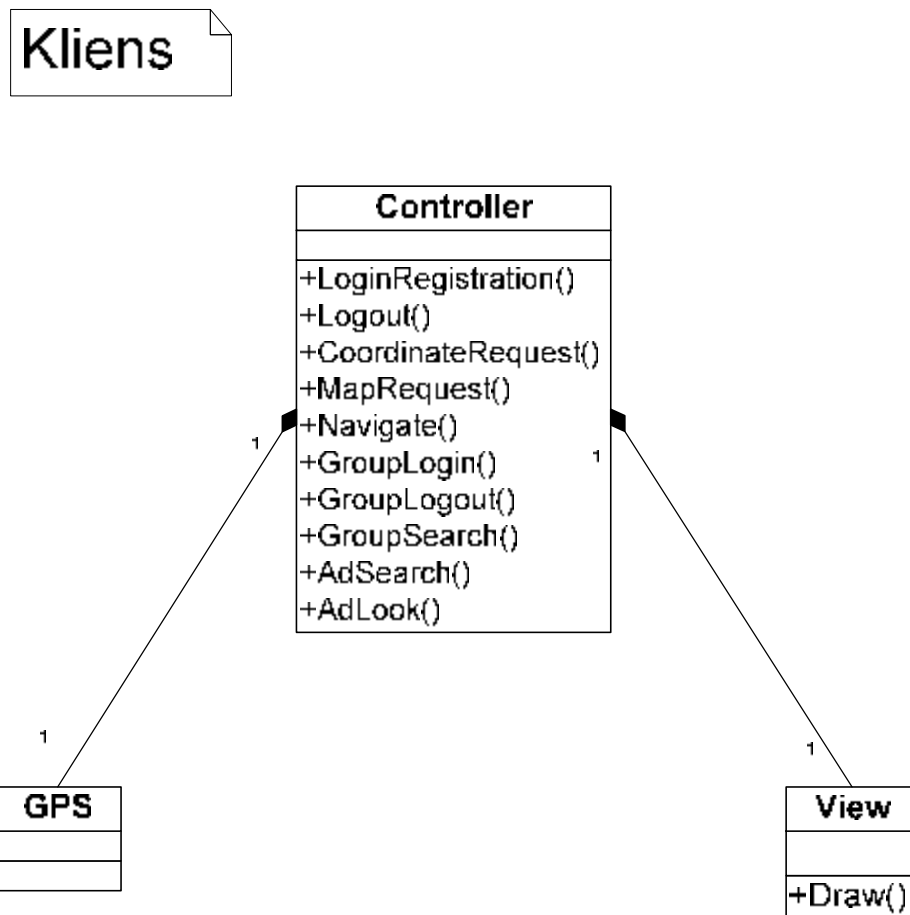
Példányok száma: 1

Komponensek: Select, Modify

Változók: -

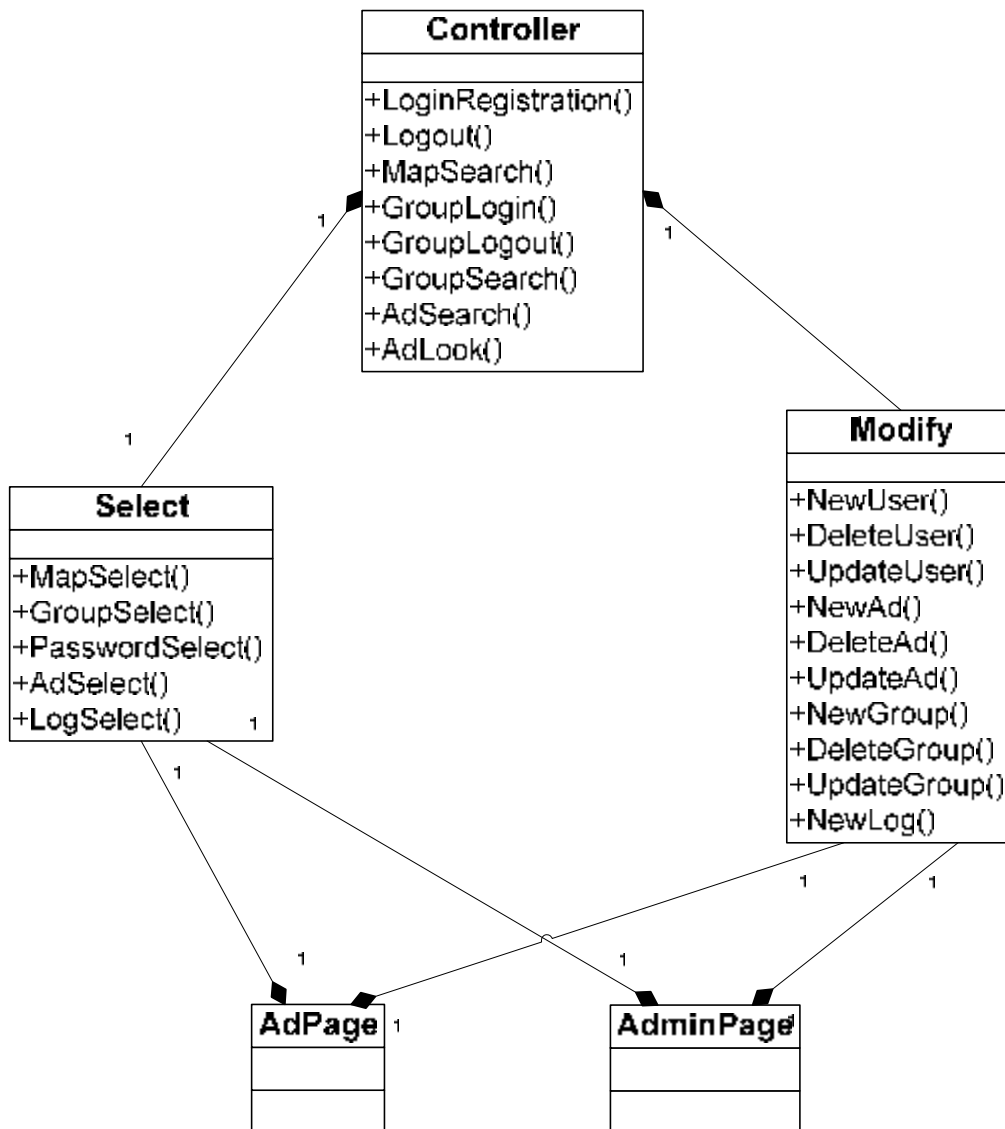
Függvények: ez egy weblap lesz, ami majd a komponens osztály függvényeit hívja.

5.2. Statikus osztálydiagramok



5. ábra Kliens statikus osztálydiagramja

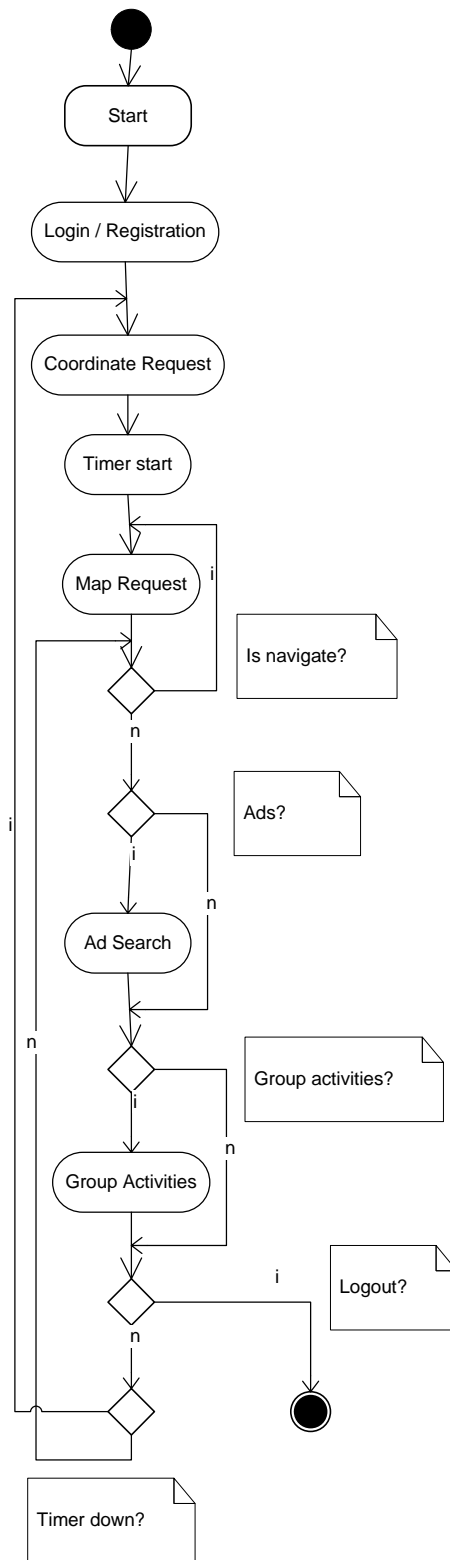
Szerver



6. ábra Szerver statikus osztálydiagramja

5.3. Kliens állapotgép

A kliens oldali program jól jellemezhető egy állapotgéppel. Az egész futás egy hurokban történik, melynek két feltétele lesz. Az egyik, hogy ki akarunk-e lépni a program futásából, a másik pedig egy időzítőt fog figyelni. Az időzítőt minden koordináta-lekérésnél fogjuk újraindítani, ha pedig lejár, akkor újrakérjük a koordinátákat és újrarajzoljuk a térképet is. Erre azért van szükség, hogy mozgás közben is követni tudjuk a helyzetünket. A többi funkciót ezen a hurkon belül hajtjuk végre. Ezek között a szekvencialitás nem feltétel, valószínűbb, hogy többszálúan fog a program futni, és eseményvezérelt lesz a hurkon belüli működés (7. ábra).

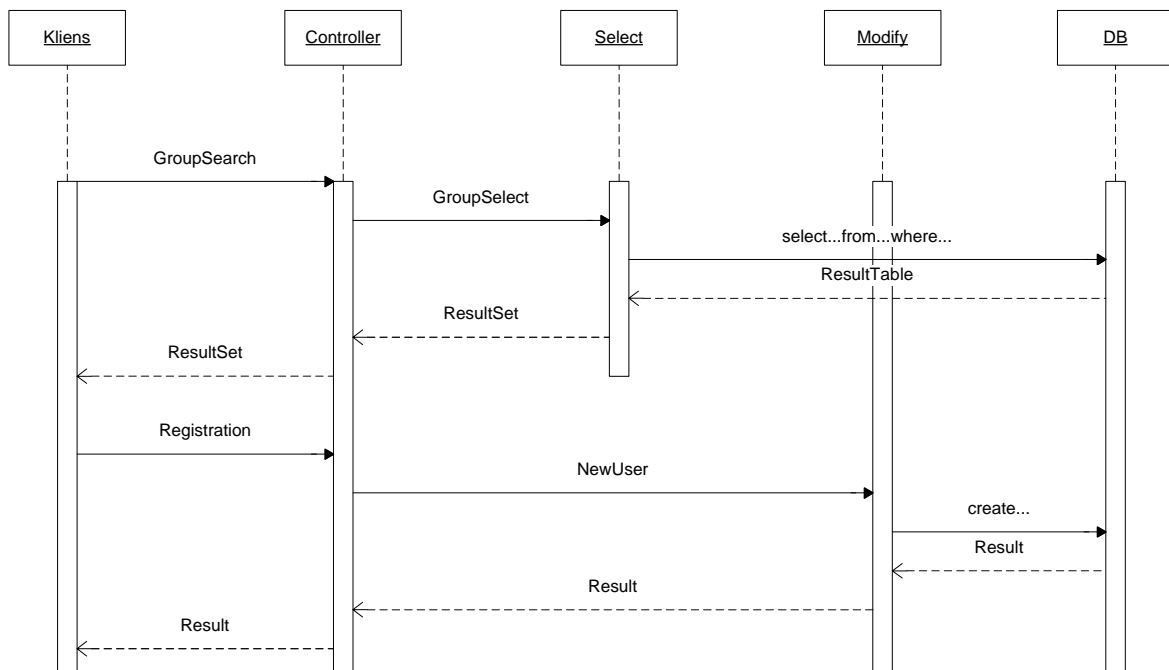


7. ábra Kliens állapotgép

5.4. Szerver szekvenciadiagramok

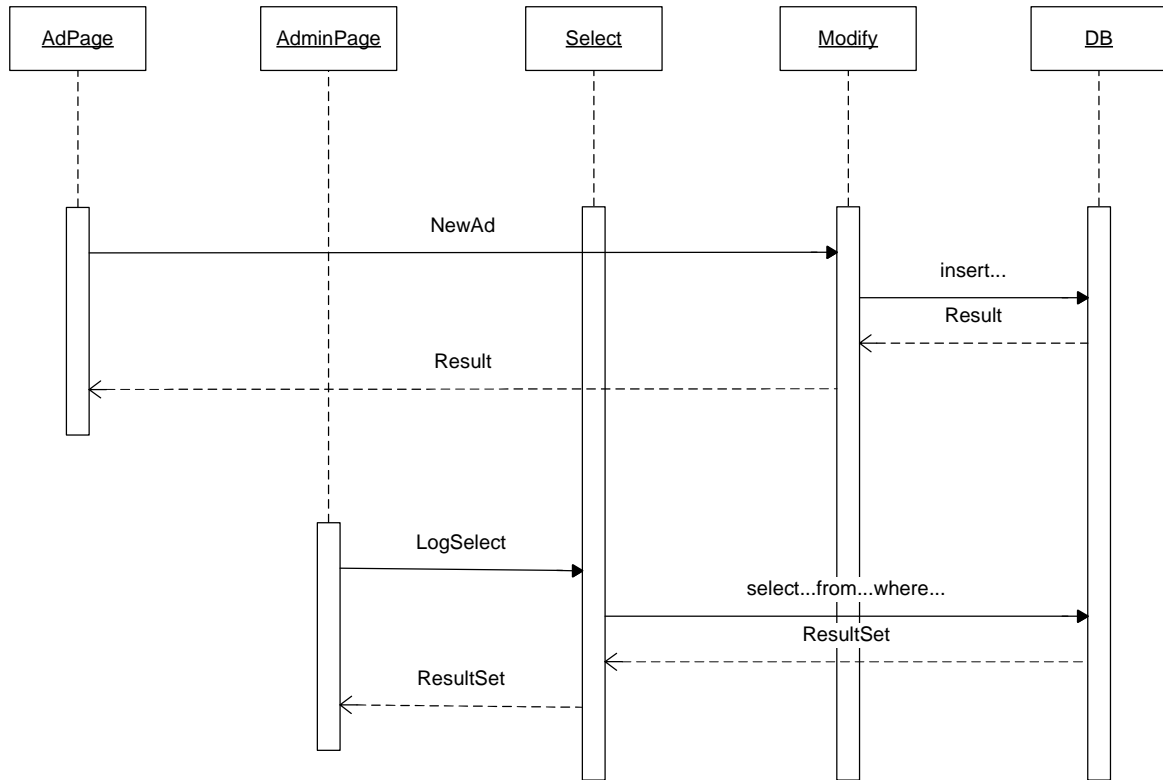
A szerver oldal sokkal jobban jellemezhető szekvenciadiagramokkal, mivel itt gyakorlatilag csak és kizárólag eseményvezérelten folynak a történések. Két esetet lehet élesen elkülöníteni.

Az egyik, amikor a kliens fordul a szerverhez. Ez az eset látható a 8. ábrán. Érkezik egy kérés, amit a szerver feldolgoz és továbbít a Select / Modify modul felé. Itt egy SQL utasítás lesz a kérésből, ami az adatbázis számára már értelmezhető. Az eredmény ugyanezen az útvonalon halad vissza a kliens felé.



8. ábra Szerver szekvenciadiagramja, amikor a kliens kérését szolgálja ki

A másik esetben a szerver oldalhoz tartozó AdPage vagy AdminPage oldalon keletkezik egy kérés. ezek közvetlenül a Select / Modify blokkhoz érkeznek. Ez látható a 9. ábrán. Itt ugyanúgy SQL utasítás lesz belőlük, majd az eredmény is hasonló módon jut vissza az oldalak valamelyikére.



9. ábra Szerver szekvenciadiagramja, amikor valamelyik weboldal kérését szolgálja ki