

Étel-érzékenyek termék választásának segítése okostelefon alkalmazásával

Tudományos Diákköri Dolgozat

Eger

2014

Tartalomjegyzék

I. Bevezetés	3. oldal
II. Szakirodalmi áttekintés	5. oldal
III. Piaci igények felmérése	25. oldal
IV. A szoftver	36. oldal
V. Szakirodalmi jegyzék	49. oldal

I. Bevezetés

*„Ami az egyik embernek táplálék,
az a másinak keserű méreg.”*

- Lucretius

Amennyiben az idézetet tekintjük, akár arra is gondolhatnánk, hogy azt a napjainkban fogalmazták meg.

A római költő és filozófus, teljes nevén Titus Lucretius Carus (Kr. e. 99 – 54) azonban Krisztus előtti évszázadban élt és dolgozott. Ebből kis kitűnik, hogy az allergia, az ételérzékenység nem a mi korunkba született.

Egy azonban biztos: az emberiség történetében még soha nem volt olyan elterjedt az ételérzékenység, mint napjainkban. Mondhatjuk, hogy korunk népbetegségévé vált az allergia, arra azonban nincs pontos válaszunk, hogy ezt mi válthatta ki.

E témakörön gondolkodva jutott eszünkbe egy olyan okostelefon alkalmazás ötlete, amely megkönnyítheti, segítheti az élet-allergiások számára a vásárlást, a megfelelő termékek választását. Az ötlet megszületése után el is kezdtünk dolgozni a szoftver megvalósításán, melynek eredménye, hogy ma már (2014 első fele) egy kész, működő alkalmazásunk van. Célunk, hogy az alkalmazás elterjedjen, ingyenes szoftverről lévén szó, azt sokan töltsék le és sokan is használják.

Látható tehát, hogy már a Krisztus előtti I. században felmerült az étel-allergia problémaköre (Lucretius a *De rerum natura* című művében ír róla), napjainkban már népbetegséggé vált. Mi azért kezdtünk el dolgozni e szoftveren, hogy valamilyen szinten segítsük és megkönnyítsük az étel-érzékenyek mindennapjait, őket alternatívákkal, személyes tanácsadással is segítsük.

A tudományos munkánk az alábbi nagyobb egységből épül fel: bevezetés, szakirodalmi áttekintés, a piaci igények felmérése, a szoftver, az összefoglalás, szakirodalmi jegyzék, valamint ábrajegyzék.

A **bevezetésben** kerül sor a további fejezetek rövid bemutatására, itt írunk a témánk (és a szoftverünk) aktualitásáról és gyakorlati hasznáról is.

A **szakirodalmi áttekintésben** témával összefüggő fogalmak és a szoftver kerül feldolgozásra és bemutatásra. A szakirodalmi áttekintést három részre osztottuk fel:

- allergia, ételérzékenység
- az Androidról, általánosságban
- az Android verziói
- fejlesztés Androidra

A harmadik pontban a **piaci körülményeket tekintjük át**, tesszük ezt a saját kérdőíves kutatásunk bemutatására, illetve a kérdéseinkre kapott válaszok kerülnek elemzése és bemutatásra. Szoftverünk kifejezetten innovatív, azonban a miénkhez hasonló alkalmazások már léteznek. E fejezetben fogunk három, a miénkhez hasonló alkalmazást elemezni és leírni.

A negyedik pontban fogjuk részletesen **bemutatni szoftverünket**, annak felépítését, működését és alappilléreit. E fejezetben fogunk írni a szoftverünk jövőjéről, illetve a jövőbeli ötleteinkről, konkrét terveinkről is.

Az utolsó, szám szerint ötödik pontban pedig a **szakirodalmi jegyzéket** fogjuk leírni.

II. Szakirodalmi áttekintés

Ebben a fejezetben meg szeretnénk vizsgálni, hogy azokhoz a kérdéskörökhöz, melyekkel mi is találkoztunk Tudományos Diákköri munkánk során, – és esetleg nézeteltéréseket, problémákat okozhatnak – hogyan viszonyulunk, hogyan értelmezzük mi azokat. E fogalommagyarázat fontos szerepet tölt be a dolgozatunkban, hiszen egyrészt az IT-szektor folyamatosan változik, fejlődik, így nem elhanyagolható az alapfogalmak értelmezése a tisztánlátáshoz. Másrészt fontosnak tartjuk, hogy azok a személyek is értsék, megértsék és értelmezni tudják Tudományos Diákköri munkánkat, akik nincsenek benne a számítástechnikában, hanem átlagos tudással rendelkeznek a technológiai szektorról.

1) Allergia, ételérzékenység

A Tudományos Diákköri munkánk címét olvasva először ötlik szemünkbe az étel-érzékeny kifejezés. Az allergiáról, ételérzékenységről nem kívánunk hosszan szólni, hanem a jelenleg számunkra legfontosabb és leglényegesebb definíciókat hozzuk.

Az emberi szervezetben végbemenő folyamatok és mechanizmusok kifejezetten összetettek, elmondható azonban, hogy mindennek meg van a maga feladata. Az immunrendszer feladata felismerni a saját és az idegen molekulákat, melyek belépnek az emberi szervezetbe.

Elmondható tehát, hogy az immunrendszer által végzett nullától huszonnégy óráig tartó munka a monitorozás, valamint az esetleges betolakodók észlelésekor az azonnali ellenreakció, vagyis a betolakodók elpusztítása és eltávolítása. Ez az összetett folyamat elengedhetetlen az emberi szervezet működéséhez. A monitorozás mellett az immunrendszer képes adatokat is tárolni: ha egy olyan károkozó sejttel találkozik, amit kár korábban eltávolított, az ellen a szervezet védetségét alakít ki. Ezt immunitásnak nevezzük.

Az egészséges szervezet immunrendszere megfelelően végzi munkáját és csak akkor „riaszt”, ha az az eset indokolt. Nem meglepő azonban, hogy ekkora munka és „adathalmaz” mellett az immunrendszer hibázik: akkor is ellenreakciót vált ki, ha az nem is indokolt, ha nem is áll fenn veszélyhelyzet. Ezen esetben a szervezet olyan ellenreakciót vált ki, mintha a nem káros sejtek is ténylegesen káros sejtek lennének. A védelmi rendszer hibázik, és előnyös funkcióik helyett károsító mechanizmusokat léptet életbe. Ez esetben lehet beszélni allergiáról.

Ahogy Tóth Gábor is megfogalmazza: „*Az allergiás reakció tehát nem más, mint az emberi*

szervezetet károsító immunreakciók gyűjtőneve. Allergiában az allergén hatására az immunrendszer kórosan fokozott mértékben aktiválódik, és valójában ez okozza a szervezet károsodását.”

Vagyis megállapítható, hogy e hiba eredményeképp tévesen ad "riasztást," a védelmi rendszer aktiválódik. Tóth Gábor megfogalmazta ennek lehetséges következményeit is:

„Amiképpen az ilyenfajta gyakori tévedések és a velük járó állandó kiadások fölemészthetik egy cég tartalékait, hasonlóképpen az emberi szervezet is leépül, elhasználódik.”

Az allergia jelensége messze nem új, hiszen már Kr. e. V. században, Hippokratész is írt élelmiszer-allergiákról, és azok kísérő tüneteiről. Lucretius (Kr. e. 95-55) így fogalmazta meg az allergia definícióját: *„Ami az egyik embernek táplálék, az a másiknak keserű méreg.”* [1]
[24-25.oldal]

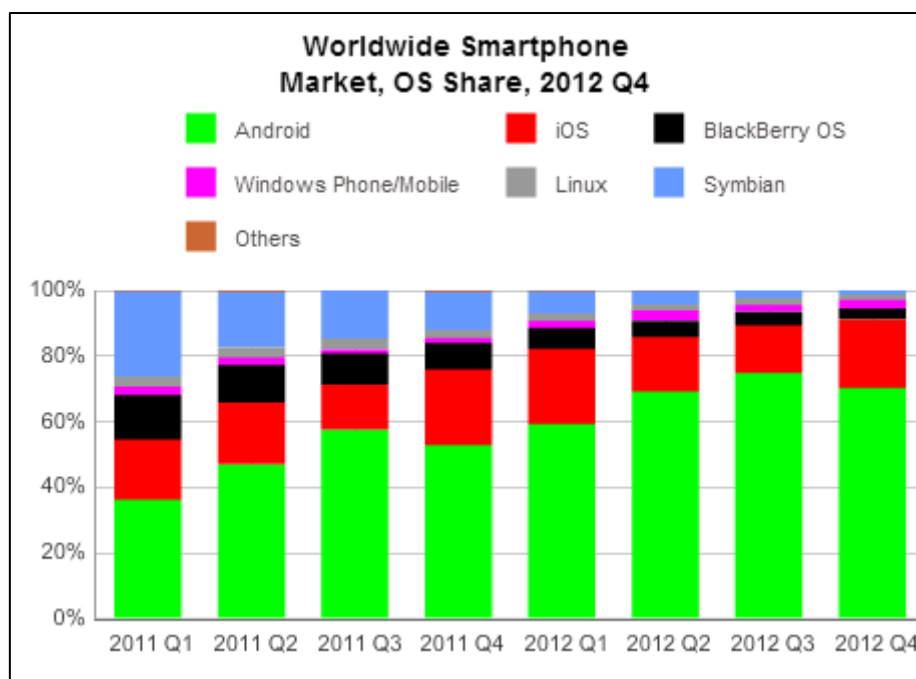
2) Android – Avagy miért pont Android?

Valóban fontos kérdés, hogy miért döntöttünk az Android mellett. Nos, az igazság az, hogy jelenleg sokkal több lehetőség rejlik az Androidban, mint riválisaiban, ami a lent is látható piaci részesedésekből, arányokból nagyon jól kitűnik. A következő alfejezetben össze fogjuk hasonlítani a három rendszert:

2.1. Összehasonlítás

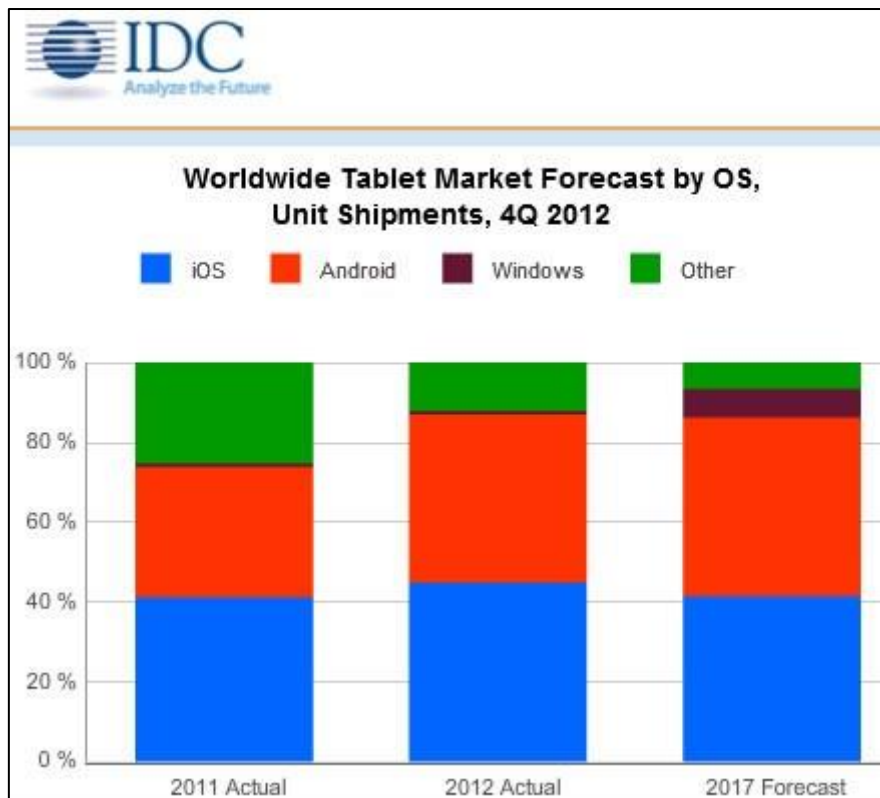
Ebben a fejezetben diagramok segítségével támasztjuk alá azon érvünket, hogy az Android jó és piacképes, nagy fejlődési potenciállal bír.

A lent látható ábra alakján megállapítható, hogy az eladások terén az Android messze lekörözi a riválisait.



1. ábra: Az Android piaci részesedésének alakulása [2]

Jól látható az Android fölénye a riválisokkal szemben. Ha valaki úgy gondolná, hogy táblagépek területén más a helyzet, azt ki kell ábrándítanunk. Az Android már itt is fontos pozíciót tölt be.



2. ábra: Táblagép értékesítések aránya [3]

Jól megfigyelhető, hogy az Android itt azért nem annyira egyértelműen uralja a piacot, de az előrejelzések alapján itt is növelni fogja piaci részesedésének arányát.

De vajon mi lehet az Android titka?

A titok elsősorban az, hogy nyílt forráskódú a rendszer, ezért szeretik a fejlesztők, mondhatjuk, hogy könnyebben boldogulnak vele. A vásárlók szempontjából pedig az lehet a siker titka, hogy sokkal kedvezőbb áron juthatnak egy Android rendszerrel működő készülékhez, mint riválisaihoz. Az olcsóbb készülék pedig nem jelent gyengébb minőséget, sőt!

2.1.1. iOS

Az Apple Inc. rendszere röviden összefoglalva a következő elvek szerint működik:

Az iOS hátránya vagy előnye – ezt mindenki döntse el maga, mi inkább hátránynak tekintjük – hogy egy teljesen zárt rendszer. Ez azt jelenti a valóságban, hogy mindenre megvannak azok a bizonyos előírások. Ha bármit szeretnénk letölteni a készülékre, csak a hivatalos

boltból tehetjük meg. Továbbá ha bármilyen adatot, zenét vagy akár filmet szeretnénk a készülékre másolni, szükségünk lesz a számítógépes iTunes-ra.



3. ábra: Apple iPhone 5s [4]

De akkor mégis miért lehet ennyire sikeres? Az ok az, hogy a felhasználónak könnyű dolga van a rendszerrel kapcsolatban. Nehéz benne eltévedni, minden kézenfekvő. Ezáltal gördülékeny a rendszer, továbbá azáltal, hogy csak hivatalos boltból tölthetjük, biztosak lehetünk benne, hogy minőségi szoftvert kapunk. Mi azonban most programozni szeretnénk, nem pedig használni a készüléket, éppen ezért mindenképpen negatívumként értékelhető a rendszer zártsága. Továbbá azt is fontos kiemelni, hogy az árképzése is jelentősen magasabb egy belépő szintű Androidos készülékkel szemben.

2.1.2 Windows Phone

A Microsoft Co. által fejlesztett készülék, hasonlóan az iOS-hez, ez is zárt rendszerrel dolgozik. A zárt rendszer nagy előnye az Androidhoz képest, hogy stabilabban és hibamentesebben működik.

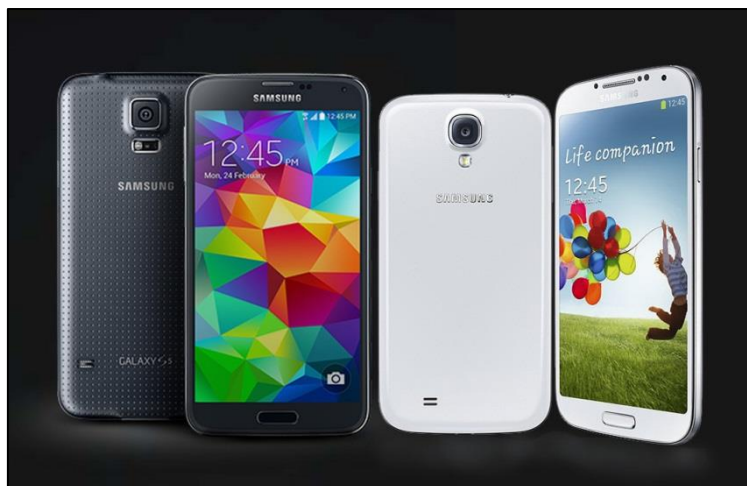


4. ábra: Nokia Lumia készülékek [5]

Ezek alapján Tudományos Diákköri munkánk fejlesztését akár Windows Phone-on is végezhattük volna. Akkor miért nem Windows Phone? A válasz egyszerű: jelenleg még nagyon kicsi a piaci részesedése és a közeljövőben jelentős növekedés nem várható az ellenfelekkel szemben. A mi meglátásunk szerint itt is megemlíthető az árképzés, mely itt is magasabb az Androidénál.

2.1.3. Android

Az Android – a másik két példával szemben – egy teljesen nyílt forráskódú rendszer, melyre bárholnan telepíthetünk, szinte bármit. Ahogy minden éremnek két oldala van, úgy a nyílt forráskódnak is megvannak a sajátos hátrányai. Elmondható, hogy az alkalmazások száma nem arányos azok minőségével, vagyis számtalan programból válogathatunk, ám ezek minősége sokszor megkérdőjelezhető. Akkor mégis miért ennyire megnyerő? A válasz szerintünk a nyílt forráskódúságban rejlik. A programoz számára pedig nem is kell más, csak egy olyan rendszer, amiben szabadon mozoghat. Ezért is készül olyan sok alkalmazás, ezáltal szinte bármilyen feladatra, problémára találunk megoldást nyújtó programot, programokat.



5. ábra: Samsung Galaxy S5 [6]

Ezt adja az Android, illetve a szintén nem elvetendő alacsony ár (a versenytársakhoz viszonyítva). Természetesen itt is megtalálhatóak a drága készülékek, melyek ugyancsak Androidot használnak, de ha valaki olyan készüléket szeretne, amin Android fut, mondhatni relatív olcsón hozzájuthat ilyen készülékhez.

Tehát összefoglalva, fejlesztői szempontból az Android a legjobb választás, illetve a piaci részesedéseket és eladásokat tekintve is pozitív jövő áll e operációs rendszer előtt.

2.2. Az Android verziói

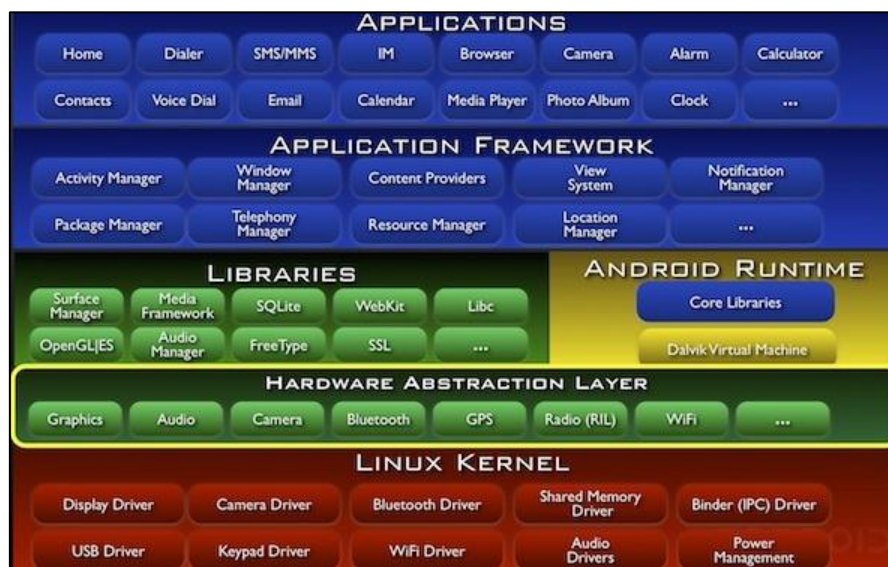
Amennyiben részletesebben szeretne a verziókról olvasni, az alábbi címen megteheti: [7]

Az egész folyamat azzal indult, hogy 2005-ben a Google felvásárolta az Android Incorporated nevű kaliforniai vállalatot. Ezután egy saját operációs rendszer fejlesztésébe kezdett, mely rendszer 2008 végére lépett piacra, a HTC G1 nevű készülékkel, melyet a T-Mobile forgalmazott.



6. ábra: HTC-G1 készülék [8]

A Google célja az volt, hogy egy olyan rendszert hozzanak létre, mely nyílt forráskódú, rugalmas és könnyen alakítható, melyhez a fejlesztők java nyelvű kódot tudjanak írni. A fejlesztés megkönnyítése érdekében minden alkalmazás, kivétel nélkül, hozzáfér a készülék összes erőforrásához, mivel teljesen egyenrangúak. Ha ez még nem lenne elég, a Google által fejlesztett java könyvtárak még könnyebbé teszik a fejlesztők munkáját. A rendszer monolitikus Linux kernelen (magon) alapul.



7. ábra: Az Android rendszer felépítése [9]

A rendszer a vörös mezőben látható Linux kernelre épül fel, ez tartalmazza a hardverekhez szükséges Drivereket. Ezeket a Drivereket azok a cégek készítik el, melyek saját készülékükön szeretnék használni az Androidot. Azért ők, mert így a legkisebb a hiba lehetősége, ugyanis a saját hardverükhöz csinálják a drivereket. Ebben a kernelben zajlik a memóriakezelés, a folyamatütemezés és a teljesítménykezelés.

A kék színnel jelölt részben Java forrást találunk, melyet a virtuális gép futtat. Összefoglalva az Android lényege tehát, hogy könnyen alakítható, rugalmas, kis hardverigényű rendszer, mely a felhasználókat látványos és letisztult felületével, és könnyű bővíthetőségével csábítja magához.

Mint mindenhol, itt is újabb és újabb verziók jelentek meg, annak érdekében, hogy az Android egyre jobb legyen és versenyképes maradjon. Nézzük, milyen változatai jelentek meg az Androidnak. Az 1.5 (Cupcake) előtti verziókról nem esik szó, hiszen azok igazából még csak arra szolgáltak, hogy a fejlesztőket segítsék, valamint, hogy az Android iránt felkeltség a cégek érdeklődését.

2.3. Verziók

2.3.1. Android 1.5 (Cupcake)

1. Jelentős újírtásnak mondható, hogy már készíthetünk videofelvételeket, melyeket fel is tölthetünk akár azonnal a YouTube-ra.
2. Új szoftveres billentyűzet, szókiegészítő/javító funkcióval.
3. Továbbá jelentős újírtásnak mondható a Bluetooth A2DP támogatás, azaz használhatóvá váltak a sztereo Bluetooth headset-ek.
4. Egyéb, kevésbé jelentős újírtások is kerültek a rendszerbe, ilyen például az új widgetek és mappák, illetve az új animációk a képernyők között.
5. Böngésző oldalról pedig elérhetővé vált a copy/paste funkció.

2.3.2. Android 1.6 (Donut)

Az 1.6-os verzió ráncfelvarrás volt az 1.5-höz, az alábbi újírtásokkal:

1. Továbbfejlesztett Android Market, mely egyszerűbb használatot és több funkciót hozott.
2. Fontos újírtás az integrált fényképező, a kamera és galéria interfész, a kamerát kezelő szoftver is jelentősen gyorsult, ezen felül pedig a galériában egyszerre többek között gyorsabb lett a keresőmotor. Technológiai újírtások is megjelentek, CD-MA/EVDO, 802.1x VPN gesztusok, illetve QVGA támogatás.

2.3.3. Android 2.0 / 2.1 (Eclair)

A 2.0-as verzió 2009 októberében jelent meg, ez már egy komolyabb változtatásokat hozó rendszer volt:

1. Ebben a két verzióban nagyon fontos újítások érkeztek. Ilyen például a hardveres optimalizáció, ebből kifolyólag gyorsabb működésre képes a rendszer, ezen felül az Android már támogatja az eltérő kijelzőfelbontást és méretet.
2. Webes téren is komoly változtatásokra került sor. HTML5 támogatása és a böntésző felület megváltoztatása, valamint a Google Maps is fejlődött, gyárilag a 3.1.2-es verzió került a rendszerbe.
3. Kényelmesebb használat érdekében már támogatja a rendszer a Multi-touch-ot, támogatja a kamerához a vakut és LED-et, valamint bekerült a kényelmesebb virtuális billentyűzet is.

2.3.4. Android 2.2 (Froyo)

A 2.2-es verzió sok-sok újítást hozott, rendszeroptimalizálást és fejlesztést, ezáltal még vonzóbbá és szerethetővé teszi az Androidot. Részletesebben, az alábbi újításokkal érkezett:

1. Optimalizálták a rendszer működését, ezáltal gyorsabb lett, JIT integrálás, eredménye a gyorsabb Java működés. Integrálták még a Chrome V8 Javascript motort a böngészőhöz, valamint az Adobe Flash 10.1-et.
2. Hálózati téren is történtek változások, mint például az USB-n megosztható internet, illetve a WiFi hotspot funkció, valamint már lekapcsolhatóvá tették a mobilinternetet.
3. Egyéb olyan változtatásokat is végeztek, melyek kisebb volumenűek voltak, de a felhasználót segítették. Ilyen például, hogy a rendszer támogatja a névjegy és hanghívás megosztását Bluetooth-on, támogatja a GIF-eket a böngészőben és nem utolsósorban feltöltési lehetőséget a böngészőben.

2.3.5. Android 2.3 (Gingerbread)

A következő verzió az Android 2.3 (Gingerbread), mely 2010. december 6-án került bemutatásra. Ebben a verzióban inkább arra törekedtek, hogy optimalizálják a rendszer működését, azaz gyorsabb, stabilabb és biztonságosabb legyen annak működése. Ezért ezt most nem kívánjuk elemezni, mert itt ténylegesen csak optimalizálásról volt szó.

2.3.6. Android 3.0 (Honeycomb)

2011. január 26-án bemutatta az Android 3.0 Honeycomb változatot. Ez viszont valóban forradalmi változást hozott, hiszen nem a korábbi verziók csiszolásával készült, hanem majdnem a nulláról újraírták az egészet. A fejlődés persze itt nem áll meg, viszont a Google mostantól két, párhuzamos szálon futtatja tovább az Androidot. A 2.x verziók lesznek az

okostelefonokra szánt változatok, míg a 3.0 és ennek újabb változatai a tabletekre készülnek. Most nézzük meg mit is változtattak:

1. Nagyon jelentős fejlesztések és újítások kerültek a rendszerbe, ilyen az új 3 dimenziós főképernyő, a "holografikus" kezelőfelület, hardveres 2D gyorsítás, vagy a Renderscript 3D grafikus motor is.
2. További pozitívum hogy az eddigi Android programokkal kompatibilis, azaz a régebbi Androidra készült programok is gond nélkül futtathatóak.
3. Ezen felül a rendszer támogatja a többmagos processzort is.
4. Felhasználói szempontból lényeges a fejlettebb szövegkijelölés, a közvetlen kapcsolódás digitális kamerához, az USB és Bluetooth billentyűzet kezelése és a gyári alkalmazások frissítése is.

2.3.7. Android 3.1 (Honeycomb)

A 3.1-es Android 2011. május 10-én jelent meg. Ezen verziónál igazából a felhasználók segítése volt a cél. Azaz olyan dolgok kerültek bele melyek segítik a rendszer használatát. Éppen ezért a háttérben nem is nagyon változott semmi az előző változathoz képest. Ilyen segítség az átméretezhető minialkalmazások, USB-s kiegészítők támogatása, valamint hogy a képernyő lezárásával nem szakad meg a WiFi kapcsolat.

2.3.8. Android 3.2 (Honeycomb)

A 3.2-es verzió 2011. július 15-én debütált a Huawei MediaPad-en. Bár az újdonságok listája egy kissé szerényebb elődjénél, néhány fontos képesség azért belekerült a rendszerbe:

1. Bővült a támogatott hardverek listája
2. A külső alkalmazások is elérhetik a memóriakártya tartalmát
3. Kisebb felbontású kijelzőre fejlesztett alkalmazások megjelenítése teljes képernyős módban, pixeltöbbszörözéssel
4. 720p felbontású televíziók támogatása

2.3.9. Android 4.0 (Ice Cream Sandwich)

2011. október 19-én jelent meg a Samsunggal közös Google Galaxy Nexus telefonon. A 4.0-as android 2011. október 19-én jelent meg. Nagyon sok újítás került bele ebbe a verzióba, nagyon sok a mindennapi használat megkönnyítésének érdekében. Ilyenek például:

1. Beépített képszerkesztő, a minialkalmazások átméretezhetővé váltak, az értesítések már egyenként is törölhetőek, illetve a weboldalak menthetőek, azaz internet kapcsolat nélkül is elérhető a mentett tartalom.

2. Jelentős újítás még az 1080p felbontású videófelvétel, illetve zoomolási lehetőség felvétel közben. Sok egyéb újítás is helyet kapott ebben a verzióban, de ezek kevésbé mérvadók.

2.3.10. Android 4.2 (Jelly Bean)

A 4.1-es Android és a 4.2-es között voltak még a 4.1-esnek egyéb változatai, de ezekre nem térnék ki részletesebben. Helyette inkább a jelenleg legújabb verziót ismertetném. A 4.2-es verzióról 2012. október 29-én mutatták be. Bár jó néhány újdonságot kaptunk, továbbra is a Jelly Bean névvel illetik a rendszerverziót.

1. Fontos újítás hogy a vakok és gyengénlátók segítésére nagyító és szövegfelolvasó rendszer került bevezetésre.
2. Szórakoztatási szempontból fontos újítás a gömbpanoráma-képkészítés, valamint átrajzolták a kamera UI-t.
3. Nem utolsó sorban pedig több felhasználói fiók is létrehozható a táblagépen.
4. Sok-sok egyéb fejlesztés is belekerült a rendszerbe, de ezekre most nem térnék ki.

2.3.11. Android 4.2.2

A jelenlegi legfrissebb verziót 2013. február 11-én mutatták be, teljesítményoptimalizálás és finomítás, ez a két fő irányvonal. Ez azt jelenti, hogy inkább egy ráncfelvarrásról van szó, azaz apró finomításokat tartalmaz a rendszer.

2.3.12. Android 4.3 (Jelly Bean)

2013. július 24.-én került bevezetésre, rengeteg újdonságot hozva magával:

Az egyik legfontosabb szoftveres változás az OpenGL ES 3.0 támogatás, gyorsabb tudunk felhasználót váltani, továbbfejlesztették a DayDream funkciót, bevezetésre került a Bluetooth Low-Energy támogatás is. Ezen felül lehetőségünk nyílik helymeghatározást végezni Wi-Fi segítségével.

Továbbá az egyik legjelentősebb újítás a Ultra HD (4K) támogatása, mellyel együtt hüvelykenként 640 képpont jelenítődik meg.

2.3.13. Android 4.4 (KitKat)

E verzió 2013. október 31-én jelent meg. Ebben a verzióban is jelentős változások történtek:

1. Sikerült a teljesítményt optimalizálni, ezáltal alacsonyabbak lettek a rendszerkövetelmények. Továbbá beépítésre került egy nyomtatási keretrendszer is, valamint egy NFC olvasó.
2. Továbbá olyan nem kevésbé jelentős kiegészítők is bekerültek, mint a DSP audio csatorna, a beépített kijelzőfelvétel, a kiterjesztett elérhetőségű API-k, valamint nem utolsónak a Bluetooth MAP támogatás.

2.3.14-15. Android 4.4.1 és 4.4.2

E verziók 2013. december 5 és december 9.-én jelentek meg, azonban jelentős változtatásokat nem hoztak, inkább csak kisebb hibajavításokat, apróbb módosításokat tartalmaztak.

2.3.16. Android 4.5 (Lollipop részletek)

A Google az alábbi fejlesztéseket hajtotta végre:

1. Ezen új verzió nem az újításra, hanem a 4.4.2-es verzióban fellépő nagyobb hibák kijavítására fektette a hangsúlyt, ennek eredményeképp a 4.5-ben javították a kamera fókuszálási képességeit.
2. Design-ját tekintve kis változás lesz csak, ugyanis a gyári alkalmazások ikonjain változtatott a Google.

A tudományos munkánk leadásakor hozta nyilvánosságra a Google, hogy sok más fejlesztés mellett (okosóra, okosautó, Android TV) fog kijönni az új Android verzió is. Erről sajnos a fentieknél is bővebb információval – a tudományos munka írásakor – nem rendelkezünk.

2.4. Fejlesztés Androidra

Ha rászántuk magunkat, hogy Androidra szeretnénk fejleszteni, jó pár programra szükségünk lesz a fejlesztés megkezdése előtt. Vegyük sorba hogy mik is ezek a programok:

1. Először is ellenőrizzük, hogy a számítógép megfelel-e a minimum követelményeknek.
2. Eclipse fejlesztői környezet telepítése.
3. SDK telepítése.
4. Eclipse ADT plugin telepítése.
5. SDK-n belül a használni kívánt Android verzió telepítése.
6. AVD emulátor létrehozása.

A következő oldalakon tekintsük meg, hogy melyik program hogyan működik és mire való.

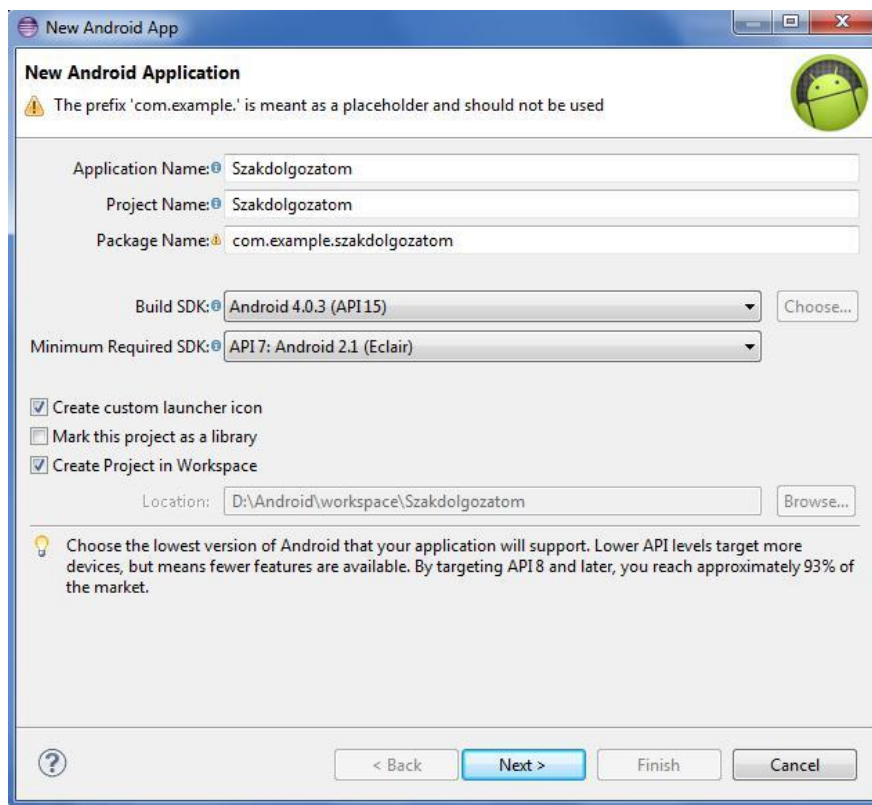
2.4.1. Szükséges szoftverek

2.4.1.1. Eclipse

<http://www.eclipse.org/downloads/>

Az Eclipse nem csak Android fejlesztésre alkalmas, hiszen fejleszthetünk benne: Java, JavaEE, C++ és még más nyelveken is. Ha feltelepítettük az SDK-t és az Eclipse plugint, akkor a projektünket egy emulátorban le tudjuk tesztelni.

Új projekt létrehozása a file/new/Android Application Project-re kattintva tehető meg.



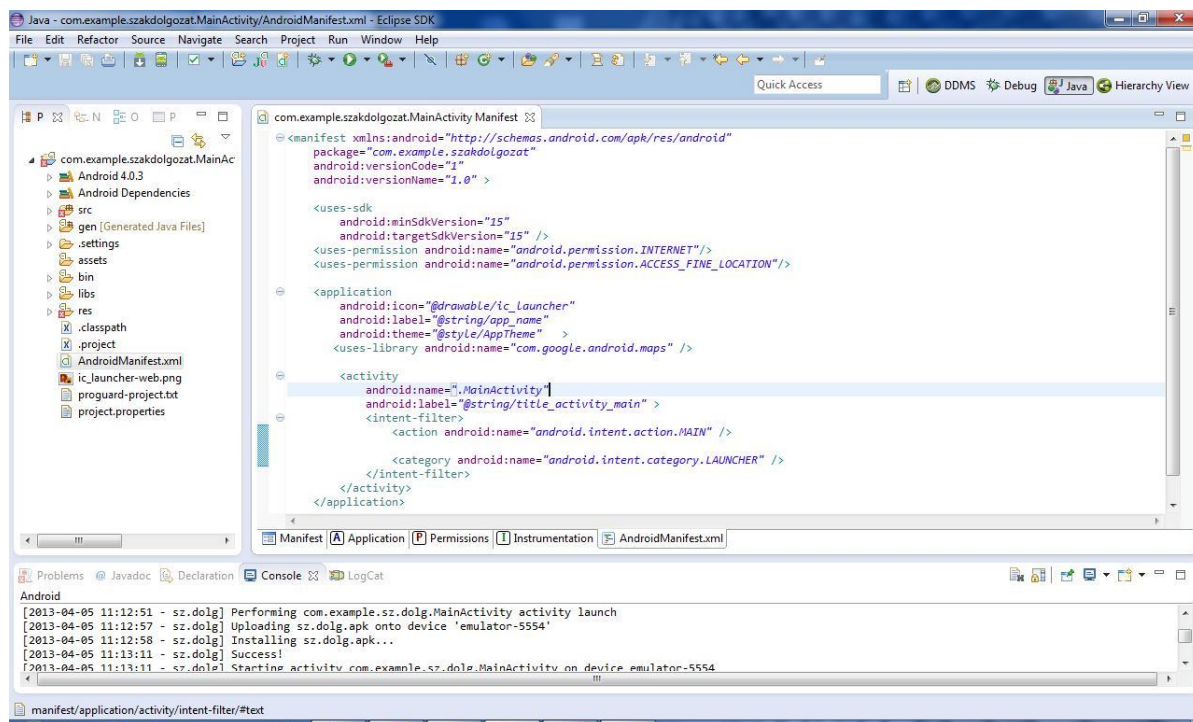
8. ábra: Új projekt létrehozása Eclipse-ben

Elég az applikáció nevét megadnunk, a projekt nevét és a csomag nevét automatikusan kitölti nekünk az Eclipse. Ezután ki kell választanunk a Build SDK-t azaz melyik verzióra szeretnénk fejleszteni, alatta pedig megadhatjuk, hogy minimum milyen android szükséges hogy a program tökéletesen működjön. Ez lehet alacsonyabb, mint a fejlesztésre kiválasztott rendszer, de vigyázni kell, mert nem biztos, hogy minden funkció működni fog egy régebbi rendszeren. Ezért ajánlatos a két verziónak egyeznie.

A következő képernyőn, ami fogad minket megadhatjuk, hogy hogy nézzen ki a program ikonja, majd azt hogy milyen elrendezésű legyen az Activity. Ezen felül még be lehet állítani az Activity nevét és egyéb dolgokat. Azért nem mutatom be részletesebben ezeket, mert tökéletesen működik, ha alapértelmezetten hagyjuk őket. Miután rákattintottunk a Finish-re, legenerálja a projektet, mely alapértelmezetten tartalmaz egy Hello Word-öt.

Eclipse felépítése

Hiába készítettünk egy projektet, ha nem tudjuk, hogy mi hol található és mire való. Éppen ezért bemutatom a fontosabb dolgokat, hogy mit hol találunk.



9. ábra: Az Eclipse felépítése

Bal oldalt található a Project Tree azaz a projektünk felépítése, innen érhetőek el a különböző részei az aktuális projektünknek.

A fenti menüsor az Eclipse beállításában segít, a képernyő közepén, azaz a túlnyomó részén a szerkesztőnézet található, az éppen aktuális állományt megjelenítve.

Jobb oldalon kicsit felülválthatunk a nézetek között, ha éppen teszteljük a programot, akkor hasznos lehet az itt található Debug, illetve a DDMS.

A DDMS lényege hogy az emulátor vagy a mobilkészülék beállításait láthatjuk, illetve ha szükséges, akkor módosíthatunk adatokat.

A képernyő alján pedig a hibákra figyelmeztethet minket az Eclipse, illetve megtaláljuk még itt a LogCat nevű ablakot is, mely futás közben folyamatosan tájékoztat az eseményekről.

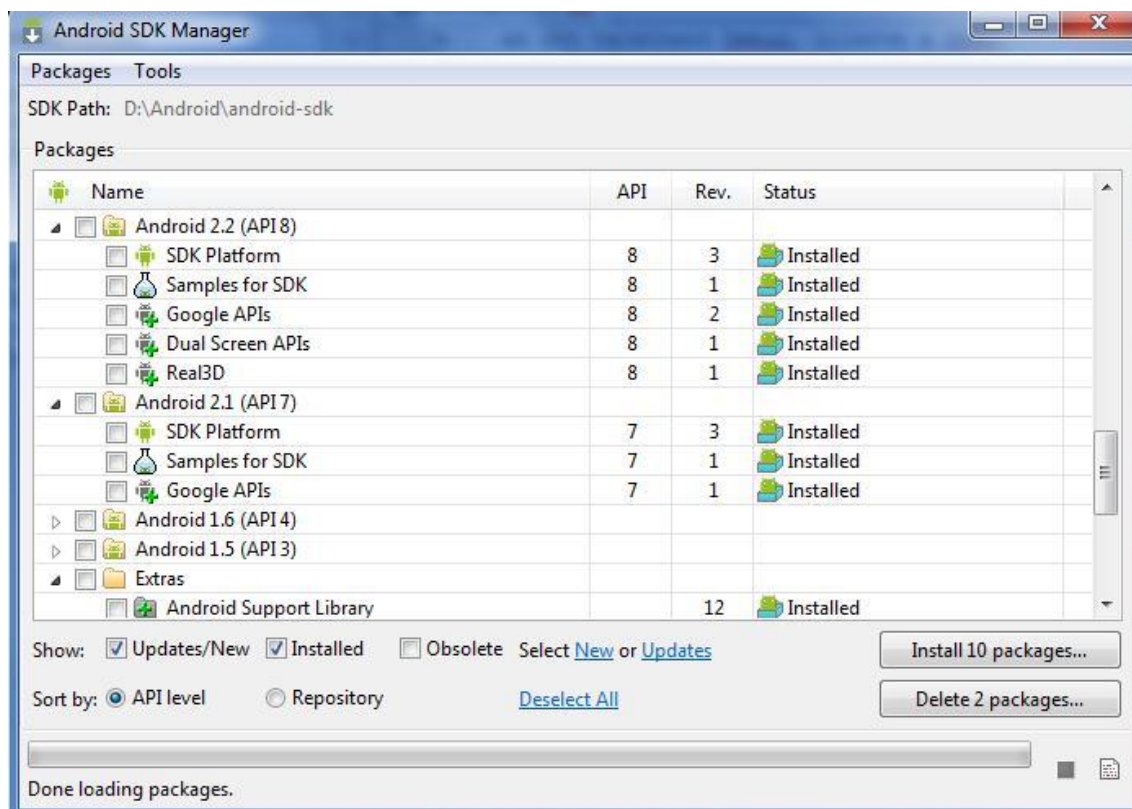
2.4.1.2. ADT-plugin

Az ADT-plugin szerepe, hogy az Eclipse-ben tudjunk Androidra fejleszteni. Telepítése után az Eclipse-ben, új projekt létrehozásánál, megjelenik az Android Application Project. Erre kattintva már elkezdhetjük a fejlesztést Androidra.

2.4.1.3. SDK-manager

<http://developer.android.com/sdk/index.html>

Mi is az az SDK-manager? Az legfőbb feladata az, hogy frissíthetjük a meglévő SDK komponenseinket, újakat tölthetünk le, valamint ami a legfontosabb, hogy az Android melyik verzióját szeretnénk letölteni. Ez azért fontos mert az AVD-managerben csak olyan emulátort hozhatunk létre, melyet letöltöttük itt.



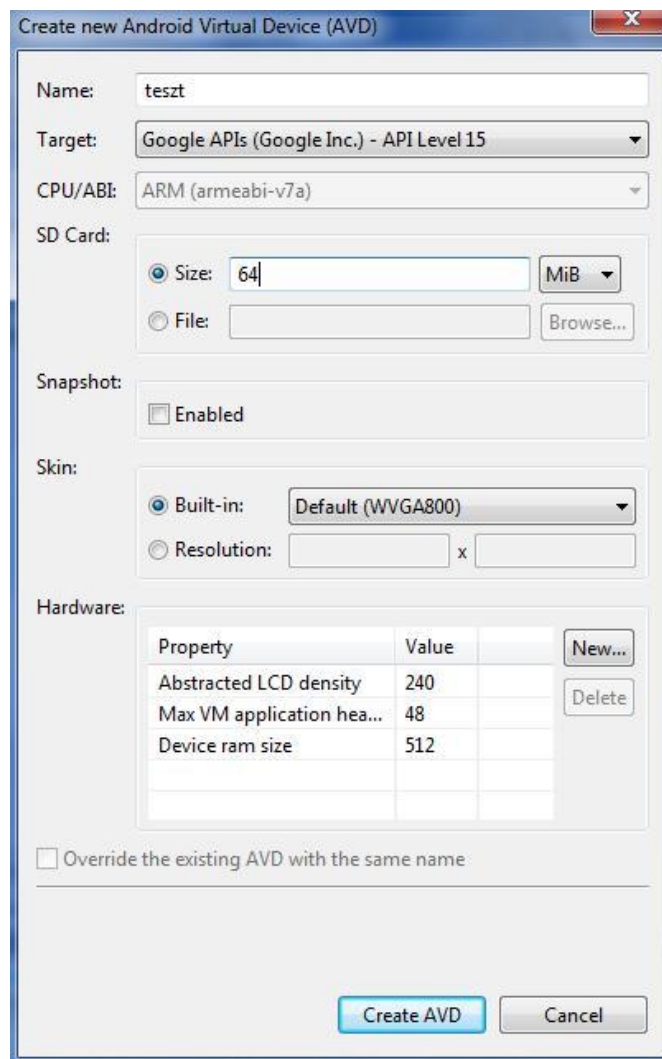
10. ábra: SDK Manager felépítése

Jól látható hogy milyen részekből áll egy adott verzió. Itt ellenőrizhetjük le, hogy valóban letöltöttük-e azt az Android verziót, melyben fejleszteni szeretnénk és melyben virtuális gépet szeretnénk indítani, az AVD-manager segítségével.

2.4.1.4. AVD-manager

Az AVD-manager használata erősen ajánlott, mivel a teszteléshez elengedhetetlen.

Igaz USB-n csatlakoztatott mobil készülékkel is tesztelhető a project, de sokkal biztonságosabb egy virtuális rendszeren tesztelni mintsem egy valóson.



11. ábra: Új AVD készítése

A Name-nél megadhatjuk a virtuális készülékünk nevét, majd alatta, hogy melyik Android verzió szeretnénk tesztelni, azaz milyen operációs rendszer fusson az emulátoron. Megadható még az SD-kártya mérete is, valamint megjelenítési mód is. Ha kész vagyunk már csak létre kell hozni, majd elindítani. Nagyon lassan fog felépülni az rendszer, cserébe egy teljes értékű Android rendszert kapunk, minden funkciójával.

2.4.2. Dalvik-gép

A Dalvik-gép részletes bemutatása megtalálható az alábbi helyen: [10]

A Dalvik-gép futtatja le az Androidra készült programokat, ugyanis ő az Android rendszer egy virtuális számítógépe, azaz rajta futnak le a programok. A futtatás indításakor a .class fájlok .dex fájlra konvertálódnak, melyet a Dalvik-gép már könnyedén tud kezelni. Nagy előnye hogy kis erőforrású készüléken is tökéletesen működik, mivel a tervezése során ezt is

figyelembe vették.

Fontos megemlíteni, hogy ellentétben a Java alapú virtuális gépekkel, nem vermet használ, hanem regiszter-bázisú architektúrát. Ezért a Dalvik-gépet úgy módosították a többi virtuális géphez képest, hogy alacsony memória-felhasználás történjen.

1. A helymegtakarítás érdekében a felesleges részeket eltávolították, azaz tényleg csak azt hagyták a virtuális gépben ami elengedhetetlen a hibátlan működéshez.
2. Továbbá csak 32 bites indexek használatát engedélyezték, éppen ezért a constant pool táblát ez alapján módosították.
3. Nem utolsó sorban pedig bevezették a 16 bites utasításkészletet. Ennek eredményeképpen, már külön utasítások nélkül is képes a virtuális gép a lokális változókkal dolgozni.

2.4.3. Az alkalmazás felépítése

Ezen alfejezet célja egy alkalmazás felépítésének bemutatása. Ugyanis ha valóban jó programot szeretnénk készíteni, akkor mindenképpen ismernünk kell, hogy hogyan is épül fel egy projekt.

Először is szeretném megemlíteni az `@override`-ot, mely egy annotáció. Akkor fogjuk használni a programban, ha az aktuális metódusunkkal az őssztály egy metódusát szeretnénk felüldefiniálni. Ezt azért fontos megemlíteni, mert a kódban nagyon sok helyen fogunk vele találkozni.

2.4.3.1. Activity

Az adott témáról bővebben itt olvasható: [11] [49.oldal]

Az alkalmazáskomponensek közül ezt használjuk legtöbbször. Önálló felülete van, de egy alkalmazáson belül több is szerepelhet belőle, sőt általában több szokott szerepelni. Az Activity életciklusa

Egy Android alkalmazás több egymáshoz kapcsolódó Activity-ből épül fel, de van egy fő Activity (MainActivity) mely indításkor látható, innen érhető el a többi Activity.

Ha egy új Activity-t indítunk akkor az előző leáll és a rendszer a Back Stacken tárolja. A Back Stack egy "last in first out" verem, mely a példányosított Activity-eket tartalmazza. Ha egy Activity-ből visszalépünk, akkor ő kikerül a veremből és az alatta lévő kerül a verem tetejére, ő lesz aktív. Ha egy Activity egy másik indítása miatt áll le, akkor mind ketten értesítést kapnak az eseményről, egy callback metóduson keresztül. Sokféle callback metódus létezik. Többet között az `onCreate()`, `onStop()`, `onDestroy()`. Az alapján fog reagálni az Activity, hogy melyik értesítést kapta meg.

Három állapotban lehet egy Activity és minden állapothoz tartozik egy megfelelő callback. A három állapot a következő:

1. Resumed: Az Activity használatban van.

2. Paused: Még él az Activity, mivel a másik Activity ami használatban van nem fedi le teljesen.
3. Stopped: Ebben az állapotban is él az Activity, de az aktív Activity teljes előtérben van.

Fontos kérdés hogy mi van akkor, ha kevés a memóriánk? Nos, ebben az esetben különböző Activitykkel más és más módon jár el a rendszer. Van egy prioritási elv, mely szerint a kevésbé fontosakat állítja le a rendszer. De melyik a kevésbé fontos? Az amelyik Stopped állapotban van. Ha továbbra sem szűnt meg a probléma, akkor következnek a Paused állapotban lévő Activityk. A tapasztalatok azonban azt mutatják, hogy memóriahiány esetén nem az Activityket állítja le a rendszer, hanem a teljes processzt.

Ezért fontos hogy ha már előre tudjuk hogy nem lesz szükség az Activityre, akkor mi állítsuk le melyet a finish() függvényhívással tehetünk meg. Ha később ismét szükség lenne rá, akkor a rendszer ismét létrehozza.

2.4.3.2. Az Összetettlista

Az összetettlistáról bővebben: [12] [110.oldal]

Az Összetettlista bemutatásához egy ListActivity osztályt használok, mely teljes képernyős listák bemutatására teljesen alkalmas. Alapértelmezetten tartalmaz egy ListView-t és egy ListAdaptert. A ListAdapter a lista elemeinek tárolásáért és megjelenítéséért felelős. Továbbá felelős az új elemeken végzett műveletekért, mint például a hozzáadás, vagy a törlés. Fontos megemlíteni a getView függvényt, melynek paramétere egy kirendelhető objektum. A működési elve a következő: Kiválaszt az XML-ben leírt elrendezést a sorra, és beállítja, hogy az abban található elemek mit tartsanak. Ennek a segítségével nem csak standard egysoros listákat hozhatunk létre, hanem bármilyet amilyet szeretnénk.

2.4.3.3. XML-feldolgozás

Az XML-feldolgozásról bővebben az alábbi helyen olvashatunk: [13] [245.oldal]

Az XML-t nagyon sokan nagyon gyakran használják, mégpedig azért, mert jól olvasható, azaz felhasználóbarát és az összetett adatstruktúrák helyettesítésére is kiválóan alkalmas. Két fő típusú XML feldolgozás használata terjedt el, ezért használatuk ajánlott.

Az első, amiről beszélni szeretnék a SAX alapú feldolgozó. Őt a SAXParse osztály reprezentálja. Sikere abban rejlik, hogy nem tölti be az teljes XML állományt a memóriába, hanem az állomány folyamatos feldolgozás alatt áll, azon módon, hogy egyes csomópontokhoz és egyéb fontos részekhez érve, különböző események generálódnak le, melyekre, ha szükséges, akkor feliratkozhatunk.

A másik feldolgozási forma a DOM. Megvalósítását a DocumentBuilder és a DocumentBuilderFactory végzi. Ellentétben a SAX-al, itt a teljes XML a memóriába kerül, egy fa jellegű strukturális állapotban, melyből lekérdezhetővé válnak az elemek.

2.4.3.4. Manifest és Erőforrás állomány

Legfőbb szerepe hogy definiálja az alkalmazás komponenseit. Ez egy XML állomány, mely információt ad a rendszernek hogy milyen komponenseket tartalmaz az alkalmazás és milyen külső események fontosak illetve mik tartozhatnak ide. Ezen felül itt van definiálva a minimális verziószám is, illetve azok a jogosultságok, melyek az alkalmazás teljes működéséhez szükségesek. Ilyen például az internetelérés, vagy a GPS is.

Minden Android projektben megtalálható egy res elnevezésű erőforráskönyvtár, melyben az XML-ben leírt erőforrások és egyéb médiaelemek találhatóak. Néhány dolog, amit tartalmazhat (ezen felül persze még sok egyéb dolog is megtalálható itt) :

1. Képek, hanganyagok illetve animációk.
2. Menük és témák, illetve stílusok.

2.4.3.5. Service és ContentProvider

A Service legfőbb tulajdonsága hogy nincs felülete, de indíthat Activityket. Az alábbi funkciókat szolgáltatásokkal valósítjuk meg:

1. Háttértartalom szinkronizálása
2. Zenelejátszás
3. Nagyobb tartalomletöltések

Az Android rendszerben a szolgáltatásoknak rendkívül nagy szerepe van, ezért fejlesztésük és nagy körültekintést és szakértelmet igényel. Gyakori hiba, hogy olyan szolgáltatást írunk, mely túlságosan lefoglalja az erőforrásokat, vagy éppen lemeríti az akkumulátorunkat.

A ContentProvider egy tartalomkiszolgáló komponens, feladata az adatforrás kezelése illetve a rá irányuló kérések kiszolgálása.

III. Piaci igények felmérése

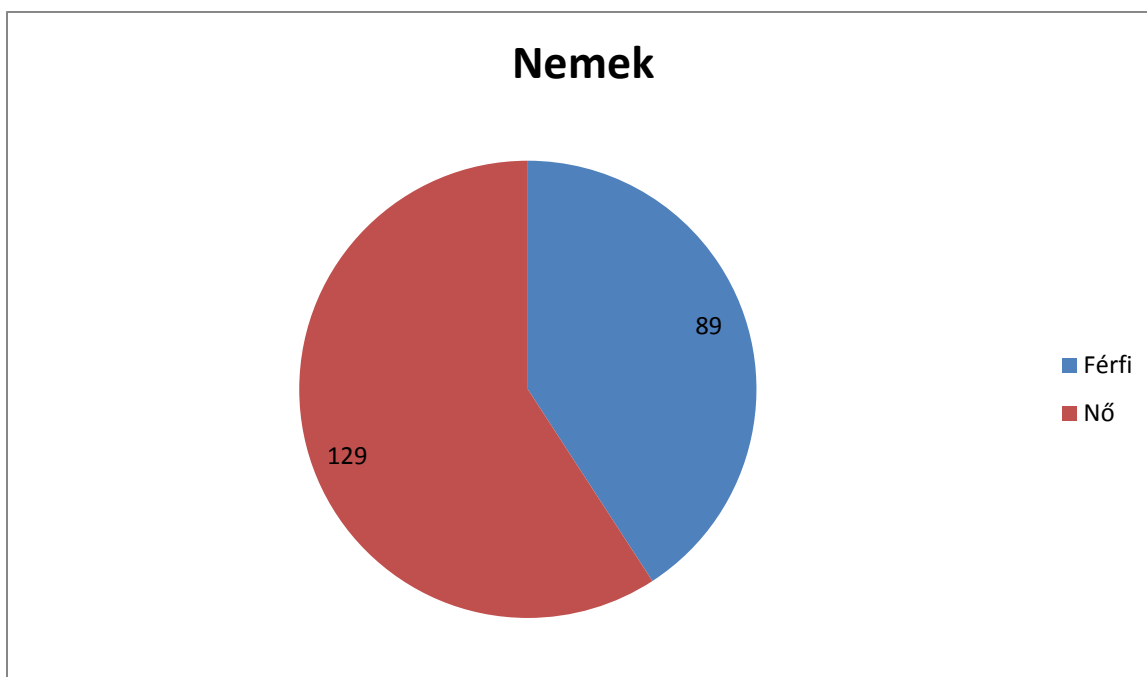
Ez a fejezet két, egymástól eltérő témakörből fog állni:

- a saját kérdőíves kutatásunk elemzése és bemutatásra,
- három, a miénkhez hasonló alkalmazás elemzése és bemutatása.

III/1. A kérdőíves kutatás eredményei

2014 első felében folytattunk kérdőíves kutatást, melyet összesen 218 fő töltött ki. A kutatás felét az egri Almagyardombi Kollégiumban töltöttük ki, míg a másik felét interneten (ww.ripet.hu segítségével). A kérdőív 15 kérdésből állt, melyeket röviden be szeretnénk mutatni:

1. Adja meg nemét!

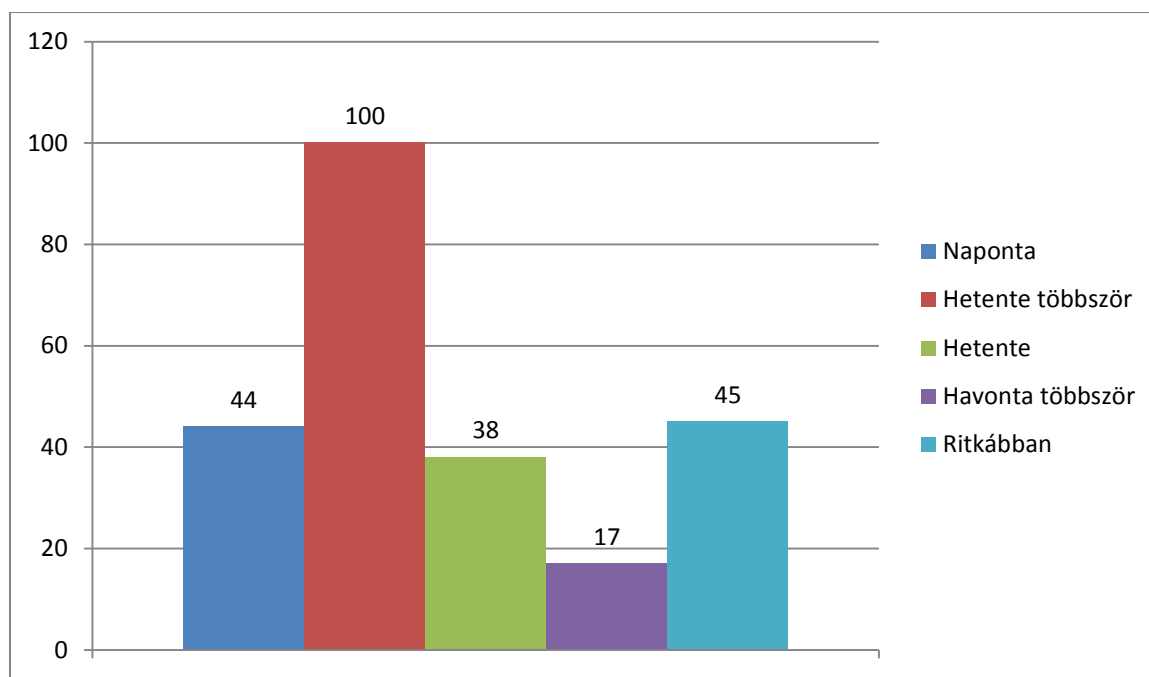


A kérdőívünket kitöltők 41%-a férfi, míg 59%-a nő volt. Ez pontosabban 89 főt és 129 főt jelent a nemek eloszlásában.

2. Adja meg életkorát!

A kérdőívet kitöltők nagyrésze a 18-49 éves korosztályba esik, akik egyébként potenciális célcsoportja szoftverünknek. A kérdésre válaszolók átlagéletkora 24,76 év.

3. Milyen gyakran sportol?

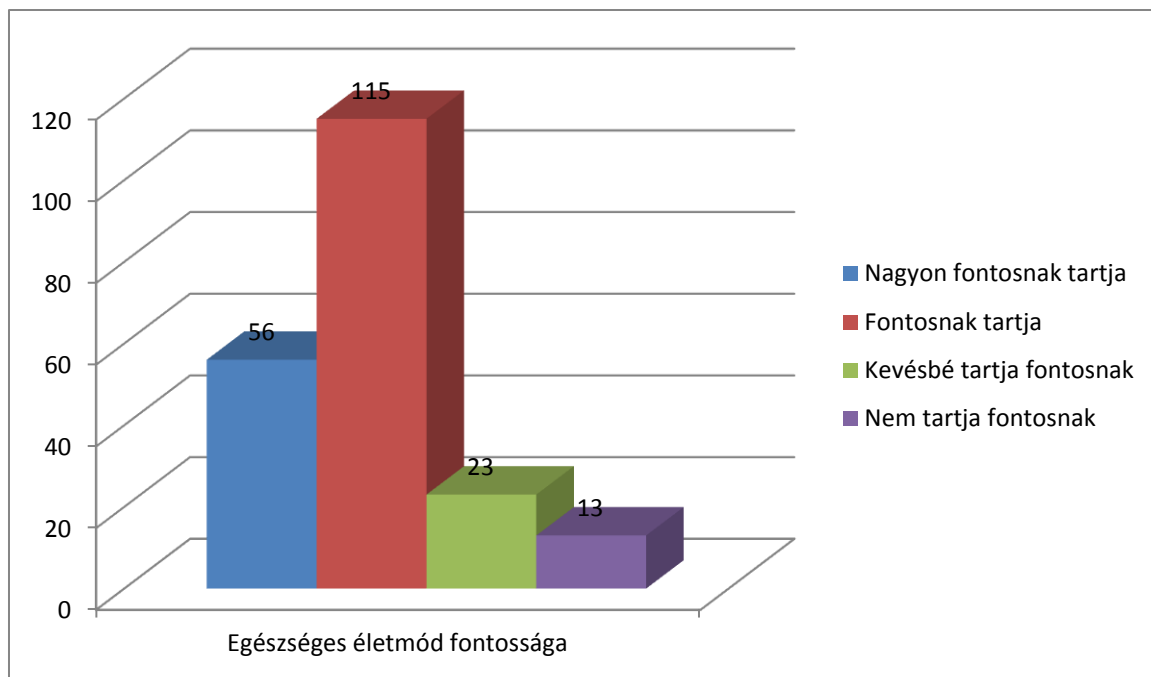


A kérdőívünket kitöltők nagy része legalább heti rendszerességgel sportol, ami egy üdvözlendő adat.

Felmerülhet a kérdés, hogy vajon ezen adatra miért van nekünk szükségünk. A válasz az, hogy (ahogy a IV. fejezet 3. pontjában is lesz szó a szoftver további lehetőségeiről) terveink közt szerepel a szoftverünk továbbfejlesztése. A biztos alapokat tekintve (adatbázis, leolvasó szoftver) megállapítható, hogy nem csak ételérzékenyek javára válhat az alkalmazásunk, ugyanis némi módosítással az aktívan sportolók számára is értékes lehet az app. Konkrét célunk, hogy az app. el is tudja menteni a naponta bevitt kalóriát, ki tudja kalkulálni, hogy az egyénnek hány kalóriát kell/kelljen napi szinten bevinni.

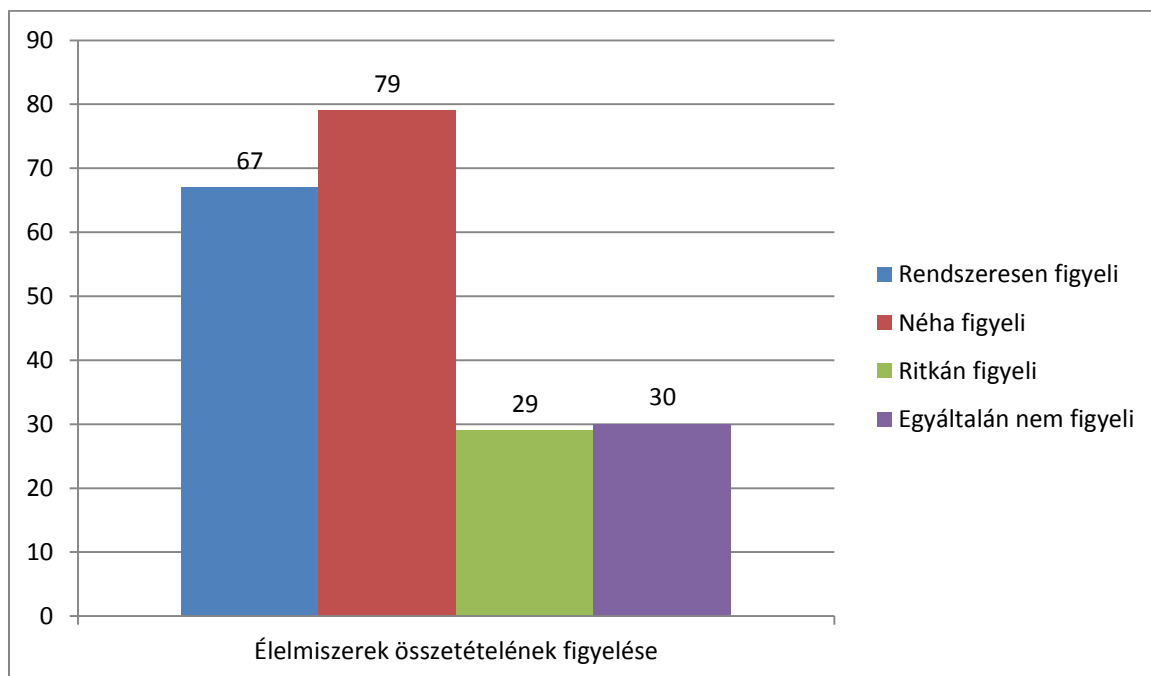
Ez azonban még előttünk áll, ezért sem kívánunk erről most bővebben beszélni.

4. Mennyire tartja fontosnak az egészséges életmódot?



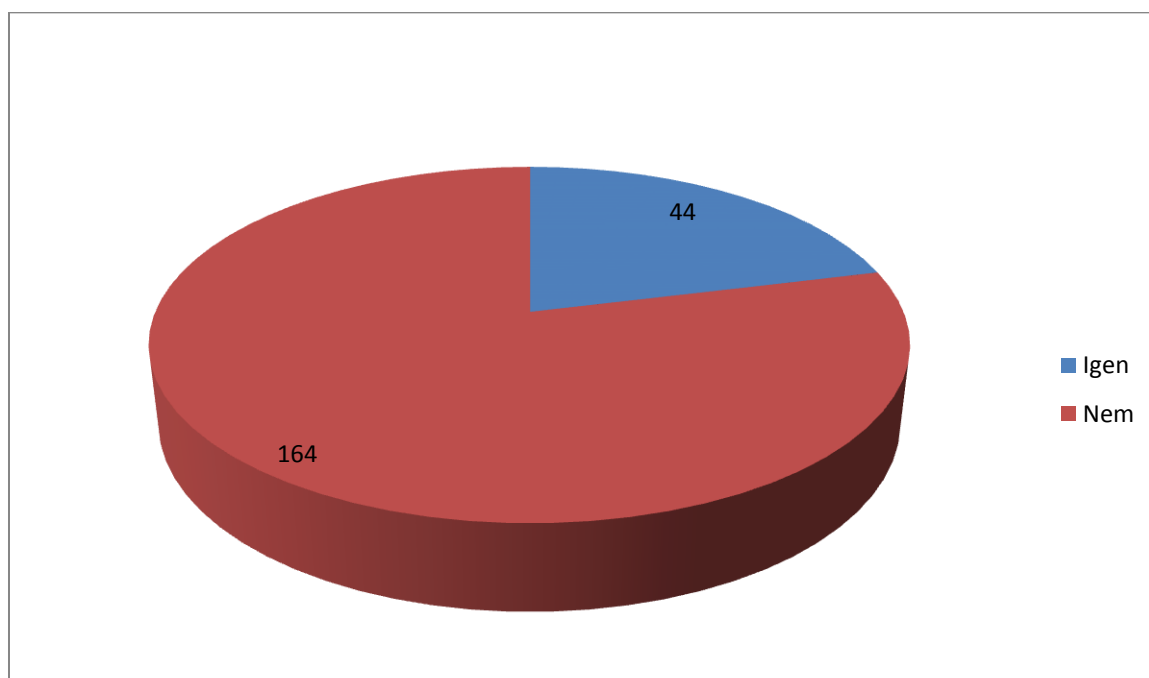
A válaszadók döntő többsége fontosnak tartja az egészséges életmódot, ami egy jó eredménynek számít a számunkra.

5. Mennyire szokta figyelni az élelmiszerek összetételét?



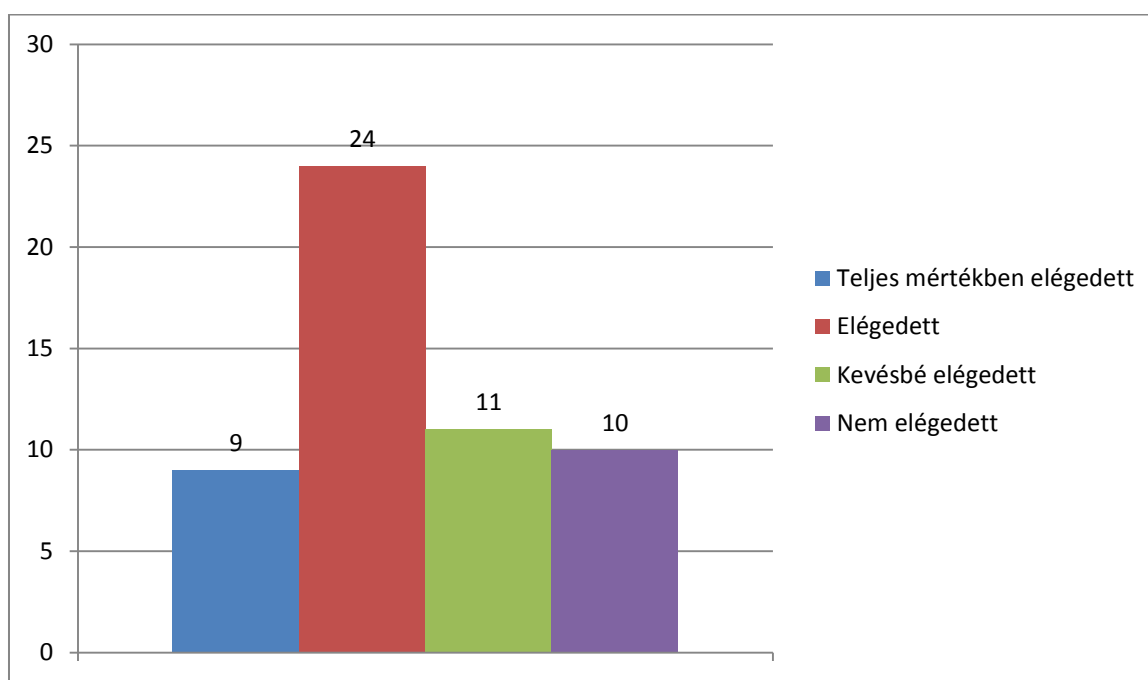
Számunkra meglepetés volt ezen kapott adat, természetesen pozitív meglepetés, hogy a válaszadók milyen nagy aránya figyel a vásárlások alkalmával az egyes élelmiszerek összetételét.

6. Használt már olyan szoftvert (programot, telefonos alkalmazást), amelyben volt lehetőség rögzíteni az elfogyasztott élelmiszereket?



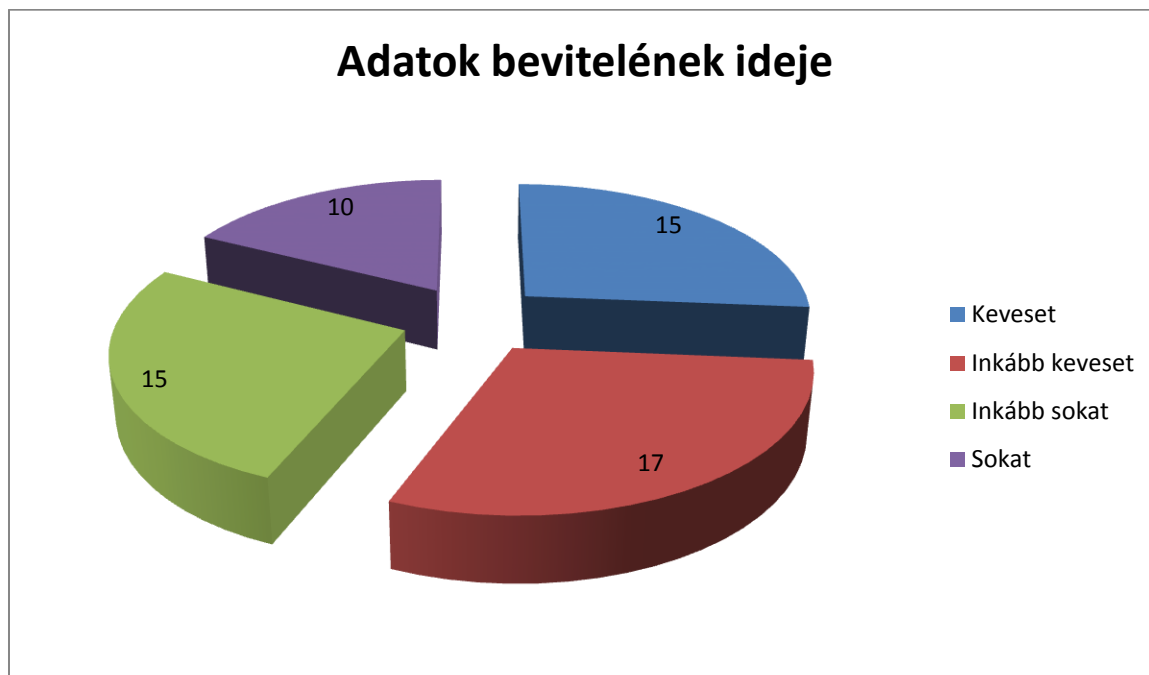
A válaszadók döntő többsége azt válaszolta, hogy még nem használt ilyen szoftvert. Ebből arra tudunk következtetni, hogy Magyarországon még nem leterjedtek az ilyen típusú okostelefon alkalmazások. Ez az előnyünkre válhat.

7. Amennyiben használt már ilyen szoftvert, mi a véleménye erről?



A válaszadók száma – a feliratokból is láthatóan – nem kifejezetten magas, azonban elmondható, hogy megosztott az ilyen típusú szoftverről az emberek véleménye. Ez nem meglepő, hiszen még az alkalmazások „gyerekcipőben” járnak.

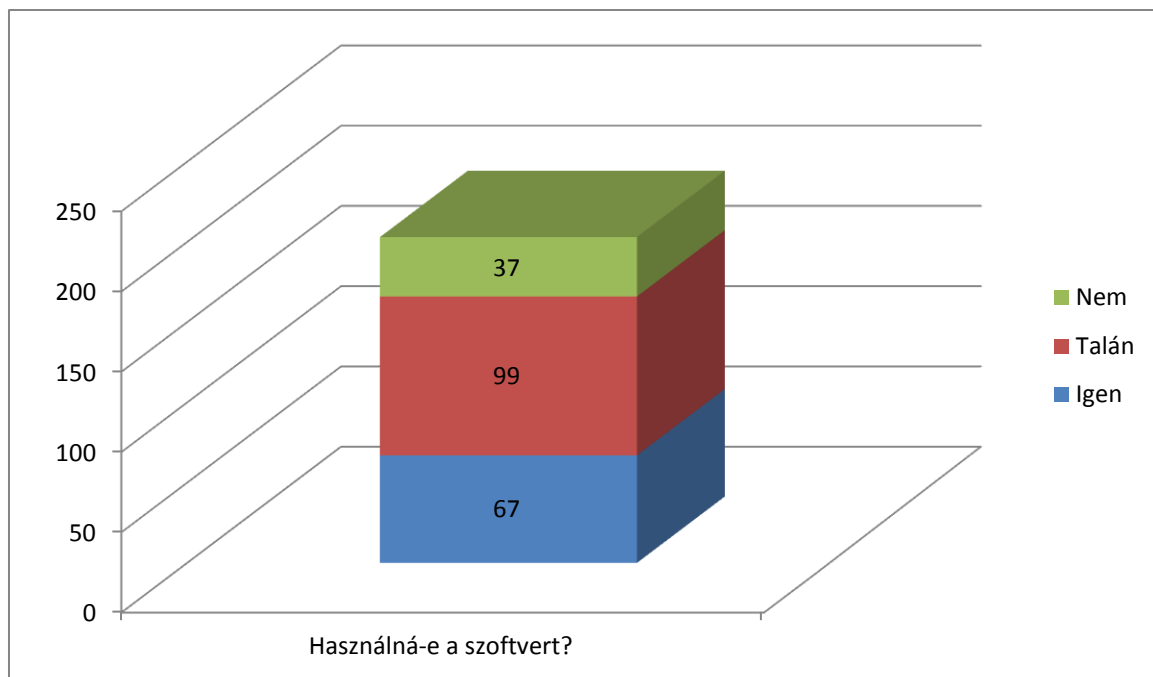
8. Mennyi időt vett igénybe az elfogyasztott élelmiszerek bevitele a szoftverbe?



Hipotézisünk az volt a kérdőív kitöltése előtt, hogy a jelenleg is használatban lévő hasonló szoftverek egyik fő problémája, hogy nagyon nehézkesen és csak nagy időráfordítással lehet az egyes adatokat bevinni.

E hipotézisünkre kapott válaszok alapján elmondható, hogy a válaszadók nagyon megosztottak ebben a kérdésben, és mivel megosztottak, mi ezt úgy értelmezzük, hogy e problémakör megoldása versenyelőnyhöz juttathat mindek, ezért ezt kiemelkedően fontos részként kezeljük a szoftverünkben.

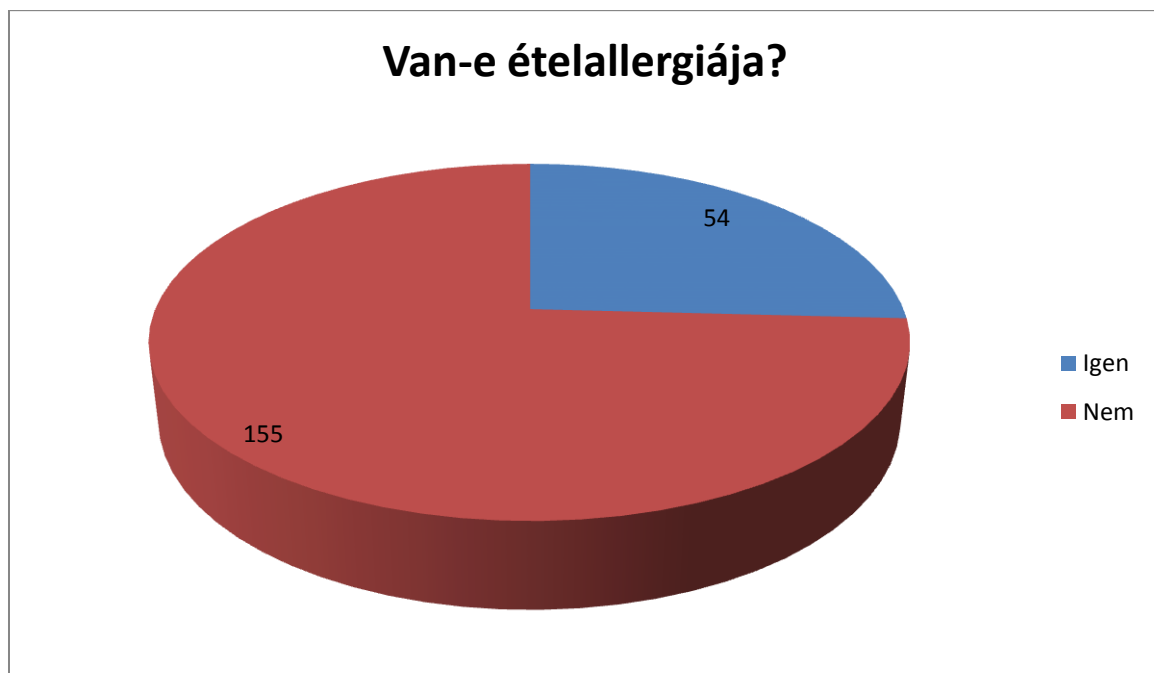
9. Ha lenne lehetősége egy olyan szoftvert használni, amivel egyszerűen és gyorsan tudná az elfogyasztott élelmiszereket rögzíteni, akkor használná?



Kérdésünk lényegre törő volt, azt szeretnénk volna megtudni, hogy a kérdőívet kitöltő emberek mekkora része használna olyan szoftvert, ami rögzíti az elfogyasztott élelmiszereket.

A kapott adatokból arra következtetünk, hogy van igény ilyen szoftverre. A mi esetünkben ez ugye a szoftver egy része lesz.

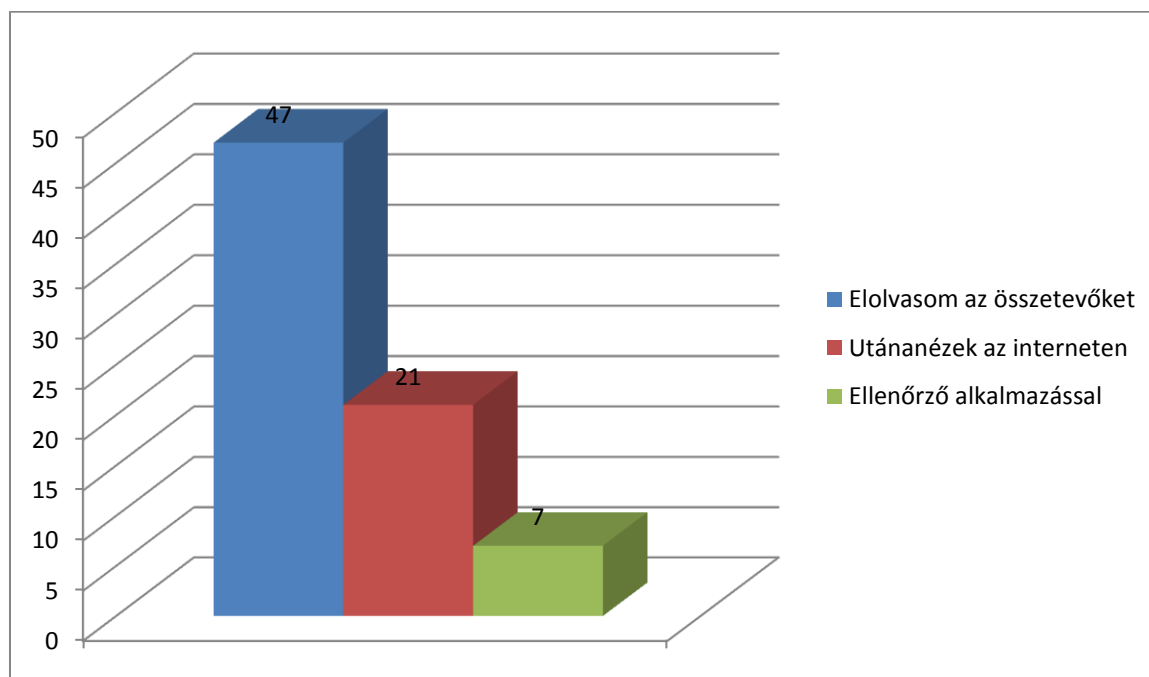
10. Van bármilyen ételallergiája?



A kérdőívünket úgy készítettük el (választottuk ki a kérdéseket, töltöttük ki), hogy az 100 %-ig reprezentatív legyen.

A kapott adatok alátámasztják az a tényt, hogy napjainkban az allergiások (ételallergiások) száma igen magas. Az általunk kapott adatok alapján a válaszolók 25,71 %-a szenved valamilyen ételallergiában. Vagyis a válaszadók negyede, ami kiemelkedően magas szám.

11. Amennyiben igen, hogyan szokta ellenőrizni az élelmiszerek összetételét?

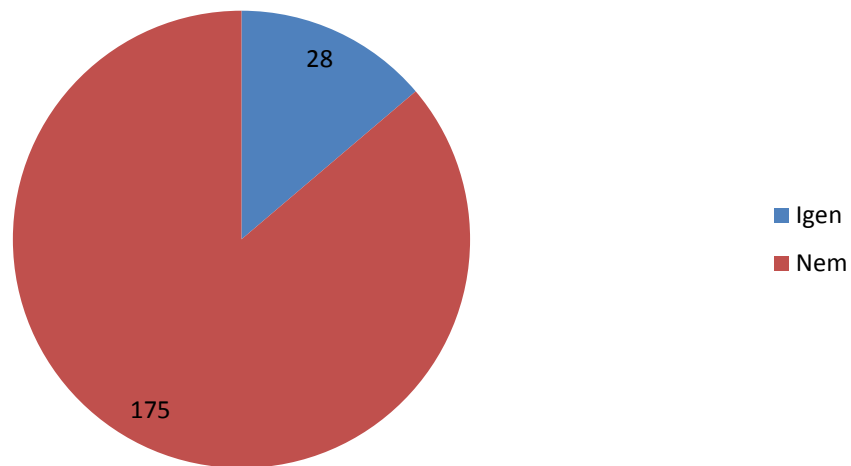


E kérdés kérdőívünk legfontosabbika. Arra voltunk kíváncsiak, hogy milyen módszereket alkalmazva tudják meg az emberek az adott élelmiszer összetevőjét.

Látható, hogy döntő többségük elolvassa az összetevőket, valamivel kevesebb utánanézne az interneten, és 7 válaszadó nyilatkozott úgy, hogy valamilyen alkalmazással ellenőrzi az élelmiszerek összetételét. A kapott válaszok itt is pozitívak voltak a projektünk szempontjából.

12. Ismer olyan szoftvert, amely megkönnyíti az élelmiszerek ellenőrzését? (Például, hogy tartalmaz-e allergén anyagot, stb.)

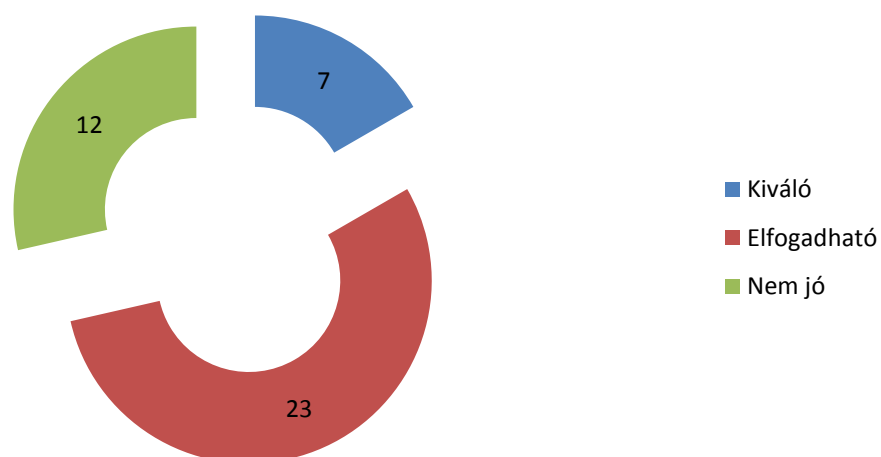
Ismer-e termékválasztást segítő szoftvert?



A kérdőív másik központi kérdése, hogy mennyire szűz a terület, hogy mennyire lesz innovatív az alkalmazásunk. A válaszok itt is kielégítőek voltak, hiszen csak közel 15 %-a a válaszadóknak tud a miénkhez hasonló szoftverről. Bízunk benne, hogy ez az adat (a mi szoftverünk hatására) meg fog a jövőben változni, pozitív irányba.

13. Ha igen, mi a véleménye erről a szoftverről?

Vélemény a miénkhez hasonló szoftverekről



Ezen eredményt többféleképpen is lehet értelmezni. Egyrészt számunkra negatívum, hogy a miénkhez hasonló szoftverekről inkább nem jó a véleménye a válaszadóknak. Másrészt ez az

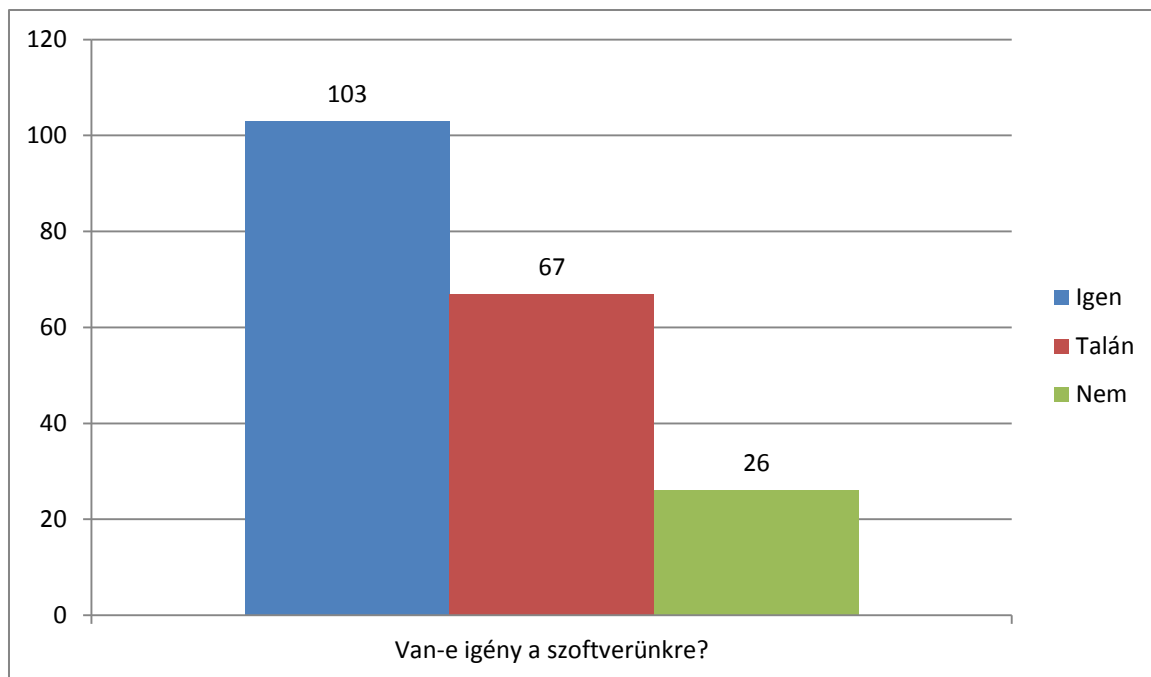
előnyünk is lehet, hiszen nem voltak megelégedve a „konkurenciánkkal”, amit a mi javunkra tudunk fordítani. A nem pozitívan válaszolók problémájának tekinthették a korábbi kérdésekben is írtakat; a nehezen kezelhetőséget, az időigényességet, a bonyolultságot. Mi e hibákat ki fogjuk a javítani.

14. Sok időt vesz igénybe az egyes élelmiszerek ellenőrzése, allergén anyagok szempontjából?



Hipotézisünk az volt (többek közt), hogy az időigényesség nagy problémája a szoftvereknek. Ezt az ötletünket azonban a válaszadók megcáfolták, hiszen az olvasható ki az adatokból, hogy többségük szerint nem időigényes az összetevők ellenőrzése.

15. Ha lenne lehetősége egy olyan szoftvert használni, amivel egyszerűen és gyorsan megtudhatná, hogy fogyaszthatja-e az adott élelmiszert, akkor használná?



Némileg sarkítva, de azt szeretnénk volna az utolsó kérdésben megtudni, van-e igény a szoftverünkre.

A kapott válaszok alapján az mondható (a másik 14 kérdést is idevéve), hogy van igény a szoftverünkre. Ezen adatok nagyon fontosak voltak a számunkra, hiszen arról adtak visszajelzést, hogy van értelme a munkánknak, van értelme ilyen okostelefon alkalmazás fejlesztésének.

III/2. Három, a miénkhez hasonló alkalmazás elemzése és bemutatása

Röviden, pár mondatban be szeretnénk mutatni már megvalósult szoftvereket, melyek hasonló célok eléréséért lettek létrehozva.

Csak angol nyelvű szoftvereket tudunk bemutatni, ugyanis még nincs Magyarországon a miénkhez hasonló szoftver.

Fontos megemlíteni, hogy kutatásunk (és szoftverfejlesztésünknek) nem célja más szoftverek másolása, vagy azokról negatív vélemény nyilvánítása. Egyszerűen csak a három, legáltalánosabb hibát (hiányosságot) szeretnénk bemutatni.

III/2/1. Healthy Food, Allergens, GMO [14]

E szoftver nagy hibája, hogy fizetős.

Az okostelefon alkalmazások 95%-a ingyenes, úgy gondoljuk, hogy nem életképes fizetős szoftver létrehozása.

A megoldásunk erre az lesz, hogy ingyenesen lehet letölteni és használni szoftverünket. Az elsődleges bevételi forrásnak a reklámokat gondoljuk, nem a fizetőssé tételét az alkalmazásnak.

III/2/2. Food Guard - Allergy Scanner [15]

E szoftver nagy hibája, hogy nincs saját adatbázisa.

Ennek is köszönhető, hogy nem kapott jó értékeléseket, hiszen olyan, nem saját adatbázisból hívja le a szoftver a szükséges adatokat, mely nem biztos, hogy hiteles és helytálló információkat tartalmaz. Illetve nagy hátránya a nem saját adatbázisnak, hogy ki van szolgáltatva az adatbázis tulajdonosának is.

A mi megoldásunk erre, hogy olyan saját adatbázist építünk, melyben olyan termékek szerepelnek, amelyek adatait, összetevőit mi vittük fel az adatbázisba. A adatbázis saját szerveren van, melynek elérhetőségét mi tudjuk befolyásolni.

III/2/3. Food Allergy Detective [16]

E szoftver nagy hibája, hogy csakis "kézzel" lehet bevinni az adatokat.

Sikertelenségének okát abban látjuk, hogy túlságosan sok időt és energiát igényel a felhasználóktól az egyes termék adatainak bevitele (pl. a termék neve), így inkább más módszerekre támaszkodnak.

A mi megoldásunk erre, hogy olyan vonalkód leolvasót építettünk be a szoftverbe, mely kézzel történő "bepötyögés" nélkül is képes a termékek adatait kimutatni, elmenteni. Ezzel sok időt és energiát tudunk megspórolni a felhasználóinknak.

IV. A szoftver

Ebben a fejezetben fogjuk bemutatni a munkánk eredményét, az okostelefon alkalmazást, a szoftvert.

A program a leolvasott vonalkód segítségével meghatározza, hogy a vonalkódhoz tartozó termékben található-e olyan összetevő, melyre a felhasználó allergiás.

Valójában a szoftver ettől jóval bonyolultabb, illetve összetettebb.

Ugyanakkor rengeteg nehézségbe is ütköztünk (és ütköztünk is) a fejlesztés során.

Az alapelv az volt, hogy a fejlesztett szoftver gyors és felhasználóbarát legyen.

Miután kutatást végeztünk, arra az eredményre jutottunk, hogy egyre nagyobb az igény egy ilyen Androidos alkalmazásra.

Elsődleges szempont a gyorsaság. A termékek ellenőrzése csak úgy valósítható meg gyorsan, ha a vonalkód-leolvasás után azonnal megkapja a felhasználó az eredményt.

De vajon ezt hogyan lehetne megvalósítani? Hosszas kutatás után rátaláltunk a zxing-re mely a GITHUB terméke. Ez egy ingyenesen használható vonalkódozó szoftver, mely beleintegrálható az alkalmazásunkba. Vannak bizonyos használati feltételek, de ezek valójában elég kötetlenek ahhoz hogy zavartalanul használható legyen. A zxing licenszéről bővebben itt olvashat: <https://github.com/zxing/zxing/wiki/License-Questions>

A Zxing-et használja a BarcodeScanner nevű alkalmazás is. Ezért mivel ingyenesen elérhető, ezt építettük be a szoftverünkbe.

A vonalkód leolvasása után egy MySql-adatbázishoz csatlakozva kikeresi a vonalkódot az adatbázisban, ha van találat, akkor megnézi, hogy a felhasználó allergiás-e valamely összetevőre. Ez úgy valósítható meg, hogy a felhasználó a regisztrációja során kijelöli, hogy mire/mikre allergiás. Ez természetesen később bármikor módosítható.

Tehát ha a termék megtalálható az adatbázisban, akkor megvizsgálja és visszajelzi a felhasználónak, hogy fogyaszthatja-e az adott terméket vagy sem.

Azonban ha a termék nem található meg az adatbázisban, akkor lehetőséget ad a szoftver a termék felvételére. Azonban itt felmerül egy kérdés: mi van akkor, ha valótlan vagy téves adatokat adnak meg? Ennek a problémának a kiküszöbölésére azt találtuk ki, hogy ne az "éles" adatbázisba tudjon feltölteni a felhasználó, hanem egy másikba, amiből csak akkor kerül át a termék miután általunk ellenőrzésre került, hogy valósak-e az ott megadott adatok.

Továbbá sok embert zavar, ha egy terméken azokat a bizonyos E-számokat látja. Ezért a szoftverben van lehetőség arra is, hogy az adott E-számmra kattintva bővebb tájékoztatást kapjanak az adott összetevőről.

Maga a szoftver kialakításánál az volt a fő szempont hogy letisztult külsőt használjunk, ezzel is azt sugallva a felhasználóknak, hogy itt inkább a tartalom a lényeg és a szoftver tudása, nem pedig a külső. A következő fejezetben részletes bemutatásra kerül a szoftver, az adatbázis illetve a fejlesztésekhez használt szoftverek.

A szoftver felépítése:

A MANIFEST állomány:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="jager.vcodereader"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="19" />

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name="jager.vcodereader.MainActivity">

        </activity>
        <activity
            android:name="jager.vcodereader.HomeActivity"> </activity>
        <activity
            android:name="jager.vcodereader.fomenu"> </activity>
        <activity
            android:name="jager.vcodereader.Informaciok"> </activity>
```

Ebben a fájlban nagyon sok fontos információt találunk. Itt tudjuk megadni, hogy milyen engedélyeket kap a szoftver, azaz hogy hozzáférhet-e a kamerához, az internethez, illetve egyéb dolgokhoz.

Ha itt nem adunk meg egy engedélyt az alkalmazásnak, akkor a szoftver nem fog tudni olyan funkciókat használni, amik ehhez tartoznak.

Továbbá itt adhatjuk meg a minimum SDK-verziót is, illetve a futtatni kívánt activity-eket is.

HOMEPAGE.XML



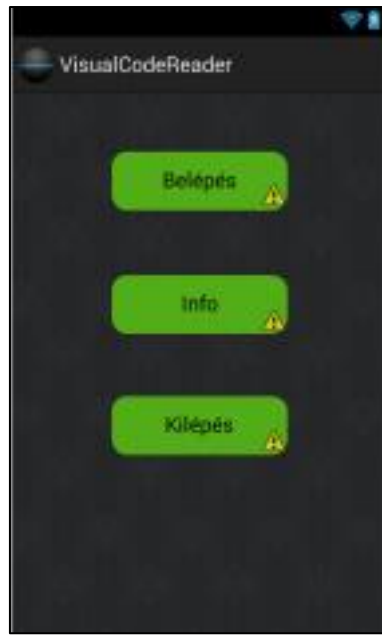
12. ábra: A szoftverünk kezdőlapja

A program indulását követően a homepage.xml fájl által definiált képernyőt fogjuk látni. Ez lesz a kezdőképernyőnk. Itt nagyon fontos, hogy a kezdőképernyőnk megfogja a felhasználót, ugyanakkor logikus, egyszerű és elegáns is legyen.

A kezdőképernyő tartalmazza a szoftverünk nevét, illetve egy Buttont, amire kattintva beléphet a szoftverbe a felhasználó.

Miután rákattintottunk a Buttonra, átkerülünk a fomenu.xml képernyőre.

FOMENU.XML



13. ábra: A szoftverünk főmenüje

Igazából ez lesz az első olyan activity aminek valós funkciója van. Itt a felhasználó kiválaszthat, hogy belép, információt szeretne a szoftverről és annak használatáról, vagy esetleg ki szeretne lépni az alkalmazásból.

Ha a belépésre kattint, akkor az Activity_main.xml fog betöltődni.

Activity_main.XML



14. ábra: Leolvasás

Itt található a szoftver legfontosabb Button-ja, a Leolvasás. Erre kattintva történik a vonalkódolvasás, illetve itt kap tájékoztatást a felhasználó, hogy a kiválasztott terméket fogyaszthatja-e vagy sem. Azonban erre csak a bejelentkezés után van lehetőség. Ezért a másik Button a regisztráció.

Ha a regisztrációra kattintunk, akkor a main.xml következik.

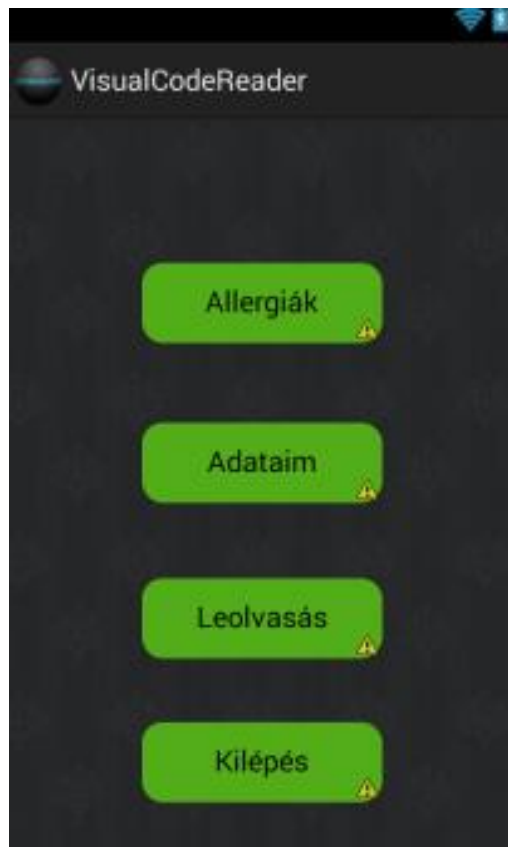
Main.xml



15. ábra: Regisztráció

Ebben az Activity-ben is két Buttont találunk: az egyik a Regisztráció, a másik a Belépés. Ha még nem regisztráltunk, akkor itt megtehetjük, ha viszont már van account-unk akkor elég csak bejelentkezni, melyet a Bejelentkezésre kattintva tehetünk meg. Ekkor a mydata.xml töltődik be.

Mydata.xml



16. ábra: MyMenu

Itt találunk négy Button-t, de amit még nem mutattunk be, az az első kettő, az Allergiák és az Adataim elnevezésűek. Az elsőre kattintva egy oldal nyílik, ahol checkbox-okat találunk. A megfelelő allergia kiválasztásához csak be kell pipálni a megfelelőt.



17. ábra: Allergia checkbox

A másik Button az adataim. Itt van megadva az e-mail cím, életkor, illetve a felhasználó neve. Ez fontos lesz a jövőben is, hiszen az így keletkező ügyfélkör adatai bevételt is termelhetnek, habár nem ez lesz az elsődleges bevételi forrásunk. Mind az allergiák mind az adatok bármikor módosíthatók.

Röviden összefoglalva jelenleg ezeket a funkciókat valósítottuk meg a szoftverben, azonban a későbbiekben további funkciók bevezetését is tervezzük.

A következő részben szeretném bemutatni, hogy az imént bemutatott xml fájlok hogyan kelnek életre. Minden xml fájlban található Button-nál valahol definiálnunk kell, hogy a Button használatakor mi történjen, illetve bármilyen folyamatot szeretnénk indítani, azt egy java állományban definiálhatjuk.

Java állományok:

Button-ok használatakor, ha azt szeretnénk, hogy egy új Activity induljon el, azaz új képernyőre szeretnénk eljutni, akkor a következő kód szükséges:

A screenshot of an IDE window showing the Java code for a class named Startpage. The code is as follows:

```
public class Startpage extends Activity implements View.OnClickListener{  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.homepage);  
  
        Button regbutton=(Button) findViewById(R.id.startbutton);  
        regbutton.setOnClickListener(new OnClickListener()  
        {  
            public void onClick(View v)  
            {  
                Intent intent = new Intent(v.getContext(),menu.class);  
                startActivityForResult(intent,0);  
            }  
        });  
    }  
    private void BelepesClick()  
    {  
        startActivity(new Intent("jager.vcodereader.fomenu"));  
    }  
    public void onClick(View v)  
    {  
        switch(v.getId())  
        {  
            case R.id.startbutton:  
                BelepesClick();  
                break;  
        }  
    }  
}
```

18. ábra: Kezdőkép

A BelepesClick-ben adjuk meg, hogy melyik java állomány induljon el a kattintásra. Hasonlóan működik a többi Button esetén is.

Viszont van egy fontos kérdés, ami felmerülhet: honnan kéri le a termék információit a vonalkód olvasó?

Nos, a válasz az hogy, egy Adatbázisból. Jelenleg egy MySql adatbázist használunk, mely már most több mint ezer darab terméket tartalmaz, minden adatával együtt. Minden adat alatt értem itt az összetevőket, gyártókat, kalóriatáblázatokat, a termék nevére és nem utolsó sorban a termékcsoportha, azaz milyen élelmiszer csoportba tartozik a termék.

Az Adatbázis:

Az adatbázis létrehozása kulcsfontosságú volt a szoftver elkészülésében. Ugyanis ha nincs adatbázis, akkor nincs honnan lekérni az adatokat.

Szerencsére kaptunk segítséget Tari Tímeától, aki a SPAR Magyarország Kft. Területi vezetője. Sikerült egy megállapodást kötnünk, melynek eredményeként sikerült több mint 1000 terméket regisztrálnunk az adatbázisunkba.

Az adatbázist a fejlesztési időszak alatt Wamp szerveren tároljuk, melynek lényege, hogy az adatbázist offline tudjuk elérni mivel az a számítógépünkön, tárolódik.

A Wampserver teljes mértékben ingyenes szoftver, így minden funkciójával együtt teljes mértékben használható. A gépünkön létrehoz egy virtuális Apache szerver, melyen MySQL adatbázist tárolhatunk, továbbá php lekérdezéseket is indíthatunk az adatbázisunk felé. Az adatbázisunkat phpmyadmin-on keresztül tudjuk kezelni. Ez egy felhasználóbarát környezet, viszont van lehetőségünk sql parancsok használatára is.

Az adatbázis a következő táblából épül fel:

Gyártó:

Ez a tábla tartalmazza a gyártók neveit illetve a hozzá tartozó ID-t.

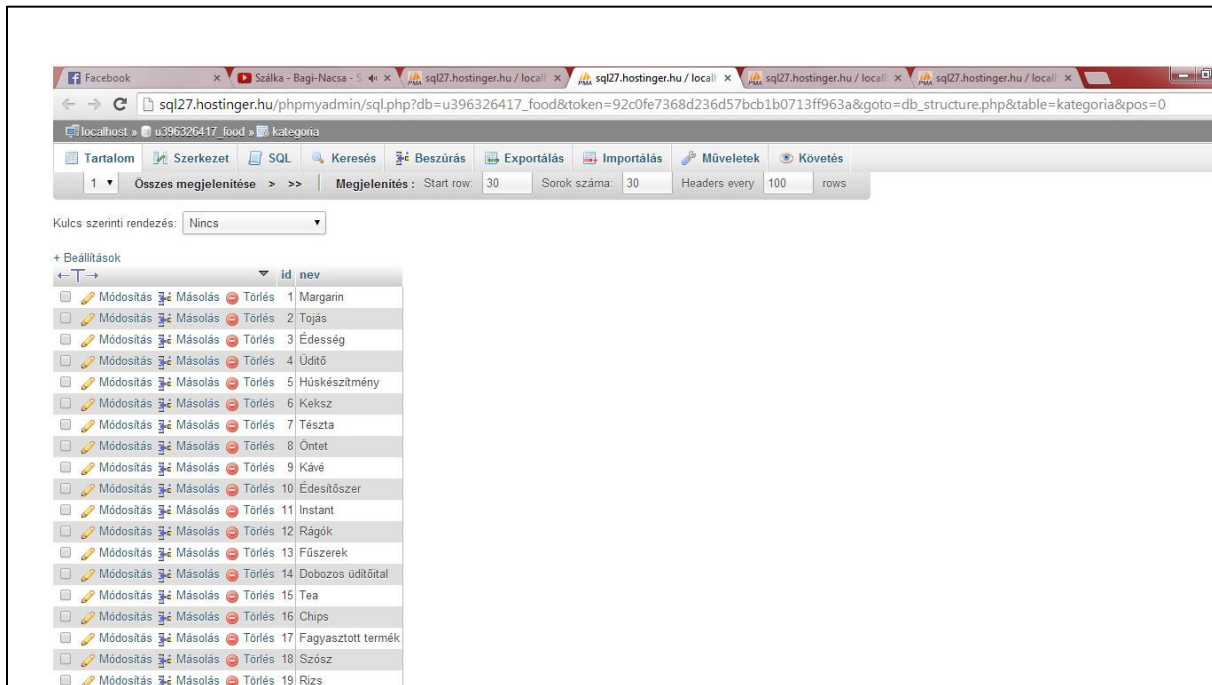


	id	nev
☐	1	Natura Margarin Kft.
☐	2	Vital Eggs Kft.
☐	3	Unilever Magyarország Kft.
☐	4	Stühmer Kft.
☐	5	Hell Energy Magyarország Kft.
☐	6	Gramex 2000 Kft.
☐	7	PA-COMP Kft.
☐	8	TELLÉR Kft.
☐	9	Merian Foods Élelmiszeripari Kft.
☐	10	Mondelez Hungária Kft.
☐	11	Penny Market Kft.
☐	12	Landena KG
☐	13	Instanta Sp. Z.o.o.
☐	14	Nissin Foods Kft.

19. ábra: Gyártók

Kategória:

Ez a tábla tartalmazza a termékkategóriákat, illetve az hozzájuk tartozó azonosítókat.



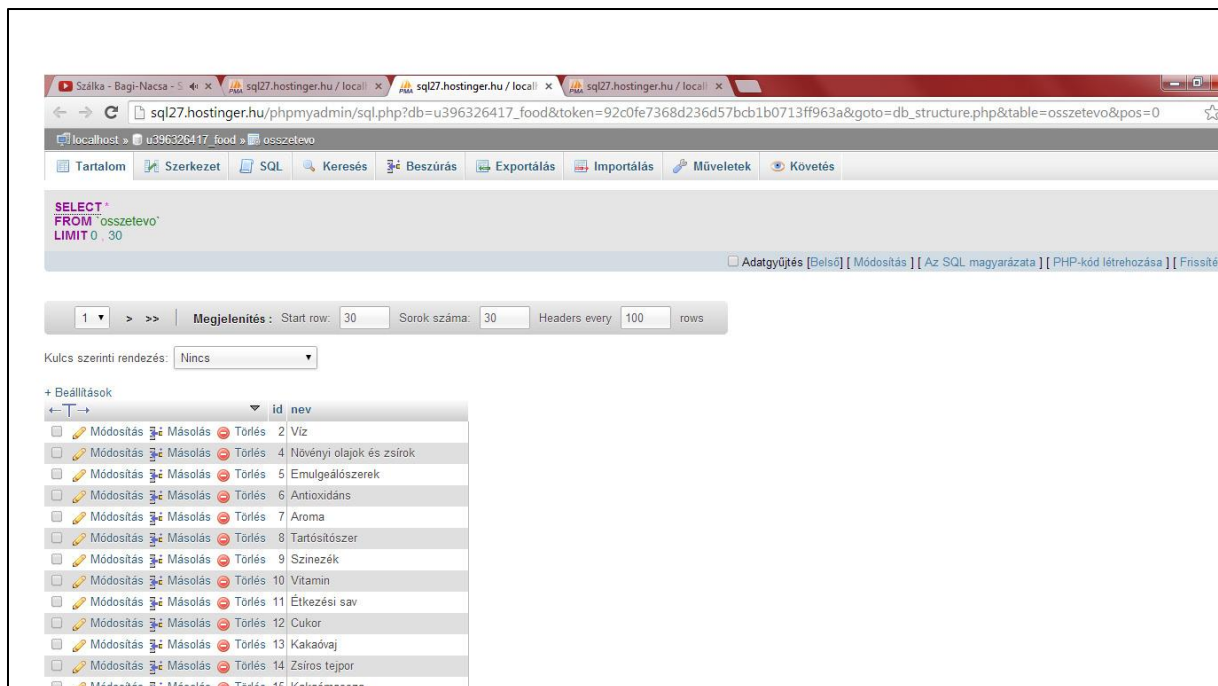
The screenshot shows a web application interface for managing a database. The browser address bar indicates the URL: `sql27.hostinger.hu/phpmyadmin/sql.php?db=u396326417_food&token=92c0fe7368d236d57bcb1b0713ff963a&goto=db_structure.php&table=kategoria&pos=0`. The interface includes a navigation bar with options like 'Tartalom', 'Szerkezet', 'SQL', 'Keresés', 'Beszúrás', 'Exportálás', 'Importálás', 'Műveletek', and 'Követés'. Below the navigation bar, there are settings for 'Megjelenítés' (Display) and 'Kulcs szerinti rendezés' (Sort by key). The main content area displays a table with the following data:

	id	nev
☐	1	Margarin
☐	2	Tojás
☐	3	Édesség
☐	4	Üdítő
☐	5	Hűskészítmény
☐	6	Keksz
☐	7	Tészta
☐	8	Öntet
☐	9	Kávé
☐	10	Édesítőszer
☐	11	Instant
☐	12	Rágók
☐	13	Fűszerek
☐	14	Dobozos üdítőital
☐	15	Tea
☐	16	Chips
☐	17	Fagyasztott termék
☐	18	Szószt
☐	19	Rizs

20. ábra: Kategóriák

Összetevő:

Ebben a táblában kaptak helyet az összetevők. Itt az összes összetevő megtalálható, ami a több mint ezer termékben megtalálható. Az összetevőket is a hozzájuk tartozó ID alapján lehet azonosítani.



21. ábra: Össztevők

Termék:

Ez a legösszetettebb tábla. Itt találhatóak a vonalkódok, ami alapján a program azonosítja a terméket. A tábla tartalmazza a termék megnevezését,

Kalóriatáblázatot, illetve a kategória és a termékcsoporthoz azonosítót.

The screenshot shows a web application interface for managing products. At the top, there's a navigation bar with options like 'Tartalom', 'Szerkezet', 'SQL', 'Keresés', 'Beszúrás', 'Exportálás', 'Importálás', 'Műveletek', and 'Követés'. Below this is a SQL query editor with the following query:

```
SELECT *
FROM 'termek'
LIMIT 0, 30
```

Below the query editor, there's a table of results. The table has columns: id, vonalkod, nev, kj, kcal, fehérje, szénhidrát, zsír, categoria_id, gyarto_id. The table contains 14 rows of product data.

id	vonalkod	nev	kj	kcal	fehérje	szénhidrát	zsír	categoria_id	gyarto_id
1	99	Tejcsokoládés Almaszirom	1996	477	4.2	64	21	16	51
2	99	Spar Premium - Kandírozott vörös áfonya étcsokolád...	1684	401	2.2	66.6	12.4	3	32
3	20009007	Belbake csokoládés puding	466	111	3.1	17.4	3.1	21	35
4	20013639	Vita D'or Sunflower Spread	2960	720	0	0	80	1	50
5	20123109	Pilos Tej	186	44	3	4.7	1.5	31	48
6	20130961	Mikodo örlött fehérbors	0	0	0	0	0	13	35
7	24044882	Sugo alla Bolognese	334	80	5	6.5	3.6	8	12
8	24541244	Chocolate Chip Cookies	2102	502	6.7	61	25	6	18
9	54491472	Coca-Cola 500ml	190	44	0	11.2	0	4	20
10	59901419	Budafok élesztő	0	0	0	0	0	38	45
11	599081783658	Szentkirályi szénsavmentes ásványvíz	0	0	0	0	0	20	30
12	3700123300380	Nestlé Aquarel Cédrus	0	0	0	0	0	20	24
13	4009900018784	Orbit Spearmint chewing gum	621	149	0	62	0	12	16
14	5051140683348	Étkezési lencse	1507	360	25	59.5	1.9	35	17

22. ábra: Termékek

Az adatbázis tartalmaz még felhasználók táblát, melyben a felhasználók adatait találhatjuk meg (erről korábban már bővebben is írtunk), illetve azon összetevők ID-ját, melyekre az adott felhasználó allergiás, intoleráns.

Ahogy korábban már írtuk, az adatbázisunk 100%-ban olyan termékeken alapul, melyek valóságok, valamint az egyes termékek minden adata, értéke megfelel a valóságnak. E megbízható alappillér igazi értéke a szoftverünknek. Emellett nem csak a szoftverünkönél lehet ezt a megbízható adatbázist felhasználni: a közelmúltban (2014. május) megkeresést kaptunk az Eger Food vállalattól. A velük folytatott tárgyalások eredményeképp sikerült megállapodást kötnünk az adatbázisunk használatára. E együttműködés mindkét fél hasznára fog válni a jövőben.

A szoftver jövője és további lehetőségei

Habár már a szoftver kész és működik is (illetve használják is), nem szeretnénk megállni a munkával.

Terveink közt szerepel, hogy a jelenleg több mint 1000 darab terméket tartalmazó adatbázisunkat tovább növeljük. A termékek számának folyamatos növelésének nincs különösebb akadályozója, hiszen szerződésben állunk a SPAR Magyarország Kft-vel, illetve a Forza Market Kft-vel is.

Nem szeretnénk megállni Magyarország határainál, tervünk, hogy nemzetközi szintre lépünk. Ehhez elengedhetetlen lesz egy nagyobb tőkeinjekció, melyet kockázati tőkebefektető bevonásával szeretnénk megvalósítani. 2014 márciusában tárgyaktunk már egy kockázati tőkebefektetővel, aki támogatta a szoftver megvalósítását, illetve további ötletünket. Ekkor abban "állapodtunk meg" (szóban), hogy amint a regisztrált felhasználóink száma elér egy bizonyos szintet (200 fő), vegyük fel ismét a kapcsolatot. Bízunk benne, hogy hamarosan már nem csak a magyar piacon leszünk jelen.

A nemzetközi terjeszkedéshez elengedhetetlen lesz, hogy már ne csak magyar nyelven legyen elérhető a szoftver. Célunk, hogy záros határidőn belül több nyelven is elérhetővé tegyük alkalmazásunkat.

Tervünk, hogy a szoftver ne "csak" ellenőrizze a termékeket az étel-allergiásoknak, de amennyiben allergén anyagot észlel az adott termékben, akkor ajánljon fel alternatívát az étel-allergiásoknak. Ez egy hasznos kiegészítője lesz alkalmazásunknak.

A szoftverhez tartozik egy weboldal (www.caneat.eu) is, ahol bővebb információk és szolgáltatások lesznek elérhetőek a regisztrált felhasználók számára. Folyamatban van a közösségi médiákban való megjelenés is.

A weboldalon szeretnénk létrehozni egy komplex és át fogó tudásbázist, ahol minden elérhető lesz, ami az étellel, az étkezéssel és az alapanyagokkal kapcsolatos.

Szintén a weboldalon lesz elérhető egy olyan modul, melynek feladata kifejezetten az étkezési tanácsadás lesz. Itt lesz majd egy általános rész, ahol a leggyakoribb kérdésekre és felvetésekre lesz válasz adva, illetve lesz egy "egyéni tanácsadás" rész, ahol a felhasználók feltehetik a témakörrel kapcsolatos kérdéseiket.

A II. fejezetben írunk bővebben az Androidról, az iOS-ről és a Windows Phone-ról. És habár ezen utóbbi két cég piaci részesedése nem olyan magas, mint az Androidé, terveink közt szerel, hogy mind Apple-, mind pedig Windows-termékre is elkészítsük alkalmazásunkat.

Szeretnénk továbbá, hogy partneri kapcsolatainkat bővítsük. Habár jelenleg már két nagy cég, a SPAR Magyarország Kft. és a Forza Market Kft. partnerünk, terveink közt szerepel, hogy más cégeket is bevonjunk.

V. Szakirodalmi jegyzék

- [1] – Tóth Gábor: Allergia- és Candida-kalauz, Pilis-Vet Bt., Pilisvörösvár, 2003.
- [2] - http://www.phonearena.com/news/IDC-says-iOS-and-Android-combined-for-87.6-of-the-2012-smartphone-market_id39808
- [3] - <http://kutyu.hu/cikk/96232>
- [4] - <http://www.wirelesscommunicationsc.com/portfolio/apple-iphone-5s/>
- [5] - <http://www.nokia.com/hu-hu/keszulekek/lumia/>
- [6] - <http://www.barato-electronico.com/no-1-s7-and-samsung-galaxy-s5-same-looking-diferente-price/>
- [7] – http://mobilarena.hu/teszt/android_fejlodestortenet/android_tortenelem.html
- [8] - <http://www.androidsis.com/android-2-2-para-los-htc-g1dream-es-ahora-mas-posible/>
- [9] - <http://www.edureka.in/blog/beginners-guide-android-architecture/>
- [10] - <http://hu.wikipedia.org/wiki/Dalvik>
- [11] – Ekler Péter, Fehér Marcell, Forstner Bertalan, Kelényi Imre, Android-Alapú Szoftverfejlesztés, SZAK kiadó, Budapest, 2012.
- [12] – Ekler Péter, Fehér Marcell, Forstner Bertalan, Kelényi Imre, Android-Alapú Szoftverfejlesztés, SZAK kiadó, Budapest, 2012.
- [13] – Ekler Péter, Fehér Marcell, Forstner Bertalan, Kelényi Imre, Android-Alapú Szoftverfejlesztés, SZAK kiadó, Budapest, 2012.
- [14] – <https://play.google.com/store/apps/details?id=com.nxtnutrio.nxtnutrio>
- [15] – <https://play.google.com/store/apps/details?id=com.mangobird.foodguard>
- [16] – <http://www.foodallergydetective.com/>