

# Címtár Felhő

## Projektfeladat specifikáció



**Informatikai Biztonsági és  
Adatvédelmi Tanácsadó Kft.**

# 1 Tartalomjegyzék

1	Tartalomjegyzék.....	2
2	Bevezetés .....	3
2.1	A feladat címe .....	3
2.2	A feladat rövid ismertetése .....	3
3	Elvárások a feladattal kapcsolatban.....	4
3.1	Operációs rendszer, környezet.....	4
3.2	Felhasználható programozási nyelv .....	4
3.3	Megoldás formátuma.....	4
3.4	Szoftverfejlesztés .....	4
3.5	Modulok.....	4
4	Szoftver specifikáció.....	6
4.1	Megjelenés.....	6
4.2	Funkciók .....	6
5	Dokumentáció .....	7
5.1	Erőforrás-terv, munkaidő nyilvántartás .....	7
5.2	Technikai dokumentáció.....	7
5.3	Forráskód dokumentáció.....	7
5.4	Felhasználói dokumentáció .....	7
6	A projekt értékelése.....	8
6.1	A feladat értékelésének felhasználó oldali szempontjai.....	8
6.2	A feladat értékelésének technikai szempontjai.....	8
6.3	Projekt megvalósításának piaci jellegű értékelése .....	8
7	Projekt adatlap.....	9

## 2 Bevezetés

### 2.1 A feladat címe

Címtár Felhő

### 2.2 A feladat rövid ismertetése

Sok cégnél vannak olyan címtárak (telefonszámok, e-mail címek, stb.), amelyekre minden dolgozónak szüksége van napi munkavégzése során. Azonban ha egy ilyen címtárban például az egyik ügyfélnek megváltozik a telefonszáma vagy az e-mail címe, akkor azt minden egyes felhasználó készülékén át kell írni, hogy az adattartalom teljes szinten konzisztens maradjon. Erre a problémára létezik megoldás: a névjegyeket egyetlen forráspéldányként egy felhőbe mentik, és azt szinkronizálják a dolgozók a telefonjukra.

A cél egy olyan applikáció készítése, amely a felhőben tárol névjegyeket. Az előre beállított címtárat szinkronizálja a készülékre, és az ott folytatott módosításokat a felhőben is végrehajtja (hozzáadás, módosítás, törlés). A készüléken lehet hívást, sms írást, e-mail írást kezdeményezni.



## 3 Elvárások a feladattal kapcsolatban

### 3.1 Operációs rendszer, környezet

- Android 4.4.+ vagy Windows Phone 8/8.1.

### 3.2 Felhasználható programozási nyelv

- Java (Android fejlesztés esetén)
- C# (Windows Phone esetében)

### 3.3 Megoldás formátuma

- C#, Java forrásállomány
- Teljes projekt környezet
- Forráskód dokumentáció
- Technikai dokumentáció (odt és pdf formátumban)
- Erőforrás-terv és munkaidő nyilvántartás
- Felhasználói dokumentáció

### 3.4 Szoftverfejlesztés

A feladat egy olyan program készítése, mely egy felhőben tárolt címtárat képes szinkronizálni a készülék címtárával. A készülék tudjon kezdeményezni hívást, sms küldést, e-mail írást a telefon beépített eszközeivel (gyári hívó, sms, e-mail kliens).

A fejlesztés közben be kell tartani a meghatározott kódolási konvenciókat, melyeket a gyakorlatvezető szab meg.

Az elkészült megoldásnak maradéktalanul meg kell valósítania a 4. fejezetben megfogalmazott követelményeket. Amelyik követelmény nincs pontosan definiálva, ott a megvalósítás során a fejlesztő szabad kezet kap. Fontos, hogy a választott megoldás megfelelő színvonalú legyen mind felhasználói, mind fejlesztői szempontból.

### 3.5 Modulok

A projekt keretében történő megvalósítás egy lehetséges felbontási lehetősége az alábbi:

- Adatbázis tervezése, kivitelezése, felhő interfész megírása
- Felhasználóbarát front end (GUI) tervezése, back end kivitelezése
  - Applikáció felület, grafikai elemek
  - Menürendszer
  - Beállítások
- Projektvezetéssel kapcsolatos dokumentációk, nyilvántartások vezetése, feladatok összehangolása, felhasználói dokumentáció elkészítése, tesztelés.



Lehetőség szerint a fejlesztői dokumentációkat minden esetben a ténylegesen fejlesztést végző projektagok készítsék el. A felhasználói dokumentáció külön egységet képezhet, melyet azonban érdemes a teszteléssel összekapcsolni a megfelelő minőség biztosítása érdekében.

## 4 Szoftver specifikáció

Az alkalmazás csak álló nézetben jelenjen meg. A design-t úgy kell elkészíteni, hogy egy 4,7"-os kijelzőn kényelmesen jelenjen meg 5-7 névjegy.

Elsődlegesen egy saját programot kell készíteni, azonban plusz funkcióként egy olyan verziót is el lehet készíteni, amely a telefon saját névjegykezelőjébe beépülve tudja megjeleníteni az elemeket.

### 4.1 Megjelenés

- A program indításakor jelenjen meg a tárolt címlista. Amennyiben még nem történt meg a felhőhöz való csatlakozás, akkor jelenjen meg olyan felület, ahol megadhatók a kapcsolat adatai. Ha létrejött a kapcsolat a felhővel, és üres az adatbázis, akkor ezt jelezze a felhasználónak, és ajánlja fel a névjegy hozzáadása lehetőségét.
- Az alkalmazás kinézete legyen hasonló, mint a már megszokott telefonok címtára.
- A címtár tetején legyen egy gomb, amivel be lehet állítani, hogy mi alapján szeretnénk rendezni a listát.
- Új névjegy hozzáadása esetén egy új ablakban jelenjen meg egy űrlap, amelyen a következő adatokat kell megadni:
  - Vezetéknév
  - Keresztnév
  - Mobil szám
  - E-mail cím
  - Megjegyzés
- Egy névjegyre hosszan kattintva jelenjen meg 2 opció, az egyik a törlés, a másik a módosítás.

### 4.2 Funkciók

- Az alkalmazás indításakor (ha van címtár definiálva) a szinkronizáció fusson le.
- Lehessen több különböző címtárhoz csatlakozni, de egyszerre csak egy címlista adatai legyenek szinkronizálva a készülékre.
- A névjegyek rendezése lehessen:
  - Vezetéknév szerint növekvő és fordított sorrendben.
  - Keresztnév szerint növekvő és fordított sorrendben.
- A lista legyen alapértelmezetten „abc” sorrendben, vezetéknév alapján.
- A megjegyzés maximum 160 karakter lehet.
- Minden művelet esetén (névjegy hozzáadása, törlés, módosítás) a program automatikusan végezze el a módosításokat a felhőben is.
- Az applikáció alapvetően nem tud például Viber, vagy Facebook üzenet írást, hívást kezdeményezni, de plusz funkcionalitásként ez is megvalósítható.

## 5 Dokumentáció

### 5.1 Erőforrás-terv, munkaidő nyilvántartás

A specifikáció birtokában a projekt résztvevői készítsenek erőforrás-tervet. Ez tartalmazza a feladatban részt vevő projektagokat, akik legyenek hozzárendelve a tervezés során azonosított részfeladatokhoz. Minden részfeladat mellé kerüljön egy munkaidő ráfordítási becslés munkaóraban számolva. Ezt a tervet a tényleges fejlesztés előtt le kell adni. A feladat megoldása során az elvégzett munkáról készüljön nyilvántartás részfeladatonként és személyenként a tényleges munkaórák számának megjelölésével. A projekt végén a két dokumentum összehasonlításra, az eltérések elemzésre kerülnek.

### 5.2 Technikai dokumentáció

Az elkészült kódot, függvényeket és osztályokat megfelelő kommentekkel kell ellátni, továbbá el kell készíteni a teljes alkalmazás dokumentációját.

A dokumentációnak a feladat bonyolultságától függő hosszúságúnak kell lennie, maximális terjedelem nincs meghatározva.

A technikai dokumentáció szövegezésénél előírás, hogy a nem hozzáértő személyek számára is feldolgozható legyen, így az egyes fogalmak, rövidítések, idegen kifejezések magyarázatát a dokumentumnak tartalmaznia kell.

### 5.3 Forráskód dokumentáció

A fontosabb függvények és osztályok előtt szerepelnie kell megjegyzéseknek, melyeknek tartalmazniuk kell az azt követő metódus rövid szöveges – akár magyar nyelvű – leírását. A forráskód dokumentációt a munka során folyamatosan kell készíteni.

### 5.4 Felhasználói dokumentáció

Az alkalmazás használatának részletes bemutatása, képernyőképekkel, funkciók pontos leírásával.

## 6 A projekt értékelése

### 6.1 A feladat értékelésének felhasználó oldali szempontjai

A működő alkalmazás tesztelése alapján a legfontosabb jellemzők:

- A kiírást teljes egészében lefedő funkcionalitás
- Ergonomikus kialakítás
- Kényelmes használat
- Igényes felhasználói felület
- Stabil működés
- Igényes felhasználói dokumentáció

### 6.2 A feladat értékelésének technikai szempontjai

Informatikai szakmai szempontból a megoldás értékelésének alapja:

- Kódkép, a kód tisztasága, a komment minősége
- Kódolási konvenciók betartása
- Fejlesztői dokumentáció színvonala
- Dokumentált tesztelés
- Erőforrás felhasználásának pontos nyilvántartása

### 6.3 Projekt megvalósításának piaci jellegű értékelése

A projekt lezárását követően összehasonlításra kerül a kezdeti erőforrás-terv, valamint a megvalósítás során dokumentált munka. Ezen dokumentumok elemzéséből levezetésre kerülnek azok a tipikus problémák, melyek a piaci környezetben jellemzően megjelennek. Végigtekintjük ezen problémák okait, következményeit, lehetséges elkerülésüknek vagy hatásuk mérséklésének módjait. A jellemző hibák ebből a megközelítésből:

- Határidő csúszása
- Nem megfelelő minőség
- Hiányos, vagy elmaradó tesztelés
- Használhatatlan, pontatlan dokumentáció
- Pontatlan erőforrás-bebecslés
- Aránytalanul magas önköltség
- Az elkészült termék továbbfejlesztésének, karbantartásának nehézségei

A fentiek értékelésén túl fejlesztői szemszögből elemezzük a megvalósítás tapasztalatait, a lehetséges továbbfejlesztés, átalakítás, támogatás kérdéseit és piaci lehetőségeit.



## 7 Projekt adatlap

Projekt neve: Címtár felhő

Feladat rövid ismertetése: Címtár adatbázis szinkronizálása felhővel, telefonhívás, sms írás és e-mail küldés kezdeményezése.

Felhasznált programozási nyelv: C# vagy Java

Specifikációt összeállította: Szabó Gábor Ferenc, Apáti László