

**Profesionālās izglītības kompetences centrs „Liepājas Valsts tehnikums”**

# Latvijas apskates objektu uzskaites sistēma

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Profesionālā kvalifikācija Programmēšanas tehniķis

Grupas nosaukums 4.PT

Projekta izstrādātājs Daņila Anfinogenovs

Eksāmena datums 2022.21 jūnijs

# Saturs

[**Titullapa**](#_heading=h.5q7t44o2gbcv) **0**

[**Saturs**](#_heading=h.y38pdrvyy47g) **0**

[**Ievads**](#_heading=h.ggbo8vfvgc54) **1**

[**1. Uzdevuma formulējums**](#_heading=h.1nbcgzvtv7mo) **2**

[**2. Programmatūras prasību specifikācija**](#_heading=h.wurq3xlf7u7q) **3**

[2.1. Produkta perspektīva](#_heading=h.1294e54g4a9z) 3

[2.2. Sistēmas funkcionālās prasības](#_heading=h.2r7xdpkg2na) 3

[2.3. Sistēmas nefunkcionālās prasības](#_heading=h.5j6pix5tkqqg) 3

[2.4. Gala lietotāja raksturiezīmes](#_heading=h.ot4qwlrc4gss) 3

[2.5. Lietoto terminu un saīsinājumu skaidrojumi](#_heading=h.7a36efs7soj8) **3**

[**3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums**](#_heading=h.y2g9rmvoicxo) **4**

[3.1. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts](#_heading=h.mplv7zoglkef) 4

[3.2. Izvēlēto risinājuma līdzekļu un valodu apraksts](#_heading=h.indyxofljngv) 4

[**4. Sistēmas modelēšana un projektēšana**](#_heading=h.dh7yrnsfddon) **5**

[4.1. Sistēmas struktūras modelis](#_heading=h.uzw3ip54uq9y) 5

[4.2. Klašu diagramma / ER diagramma (Obligāti, ja sistēmā ir klases un/vai](#_heading=h.o93vtpm3n1sm) 5

[tabulas)](#_heading=h.2t3r1o37nbz3) 5

[4.3. Funkcionālais un dinamiskais sistēmas modelis](#_heading=h.yea45936zvas) 5

[4.4. Aktivitāšu diagramma (Activity)](#_heading=h.fl6g34tmux5q) **5**

[4.5. Lietojumgadījumu diagramma (Use Case)](#_heading=h.7rypyaa3rzpz) **5**

[4.6. Sistēmas moduļu apraksts un algoritmu shēmas](#_heading=h.w5cwywmwslkp) **5**

[**5. Lietotāju ceļvedis**](#_heading=h.mzghfh7w56df) **6**

[**6. Testēšanas dokumentācija**](#_heading=h.v4il8voh266u) **7**

[6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums](#_heading=h.fnylku50uige) 7

[6.2. Testpiemēru kopa](#_heading=h.encg50immflm) 7

[6.3. Testēšanas žurnāls](#_heading=h.lyp4rvxbqa7w) **7**

[**7. Secinājumi**](#_heading=h.meg7d2wdzymw) **8**

[**8. Lietoto saīsinājumu saraksts**](#_heading=h.ffnz9afbvpzo) **9**

[**9. Literatūras un informācijas avotu saraksts**](#_heading=h.yo7xx8ut09ke) **10**

[**Pielikums/-i**](#_heading=h.1ycf6dgwsr6k) **11**

# Ievads

Šis dokuments satur „Latvijas apskates objektu uzskaites sistēma” tehnisko dokumentāciju. Dokumentā ietvertas vispārējās prasības, atsevišķas iespējas sistēmā un konkrētie analītisko atskaišu veidi. “Latvijas apskates objektu uzskaites sistēma” ir iedzīvotāju atvieglots veids, kā atrast interesējošos apskates objektus Latvijā uzzināt kur tie atrodas un informācija par tiem

# 1. Uzdevuma formulējums

Izveidot web aplikāciju kuras lietotājam būs iespējams atlasīt un apskatīt interesējošās pilsētās apskates objektus, kā ari skatīt populārakos apskates objektus Latvijā

# 2. Programmatūras prasību specifikācija

## **2.1. Produkta perspektīva**

Risinājums padarīs Latvijas iedzīvotājiem un tūristiem, kas ieceļo Latvijā web aplikācijā pieejamu informāciju par populārākiem Latvijas apskates objektiem

## **2.2. Sistēmas funkcionālās prasības**

* Mājaslapa ļaus iedzīvotājiem atlasīt apskates objektus, apskatīt tos detalizēti.
* Mājaslapa ļaus mainīt lapas stilu uz nakts vai attiecīgi dienas režīmu.
* Lietotājam būs iespējams meklēt apskates vietu pēc nosaukuma

## 2.3. Sistēmas nefunkcionālās prasības

* Web aplikācijai ir jāizvada meklēšanas rezultāti 1-3 sekunžu laikā pēc saņemšanas meklēšanas pieprasījuma.
* Web aplikācijai jābūt viegli izmantojamai, lai lietotājs varētu ātri un viegli atrast sev nepieciešamo pilsētu un apskates objektu.
* Web aplikācija ir uzticama informācija.

## **2.4. Gala lietotāja raksturiezīmes**

* Lietotājam jābūt uzinstalētai pārlukprogrammai
* Pamata prasmses pārlūkprogrammas lietošanā, lai varētu meklēšanas rindā ierakstīt attiecīgo domēnu.

## **2.5. Lietoto terminu un saīsinājumu skaidrojumi**

# 3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums

## **3.1. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts**

Iespējamie alternatīvie risinājuma līdzekļī varēja būt dažādi, tā kā projekts Latvijas apskates objektu uzskaites sistēma ir web aplikācija, kura veidota reactJS bibliotēkā. Protams, bija iespēja veidot darbu html, php un javascript valodās, bet prakses laikā tiek apmācīts reactJS, un jūtos pārliecināts, ka pietiks zināšanas, lai izveidotu šo projektu.

## **3.2. Izvēlēto risinājuma līdzekļu un valodu apraksts**

Visual Studio Code - ir pirmkoda redaktors, ko var izmantot ar dažādām programmēšanas valodām, tostarp Java, JavaScript, Go, Node.js, Python un C++. Tā ir balstīta uz Electron ietvaru, ko izmanto, lai izstrādātu Node.js tīmekļa lietojumprogrammas, kas darbojas ar Blink izkārtojuma dzinēju. Izvēlējos šo valodu, jo apguvu to lietošanu apmācības laikā.

ReactJS - React ir uz JavaScript balstīta lietotāja saskarnes izstrādes bibliotēka. To pārvalda Facebook un atvērtā koda izstrādātāju kopiena. Lai gan React ir bibliotēka, nevis valoda, tā tiek plaši izmantota tīmekļa izstrādē. Bibliotēka pirmo reizi parādījās 2013. gada maijā, un tagad tā ir viena no visbiežāk izmantotajām priekšgala bibliotēkām tīmekļa izstrādei. Šo bibliotēku izvēlējos, jo tā ir visbiežāk izmantojamā bibliotēka tīmekļu izstrādē un prakses laikā ar to apmāca

Node.js - ir atvērtā pirmkoda, vairāku platformu, aizmugures JavaScript izpildlaika vide, Node.js ļauj izstrādātājiem izmantot JavaScript, lai rakstītu komandrindas rīkus un servera puses skriptus — palaist skriptus servera pusē, lai izveidotu dinamisku tīmekļa lapas saturu, pirms lapa tiek nosūtīta uz lietotāja tīmekļa pārlūkprogrammu. Līdz ar to Node.js ir JavaScript paradigma, kas apvieno tīmekļa lietojumprogrammu izstrādi vienā programmēšanas valodā, nevis dažādās valodās servera un klienta puses skriptiem.

CSS - Cascading Style Sheets (CSS) ir stila lapu valoda, ko izmanto, lai aprakstītu dokumenta noformējumu, kas rakstīts iezīmēšanas valodā, piemēram, HTML. CSS ir globālā tīmekļa stūrakmens tehnoloģija līdzās HTML un JavaScript.CSS ir izstrādāts, lai nodrošinātu prezentācijas un satura nodalīšanu, tostarp izkārtojumu, krāsas un fontus. Izvēlējos šo valodu, jo to apmācīja skolā un ir ļoti grūti un neērti rakstīt stila izkārtojumu html vai javascript failos.

HTML - ir iezīmēšanas valoda, kas ir vērsta uz vienas lappuses teksta dokumentu attēlošanu speciālās programmas HTML lietotāja aģentos, kuru visizplatītākais paveids ir tīmekļa pārlūkprogramma. HTML sniedz iespēju dokumenta pamatsaturu sasaistīt ar dažāda veida metadatiem un atveidei nepieciešamo informāciju. Atveides informācija var būt gan nelielas teksta dekorācijas, kā piemēram atsevišķa vārda pasvītrošana vai attēla ievietošana tekstā, gan sarežģīti skripti, attēlkartes un formu. Izvēlējos tieši šo valodu, jo to apguvu apmācības laikā. Pēc manām domām ir interesanta un diezgan viegla valoda, jo ir daudz iespējas web izstrādē

ANT Design - ir React UI bibliotēka, kurā ir daudz viegli lietojamu komponentu, kas ir noderīgi, veidojot elegantas lietotāja saskarnes. Ķīnas konglomerāta Alibaba izveidoto Ant Design izmanto vairāki lieli nosaukumi: Alibaba (protams), Tencent, Baidu un citi. Lai gan Material-UI joprojām ir vispopulārākā React UI bibliotēka ar vairāk nekā 40 000 zvaigznēm vietnē Github. Izvēlējos šo bibliotēku, jo prakses vietā kolēģi izmanto to un apmācīja, ka izmantot šo bibliotēku.

Font Awesome – ir interneta ikonu bibliotēka un rīkkopa, ko izmanto miljoniem dizaineru, izstrādātāju un satura veidotāju. Font Awesome pamatā ir CSS un Less. No 2020. gada Font Awesome izmantoja 38% vietņu, kurās tiek izmantoti trešo pušu fontu skripti, ierindojot Font Awesome otrajā vietā pēc Google fontiem.Visvairāk izmantoju font awesome icons, jo ikonas tajā ir visādas, protams, manā darbā nav tik daudz viņas, bet tās ir nepieciešamas

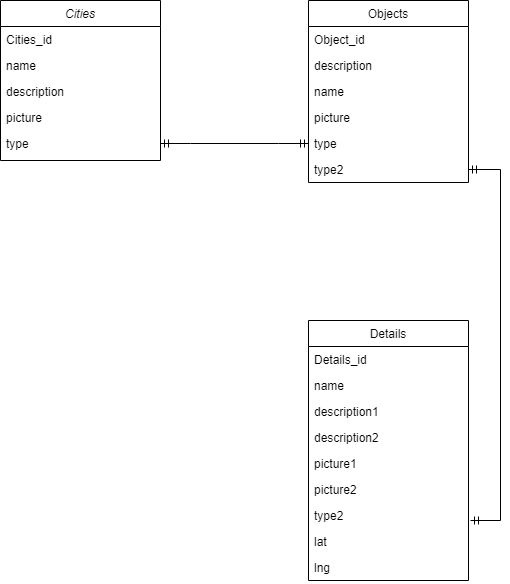
# 4. Sistēmas modelēšana un projektēšana

## 4.1.Sistēmas struktūras modelis

## 4.2. Klašu diagramma / ER diagramma (Obligāti, ja sistēmā ir klases un/vai

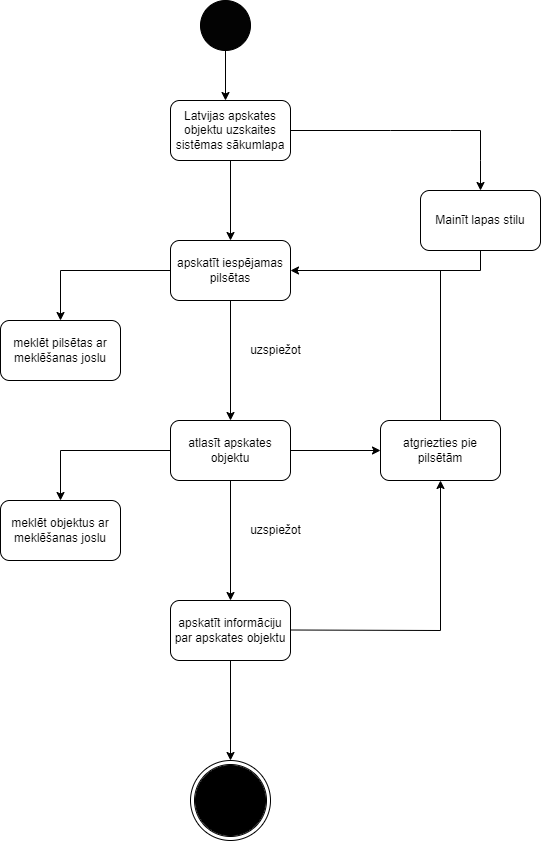
## tabulas)

Skatoties pēc datubāzes tika izveidota ER diagramma ar 3 tabulām, Cities jeb pilsētas, Objects jeb objekti un Details jeb detaļas.

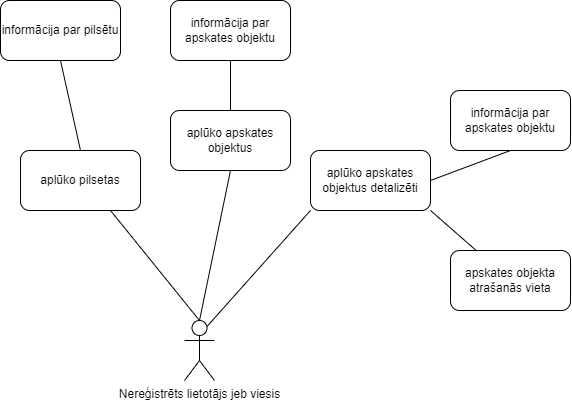


## 4.3.Funkcionālais un dinamiskais sistēmas modelis

## 4.4. Aktivitāšu diagramma (Activity)



## 4.5. Lietojumgadījumu diagramma (Use Case)



## 4.6.Sistēmas moduļu apraksts un algoritmu shēmas

# 5. Lietotāju ceļvedis

# 6. Testēšanas dokumentācija

## **6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums**

## **6.2. Testpiemēru kopa**

## 6.3. Testēšanas žurnāls

# 7. Secinājumi

# 8. Lietoto saīsinājumu saraksts

ER

# 9. Literatūras un informācijas avotu saraksts

<https://en.wikipedia.org/wiki/Visual_Studio_Code>

<https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>

<https://en.wikipedia.org/wiki/Node.js>

<https://en.wikipedia.org/wiki/CSS>

<https://lv.wikipedia.org/wiki/HTML>

# 

# Pielikums/-i