**Profesionālās izglītības kompetences centrs**

**„Rīgas Valsts tehnikums”**

**Izglītības programma: Programmēšana**

**KVALIFIKĀCIJAS DARBS**

**"OneStop" transporta sarakstu sistēma**

**Audzēknis: Daniils Fjodorovs**

**Grupa: DP4-3**

**Rīga**

**2023**

# Ievads

Sabiedriskā transporta kustība ir viens no pamata veidiem, kā cilvēki pārvietojas pilsētās un valstīs. Lai gan sabiedriskais transports ir ļoti efektīvs, lai samazinātu satiksmes sastrēgumus un radītu mazāku ietekmi uz vidi, bieži vien tas var būt neskatoties uz to sarežģīts un neērts, jo daudzi cilvēki cīnās ar sliktiem sarakstiem un neprecīzām informācijas avotiem. Lai uzlabotu šo situāciju un nodrošinātu precīzu un uzticamu informāciju par sabiedrisko transportu, tiek izmantotas Sabiedriskā transporta sarakstu skatīšanas sistēmas. Šīs sistēmas ne tikai atvieglo cilvēku dzīvi, bet arī palīdz samazināt satiksmes sastrēgumus, uzlabo vidi un palielina sabiedrības mobilitāti. Šajā ievadā mēs apskatīsim Sabiedriskā transporta sarakstu skatīšanas sistēmas nozīmi, priekšrocības un to, kā tās var uzlabot cilvēku pieredzi ceļojot ar sabiedrisko transportu.

# 1. Uzdevuma nostādne

Jāizveido Sabiedriskā transporta sarakstu skatīšanas sistēmu, kas būs spējīga nodrošināt precīzu informāciju par sabiedrisko transportu. Sistēmai vajadzētu ļaut lietotājiem viegli atrast un skatīt sabiedrisko transportu un maršrutu informāciju, kā arī informāciju par kustību, kavējumiem un atcelšanām. Sistēmai vajadzētu arī nodrošināt iespēju meklēt informāciju par sabiedrisko transportu, izmantojot dažādus kritērijus, piemēram, maršruta numuru, pieturu nosaukumu vai atrašanās vietu.

Funkcionālās prasības Sabiedriskā transporta sarakstu skatīšanas sistēmai:

* Lietotāja autentifikācija: Sistēmai vajadzētu nodrošināt lietotāja autentifikācijas funkciju, lai lietotāji varētu izveidot un piekļūt savam personīgajam profilam, saglabāt savus iecienītākos maršrutus un iestatīt paziņojumus par kavējumiem un atcelšanām;
* Sabiedriskā transporta sarakstu meklēšana: Sistēmai vajadzētu ļaut lietotājiem meklēt sabiedriskā transporta sarakstus, izmantojot dažādus kritērijus, piemēram, maršruta numuru, pieturu nosaukumu vai atrašanās vietu. Lietotājiem vajadzētu arī varēt meklēt sarakstus, kas atbilst noteiktam laika intervālam un datumiem;
* Sabiedriskā transporta kustības informācijas atjaunināšana: Sistēmai vajadzētu nodrošināt ātru un precīzu informācijas atjaunināšanu par sabiedrisko transportu, ieskaitot kavējumus, atcelšanas, maršrutu izmaiņas un citu informāciju, kas varētu ietekmēt lietotāju ceļojumu plānošanu;
* Transporta maršrutu un grafiku pārvaldīšana: Sistēmai vajadzētu ļaut administratoriem pārvaldīt sabiedriskā transporta maršrutus un grafikus, kā arī pievienot jaunus maršrutus un sarakstus;
* Notikumu paziņošana: Sistēmai vajadzētu nodrošināt iespēju paziņot lietotājiem par notikumiem, kas varētu ietekmēt sabiedriskā transporta kustību, piemēram, kavējumiem, atcelšanām vai maršrutu izmaiņām;

# ER-diagramma

# 2. PRASĪBU SPECIFIKĀCIJA

## 2.1 Ieejas un izejas informācijas apraksts

Ieejas informācijas apraksts:

1. Lietotāja konta informācija:

* Lietotāja vārds: alfabēta simboli, līdz 20 simboliem
* Lietotāja uzvārds: alfabēta simboli, līdz 20 simboliem
* E-pasta adrese: formāts jābūt "vārds@domēns.lv", līdz 50 simboliem
* Parole: līdz 20 simboliem, atļauti alfabēta un ciparu simboli

1. Ziņojumu informācija:

* Ziņojuma teksts: burtu teksts ar izmēru līdz 500 simboliem
* Datums un laiks: formāts "dd.mm.yyyy hh:mm", datums jābūt aktuālam

1. Komentāru informācija:

* Komentāra teksts: burtu teksts ar izmēru līdz 500 simboliem
* Datums un laiks: formāts "dd.mm.yyyy hh:mm", datums jābūt aktuālam

1. Kartes informācija:

* Adrese: līdz 100 simboliem, atļauti alfabēta un ciparu simboli
* Kartes koordinātas: formāts "latitūde, longitudāde", piemēram, "56.9539, 24.1052"

Izejas informācijas apraksts:

1. Attēli un teksts, kas parāda lietotājam informāciju par maršrutiem, pieturām, transporta sarakstiem, jaunumiem un citiem saistītiem jautājumiem

## 2.2 Funkcionālās prasības

1. Lietotājam jāspēj meklēt sabiedriskā transporta maršrutus un laika grafikus, ievadot sākuma un galamērķa adresi.
2. Lietotājam jāspēj skatīt informāciju par sabiedriskā transporta pieturvietām, tostarp to atrašanās vietu un tuvākajiem maršrutiem.
3. Lietotājam jāspēj skatīt precīzu informāciju par sabiedriskā transporta maršrutiem, tostarp to nosaukumiem, numuriem, laika grafikiem un galamērķiem.
4. Lietotājam jāspēj reģistrēties un pārvaldīt savu kontu, lai izveidotu un rediģētu savus iecienītākos maršrutus.
5. Lietotājam jāspēj izveidot un rediģēt sava profila informāciju, tostarp savu vārdu, e-pasta adresi un paroli.
6. Lietotājam jāspēj lasīt un rakstīt komentārus par sabiedriskā transporta pieturvietām un maršrutiem.

## 2.3 Nefunkcionālās prasības

1. Sistēmai jābūt drošai un aizsargātai pret kaitēkļiem, kā arī jānodrošina datu privātums un konfidencialitāte.
2. Sistēmai jābūt stabila un viegli pielāgojamai jaunām funkcijām un uzlabojumiem.
3. Sistēmai jānodrošina ātrs un efektīvs meklēšanas process, lai lietotājiem nebūtu jāgaida ilgi.
4. Sistēmai jābūt viegli lietojamai un intuitīvai, lai ar to varētu strādāt gan pieredzējuši, gan mazāk pieredzējuši lietotāji.
5. Sistēmai jābūt pieejamai un saderīgai ar vairākiem operētājsistēmām un mobilajiem tālruņiem.
6. Sistēmai jābūt skalējamai un jāspēj apkalpot lielu lietotāju skaitu, nezaudējot veiktspēju.

## 2.4. Uzdevuma risināšanas līdzekļu izvēles pamatojums

React Native: React Native ir populārs ietvars mobilo lietotņu izstrādei gan iOS, gan Android platformām. Tas ļauj izveidot šķērsojumu platformas lietotnes, kas nozīmē, ka vienā kodu bāzē var izveidot lietotnes abām platformām, taupot izstrādes laiku un pūles. Turklāt tam ir liela kopiena un ekosistēma, kas var nodrošināt noderīgus resursus un atbalstu.

TypeScript: TypeScript ir JavaScript paplašinājums, kas nodrošina izvēles iespējas statiskajai tipizācijai un citus valodas elementus, kas var palīdzēt uzlabot koda kvalitāti, uzturamību un mērogojamību. Tas var agrīnā izstrādes procesā atrast kļūdas, padarot kodu vieglāk uzturēt ilgtermiņā.

PostgreSQL: PostgreSQL ir populārs atvērtā koda relāciju datu bāzu pārvaldības sistēma, kas nodrošina augstu mērogojamību, datu integritāti un uzticamību. Tam ir plašs funkciju klāsts, ieskaitot ACID atbilstību, JSON datu atbalstu un paplašināmību, izmantojot lietotāju definētās funkcijas.

GraphQL: GraphQL ir vaicājuma valoda un izpildlaiks, kas nodrošina efektīvāku un elastīgāku veidu, kā vaicāt datus salīdzinājumā ar tradicionālajām REST API. Tas ļauj klientiem precīzi norādīt, kādus datus viņi vēlas, samazinot datu daudzumu, kas tiek pārsūtīts pār tīklu, un uzlabojot veiktspēju.

Docker: Docker ir konteineru platforma, kas ļauj izveidot un izvietot lietotnes viegli pārvietojamos konteineros. Tas nodrošina vienmērīgu darbības vidi, padarot lietotņu izvietošanu un mērogošanu dažādās platformās vieglāku.

Šie tehnoloģiju risinājumi nodrošina vairākas priekšrocības, piemēram, iespēju izveidot vienotu koda bāzi, kas strādā gan iOS, gan Android platformās, tādējādi samazinot izstrādes laiku un pūles. TypeScript piedāvā iespēju izmantot statisku tipizēšanu un citus valodas līmeņa risinājumus, kas palīdz uzlabot kodu kvalitāti, uzturamību un skalējamību. PostgreSQL ir uzticams un augsti skalējams datubāzes pārvaldības sistēma, kas piedāvā plašas iespējas, piemēram, atbalstu JSON datiem un lietotāju definētām funkcijām. GraphQL ir moderns vaicājumu valoda un izpildes vide, kas nodrošina efektīvāku un elastīgāku veidu, kā piekļūt datiem salīdzinājumā ar tradicionāliem REST API. Docker piedāvā vienotu izpildes vidi, kas atvieglo aplikāciju izstrādi un izvietošanu dažādās platformās, savukārt e-pasta verifikācijas bibliotēka var palīdzēt uzlabot lietotāju datu drošību un integritāti, verificējot, ka lietotāju ievadītās e-pasta adreses ir derīgas un pieder tiem.

Kopā šie tehnoloģiju risinājumi veido spēcīgu un efektīvu izstrādes rīku komplektu, kas var palīdzēt izveidot kvalitatīvu un lietotājam draudzīgu publiskā transporta informācijas sistēmu.