# Projekta uzdevums

## 1.1. Projekta tēma

Projekta tēma ir pilsētas sabiedriskā transporta satiksmes vadības sistēma vadības sistēma, kuru izmantos satiksmes pārvalde.

## 1.2. Projekta mērķis

Mērķis ir izstrādāt funkcionālu un ērti lietojamu vadības sistēmu, ar kuras palīdzību var pievienot, dzēst, mainīt, apskatīt informāciju par transportiem, vadītājiem, maršrutiem.

## 1.3. Uzdevumi

1. Savienojuma izveide ar datu bāzi, informācijas saglabāšana datu bāzē;
2. Izstrādāt lietojumprogrammas struktūru. Izstrādāt datu struktūras. Katram objektam jābūt savai struktūrai.
3. Izstrādāt lietotāja interfeisu – attēlojumu, kā tiks attēloti dati, kā tiks atlasīta tā vai cita darbība ar datiem, izvēlnes utt.
4. Izstrādāt algoritmus, kas veic programmas galvenās metodes (meklēšana, aprēķinātie raksturlielumi, dzēšana, kārtošana utt.)
5. Uzrakstīt programmas kodu.
6. Testēt programmu.
7. Uzrakstīt atskaiti.

## 1.4. Programmas prasības

1. Iekļautas metodes:

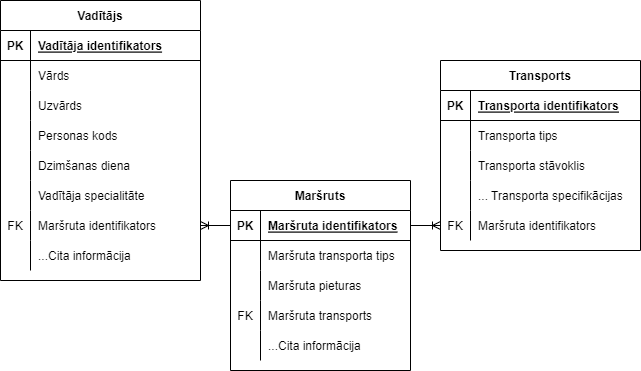
* Pievienot jaunus datu ierakstus;
* Saglabātās informācijas datu attēlošana;
* Datu dzēšana pēc noteikta kritērija;
* Datu meklēšana un filtrēšana ar iespēju uzstādīt dažādus kritērijus (vismaz 3);
* Datu kārtošana pēc vairākiem kritērijiem.

1. Atbilstība pēc dotajiem dokumentiem

# 2. Priekšmeta jomas analīze

## 2.1. Priekšmeta jomas apraksts

Priekšmeta joma ir satiksmes vadības sistēma, kuru izmantos satiksmes pārvalde. Tas dos iespēju mainīt, rediģēt, pievienot un dzēst maršrutus, kurus vada nodarbinātie vadītāji un katram maršrutam ir nozīmēti savi transportlīdzekļi.

Projektā tiks izmantotas 3 atsevišķas bāzes klases – Vadītājs, Maršruts un Transports. Transporta klase tiks sadalīta 4 apakšklasēs, kas norādīs katra transporta veidu, t.i. Autobuss, Tramvajs, Trollejbuss, Mikroautobuss.

## 2.2. Objektu apraksts

Apakšā tiek dota informācija par objektu glabātiem metadatiem un to datu tipiem. Šie objekti iekļauj:

1. Vadītājs – Informācija par transporta vadītāju jeb šoferi. Vadītājam tiks piešķirts maršruts.

* Vārds – *string* datu tips, glabā informāciju par vadītāja vārdu;
* Uzvārds – *string* datu tips, glabā informāciju par vadītāja uzvārdu;
* Personas kods – *string* datu tips, glabā informāciju par vadītāja personas kodu;
* Dzimšanas diena – *date* datu tips, glabā informāciju par vadītāja dzimšanas dienu;
* Vadītāja specialitāte – *string[] vai List<string>* datu tips, glabā informāciju par vadītāja specialitāti. Vadītājs varēs vadīt tikai tos transportlīdzekļus, uz kuriem vadītājam ir specialitāte.
* Maršruta identifikators\* – *int* datu tips, glabā piešķirtā maršruta identifikatoru.

2. Maršruts – Informācijas par maršrutu. Pie katra maršruta tiks pievienoti atbilstošie vadītāji un transporti.

* Maršruta transporta tips: *string[] vai List<string>* datu tips, glabā informāciju par maršruta transporta tipu, norāda maršruta tipus, kas var veikt šo maršrutu. Piemēram, vienu maršrutu var veikt gan autobuss, gan mikroautobuss, taču citu maršrutu var veikt tikai tramvajs;
* Maršruta pieturas: *string[] vai List<string>* datu tips, glabā informāciju par maršruta veicamo pieturu nosaukumiem.

3. Transports – Informācija par transportu. Transportam tiks piešķirts veicamais maršruts.

* Transporta tips: *string* datu tips, glabā informāciju par transporta tipu. Transportu varēs piešķirt tikai pie tāda veicamā maršruta, kas atbilstīs ar transporta tipiem. Piemēram, tramvajs nevarēs veikt autobusa maršrutus.
* Transporta stāvoklis: *string* datu tips, glabā informāciju par transporta stāvokli. Tas var būt braucamā stāvoklī, bojāts, tiek remontēts u.t.t.
* Maršruta identifikators\*: *int* datu tips, glabā informāciju par maršruta identifikatoru, kas ir piešķirts dotajam transportlīdzeklim.

\*- Identifikatori tiks saglabāti datu bāzē. Kodā, iespējams, maršruta objektam tiks piešķirts saraksts ar atbilstošā maršruta vadītājiem un transportlīdzekļiem.