



**Budapesti Műszaki és Gazdaságtudományi Egyetem
Távközlési és Médiainformatikai Tanszék**

TDK dolgozat

**Üzenetkezelő rendszer vak és látássérült felhasználók
részére mobil eszközökön**

Készítette:

Pandur Balázs, V. Inf.

Konzulensek:

Dr. Németh Géza, Távközlési és Médiainformatikai Tanszék

Tóth Bálint Pál, Távközlési és Médiainformatikai Tanszék

Budapest, 2011. október

Tartalomjegyzék

A dolgozat kivonata	4
1. Bevezetés.....	4
2. Probléma felvezetés	5
2.1 Okostelefonok	5
2.2 Mobil operációs rendszerek	6
2.2.1 A Google Android platform bemutatása	6
2.2.2 Az Apple iOS platform bemutatása	7
2.2.3 A Symbian platform bemutatása.....	8
2.3 Okostelefonok megjelenése a piacon	9
2.4. TTS (Text-to-Speech) modul	10
2.4.1 A beszéd szintézis	10
2.4.2 Kötött szótáras beszédfelismerők	10
2.4.3 Az általános szövegfelolvasó megvalósítási sémája	10
2.4.4 A szintetizált beszéd alkalmazási területei	11
2.4.5 A TTS modul és az operációs rendszerek	12
2.5. Beszédfelismerők	12
2.5.1 Az általános beszédfelismerő	12
2.5.2 Mintaillesztési eljárások	13
2.6 Már meglévő beszédtechnológiai alkalmazások vakok és gyengén látók számára	13
2.6.1 Beszélő bankautomaták	14
2.6.2 NaviSpeech – beszélő navigátor látássérült gyalogosoknak	14
2.7 A piacon található üzenetkezelő rendszerek vizsgálata	14
2.7.1 Google Android.....	15
2.7.1.1 SMS Reader	15
2.7.1.2 SMS popup	16
2.7.1.3 SMS speaker	16
2.7.1.4 K-9 Mail.....	17
2.7.1.5 Handcent SMS	18
2.7.1.6 GO SMS Pro	20
2.7.2 Apple iOS	21
2.7.2.1 Email'nDrive - Hands Free Email Message Reader	21
2.7.2.2 Text'nDrive - Speech To Text Hands Free Message Reader	22
2.7.3 Symbian	22
2.7.3.1 Talking SMS	22
3. Saját email és SMS felolvasó rendszer specifikációja	23
4. A rendszer megvalósítása	24
4.1 mobil platform választása	24
4.2 Egy Android alkalmazás felépítése	24
4.2.1 Biztonsági megoldások Android platformon	24
4.2.2 Alkalmazás komponensek	25
4.3 Az üzenetkezelő rendszer működési elvének megtervezése	27
4.4 Az üzenetkezelő rendszer fejlesztése során szerzett tapasztalatok.....	27

4.5 Az üzenetkezelő rendszert megvalósító osztályok és leírásaik	28
4.6 Az üzenetkezelő rendszer UML diagramja	30
4.7 Az üzenetkezelő rendszer nézeteinek megtervezése.....	32
4.8 Eredmények, fejlesztési tervek.....	38
Irodalomjegyzék	39

A dolgozat kivonata

Napjainkban nagymértékben elterjedtek a telefonok piacán az ún. okostelefonok, amelyek az alapfunkciókon kívül már érintőképernyővel, internetkapcsolattal, és egyéb kiegészítő funkciókkal is rendelkeznek. A piac mai állása szerint ezeken a készülékeken többségben Google Android operációs rendszer található. [1]

Habár ez az operációs rendszer igen elterjedt, a vakok és látássérültek számára nehezen kezelhető, vagy egyáltalán nem elérhető funkciókat is tartalmaz, hiszen a legtöbb információ vizuális formában érkezik a készülékre. Ilyen funkciók lehetnek pl. az SMS, MMS és levél üzenetek fogadása, és küldése, amelyek használata már-már alapvetővé vált a mai modern társadalomban. További problémákat okoz a vak és gyengén látó felhasználó számára az érintőképernyő megfelelő kezelése is, hiszen a billentyűzettel rendelkező telefonokkal ellentétben, ebben az esetben mindössze egy tenyér méretű felület áll rendelkezésére.

A hátrányok kiküszöbölésére megoldást jelenthet egy olyan rendszer fejlesztése, amely mesterséges szövegfelolvasó és beszédfelismerő segítségével egy olyan környezetet valósít meg, amely lehetővé teszi a vak és látássérült felhasználó számára, hogy a készüléken található SMS, MMS és levél funkciókat maradéktalanul tudja használni. Ilyen funkció a beérkező üzenetek jelzése rezgés, ill. hang formájában, annak felolvasása, ill. üzenetek létrehozása, majd elküldése a címzettnek.

A rendszer középpontjában a szövegfelolvasó, TTS (Text-To-Speech) [2] és a beszédfelismerő, un. ASR (Automatic Speech Recognition) [3] motor használata áll. A TTS motor lehetőséget nyújt szöveges tartalmak felolvasására, ill. megkönnyíti a menüben való navigálást. Az ASR motor segítségével pedig megkönnyíthetjük az SMS, és e-mail üzenetek írását. Az említett funkciók mellett fontos lenne egy olyan felhasználói felület kialakítása is, amely az említett célközönség számára kényelmes megoldást nyújt a rendszer használatakor, pl. a képernyő megfelelő elrendezésével, kiválasztott menüpont címének felolvasásával, és így tovább.

A dolgozatomat négy fő részre osztottam fel. A bevezetés követően a második fejezetben áttekintem a manapság népszerűnek mondható mobil operációs rendszereket, piaci elhelyezkedésüket, néhány már elérhető üzenetkezelő alkalmazást, az üzenetkezelő rendszer által használt beszédszintézis, és beszédfelismerés alapjait, ill. egyéb megoldásokat, amiket kifejezetten vakok és gyengén látó felhasználók számára fejlesztettek ki.

A tesztelt alkalmazások vizsgálata során számos hiányosságot tapasztaltam, ami nehézséget okozhat egy vak és gyengén látó felhasználó számára. Ezekre a harmadik fejezet elején térek ki, amit az üzenetkezelő rendszer pontos specifikációja követ.

A negyedik és egyben utolsó fejezet a rendszer megvalósításával kapcsolatos döntéseket, tapasztalatokat, a rendszer felépítését, a tesztelési eredményeket, ill., a fejlesztési terveket tartalmazza.

1. Bevezetés

A mobiltelefon nagymértékű fejlődésen ment keresztül a 20. században. Az 1973. április 3-án elindított első hívás óta napjaink egyik legmeghatározóbb kommunikációs eszközévé vált. Ezek a készülékek manapság már nem csak hívásokra, hanem egyéb más szolgáltatásokra is használhatóak. A manapság igen elterjedtnek számító okostelefonok már saját operációs rendszerrel, internettel és GPS kapcsolattal is rendelkeznek, amelyek új funkciókat biztosítanak a felhasználó számára. Az okostelefonok talán egyik leghasznosabb funkciója, hogy változatos kommunikációs lehetőségeket biztosítanak. Mivel az ilyen készülékek rendelkeznek internetkapcsolattal, ezért elérhetővé válnak az e-mail üzenetek, vagy az aktuális hírek olvasása, esetleg csevegés barátokkal, ismerősökkel. Ezek a telefonok főként hatalmas kijelzővel és többnyire néhány gombbal rendelkeznek, ezért bizonyos

felhasználói körök, mint például a vakok és gyengén látók nehezen tudják kezelni a készüléket. Célszerű tehát egy olyan rendszer kifejlesztése, amely ezeknek az embereknek nyújt segítséget a mindennapjaikban, és kiterjeszti a kommunikációs lehetőségeiket azáltal, hogy az előbb említett funkciókat elérhetővé teszi számukra.

Dolgozatom fő témája egy olyan üzenetkezelő rendszer fejlesztése, és bemutatása, ami segítséget nyújthat a vak és gyengén látó felhasználó számára, lehetőséget biztosítva SMS és e-mail üzenetek küldésére és fogadására.

A mai okostelefonokra számos operációs rendszer elérhető, így magának a platformnak a kiválasztása sem egyértelmű. A fejlesztői munkálatok előtt mindenképpen szükségesnek tartottam hasonló üzenetkezelő rendszerek kipróbálását és vizsgálatát is, figyelembe véve, hogy ezek mennyire használhatóak egy vak és gyengén látó felhasználó számára.

A rendszer fontos része, hogy képes legyen szövegek felolvasására, ill. felismerésére. Ezek a funkciók a TTS (Text-To-Speech) modul és az ASR (Automatic Speech Recognizer) motor segítségével valósíthatók meg. Ezek működési elvéről az ötödik és hatodik fejezetben talál áttekinthető információt az olvasó.

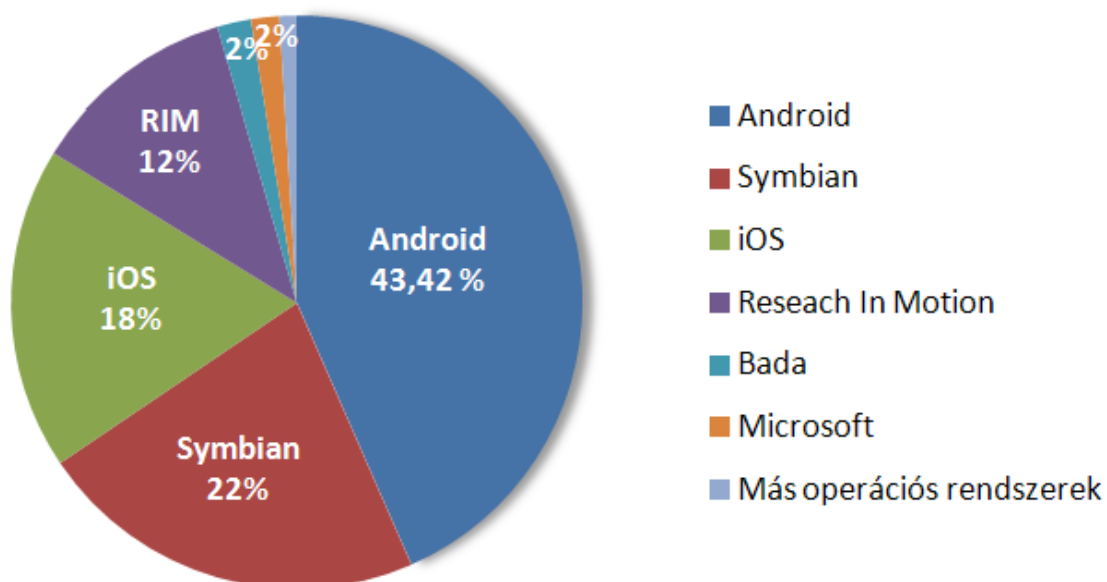
A rendszer fejlesztése előtt végzett kutatómunka jó alapot adott egy vakoknak és látás-sérülteknek készülő üzenetkezelő rendszer megvalósításához. Végző célként egy olyan alkalmazás elkészítését tűztem ki, ami egyszerre teszi lehetővé SMS-ek, és e-mailek kezelését olyan elrendezésben amit, az említett felhasználói kör is gond nélkül tud használni.

2. Probléma felvezetés

2.1 Okostelefonok

Az okostelefonok, manapság a legjobban elterjedt kommunikációs eszközök, amelyek ötvözik a PDA-k (personal digital assistant) és a mobil telefonok funkcióit.

Fontos megjegyezni, hogy az okostelefonok saját operációs rendszert használnak. Ezek közül a legerjedtebbek a Google Android, Apple iOS, Microsoft Windows Phone 7, és a Symbian (2-1. ábra).



2-1. ábra 2011 második negyedéves okostelefon eladások világszerte a Gartner alapján [1]

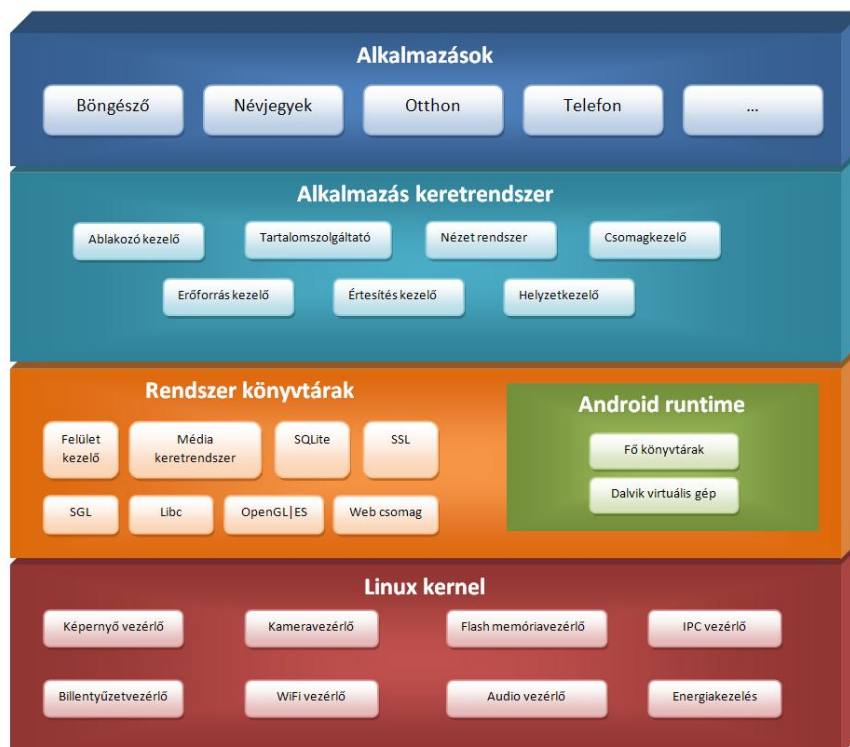
2.2 Mobil operációs rendszerek

A régebbi hagyományos mobiltelefonoknak nem volt szükségük fejlett operációs rendszerre. Kezdetben mindössze hardver közeli programokat használtak, azonban a folyamatos technikai fejlődés miatt szükségessé váltak külön a mobiltelefonokra kifejlesztett operációs rendszerek, amelyek az alapfunkciók mellett (SMS, MMS) egyéb multimédiás funkciókat is elláttak.

A mobil operációs rendszerek működése és felépítése nagyban hasonlít az asztali gépeken megtalálható operációs rendszerekéhez, például a Windowshoz, OS_X-hez (Apple iOS), vagy a különböző Linux disztribúciókéhoz (Google Android). A hasonlóság ellenére ezek a rendszerek felépítésben mégis egyszerűbbek, főként a vezeték nélküli kapcsolattartásra és multimédiás tartalmak megjelenítésére, kezelésére koncentrálnak. A felhasználó szemszögéből a kommunikációs eszköz funkciója mellett szórakoztatóelektronikának is tekinthetőek.

2.2.1 A Google Android platform bemutatása

A Google Android platform megjelenéséig még nem létezett a mobil eszközök világszerte széleskörűen használt egységes nyílt forráskódú operációs rendszere. Az platform alapja egy Linux alapú operációs rendszer, amit úgy alakítottak át, hogy képes legyen problémák nélkül kezelni a mobil eszközök integrált hardvereit (érintőképernyő, WiFi, HSDPA, Bluetooth, és így tovább, ld 2-2. ábra). Miután 2005-ben a Google felvásárolta az Android, Inc.-et a fejlesztések eredményeként a Linux kernel fölé egy virtuális gép került. Ez a virtuális gép felelős a felhasználói felület, és az alkalmazások futtatásáért. Az Android platformot 2007. november 5-én az Open Handset Alliance jelentette be. Az Open Handset Alliance (OHA) olyan cégek konzorciuma, akiknek a tagjai érdekeltek egy szabad és nyílt forrás kódú platform bevezetésében.



2-2. ábra Az Android architektúrája

A platform alapját a Linux kernel adja, amely tartalmazza a kezelendő eszközök meghajtó programjait (ezeket azok a cégek készítik el, amelyek a platformot saját készülékükön szeretnék használni), feladata továbbá a memóriakezelés, a folyamatok ütemezése és a teljesítmény-kezelés is. Utóbbi az alacsony fogyasztás segíti elő.

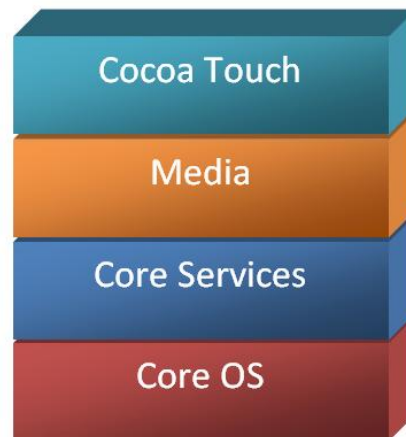
A kernel szolgáltatásait a Linux rendszerekben megtalálható programkönyvtárak használják. Ilyenek például a libc, az SSL, vagy az SQLite is. Ezek a könyvtárak C/C++ nyelven íródtak és közvetlen a Linux kernelen futnak. Részben ezekre épül a Dalvik virtuális gép is, ami egyáltalán nem kompatibilis a Sun virtuális gépével, hiszen teljesen más utasítás készletet, és bináris programot futtat.

A kék színnel jelölt részekben már csak Java forrásokat találunk, amelyet a virtuális gép futtat. Lényegében ez jeleníti meg a felhasználó számára az operációs rendszer felületét, illetve az általa használt alkalmazásokat.

2.2.2 Az Apple iOS platform bemutatása

Az iOS vagy korábbi nevén iPhone OS az Apple operációs rendszere. Eredetileg az iPhone-ra fejlesztették, de azóta kiterjesztették, hogy egyéb Apple eszközöket is támogasson, mint például az iPad-et, vagy az Apple TV-t.

Az operációs rendszert ugyan a MAC OS X-ből származtatták, az iOS-t úgy tervezték, hogy mobil környezetben is megfelelően lehessen használni, hiszen a felhasználók igényei mobil eszközök használatakor eltérőek lehetnek.



2-3. ábra Az iOS rétegszerkezete

Mint 2-3 ábrán is látható az iPhone OS négy fő rétegből tevődik össze: Core OS, Core Services, Media és a Cocoa Touch. A Core OS és a Core Services réteg tartalmazza az alapvető interfészeket az iOS számára, beleértve a fájlok elérését, alacsony szintű adattípusokat, a Bonjour szolgáltatásokat és így tovább. Ezek az interfészek főként C alapúak és olyan technológiákat foglalnak magukba, mint például Core Foundation, CFNetwork, SQLite, hozzáférés a POSIX szálakhoz és POSIX socketekhez.

A Core OS, és a Core Services rétegek felett található a Media réteg, amelyben már sokkal fejlettebb technológiai megoldásokat találhatunk. Ez a réteg C-ben és Objective-C-ben íródott interfészeket és olyan alapvető technológiákat használ, amelyek támogatják a 2D-s, és 3D-s rajzolást, ill. a hang és videó tartalmak kezelését. A réteg ugyanakkor tartalmaz további C alapú technológiákat is, mint például OpenGL ES, Quartz és Core Audio, ill. az Objective-C-ben íródott Cora Animation-t, amely egy fejlett animációs motor.

A rétegszerkezet legfelső eleme a Cocoa Touch réteg, amelynek technológia megoldásai főként az Objective-C-t használják. A réteg keretrendszerei biztosítják az alapvető infrastruktúrát, amit az alkalmazások használnak. Ilyen keretrendszer például a Foundation, ami objektum orientál támogatást nyújt fájlkezelésekhez, hálózati műveletekre és így tovább. A UIKit biztosítja az alkalmazás vizuális megjelenítését, beleértve az ablakok osztályait, nézeteket, és a kontrollert, ami menedzseli ezeket az objektumokat. Más ezen a szinten lévő keretrendszerek hozzáférést nyújthatnak a felhasználó névjegyzékéhez, fotókhoz, gyorsulásmérőhöz, és más egyéb hardver eszközökhöz, amelyek a készüléken megtalálhatóak. [4]

2.2.3 A Symbian platform bemutatása

A Symbian platform a Symbian OS és a Nokia Series 60 utódja. A legújabb verzió, a Symbian^3, amit hivatalosan 2010 év végén adtak ki, és először a Nokia N8-nál használták. Az utolsó frissítések 2011 májusában a Symbian Anna-val, majd ezt követően 2011 augusztusában a Symbian Belle-vel jelentek meg.

A Nokia 2011. február 11-én bejelentette, hogy az a Symbian-t Windows Phone 7-re cseréli.

Az operációs rendszer rétegszerkezete (fentről lefelé haladva):

1. UI Framework Layer
2. Application Services Layer
 - JAVA ME
3. OS Services Layer
 - generic OS services
 - communications services
 - multimedia and graphics services
 - connectivity services
4. Base Services Layer
5. Kernel Services & Hardware Interface Layer

A Base Services Layer a legalacsonyabban található szint, amelyet a felhasználói oldalról még el lehet érni. Itt található a File Server, a User Library, és a Plug-In Framework is, ami menedzseli az összes plugint, központi adatbázist, a DBMS-t (Database Management System – adatbázis kezelő rendszer), és titkosítási szolgáltatásokat.

A Symbian mikrokernél architektúrával rendelkezik, tehát csak néhány alapvető funkciót valósít meg, a nélkülözhetetlen operációs rendszerszolgáltatások realizálásához. Mindössze a címtartomány menedzsment, szál menedzsment, ütemező és memória menedzsment feladatát látja el, ill. néhány egyéb szolgáltatást, mint például a hálózatépítés, telefonálás, és fájlrendszer támogatása. Ezek a feladatok az OS Services, vagy a Base Services réteghez tartoznak.

Egy kimutatás szerint a Symbian operációs rendszerrel ellátott készülékek eladása nagymértékben uralja a piacot. 2009-2010-ig az eladott készülékek nagy részén ilyen mobil platform volt megtalálható. 2011-ben az eladott készülékek aránya csökken a többi vetélytárséhoz képest, azonban még mindig meghatározó. Az előrejelzés szerint 2014-ben már előnye nem lesz olyan meghatározó, azonban vezető pozícióját még mindig tartani fogja.

Mobil eszközök eladási statisztikáinak előrejelzése operációs rendszer szerint 2009-2014 (Ezer / Egység)				
Operációs rendszer	2009 piaci részesedés (%)	2010 piaci részesedés (%)	2011 piaci részesedés (%)	2014 piaci részesedés (%)
Symbian	46.9	40.1	34.2	30.2
Android	3.9	17.7	22.2	29.6
Research In Motion	19.9	17.5	15.0	11.7
iOS	14.4	15.4	17.1	14.9
Windows Phone	8.7	4.7	5.2	3.9
Más operációs rendszerek	6.1	4.7	6.3	9.6

2-4. ábra Az Symbian még mindig a legnagyobb darabszámban használt okostelefon platform [6]

2.3 Okostelefonok megjelenése a piacon

Az okostelefonok eladása évről-évre rohamosan nő (ld. 2-5. ábra). Egy kimutatás szerint a világszerte eladott készülékek száma 2011 második felében 428.7 millió egység volt, ami 2010 második feléhez képest 16.5 %-os növekedést jelent.

2011. második negyedéves mobilkészítők eladások világszerte gyártók szerint (Ezer / Egység)				
Operációs rendszer	2Q11 Eladott Egység	2Q11 piaci részesedés (%)	2Q10 Eladott egység	2Q10 Piaci részesedés
Nokia	97 869,30	63,69	111 473,70	107,8
Samsung	69 827,60	16,29	65 328,20	17,75
LG	24 420,80	5,7	29 366,70	7,98
Apple	19 628,80	4,58	8 743,00	2,38
ZTE	13 070,20	3,05	6 730,60	1,83
Reseach In Motion	12 652,30	2,95	11 628,80	3,16
HTC	11 016,10	2,57	5 908,80	1,61
Motorola	10 221,40	2,38	9 109,40	2,48
Huawei	9 026,10	2,11	5 276,40	1,43
Sony Ericsson	7 266,50	1,7	11 008,50	2,99
Más gyártók	153 662,10	35,85	103 412,60	28,1
Összesen	428 661,20	100	367 986,70	100

2-5. ábra Eladott készülékek száma 2011 második felében [1]

Ennek a drasztikus növekedésnek az oka lehet, hogy a gyártók újabb és újabb fejlesztésű telefonokat dobnak piacra, amelyek csábító funkciókkal és felhasználói felülettel rendelkeznek, áruk egyre inkább megfizethető, és nem mellékesen rengeteg letölthető alkalmazás áll a felhasználó rendelkezésére.

2011. második negyedéves okostelefon eladások világszerte operációs rendszerek szerint (Ezer / Egység)				
Operációs rendszer	2Q11 Eladott Egység	2Q11 piaci részesedés (%)	2Q10 Eladott egység	2Q10 Piaci részesedés
Android	46 775,90	44,523	10 652,70	529,75
Symbian	23 853,20	22,14	25 386,80	40,91
iOS	19 628,80	18,22	8 743,00	14,09
Reseach In Motion	12 652,30	11,74	11 628,80	18,74
Bada	2 055,80	1,91	577	0,93
Microsoft	1 723,80	1,6	3 058,80	4,93
Más operációs rendszerek	1 050,60	0,98	2 010,90	3,24
Összesen	107 740,40	100	62 058,00	100

2-6. ábra Eladott készülékek operációs rendszerei 2011 második felében [1]

2011-ben az okostelefonok ökoszisztémájának egyértelmű győztesei a Google és az Apple (ld. 2-6. ábra). Együttes piaci részesedésük 2011 második felében közel megduplázódott 2010-hez képest

(közel 62 %). Ennek oka lehet, hogy ezek az operációs rendszerek a felhasználó számára használható, élvezhető felületet és funkciókat nyújtanak.

2.4. TTS (Text-to-Speech) modul

2.4.1 A beszéd szintézis

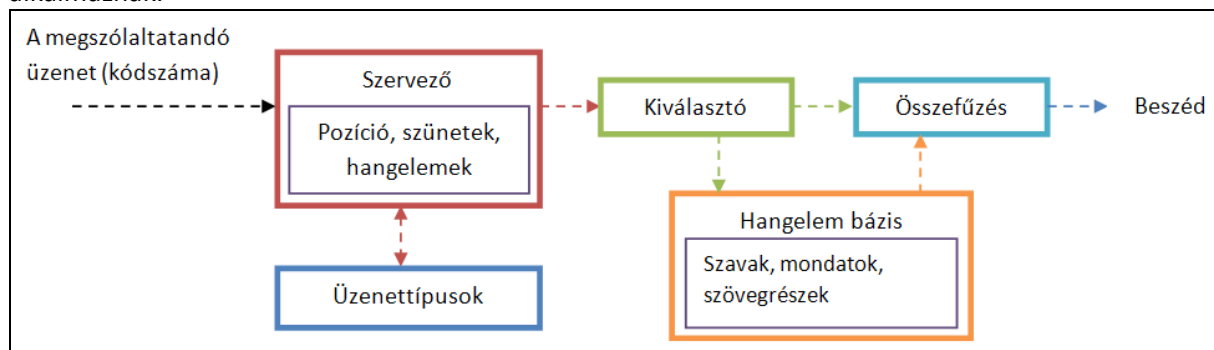
A beszéd szintézis az emberi beszéd mesterséges előállítását jelenti. A számítógép rendszert, amelyet erre a célra használunk, beszéd szintetizátornak nevezzük. Ez implementálható hardveres és szoftveres úton egyaránt.

A Text-To-Speech rendszer a normál nyelvi szöveget alakítja beszéddé. A szintetizált beszéd létrehozható felvétel töredékek összefűzéséből, amiket adatbázisban tárolnak. Az egyes rendszerek különbözhetnek a tárolt beszéd részletek hosszában, amelyek hatással lehetnek a szintézis minőségére is. Felhasználási területtől függően tárolhatnak szavakat, vagy egész mondatokat a kimenet minőségének javítása érdekében, emellett belefoglalhatják a modellbe az emberi hang karakterisztikáját is.

2.4.2 Kötött szótáras beszéd felismerők

A 20. század közepén felmerült az igény a gépi beszéd alkalmazására, például egyszerű távközlési szolgáltatásoknál, ill. megjelentek a számítógépes hangkártyák is. Ez lehetővé tette, hogy előre felvett hangüzeneteket automatikusan távközlési csatornára továbbítsanak. Ez a technológia ma is széles körben elterjedt, például pontos idő, megállónév, indulási idő stb. bemondására, de ilyen rendszerek működnek a hívasközpontokban is, és más egyéb egyszerű automatikus információs rendszerekben.

A kötött szótáras beszéd szintetizáló emberi beszédből kivágott hullámforma részletek összekapcsolásával hozza létre a kimondandó szöveget, azonban jelfeldolgozási algoritmusokat nem alkalmaznak.



2-7. ábra Az általános kötött szótáras beszéd szintetizátor bloksémája

A rendszer fő elemei (ld. 2.7. ábra):

- elembázis: üzenetek, üzenetrészek hullámformáinak hangelemtára
- elemző és kiválasztó algoritmus: meghatározza, hogy az elemtárból mely összefűzendő hangelemeket (üzenetrészeket) kell kiválogatni a bejövő parancs alapján.
- elemösszefűző: feladata értelemszerűen az kiválasztott elemek összefűzése
- megszólaltató egység: feladat az elemösszefűző által adott eredmény felolvasása

[5] 421. o. 10.2

2.4.3 Az általános szövegfelolvasó megvalósítási sémája

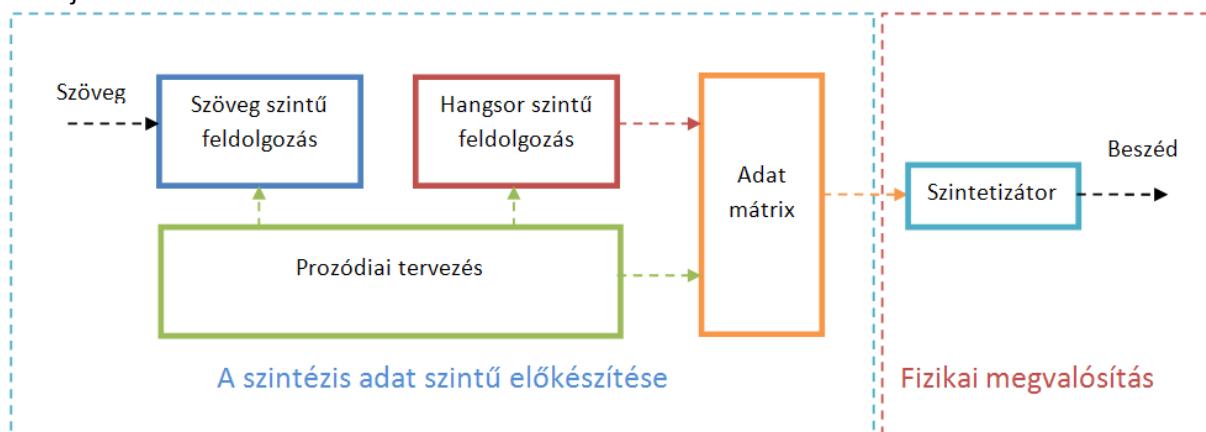
Egy szövegfelolvasó megvalósításához két jól elkülönült modul kell megvalósítani. Az első modul feladata a rendszer bemenetére kapott szöveg feldolgozása, adat szintű előkészítése a

szintézishez, míg a második rész a szintézis fizikai megvalósításáért felelős. A modulrendszerű felépítés lehetővé teszi, hogy a fejlesztés során ellenőrizhető legyen az egyes modulok (jelfeldolgozási szintek) helyes működése, külön-külön, csoportosítva, vagy teljes egészében rendszert alkotva is.

A szöveg előkészítésének különböző mélységű szintjei vannak. Értelemszerűen a szöveg minden elemét, így a szavakat és írásjeleket is értelmezni kell. A nyelvi értelemben vett szavak esetében ez nem igényel bonyolult rendszert, azonban olyan karaktersorozatoknál, amelyek nem tartoznak szavak e csoportjába, meg kell fejteni a karaktersorozat jelentését, majd értelmes szavak sorozatává kell átalakítani. (Ilyenek lehetnek a rövidítések, a betűszók, és így tovább.) A bejövő szöveg átalakítása után a kapott eredmény már csak betűket tartalmaz.

A feldolgozás következő szintje, hogy a szövegbe bejelöljük, hogy milyen dallamot, ritmust, hangsúlyt kaphat az adott szó, a szövegrész, vagy mondat. A feldolgozásnak ezt a szintjét a prozódiai tervezésnek nevezzük. A prozódiai előrejelző jelek meghatározásához csak a bemenő szöveg áll rendelkezésre, ezért főként szövegelemzés szintje határozza meg, hogy mennyire pontosan lehet a mondatdallamot, a hangsúlyozást, a beszédsebesség esetleges változását meghatározni.

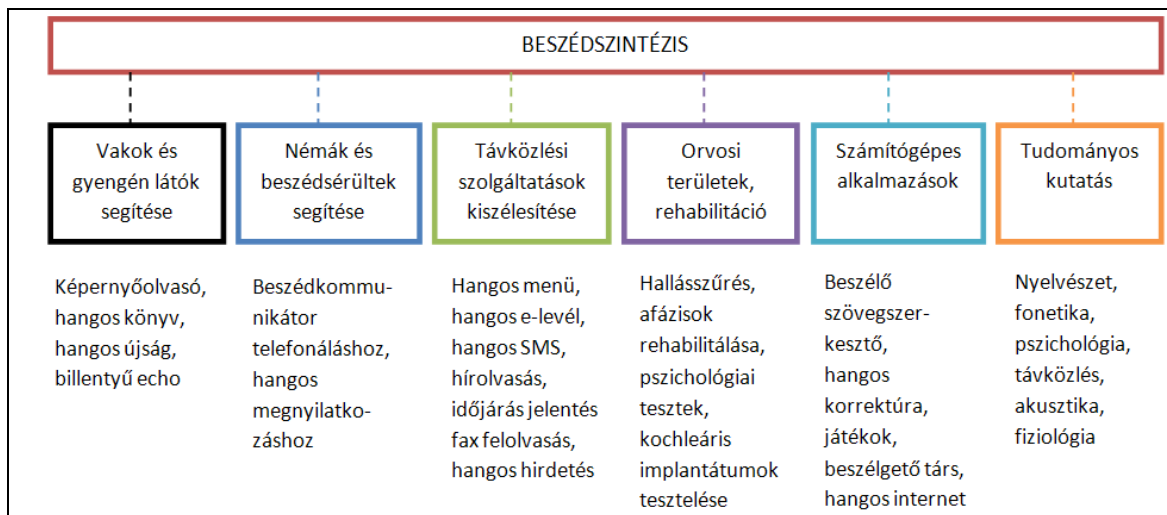
A szintézis adat szintű előkészítésének utolsó lépéseként a szöveget hangkódokká alakítjuk. Ennek végeredménye a hangzó forma adatszintű megvalósítása, amit egy speciális mátrixban összegzünk. Ez a mátrix képviseli a kapcsolódási felületet a tényleges beszéd szintetizátorral, ami képes az előkészített adatokból a beszéd előállítására. Egy általános szövegfelolvasó megvalósítási sémája a 2-8. ábrán látható.



2-8. ábra Az általános szövegfelolvasó megvalósítási sémája

2.4.4 A szintetizált beszéd alkalmazási területei

A beszéd szintézis technológiája a 21. század elejére olyan szintre fejlődött, hogy a társadalom számos területén alkalmazható. Ilyen terület például a vakok és a gyengén látó emberek, vagy a némák és beszéd-sérültek segítése. Ezeknek az embereknek hatalmas segítséget nyújtanak ezek a technológiai vívmányok. A vakok és gyengén látók számára lehetővé válik szöveges üzenetek olvasása, vagy akár a számítógép használata is, vagy a beszéd-sérültek olyan kisméretű eszközt kaphatnak, amellyel a leírt gondolataikat beszéddé tudják alakítani, ezáltal akár telefonálhatnak is. A szövegfelolvasóknak rengetegféle gyakorlati alkalmazásuk lehetséges. Néhányat ezek közül a 18. ábra ismerteti. (2-9. ábra)



2-9. ábra A szintetizált beszéd felhasználási területei

2.4.5 A TTS modul és az operációs rendszerek

Számos operációs rendszer részét képezi a TTS modul. Ilyen modult használó programok már például a Windows 95-höz is elérhetőek voltak. Ilyen volt a Narrator, vagy a CoolSpeech amely képes volt szövegek hangos felolvasására. A TTS modul az okostelefonok piacán is jelen van. Például az Android operációs rendszeren az 1.6-os változattól elérhető, és Angolul (az amerikai és a brit akcentus ugyancsak támogatott), Franciául, Németül, Olaszul és Spanyolul és képes szövegek felolvasására.

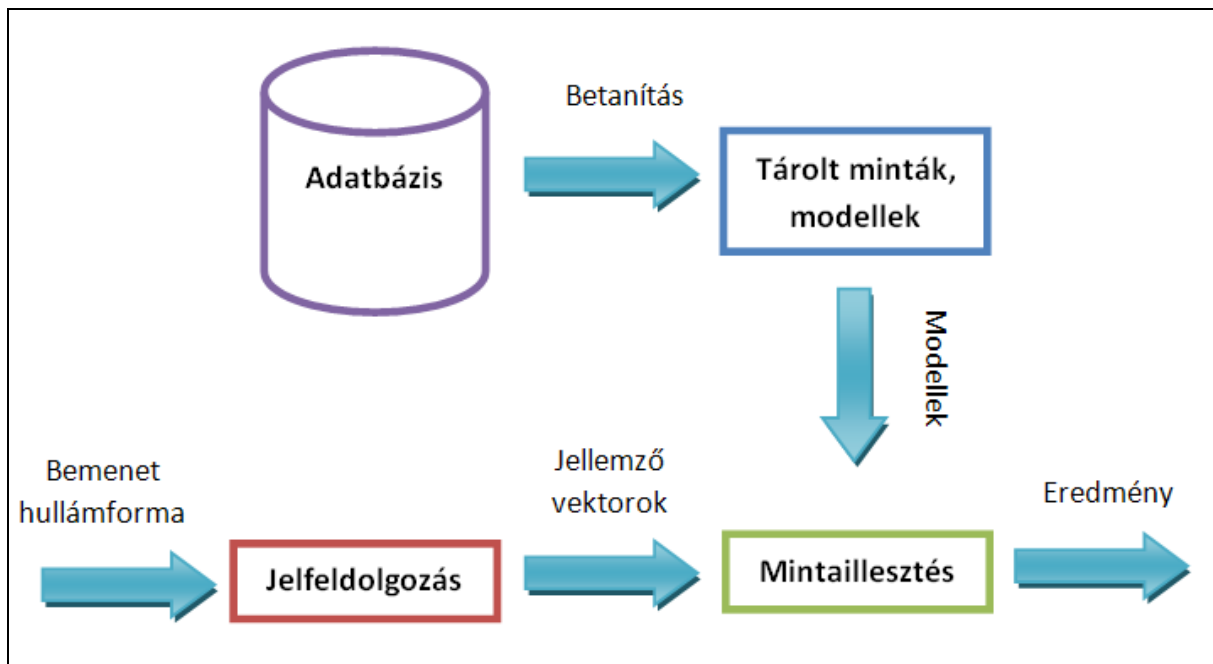
2.5. Beszédfelismerők

2.5.1 Az általános beszédfelismerő

A beszédfelismerő rendszerek alapvető feladata, hogy a beszédjelet a számítógép, illetve a felhasználó által értelmezhető alakra, például szöveggé alakítsák. A beszéd bonyolult, összetett jel, így analízise is komplex eljárást igényel.

A ma ismert beszéd felismerési eljárások összehasonlításokon alapulnak, amik előre eltárolt mintákat hasonlítanak össze a bejövő jeltől képzett mintasorozattal.

A bemeneti hullámforma egy jelfeldolgozó egységbe kerül, ami a folyamatos beszédjelből a gép számára értelmezhető formátumot, jellemzően tulajdonságvektorok-sorozatát állítja elő. Az így képzett vektor-sorozatok a mintaillesztő egységbe kerülnek. Ezeket a rendszer összehasonlítja a már tárolt mintákkal és modellekkel (a modellek beszédatadabázisok alapján kerülnek betanításra), majd végezetül a beszédfelismerő kimenetén megjelenik az értelmezett szöveg. Egy ilyen általános beszédfelismerő a 2-10. ábrán látható.



2-10. ábra Az általános beszédfelismerő

2.5.2 Mintaillesztési eljárások

A mintaillesztő feladata a bemenésként kapott tulajdonságvektor-sorozat leképezése felismert szimbólumok (nyelvi elemek) sorozatává. A mintaillesztő talán a beszédfelismerő egyik legfontosabb része, hiszen a beszédjelből képzett tulajdonságvektor-sorozat itt kerül illesztésre/felismerésre.

Két alapvető mintaillesztési technika alakult ki:

- a sablon (template) bázisú mintaillesztés
- statisztikai mintaillesztés

Az első esetben a mintaillesztés során az előre tárolt mintákhoz hasonlítja a rendszer a tulajdonságvektor-sorozat egyes elmeit. Ezt a mintaillesztési módot jellemzően kisszótáras felismerőkben használják. A korszerű felismerők túlnyomó többsége azonban a statisztikai mintaillesztést alkalmazza (A gyakorlatban leginkább a tisztán rejtett Markov-modelles felismerők terjedtek el).

A mintaillesztési eljárások alapja, hogy a felismerni kívánt szavakból, beszédelemekből, vagy ezeknek láncairól referenciamodelleket hozunk létre, ezeket adatbázisban tároljuk, majd az illesztés során megkeressük a felismerő bemenetére érkező beszédjel és a referenciamodellek közötti legjobb illeszkedést. Végezetül valamennyi referenciamintára kiszámítjuk a hasonlóság mértékét, az adott referencia és a bemeneti beszédminta között, majd a mintaillesztő kimenetére azokat a referenciamodelleket küldjük, amelyek a legnagyobb hasonlósági mértékkel rendelkeznek. [5] 344. o 9.4

2.6 Már meglévő beszédtechnológiai alkalmazások vakok és gyengén látók számára

A BME Távközlési és Médiainformatikai Tanszék-én már készült néhány vakokat támogató rendszer. Ilyenek például a *Beszélő bankautomaták*, vagy a látássérült gyalogosokat segítő *NaviSpeech*. Előbbi a bankautomaták használatakor nyújt segítséget, utóbbi pedig egy GPS alapú helymeghatározó rendszert nyújt a felhasználó számára.

2.6.1 Beszélő bankautomaták

Készítették: Németh Géza-Kiss Géza-Bartalis Mátyás

A bankautomata beszéddel tájékoztatja a felhasználót, a képernyőn megjelentekről. A felvenni kívánt pénzüsszeg, vagy hibaüzenet felolvasásával könnyíti a tájékozódást. Az automatát fejhallgató-csatlakozóval is ellátták, hogy a készülék beszédét csak az ügyfél hallja, emellett szükséges volt a beszélőszoftver beillesztése az automata szoftverrendszerébe is. [5] 590.o 12.7.4-es fejezet

2.6.2 NaviSpeech – beszélő navigátor látássérült gyalogosoknak

Készítették: Viktóriusz Ákos-Németh Géza-Tóth Bálint

Az alkalmazás grafikus felülete szöveges megjelenítésre van korlátozva, és fő jellemzője a beszédalapú irányítás. A rendszer számos funkcióval rendelkezik, mint például az aktuális pozíció lekérdezése, útvonalfelvétel, útvonaltervezés és így tovább. A rendszer egyik fő építőelemét a BME-TMIT Beszédtechnológiai Laboratóriumában kifejlesztett NaviSpeech rendszer jelenti (Tóth-Németh 2007). Ez egy Symbian alapú okostelefonokon futtatható helymeghatározó rendszer, ami a helymeghatározást GPS (Global Positioning System) alapján végzi. Az alkalmazás a navigáció során beszéddel tájékoztat az aktuális pozícióról, és utasításokat adva vezeti a látássérültet a helyes irányba. A képernyőn megjelenik a kimondott utcanév. A betűk mérete nagyítható, így a gyengén látó felhasználók újra elolvashatják a képernyőn megjelent információkat. A rendszer emellett funkciók mellett képes még előre megtervezett útvonalak betöltésére, és elmentésére is. [5] 591. o. 12.7.5

2.7 A piacon található üzenetkezelő rendszerek vizsgálata

Az előző fejezetekben már olvasható volt, hogy manapság az okostelefonok, és operációs rendszereik igen elterjedtek számítanak. Hordozhatóságuk, és felhasználói élményük sok vásárlót csábítanak, azonban kiemelt funkciójuk mégis talán a kommunikációs lehetőségek magasabb szintre hozásában rejlik. Gondoljunk csak arra, hogy az SMS, és MMS funkciókon kívül email és egyéb szöveges üzeneteket is fogadhatunk, nem beszélve arról, hogy azonnal értesülhetünk a világban történt eseményekről is. Manapság ugyancsak elterjedtek a közösségi oldalak, amelyek segítségével bárki tarthatja a kapcsolatot barátaival, munkatársaival, akár chat, vagy egyéb üzenetek formájában.

Habár ezek az operációs rendszerek igen elterjedtek, és fejlett kommunikációs eszközökkel rendelkeznek a vakok és látássérültek számára nehéz, vagy egyáltalán nem elérhető funkciókat is tartalmaznak, hiszen a legtöbb információ vizuális formában érkezik a készülékre. Ahogy az előbb említettem ezek a mai társadalom fontos részét képezik, hiszen az emberek nagy százaléka valamilyen formában már használ ilyen készüléket. A vak és látássérült felhasználó számára ezek a funkciók nem, vagy csak nehézkesen elérhetőek, emellett további problémát okoz az érintőképernyő megfelelő kezelése is, hiszen a billentyűzettel rendelkező telefonokkal ellentétben, ebben az esetben mindössze egy tenyér méretű felület áll a felhasználók rendelkezésére, ami számukra nem teszi lehetővé a készülék megfelelő, kényelmes használatát.

Ezek kiküszöbölésére megoldást jelenthet egy olyan rendszer fejlesztése, amely mesterséges szövegfelolvasó és beszédfelismerő segítségével egy olyan környezet valósít meg, amely, lehetővé teszi a vak és látássérült felhasználó számára, hogy a készüléken található SMS, MMS és e-mail funkciókat maradéktalanul tudja használni. Ilyen lehet egy beérkező SMS, MMS, vagy egy e-mail üzenetek jelzése rezgés, ill. hang formájában, annak felolvasása, ill. üzenetek létrehozása, majd elküldése a címzettnek.

Egy ilyen rendszer fejlesztéséhez elengedhetetlen más, hasonló működésű alkalmazások vizsgálata. Ehhez néhány, már a piacon megtalálható SMS, ill. e-mail felolvasó alkalmazást töltöttem

le, és teszteltem. Ezek néhány pozitív, ill. negatív jellemzőjét próbáltam összegyűjteni, figyelembe véve azt, hogy egy vak és gyengén látó felhasználó, hogyan lenne képes használni az alkalmazást.

A tesztelt programok a következők:

Google Android: SMS Reader, SMS Popup, SMS Speaker, K-9 Mail, Handcent SMS, GO SMS Pro

Apple iOS: Email'nDrive - Hands Free Email Message Reader, Text'nDrive - Speech To Text Hands Free Message Reader

Symbian: Talking SMS

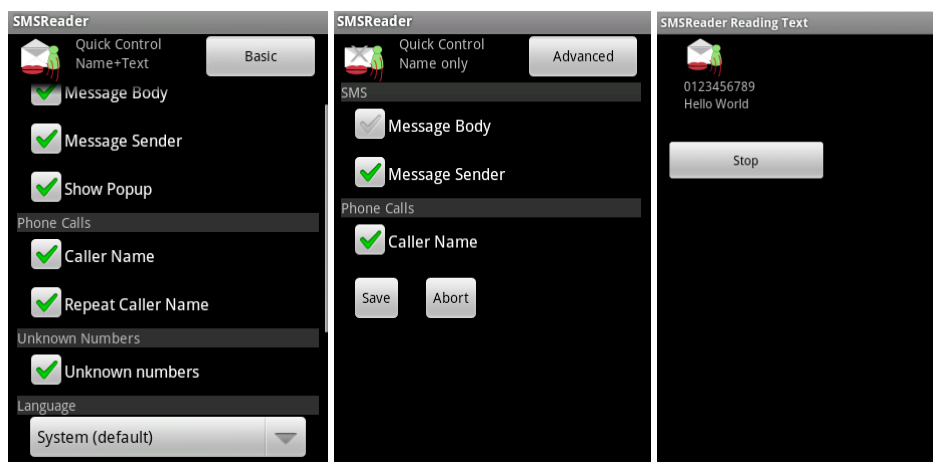
2.7.1 Google Android

2.7.1.1 SMS Reader

Leírás:

Az alkalmazás a bejövő SMS üzenetek figyelője, a felhasználó beállításai szerint felolvassa annak törzsét, és/vagy annak küldőjét, ill. képes egy bejövő hívásnál a hívó fél nevének kimondására.

Képek a programról:



2-11. ábra Az SMS Reader alkalmazás felhasználói felülete

Előnyök:

- A program fejlesztői változatos beállítási lehetőségeket kínálnak. Ilyen például az értesítések beállításai. Beállíthatjuk, hogy SMS üzenet érkezésekor az üzenet mely részeit olvassa fel, ill., hogy jelenjen-e meg felugró ablak az eseményről.
- Lehetőséget nyújt nyelvi beállításokra.
- Megállítható az üzenet felolvasása.

Hátrányok:

- A csendes üzemmódot figyelmen kívül hagyja. Az üzeneteket ugyanúgy felolvassa, ill. hívás esetén a hívó nevét is kimondja.
- Nincs megoldva az e-mailek kezelése.
- Nem biztosít megfelelő felhasználói felületet egy vak és gyengén látó felhasználó számára.

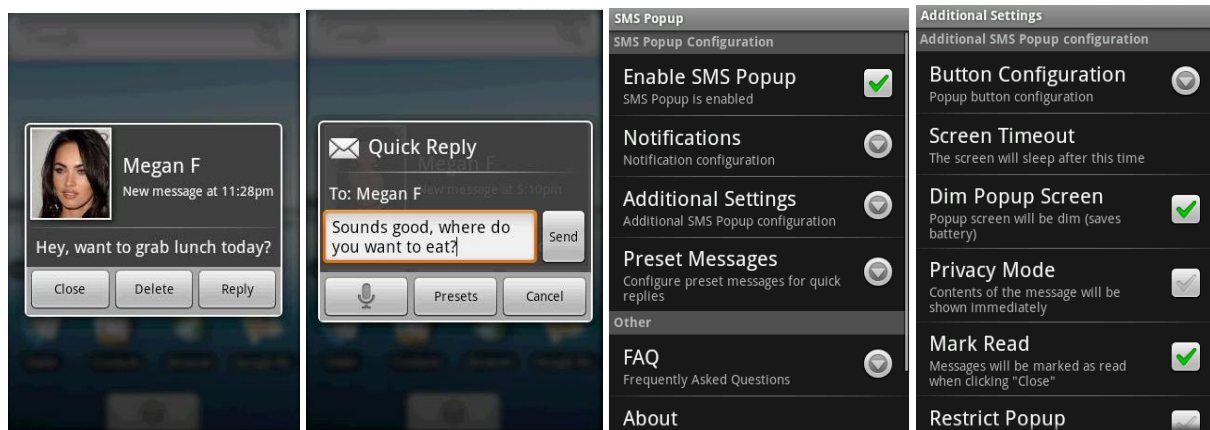
- Nem egy komplett üzenetkezelő rendszer, csak annak egy töredéke. Mindig az utoljára beérkezett üzenetet olvassa fel.
- Nem jeleníthetők meg vele üzenetek.

2.7.1.2 SMS popup

Leírás:

Egy felugró ablak segítségével rögtön olvashatóvá, és kezelhetővé válnak az üzenetek. Új üzenet érkezésekor bezárhatjuk azt, törölhetjük, vagy azonnal válaszolhatunk is rá, emellett alkalmunk van az üzeneteket felolvasni a Text-To-Speech modul segítségével is.

Képek a programról:



2-12. ábra Az SMS popup alkalmazás felhasználói felülete

Előnyök:

- Új üzenet érkezését vibrálással, hanggal, és/vagy fényjelzéssel jelzi.
- Kényelmesen használható felhasználói felület.
- Üzenet érkezésekor egy dialógus ablak jelenik meg, ahol azonnal törölhetjük, ill. válaszolhatunk a beérkező üzenetre.
- Üzenet írásakor használható az ASR beszédfelismerő motor.
- Változatos beállítási lehetőségek.

Hátrányok:

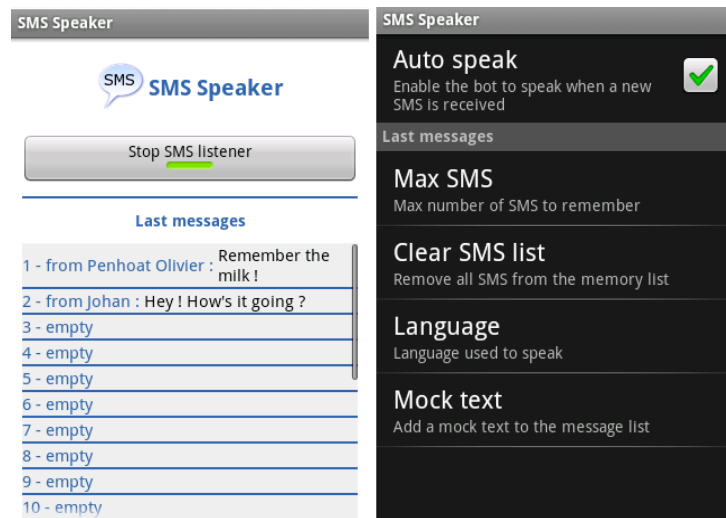
- Ha több SMS érkezik, mindig csak az utolsót mutatja.
- Nincs megoldva az e-mailek kezelése.
- Nem biztosít megfelelő felhasználói felületet egy vak és gyengén látó felhasználó számára.
- Nem egy komplett üzenetkezelő rendszer, csak egy felugró ablak, amely segítségével gyorsan válaszolhatunk az éppen beérkezett üzenetre.

2.7.1.3 SMS speaker

Leírás:

Az SMS speaker egy olyan alkalmazás, amely figyeli a beérkező SMS-eket, kimondja a küldő nevét, ha megtalálható a telefonkönyvben, ill. kilistázza a beérkezett üzeneteket.

Képek a programról:



2-13. ábra Az SMS Speaker alkalmazás felhasználói felülete

Előnyök:

- Leállítható az SMS felolvasása.
- Küldő nevének kimondása, ha az megtalálható a névjegyzékben.
- Mutatja a beérkezett üzenetek listáját.
- Nem olvas fel, amíg a felhasználó telefonál, vagy csendes üzemmódban van.

Hátrányok:

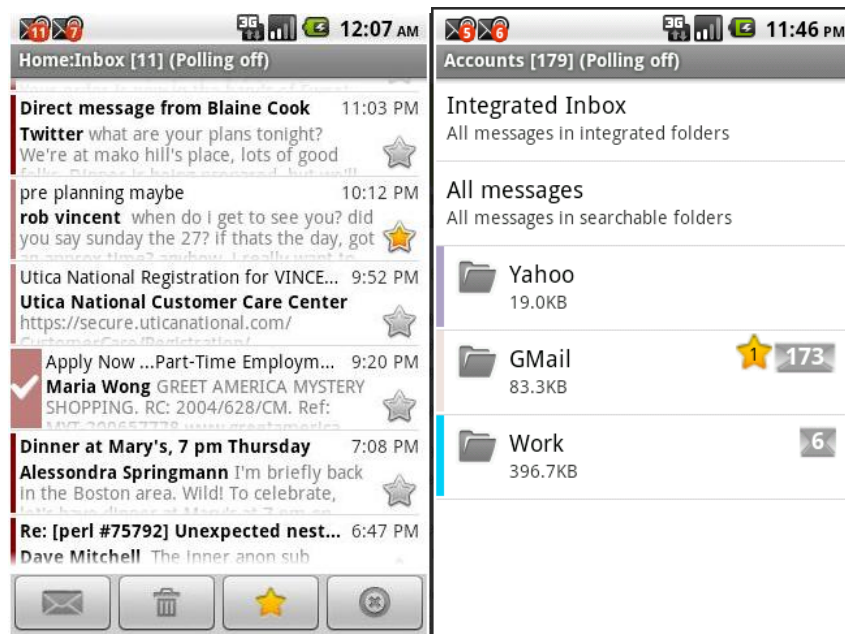
- A felhasználó kevés opció közül választhat.
- Szegényes felhasználói felület.
- Nincs megoldva az e-mailek kezelése.
- Nem képes üzenetek írására, küldésére.
- Nem biztosít megfelelő felhasználói felületet egy vak és gyengén látó felhasználó számára.
- Nem egy üzenetkezelő rendszer, hiszen csak a beérkező üzenetek figyeli, és megjeleníti azokat.

2.7.1.4 K-9 Mail

Leírás:

Egy nyílt forráskódú e-mailkliens kereséssel, mappák szinkronizálásával és így tovább. A K-9 támogatja az IMAP-ot, POP3-at és az Exchange 2003/2007-et is. Ez az alkalmazás teljes mértékben megvalósítja azt, amit egy e-mail klienstől elvárhat a felhasználó.

Képek a programból:



2-14. ábra A K-9 Mail alkalmazás felhasználói felülete

Előnyök:

- Minden email funkciót megvalósít.
- Több email fiókkal is bejelentkezhetünk, amelyeket külön mappába rendez
- Új üzenetek érkezésekor értesítés küldése.
- Előre beállított aláírást állíthatunk be az e-mailjeinknek

Hátrányok:

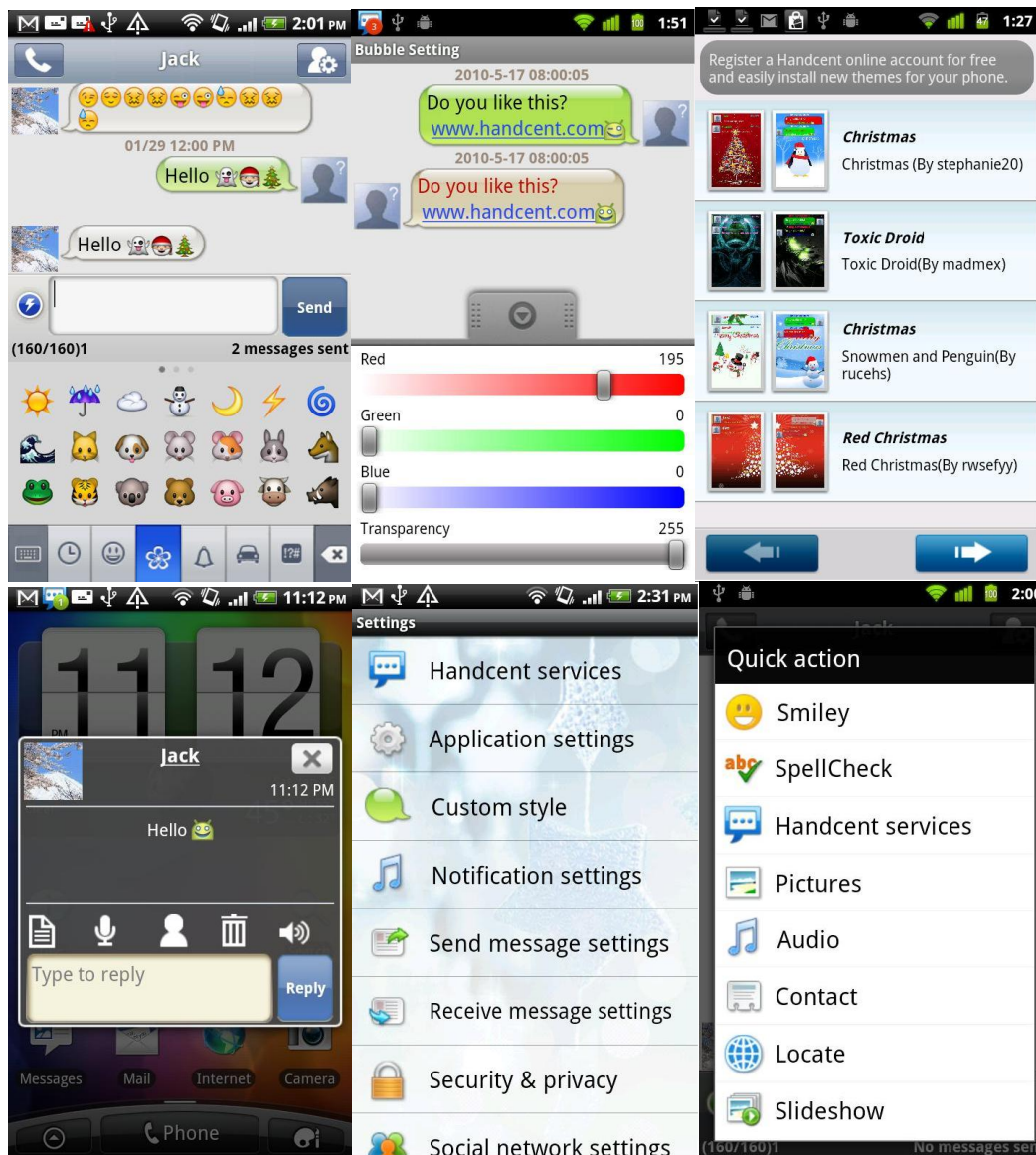
- Nem képes üzenetek felolvasására.
- Nem biztosít megfelelő felhasználói felületet egy vak és gyengén látó felhasználó számára.
- Üzenet írásakor a felhasználó nem használhat ASR modult.
- Nem kezeli az SMS üzeneteket.

2.7.1.5 Handcent SMS

Leírás:

Talán a legnépszerűbbnek mondható SMS kezelő alkalmazás a Market-en (eddig 393.000 felhasználó töltötte). A program rengeteg funkcióval rendelkezik, és szinte minden igényt kielégít, ami elvárható egy SMS üzeneteket kezelő alkalmazástól.

Képek a programból:



2-15. ábra A Handcent alkalmazás felhasználói felülete

Előnyök:

- SMS-ek, MMS-ek kezelése.
- Integrált helyesírás ellenőrzés.
- Különböző üzenetküldő protokollok és nyelvek támogatása.
- Az SMS-ek és MMS-ek elmenthetőek a My Handset Online segítségével. Ezek később természetesen visszaállíthatóak, például újratelepítés esetén.
- Feketelista vezethető azokról az ismerősökről, telefonszámokról, akitől nem szeretnék üzeneteket fogadni (ezek a SPAM mappába kerülnek).
- Rengeteg kiegészítő található hozzá (ilyen például az MMS+ ami lehetővé teszi, 250 Mb-nál nagyobb üzenet elküldését, amit az Android alap esetben nem engedne meg).
- egyedi megjelenítés, és üzenetküldő elrendezés alakítható ki.
- támogatja üzenetek felolvasását és a beszédfelismerést is.

Hátrányok:

- Nem kezel e-mail üzeneteket

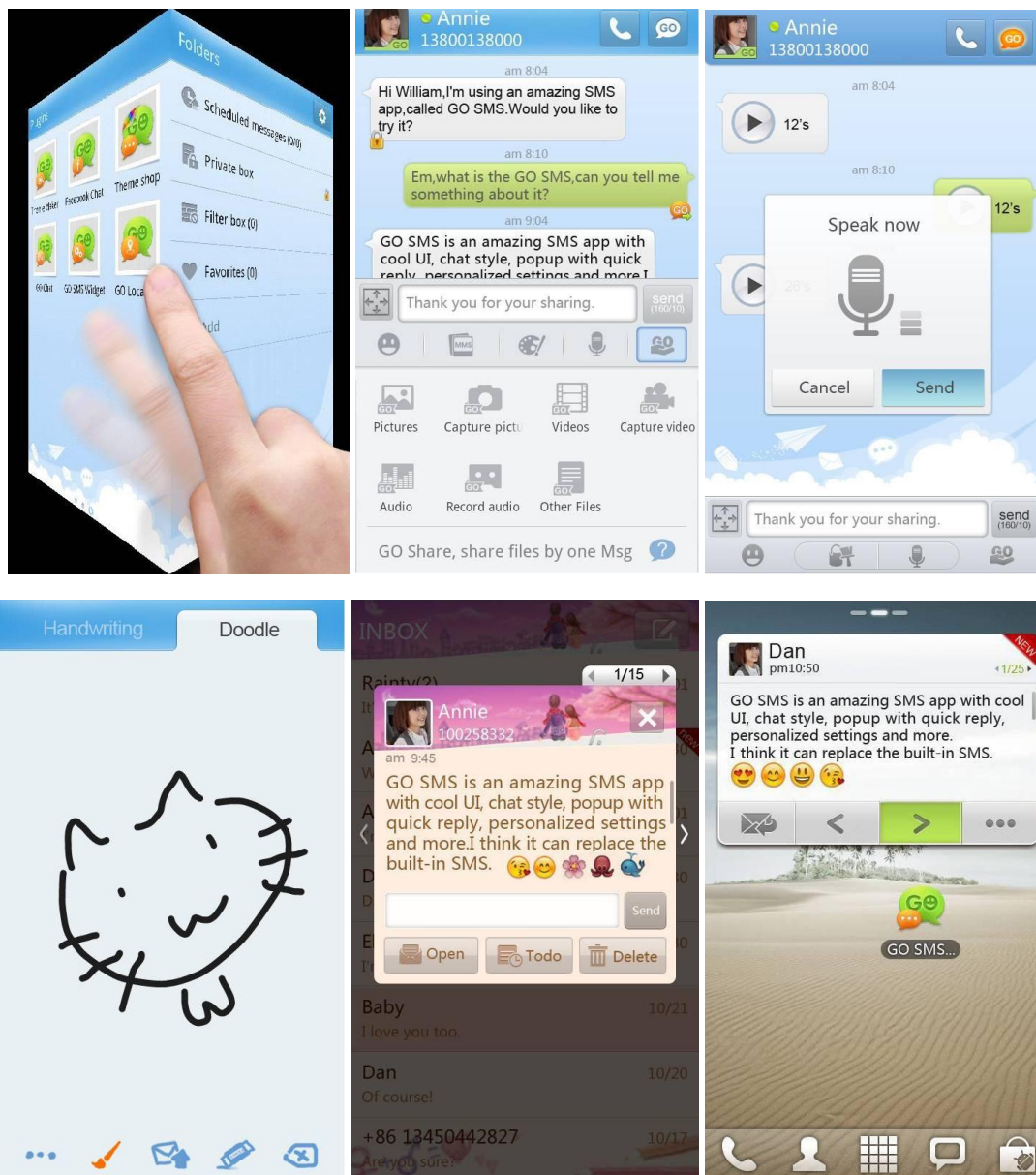
- Nem biztosít megfelelő felhasználói felületet egy vak és gyengén látó felhasználó számára.

2.7.1.6 GO SMS Pro

Leírás:

Egy másik igen népszerűnek mondható SMS/MMS üzenetkezelő alkalmazás. Az SMS és MMS üzenet küldésénél kívül számos kiegészítő funkcióval is rendelkezik. Ilyenek például a Facebook chattámogatás, SMS ütemező, SMS-ek biztonsági mentése és helyreállítása, feketelista támogatása, csoportos üzenet küldése és így tovább.

Képek a programból:



2-16. ábra A GO SMS Pro alkalmazás felhasználói felülete

Előnyök:

- Facebook csevegés támogatása.
- TTS (Text-To-Speech) modul támogatása.
- Használható az ASR modul üzenetek írásakor.

- Több nyelv támogatása.
- Rengeteg kiegészítő funkció áll rendelkezésre (például hangulatjelek hatalmas választéka).
- Facebook profilképek szinkronizálása
- Használható feketelista.
- Üzenetek biztonsági mentése, és visszaállítása.

Hátrányok:

- Nem kezel e-mail üzenetek
- Nem biztosít megfelelő felhasználói felületet egy vak és gyengén látó felhasználó számára.

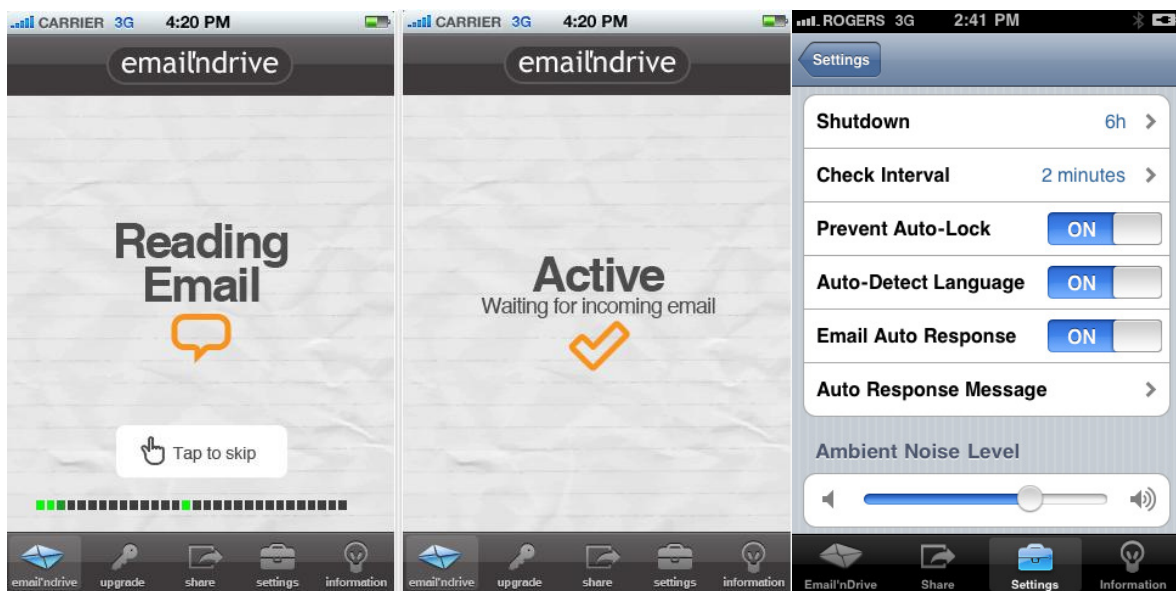
2.7.2 Apple iOS

2.7.2.1 Email'nDrive - Hands Free Email Message Reader

Leírás:

Egy olyan alkalmazás, amely vezetés közben tájékoztatja a felhasználót új e-mail üzenet érkezéséről. A program képes beérkező email üzenetek felolvasására a TTS modul, ill. válaszüzenet írására az ASR motor segítségével.

Képek a programból:



2-17. ábra Az Email'nDrive - Hands Free Email Message Reader felhasználói felülete

Előnyök:

- Bejövő e-mail üzenetek felolvasása a TTS modul segítségével.
- Üzenet írása az ASR motor segítségével

Hátrányok:

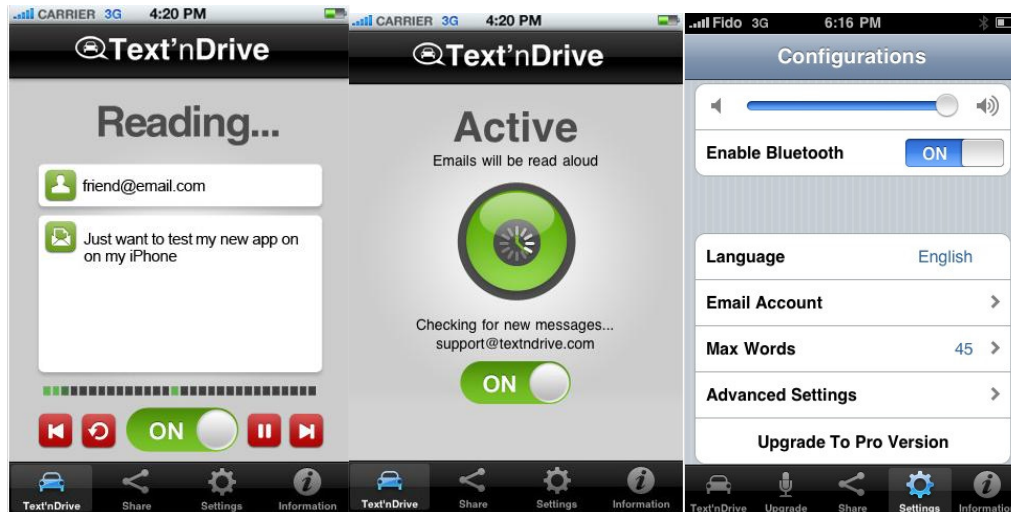
- Nem kezeli az SMS üzenetek.
- Nem egy komplett e-mail kliens, mindig csak az utoljára beérkezett email üzenetet kezelésére szolgál.
- Nem kezel SMS üzeneteket.
- Csak részben biztosít megfelelő felhasználói felületet egy vak és gyengén látó felhasználó számára.

2.7.2.2 Text'nDrive - Speech To Text Hands Free Message Reader

Leírás:

Az Email'nDrive programhoz hasonlóan működik. Figyelmeztet újonnan érkezett e-mail üzenetet esetén. Képes beérkező email üzenetek felolvasására a TTS modul, ill. válaszüzenet írására az ASR modul segítségével.

Képek a programból:



2-18. ábra A Text'nDrive - Speech To Text Hands Free Message Reader alkalmazás felhasználói felülete

Előnyök:

- Bejövő e-mail üzenetek felolvasása.
- Automatikus értesítés üzenet küldése, hogy a felhasználó éppen vezet-e.
- Üzenet írása az ASR motor segítségével.
- Beérkezett üzenet felolvasása.

Hátrányok:

- Nem kezeli az SMS üzenetek.
- Nem egy komplett e-mail kliens, mindig csak az utoljára beérkezett email üzenetet kezelésére szolgál.
- Csak részben biztosít megfelelő felhasználói felületet egy vak és gyengén látó felhasználó számára.

2.7.3 Symbian

2.7.3.1 Talking SMS

Leírás:

Egy SMS felolvasó program, amely figyeli a beérkező üzeneteket, és azonnal kimondja a küldő nevét, és felolvassa a teljes SMS-t.

Képek a programból:



2-19. ábra A Talking SMS alkalmazás felhasználói felülete

Hátrányok:

- Nem egy komplett üzenetkezelő rendszer, csak egy alkalmazás, amely felolvassa az utolsóként beérkezett SMS üzenetet.
- Nem biztosít megfelelő felhasználói felületet egy vak és gyengén látó felhasználó számára.
- Az e-mail üzenet kezelését nem támogatja.

3. Saját email és SMS felolvasó rendszer specifikációja

Az előző pontban vizsgáltam néhány alkalmazást, amely üzenetek kezelésével foglalkozik. Mivel egyik sem kifejezetten vakok és gyengén látók számára készült, ezért az esetek többségében hiányzott a számukra elvárt felhasználói felület, amely lehetővé teszi a rendszer egyszerű használatát. Ilyen volt például, hogy a beérkezett üzeneteket valamilyen listában jelenítette meg, vagy egyazon oldalon egymás alatt több komponenst helyezett el az alkalmazás fejlesztője. Ezekben az esetekben nem biztosítható, hogy egy vak és látássérült felhasználó gond nélkül használhassa a rendszert, hiszen könnyedén eltévesztheti a kívánt menüpont, vagy üzenet elérését. Másik problémát jelenthet, hogy a felhasználó nem kap visszajelzést az alkalmazás használata közben hang formájában (például a programok nem jelezték egy új menüpont kiválasztását). Sok esetben hiányosság volt, hogy az alkalmazás nem egy komplett üzenetkezelő rendszert valósított meg, hanem annak csak egy töredékét, például felolvasta a beérkező üzenetet, viszont a régebben fogadottakat már nem tudta kezelni, vagy esetleg hiányzott belőle a küldés funkció.

Mivel olyan alkalmazást nem találtam, amely az SMS és az e-mail üzenetek kezelését egyszerre oldaná meg, és emellett használható felhasználói felületet is kínálna a vakok és gyengén látó emberek számára, ezért célszerűnek tartom egy ilyen rendszer fejlesztését.

Az okostelefonok egyik fontos funkciója, hogy rendelkeznek internet kapcsolattal, így tehát e-maileket és egyéb szöveges üzenetek is olvashatunk velük. Sajnos azonban ezek használata a vak és gyengén látó emberek számára nem megoldható segítség nélkül, ugyanis ellentétben a hagyományos mobiltelefonokkal itt mindösszesen egy tenyérnyi méretű képernyő és néhány gomb áll a felhasználó rendelkezésére.

Fontos tehát egy olyan rendszer kialakítása, amely egyszerre képes e-mailek és SMS-ek fogadására, küldésére, és menedzselésére, és emellett még ergonomikus felhasználói felületet is kínál a célközönség számára.

A rendszer az alábbi funkciókat képes majd ellátni:

- Beérkező üzenetek figyelése.
- Beérkező üzenet jelzése rezgés, és hangüzenet formájában.
- e-mail és SMS üzenetek szétválasztása.
- Üzenetek kezelése mappák szerint (Beérkező, Elküldött, Piszkozatok).
- Üzenetek hangos felolvasása TTS (Text-To-Speech modul segítségével).
- Új üzenet írásának / küldésének lehetősége (ASR motor segítségével).
- Megfelelő felhasználói felület vakok és látás sérült felhasználók számára.
- Névjegyzék szinkronizáció.
- Csendes üzemmód esetén a riasztás későbbre halasztása.

4. A rendszer megvalósítása

4.1 mobil platform választása

A 3.3-as fejezetben már említettem, hogy a legnépszerűbb mobil operációs rendszerek a Google Android, az Apple iOS, és a Symbian. A választásom az Android platformra esett, mert azon felül, hogy manapság az egyik legnépszerűbb mobil operációs rendszer, a statisztikák szerint terjedése rohamos növekedést mutat, a piaci részesedése folyamatosan növekszik, a gyártók nagy többsége ezt használja, és nem utolsósorban manapság már elérhető áru készülékek is találhatóak a piacon.

4.2 Egy Android alkalmazás felépítése

4.2.1 Biztonsági megoldások Android platformon

Az Android operációs rendszer egy több-felhasználós Linux rendszer, ahol minden alkalmazás különböző felhasználónak tekinthető.

Alapvetően a rendszer minden alkalmazást egy egyedi Linux felhasználó azonosítóval lát el (ez az azonosító csak a rendszer által használható és az alkalmazás számára ismeretlen). A rendszer engedélyeket köt az összes fájlhoz az alkalmazáson belül, lehetővé téve, hogy csak az alkalmazással megegyező azonosítóval rendelkező alkalmazás férjen hozzájuk. Minden egyes folyamatnak saját virtuális gépe van (VM-Virtual Machine), így minden egyes alkalmazás kódja a többitől függetlenül fut. Alapvetően minden alkalmazás a saját Linux folyamatán belül fut. Az operációs rendszer akkor indít el egy folyamatot, amikor minden alkalmazáshoz tartozó komponens rendelkezésre áll, ami a futásához szükséges. Befejezni pedig akkor fogja, amikor már nincs rá szükség, vagy a rendszer más alkalmazások számára szabadít fel memóriát.

Ilyen módon az Android operációs rendszer a legkisebb jogosultság elvét valósítja meg. Alapvetően ugyanis minden alkalmazás csak az általa igényelt erőforrásokhoz fér hozzá. Ez az elgondolás egy nagyon biztonságos környezetet eredményez, ahol az alkalmazások nem férnek hozzá a rendszer azon részeihez, ahová nincs jogosultságuk.

Azonban létezik megoldás arra is, hogy az alkalmazások megoszthassanak adatokat más alkalmazásokkal, vagy hozzáférjenek a rendszer szolgáltatásaihoz. Lehetséges ugyanis, hogy két alkalmazás azonos Linux felhasználói azonosítót osszon meg egymással. Ebben az esetben képesek lesznek hozzáférni egymás fájljaihoz. A rendszer erőforrásainak kímélése érdekében lehetőség van az azonos felhasználói azonosítóval rendelkező alkalmazásokat azonos Linux folyamatban futtatni és megosztani a hozzájuk tartozó VM-et is (ebben az esetben azonban az alkalmazásoknak azonos bizonyítvánnyal is kell rendelkezniük).

Az alkalmazások hozzáférési jogot is kérhetnek a készüléken található adatokhoz, például a telefonon található névjegyzékhez, SMS üzenetekhez, SD kártyához, kamerához, Bluetooth-hoz, és így tovább, azonban ezekhez telepítéskor a felhasználónak engedélyt kell adnia.

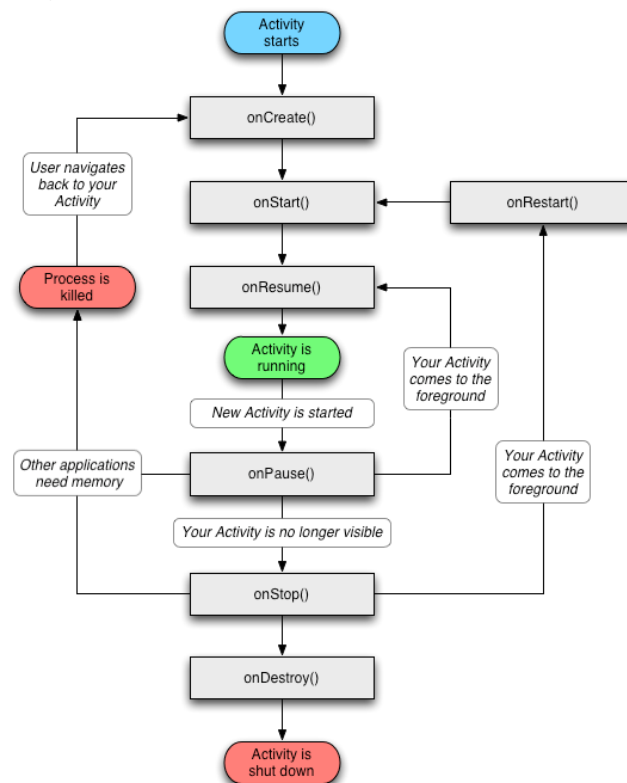
4.2.2 Alkalmazás komponensek

Az alkalmazás összetevők alapvetői építőkövei egy Android alkalmazásnak. Minden egyes komponens egy külön pont, amin keresztül az operációs rendszer hozzáférhet a fejlesztő által írt alkalmazáshoz. Négy különböző alkalmazás összetevő létezik. Mindegyik egyedülálló céllal, és életciklussal rendelkezik, amely meghatározza, hogy az alkalmazás mikor jön létre, és pusztul el.

A négy alkalmazás komponens a következő:

1)Activity

Az Activity osztályok egy-egy képernyőt reprezentálnak a mobil eszköz kijelzőjén. Egyszerre mindig csak egy látható a kijelzőn, de természetesen egy alkalmazáshoz több képernyő is tartozhat, amelyek futás közben események hatására cserélhetünk. Ekkor az Activity-k különböző eseményei és állapotai hívódnak meg (ld. 4-1. ábra).



4-1. ábra Az Android Activity életciklus

onCreate:

Ez a metódus akkor hívódik meg, amikor az alkalmazás létrejön. Általában ezen belül hozzuk létre a felhasználói felületet, inicializáljuk a változóinkat, rendelünk eseményeket az Activity gombjaihoz és így tovább.

onStart:

Ez a metódus egy Activity életciklusa alatt többször is meghívódhat. Első alkalommal a létrehozás után, a többi alkalommal pedig közvetlenül az `onRestart()` metódus után.

onResume:

A metódus meghívása közvetlenül az Activity képernyőre kerülése előtt, vagy előtérbe kerülése után történik (ezt egy `onRestart()` és egy `onStart()` metódushívás előzi meg).

onPause:

Ez a metódus közvetlenül azután hívódik meg, ha az alkalmazás egy új Activity-t indított el. Ebből az állapotból lehetséges:

- onResume() metódushívás, abban az esetben, ha az újonnan meghívott Activity futása befejeződött.
- hogy, más alkalmazás erőforrás igénye miatt az ütemező kilövi. Ilyenkor onCreate() metódushívás következik, ha felhasználó ismételten erre a nézetre navigál.
- onStop() metódushívás, ha az Activity már nem látható.

onStop:

Ez a metódus akkor hívódik meg, amikor egy másik Activity kerül előtérbe. Az Activity lényegében ezzel a metódushívással értesül arról, hogy a háttérbe szorították. Itt is előfordulhat, hogy az ütemező memóriahiány miatt kilövi, majd később szükség esetén újra létrehozza.

onRestart:

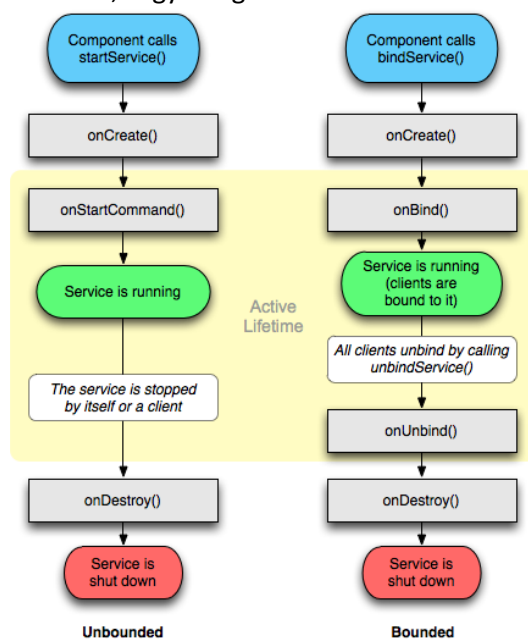
Az onStop hívása után kerül meghívásra, ha a háttérbe szorított Activity újra előtérbe kerül.

onDestroy:

Ha bezárják az alkalmazásunkat, akkor meghívódik ez a metódus, amely lefutása után a platform felszabadítja a lefoglalt memóriaterületet. Ebbe az állapotba csak az onStop metódus meghívása után kerülhet az Activity.

2) Service

Sok alkalmazásnál szükség lehet arra, hogy bezárt ablakkal is képes legyen a háttérben futni. Az Android erre kínálja az ún. Service-ek, vagy szolgáltatások használatát.



4-2. ábra Az Android Service életciklusa

Mint ahogyan a 4-2. ábrán látható a service komponensnek két különálló életciklusa is létezik. (7-2. ábra) A baloldalon látható az a Service, amelyet startService()-el, a jobb oldalon pedig, amelyet bindService()-el hoztak létre. Működésük hasonló, az Activity-nél leírtakkal.

3) Content provider

Content Provider lehetőséget nyújt arra, hogy az adatok állományokba vagy SQLite adatbázisba menthetőek legyenek ill., hogy két alkalmazás adatokat cserélhessen egymással.

4) Broadcast receiver

A rendszer ezen a komponensen keresztül tájékoztat alkalmazásokat bizonyos események bekövetkezéséről, például képernyő kikapcsolásáról, az akkumulátor alacsony töltésszintjéről, vagy fényképek készítéséről. Alkalmazások is kezdeményezhetnek ilyen üzenetszórásokat, például egy alkalmazás értesíthet másokat, hogy bizonyos adatok letöltésre kerültek és az elérhetőek a számukra.

4.3 Az üzenetkezelő rendszer működési elvének megtervezése

Az üzenetkezelő rendszeremet úgy terveztem, hogy a vak és látássérült felhasználó egyszerre tudja kezelni az SMS és e-mail üzeneteit. A 2.4-es pontban már vizsgáltam és elemeztem néhány a piacon megtalálható üzenetkezelő alkalmazást. Ezek egyike sem rendelkezett olyan felhasználói felülettel, ami megfelelt volna egy vak és gyengén látó felhasználó számára és emellett egyszerre tette volna lehetővé SMS-ek és e-mail üzenetek menedzselését.

A végső cél tehát egy olyan alkalmazás megtervezése és kifejlesztése volt, ami ezeket a funkciókat ellátja, és a vak és gyengén látó felhasználó számára is használható megoldást nyújt. Ebben a fejezetben inkább a belső működésre, mintsem a felhasználó felület leírására szeretnék koncentrálni (ezt a 7.7-es fejezetben teszem meg).

A fejlesztett üzenetkezelő rendszer egyszerre valósít meg egy SMS, és egy e-mail klienst alkalmazást. Ezek egymástól függetlenül működnek, nem kommunikálnak egymással, azonban közös jellemzőjük, hogy figyelik a beérkező üzeneteket, ill. képesek a beérkezett, elküldött és a piszkozatok mappák kezelésére.

Fontosnak tartottam, hogy a rendszer minden olyan feladatot el tudjon látni, amit a telefonon megtalálható gyári üzenetkezelő alkalmazások is tudnak, sőt ezeket kiegészítve próbáltam olyan plusz funkciókkal ellátni, amik segíthetik a vak és látás sérült emberek mindennapjait. Ilyenek például a TTS modul, vagy a beszédfelismerő motor használata is, amelyek üzenetek felolvasásakor, navigálásakor, vagy üzenetek írásakor nyújtanak segítséget.

4.4 Az üzenetkezelő rendszer fejlesztése során szerzett tapasztalatok

Egy üzenetkezelő rendszer működésének megtervezésénél/elkészítésénél számos apró problémát meg kell oldanunk. Ilyenek például az újonnan érkezett üzenetek figyelésének módja, a beérkezett üzenetek kezelése, törlése, továbbküldése, új üzenet létrehozása, félbehagyott üzenet írás esetén azok piszkozatok mappába való elmentése, és így tovább. Különös kihívást jelentett a felhasználói felület megtervezése, amit úgy kellett kialakítani, hogy a vak és gyengén látó felhasználó számára is használható megoldást nyújtson.

Habár a manapság kapható okostelefonok fejlett hardverrel rendelkeznek, mégis fontos figyelembe vennünk, hogy ezeken a készülékeken jóval korlátozottabb méretű erőforrások találhatóak, mint egy asztali számítógépen. Fontos lehet tehát figyelembe venni azt is, hogy melyek azok a költségesebb feladatok, amelyek futtatását célszerűbb esetleg háttérszálon futtatni, hiszen ellenkező esetben előfordulhat, hogy az alkalmazás ablaka megfagy, vagy lassabban működik az elvártnál.

A Google Android fejlett SDK csomagot (Software Development Kit) kínál a fejlesztők számára, azonban ez nem tartalmazza például az e-mail üzenetek kezelését. Utóbbi megvalósításában a JavaMail API volt a segítségemre.

A problémák felvezetése után talán az üzenetkezelő rendszer egyik legalapvetőbb funkciójának megvalósításával kezdeném, az újonnan érkezett üzenetek fogadásával. A beérkező SMS és e-mail üzeneteket az üzenetkezelő rendszerem különböző módon észleli. Az Android operációs rendszer új esemény bekövetkeztekor üzenetekkel (Intentekkel) értesíti a futó programokat. A Broadcast receiver komponens lehetővé teszi ezeknek az eseményeknek a figyelését. Ilyen például az „android.provider.Telephony.SMS_RECEIVED” esemény is, amelyre egy szűrőt beállítva, észlelhetővé és megfelelően lekezelhetővé válnak a beérkező SMS üzenetek.

Az e-mail üzenetek fogadása sajnos már nem ilyen triviális feladat. Ennek megvalósításához a JavaMail API-ra volt szükségem, amit az SDK alapvetően nem tartalmaz, emellett ellentétben az SMS üzenetek kezelésével ebben az esetben az új üzenetek érkezésének figyelése nem volt megoldható egyszerű szűrőkön keresztül. A probléma megoldásában az AlarmManager osztály volt a segítségemre, aminek feladata, hogy ütemezhetővé tegyen egy alkalmazást, tehát, előre meghatározott időközönként meghívja annak bizonyos metódusait. A másik probléma, hogy frissítéskor az alkalmazás minden alkalommal a szervertől tölt le információkat, ami költséges feladat, tehát ennek a háttérszálon való futtatását is meg kellett oldanom.

Egy üzenetkezelő rendszer feladata az üzenetek fogadásán és küldésén kívül, hogy azokat rendszerezetten tárolja. A fejlesztés során próbáltam minél nagyobb hangsúlyt fektetni ezeknek a betartására. Ahogy az üzenetek fogadásánál itt is szembesültem hasonló problémákkal, ugyanis amíg az SMS-ek esetében rendelkezésemre állt az SDK által nyújtott ContentResolver osztály, addig az e-mail üzenetek kezelésénél ugyancsak a javaMail API-ra kellett támaszkodjak, ahol ismét erőforrás problémákba ütköztem.

Összességében a fejlesztés során a legtöbb gond akkor adódott, amikor egy e-mail üzenethez tartozó problémát akartam orvosolni. A fő gondot szinte minden esetben a feladat komplexitásának nagysága okozta, de a legtöbb esetben ez megoldható volt a feladat háttérszálon való futtatásával.

4.5 Az üzenetkezelő rendszert megvalósító osztályok és leírásaik

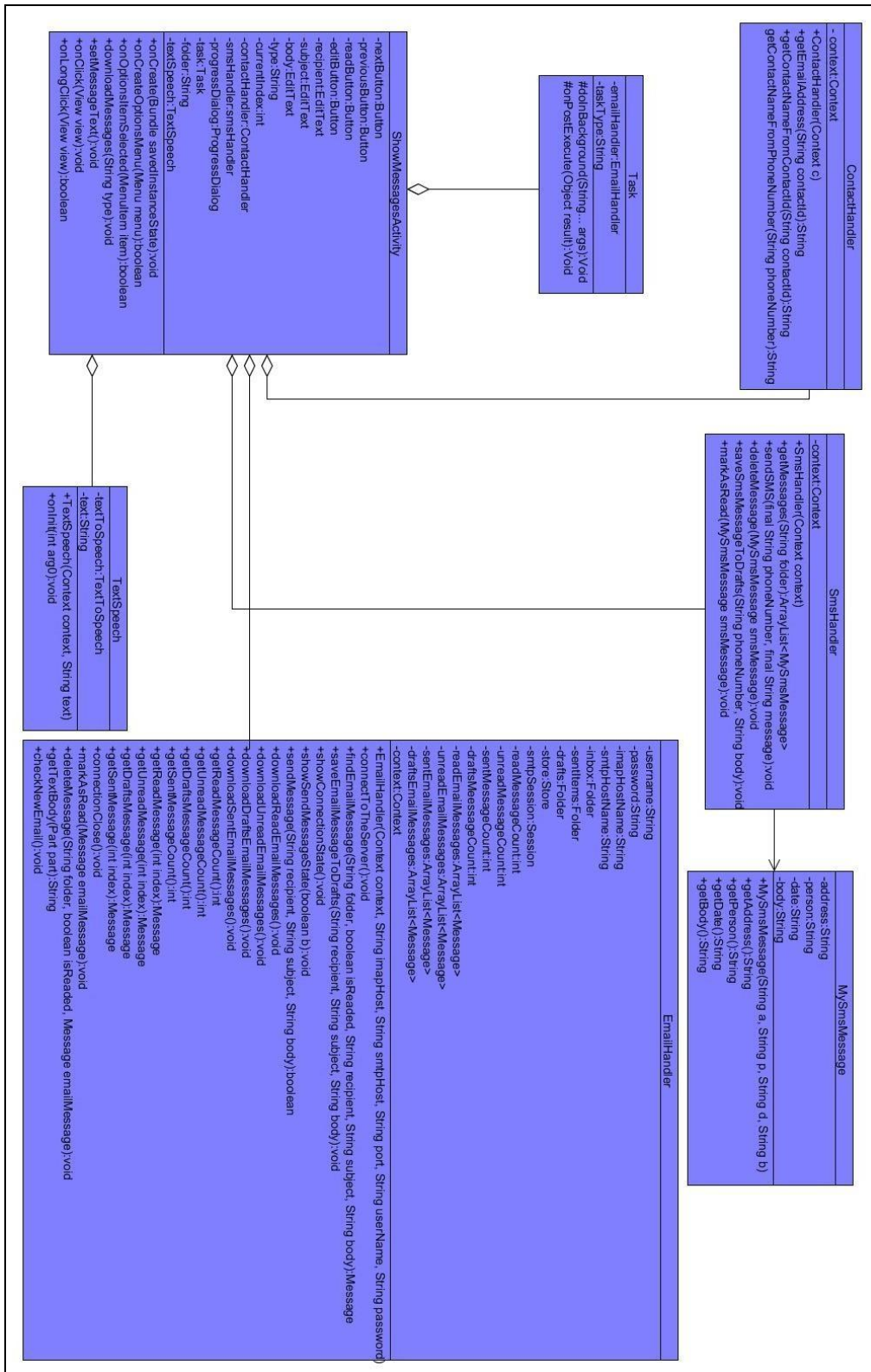
A rendszer jelenleg 24 osztályt tartalmaz.

Ezek listája és rövid leírása:

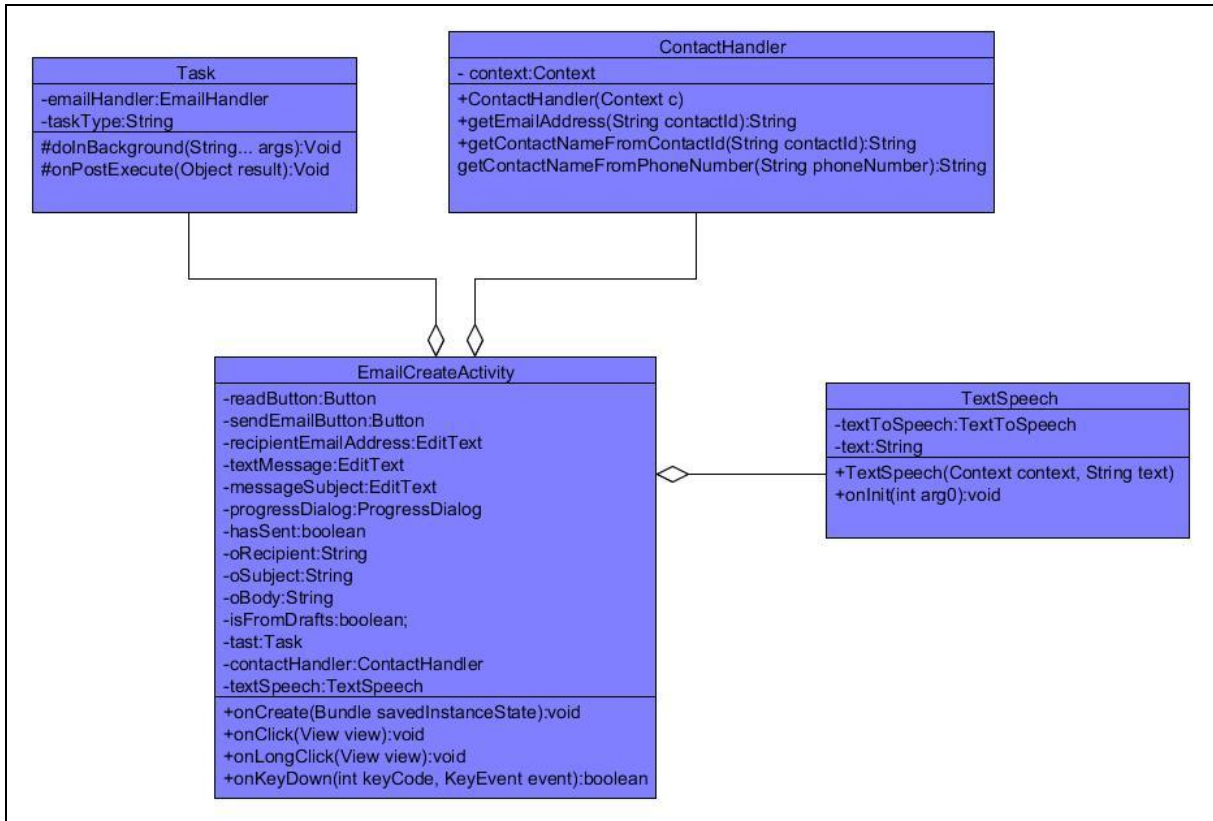
- *TextSpeech*:
Szövegek felolvasásáért felelős osztály, amely a Google Android által nyújtott beépített TTS (Text-To-Speech) modult használja.
- *SpeechRecognizerActivity*:
A szövegfelismerő modult megvalósító osztály. Ez a modul a Google Android által nyújtott beépített ASR (Automatic Speech Recognition) motort használja.
- *SmsReceiver*:
A beérkező SMS üzeneteket figyelő osztály.
- *SmsMenuActivity*:
Az SMS üzenetek főmenüjét megvalósító osztály. A felhasználó itt választhatja ki, hogy a beérkezett, elküldött üzenetek, vagy a piszkozatokat mappát szeretné-e böngészni, vagy új üzenetet szeretné-e létrehozni.
- *SmsInboxActivity*:
A beérkezett SMS üzenetek menüjét megjelenítő osztály. Itt választhat a felhasználó, hogy az olvasott, vagy az olvasatlan üzeneteket szeretné-e megnézni.
- *SmsHandler*:
Az SMS-ek kezeléséért, menedzseléséért felelős osztály.
- *SmsCreateActivity*:
Új SMS létrehozásáért, elküldéséért, vagy a piszkozatok mappába mentéséért felelős osztály.
- *ShowSpeechRecognizerResultActivity*:
A beszédfelismerő által talált egyezések lehetséges listáját megjelenítő osztály.
- *ShowMessagesActivity*:
Az e-mail és SMS üzenetek megjelenítéséért felelős osztály.
- *ShowEmptyFolderActivity*:
Üres mappát megjelenítő nézet.
- *ShowDraftMessagesActivity*:
A piszkozatok mappát megjelenítő osztály.
- *NetworkSettingsActivity*:
Abban az esetben, ha a felhasználó olyan szolgáltatást szeretne igénybe venni, amihez internet kapcsolat is szükséges, akkor ennek az osztálynak a felelőssége, hogy elnavigáljon a hálózati beállításokhoz.
- *MySmsMessage*:
Ez az osztály egy SMS üzenetet valósít meg.
- *MessageNotification*:
Az értesítések megjelenítéséért felelős osztály.

- *MainMenuActivity:*
Az alkalmazás főmenüjét megjelenítő osztály. Itt választhat a felhasználó, hogy az SMS, vagy az e-mail üzeneteit szeretné-e használni.
- *EmailReceiver:*
A beérkező e-mail üzenetek figyeléséért felelős osztály
- *EmailMainMenuActivity:*
Hasonlóan az SmsEmailMenuActivity osztályhoz ez az osztály az e-mail üzenetek főmenüjét valósítja meg. A felhasználó itt választhatja ki, hogy a beérkezett, elküldött üzenetek, vagy a piszkozatokat szeretné-e böngészni, vagy új üzenetet szeretne-e létrehozni.
- *EmailLoginActivity:*
A felhasználó ennek az osztálynak a segítségével jelentkezhet be az e-mail fiókjára.
- *EmailInboxActivity:*
A beérkezett e-mail üzenetek menüjét megjelenítő osztály. Itt választhat a felhasználó, hogy az olvasott, vagy az olvasatlan üzeneteket szeretné-e megnézni.
- *EmailHandler:*
E-mail üzenetek kezeléséért, menedzselésért felelős osztály.
- *EmailCreateActivity:*
Új e-mail üzenet létrehozásáért felelős osztály.
- *ContactHandler:*
A telefonon található névjegyzékeket menedzselő osztály.
- *BootUpReceiver:*
Ez az osztály felelős azért, hogy az operációs rendszer felállása után az üzenetkezelő rendszer is automatikusan elinduljon.
- *Task:*
Bizonyos műveletek háttértáron való futtatásáért felelős osztály.

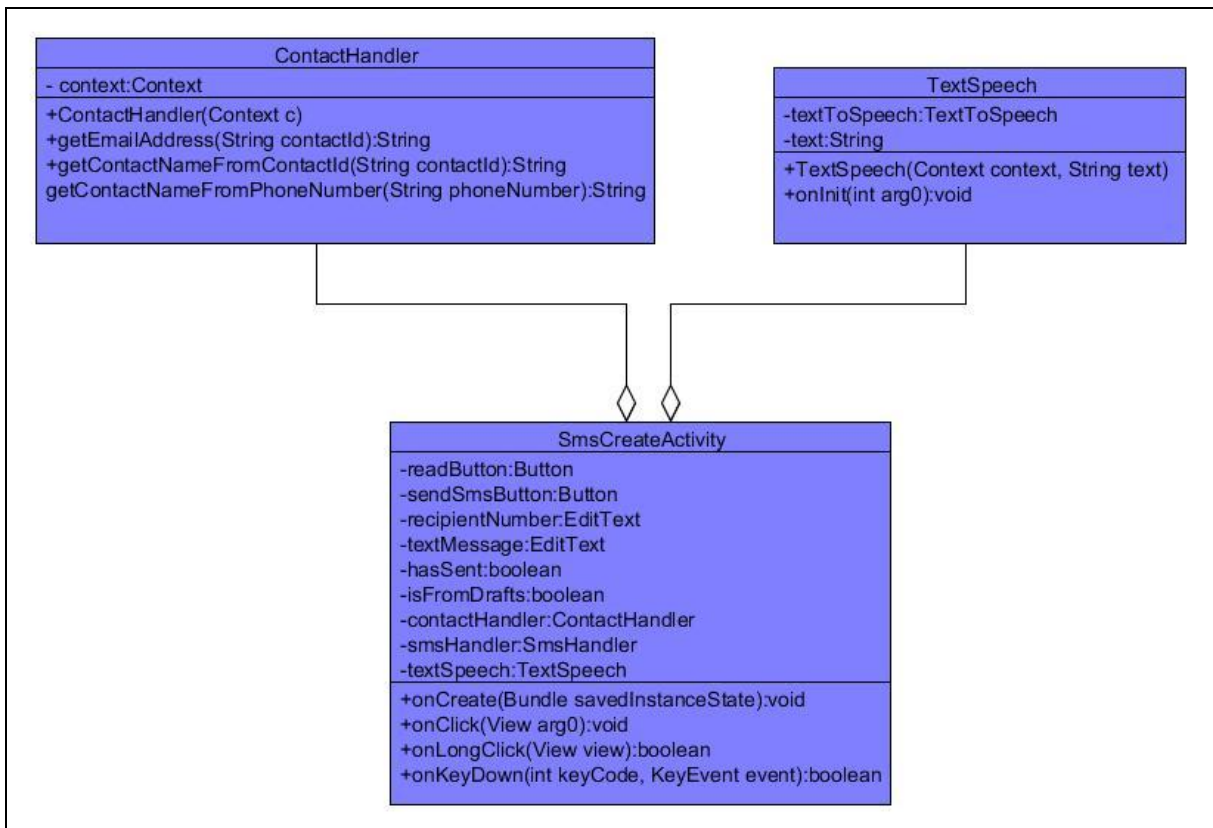
4.6 Az üzenetkezelő rendszer UML diagramja



4-3. ábra A üzenet megjelenítéséért felelős UML osztály diagram



4-4. ábra Új e-mail üzenet létrehozásáért felelős UML osztály diagram



4-5. ábra Új SMS üzenet létrehozásáért felelős UML osztály diagram

4.7 Az üzenetkezelő rendszer nézeteinek megtervezése

Az üzenetkezelő rendszer felhasználói felületét próbáltam úgy megtervezni, hogy egy vak és gyengén látó felhasználó is könnyedén tudja kezelni. A menükben való lépkedéshez főként gombokat használtam. Általában ezeket a képernyő sarkaiban helyeztem el valamilyen formában, hogy könnyedén megjegyezhetőek és használhatóak lehessenek. A navigálást segíti, hogy minden gombot hosszan tartva, az alkalmazás felolvassa annak nevét, ill. a menübe való belépéskor felhasználó információt kap, hogy éppen melyik menüben tartózkodik.

A rendszer indításakor a felhasználó választhat, hogy az SMS, vagy az e-mail funkciókat szeretné-e használni (ld. 4-6. ábra). Ekkor természetesen, már képes a rendszer e-mailek és SMS-ek fogadására. Ezt külön nem kell engedélyezni. Új üzenet érkezését a rendszer hang, és rezgés formájában jelzi a felhasználó számára, ilyenkor, ha talál egyezést a névjegyzékben, kimondja a küldő nevét, és az üzenet típusát. (Természetesen ügyelve arra, hogy a csendes üzemmódban ezt ne tegye meg.)



4-6. ábra Az üzenetkezelő rendszer főmenüje

Attól függően, hogy a felhasználó melyik opciót választotta, bekerül az üzenetkezelő főmenüjébe (ld. 4-7. ábra).

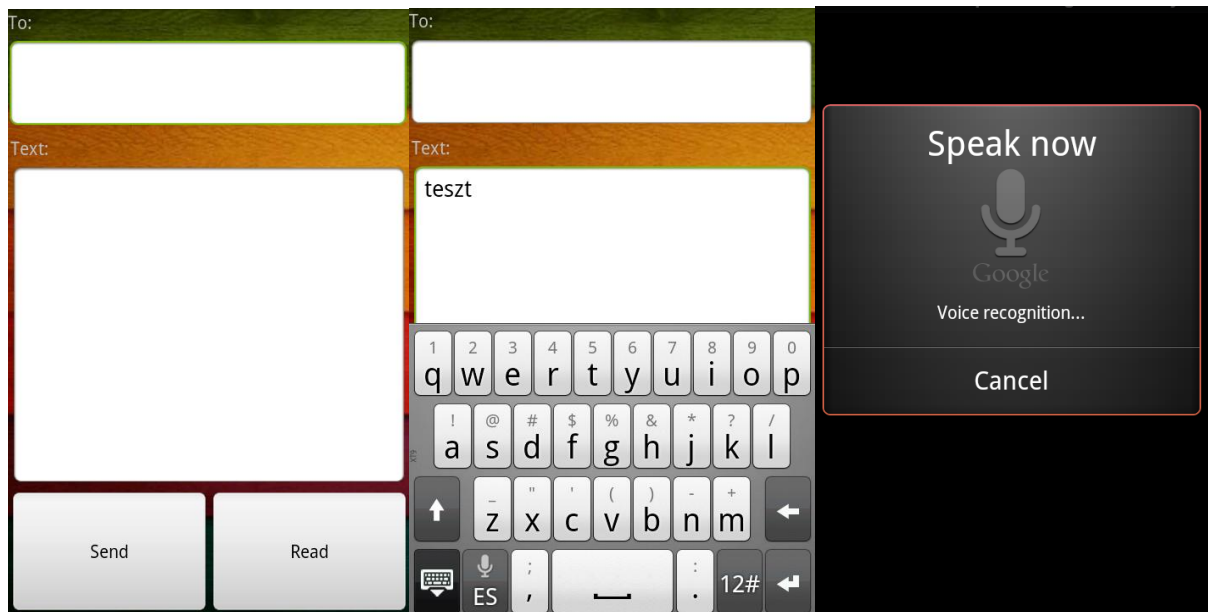


4-7. ábra Az SMS, vagy e-mailkezelő főmenüje

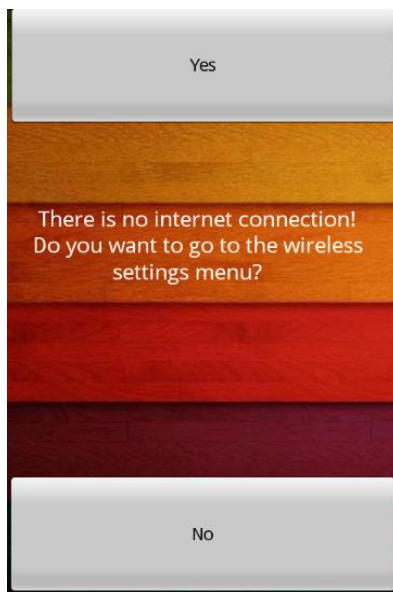
A főmenüben a felhasználó választhat, hogy új üzenet szeretne-e létrehozni, vagy esetleg a bejövő, elküldött, vagy a pizskozatok mappát böngészni.

Új SMS üzenet írásakor a 4-8 ábrák láthatóak látható. Értelmszerűen meg kell adni a címzett, vagy címzettek nevét, és az üzenet törzsét. A címzettek és az üzenet megadására egyaránt használható billentyűzet, ill. a beépített beszédfelismerő. A billentyűzet előhívásához elegendő röviden, a beszédfelismerő elindításához pedig hosszan nyomva tartani a kívánt mezőt.

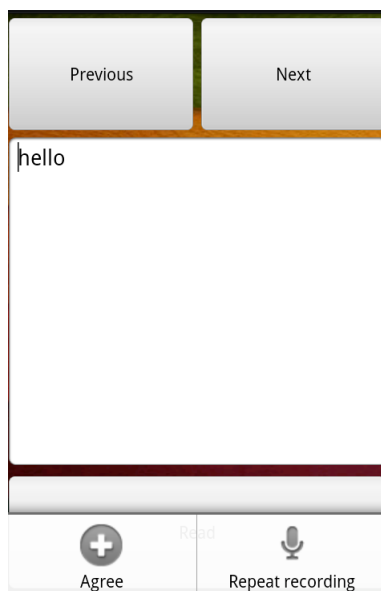
A beszédfelismerő motor használatához szükség van internet kapcsolatra (beépített beszédfelismerő a felvett mintát a központi adatbázisban megtalálhatókkal hasonlítja össze). Abban az esetben, ha éppen ez nincs bekapcsolva, akkor az alkalmazás felkínálja az hálózati beállítások menübe való navigálást (ld. 4-9. ábra), ellenkező esetben pedig megjelenik a beszédfelismerő. A szöveg bemondása után, a felismert szó egy külön nézetben jelenik meg. Ilyenkor általában a szövegfelismerő több lehetőséget is talál, ekkor a felhasználó kiválaszthatja a felkínáltak közül a legmegfelelőbbet, majd beszúrhatja abba a mezőbe, amire a felismerőt meghívta (ld. 4-10. ábra). Olyan eset is előfordulhat, hogy a beszédfelismerő nem talál a felhasználó számára megfelelő egyezést. Ilyenkor lehetősége van a felvétel megismétlésére.



4-8. ábra SMS üzenet írása



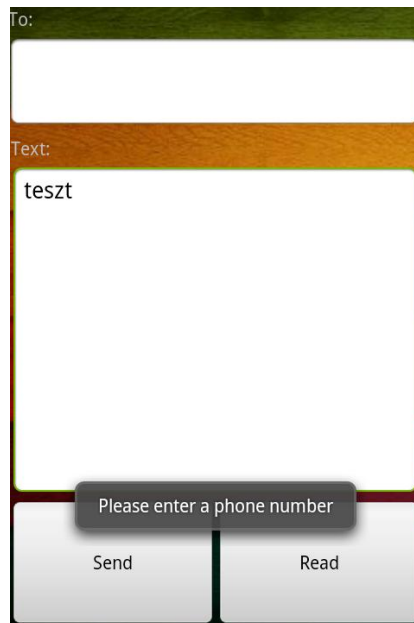
4-9. ábra Ha az alkalmazás észleli, hogy a felhasználó olyan műveletet szeretne végrehajtani, amihez szükséges internetkapcsolat is, azonban ez éppen ki van kapcsolva, akkor felajánlja a hálózati beállítások menübe való elnavigálást.



4-10. ábra A beszédfelismerő eredményének megjelenítése. A felhasználó a menü gomb megnyomása után választhat, hogy elfogadja-e a felkínált szót, mondatot, vagy megismétli a felvételt.

Abban az esetben, ha a felhasználó olyan üzenetet szeretne elküldeni, ahol korábban nem adott meg címzettet, azt egy felbukkanó ablak, és egy hangos értesítés jelzi (ld. 4-11. ábra).

Egy másik eset, ha az üzenet nem került elküldésre, viszont a felhasználó kilép ebből a nézetből, ekkor az üzenet a piszkozatok mappába kerül (ezt szintén hang, ill. egy felugró ablakkal jelzi), ahonnan bármikor újra szerkeszthető, vagy elküldhető lesz a későbbiek folyamán.



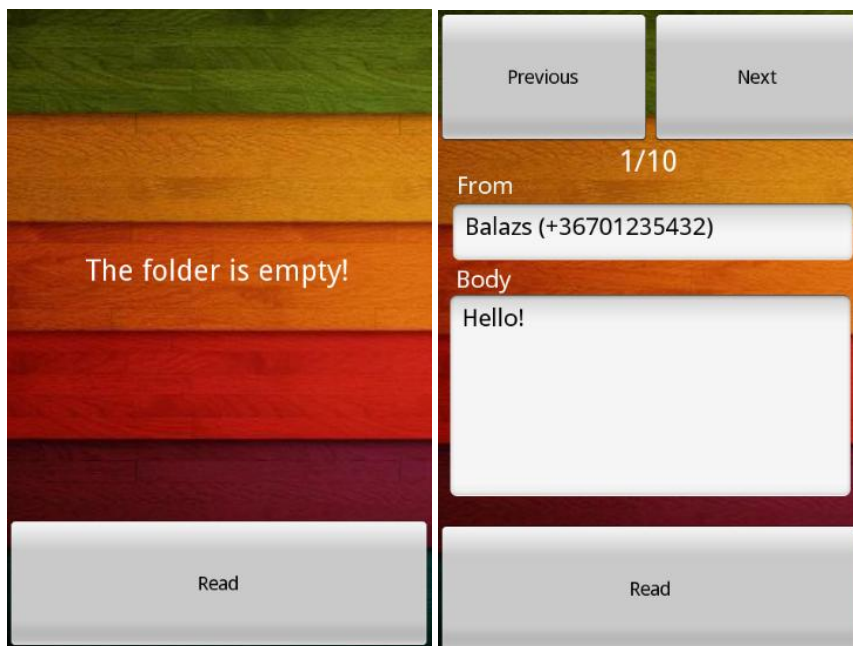
4-11. ábra A felhasználó értesítése, ha nem adott meg legalább egy címzettet

A küldés sikerességéről az alkalmazás hasonló üzenet és hang formájában tájékoztatja a felhasználót. Amint a 4-12. ábrán látható a beérkezett üzeneteknél a felhasználónak lehetősége van kiválasztani, hogy az olvasatlan, vagy olvasott üzeneteket szeretne-e böngészni.



4-12. ábra A bejövő üzenetek menüje. Itt választhat a felhasználó, hogy az olvasott, vagy az olvasatlan üzeneteit szeretné-e böngészni.

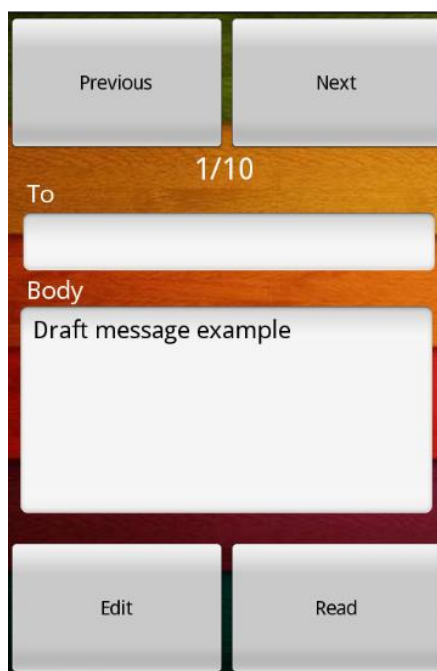
A kívánt mappa kiválasztása után az üzeneteket megjelenítő nézet jelenik meg. Ha a mappa üres, akkor a rendszer ezt külön jelzi a felhasználó számára (ld. 4-13. ábra). Ellenkező esetben az üzenetek egy speciális elrendezésben lesznek megjelenítve. Az alábbi ábrán látható, hogy a korábban tesztelt listás megjelenítést biztosító üzenetkezelő rendszerekkel ellentétben, a vak és látás sérült felhasználók számára kényelmesebb megoldást nyújt. A felhasználó előre és hátrafelé is lépkedhet az üzenetek listájában, majd ezeket fel is olvastathatja az alkalmazással. Ebben a menüpontban ugyancsak lehetőség nyílik üzenetek törlésére, ill. a küldő fél felhívására. Ezeket az opciókat felhasználó a menü gomb megnyomásával hívhatja elő, hasonló módon, ahogy a beszédfelismerő eredményét megjelenítő ablakban.



4-13. ábra Bejövő üzenetek megjelenítése menüpont

Az elküldött üzenetek megjelenítése is hasonló elven működik, így azt külön nem részletezném.

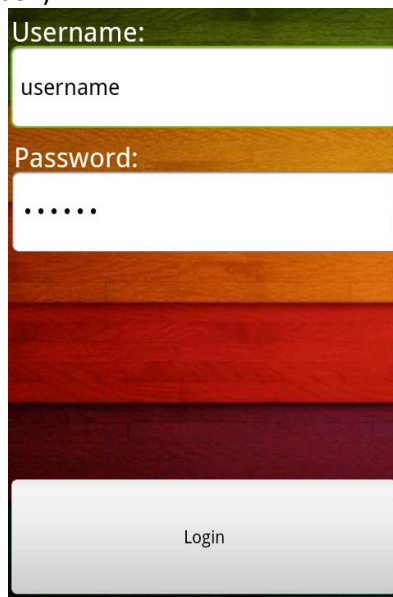
Az alkalmazás a piszkozatok mappa elemeit hasonlóképpen jeleníti meg, mint a beérkezett üzeneteket, azzal a különbséggel, hogy itt lehetőségünk van az üzenet további szerkesztésére is (ld. 4-14. ábra).



4-14. ábra Egy piszkozat mappában lévő üzenet megjelenítése

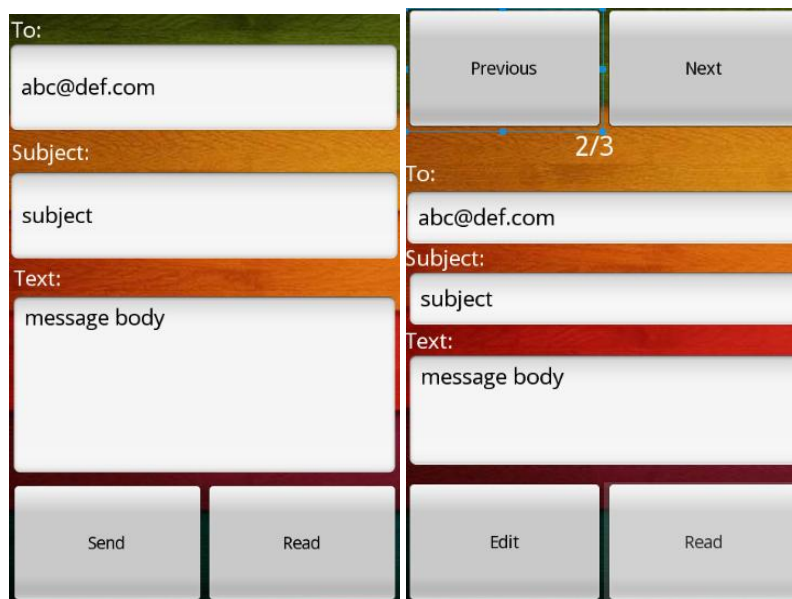
Az e-mail üzenetek menedzselése néhány aprósággal ellentétben hasonlóan történik az SMS üzenetekéhez. A program első indításakor megjelenik egy bejelentkező ablak (ld. 4-15. ábra). Itt léphetünk be a Gmail fiókunkba. A megadott felhasználónév-jelszó párost ezután megjegyzi a rendszer. Erre a bejelentkező nézetre azért volt szükség, mert biztonsági okok miatt sajnos nem tudtam megoldani, hogy a készüléken megtalálható alapértelmezettként megadott Gmail fiók

jelszavát megszerezem, mert maga az operációs rendszer nem teszi lehetővé egy bizonyos alkalmazás által elmentett jelszavak lekérdezését egy másik számára (Ellenkező esetben ez hatalmas biztonsági rést hagyna a rendszerben).



4-15. ábra Bejelentkező ablak Gmail fiókhoz

Az e-mail üzenetek írása, olvasása, ill. piszkozatba mentése hasonló módon történik, az SMS üzenetekéhez, azzal a különbséggel, hogy ebben az esetben az üzeneteknek tárgyat is adhatunk, ill. beérkezett üzenetek esetén lehetőség van továbbküldésre is (ld. 4-16. ábra).



4-16. ábra Üzenetek írása, és piszkozatok megtekintése e-mail üzenet esetén

4.8 Eredmények, fejlesztési tervek

Az általam fejlesztett rendszer jelenleg képes ellátni azokat a funkciókat, amik egy ilyen rendszertől alapvetően elvárhatók. Azonban a projekt részét képezi néhány még meg nem valósított ötlet is.

A rendszer jelenlegi formájában képes a specifikációban leírtak teljesítésére:

- Beérkező üzenetek figyelése.
- Beérkező üzenet jelzése rezgés, és hangüzenet formájában.
- e-mail és SMS üzenetek szétválasztása.
- Üzenetek kezelése mappák szerint (Beérkező, Elküldött, Piszkozatok).
- Üzenetek hangos felolvasása TTS (Text-To-Speech modul segítségével).
- Új üzenet írásának / küldésének lehetősége (ASR motor segítségével).
- Névjegyzék szinkronizáció.
- Csendes üzemmód esetén a riasztás későbbre halasztása.

A specifikációban leírtakon felül a közeljövőben szeretném továbbfejleszteni az üzenetkezelő rendszert, hogy valóban segítséget nyújthasson a vak és látás sérült felhasználók mindennapjaiban.

Az alkalmazás jelenlegi formájában csak angolul tud, ugyanis mindössze az Android-ba beépített TTS-t és ASR-t használja. Ezzel a magyar látássérültek egy része kiesik a rendszer felhasználói köréből. Szeretném, ha a továbbfejlesztések során ez a probléma megoldódna, és a rendszer képes lenne magyarul is szövegek felismerésére és felolvasására.

Elképzelhető az is, hogy a rendszert használó személy angolul és magyarul is egyaránt küld és fogad üzeneteket. Ilyenkor csak a megfelelő szövegfelolvasó és felismerő használható, ezért például célszerű lenne nyelvdetekciót is beépíteni az alkalmazásba.

Egy másik ötlet az akusztikus ikonok használata. Ezek a mindennapi szituációk kifejezésére használt hangjelzések (például nevetés, sírás, sziréna). Ezek használata véleményem szerint kiszínesíthetik, hangulatosabbá tehetik egy felolvasott szöveg tartalmát. [7]

Összegezve úgy gondolom, ha a jelenlegi alkalmazást sikerül kiegészítenem az imént említett javaslatokkal és ötletekkel, akkor egy olyan használható rendszer lehet a végeredmény, ami segíthet ezeknek az embereknek a mindennapi életük színesebbé tételében.

Irodalomjegyzék

[1] Gartner Says Sales of Mobile Devices in Second Quarter of 2011 Grew 16.5 Percent Year-on-Year; Smartphone Sales Grew 74 Percent,
honlap:

<http://www.gartner.com/it/page.jsp?id=1764714> 2011. október 27.

[2] Tóth, B., Németh G., Hidden Markov Model Based Speech Synthesis System in Hungarian, Infocommunications Journal, Volume LXIII, 2008/7, July 2008, pp. 30-34

[3] Péter Mihajlik, Tibor Fegyó, Bottyán Németh, Zoltán Tüske, and Viktor Trón, „Towards Automatic Transcription of Large Spoken Archives in Agglutinating Languages – Hungarian ASR for the MALACH Project” in Proc. of TSD 2007, under the Springer (Heidelberg) series LNCS-LNAI, „Text, Speech and Dialogue”, V. Matousek and P. Mautner (Eds.), pp.: 342-350, Pilsen, Czech Republic, September 2007

[4] iOS Technology Overview: About iOS Development,
honlap:

<http://developer.apple.com/library/ios/#DOCUMENTATION/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSOverview/iPhoneOSOverview.html>

[5] Németh Géza, Olasz Gábor: *A magyar beszéd*, Akadémiai kiadó 2010

[6] Gartner Says Android to Become No. 2 Worldwide Mobile Operating System in 2010 and Challenge Symbian for No. 1 Position by 2014,
honlap:

<http://www.gartner.com/it/page.jsp?id=1434613> 2011. október 28

[7] A magyar beszéd,

honlap:

<http://magyarbeszed.tmit.bme.hu/index.php?p=spemoticon>