Tankolás nyilvántartó

Tartalomjegyzék

[1. Szereplők és igényeik 1](#_Toc100905946)

[2. Use-case- funkció lista 1](#_Toc100905947)

[3. Magas szintű rendszerterv 2](#_Toc100905948)

[4. Screenek 2](#_Toc100905949)

[5. Modellek 2](#_Toc100905950)

[6. Alkalmazások kiválasztása 2](#_Toc100905951)

[7.Routing 2](#_Toc100905952)

[8. Middlewarek (algoritmusok) 2](#_Toc100905953)

[9.Imlpementáció 2](#_Toc100905954)

[10. Tesztelés 2](#_Toc100905955)

[11. Bevezetés 2](#_Toc100905956)

# 1. Szereplők és igényeik

A feladatunk egy alkalmazás készítése, amelyet egyetlen felhasználó használ. Ez a felhasználó nyilván akarja tartani a tankokásait. Több autója van, melyek előre rögzítve vannak. Az autók adottak, nem akarunk módosítani az adataikon.

Az adott autó kiválasztását követően tudjunk új tankolást rögzíteni. A rögzítendő adatok: tankolás dátuma (datatime), tankolt üzemanyag mennyisége (liter, a felhasználó kerekíti igényei szerint), kilométeróra állása.

A rögzített adatokat tudjuk kilistázni összesen és autónként. Lehessen tankolást kitörölni. Tudjuk módosítani egy kiválasztott tankolás esetén a km állást és az üzemanyag mennyiséget.

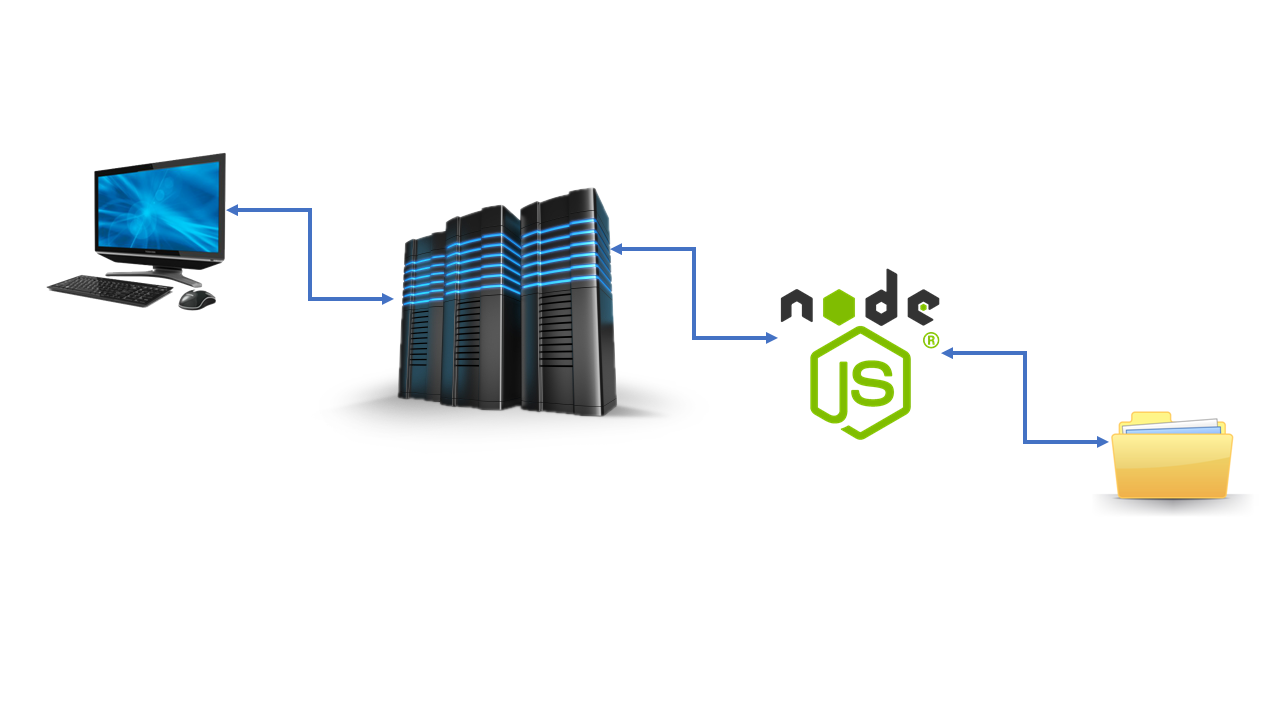
# 2. Use-case- funkció lista

Felhasználó:

* A rögzített tankolási adatok kilistázhatóak összes autóra, rögzítés sorrendjében. (Későbbi megvalósítási lehetőség az időrendbeli listázás)
* A rögzített tankolási adatok kilistázhatóak autónként. (Későbbi megvalósítási lehetőség az időrendbeli listázás)
* Új tankolás rögzítése.
* Egy rögzített tankolás esetén a km állás és az üzemanyag mennyisége módosítható
* Egy rögzített tankolás törölhető

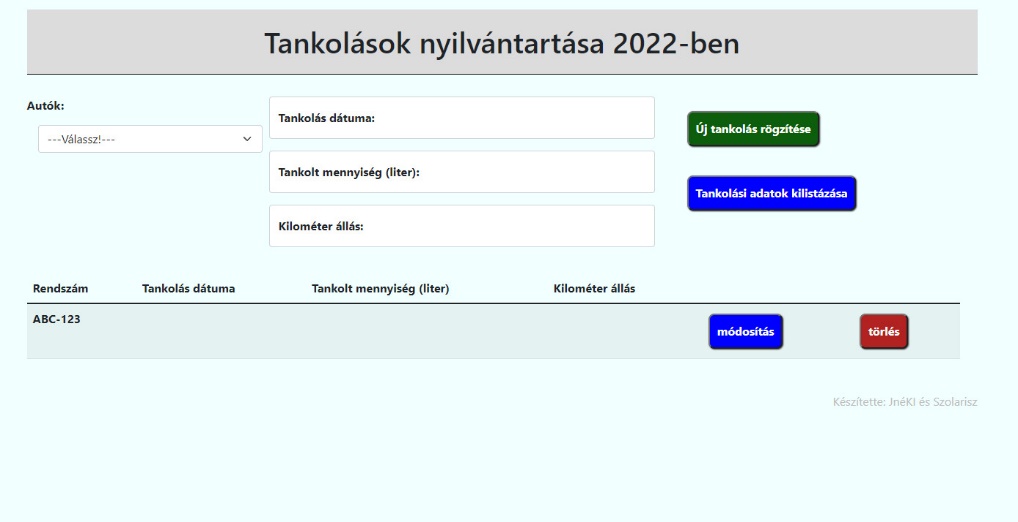
A rendszert egyetlen felhasználó használja. Nem nyilvános, így hitelesítésre nincs szükség. Nincs több felhasználó, így további funkciók sem merülnek fel

# 3. Magas szintű rendszerterv



Kliens-Szerver-szerveren node JS-mentés filerendszerbe

# 4. Screenek

  
Az autó kiválasztására és az adatok rögzítésére szolgáló képernyő. Ez egyben a nyitóképernyő is



Az adatok listázására szolgáló képernyő. Az egeydi és az összesen listához is ezt a képernyőtervet használjuk, ezért a rendszámot az első oszlopban feltüntetjük.

# 5. Modellek

Az adatok tárolására egyetlen modellre van szükség:

**tankolás**

**{**

**id : A tankolás azonosítója(uuid),**

**rendszam : A jármű forgalmi rendszáma**

**idopont : tankolás időpontja,**

**mennyiseg : a tankolt üzemanyag egész literre kerekítve (nem a rendszer kerekít)**

**kmora : A tankolás időpontjában az adott gépjármű kilométerórájának az állása**

**}**

# 6. Alkalmazások kiválasztása

* node JS
  + express
  + LokiJS
  + EJS
  + moment
  + body-parser
  + további modulokra is szükség lehet, ezeket a tervezési fázisban nem látjuk előre

# 7.Routing

GET /

|  |  |
| --- | --- |
| tervezett feladat | kezdőképernyő betöltése |
| bemenet | - |
| kimenet | a renderelt állomány |

GET /auto/:frgsz

|  |  |
| --- | --- |
| tervezett feladat | Adott azonosítójú autó tankolásainak listája |
| bemenet | frgsz (az autó rendszáma) |
| kimenet | az adott autó tankolásainak a renderelt listája |

GET /auto

|  |  |
| --- | --- |
| tervezett feladat | Az összes autó tankolásának a listája |
| bemenet | - |
| kimenet | a tankolások renderelt listája |

PUT /tankolas

|  |  |
| --- | --- |
| tervezett feladat | Egy tankolás rögzítése |
| bemenet | A tankolás adatai űrlapról |
| kimenet | A rögzített adatok JSON formátumban |

DELETE /tankolas/:id

|  |  |
| --- | --- |
| tervezett feladat | Egy tankolás törlése |
| bemenet | A tankolás azonosítója |
| kimenet | A törölt adatok JSON formátumban |

PATCH /tankolas/:id

|  |  |
| --- | --- |
| tervezett feladat | Egy tankolás adatainak a módosítása |
| bemenet | tankolás azonosítója |
| kimenet | A módosított adatok JSON formátumban |

# 8. Middlewarek (algoritmusok)

# 9.Imlpementáció

# 10. Tesztelés

# 11. Bevezetés