

Tankoláskönyvelő

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 | Szoftverspecifikáció..... | 6 |
| 4.1 | Megjelenés..... | 6 |
| 4.2 | Funkciók..... | 6 |
| 5 | Dokumentáció | 8 |
| 5.1 | Erőforrásterv, munkaidő-nyilvántartás..... | 8 |
| 5.2 | Üzemeltetői dokumentáció..... | 8 |
| 5.3 | Forráskód dokumentáció..... | 8 |
| 5.4 | Felhasználói dokumentáció | 8 |
| 6 | A projekt értékelése..... | 9 |
| 6.1 | A feladat értékelésének felhasználó oldali szempontjai..... | 9 |
| 6.2 | A feladat értékelésének technikai szempontjai..... | 9 |
| 6.3 | Projekt megvalósításának piaci jellegű értékelése | 9 |
| 7 | Projekt adatlap..... | 10 |

2 Bevezetés

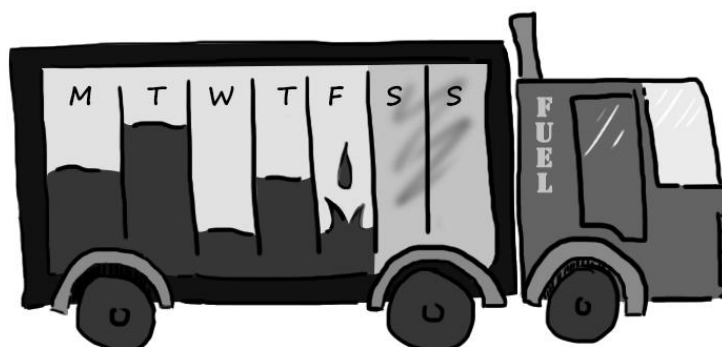
2.1 A feladat címe

Tankoláskönyvelő

2.2 A feladat rövid ismertetése

Az autótulajdonosok többsége szereti kiszámolni az autó aktuális fogyasztását. Vannak, akik ezért feljegyzik minden tankoláskor a tankolt üzemanyag mennyiségét és az autó aktuális kilométeróra-állását.

A kockás füzetek világát már rég elhagytuk, ezért merült fel az igény egy olyan mobiltelefonos alkalmazásra, amiben a tankolási adatainkat fel tudjuk jegyezni.



Ez az alkalmazás nemcsak az adatok tárolására lenne alkalmas, hanem ezen adatokból statisztikákat készítené. Ezzel a tulajdonos jobban megismeri a vezetési stílusához tartozó üzemanyag-fogyasztást.

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 készítése, mely az autó tulajdonosának, használójának segít nyilvántartani, hogy mikor, mennyi és milyen típusú üzemanyagot tankolt, valamint segít megismerni, hogy adott vezetési stílus esetén milyen fogyasztást produkál az autó. Ezen felül egyéb statisztikai adatokkal szolgálja a gépjármű tulajdonosát, használóját.

A fejlesztés közben be kell tartani a Clean Code alapszabályait.

Az elkészült megoldásnak maradéktalanul meg kell valósítania az 4-es fejezetben megfogalmazott követelményeket. Amelyik követelmény nincs pontosan definiálva, abban a megvalósítás során a fejlesztő szabad kezet kap. Fontos viszont, 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 tervezés, kivitelezés, DB interfész megírása
- Felhasználóbarát front end (GUI) tervezése, kivitelezése
 - Programfelület, grafikai elemek
 - Menürendszer
 - Statisztikák
 - Beállítások

- Back end osztályok írása, melyek a program logikáját valósítják meg
 - Programmotor megvalósítása
 - Statisztikák összegyűjtése
- 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 csak portrait módon jelenjen meg. A felületet úgy kell elkészíteni, hogy kényelmesen lehessen navigálni egy szokványos kijelzőméretű készüléken, megfelelő rezponzivitást nyújtva.

4.1 Megjelenés

- A program indításakor jelenjenek meg a következő adatok:
 - Legutolsó tankolás óta eltelt idő
 - A legutolsó tankolás adatai
 - Átlagok a tankolási adatokból (pl.: átlagos fogyasztás, egy tankolással átlagosan megtett kilométer)
 - Az összes megtett kilométer
- A megjelenés táblázatos legyen.
- A statisztikai adatok valamilyen grafikonos formában is jelenjenek meg (pl.: vonaldiagram, kördiagram stb.).

4.2 Funkciók

- Lehessen felvinni tankolási adatokat, amik alapvetően a következőket tartalmazzák: (ezen adatok kitöltése kötelező)
 - Aktuális kilométeróra-állás
 - Tankolt üzemanyag típusa (benzin 95 vagy 98, diesel stb.)
 - Üzemanyag ára (Ft/l)
 - Tankolt üzemanyag mennyisége literben
 - Tankolás dátuma
- Az alapprogramban az előzőleg leírt adatokat kérje be, de plusz funkcionalitásként a fejlesztő általa fontosnak vélt információt is bekérhet, mint például a benzinkút típusa.
- A program tartalmazza a következő alapstatisztikákat:
 - Az autó átlagfogyasztása (l/100 km)
 - Átlagban hetente, vagy havonta a tankolások száma
 - Hónapra, vagy hétre bontva a tankolások száma
 - Az utolsó, és az azt megelőző tankolás között megtett út hossza kilométerben megadva
- Az előzőekben felsorolt statisztikákon kívül plusz funkcionalitásként a fejlesztő, számára fontosnak vélt információt tartalmazó statisztikát is készíthet (pl.: Hónapokra lebontva mennyi volt az összes üzemanyagköltség).
- Amennyiben még nincs semmilyen adat, akkor ezt jelezze a felhasználónak és adjon lehetőséget, hogy egyből felvihesse az első tankolás adatait.

- Visszamenőleg ne lehessen módosítani tankolási adatot.
- El lehessen menteni (exportálni) a teljes adatbázist.
- Egy régebbi adatbázist be tudjon tölteni a program.
- A jelenlegi adatbázist lehessen törölni.
- Egyszerre csak egy adatbázist tudjon kezelni a program, ami azt jelenti, hogy ha egy másik adatbázist tölt be, akkor az előzőt teljesen töröli. A törlésre a felhasználót figyelmeztetni kell. Természetesen ebben az esetben lehetőséget kell adni a felhasználónak, hogy a régi adatbázist elmentse (exportálja).

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: Tankoláskönyvelő

Feladat rövid ismertetése: Tankolások könyvelése, s ebből statisztikák készítése.

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