

SZAKDOLGOZAT

Bettenbuk Zoltán

Debrecen
2007

Debreceni Egyetem
Informatikai Kar

Mobiltelefonok programozása

A mobil internet lehetőségei

Témavezető:
Bátfai Norbert

Készítette:
Bettenbuk Zoltán
Programtervező informatikus

Debrecen
2007

Tartalomjegyzék:

1. Bevezetés és háttér.....	4
1.1 Előszó.....	4
1.2 Mobil eszközök.....	6
Mi az a mobil eszköz?.....	6
Kialakulásuk.....	7
Terjedésük.....	13
Szolgáltatásaik.....	14
2. Mobil internetes alkalmazások.....	17
2.1 Fogalma, megvalósíthatósága.....	17
2.2 Jelentősége, létjogosultsága.....	22
3. Egy mobil internetes alkalmazás.....	24
3.1 iSMS – az ötlet és a fejlesztés lépései.....	24
Tervezés.....	25
Szerveroldal.....	28
Mobiloldal (kliensoldal).....	33
3.2 Jó mobil program.....	40
4. A jövő.....	42
5. Irodalomjegyzék.....	44
Könyvek.....	44
Internet.....	44
6. Függelék.....	45

1. Bevezetés és háttér

1.1 Előszó

Napjaink elengedhetetlen használati tárgyává váltak a mobil eszközök. Irányítják mindennapi életünket, segítenek minket mindennapos dolgaink intézésében, és adott esetben már akár munkánkat sem tudnánk elvégezni nélkülük. Próbáljuk meg végiggondolni egy üzletember átlagos napját. Felébred reggel, valószínűleg a reggeli ébresztőt is a telefonján állítja be. Ezután reggeli közben elolvassa az esetleg éjszaka a tengerentúlról érkezett e-mailjeit (az időeltolódás miatt). Ezt természetesen komoly üzletember lévén nem az íróasztala mellett egy asztali gépről teszi meg, hanem a reggelizőasztalnál egy laptop segítségével. Vajon milyen internetet elérést használ a laptopon? Vagy van neki egy vezeték nélküli hálózat a lakásában, vagy egyszerűen Bluetooth segítségével kapcsolódik 3G vagy HSDPA képes telefonjához, aminek segítségével széles sávú internethez juthat bárhol. Nem hiszem, hogy vezetékek szaladgálnának át a reggelizőasztalon. Reggeli után beülve az autóba, munkába menet elintézi a legsürgősebb telefonjait szintén mobiltelefon segítségével, persze csakis kihangosítóval. A dugóban előveszi a PDA-ját, és elkezdi átnézni az aznapi teendőit, naptárbejegyzéseit, és természetesen az imént elintézett telefonok eredményességét is lejegyzí. Még be sem ért a munkahelyére, de már 3-4 olyan mobil eszközt használt, aminek megléte nélkül szinte lehetlenné válna munkájának elvégzése. És akkor nem is beszéltünk még az utazást elősegítő mobil helymeghatározó eszközökről (GPS, vagy manapság PNA), illetve a munkába érés után használt egyéb mobil dolgokról sem. Ez nem egy sarkalatos példa, gondoljunk csak bele mi is, hány helyen használunk hasonló eszközt életünk során, és aztán gondlunk végig azt is, hogy elvakult üzletemberek vagyunk-e, vagy nem. Én nem vagyok. Mégis számtalan ilyen példát tudnék felsorolni pár perc alatt. Jó, most mondhatjuk azt is, hogy azért, mert informatikus vagyok, de szerintem ez nem ezen múlik.

Ezen dolgozat pontosan az ilyen eszközök használatáról, terjedéséről szól, illetve az internet szolgáltatásairól, és kettejük kapcsolatáról. Igyekszem feltérképezni a mobil

eszközök képességeit, illetve visszatekintünk az idáig vezető útra. Megvizsgáljuk ezen eszközök korlátait is, illetve azt, hogy ezen korlátok hogyan befolyásolják használatukat. Megnézzük, hogyan kerül a képbe az Internet, milyen szolgáltatásokat érünk el mobil eszközök segítségével, illetve ezen szolgáltatások hogyan könnyítik meg életünket, hogyan tolják ki mobilitásunk határait. Rövidke áttekintést kapunk a manapság használatos vezeték nélküli hálózati/internetes szabványokról, illetve, és ez a dolgozat fő témája, mik azok a Mobil Enterprise Alkalmazások, miért és hogyan használhatjuk őket mindennapjainkban, hogyan segíthetik életünket, és hogyan gyorsíthatják az üzleti életet. Megvizsgálunk egy konkrét alkalmazást, és betekintést nyerünk ezek fejlesztésének lépéseibe, mikéntjeibe. Zárásképpen megpróbálunk belelátni a jövőbe, mind a mobil eszközök piacán, mind a vezeték nélküli szolgáltatások tekintetében.

A dolgozatban igyekszem párhuzamos síkon feltárni a mobil eszközök és az Internet kapcsolatát, hogy jól átlátható legyen kettejük szimbiózisa és az az erő, amely mindkét iparág fejlődését egyre csak gyorsítja, ez pedig nem más, mint a felhasználók tábora és az igény.

1.2 Mobil eszközök

Mi az a mobil eszköz?

Az előszóban már említettünk néhány mobil eszközt, de pontosan még nem is tudjuk, hogy mit jelent ez a kifejezés. Mivel a dolgozat gyakorlatilag róluk fog szólni, nem árt előbb tisztázni ezt. Ez azonban nem is olyan egyszerű. Ha megkérdezzük valakit arról, hogy vajon mi az a mobiltelefon, lehet, hogy valami olyasmi definíciót adna rá, hogy „*olyan telefon, ami hordozható*”... Jól megmagyaráztuk. És mi van, ha én az asztali telefont fölemelem, és hordozom? Az is mobiltelefon? Ja, mondjuk, hogy „... *hordozható és közben működik is...*”. Működik az is, amíg a vezeték elér. Akkor „... *működik és nincs vezetéke...*”. Vannak ám hordozható, vezeték nélküli vonalas telefonok is. Na itt jól bejöttünk a zsákutcába, se ki, se be. És akkor még csak a mobiltelefont próbáltuk meghatározni, nem is beszéltünk az általános, úgynevezett „mobil eszközökről”. Hát akkor próbáljunk meg ezekre egy általános definíciót adni. *Mobil eszköz olyan számítástechnikai eszköz, ami fizikailag könnyedén és szabadon mozgatható, és számítási képességei használhatóak mozgás közben is.* Úgy gondolom, ez jól lefedi a mobil eszköz fogalmát, habár nem tűnik valódi informatikai meghatározásnak, inkább olyan, mintha valami fizika könyvből emeltük volna ki. Gondoljunk csak bele, az előbb említett hordozható vonalas telefon tehát mobiltelefon? Nem, mivel fizikailag nem könnyen és szabadon mozgatható, ugyanis az asztali egységtől nem nagyon vihetjük 50-100m-nél messzebbre. Ezzel szemben a mobiltelefon most már szinte mindenhol képes (így vagy úgy) működni. Mi számít akkor mobil eszköznek? A már említett mobiltelefon határozottan mobil eszköz. De itt vannak még a laptopok, és ezek összes alfajai (subnotebook, tablet PC...stb.), az összes kézi számítógép (angolul handheld device, vagy personal digital assistant – PDA), navigációs egységek (GPS, vagy personal navigation assistant – PNA), de kis fantáziával ide sorolhatjuk a ma már elég intelligensnek számító hordozható zene és filmlejátszó eszközöket is. Amint látható, ugyanazon definíció alatt merőben eltérő

méretű, formájú, funkcionalitású eszközök találhatóak. Ennek oka feladatuk különbözősége, amire tervezték, gyártották őket. Lássuk tehát hogyan alakultak ki mai formáik és funkcionalitásaik.

Kialakulásuk

A kezdetek és a nulladik generáció.

A mobil eszközök fejlődése a mobiltelefonnal kezdődött, és mivel ezen eszközök, illetve a PDA-k a dolgozat fő témája, jobbra ezek történetét fogom vizsgálni. A hordozható telefon igénye nem mai keletű. Az első mobiltelefonok gyakorlatilag kétirányú adó-vevő készülékek voltak, illetve abból fejlődtek ki. Ekkor még a mobiltelefon, mint szó sem igazán létezett, jobbra rádiótelefonnak hívták, ami szintén a származására utalt. Európában a rádiótelefonálás, mint fogalom, először a Berlin és Hamburg között járó első-osztályú vonaton jelent meg 1926-ban, majd ugyanebben az évben az utasszállító repülőkön is. Később a Második Világháború idején a német tankok ennek segítségével tartották a kapcsolatot. Ezen eszközök használata akkoriban nagyon komoly képzettséget igényeltek, és csak arra szakosodott, képzett ember volt képes kezelni őket. Az első valóban „mobil” telefon 1950-ben jött ki, aminek használatához már nem kellett nagy hozzáértés. 1956-ban a svéd illetőségű Ericsson adta ki a világ első automata mobiltelefonos rendszerét (MTA – Mobile Telephone system A), viszont hátránya nagy súlya volt, a készülék közel 40kg-ot nyomott. Ezt továbbfejlesztve az Ericsson kiadott egy tranzistoros változatot, ami már csak 9 kg volt, és 150 előfizetővel büszkélkedhetett eleinte, viszont ez a szám nem haladta meg a 600-at sem a projekt lezárásáig 1983-ban. Akkor még nem volt automata átjárhatóság a cellák között, azaz egy telefonhívás erejéig egy cellában kellett tartózkodni. 1970-ben oldották meg a Bell Labs-nál ezt a problémát. Ezután 1971-ben indult útjára az első, valóban sikeres kereskedelmi mobiltelefon szolgáltatás Finnországban, ARP néven. Az ARP -t a szakirodalom sokszor nulladik generációnak emlegeti.

Első generáció

Az első, kereskedelmi forgalomban kapható mobiltelefon az 1983-ban megjelent Motorola DynaTAC 8000X volt. Elterjedésüket az akkorra már megoldott „handover” protokollok segítették, azaz már nem szakadtak meg a hívások a cella elhagyásakor, a telefon automatikusan képes volt átkapcsolni két cella, illetve két bázisállomás között. Ezek a telefonok természetesen némiképp nagyobbak voltak a mai telefonoknál, ezeket jobbra fixen az autóba szerelhető kivitelben gyártották, csak később alakultak



*Martin Cooper (Motorola - 1973)
kezdeményezi az első kereskedelmi
mobiltelefonhívást az Egyesült
államokban (Kép: Rico Shen)*

ki az akkora méretű „hordozható” telefonok, az első valóban kézben hordozható mobiltelefonok pedig csak jóval később, szintén a Motorola jóvoltából jelentek meg. Amint látható, a Motorola úttörő szerepet vállalt a mobiltelefonok kialakulásában, illetve elterjedésükben, és ezt a vezető szerepét az amerikai piacokon ma is őrzi. Ezek a telefonok analóg átviteli szabványt használtak, többféle elnevezés is kialakult ezekre a technológiákra, általában gyártó, vagy kisebb eltérések alapján (NMT, AMPS, TACS...stb.), viszont közösen ezeket nevezzük első

generációs telefonoknak. Hogy hozzánk közeli példát hozzak, Magyarországon az analóg mobiltelefonok piaca nagyon keskeny volt, és elég rövid ideig virágzott, ha virágzott egyáltalán. Szolgáltató terén a Westel0660 elnevezésű, azóta már megszűnt vállalat volt egyedül fellelhető, ezen cég is csak 1-2 évig működött. Annyira gyorsan fejlődtek az 1G-s telefonok, és olyan hatalmas fejlettségi különbség volt a két rendszer között, hogy még a 450Mhz-es analóg telefonok elterjedése előtt kiszorították őket a piacról. Összehasonlításképpen elmondható, hogy az akkor legmodernebb és legújabb kiadású analóg telefon (NMT rendszerű Nokia 540-es típus) súlya 189 gramm volt és hatalmas külső antennájával együtt megközelítette a 20 centiméteres magasságot, addig a 2G-s Motorola V3690 súlya csupán 83 gramm volt, magassága pedig 10 centiméter, és nem tartozott az akkori legmodernebb telefonok közé.

Második generáció

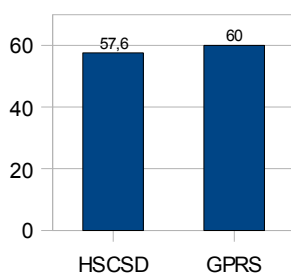
A kilencvenes évek elején jelentek meg a GSM, TDMA és CDMA szabványok, és velük a mobiltelefonok második generációja. GSM rövidítés egykor az Európai Posta és Távközlési Hivatalok (CEPT) által 1982-ben életre hívott Groupe Spécial Mobile munkacsoportjának nevét takarta, majd 1987-ben alakult ki belőle a mai is ismert Global System for Mobile Communications páneurópai szabvány. A TDMA (Time Division Multiple Access, IS-136) egy közeghozzáférési technológia neve, lehetővé teszi több felhasználónak ugyanazon frekvenciatartomány egyidejű használatát azáltal, hogy minden felhasználó adott időintervallumban továbbítja a saját adatát, így több adatframe is képes látszólag egy időben utazni. A CDMA (Code Division Multiple Access, IS-95) kódosztásos technológia, ami azon alapul, hogy azonos időben, azonos csatornán kommunikáló eszközök különböző kódokat használnak kommunikálásra, így csak az „azonos nyelveken beszélő” felek értik meg egymást.

A második generációs mobiltelefonok már digitális áramkör-kapcsolt átvitelt használtak az analóg első generációsokkal szemben. Mivel Európában előfordult, hogy ugyanazon frekvencián üzemeltek digitális 2G telefonok és analóg 1G telefonok, ezért ezen 1G-s rendszereket rendre megszüntették, hogy ne foglalják a 2G-s telefonoknak szánt sáv szélességet. Ez volt a sokáig szabványként használt 900Mhz-es sáv. Ekkor kezdtek eltűnni a piacról az óriási, úgynevezett téglatelefonok, és jelentek meg a 100-200g-os apró mobiltelefonok. Itt megjegyezném, hogy a magyar (és feltételezhetően egyéb) köztudatban viszonylag tévesen jelennek meg ezen 1G, 2G...stb. megnevezések. Sok helyen a 2G-s telefonokat 1G-snek jelölik, és csak azon telefonokat tekintik 2G-snek, amik már csomagkapcsolt átvitelre képesek. Ez egy téves gondolat, ugyanis a csomagkapcsolt átvitelt a 2G-s rendszerek kibővítéseként fejlesztették ki, mint egy átmenetet a 3G-hez, és itt alakultak ki a köznyelvben 2.5G és 2.75G néven nevezett „generációk”, ám fontos tudni, hogy míg az 1G, 2G és a 3G rendszerek alapelvei hivatalosan meg vannak határozva, és ezen alapelvek ilyen néven szerepelnek a szabványleírásokban, addig a 2.5G és 2.75G elnevezések csak

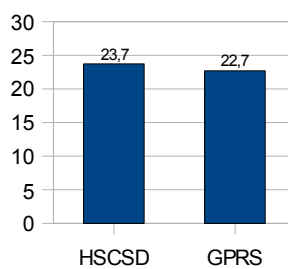
marketingcélből jöttek létre, megkülönböztetve a termékeket egymástól. Ezen szabványok lefektetése nem történt meg sehol hivatalosan.

Már említettük, hogy a 2.5G elnevezés a csomagkapcsolt adatátvitel megjelenésekor jött létre, így különböztették meg az erre képes eszközöket. A csomagkapcsolt adatátvitel a GPRS (General Packet Radio Service) megjelenésével terjedt el, ami az akkori viszonyokhoz képest gyors és olcsó adatátvitelt jelentett. Azért csak viszonylag, mert a GPRS maximális elvi átviteli sebessége 20kbit/s csatornánként, azaz 4 letöltési csatorna esetén 80kbit/s -ra képes, azonban hasonló sebességet már az előtte is létező HSCSD (High Speed Circuit-Switched Data) szabvány is biztosított, azonban míg a GPRS csomagkapcsolt, és ezzel egyidejűleg olcsó átvitelt biztosított, a HSCSD vonalkapcsolt volt, ráadásul általában az átlagos telefonhívás percdíjánál magasabb áron volt elérhető. Hogy prezentáljuk a GPRS elterjedésének hatalmas előnyét, nézzünk egy egyszerű számítást:

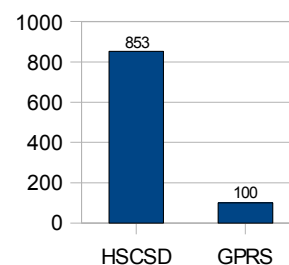
<i>Technológia:</i>	<i>Átviteli sebesség:</i>	<i>10Mb átviteli ideje:</i>	<i>10Mb átviteli költsége:</i>
HSCSD	57.6 kbit/s (4 slot)	23.7 perc	853 Ft
GPRS	60 kbit/s (3+2 slot)	22.7 perc	100 Ft



Átviteli sebesség



Átviteli idő



Átviteli költség

Amint látható a GPRS nem hozott óriási előrelépést az átviteli sebességben, ellenben hatása az átviteli költség csökkenésében volt erőteljesen érezhető. Ennek köszönhető a WAP (Wireless Application Protocol), illetve a mobil internet teret nyérése is, ami a dolgozat egyik fő témája lesz. Nem általánosan ismert tény, de az MMS (Multimedia

Message Service – Multimédiás üzenetküldés) szolgáltatás megjelenése is a GPRS-nek köszönhető, mivel ez biztosította az üzenet mérete alapján számlázott átalánydíj bevezetésének lehetőségét is. A 2.75G fantázianevű marketing-generáció az EGPRS (Enhanced GPRS) és az EDGE (Enhanced Data rates for GSM Evolution) átviteli szabványok megjelenésével jelent meg, melyeknek egyetlen céljuk az átviteli sebesség további növelése, és ezzel egyidejűleg természetesen az átviteli idő (tétlen várakozás) tovább csökkentése volt. Magyarországon kevésbé terjedtek el ezen technológiák, habár két magyar szolgáltató is bevezette ilyen jellegű elérését. Ennek oka az volt, hogy bár elvi átviteli sebességük valóban meglepően nagy volt (akár 236.8 kbit/s letöltési sávszélesség), ez ritkán volt teljes egészében elérhető, illetve ezen elérést főleg a vidéki előfizetők használták ki, mivel a nagyvárosokban hamar elterjedt a 3G gyors adatátviteli szolgáltatása, ami viszont mobil internethez valóban elég gyors volt, és használata a GPRS -hez képest nem került felárba (egyébként az EDGE -é sem).

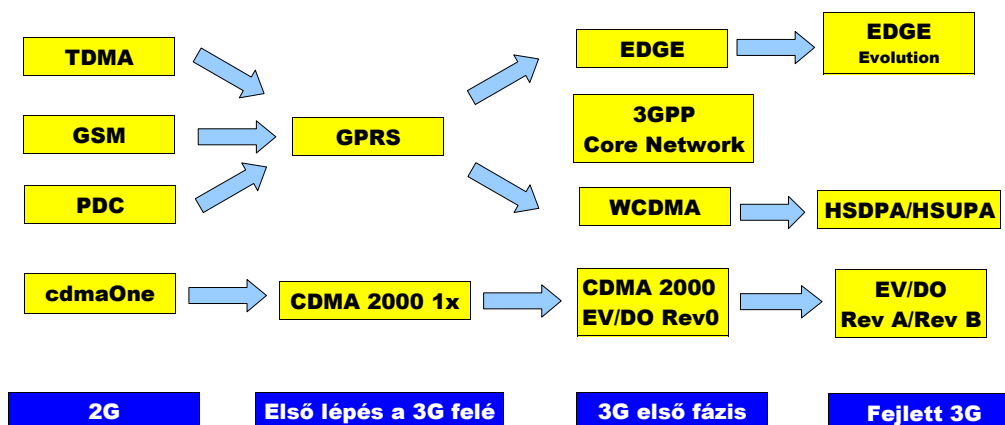
Harmadik generáció

A mobil eszközök harmadik generációjának kialakulása már nagyon röviddel a második generáció megjelenése után elkezdődött. Tulajdonképpen már a második generáción érződött a hatása, gondolok itt a 2.5 és a 2.75G megjelenésére, ami teljes egészében a 3G előszelének nevezhető. Ezen generáció már korántsem volt annyira szabványos, egyhangú, mint a második, ennek oka pedig az volt, hogy a 3G szabvány megfogalmazásakor nem a technológiát standardizálták, hanem az ezt megvalósító rendszer felé támasztott követelményeket foglalták írásba. Ezzel azonnal megdőlt egy világméretű szabvány kialakulásának lehetősége, és több különböző technológia kezdett megjelenni. Már az EGPRS, EDGE szabványok is eleget tettek a 3G által megfogalmazott követelmények nagy részének, viszont az elvárt, valóban nagy sebességű adatátvitel és a széleskörű multimédiás képességek megjelenésére még várnunk kellett a 3G megjelenéséig és elterjedéséig. Ez egyébként napjainkig is tart, teljes körű 3G szolgáltatás még csak a legnagyobb városokban van Magyarországon

(pl. Budapest, Debrecen). Elterjedésüket az is nagyban hátráltatja, hogy mivel a technológia egy új, eddig nem használt frekvenciatartományt használ, ezen új hálózat kiépítése, és az új frekvenciatartomány licencelése hatalmas költségeket ró a szolgáltatókra.

Európában az Egyesült Királyság és Olaszország volt az első, akik 3G hálózatot indítottak 2003 márciusában. 2007-re a 3G szolgáltatás előfizetői száma meghaladta a 200 milliót, ez azonban még mindig kevesebb, mint 7 százaléka a 3 milliárd mobiltelefon előfizetőnek, azonban azokban az országokban, ahol a szolgáltatás hamar elindult (mint pl. Japán és Dél Korea) ez az arány eléri az 50%-ot is.

A 3G hálózatok bevezetése több szempontból is hatalmas előrelépés volt. Ezek a hálózatok sokkal nagyobb előfizetőszámot tudnak kezelni, amely probléma főleg a sűrűn lakott nagyvárosokat érintette hátrányosan, illetve képesek voltak valóban széles sávú internetelérést biztosítani, aminek köszönhetően multimédiás szolgáltatások hada indult útjára ezen hálózatokon. Mivel a szolgáltatásokra még később kitérek, ezért most csak egyet említenék a sok közül, ami viszont összeforrt a 3G névvel, ez pedig az élő videó és hangátvitel, azaz a videótelefonálás. A 3G képes telefonokat elülső, általában kisebb felbontású kamerával is ellátják, aminek köszönhetően a hangunk mellett hívott fél láthatja az arcunkat is, és viszont. Innen már csak egy kis lépés az élő televízióadások, vagy hasonló, nagy átviteli sebességet igénylő szolgáltatások elterjedése mobiltelefonon, de ne ugorjunk ennyire előre.



Terjedésük

A mobiltelefonok terjedését nagyon egyszerű összefoglalni. A 80-as években a 0-ról indult szolgáltatás mára 3 milliárd előfizetővel rendelkezik, és növekvése töretlen, ma is napjaink egyik legmeghatározóbb üzletága a mobiltelefon, és minden azzal kapcsolatos iparág. Eleinte természetesen a gazdagabb rétegek kiváltsága volt, mivel mind a készülék, mind a szolgáltatás ára csillagászati nagyságrendű volt, de aztán 1998 környékén robbanásszerűen kezdett nőni a mobiltelefon előfizetők száma, aminek egy nagyon fontos oka volt: megszületett a Magyarországon csak „kártyás telefon” -nak nevezett szolgáltatás, azaz a prepaid mobile service. A szolgáltatást 1995-ben a portugál Telecomunicações Móveis Nacionais nevű szolgáltató találta ki eredetileg azért, hogy a plazákban vásárló népesség gyorsan, mindenféle papírmunka nélkül tudjon mobiltelefont venni. Lényege, hogy nem kell a hagyományos értelemben vett előfizetés a telefonhoz, azaz nincs fix havidíj, nincs közttség és hóvégi számla, ellenben pontosan olyan értékben vehető igénybe a szolgáltatás, amilyen értékű „kártyát” vásárol a tulajdonos előre. Ezzel akkor még nem is sejtették, hogy milyen lavinát indítanak el. A fiatal, vagy kevésbé tehetős réteg számára is megnyílt a mobilkommunikáció lehetősége, így a potenciális előfizetők számát megsokszorozták. A többi országban 1998 körül terjedtek el az ilyen jellegű szolgáltatások, ami okozta a fent említett robbanásszerű előfizetős szám növekedést. Európában jelenleg a mobiltelefon előfizetések 40%-át teszik ki az ilyen, úgynevezett prepaid szolgáltatások, de vannak olyan országok, mint például Mexikó, ahol ez a szám eléri a 90%-ot is. Ez a lehetőség azután gyors fejlődésnek indította a mobiltelefon-bűnözést is, amit ugye az tett lehetővé, hogy ezen szolgáltatások igénybevételéhez nem kellett szerződést aláírni, azaz megadni személyes adatainkat, és esetleg valamilyen hivatalos okmánnyal alátámasztani azok hitelességét, elegendő volt kifizetni a mobiltelefon-készüléket, illetve az alapsomag által tartalmazott szolgáltatás-mennyiség értékét, és már lehetett is telefonálni a készlettel. Így a hatóságok felismerve ezt, néhány éven belül a legtöbb országban bevezették, hogy valamiféle szerződéskötésre kötelezték az

ilyen szolgáltatás igénybevevőit is, azaz a telefontársaságok pontosan nyilvántartották az előfizető személyes adatait, amit bűncselekmény esetén a hatóságok rendelkezésére tudtak bocsátani. Ezzel persze pont a portugál szolgáltató eredeti céljait lehetetlenítették el, azaz komolyabb papírmunka kellett az ilyen előfizetésekhez is, de akkorra már nem is ez volt ezen szolgáltatások lényege, hanem az, hogy mindenki mobiltelefonhoz jusson, és a prepaid előfizetések ma is ennek köszönhetik töretlen népszerűségüket. Hogy a hagyományos, utólag fizető (azaz postpaid) előfizetések népszerűsége se csökkenjen, a szolgáltatók ügyeltek arra, hogy az ilyen szolgáltatások tarifái, percdíjai mindig a prepaid díjai alatt maradjanak, azaz ha az előfizető sokat használja a telefonját, akkor elérheti azt, hogy, habár fizetnie kell előfizetési díjat is, összességében még mindig kevesebbet fizet, mintha prepaid -re váltana. Ezen kívül eleinte néhány exkluzív szolgáltatást nem tettek elérhetővé prepaid kártyával rendelkező ügyfelek részére, de mára ez megváltozott. Így alakult ki az a hozzáállás, hogy a prepaid a diákok és fiatalok előfizetése, a postpaid pedig az idősebbeké.

Szolgáltatásaik

Kezdetben a mobiltelefonoknak egyetlen szolgáltatása volt, mégpedig a kétirányú full-duplex hangátvitel. Természetesen úgynevezett offline funkciókat már a régi telefonokba is integráltak, úgymint telefonkönyv, később csengőhang választás és hasonló, de ezeket a funkciókat a készülékgyártók valósították meg alkalmas hardver és szoftver segítségével. A valódi hálózati szolgáltatások még vártak magukra egészen 1992 december 3-ig, amikor Angliában a Vodafone hálózatán elküldésre került az első SMS (Short Message Service) üzenet. Eleinte nem volt nagy sikere az SMS-nek, 1995-ös felmérések szerint egy előfizető átlagosan 5 SMS-t küldött évente. Ebben közre játszott az is, hogy a szolgáltatás technikai háttere nem volt tökéletes. Ezen problémák kiküszöbölése után 2000-ben már több, mint 400 üzenetet küldött egy előfizető évente, 2007-re pedig csak New Yorkban, újév napján 9 millió üzenetet küldtek az emberek egymásnak. Az üzenetkezelési szolgáltatások forradalma tehát így

indult. A következő lépés az EMS volt, ami az SMS 160 karaktere mellett fekete-fehér kép átvitelére is képes volt, egy vagy több üzenet felhasználásával. A GPRS megjelenésével pedig elindult az MMS szolgáltatás, ami mára gyakorlatilag bármilyen fájlformátumot képes továbbítani, legyen az kép, videó, hang, vagy futtatható állomány. Az MMS terjedését természetesen az is nagyban segítette, hogy megjelentek, és nagyon gyorsan elterjedtek a fényképezni, videót rögzíteni tudó telefonok, amiknek segítségével gyorsan megoszthattuk élményeinket egymással.

Az adatátviteli funkciók körében a WAP-ra (Wireless Application Protocol) épülő szolgáltatások voltak az elsők, amik nagy sikernek örvendtek már bevezetésükkor is, de a GPRS és ezzel az olcsó internetelés megjelenésével tovább nőtt népszerűsége. A WAP és egy WAP képes böngésző segítségével az előfizető mobil eszközökre optimalizált (általában WML nyelven írt) weblapokat tekinthetett meg, azaz mobiltelefonjáról böngészhetett az Interneten, e-mailt olvashatott...stb., de mindezt sajnos azon megkötés mellett, hogy csak az erre alkalmas lapokat tekinthette meg. Ezt a megkötést szüntették meg a mobiltelefonokra fejlesztett internet böngészők, amik a WAP mellett már HTTP és egyéb protokollokon is képesek voltak működni, és cHTML, HTML lapokat is megjeleníteni.

Habár az üzenetkezeléshez tartozna szervesen, de a mobil internet megjelenéséhez is köze van, ezért külön említeném meg a telefonok e-mail képességeit. Már nagyon korán megjelentek a kezdetleges e-mail kliensek a telefonokon, amiknek előnye volt, hogy nem az e-mail szolgáltató webes elérését kellett használnia a felhasználónak, hanem az asztali számítógépek mintájára működő kliensprogrammal letölthette és később, akár offline is olvashatta leveleit. Kezdetben csak a POP3 protokollt támogatták, de gyorsan megjelent az IMAP és az SMTP, később a Windows Operációs rendszerrel szerelt telefonokon a MAPI támogatás is.

A mobil eszközök multimédiás képességei nem tartoznak ugyan a hálózati szolgáltatásokhoz, de népszerűségük indokolja, hogy pár szóban megemlítsük őket. A fejlődés talán a többszólamú csengőhangok megjelenésével kezdődött, aminek jelenlegi helyzete az MP3, WMA és hasonló zenei formátumokat lejátszani képes,

illetve beépített rádióvevővel rendelkező telefonokkal jellemezhető. Közben egy másik szálon induló fejlesztés első lépcsője a beépített digitális fényképezővel szerelt telefon volt, ahonnan egy rövid szoftveres fejlesztés után pillanatok alatt elértünk a videót rögzíteni képes telefonokig. Ezt egy gyors internet eléréssel kombinálva (ami akkorra már létezett is) kapjuk a videóhívás funkciót, ami Magyarországon ugyan még gyerekcipőben jár, de olyan országokban, ahol korábban jelent meg a szolgáltatás, már kialakult kultusza van. Emellett a széles sávú internetelérés az internetes-multimédiás szolgáltatások megjelenésének is kedvezett, mint az élő internetes televízió- és rádióadások. Amikor a mobiltelefon hardver elért egy elfogadható szintet teljesítményben, megjelenhettek a nagyobb számítási igényt kívánó szolgáltatások is, mint például a jó minőségű videó lejátszása (gondolok itt MPEG4 vagy a közkedvelt DivX formátumra), vagy az összetettebb grafikával rendelkező játékok futtatása.

A mobil eszközöknek ezen kívül manapság nagyon fontos funkciója a számítógéppel való szinkronizálás, összeköttetés. Olyannyira, hogy míg kezdetben a számítógép volt az, ami szolgáltatásokat tudott nyújtani a telefonnak (levelek szinkronizálása, archiválás...stb.), ma már megesk, hogy a számítógép szorul a telefon valamely funkciójára. Ilyen például a telefonoknak az a képessége, hogy mivel a modern telefonok már rendelkeznek valamilyen széles sávú interneteléréssel (lásd: HSDPA, WiFi), képesek ezen kapcsolatukat megosztani a hozzájuk kapcsolt számítógéppel. A számítógéppel való kapcsolódás történhet kábel segítségével, vagy vezeték nélkül, pl. infravörös, Bluetooth, vagy WiFi használatával is. Ezen vezeték nélküli kapcsolatok egyébként is jól jönnek akár két telefon, vagy telefon és egyéb eszköz (pl. headset, kihangosító, GPS vevő) összekapcsolásánál. A mobil internet annyira integrálva van már a modern telefonokba, hogy jelenléte néha alig vehető észre, mi csak a kényelmi funkció működését látjuk:



Az asztali Windows Media Playerből megszokott funkciót láthatjuk mobiltelefonon, azaz zene lejátszása közben beszerzi az internetről a zene adatait, és az album borítóját.

2. Mobil internetes alkalmazások

2.1 Fogalma, megvalósíthatósága

A mobil internetes alkalmazások definíciója egyszerűbb, mint amilyenek nevéből gondolnánk. Egyszerűen olyan, mobiltelefonokra írt programokról van szó, amik valamilyen interneten keresztül elérhető alkalmazáserverrel tartják a kapcsolatot, adatot cserélnek, stb. Gyakorlatilag a kapcsolat másik végén még alkalmazáserverre sincs szükség, mivel szerveren tárolt statikus adatfájlokból is olvashatnánk adatokat, illetve megfelelő jogosultsággal írhatnánk is oda, de ilyen jellegű szervert manapság nehezen találunk. A szerveroldal lehet bármilyen alkalmazáserver, ami képes olyan kimenetet generálni, ami egy mobiltelefon számára könnyen és gyorsan feldolgozható, vagy egyáltalán olyan, amire a mobiltelefonnak az adott pillanatban szüksége van. Lehet ez például XML, vagy properties fájl, adott esetben egy kép, vagy hangállomány, de ezen kívül még rengeteg formátum szóba jöhet, a lényeg, hogy az adott formátum az aktuális internetes tranzakció feltételeinek eleget tegyen.

Ahogy már szó volt róla, az átvitt formátumnak gyorsan letölthetőnek, gyorsan feldolgozhatónak, egyértelmű, jól strukturálnak, és mivel vezeték nélküli átvitelről

van szó, adott esetben (a felhasználástól függően) biztonságosnak is kell lennie. Mindezen követelmények egyszerűen megvalósíthatóak, és hogy példát is hozzak megvalósításukra, vizsgáljunk meg pár lehetőséget.

A gyors letölthetőségnek két feltétele van, az egyik a gyors internetkapcsolat, a másik az átvitt adat mennyiségének csökkentése. A mobiltelefon oldalt (azaz például a gyors letöltést) most nem vizsgálom, mivel az a következő pont témája lesz, maradjunk most szerveroldalon. A kis adatmennyiség legegyszerűbben kétféleképp valósítható meg, az egyik a jól megválasztott, tömör átviteli szerkezet (nevezzük ezeket innentől fájloknak), a másik pedig ezen fájlok tömörítése. Természetesen mindkét lehetőséget választhatjuk, de mivel a hálózati adatforgalom tömörítése jó néhány hálózatban már megvalósított funkció, illetve nem is ez a dolgozat fő témája, maradjunk a jól megválasztott, tömör fájlformátumnál. Én személy szerint az XML formátumot nagyon kedvelem, a szabvány egyből eleget tesz a strukturáltság és az egyértelműség követelményeknek, de vizsgáljuk meg a többit is. Feldolgozás szempontjából már nem ennyire rózsás a helyzet. A Java nagyon jó XML parser keretrendszereket tartalmaz, de ezeket a mobiltelefonokra szánt Java verziókból rendre kihagyták, így nem használhatóak. Az Interneten természetesen nem is egy ilyen, mobiltelefonokra írt XML feldolgozó csomagot lehet találni, és letölteni, de ezek felhasználása egy esetleg üzleti célra írt programnál már jogi kérdéseket is felvet. Megírhatjuk természetesen a saját XML csomagunkat is, az alap XML funkciók támogatását implementálni nem tart sokáig, de ha ezen követelmények teljesítése után megállunk, nem fejlesztjük tovább a parsert, akkor a szerveroldali, XML kimenetet generáló komponenst is úgy kell felkészíteni, hogy ne lépje át a feldolgozó korlátait. Teljes XML parser írása pedig nagyobb cégeknek is hetek, hónapok munkája lehet, nem biztos, hogy szánunk ennyi fejlesztői időt a rendszer egy ilyen kis részének implementálására.

Ezen kívül az XML ilyen rendszereknél történő alkalmazásának van még egy kis szépséghibája. Mivel az egyértelműséget és jól strukturáltságot úgy oldották meg, hogy szigorú szabályokat alkottak az XML formátumára, ez olykor nagy százaléku

többszörös adatátvitelét jelentheti. Ha egy egyszerű, pár karakteres adatot szeretnénk továbbítani XML formátumban, az a következőképpen néz ki:

```
<root>
  <element>
    adat
  </element>
</root>
```

Az egyszerű, 4 karakterből álló „adat” szó átviteléhez felhasználtunk 36 karaktert (formázás nélkül). Ez kilencszeres adatátvitelt jelent, de egy jól megtervezett adatátviteli folyamatnál természetesen nem jön ki ilyen sarkítottan.

Egy másik számításba vehető fájlformátum a properties fájl. Nagyon egyszerű formátum van, ráadásul a szabvány több lehetőséget is támogat. Formátuma:

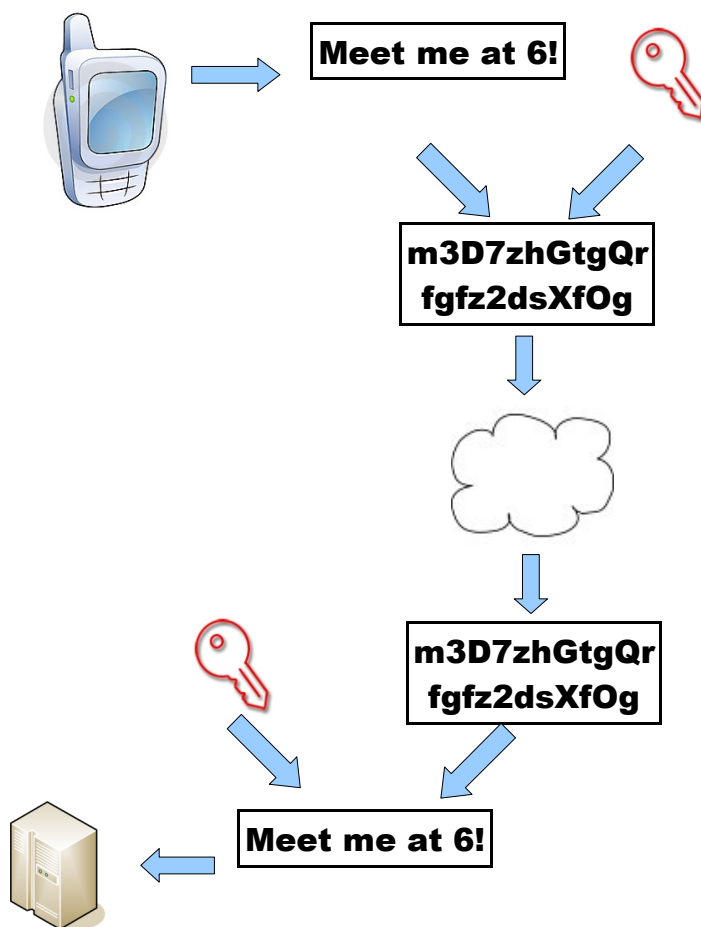
```
key1=value1
key2:value2
key3 = value3
```

Kulcs és érték párokból áll, amit választhatóan kettőspont vagy egyenlőségjel választ el egymástól. Az egyenlőségjeles változat elterjedtebb. Az úgynevezett töltelékkadat mennyisége sokkal kisebb, mint az XML-nél, bár kevésbé strukturált, és nem támogatja a sortöréseket az értékekben, így azt valami más módon jelölni kell. A feldolgozásához használt `java.util.Properties` osztály szintén kimaradt a J2ME-ből, de egy properties fájl beolvasását és feldolgozását végző rutin megírása középiskolai informatikaórán végzett gyakorlat is lehet.

Természetesen bármilyen más, akár általunk kitalált és implementált fájlformátumot is használhatunk, ha ezt feladatunk engedi, és/vagy indokolja, sőt, akár különböző technikákat keverhetünk össze, az adott feladat legjobb optimalizálása érdekében, de ez esetekben átláthatatlan kódot, struktúrát eredményezhet.

Az átviteli biztonság kérdésének tárgyalását szándékosan a végére hagytam, mivel az bármelyik formátum esetén azonos lehet. Mivel vezeték nélküli adatátvitelről van szó, az adat egy olyan átviteli közegben közlekedik, amihez mindenki tetszés szerint

hozzáfér. Az adatok biztonsága ilyen esetben kulcsfontosságú lehet, még akkor is, ha nem személyes adatainkat, vagy bankszámlaszámunkat közöljük az alkalmazáserverrel az éteren keresztül, hanem csak vicces üzeneteket küldünk, vagy moziműsort nézünk. Mint a fájlformátumra, a biztonság megvalósítására is több megoldást találhatunk, melyek változóan lehetnek jók, vagy gyengébbek is. Az egyik ilyen, és általam is kedvelt megoldás a privát kulcs alkalmazása mindkét oldalon. A lényeg, hogy az átvitt adat mindkét oldalon (szerver és kliens – mobiltelefon) kódolva lesz, és csak a privát kulcs segítségével lehet azt dekódolni és értelmezni.



Az azonos privát kulcs mindkét oldalon tárolva van, így használható mind kódolásra, mind dekódolásra. Az egyik fél üzenetküldésnél a küldendő üzenetet a kulcs segítségével kódolja, majd a kódolt üzenetet küldi el az interneten keresztül. A másik fél megkapja ezt a kódolt üzenetet, dekódolja a kulccsal, ami neki is meg van, majd az

így már értelmezhető üzenetet feldolgozza, majd ugyanilyen módon válaszol a címzettnek. A kliensen, azaz a mobiltelefonon egyszerűen tároljuk a privát kulcsot valamilyen konfigurációs állományban, a szerveren pedig az autentikáció után letárolásra kerül, hogy az adott klienssel milyen közös jellel (publikus kulccsal) azonosítják egymást, ami alapján a kliens fix privát kulcsa előkereshető az adatbázisból, és felhasználható kommunikációnál.

2.2 Jelentősége, létjogosultsága

Már láthattuk, hogy a biztonságos és hatékony mobil internetes alkalmazás megvalósítható. Most vizsgáljuk meg azt is, hogy van-e értelme ilyen dologba hosszú fejlesztői órákat és természetesen nem kevés pénzt ölni. Azt kell mondanom, hogy van. A mai mobil társadalom megkívánja ezek jelenlétét, a mobiltelefongyártók és szolgáltatók pedig elősegítik ezek fejlődését és terjedését. Az előbb elhangzott néhány technikai részlet, amit megvalósíthatóság szempontjából vizsgáltunk, értékeltünk, most nézzük meg, hogy hogyan segítik elő ezek megvalósítását a mobilgyártók, illetve szolgáltatók.

A mobiltelefon hardver folyamatos fejlődés alatt áll. Kezdetben, ahogy a bevezetőben is elhangzott, csak telefonálásra voltak alkalmasak ezek a készülékek, most már ott tartunk, hogy sok ember egy olyan teljesítményű hardvert hordoz a zsebében, ami 15 éve egy asztalon sem fért el, sőt nem is létezett nagyban sem. Én az első IBM kompatibilis számítógépet 1993-ban kaptam. Maga a gép egy 7 MHz-es processzorral és 1 MB memóriával szerelt számítógép volt, fekete-fehér monitorral, 20 MB háttértárolóval, és olyan súllyal, hogy 11 évesen álmodni sem mertem, hogy valaha felemelek olyan súlyt. Akkor átlagosnak számított. Most, amikor ezt a dolgot írom, a következő „konfigurációjú” mobiltelefon lapul az asztalom mellett: 220 MHz-es processzor, 64 MB memória, 128 MB beépített „háttértár”, ami tovább bővíthető 2 GB-ig, a súlya pedig 105 gramm. Most elkezdhetnénk számolgatni, hogy melyik érték hányszorosára nőtt, vagy hányadára csökkent, de szerintem felesleges, a számok magáért beszélnek. A mobiltelefonok manapság olyan számítási teljesítményt hordoznak magukban, ami pár éve még asztali gép formájában volt elérhető, és azokon az asztali gépeken gond nélkül futottak éppúgy a 3 dimenziós játékok, mint a webalkalmazások. Pedig néhány éve még az asztali gépeken sem volt elterjedt a széles sávú internet. A mobiltelefonok most szinte egytől egyig „szélessáv-képesen” kerülnek piacra, a szolgáltatók pedig fillérekért kínálják a gyors, vezeték nélküli interneteléréseket. Ezek mellett az adatátvitel gyors és egyszerű, a telefonok

hardvererősségei miatt pedig az átvitt adatok feldolgozása, kódolása, dekódolása pedig gyerekjáték.

A hardver és a hálózati háttér megvan tehát az ilyen alkalmazásokhoz, de vajon az emberek is igénylik? Ez attól függ, hol nézzük. Azokban az országokban, ahol korán bevezetésre került a gyors és olcsó mobil internet elérés, már egész nagy tábor használja, akár mindennapi kommunikációjához ezt a módszert. Ott viszont, ahol késett ennek bevezetése (Magyarországon is), még igencsak gyerekcipőben jár, és sokszor idegenkedve nézik azt is, akiről kiderül, hogy ilyet használ. Pedig sokkal egyszerűbbé és olcsóbbá teheti életünket. Vegyük előbb az egyik oldalt, az olcsóságot: Ma az egyik vezető hazai mobilszolgáltatónál 2000 Ft-ért 200 MB-os internet hozzáférést vehetünk. Ez pontosan 209715200 byte adat. Egy SMS 160 karaktert tartalmazhat, ami 1 byte-os karakter kódolással 160 byte, plusz vegyük a címzett megjelölését, plusz egyéb, az üzenet elküldéséhez használt hálózati adatforgalmat, majd kerekítsünk, így megkapjuk, hogy optimális adatátvitellel 300 byte-ból könnyedén kihozható egy SMS elküldése. Ez az előbb említett mobil internet hozzáféréssel 699050-szer tehető meg. Ennyi SMS elküldése egy átlagos, mondjuk 20 Ft/SMS áron 13.981.013 Ft lenne a hagyományos módon. De egy havidíj nélküli mobil internet elérés esetén is 1,2 Ft fizetendő 10 kB-onként, ami így is több, mint 30 SMS elküldését tenné lehetővé. Ezek természetesen csak elméleti értékek, az eredmények nagyban függhetnek az adatátvitel optimalizáltságán és a felhasználás módján.

Ha az életünk megkönnyítését tűzzük ki célul, azt kell mondanom, ebben is számottevő szerepet játszhat a mobil internet technológia. Az ok a nevének egyik szavában keresendő: mobil. Mindig nálunk lehet, bárhol elérhető, alkalmas szoftver vagy mobilképes weboldal segítségével autóból, karosszékből intézhetjük mindennapi teendőinket anélkül, hogy egy nagy, nehezen hordozható számítógép vagy laptop segítségére kényszerülnénk.

3. Egy mobil internetes alkalmazás

3.1 iSMS – az ötlet és a fejlesztés lépései

Először is el kell mondanom, hogy az ötlet nem új. Már több neves szoftvergyártó megjelent ezen a piacon, így részt találni rajta nagyon nehéz, majdnem lehetetlen. Mégis azt kell mondjam, hogy van egyediség az ötletben, ami azt eredményezi, hogy ilyen programot elterjedve máshol még nem nagyon láttam. Az ötlet egyszerű, és az előzőekben már említettem is. Használjuk ki a mobil internet adta lehetőségeket a mobilszámlánk csökkentésére. Az internetes telefonálás nem új dolog, és igazából akkora téma, hogy arról is meg lehetne tölteni egy vastagabb könyvet. Maradjunk most azoknál az embereknél, akik sok ezer forintot kifizetnek havonta SMS küldésre. Régen én is ilyen voltam, és elmondhatom, hogy nem nehéz ilyen számlát csinálni SMS-ből. Napi 3 üzenet, máris 2-3000 Ft költséget eredményez. Jogos hát az igény, hogy miért ne használjuk ki azt, hogy havidíj nélküli mobilinternet elérés esetén 10 kB adatot 1,2 Ft-ért küldhetünk. Ahogy számoltuk, egy SMS 300, de maximum 350 byte adatból áll, ez 30-35 üzenetet jelent 1,2 Ft-onként. Kellően olcsó. Mondhatnánk, hogy ha az ember már úgymint internetezik, miért ne írjon e-mailt? Minden telefon tud már e-mailt kezelni, küldeni, külön program sem kell már hozzá, és ugyan úgy kihasználja az internet lehetőségeit. Azért ez nem egészen így van. Pár éve, amikor megjelent az MMS szolgáltatás, a díjazás emlékeim szerint úgy alakult, hogy 32 Ft volt egy maximum 160 karakteres SMS, és 55 Ft volt egy MMS, amibe viszont több ezer karaktert küldhettünk. Nem volt nagy különbség az árban, de ár/érték arányban számottevő volt, mégsem terjedt el soha annyira, mint az SMS. Az embereknek egyszerűen az SMS szolgáltatásra volt szüksége, annak minden korlátjával, és előnyével együtt. Az is közre játszhatott ebben, hogy 160 karakterben el tudunk küldeni egy érzést, vagy gondolatot, de a mobiltelefon billentyűzetét használva senkinek nem volt kedve több ezer karakteres üzeneteket küldeni. Később az SMS konkaténációjának lehetősége sem váltotta be a neki fűzött reményeket, mindenki kínosan ügyelt arra, hogy ne lépje át az

1 SMS határt, bár ez gondolom nagyrészt pénzkérdés is volt. Ezen kívül az is besegített az SMS terjedésébe, hogy ezek az üzenetek gyakorlatilag offline üzenetek voltak, azaz az előfizető bármikor kaphatott SMS-t bárkitől, ő akkor olvasta el, és válaszolta meg azt, amikor volt ideje rá, nem kellett mindig telefonközelben lennie, nem várt el azonnali interaktivitást, mint például a manapság asztali gépeken nagyon elterjedt MSN Messenger és hasonlóak. Ezek az úgynevezett azonnali üzenetküldő szolgáltatások egyébként szintén erősen terjednek mobiltelefonon is, sok IM kliens van már több mobil platformra is, de ezek is akkor használhatóak ki igazán, ha az ember real-time üzenetküldésre használja, azaz állandó, online kapcsolata van, és azonnal reagál az üzenetre.

Próbáljuk meg akkor összeszedni a rendszer felé támasztott követelményeinket. Az alap tehát adott, legyen egy olyan üzenetkezelő rendszer, ami üzenetküldésre és fogadásra alkalmas, az elküldött üzeneteket pedig tárolja egy központi szerver, amennyiben a címzett felhasználó nem érhető el. Ne legyen szükség az előfizető állandó kapcsolódására, tehát következő kapcsolódáskor letölthetőek legyenek az offline idő alatt kapott üzenetek. Az üzenetek és egyéb adatok titkosítva utazzanak a hálózaton, a rendszer tehát igényeljen regisztrációt, amikor az előfizető számára generálódik egy privát, és egy publikus kulcs. A kliensprogram tehát rendelkezzen egy tetszetős felhasználói felülettel, ahol minden beállítás elvégezhető, és az üzenetek kezelése egyszerű és gyors. Ezek az alapkoncepciók, nyilván ezeket később pontosítani kell. Még van valami, amit előre meg kell fogalmazni, ez pedig fejlesztői, és futtató környezet. A fejlesztői környezet Netbeans 5.5, a Mobility Pack kiegészítővel karöltve, a futtató és tesztkörnyezet pedig szerveroldalon Jboss alkalmazáserver és Oracle 10g Express Edition adatbázis, a kliens oldalon pedig a Netbeans Mobility pack beépített emulátora lesz.

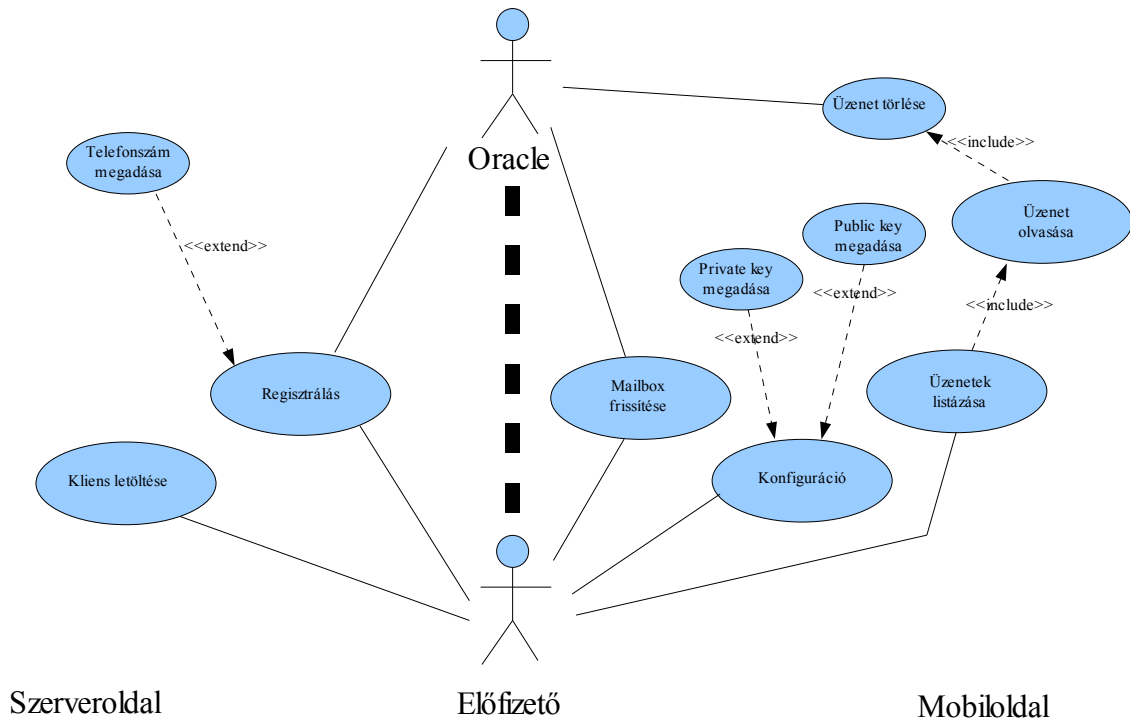
Tervezés

Foglaljuk össze tehát, hogy a rendszert milyen lépések során fogjuk használni, hogy képet kapjunk a megvalósítandó rendszer funkcióiról:

1. Az előfizetőnek a használat előtt a rendszer regisztrációs weboldalán regisztrálnia kell, ahol a mobiltelefonszáma és e-mail címe megadásával létrehoz egy postafiókot, és megkapja a rendszer által generált privát és publikus kulcsát, amit a rendszer el is küld a megadott e-mail címre.
 - Szükség lesz tehát egy űrlapra, ahol az előfizető megadja az adatait.
 - A rendszernek két egyedi kulcsot kell generálnia.
 - Létre kell hozni az előfizetőhöz tartozó bejegyzéseket az adatbázisban.
2. Szintén az adott weboldalról letölthető a mobiltelefonra feltelepítendő kliensprogram.
 - Egy egyszerű linkkel megoldható, programozói szempontból nem fontos szempont.
3. A letöltött alkalmazást telepíteni kell a mobiltelefonra, használat előtt.
 - Programozási szempontból ez sem lényeges pont.
4. A feltelepített kliensprogramban a privát és publikus kulcsok megadása
 - Szükség van egy konfigurációs lapra, ahol az előfizető megteheti ezt.
5. Üzenetkezelés (postafiók frissítése, olvasás, törlés)
 - A postafiók aktív internetkapcsolat esetén frissíthető, a letöltött üzenetek nem vesztettek el a telefonról lekapcsolás után sem, mivel letöltés után a szerverről törlődnek, ezért perzisztensen meg kell oldani tárolásukat.
 - Szükség van egy olyan nézetre, ahol a letöltött üzenetek ki vannak listázva, kiválaszthatóak.
 - Egy üzenet kiválasztása után az olvashatóvá válik.

- Egy megnyitott üzenet szükség esetén törölhető, ami az adatbázisban is végleges törlést eredményez.

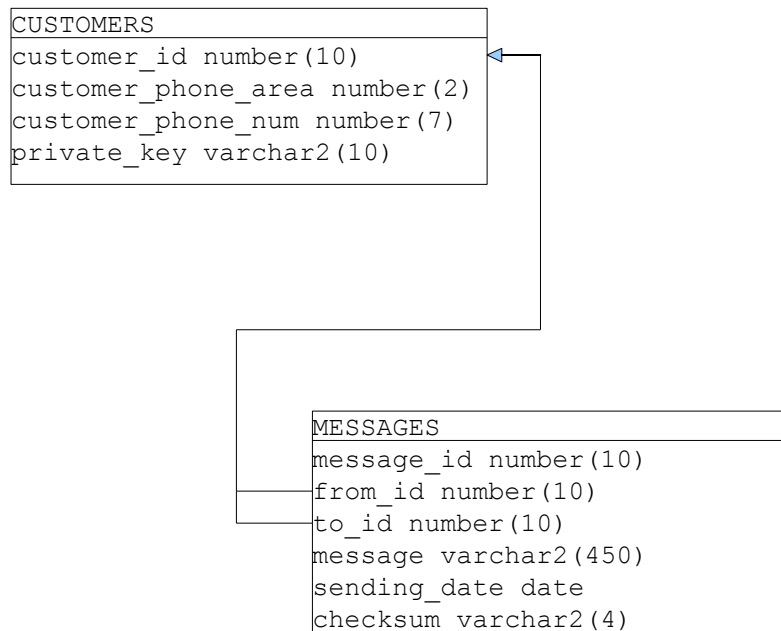
A rendszernek nincs szüksége az előfizetőről a mobiltelefonszámán kívül másik azonosítóra, mivel az mindenkinek egyedi. A mobilszám jogos tulajdonának vizsgálatát a rendszer jelenleg nem teszi meg, de éles helyzetben nyilván szükséges lehet. Megerősítő SMS-el, vagy közvetlen kapcsolattal a szolgáltató szerveréhez ez is megoldható lenne. Az előbb összegyűjtött információk alapján már elkészíthetjük a „use case” diagrammot:



Amint látható, a rendszer nem túl bonyolult, a szerveroldal egészen egyszerű, kevés komponense van. Ami a use case-en nem látszik, kell majd a szerveroldalra néhány servlet, amiket interfészként fogunk használni a kommunikáció során, valamint a kódoló/dekódoló modulok. A kliens oldal sem túl bonyolult, a nehezebb része talán az optimális adattárolás lesz, tudván, hogy a Java Micro Edition perzisztens adattárolása a RecordStore osztály véges képességeiben kimerül.

Szerveroldal

Kezdjük talán a szerveroldallal, mert az a kliens nélkül is tesztelhető. Az adatbázis Oracle, a táblastruktúra nem lesz túl bonyolult, mivel kevés adatot kell nyilvántartanunk. Mindössze egy CUSTOMERS táblára van szükségünk, ami az előfizetők adatait tartalmazza (azonosító, amit publikus kulcsként is használhatunk, mobilszám, illetve a privát kulcs), valamint egy MESSAGES tábla, ami pedig az elküldött, de a címzett által még le nem töltött üzenetek tárolja (üzenet azonosító, a küldő azonosítója, a címzett azonosítója, az üzenet, a küldés időpontja, illetve egy checksum-ot, amire később, a hitelesítésnél lesz szükség).



Az adatbázisoldalon megvalósítandó funkciók PL/SQL-ben lesznek programozva, mivel az használja ki jól az Oracle és az EJB képességeit. Egy rövid lista az adatbázisban megvalósított funkciókról (a teljesség igény nélkül):

- ```
function isCustomerExists(
 area_code CUSTOMERS.customer_phone_area%type,
 phone_num CUSTOMERS.customer_phone_num%type)
return boolean;
```

- Megvizsgálja, hogy létezik-e már az előfizető.
- ```
function registerCustomer(
    area_code CUSTOMERS.cutomer_phone_area%type,
    phone_num CUSTOMERS.customer_phone_num%type,
    private_key CUSTOMERS.private_key%type)
return number;
```

 - Regisztrál egy új előfizetőt, és visszaadja a hozzárendelt azonosítót.
- ```
function getMessagesForCustomer(
 cust_id MESSAGES.to_id%type)
return isms_cursor;
```

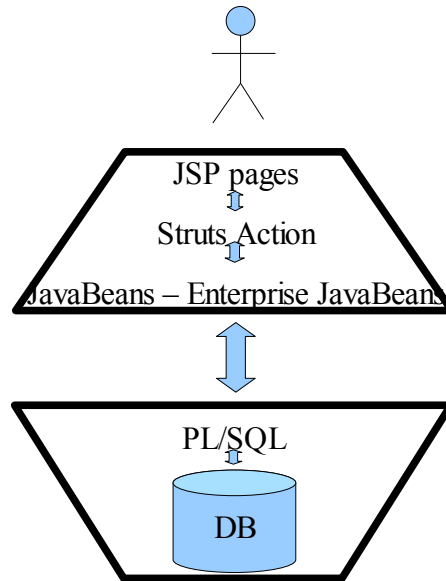
  - Visszatér az adott előfizető összes, eddig le nem töltött üzenetével. Az isms\_cursor egy gyenge ref cursor típus.
- ```
procedure addMessage(
    from_id MESSAGES.from_id%type,
    to_id MESSAGES.to_id%type,
    message MESSAGES.message%type,
    checksum MESSAGES.checksum%type);
```

 - Új üzenetet hoz létre a megadott adatokból. Az azonosító egy szekvenciából érkezik, a dátum pedig az SQL sysdate-jéből.
- ```
procedure deleteMessage(
 message_id MESSAGES.message_id%type);
```

  - Letöröl egy üzenetet. Akkor hívódik meg, mikor az előfizető sikeresen letöltötte.
- ```
function getPrivateKeyForCustomer(
    cust_id CUSTOMERS.customer_id%type)
return varchar2;
```

 - Visszatér az előfizetőhöz tartozó privát kulccsal üzenet kódolásához, vagy dekódolásához.

Az alkalmazás az MVC tervezési mintát követi. A megjelenítés JSP lapokkal történik, a controller réteg a Struts frameworkben használatos Action osztályokkal, a modell réteg pedig JavaBeans komponensekkel valósul meg. Az adatbázist egy ráépített EJB rétegen keresztül érjük el.



A webes felület csak a postafiók üzembehelyezéséhez szükséges funkciókat látja el, a rendszer lényegi részét a szerveren háttérben futó Java EE alkalmazás és a mobilon futó Java ME kliensalkalmazás végzi. A szakdolgozat témája a mobiltelefonok programozása, ezért a szerveroldal működésén csak gyorsan végigfutok, néhol képekben, néhol vázlatos magyarázatokkal.

Amint már elhangzott, a postafiók egy egyszerű regisztrálással létrejön, ehhez csak a telefonszám megadása szükséges, az e-mail cím megadása opcionális, erre el lesz küldve a generált privát kulcs. A kulcs e-mail cím hiányában a weblapról is lementhető.

Regisztráció

Mobiltelefon körzetszám (2 jegyű):

Mobiltelefonszám (7 jegyű):

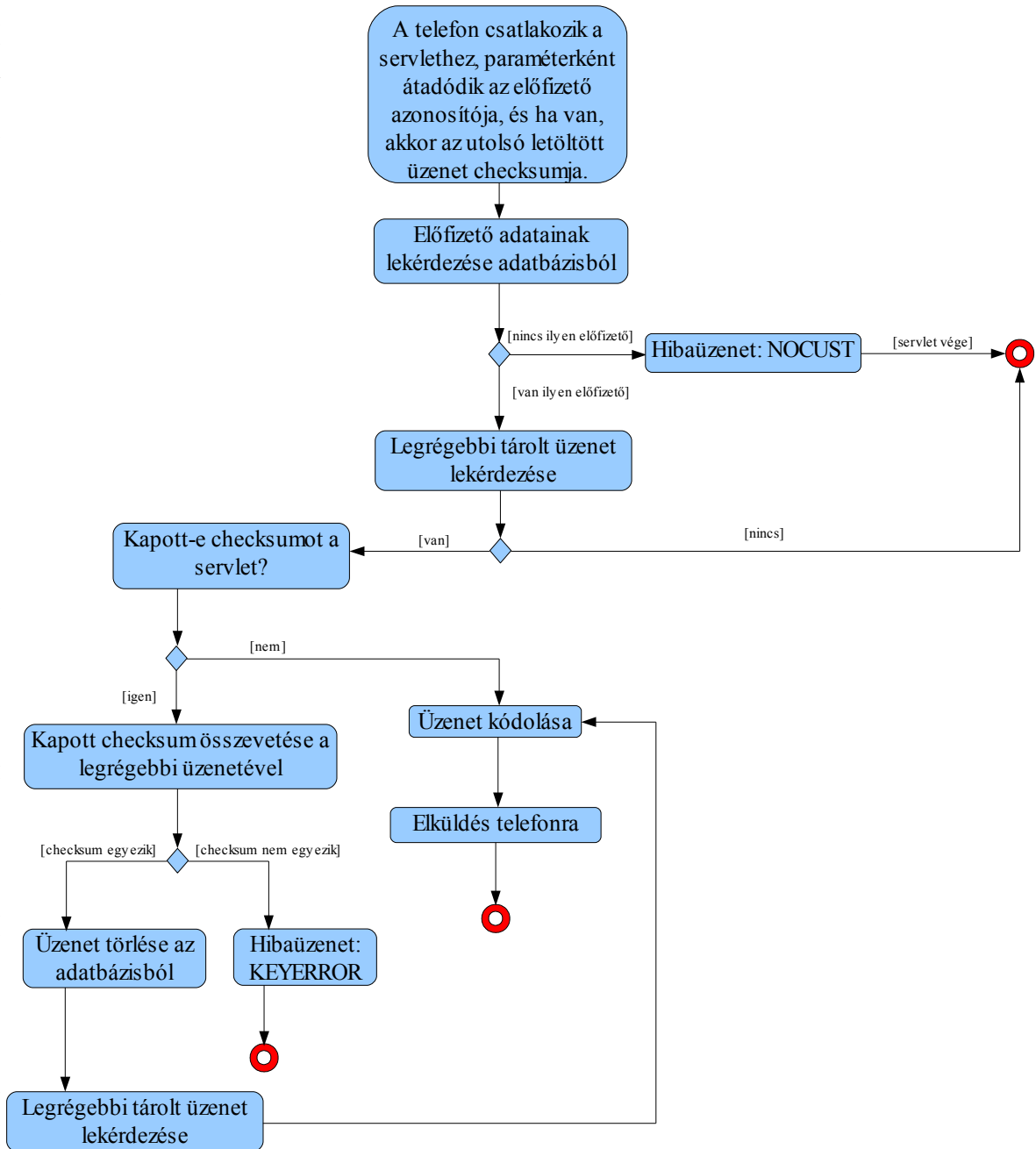
E-mail cím (a kulcs ide kerül elküldésre):

Privát kulcs: hz5Te4FD4v

Azonosító: 5234

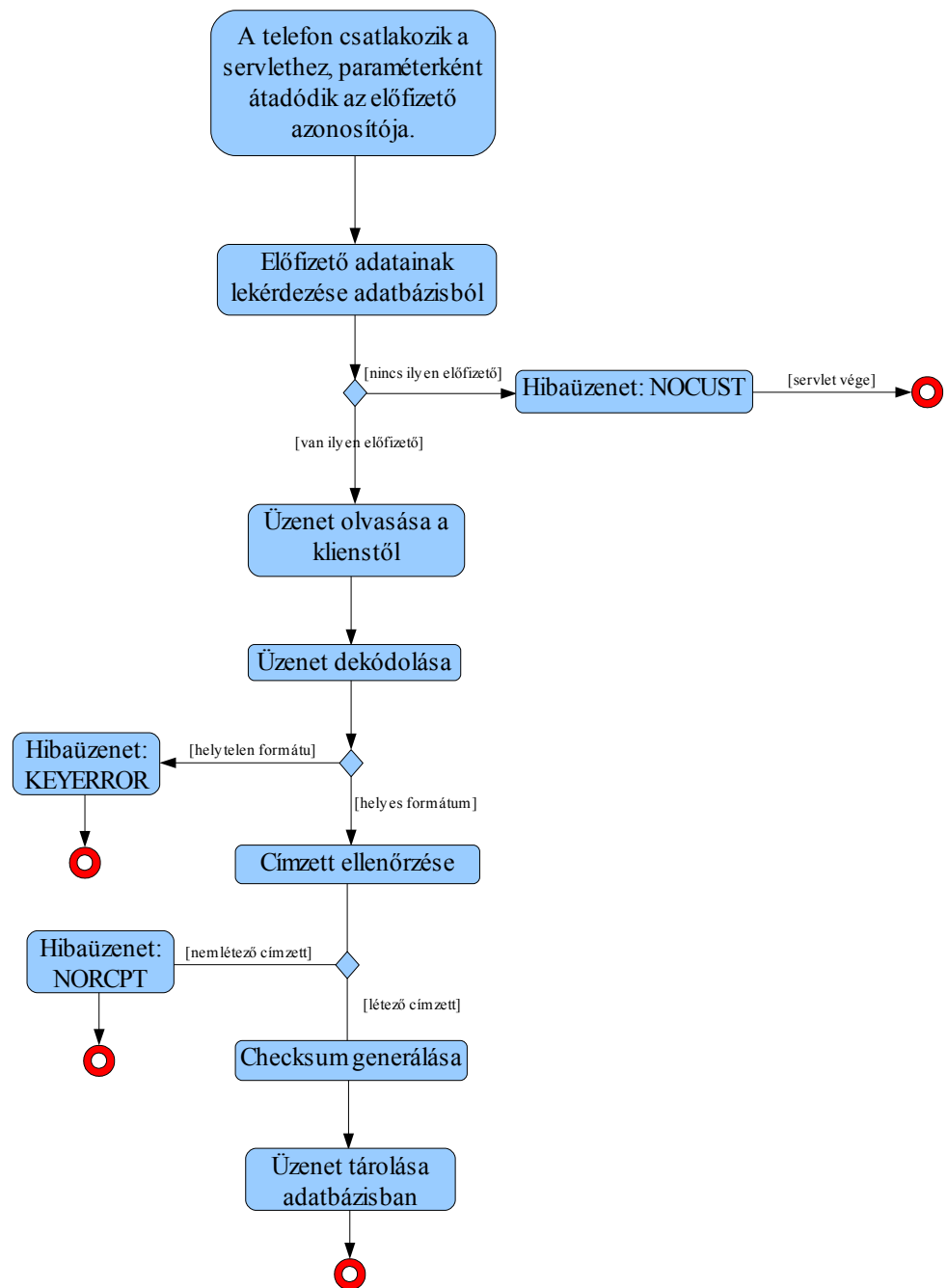
A regisztráció eredménye az adatbázis CUSTOMERS táblájában egy új sor, ami egy azonosítót, az előfizető mobilszámát és a generált privát kulcsot tartalmazza. Innentől a szerep a háttérben és a mobiltelefonon futó alkalmazásé:

SynchroniserServlet
(Az üzenetek szinkronizálását végzi)



Nézzük meg az üzenetküldő komponens működését is:

MessageSenderServlet
(Az üzenetek fogadását, tárolását végzi)



A diagram eleje megegyezik az előzővel, így azt nem tárgyaljuk, az érdekesebb részek az üzenet dekódolásánál kezdődnek. Mivel itt sem használunk hitelesítést, az üzenet validálása az alapján történik, hogy az előfizető privát kulcsával való dekódolás

után érvényes üzenetformátumot kapunk-e. Az üzenetformátum ellenőrzése úgy történik, hogy mivel az üzenet egy karakterlánc formájában utazik, a különböző mezőket úgy különböztetjük meg egymástól, hogy az első 9 karakter a címzett mobiltelefonszáma, a többi az üzenet. Ha egy rossz privát kulccsal történik az üzenet dekódolása, elég kicsi az esély arra hogy helyes formátumot kapunk a dekódolás után. Ha pedig ez mégis megtörténne, akkor sem lehetséges már nevében a hamis üzenetküldés, mert a címzett valószínűleg csak értelmetlen karaktersorozatot kapna üzenet helyett. Ha a címzettet nem találja az adatbázisban, „NORCPT” üzenetet küld, a telefon pedig hibaüzenetet ad, miszerint „Nem létező címzett”.

Mobiloldal (kliensoldal)

Azt már összeszedtük, milyen funkciók kellenek a mobiloldalra. Nézzük most meg ezeket a funkciókat közelebbről. Lenni kell egy konfigurációs lapnak, ahol az előfizető beállíthatja az előfizető azonosítóját, illetve a privát kulcsát, amit a regisztráció során kapott. Ezen kívül csak egy üzenetek menüpont kell, ahol az összes, az üzenetkezeléssel kapcsolatos funkció elérhető.

A JavaME MIDlet osztályokat használ programok reprezentációjáta. Egy MIDlet egy olyan osztály, ami kiterjeszti a `javax.microedition.midlet.MIDlet` absztrakt osztályt, és implementálja a `startApp()`, `pauseApp()` és `destroyApp()` metódusokat, amik hasonlóak a `java.applet.Applet` osztály `start()`, `stop()`, és `destroy()` metódusainak működéséhez. Ezen kívül egy MIDlet egyéb közönséges Java osztályokat is tartalmazhat, amiket együtt lefordítva és `.jar` fájlba csomagolva kapjuk az úgynevezett MIDlet csomagot. Tovább bővítve a lehetőségeket, egy `.jar` MIDlet csomag akár több MIDlet-et is tartalmazhat, amik megosztva használhatják a `.jar` fájlba csomagolt többi erőforrást, osztályt...stb. Egy MIDlet-nek három állapota lehet: `active`, `paused` vagy `destroyed`, azaz aktív, felfüggesztett vagy befejezett. Az aktív állapot azt jelzi, hogy a MIDlet elindult, és fut,

azaz a benne implementált funkciók működésben vannak. Ez az állapot a JVM által meghívott `startApp()` metódus elindulásával veszi kezdetét. Felfüggesztett állapotban (amit a `notifyPaused()` metódus kezdeményez) minden, az addig a MIDlet-hez társított erőforrás felszabadul, viszont a program készen áll azok azonnali, újbóli lefoglalására. A `notifyDestroyed()` metódus hatására a MIDlet destroyed állapotba kerül, véglegesen felszabadítja az összes, általa lefoglalt erőforrást, és a GarbageCollector aktiválására vár. Egy MIDlet osztály létrehozásánál az előbb említett Applet osztályhoz hasonlóan (bár ezt a hasonlóságot nem lehet túlzottan erőltetni) lehet komponenseket tenni a telefon grafikus kijelzőjére. Például létrehozhatunk egy form-ot, és arra komponenseket pakolhatunk, majd a form-ot beállítjuk aktuális képernyőnek:

(A minden programozó szívét megdobogtató Hello World példa mobiltelefonra)

```
//importáljuk, amire szükségünk lesz
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;

//hozzuk létre a HelloWorld nevű MIDlet osztályt, kiterjesztve az előbb
//említett MIDlet absztrakt osztályt (a CommandListener osztály az
//események kezelésére szolgál)
public class HelloWorld extends MIDlet implements CommandListener{
//a Form osztálybeli form adattagunk
private Form form;
//HelloWorld konstruktor
public HelloWorld(){
//form példány
form = new Form("Hello World MIDlet");
//statikus szöveg felvitele a formra
form.append("Hello World!");
//kilépés gomb hozzáadása
form.addCommand( new Command( "Exit", Command.EXIT, 1 ) );
```

```

        //CommandListener regisztrálása, hogy a kilépés gomb működjön is
        form.setCommandListener( this );
    }
    //a MIDlet indításához szükséges metódus
    public void startApp()
    {
        //kijelző-hivatkozás lekérése
        Display display = Display.getDisplay(this);
        //aktuális kijelző megadása
        display.setCurrent( form );
    }
    //a pauseApp metódusra most nincs szükségünk, de implementálni kell
    //mindig, mivel absztrakt metódus
    public void pauseApp() { }

    //a destroyApp metódus törli a form referenciát a MIDlet befejezésekor.
    //Erről bővebben egy későbbi fejezetben
    public void destroyApp(boolean unconditional)
    {
        form = null;
    }
    //A CommandListener interface implementálásához egy metódust kell
    //implementálnunk, ez pedig a commandAction
    public void commandAction(Command c, Displayable d)
    {
        //befejezi a MIDlet futását
        destroyApp(true);
        //értesíti az alkalmazáskezelőt, hogy a MIDlet befejeződött
        notifyDestroyed();
    }
}

```

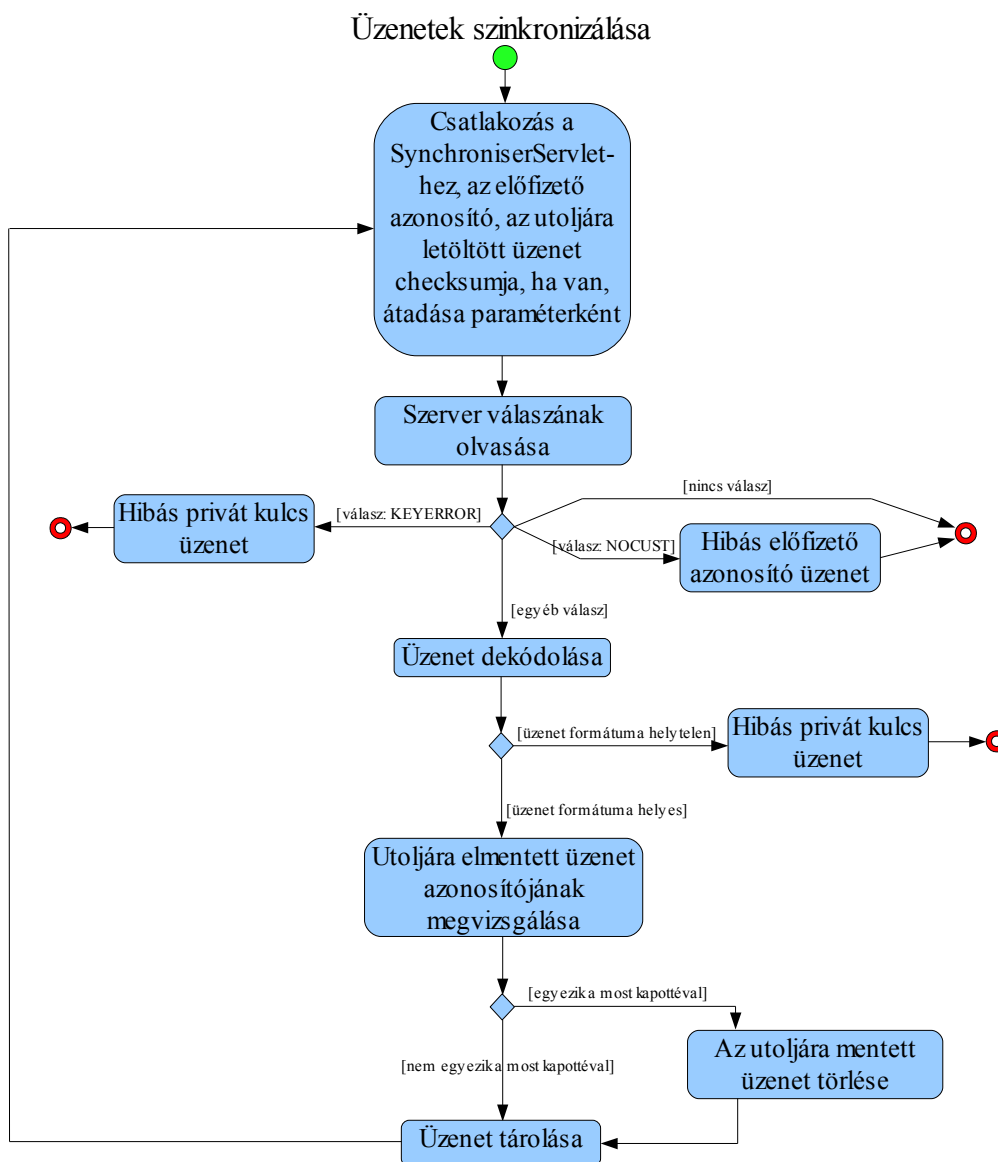
Most, hogy láttuk az alapjait egy mobil program felépítésének, nézzük, jelen esetben mire lesz szükségünk. A konfigurációs oldal két mezőt kell tartalmazzon, az egyikkel az előfizetői azonosítót tároljuk (amit a korábbiak szerint publikus kulcsként használunk), a másikkal a privát kulcsot. A perzisztens tárolás nem csak az üzenetekre

vonatkozik, azaz a beállított azonosítót, és a kulcsot is le kell tárolnunk. Nézzük meg tehát a J2ME perzisztens tárolóját, azaz a `RecordStore` osztályt. Ez az osztály pontosan azért jött létre, hogy adatot tudjunk tárolni Java oldalról úgy, hogy az túléljen bármilyen kikapcsolást, akkumulátor lemerülést, stb. Létrehozása már egy név megadásával is lehetséges, de további paraméterként átadhatunk neki olyan tulajdonságokat, mint autorizációs mód (melyik MIDlet suite férhet hozzá a store-hoz), vagy más MIDlet suite-ok ezen store-ba való írásának engedélyezése, vagy sem. Szerkezetéről elég annyit tudni, hogy egy-egy byte-tömböt képes tárolni egy rekordban, a rekordoknak egyedi azonosítójuk van, aminek segítségével később közvetlenül hozzáférhetünk a tárolt adatunkhoz, de enumerátor segítségével is bejárhatjuk, a szükséges adatot keresve.

Konfiguráció elmentése:

```
javax.microedition.rms.RecordStore configStore =  
    javax.microedition.rms.RecordStore.openRecordStore(configStoreName, true);  
configStore.addRecord(customerId.getBytes(), 0, 0);  
configStore.addRecord(privateKey.getBytes(), 0, 0);
```

Az üzeneteket listázó ablak a `javax.microedition.lcdui.List` osztály egy példánya, a listaelemek feliratát az üzenetek adataiból hozzuk létre konkaténációval. Alulra mindig kiteszünk egy extra választható sort, az lesz a „Frissítés...” gomb, a szerveren lévő üzenetek letöltésére. Működése:



A tárolás módját már megnéztük a konfiguráció elmentésénél, nézzük meg most a szerverkapcsolat nyitásának módját. A J2ME-ben a

`javax.microedition.io.Connector` osztállyal lehet a J2SE-ben lévő `URLConnection`-höz hasonló kapcsolatot nyitni. A kapcsolat

```

javax.microedition.io.Connection conn =
    javax.microedition.io.Connector.open(url);
  
```

szintaxissal jön létre, majd ezután ezt a példányt lehet cast-olni

`javax.microedition.io.HttpConnection` típusúvá:

```

javax.microedition.io.HttpConnection httpConn =
    (javax.microedition.io.HttpConnection) conn;

```

Ezután lehet OutputStream-et nyitni a kapcsolatra, majd ennek lezárása után

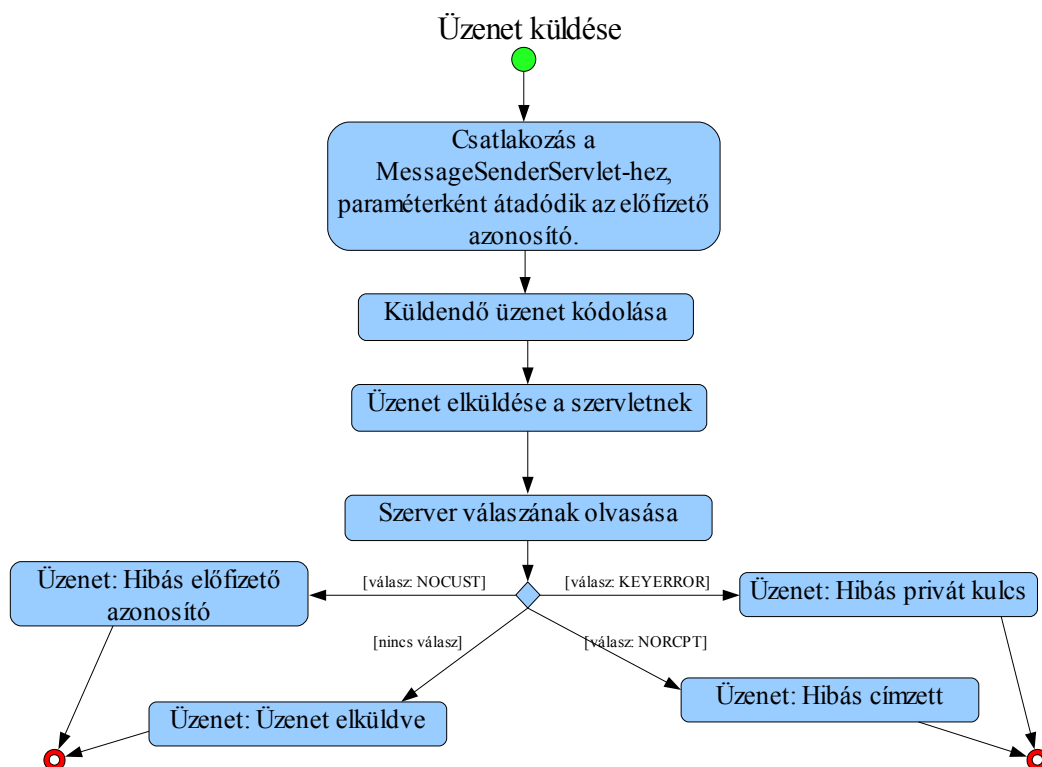
InputStream segítségével olvasható a szerver válasza:

```

java.io.OutputStream os = httpConn.getOutputStream();
//itt írható a kimenet
//pl.: os.write(„Hello World”.getBytes());
//majd le kell zárni
os.close();
//ezután nyitható InputStream
java.io.InputStream is = httpConn.openInputStream();
//a válasz ezután olvasható
//pl.: int readed = is.read();
//majd az olvasás befejeztével ezt is le kell zárni
is.close();
//majd illő a kapcsolatot is
httpConn.close();

```

Adós vagyok még az üzenetküldés aktivitásdiagramjával, így hát ezt is nézzük még meg:



Ez már viszonylag egyszerű. A függelékben ezen kívül megtalálhatóak a kliensalkalmazás legfontosabb kijelzőképei.

3.2 Jó mobil program

Végezetül nézzünk meg néhány direktívát arra vonatkozóan, hogy milyen alapvető szempontokat kell szem előtt tartani annak érdekében, hogy egyszerű, átlátható és gyorsan futó alkalmazásokat fejlesszünk mobil eszközökre. Már tervezéskor elő kell venni néhány tételt ezek közül, mert sok szempont már ott eldőlt. Igyekezzünk egyszerű alkalmazást tervezni, hagyjunk ki minden fölösleges funkciót. Ha egy nagyobb funkciócsoportot kihagyunk, ott már érdemes elgondolkodni azon, hogy ideje azokat a funkciókat összegyűjteni, és belőlük már új alkalmazás fejleszhető. Természetesen kisebb alkalmazások kevesebb memóriát igényelnek, amiből így sincs sok a JVM számára. Igyekezzünk úgy megtervezni az algoritmusainkat, hogy ahol lehet, használjunk skalár, primitív Java típusokat objektumok helyett, ahol pedig objektumot használunk, igyekezzünk felszabadítani a lefoglalt memóriát (null-ra állítani az objektumot) magunk, mielőtt nincs tovább szükség az adott példányra, ne hagyatkozzunk a GarbageCollector-ra. Ahol lehet, használjuk az úgynevezett lazy instantiation technológiát, akkor példányosítsunk amikor szükséges. Bár a kivételek a Java kommunikációjának alapkövei, igyekezzünk mobil alkalmazásokban minél kevesebbet használni. Ha van rá mód, alkalmazzunk szerveroldalt nagyobb számításokra, vagy adattárolásra, feldolgozásra. Sok esetben ez elkerülhetetlen is, a korlátozott JVM erőforrások miatt. Vegyük figyelembe, hogy nem csak a Java programozási nyelv létezik mobil eszközökön. Sok telefon képes lefuttatni C++-ban írt (pl Symbian) programokat. Sok esetben ez a nyelv jobb választás. Igyekezzünk mobiloldalon is megvalósítani valamilyen két vagy három rétegű tervezési mintát, mint például az MVC-t. Ahol lehet, használjunk lokális változókat, mert ezeket a JVM gyorsabban kezeli, mint az adattagokat. Nagyméretű String-eken végzett műveletnél kerüljük a String osztály használatát, mivel lassú, és sok memóriát használ. Helyette alkalmas például a StringBuffer osztály használata. Ha kell, ne féljünk szálakat használni, de kerüljük a szálszinkronizációt, mert az is nagyban csökkenti a teljesítményt.

Ezen gyors felsorolás természetesen nem teljes, rengeteg más olyan trükk van még, amit érdemes megfogadni, alkalmazni mobil eszközökre való fejlesztésnél, de ezek kitárgyalása is átlépné az erre a dolgozatra szánt keretet.

4. A jövő

Legvégül tekintsünk kicsit a jövőbe, hogy teljes képünk legyen erről a világról. A dolgozat a mobilkommunikáció múltjával kezdődött, a programozással és a technológia jelenlegi állapotával megnéztük a jelent, hát most nézzük meg, mi következik. Ahhoz viszont, hogy ezt eredményesen megtehessük, ismerni kell a múltat. Az írás elején említettük, hogy a mobiltelefonok fejlődése nagyon döcögösen indult, a tényleges elterjedése váratott magára. Illetve nem is magára, hanem néhány technikai probléma megoldására, ami gátolta terjedését. Úgy gondolom, az egyik ilyen legfontosabb volt 1970-ben a cellák automata átjárhatóságának kifejlesztése volt. Utána kevesebb, mint egy évvel már útjára is indult az első valóban hivatalos, üzleti mobilszolgáltatás, a fejlődése pedig onnantól kezdve töretlen volt. Valószínűleg a jövőben is valami hasonló fejlődésen kell átesnünk ahhoz, hogy kifejlődjenek a mobiltelefonok és mobilszolgáltatások legújabb generációi. Nagyon a jövőbe nem is kell már néznünk, egy kis hárombetűs rövidítés lebeg szemünk előtt, amit most még ugyan nagyon kevesen ismernek, de úgy gondolom, hamarosan (egy kis továbbfejlesztés után) megváltoztatja a mobilkommunikációt, csakúgy, mint 1970-ben az átjárhatóság lehetősége. És hogy miért gondolom így? Mert az említett találmány nagyon hasonló az 1970-eshez. Persze nem a cellák átjárhatóságáról van szó, azt már megoldották akkor, sőt, most már az is gond nélkül működik, hogy ha a mobilinternetező elhagyja a 3G/HSxPA területet, a telefonja adatvesztés nélkül átkapcsol GPRS-re. De azért nagyon hasonló az elv. A rövidítés UMA, ami Unlicensed Mobile Access-t takar. A technológia lényege nagy vonalakban, hogy a telefon képes legyen akadálymentesen átjárni GSM és 802.11 vezeték nélküli hálózatok között. A technológia már be is van építve néhány új mobiltelefonba, mint például a Nokia 6103-ban, de sajnos szolgáltatás még nem nagyon van rá. Pedig valószínűleg ez a jövő mobilkommunikációjának útja. Az ember ingyenesen (havi internetes átalánydíjért) beszélhet WiFi hálózaton, és ha esetleg betéved az őserdőbe, ahol nincs

WLAN, akkor még mindig ott van neki a GSM, mondjuk segélyhívásra. Ehhez persze az kell, hogy a szolgáltatók képesek legyenek IP alapú hangkommunikációt közvetíteni minden egyéb szoftver telefonra való telepítése nélkül, az előfizetők pedig bízzanak a technológiában, mert nélkülük nem működik. Mindent összevetve azt mondhatom, hogy a mobil internet nevű „hóbort” mára akkorává nőtte ki magát, hogy nem csak hogy megállíthatatlan, de hamarosan feltehetőleg a mobilkommunikáció (és talán a vezetékes is) nélkülözhetetlen alapja lesz.

5. Irodalomjegyzék

A dolgozat írásához felhasznált könyvek és internetcímek gyűjteménye

Könyvek

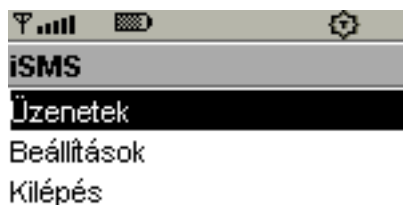
- Reza B'Far – Mobile Computing Principles (Cambridge 2005)
- Vartran Piroumian – Wireless J2ME Platform Programming (The Sun Microsystems Press 2002)

Internet

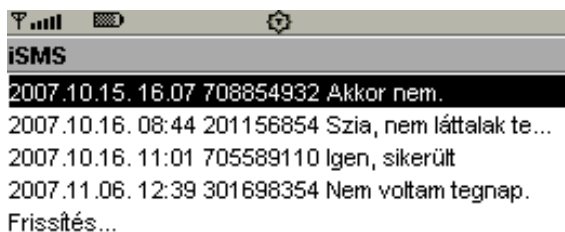
- Cellular Convergence: Evolution, Revolution and Speculation (<http://techbiz.blog.com/1812247/>)
- GSA Mobile facts (<http://www.gsacom.com/news/statistics.php4>)
- Netbeans 5.5.1 (<http://www.netbeans.info/downloads/index.php>)
- Netbeans Mobility Pack (<http://www.netbeans.info/downloads/index.php?rs=22&p=4>)
- Jboss (<http://labs.jboss.com/jbossas/downloads/>)
- Windows Mobile (<http://www.microsoft.com/windowsmobile/default.mspix>)
- Windows Media Player Mobile (<http://www.microsoft.com/windowsmobile/software/mediaplayer.mspix>)

6. Függelék

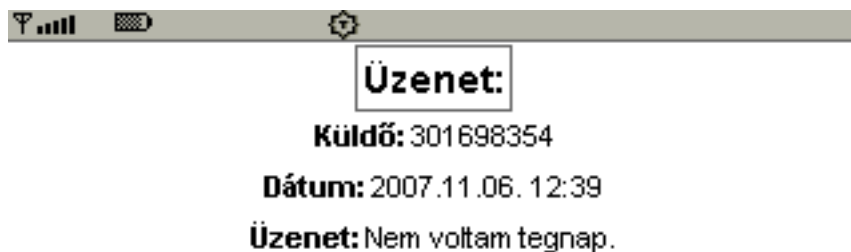
A kliensalkalmazás kijelzőképei.



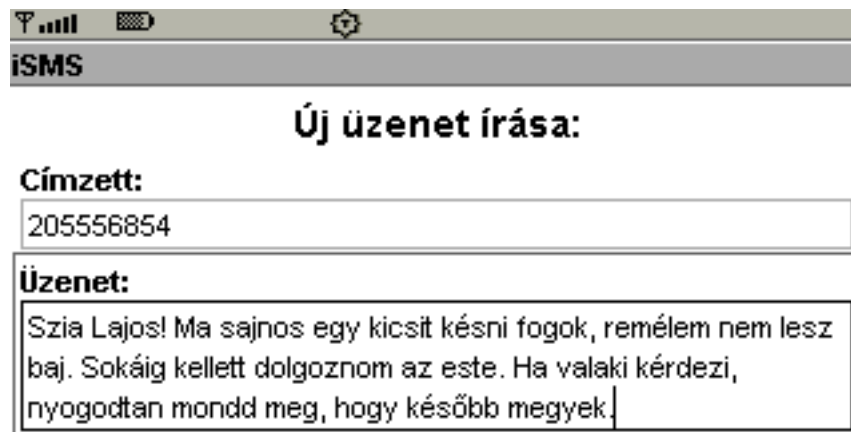
1. ábra: Főmenü



2. ábra: Üzenetek



3. ábra: Egy érkezett üzenet



4. ábra: Üzenet írása