

Entitások

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
2.3	Entitások használatának leírása.....	4
2.4	Fogalmak.....	5
3	Elvárások a feladattal kapcsolatban.....	6
3.1	Operációs rendszer, környezet.....	6
3.2	Felhasználható programozási nyelv	6
3.3	Megoldás formátuma.....	6
3.4	Szoftverfejlesztés	6
3.5	Modulok.....	6
4	Szoftverspecifikáció.....	8
4.1	Megjelenés	8
4.2	Funkciók.....	8
5	Dokumentáció	9
5.1	Erőforrásterv, munkaidő-nyilvántartás	9
5.2	Üzemeltetői dokumentáció	9
5.3	Forráskód dokumentáció.....	9
5.4	Felhasználói dokumentáció	9
6	A projekt értékelése.....	10
6.1	A feladat értékelésének felhasználó oldali szempontjai.....	10
6.2	A feladat értékelésének technikai szempontjai.....	10
6.3	Projekt megvalósításának piaci jellegű értékelése	10
7	Projekt adatlap.....	11

2 Bevezetés

2.1 A feladat címe

Entitások

2.2 A feladat rövid ismertetése

Sokszor egyszerűsítene az életet, ha lenne egy olyan alkalmazás, mellyel a ritkán használt, de fontos adatainkat tudnánk eltárolni. Előfordulhat, hogy hivatalos ügyek elintézésakor olyan adatot kell megadni, amit fejből nem tud az ember. Egy elveszett bankkártya letiltásakor vagy egy hivatalos dokumentum kitöltésekor például az adószámunk azonnali előkereshetőségének lehetősége hasznos segítség lehet. Ehhez kellene egy olyan szoftver, melyben a különféle eltárolandó adathalmazokhoz (pl.: személyes adatok, bankkártyák, website bejelentkezési adatok, különféle belépőkódok stb.) saját entitások létrehozásával lehet különböző típusú adatokat eltárolni. Az alkalmazás a tároláson túl a rendszerezésben is tudna segíteni.



2.3 Entitások használatának leírása

Példa: a felhasználónak van több bankkártyája, melyek külön banknál vezetett bankszámlákhoz kapcsolódnak. Az alkalmazásban – más jellegű adatokkal együtt – a bankkártyák adatait szeretné eltárolni.

Megoldás: Első lépésként létre kell hozni egy entitás sémát bankkártya adatokhoz. Itt meg lehet adni, hogy ez a séma milyen adatmezőket tartalmazzon, és mi legyen az entitás neve, ami alapján az ügyfél majd azonosítani fogja, valahogy így:

„Bankkártya” entitás séma:

- „Bankszámla száma”, String
- „Bankkártya száma”, String
- „Bankkártya készítés dátuma”, Dátum
- „Bankkártya lejáratára”, Dátum
- „Pin kód”, Int
- „Megjegyzés”, String

Új példány létrehozása: Valós adatok tárolásához kiválasztva a megfelelő entitás típust (jelen esetben a „Bankkártya” entitást) a program létrehoz egy új példányt, melyet fel lehet tölteni a tárolandó adatokkal. Az alábbi példa szerint a kártya a „Nagytestvér” banknál használatos. (A „Nagytestvér” az entitás példány neve.)

A „Nagytestvér” bankkártya példány elemei a következők:

- Bankszámla száma: 11223344-55667788-99887766
- Bankkártya száma: 1111-2222-3333-4444-5555
- Bankkártya készítés dátuma: 2009/12/03
- Bankkártya lejáratára: 2019/12/03
- Pin kód: 1234
- Megjegyzés: Megtakarításra használom.

Példánkban van egy másik bankkártya, ami az „Országos Bank”-nál érvényes.

Az „Országos Bank” elemei a következők:

- Bankszámla száma: 99887766-55443322-11223344
- Bankkártya száma: 1234-5678-9876-5432-1100
- Bankkártya készítés dátuma: 2006/09/24
- Bankkártya lejáratára: 2016/09/24
- Pin kód: 1598
- Megjegyzés: Ide jön a fizetés minden hónap 10.-éig.

Amikor bankkártya adatokra van szükség, a felhasználó a bankkártya entitások közül a megfelelő példányt kiválasztva minden eltárolt információt megtekinthet. Természetesen

bankkártyákon kívül van még sok más hasonló adatkör, amit el lehetne tárolni. Például autók adatai, személyes adatok, melyekhez már más felépítésű entitások szükségesek.

2.4 Fogalmak

Entitás: Az entitások elemekből épülnek fel.

Entitás példány: Egy entitásból lehet készíteni példányt, ez az entitás példány.

Elem: Egy elemnek két tulajdonsága van: egy neve és egy típusa. Például: „Bankszámla száma”, String típusú.

Típus: Egy elemnek a típusa meghatározza, hogy egy elem milyen formában tud adatot tárolni. Pl.: egész szám, szöveg, dátum.

3 Elvárások a feladattal kapcsolatban

3.1 Operációs rendszer, környezet

- Minimum Android 11.0 (min. API level 30)

3.2 Felhasználható programozási nyelv

- Nincs megkötés
- Ajánlott nyelvek, keretrendszerek: Java, Kotlin, Dart; Flutter

3.3 Megoldás formátuma

- Forráskód állományok
- Teljes projekt környezet
- Forráskód dokumentáció
- Üzemeltetői dokumentáció (odt/docx és pdf formátumban)
- Felhasználói dokumentáció
- Erőforrásterv és munkaidő-nyilvántartás

3.4 Szoftverfejlesztés

A feladat egy olyan mobilalkalmazás fejlesztése, melyben tetszőleges számú és felépítésű entitást lehet létrehozni.

A fejlesztés közben be kell tartani a Clean Code alapszabályait.

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, DB interfész megírása
- Felhasználóbarát front end (GUI) tervezése
 - Applikáció felület, grafikai elemek
 - Menürendszer
 - Beállítások
- Back end osztályok írása, melyek a program logikáját valósítják meg
 - Programmotor megvalósítása
- Projektvezetéssel kapcsolatos dokumentáció, 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, érdemes a teszteléssel összekapcsolni a megfelelő minőség biztosítása érdekében.

4 Szoftverspecifikáció

Az alkalmazás portrait (álló) és landscape (fekvő) nézetben is jelenjen meg. A felületet úgy kell elkészíteni, hogy egy szokványos kijelzőméretű készüléken kényelmesen lehessen kezelni. Az alkalmazásnak megfelelő reszponzivitást kell nyújtania.

4.1 Megjelenés

- A program indításakor jelenjenek meg az entitás példányok. Amennyiben még nincs entitás példány, akkor azt jelezze a felhasználónak.
- Különálló listában legyenek láthatók a már definiált entitások (sémák). Innen kényelmesen lehessen új példányt létrehozni.
- Új entitás példány létrehozásakor minden, az entításban definiált elemet lehessen kitölteni.
- Az entitás példányok entitás csoportokba rendezve jelenjenek meg a felhasználói felületen, nevük szerint növekvő sorrendben.

4.2 Funkciók

- Lehessen új entitást és entitás példányt létrehozni.
- Egy példány adatainak megtekintése során módosítást csak külön utasításra lehessen végezni. Tehát alapból az adatok csak olvashatóak legyenek, a módosításhoz külön interakció szükséges.
- Egy entitásból több példány is készülhet.
- Minden entitásnak saját elemei vannak.
- Másik entitásból nem lehet elemet átvenni egy adott entitásba.
- Rendezni lehessen „abc” szerint növekvő és csökkenő sorrendbe is az entitásokat.
- Módosítani lehessen a példány adatait.
- Entitásokat is lehessen módosítani, de ha már van belőle példány létrehozva, akkor maximum új elemet lehessen felvenni hozzá. (Ebben az esetben törölni és módosítani nem lehet az entitás meglévő elemeit.)
- Olyan entitást, amelyből még nincs példány, lehessen törölni, valamint korlátozás nélkül módosítani.
- Minden művelet végén felhasználói megerősítés (ez lehet pl. egy „mentés” gomb megnyomása, esetleg adatmentés megerősítésének külön kérése) esetén a program rögzítse a módosításokat az adatbázisban.
- Adatok változtatásával záruló műveleteknél (új elem létrehozása, módosítása, törlése) legyen lehetőség a művelet elvetésére.

5 Dokumentáció

5.1 Erőforrásterv, munkaidő-nyilvántartás

A specifikáció birtokában a projekt résztvevői készítsenek erőforrástervet. 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 Üzemeltetői dokumentáció

Az üzemeltetői dokumentáció célja, hogy a rendszer üzemeltetőinek támogatást adjon a termék üzemeltetésének elsajátításához. Tartalmazza többek között a rendszer architekturális felépítését (alkalmazás stack elemei és azok közti kapcsolat leírása), az alkalmazás fordításához, fejlesztéséhez és futtatásához szükséges követelményeket, technológiákat, továbbá a konfigurációs állományok leírását (ha vannak).

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

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 az alábbiak a legfontosabb jellemzők:

- Kiírást teljes egészében lefedő funkcionalitás
- 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, kommentelés minősége
- Kódolási konvenciók betartása (Clean Code)
- Dokumentációk színvonala
- Dokumentált tesztelés
- Erőforrás felhasználásának pontos nyilvántartása

6.3 Projekt megvalósításának piaci jellegű értékelése

A projekt lezárultával összehasonlításra kerül a kezdeti erőforrásterv, valamint a megvalósítás során dokumentált munka. Ezen dokumentumok elemzéséből levezetésre kerülnek azok a 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 becslés
- 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: Entitások

Feladat rövid ismertetése: Saját entitásokat létrehozva adatok tárolása és rendszerezése.

Specifikációt összeállította: Szabó Gábor Ferenc, Apáti László, Varga Bence