

TollÚtdíj

Algoritmus



Készítette:

Bobán Petrik József,

Nánási Richárd,

Simicz Zalán Zoltán

Tartalomjegyzék

1.	Login algoritmus.....	4
1.1	Célja:	4
1.2	Folyamatábra:.....	4
2.	Role-based főmenü / modulválasztás	5
2.1	Célja:	5
2.2	Folyamatábra:.....	5
3.	Logout (Kilépés / Kijelentkezés)	6
3.1	Célja:	6
3.2	Folyamatábra:.....	6
4.	App indítás + autologin.....	7
4.1	Célja:	7
4.2	Folyamatábra:.....	7
5.	Cégek CRUD	8
5.1	Célja:	8
5.2	Folyamatábra:.....	8
6.	Sofőrök CRUD.....	9
6.1	Célja:	9
6.2	Folyamatábra:.....	9
7.	Jogosítványok CRUD	10
7.1	Célja	10
7.2	Folyamatábra:.....	10
8.	Tracker eszközök CRUD	11
8.1	Célja:	11
8.2	Folyamatábra:.....	11
9.	Járművek CRUD	12
9.1	Célja:	12
9.2	Folyamatábra:.....	12
10.	RFID kártya + hozzárendelés.....	13
10.1	Célja:	13
10.2	Folyamatábra:.....	13
11.	Menet indítás	14

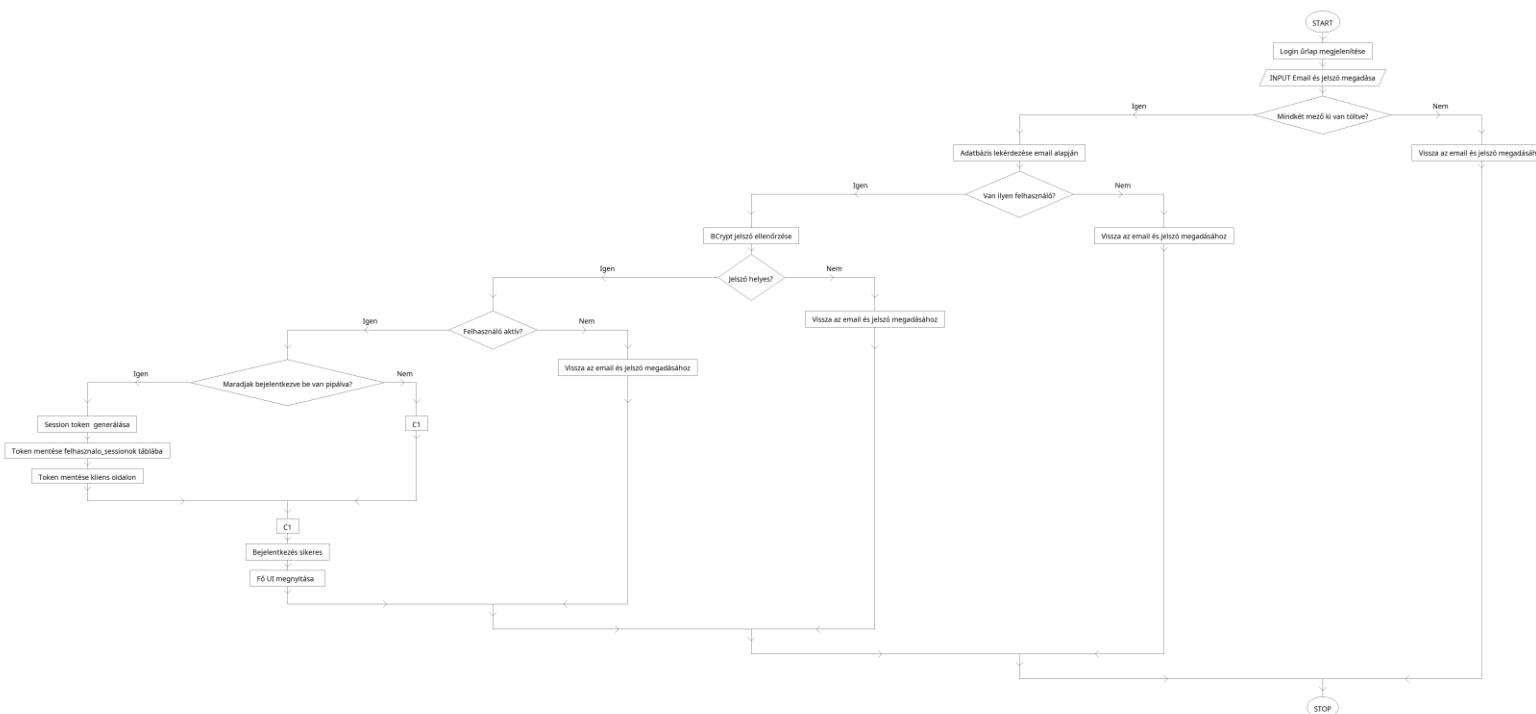
11.1 Célja:	14
11.2 Folyamatábra:.....	14
12. Menet lezárás	15
12.1 Célja:	15
12.2 Folyamatábra:.....	15
13. RFID azonosítás belső logika	16
13.1 Célja:	16
13.2 Folyamatábra:.....	16
14. GPS pozíció fogadás	17
14.1 Célja:	17
14.2 Folyamatábra:.....	17
15. Statisztikák	18
15.1 Célja:	18
15.2 Folyamatábra:.....	18
16. Útvonal tervezés.....	19
16.1 Célja:	19
16.2 Folyamatábra:.....	19
17. Útdíj kalkuláció	20
17.1 Célja:	20
17.2 Folyamatábra:.....	20
18. Jogosultság lekérdezés	21
18.1 Célja:	21
18.2 Folyamatábra:.....	21

1. Login algoritmus

1.1 Célja:

A login algoritmus célja, hogy a felhasználó által megadott email–jelszó párost biztonságosan ellenőrizze az adatbázisban tárolt, hash-elt jelszóval szemben. Csak aktív felhasználót enged be, szükség esetén létrehoz egy “maradjak bejelentkezve” session tokent, majd a jogosultsági szintjének megfelelő főfelületre irányítja. Hibás adat vagy inaktív fiók esetén hibaüzenetet ad, és visszatér a bejelentkezési űrlaphoz.

1.2 Folyamatábra:

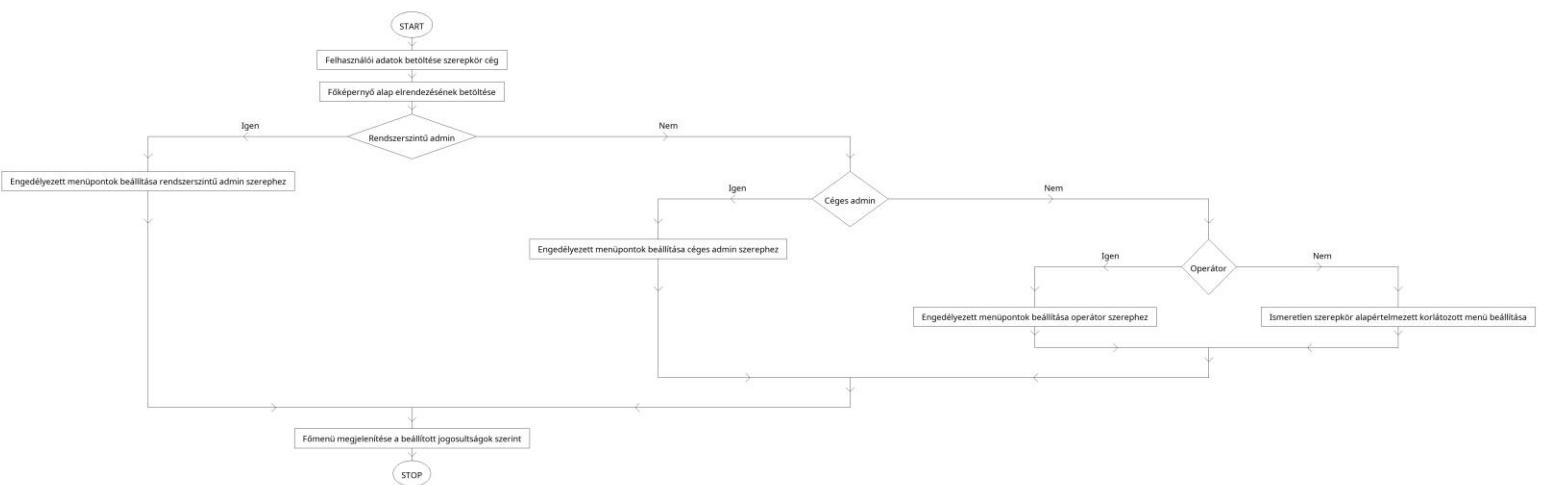


2. Role-based főmenü / modulválasztás

2.1 Célja:

Ennek az algoritmusnak a célja, hogy a bejelentkezett felhasználó szerepköre alapján eldönthse, melyik menüpontok és modulok legyenek elérhetők a főképernyőn. Így egy rendszer szintű adminisztrátor minden modult lát és kezel, a céges adminisztrátor csak a saját cégehez kapcsolódó admin funkciókat, míg egy operátor csak a számára engedélyezett, szűkebb funkcionálitást éri el. A folyamatábra ennek a döntési logikának a teljes, pontos “terképe” lesz.

2.2 Folyamatábra:

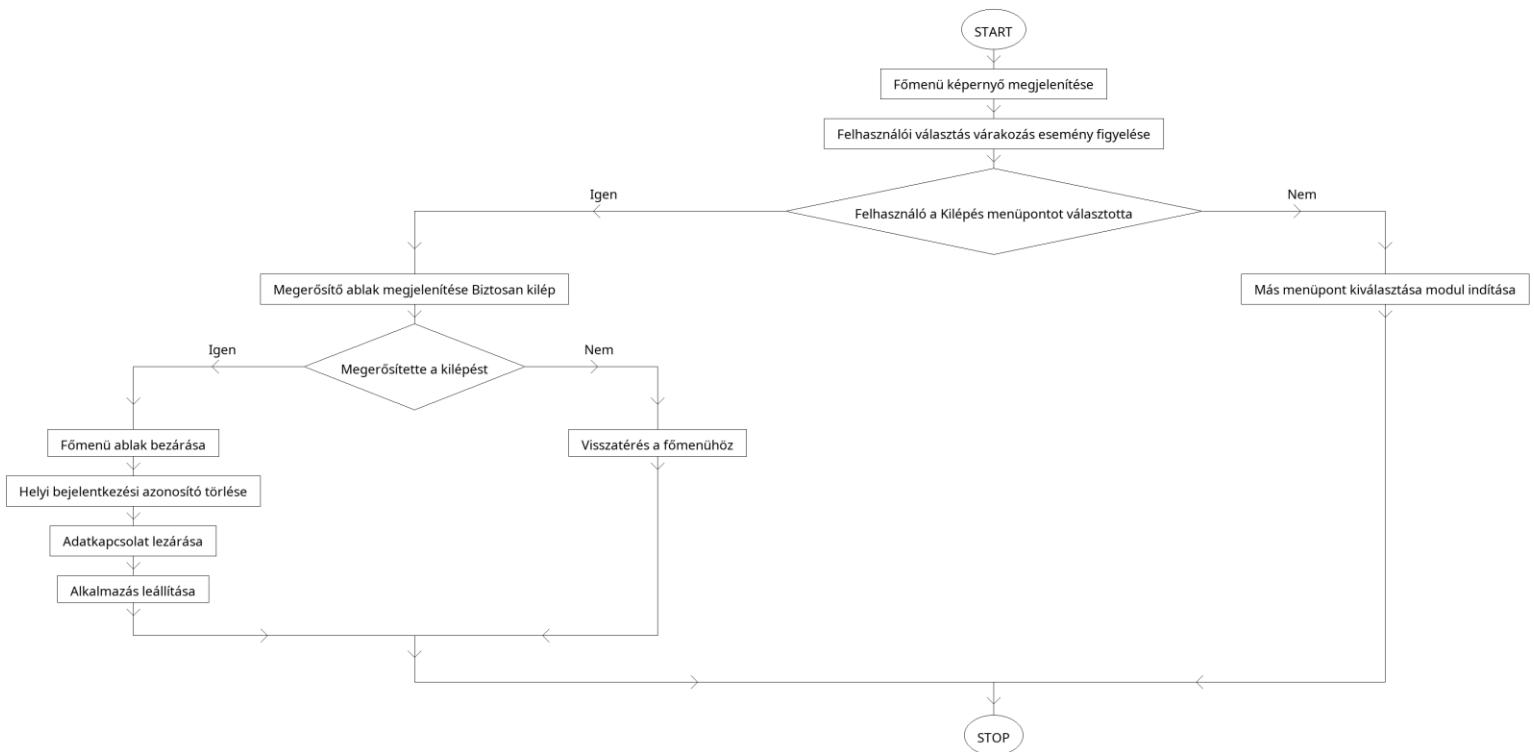


3. Logout (Kilépés / Kijelentkezés)

3.1 Célja:

A logout algoritmus célja, hogy a főképernyőről biztonságosan kiléptesse a felhasználót a rendszerből. Ez a folyamat gondoskodik róla, hogy a főmenü bezáródjon, a háttérben futó ciklus leálljon, és az alkalmazás rendezett módon leálljon (vagy visszatérjen a bejelentkezési felülethez – a logika ettől függetlenül ugyanúgy épül fel). A folyamatábra ennek a “Kilépés” útvonalnak az egyértelmű, lépésről lépévre történő ábrázolása.

3.2 Folyamatábra:

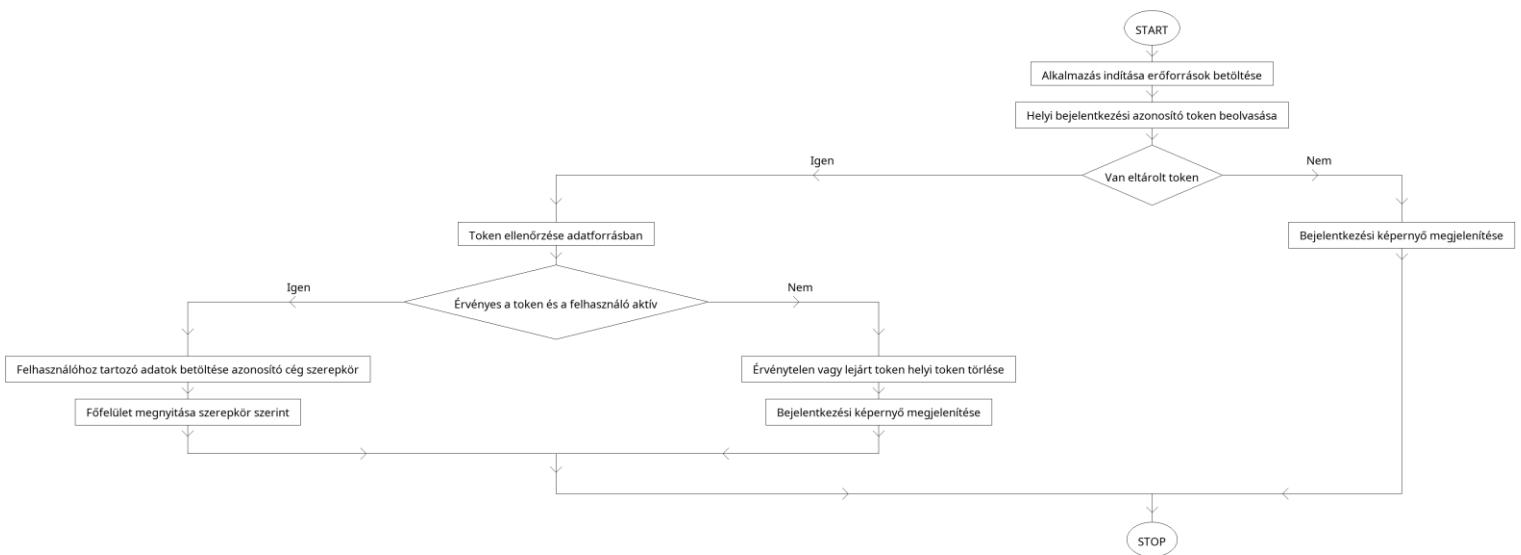


4. App indítás + autologin

4.1 Célja:

Ennek az algoritmusnak a célja, hogy az alkalmazás indulásakor automatikusan ellenőrizze, van-e korábban eltárolt, még érvényes bejelentkezési azonosító (session token), és ha igen, a felhasználót jelszó megadása nélkül beléptesse a megfelelő főfelületre. Ha nincs ilyen azonosító, vagy az érvénytelen, akkor az algoritmus biztonságosan visszairányít a bejelentkezési képernyőre, ahol a normál login folyamat indul.

4.2 Folyamatábra:

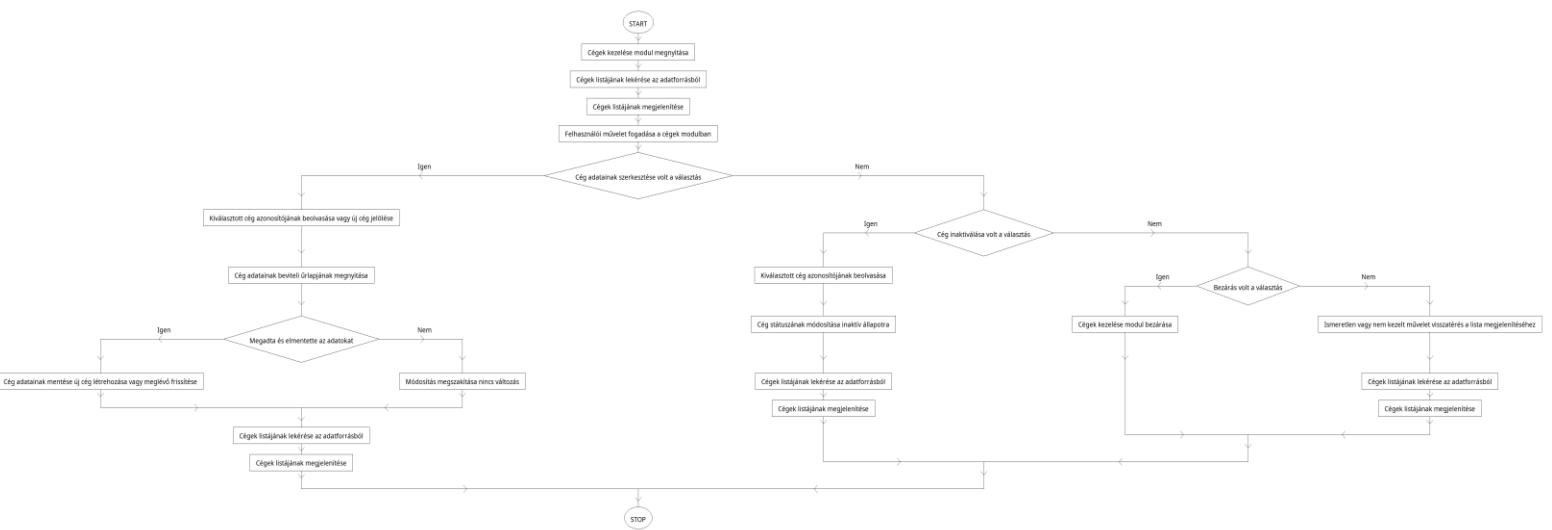


5. Cégek CRUD

5.1 Célja:

Ennek az algoritmusnak a célja, hogy egy külön „Cégek kezelése” modulban a felhasználó a cégek listáját megtekinthesse, új cégeket vehessen fel, meglévő cégek adatait módosíthassa, illetve szükség esetén a cégeket inaktív státuszba tegye. A folyamat úgy van felépítve, hogy minden módosítás után frissüljön a lista, a felhasználó pedig a modul bezárásáig ismételten végezhessen műveleteket.

5.2 Folyamatábra:

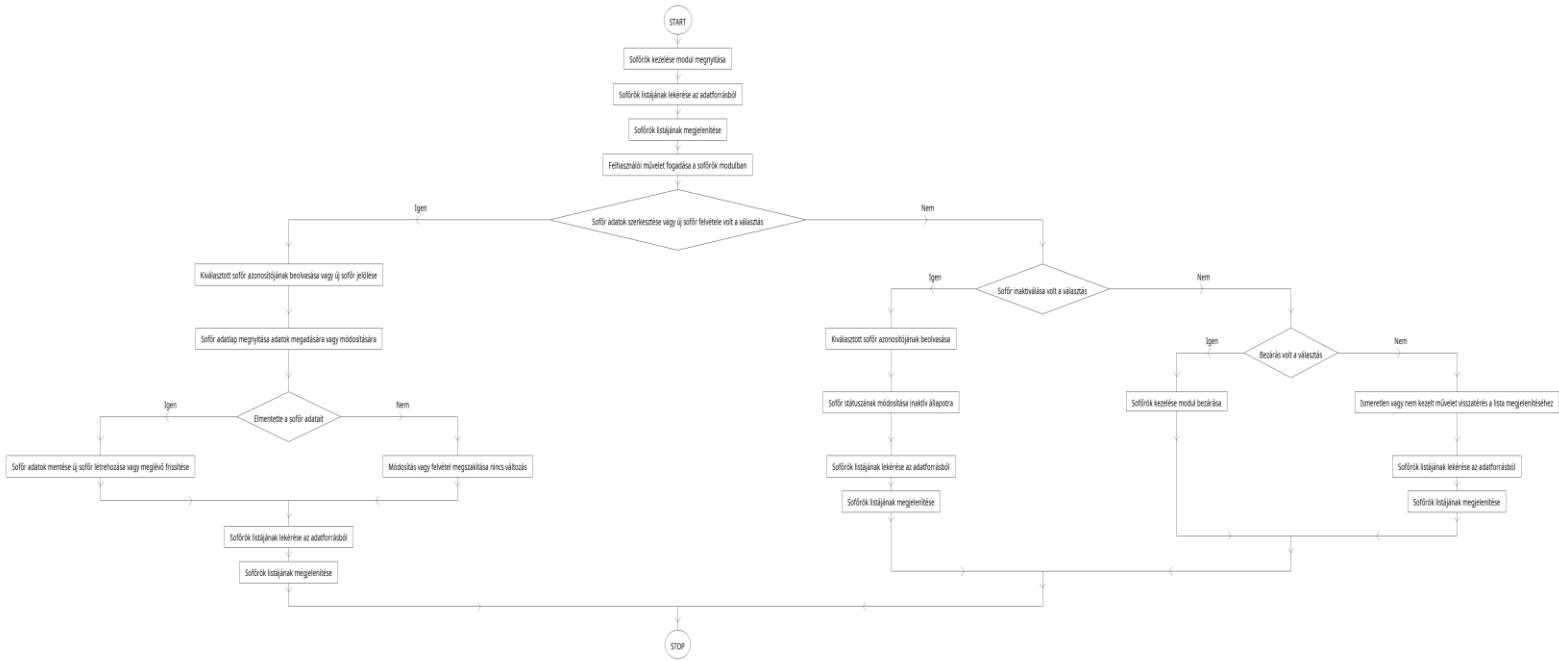


6. Sofőrök CRUD

6.1 Célja:

A sofőrok modul célja, hogy egy adott cégezhez tartozó sofőrok adatait biztonságosan és áttekinthetően lehessen kezelni. A folyamat lehetővé teszi a sofőrok listázását, új sofőr felvételét, meglévő sofőr adatainak módosítását, illetve szükség esetén a sofőrok inaktiválását (helyettesítve a törlést). A modul ciklikusan működik: minden művelet után friss listát mutat, egészen addig, amíg a felhasználó be nem zárja a modult.

6.2 Folyamatábra:

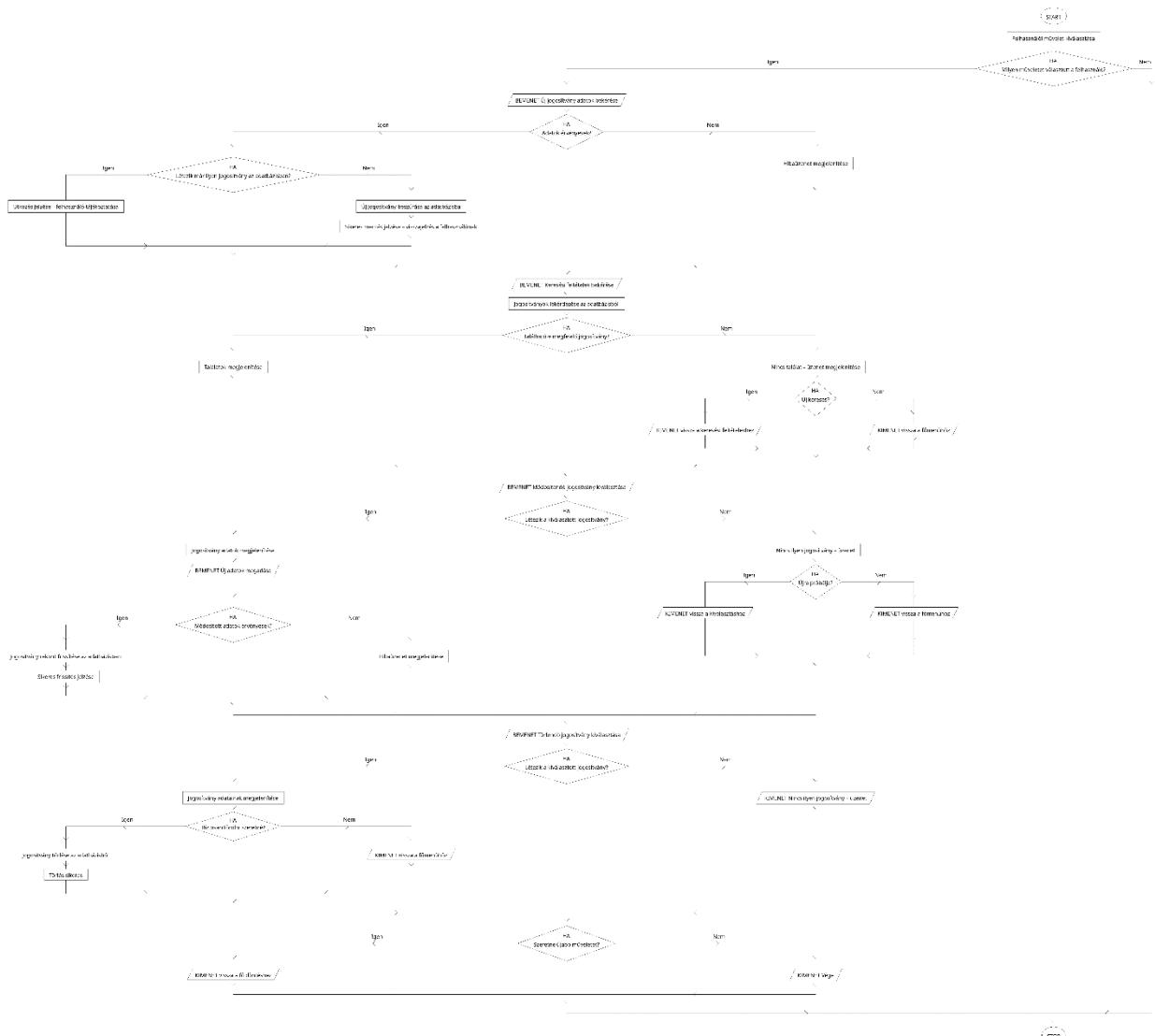


7. Jogosítványok CRUD

7.1 Célja

Az algoritmus célja, hogy a jogosítványokra vonatkozó adatokat (személy, kategória, érvényesség, stb.) egy adatbázisban biztonságosan kezelje: új rekordot hozzon létre, meglévőt megjelenítsen, módosítson vagy töröljön. A folyamatábra áttekinthetően mutatja be, hogy a felhasználó minden lépésekben tud CRUD-műveleteket végezni, és milyen ellenőrzések (pl. adatellenőrzés, jogosítvány létezése) történnek közben.

7.2 Folyamatábra:

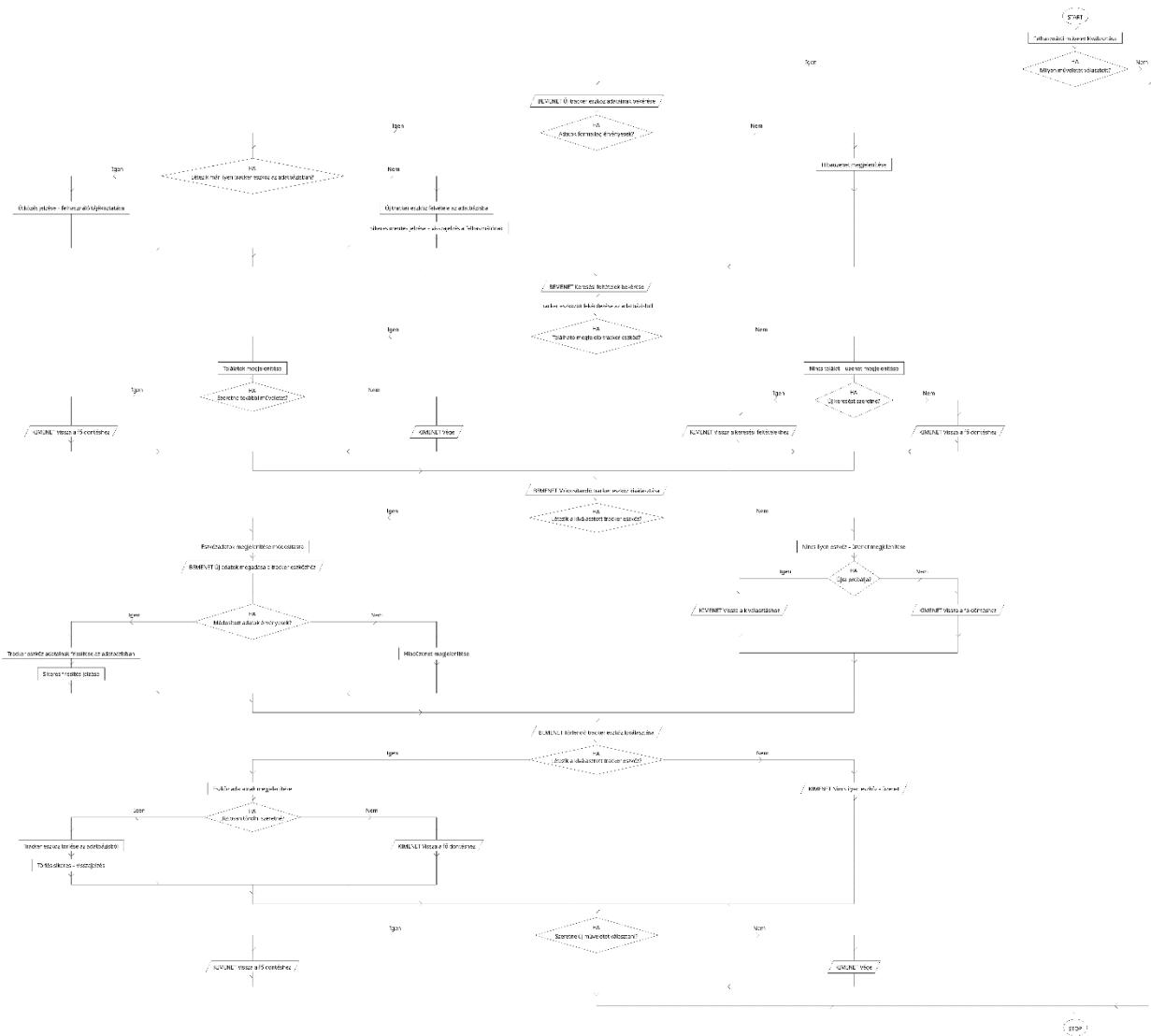


8. Tracker eszközök CRUD

8.1 Célja:

Az algoritmus célja, hogy a tracker eszközök adatait (név, azonosító, típus, státusz stb.) egy központi adatbázisban egységesen kezelje: új eszköz felvétele, meglévő adatainak megtekintése, módosítása és törlése. A folyamatábra szemléletesen mutatja be, hogyan lép végig a felhasználó ezeken a CRUD-műveleteken, és hol történnek ellenőrzések (pl. létezik-e már az eszköz, helyesek-e az adatok).

8.2 Folyamatábra:

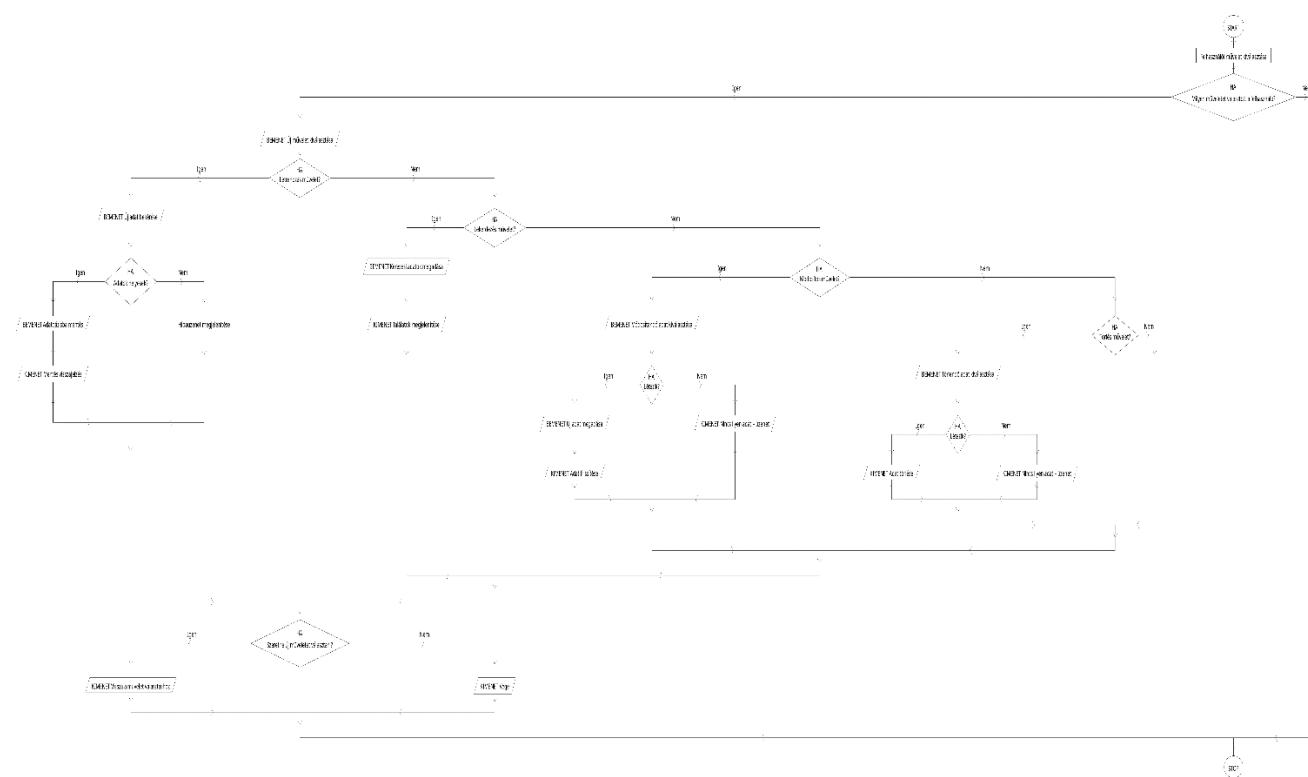


9. Járművek CRUD

9.1 Célja:

Az algoritmus célja, hogy a járművek adatait (pl. rendszám, típus, évjárat, tulajdonos, státusz) egy adatbázisban egységesen kezelje: új jármű felvétele, meglévő adatainak megtekintése, módosítása és törlése. A folyamatábra vizuálisan, lépésről lépésre mutatja be, hogyan választ a felhasználó CRUD-műveletek között, milyen ellenőrzések történnek (érvényes adatok, létező jármű, stb.), és hogyan tér vissza a fő menübe.

9.2 Folyamatábra:

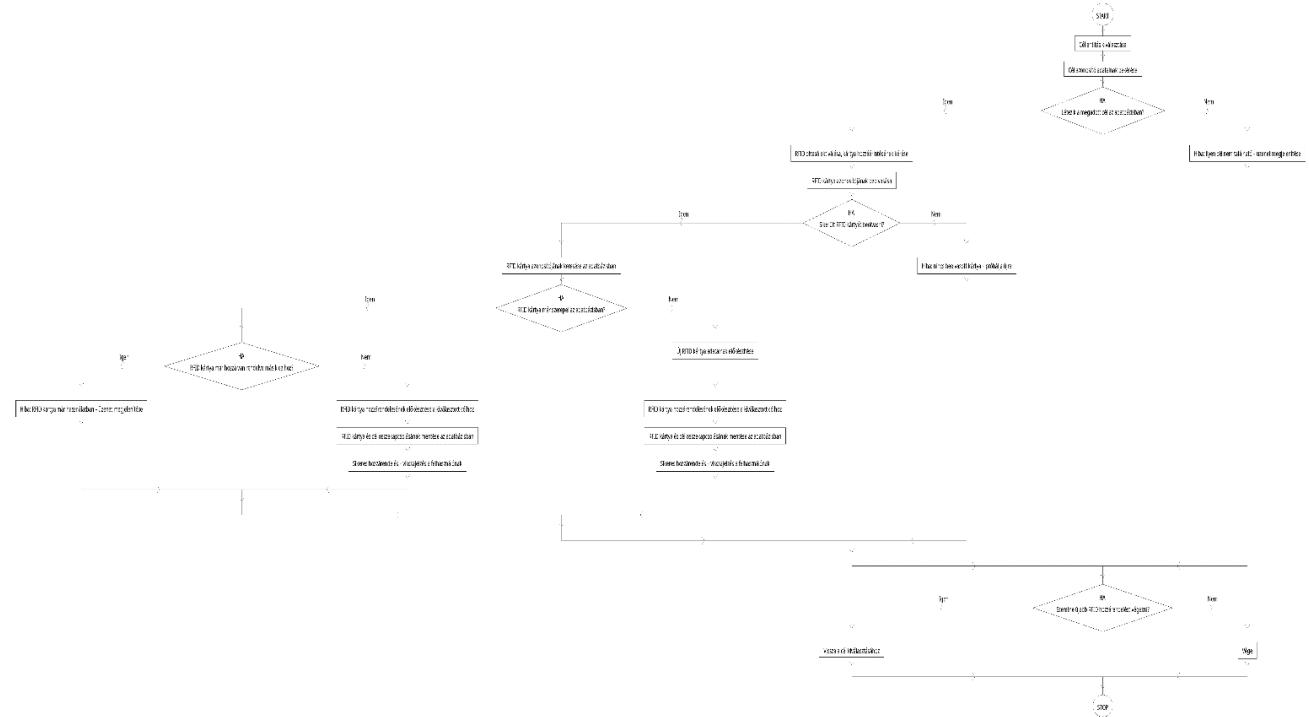


10. RFID kártya + hozzárendelés

10.1 Célja:

Az algoritmus célja, hogy egy RFID kártyát biztonságosan hozzárendeljünk egy kiválasztott személyhez vagy eszközözhöz (pl. dolgozó, jármű, tracker). A folyamatábra lépésről lépésre mutatja meg, hogyan választja ki a felhasználó a célt, hogyan olvassa be a rendszer az RFID kártyát, hogyan ellenörzi, hogy a kártya már használatban van-e, és hogyan történik a mentés.

10.2 Folyamatábra:

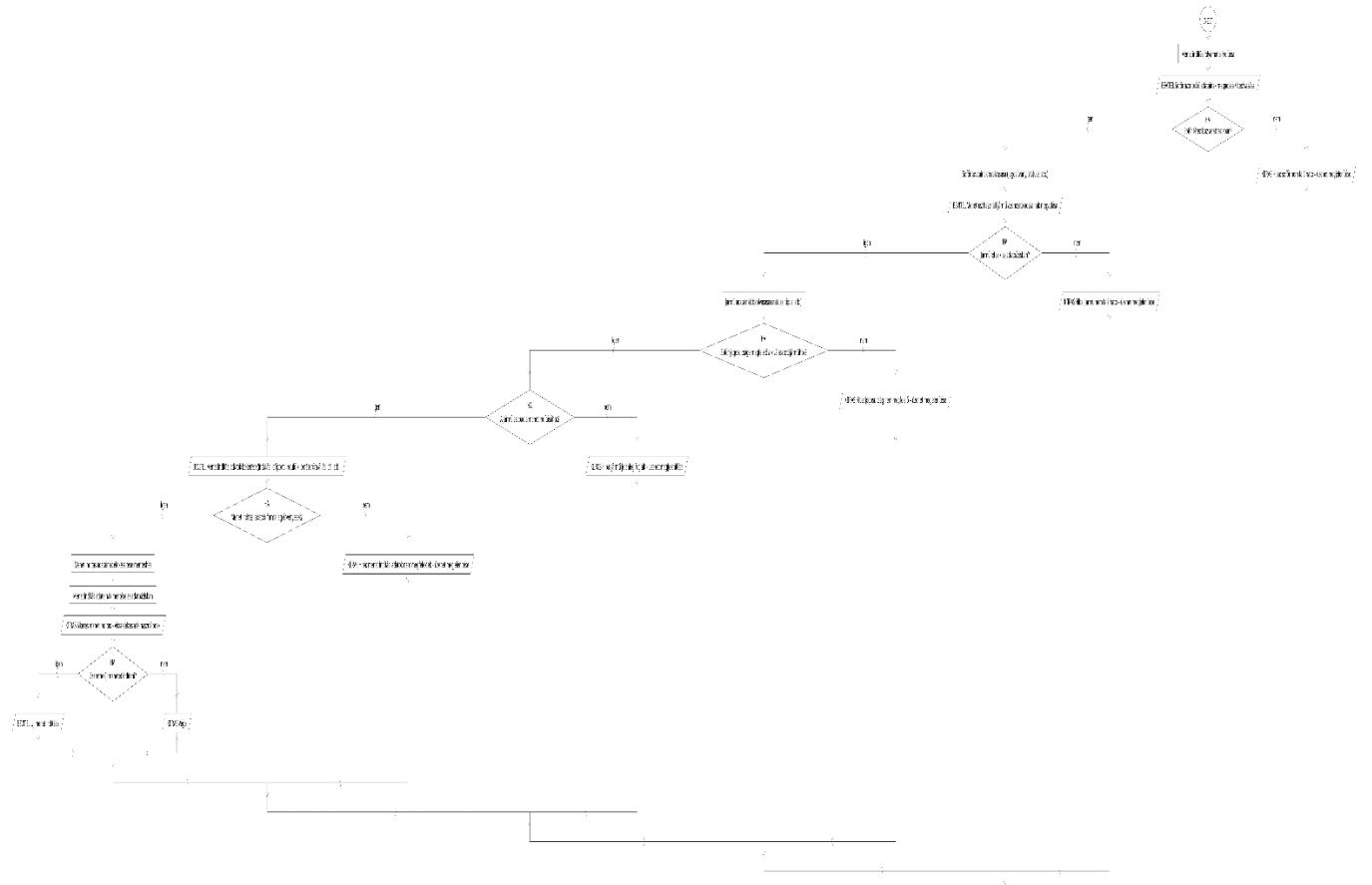


11. Menet indítás

11.1 Célja:

Az algoritmus célja, hogy a menet indítását (pl. jármű elindulása) szabályozottan, ellenőrzötten hajtsa végre: azonosítja a sofőrt, kiválasztja a járművet, ellenőrzi a jogosultságokat (érvényes jogosítvány, hozzárendelt RFID kártya stb.), majd eltárolja a menet kezdetének adatait. A folyamatábra lépésről lépésre bemutatja, hogyan halad végig a felhasználó ezen a folyamatton, hol történnek ellenőrzések, és mikor indulhat el ténylegesen a menet.

11.2 Folyamatábra:

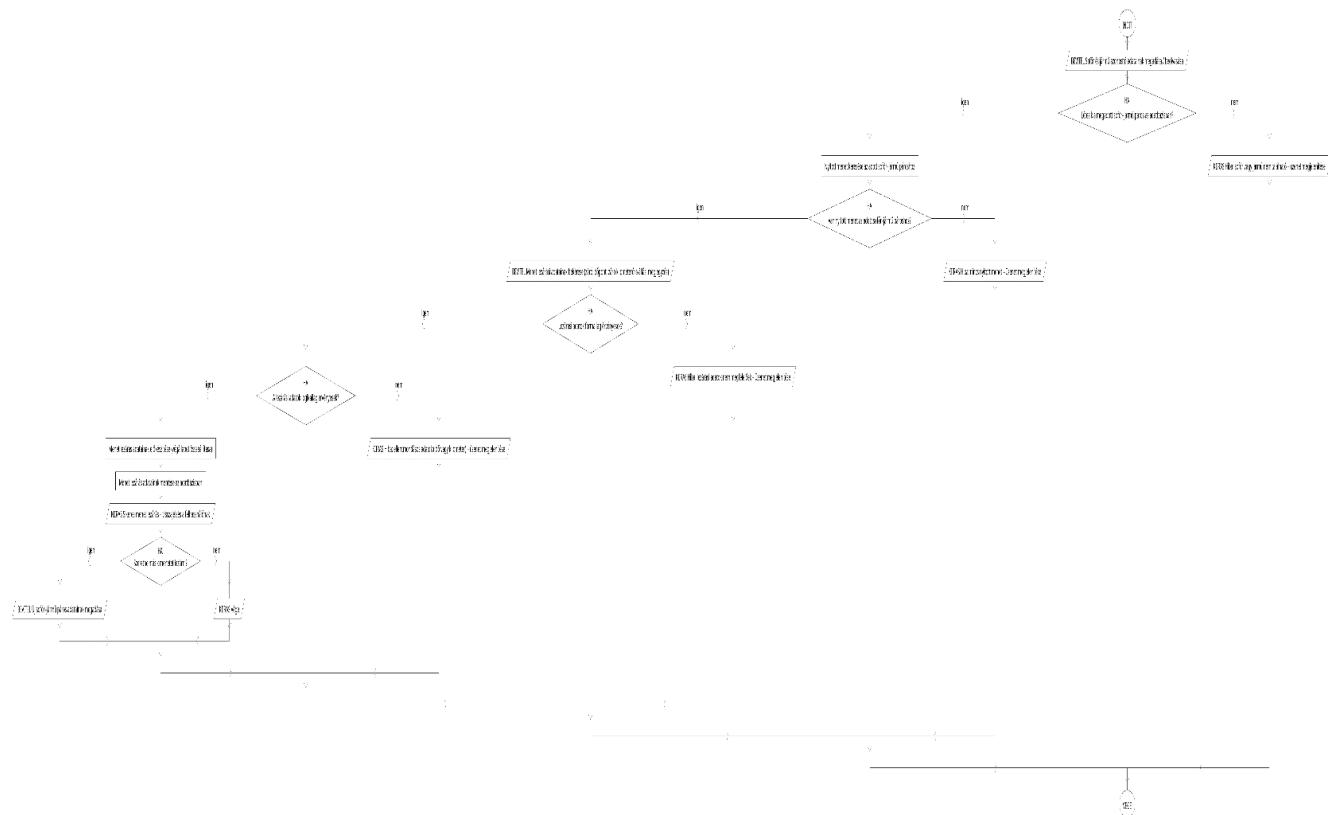


12. Menet lezárás

12.1 Célja:

Az algoritmus célja, hogy egy már futó menetet szabályosan lezárjon: megkeresi az adott sofőrhöz és járműhöz tartozó nyitott menetet, bekéri a lezáráshoz szükséges adatokat (záró időpont, záró kilométeróra-állás, esetleges hibák), ellenőrzi azok érvényességét, majd lezárja és elmenti a menetet. A folyamatábra bemutatja, milyen lépésekben történik a lezárás, milyen ellenőrzéseket végez a rendszer (van-e nyitott menet, konzisztens-e a távolság, hiánytalanok-e az adatok), és hogyan tér vissza a fő menübe vagy fejeződik be a művelet.

12.2 Folyamatábra:

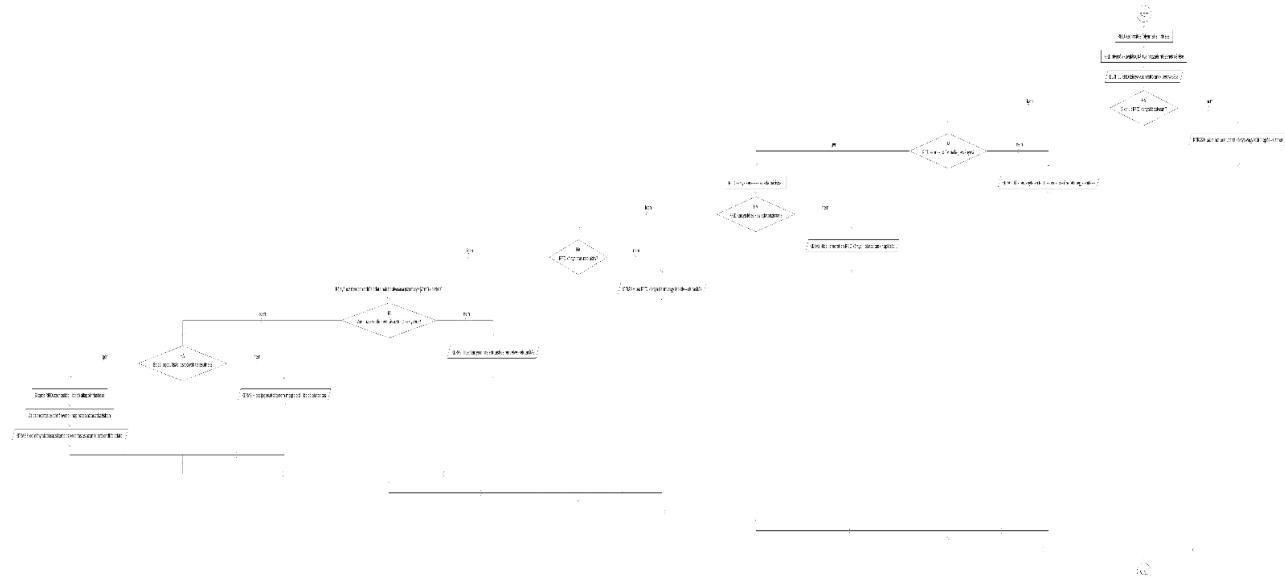


13. RFID azonosítás belső logika

13.1 Célja:

Az algoritmus célja, hogy egy RFID kártya beolvasása után megbízhatóan eldöntse, hogy az adott kártya érvényes-e, engedélyezett-e, és melyik belső entitáshoz (pl. személyhez, járműhöz, eszközhöz) tartozik. A belső logika ellenőrzi a kártya azonosítójának formátumát, meglétét az adatbázisban, státuszát (aktív, tiltott, lejárt), valamint az adott helyzetre vonatkozó jogosultsági szabályokat, majd ennek alapján engedélyez vagy megtagad egy további műveletet.

13.2 Folyamatábra:

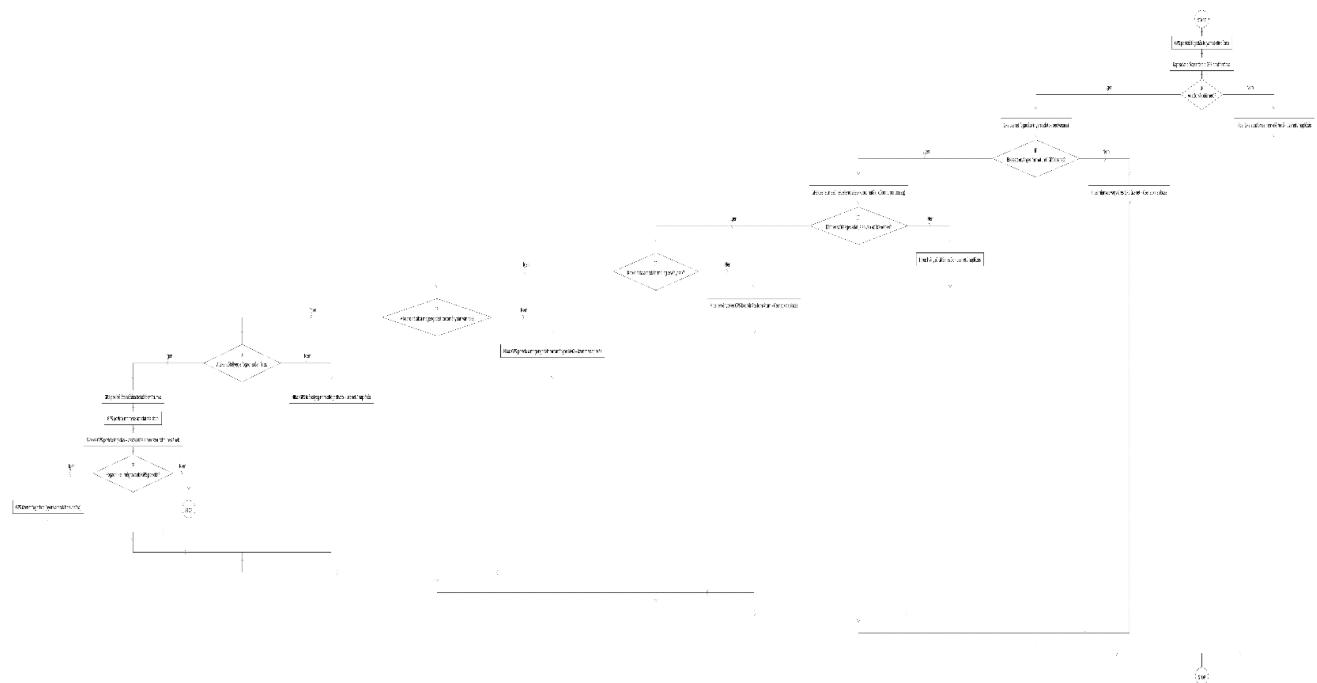


14. GPS pozíció fogadás

14.1 Célja:

Az algoritmus célja, hogy egy külső forrásból (pl. jármű fedélzetén egységéből vagy tracker eszközből) érkező GPS pozíciót fogadjon, ellenőrizze annak érvényességét (formátum, időbélyeg, koordináták tartománya), majd belső, egységesített formában eltárolja. A folyamatábra bemutatja, hogyan jön be az adat, milyen lépésekben történik a hibakezelés (hiányos üzenet, rossz formátum, túl régi pozíció), és mikor tekinti a rendszer a pozíciót sikeresen fogadottnak.

14.2 Folyamatábra:

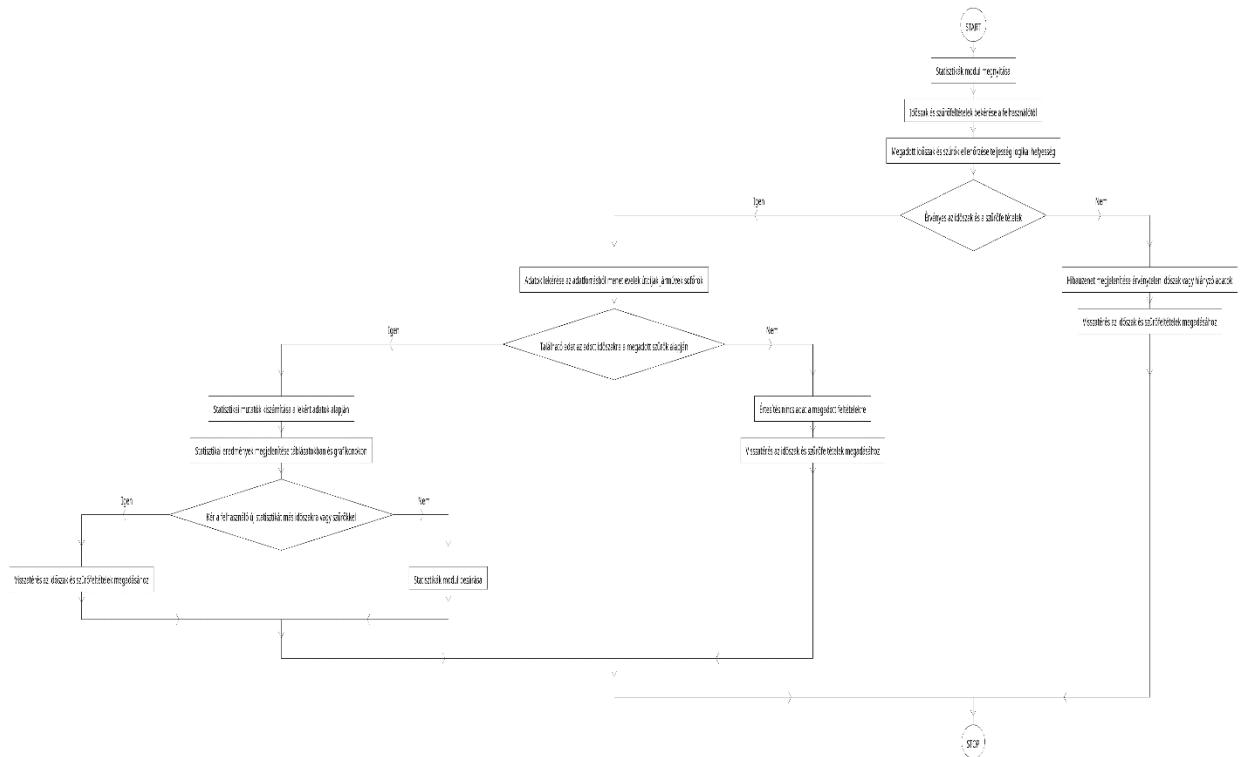


15. Statisztikák

15.1 Célja:

A statisztika modul célja, hogy egy megadott időszakra és szűrőfeltételekre (pl. cég, jármű, sofőr) összesítéseket készítsen a rendszer adataiból. Tipikusan járművenkénti futásteljesítményt, útdíj költségeket és sofőrönkénti munkaidőt számol, majd ezeket táblázatosan (és akár grafikonon) jeleníti meg. A folyamat addig ismételhető, amíg a felhasználó újabb időszakra, újabb feltételekkel nem kér statisztikát, vagy be nem zárja a modult.

15.2 Folyamatábra:

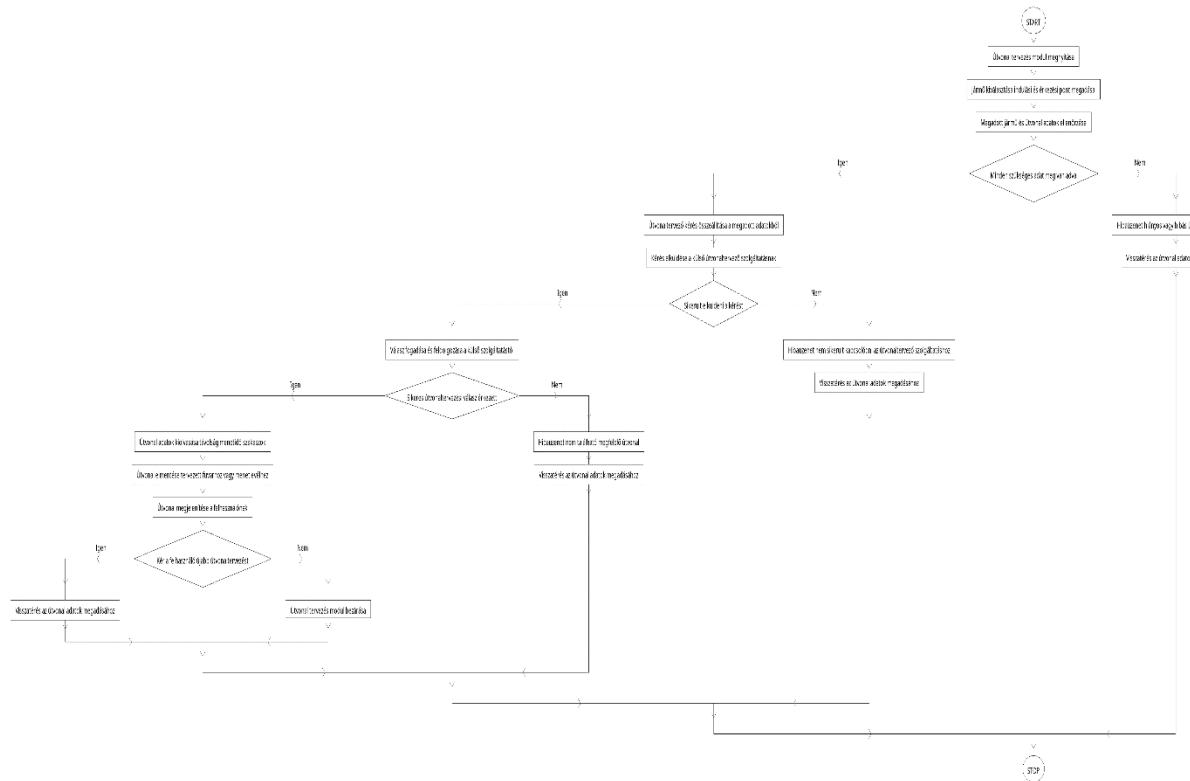


16. Útvonal tervezés

16.1 Célja:

Az útvonaltervezés algoritmus célja, hogy egy kiválasztott járműhöz és megadott indulási-érkezési ponthoz (esetleg köztes megállókkal) kiszámítson egy javasolt útvonalat. A folyamat ellenőrzi a megadott adatok helyességét, szükség esetén külső útvonaltervező szolgáltatást hív, majd az eredményt (távolság, idő, szakaszok) megjeleníti, és opcionálisan elmenti egy későbbi felhasználásra (menetlevélhez, útdíj kalkulációhoz). Hibák vagy hiányos adatok esetén biztonságosan visszavezet a beviteli lépésekhez.

16.2 Folyamatábra:

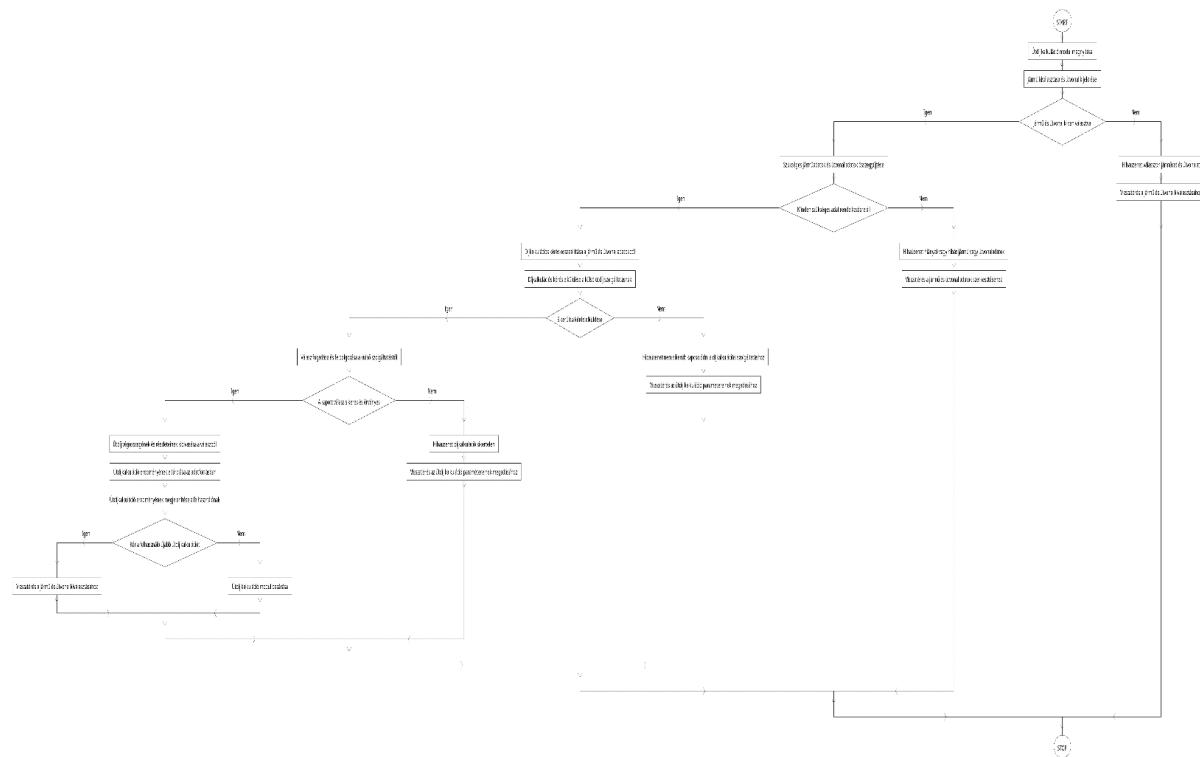


17. Útdíj kalkuláció

17.1 Célja:

Az útdíj kalkuláció algoritmus célja, hogy egy kiválasztott járműre és egy kiválasztott útvonalra / fuvorra megbecsülje és lekérje a fizetendő útdíj összegét egy külső díjkalkulációs szolgáltatás segítségével. Az algoritmus ellenőrzi, hogy minden szükséges adat (jármű paraméterek, útvonal adatok) rendelkezésre áll-e, biztonságosan kezeli a külső szolgáltatás válaszait, és az eredményt eltárolja, illetve megjeleníti a felhasználónak. Hiba vagy hiányos adatok esetén nem enged hibás kalkulációt, hanem egyértelmű visszajelzést ad és visszavezeti a felhasználót a szükséges szerkesztési pontokra.

17.2 Folyamatábra:



18. Jogosultság lekérdezés

18.1 Célja:

Ennek az algoritmusnak a célja, hogy egy felhasználó azonosítója alapján megbízhatóan megállapítsa a hozzá tartozó szerepkört / jogosultsági szintet. Ha a felhasználó nem található az adatforrásban vagy a szerepkör nem értelmezhető, az algoritmus egy alapértelmezett, biztonságos szerepkört ad vissza (pl. „ismeretlen” vagy minimális jogosultság). Ez a lekérdezés az összes további modul (főmenü, admin funkciók, stb.) jogosultság-ellenőrzésének alapja.

18.2 Folyamatábra:

