Rīgas 1.Tālmācības vidusskola

**Kinoteātra uzskaites sistēmas izstrāde**

Programmēšana I

Autors: Toms Breijers

Rīga 2024

**Saturs**

1. Problēmas izpēte un analīze 3

2. Programmatūras prasību specifikācija 4

2.1. Risinājuma mērķauditorijas izvēle un tās raksturojums 4

2.2. Programmatūras produkta un tā funkciju apraksts 5

2.3. Entītiju relāciju modelis 5

2.4. Programmatūras produkta skice 6

2.5. Piemērotās licences pamatojums 11

3. Programmatūras izstrādes plāns 12

4. Atkļūdošanas un akcepttestēšanas pārskats 12

5. Lietotāja ceļvedis 13

Pielikumi 15

1.pielikums. Programmatūras kods 15

2.pielikums. Programmatūras struktūrskice 24

* **Problēmas izpēte un analīze**

Ainažu pagastā, Salacgrīvas novadā tiek plānots atklāt jaunu kinoteātri “Filmey”. Pagaidām ir pieteikušies strādāt 20 cilvēki, ņemot vērā to, ka pagasts ir diezgan mazs un populācija apmēram 700 iedzīvotāju. Kinoteātrim ir izveidots dizains un izkārtojums.Visiem visi darba amati ir zināmi un sludinājumi ir izvietoti, lai cilvēki varētu pieteikties darbam.

Novads sprieda ka kinoteātris Ainažos palīdzētu piesaistīt vietējos un ārzemniekus (tūristus), kā arī pagasta iedzīvotājiem tā būtu tuvākā izklaide kuru skaits šobrīd ir zems, taču kinoteātra dibinātājiem nav atrasts neviens programmētājs un dizaineris kurš spētu izveidot interneta lapu, kur var rezervēt biļetes un skatīties visus nākamos seansus. Mājaslapai ir nepieciešams viegls un saprotams izskats, lai visu vecuma cilvēki to varētu izmantot, jābūt arī autentifikācijas sistēmai darbiniekiem, lai ir piekļuve redzēt informāciju par rezervācijām.

* **Izpētes metodes izvēle un pamatojums**

Tā kā kinoteātrim ir 1 galvenais dibinātājs, kuram ir savas idejas un vēlmes, vispiemērotākā izpētes metode būs intervija. Intervija tiks ierakstīta, lai pēc tam veicot darbu ir audio kuru var noklausīties atkārtoti, ja kaut kas aizmirstas. Intervijas jautājumu, piemēri, kas tiks jautāti kinoteātra dibinātājam.

***Intervijas jautājumi***

* ***Kādu krāsu dizains ir jūsu kinoteātrim?***
* ***Vai ir nepieciešama lietotāju autorizācija?***
* ***Cik detalizētu mājas lapu jūs vēlētos?***

Un daži precizējošie jautājumi par kioska programmas izveidi.

Intervijas gala detalizētās atbildes tiks izmantotas turpmākai kioska programmas izveidei.

* **Izpētes procesa apraksts**

**Sagatavot un kārtīgi noformulēt jautājumus kuri tiks jautāti kinoteātra dibinātājam. Iegūstot atbildes kuras tiks ierakstītas ar telefona aplikācijas palīdzību.**

* **Izpētes datu apkopojums**

Dati tika iegūti no kinoteātra dibinātāja. Dibinātājs aprakstīja kinoteātra krāsu izvēli, izmantotās krāsas ir pelēcīgi violeta (#90657A), tonēta rozā (#B5838D), maigi tonēta rozā (#E5989B), persiku krāsa (#FFB4A2) un gaiša persiku krāsa (#FFCDB2), šīs krāsas palīdzēs mums programmēt mājas lapas izskatu un dizainu. Lietotāju autorizācija ir nepieciešama, lai rezervācijas var identificēt pēc tālruņa numura, vārda un uzvārda. Seansu lapas sadaļā būtu jāievieto filmas režisorus, pieejamās un aizņemtās sēdvietas, saubtitru un audio valodu.

SR: analizē dažādus ikdienas darba procesus, saskata tajos vai to daļās automatizācijas iespējas.

…

* **Programmatūras prasību specifikācija**

Lai noteiktu, kādas prasības tiek pieprasītas no darbinieka, ko vēlas redzēt mājaslapā, ir jāiziet intervija. Pēc intervijas atbildēm tiks izveidota programmatūras vizuālā modeļa prototips, un entītiju relāciju prototips, kas tiks parādīts (attēli), kādā veidā iespējams izskatīsies mājaslapas galaprodukts.

….

* **Risinājuma mērķauditorijas izvēle un tās raksturojums**

Galvenā mērķauditorija ir kinoteātra darbinieki un kinoteātra apmeklētāji, galvenokārt personas kuras māk rīkoties ar dažādām ierīcēm un/ vai tīmekļa vietnēm, bet būs arī tradicionāls iegādāšanās veids pie kases, ja ir seniori kuri nemāk izmantot elektronikas ierīces.

* **Programmatūras produkta un tā funkciju apraksts**

Kioskā būs iespēja iegādāties biļetes un apmaksāt izvēlētas uzkodas un turpat uz vietas visu apmaksāt.

Kiosku vajadzēs izmantot skārienjūtīgā veidā, kas nozīme ir vajādzīgs noteikts skārien sensors un displejs. Lai veiksmīgi veiktu apmaksu pie kioska, būs jāiebūvē apmaksas termināls ar kuru varēs apmaksāt izmantojot NFC (near field communication), vai tradicionālā veidā - ievietot karti terminālā. Kā arī, lai veiksmīgi noskānēt QR kodu no telefonam bus jāiebūvē noteiks sensors, kurš var noskenēt telefona ekrānu, vai papīra lapu.

* **Saskarsnes modelis**

Pēc dotās programmas prasību specifikācijā, tik izveidots saskarsnes modelis *(skat.1.att),* kas parāda kādā veidā tiks saistītas tīmekļa lapu funkcijas.

A screenshot of a computer

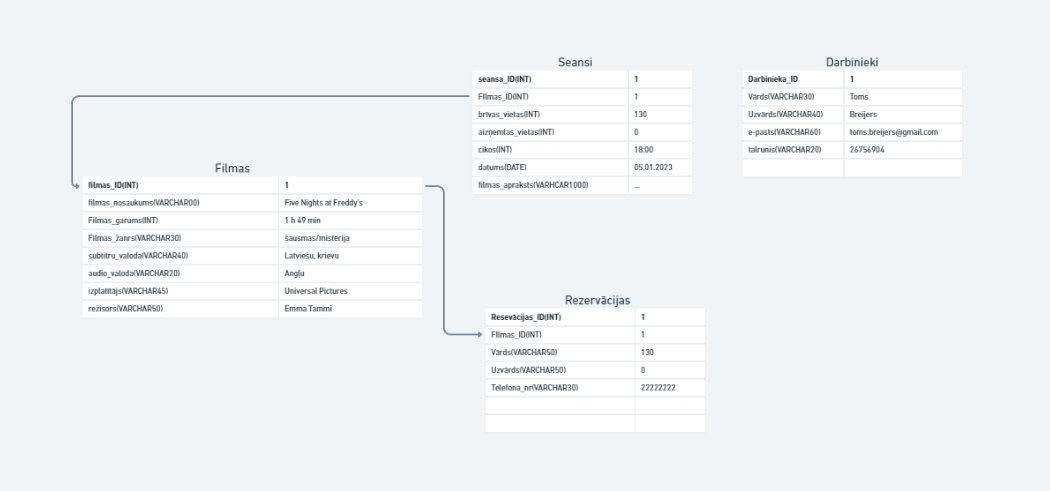
Description automatically generated

*1.att saskarsnes modelis mājaslapas sistēmai*

Saskarsnes modelis shematiski parāda katras lapas funkcijas un kādā secībā lapas aizies no vienas uz otru. Sākumā cilvēki redzēs lapu kura pasveicina lietotāju un aicina doties uz biļešu rezervāciju, kur viņi tiks aizvesti uz lapu kur ir seansu saraksts ar laikiem un brīvajām vietām, ja ir seanss kurš interesē skatītājam tiks pieejama poga lai veikt rezervāciju, pēc izvēlēšanās, lietotājs tiks virzīts uz lapu kur būs jāievada lietotāja informācija kā vārds, uzvārds, mobilais tālrunis un seanss, lai klātienē var identificēt cilvēku, kad rezervācija būs veiksmīga, tad lietotājs būs virzīts uz lapu paziņojot ka rezervācija ir veiksmīga.

Autorizēšanās opcija ir pieejama tikai darbiniekiem, uzspiežot pogu autorizēties, darbinieks tiks pārvirzīts uz lapu kur vajadzēs ievadīt e-pastu un paroli, ja darbinieka informācija eksistē ienākšanas datubāzē, tad tiks pārvirzīts uz lapu, kur var redzēt datubāzi ar rezervācijām uz filmām, visus seansus, visas filmas, kā arī lapas kurās var pievienot jaunas filmas un jaunus seansus.

Kā minēts iepriekš, tīmekļa vietnei arī pastāv datubāze kura apkopo seansus, filmas, un to detaļas, kā, piemēram, filmām tiek norādīts to saukums, garums, audio valoda, subtitru valoda, izplatītājs, režisors un žanrs.

****

*2.att. Datubāze*

* **Programmatūras produkta skice**

Lai kinoteātra darbiniekiem un klientiem būtu vieglāk paskaidrot, ko tie vēlas redzēt tīmekļa lapā, tika izveidota vizuālā skice, kuru var jebkurā brīdī rediģet, lai veiktu labojumus. Tīmekļa vietne tika izveidota pēc kinoteātra logo krāsu estētikas, lai varētu ar acu skatienu atšķīrt, kuram kinoteātrim tas pieder, gala rezultāts vienmēr atšķirsies no skicēm.

A purple background with white text

Description automatically generated

*3.att. Tīmekļa vietnes sākuma lapa*

A purple rectangle with yellow text

Description automatically generated

*4.att. Izvēle starp seansiem.*



*5.att. Rezervācijas veikšanas lapa.*

A screenshot of a video game

Description automatically generated

*6.att. veiksmīgas rezervācijas lapa.*

A screenshot of a computer

Description automatically generated

*7.att. Darbinieku autorizēšanās lapa.*

A screenshot of a computer

Description automatically generated

*8.att. Rezervāciju, datubāzes lapa.*

A screenshot of a video game

Description automatically generated

*9.att. Jaunas filmas pievienošanas lapa.*

A screenshot of a computer

Description automatically generated

*10.att. Jauna Seansa pievienošanas lapa.*

* **Piemērotās licences pamatojums**

Lai izveidotu šāda veida tīmekļa lapu uzņēmumam, būs neipieciešamas daudz aplikācijas kurām ir daudz dažādas iespējas. Tiks izmantots visual studio code, izmantotā programmēšanas valoda būs python, HTML, CSS ar mājas vietnes savienojamību, datubāzes izveidošanai tiks izmantots sqlite3. Visual studio code priekšrocības:

1. Programma pieejama bezmaksas, kas nozīmē to, ka nebūs jāizmanto sava nauda.

2. Visual studio code ir iespējams strādāt ar vairākām programmešanas valodām

3. Programma ir uzticama, jo izmanto lieli uzņēmumi.

4. Programma norāda kurās vietās tika izveidotas kļūdas, kuras var viegli identificēt un novērst, kas palīdzēs ar laika ietaupīšanu.

5. Programma ir pieejama uz visām operētājsistēmam, kas nozīme, ka savu darbu varu turpināt, jeb kad, un jebkur.

Programmas veidošanai tiks izmantota python valoda, kura ir pieejama Visual studio code:

1. Python ir viena no vienkāršākajām valodām ko izmantot, jo tā lielākoties sastāv no angļu vārdiem
2. Python kods ir viegli salasāms un viegli izmantojams
3. Python valoda ir iedeāla lai veiktu dažādus skriptus.
4. Python valodai ir augsta drošība, kas nozīme, uzlaušana būs sarežģīta, citi cilvēki nespēs to rediģēt.
5. HTML tiek izmantots lai izveidotu lapas veideni, kura pēctam tiks rediģēta izmantojot CSS failus, kuri piešķirs katrai lapai savu izskatu.
6. Sqlite3 bibliotēka tiek izmantota, jo tā ir vienkārša bibliotēka ar ko var veidot datubāzes un pielietot mājas lapu funkcionalitāti, kur datubāzi var papildināt caur HTML lapu.

Programmas skices tika veidotas lietojumprogrammā Adobe Photoshop, kaut programma nav bezmaksas, tā ir viegli saprotama un visas veidotās skices saglabājas mākoņ serverī, nodrošinot to, ka skices nepazudīs.

Programmas kods ir bezmakas, to var izmantot un rediģēt ikviens, taču vajag norādīt atsauci uz koda īpašnieku, Artu Višķeri.

SR: ikdienas darba procesos vai to daļās saskata automatizācijas iespējas un to, kā pasūtītājs formulē darba uzdevumu programmatūras izstrādātājam.

SR: sastāda vienkāršotu programmatūras prasību specifikāciju atbilstoši konkrētajam uzdevumam, izvērtējot mērķauditorijas specifiku un vajadzības.

SR: veic vienkāršotu programmatūras projektēšanu (t. sk. lietotāju saskarnes un vienkāršotu datu modeļa izveidi) atbilstoši programmatūras prasību specifikācijā izvirzītajām funkcionālajām un nefunkcionālajām prasībām.

SR: izvēlas programmēšanas valodu un programmatūras izstrādes vidi programmatūras izstrādē atbilstoši uzdevuma specifikai, pamato savu izvēli.

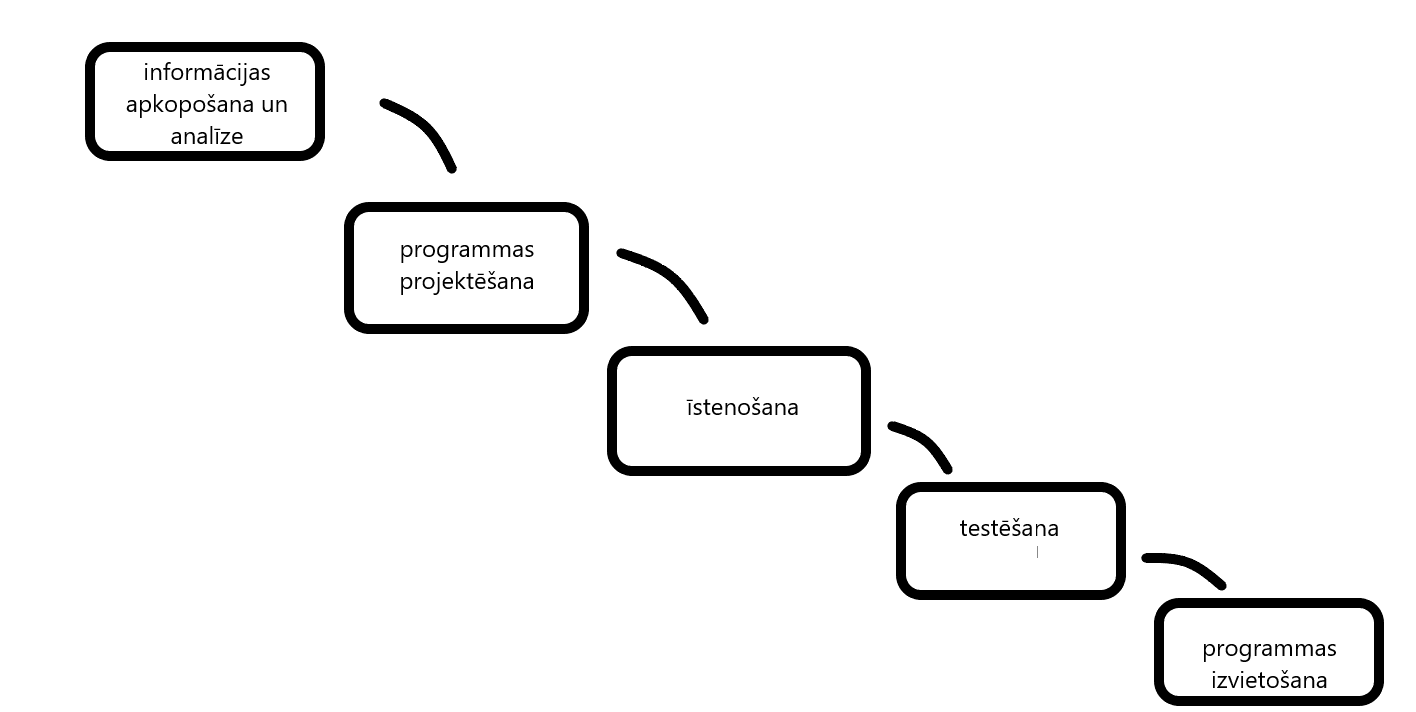
SR: salīdzina atvērto kodu licences un to atšķirības, izmanto un piemēro atbilstošāko no licencēm savam programmatūras projektam.

..

* **Programmatūras izstrādes plāns**

Lai izveidotu kinoteātra uzkaites sistēmu, ir vajadzīgs programmas izveides plāns, kūrā ir norādīts darba plāns un veicamie uzdevumi. Programmatūras izstrādei tika izvēlēta ūdenskrituma izstrādes modelis (waterfall).

Ūdenskrituma modelis tika izvēlēts, jo ūdenskrituma modelis ir diezgan viegli saprotams, elastīgs un ērts. Modelī var viegli saprast kādā secībā notiek visa darba gaita, katram darba gaitas punktam ir darba apraksts, ar darba soļiem.



*12.att. Uzskaites sitēmas ūdenskrituma modeļa skice*

…

SR: salīdzina un izvēlas piemērotāko programmatūras izstrādes modeli konkrētā uzdevuma atrisināšanai, pamato savu izvēli.

SR: veic vienkāršotu programmatūras izstrādes plānošanu (bez darbietilpības novērtējuma).

…

* **Atkļūdošanas un akcepttestēšanas pārskats**

Pēc programmatūras izveides, ir jāveic programmas testēšana, lai pārliecinātos, ka viss strādā kā vajag un drīkt programmu rādīt mērķauditorijai.

Izvēlētais testēšanas paņēmiens bija ‘’Melnās kastes testēšana”, kur testē programmas funkcijas, kādus datus jāievada noteiktajā funkcijā, jāspecificē kādiem rezultātiem ir jābūt, tad pēc testēšanas ir jāievada iegūtais rezultāts, vai testēšana notika veiksmīgi.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Testa nr. | Funkcija | Testa Dati | Sagaidāmais rezultāts | Iegūtais rezultāts | Tests iziets veiksmīgi |
| 1. | Filmu rezervācija | Rezervēt biļetes | ja lietotājs uzspiež uz “rezervēt biļetes”, tad lietotājs tiek pārvirzīts uz seansu lapu (reservetickets) | Atveras lapa kur lietotājs redz seansu sarakstu | veiksmīgs |
| 2. | Rezervācijas funkcionalitāte | Lietotāja iepriekš izvēlētā filma un lietotāja dati | Ja lietotājs ir pārliecināts par izvēli, lietotājs ievada savu informāciju un tiek pārvirzīts uz pēdējo lapu. | Uzpiežot pogu rezervēt, lietotājs tiek pārvirzīts uz ‘rezervācija pabeigta” lapu | Veiksmīgs |
| 3. | Darbinieku autorizēšanās | Darbinieku ieejas dati | Ja darbinieks uzspiež uz pogas “ienākt” pēc e-pasta un paroles ievadīšanas, ja dati ir korekti, darbinieks tiks pārvirzīts uz datubāzes skatu. | Uzspiežot pogu, darbinieks tiek pārvirzīts uz datubāzes skatu. | Veiksmīgs |

Pēc tabulas pētīšanas, drīkst secināt ka programma darbojas korekti un drīkst to prezentēt un pasniegt darbiniekiem, ļaut viņiem to izmēģīnāt un uzzināt vai darbinieki ir apmierināti ar to.

SR: veic programmatūras vienību izstrādi un vienībtestēšanu, izstrādājamās programmatūras vienību apvienošanu, integrācijas un akcepttestēšanu atbilstoši izstrādātajai programmatūras prasību specifikācijai un projektējuma aprakstam.

* **Lietotāja ceļvedis**

Šajam ceļvedim, tiks izmantotas mājas lapu skices un mājaslapu novirzīšanas attribūti.

**Ceļvedis skatītājiem, klientiem**: Ienākot tīmekļa vietnē, lietotājs sastaps sākuma lapu (/) (skat.3att.), kur ir izvēle starp biļešu rezervāciju un autorizēšanos, autorizēšanās ir domāta darbiniekiem. Kad lietotājs uzpiedīs ‘’rezervēt biļeti’’ pogu, tad lietotājs tiks pārvirzīts uz seansu sarakstu /all\_seansi (skat.4.att.). Pēc pogas “rezervēt biļeti” uzspiešanas ir izvēles vienu seansu starp daudziem seansiem, kas varētu interesēt skatītāju, kad lietotājs izvēlas sev tīkamo filmu seansu, uzspiežot pogu rezervēt tas tiek pārvirzīts uz rezervācijas veikšanas lapu /reserveInfo (skat.5.att.). Šajā /reserveInfo lapā ir jāievada sava informācija – vārds, uzvārds, mobilais tālrunis un jāizvēlas sev tīkamais seanss, šī informācija ir vajadzīgā, lai klātienē identificētu skatītāju un veikt samaksu. Ja tīmekļa lapas servisi darbojas korekti, tad lietotāji pēc informācijas apstiprināšanas sastaps lapu, kas pateiks ka rezervācija ir veiksmīga, “/successful”. (skat.6.att.). Nospiežot pogu “atgriezties”, klients tiks pārvests uz sākuma lapu (/) (skat.3.att.). Kā bija redzams, katrā lapā bija opcija “atgriezties”, kas vienmēr pārvedīs uz iepiekšējo lapu, lai pārbaudīt un pārliecinātos par izvēli.

**Ceļvedis darbiniekiem:** lai autorizētos darba vidē, sākuma lapā (/) (skat.3.att) ir jānospiež poga autorizēties, kura pārvirzīs uz lapu /authorize, darbiniekiem būs jāievada savs e-pasts un parole ar kuru tika reģistrēts saitē, ja darbinieka dati eksistē datubāzē, tie tiks pārnesti uz lapu /manage (skat.9.lpp), /manage lapā būs pogas uz kurām ir pateikta viņu funkcija, poga “jauna filma” pārnesīs uz lapu /new\_film (skat.9.att), kur darbinieks var pievienot informāciju par jaunu filmu, poga “saglabāt”, ievietos filmas datu datubāzē, kā parasti ir poga “atpakaļ” kas pārnesīs darbinieku uz /manage lapu, “jauns seanss” pārnesīs uz lapu /new\_seans kur darbinieks var ievadīt informāciju par jaunu seansu, izvēloties kādu no iepriekš pieviennotajām filmām. Tad ir pogas kur var apskatīties visas filmas (/all\_films), rezevācijas (/all\_reserv) un visus seansus (all\_seansi\_authorized). /all\_films Lapā būs visas filmas sarakstā, /all\_reserv būs visas reervācijas sarakstā, /all\_seansi\_authorized būs visi seansi sarakstā. Kad darbinieks ir paveicis savu darbu, tad var klikšķināt uz pogas “iziet”, kas pārvirzīs uz sākuma lapu (/).

SR: izstrādā un prezentē izveidotās programmatūras vienkāršotu izvēršanas (t. sk. ieviešanas) plānu, lietotāja ceļvedi un uzturēšanas plānu, ievērojot tās lietotāju mērķauditorijas specifiku.

…

**Pielikumi**

**1.pielikums. Programmatūras kods:**

https://github.com/tibbyzz/kinoteatra\_sistema

**Main.py**

from flask import Flask, render\_template, request, redirect, url\_for

import sqlite3

app = Flask(\_\_name\_\_, template\_folder='templates')

DATABASE = 'kino.db'

def create\_tables():

  print("Creating tables...")

  conn = sqlite3.connect(DATABASE)

  c = conn.cursor()

  c.execute('''CREATE TABLE IF NOT EXISTS films (

              id INTEGER PRIMARY KEY AUTOINCREMENT,

              film\_name TEXT NOT NULL,

              film\_length TEXT NOT NULL,

              film\_genre TEXT NOT NULL,

              sub\_lang TEXT NOT NULL,

              audio\_lang TEXT NOT NULL,

              distributor TEXT NOT NULL,

              director TEXT NOT NULL,

              film\_description TEXT NOT NULL)''')

  c.execute('''CREATE TABLE IF NOT EXISTS seansi (

              id INTEGER PRIMARY KEY AUTOINCREMENT,

              films\_id INTEGER,

              available\_seats INTEGER,

              taken\_seats INTEGER,

              time TEXT NOT NULL,

              date TEXT NOT NULL,

              FOREIGN KEY (films\_id) REFERENCES films(id))''')

  c.execute('''CREATE TABLE IF NOT EXISTS workers(

              id INTEGER PRIMARY KEY AUTOINCREMENT,

              name TEXT NOT NULL,

              last\_name TEXT NOT NULL,

              email TEXT NOT NULL,

              password TEXT NOT NULL,

              mobile\_number TEXT NOT NULL)''')

  c.execute('''CREATE TABLE IF NOT EXISTS reservations (

              id INTEGER PRIMARY KEY AUTOINCREMENT,

              films\_id INTEGER,

              name\_r TEXT NOT NULL,

              last\_name\_r TEXT NOT NULL,

              mobile\_number\_r TEXT NOT NULL,

              FOREIGN KEY (films\_id) REFERENCES films(id)) ''')

create\_tables()

#1.page

@app.route('/')

def start\_page():

  return render\_template('start\_page.html')

#2.page

def get\_all\_films():

  conn = sqlite3.connect(DATABASE)

  c = conn.cursor()

  c.execute(

      "SELECT id, film\_name, film\_length, film\_genre, sub\_lang, audio\_lang, distributor, director FROM films"

  )

  films = c.fetchall()

  conn.close()

  return films

@app.route('/all\_seansi')

def all\_seansi():

  sensi = get\_all\_seansi()

  return render\_template('all\_seansi.html', seansi=sensi)

@app.route('/all\_seansi\_authorized')

def all\_seansi\_authorized():

  sensi = get\_all\_seansi()

  return render\_template('all\_seansi\_authorized.html', seansi=sensi)

#3.page

def get\_film\_desc():

  conn = sqlite3.connect(DATABASE)

  c = conn.cursor()

  c.execute("SELECT id, film\_description FROM films")

  films = c.fetchall()

  conn.close

  return films

#4.page

@app.route('/reserveInfo', methods=['GET', 'POST'])

def reserveInfo():

  if request.method == 'POST':

    name\_r = request.form['name\_r']

    last\_name\_r = request.form['last\_name\_r']

    mobile\_number\_r = request.form['mobile\_number\_r']

    films\_id = request.form['films\_id']

    conn = sqlite3.connect(DATABASE)

    c = conn.cursor()

    c.execute(

        "INSERT INTO reservations (name\_r,last\_name\_r,mobile\_number\_r,films\_id) VALUES (?,?,?,?)",

        (name\_r, last\_name\_r, mobile\_number\_r, films\_id))

    conn.commit()

    conn.close()

    print(name\_r, last\_name\_r, mobile\_number\_r, films\_id)

    return redirect("/successful")

  else:

    films = get\_all\_films()

  return render\_template("reserveInfo.html", films=films)

#5.page

@app.route('/successful')

def successful():

  return render\_template('successful.html')

#6.page

@app.route('/authorize', methods=['GET', 'POST'])

def authorize():

  if request.method == 'POST':

    email = request.form.get('email')

    password = request.form.get('password')

    if check\_credential(email, password):

      return redirect(url\_for('manage\_page'))

    else:

      return redirect(url\_for('authorize'))

  else:

    return render\_template("authorize.html")

#7page

@app.route('/manage')

def manage\_page():

  return render\_template("manage.html")

#8.page

@app.route('/new\_film', methods=['GET', 'POST'])

def new\_film():

  if request.method == 'POST':

    film\_name = request.form['film\_name']

    film\_length = request.form['film\_length']

    film\_genre = request.form['film\_genre']

    sub\_lang = request.form['sub\_lang']

    audio\_lang = request.form['audio\_lang']

    distributor = request.form['distributor']

    director = request.form['director']

    film\_description = request.form['film\_description']

    conn = sqlite3.connect(DATABASE)

    c = conn.cursor()

    c.execute(

        "INSERT INTO films (film\_name, film\_length, film\_genre, sub\_lang, audio\_lang, distributor, director, film\_description) VALUES (?,?,?,?,?,?,?,?)",

        (film\_name, film\_length, film\_genre, sub\_lang, audio\_lang, distributor,

         director, film\_description))

    conn.commit()

    conn.close()

    print(film\_name, film\_length, film\_genre, sub\_lang, audio\_lang,

          distributor, director, film\_description)

    return redirect("/manage")

  return render\_template("new\_film.html")

@app.route('/new\_seans', methods=['GET', 'POST'])

def new\_seans():

  if request.method == 'POST':

    conn = sqlite3.connect(DATABASE)

    c = conn.cursor()

    films\_id = request.form['films\_id']

    available\_seats = request.form['available\_seats']

    taken\_seats = request.form['taken\_seats']

    time = request.form['time']

    date = request.form['date']

    c.execute(

        "INSERT INTO seansi (films\_id, available\_seats, taken\_seats, time, date) VALUES (?,?,?,?,?)",

        (films\_id, available\_seats, taken\_seats, time, date))

    conn.commit()

    conn.close()

    return redirect("/manage")

  else:

    films = get\_all\_films()

    return render\_template("new\_seans.html", films=films)

#9.page

def get\_all\_films():

  conn = sqlite3.connect(DATABASE)

  c = conn.cursor()

  c.execute("SELECT \* FROM films")

  films = c.fetchall()

  conn.close()

  return films

@app.route('/all\_films')

def all\_films():

  films = get\_all\_films()

  return render\_template('all\_films.html', films=films)

#10.page

def get\_all\_reservations():

  conn = sqlite3.connect(DATABASE)

  c = conn.cursor()

  c.execute('''SELECT f.film\_name, r.name\_r, r.last\_name\_r, r.mobile\_number\_r

            FROM reservations r

            JOIN films f ON r.films\_id=f.id''')

  reservations = c.fetchall()

  conn.close()

  return reservations

@app.route('/all\_reserv')

def all\_reserv():

  reservations = get\_all\_reservations()

  return render\_template('all\_reserv.html', reservations=reservations)

#11.page

def get\_all\_seansi():

  conn = sqlite3.connect(DATABASE)

  c = conn.cursor()

  c.execute(

      '''SELECT f.film\_name, s.available\_seats, s.taken\_seats, s.time, s.date, f.film\_description

            FROM seansi s

            JOIN films f ON s.films\_id=f.id ''')

  seansi = c.fetchall()

  conn.close()

  return seansi

@app.route('/seansi')

def seansi():

  seansi = get\_all\_seansi()

  return render\_template('seansi.html', seansi=seansi)

@app.route('/register\_worker', methods=['GET', 'POST'])

def register\_employee():

  if request.method == 'POST':

    mail = request.form['email']

    password = request.form['password']

    name = request.form['workerName']

    lastName = request.form['workerLastName']

    mobile = request.form['workerMobile']

    conn = sqlite3.connect(DATABASE)

    c = conn.cursor()

    c.execute(

        "INSERT INTO workers (name, last\_name, email, password, mobile\_number) "

        "VALUES (?,?,?,?,?)", (name, lastName, mail, password, mobile))

    conn.commit()

    conn.close()

  return render\_template('register\_worker.html')

def check\_credential(email, password):

  conn = sqlite3.connect(DATABASE)

  c = conn.cursor()

  c.execute("SELECT \* FROM workers")

  all\_rows = c.fetchall()

  for row in all\_rows:

    print(row)

  c.execute("SELECT \* FROM workers WHERE email=? AND password=?",

            (email, password))

  result = c.fetchone()

  conn.close()

  print(result is not None)

  return result is not None

if \_\_name\_\_ == '\_\_main\_\_':

  app.run(debug=True, host='0.0.0.0')

**2.pielikums. Programmatūras struktūrskice:**

**https://whimsical.com/kinoteatra-uzskaites-sistema-L9wrSk8BwhK7aHnuhgjd6u**

**Minimālās prasības programmatūras produktam:**

**1. Pielieto datu bāzi ar vairākām tabulām;**

SR: 2.3.2. Plāno datubāzi, t. sk. izveido ER modeli konkrētā uzdevuma datu apstrādes risinājumam.

SR: 2.4.17. Izveido vienkāršu datu apstrādes programmatūru (sistēmu), datu uzglabāšanai izmantojot paša veidotu datubāzi ar vairākām tabulām.

**2. Pielieto vismaz vienu bibliotēku un/vai API;**

SR: 2.4.10. Izmanto programmēšanas valodas un tās bibliotēku dokumentāciju un palīdzības sistēmu, lai patstāvīgi apgūtu citas to piedāvātās iespējas, kas nepieciešamas konkrētās programmatūras izstrādei.

SR: 2.4.11. Meklē un pievieno atvērtā koda bibliotēkas un lieto API (programmsaskarni) specializētu funkciju veikšanai sava programmēšanas projekta īstenošanai.

SR: 2.4.13. Izvēlas un lieto atbilstošas programmēšanas valodas konstrukcijas, datu tipus un dažādas bibliotēkas, veidojot programmas doto uzdevumu un problēmu risinājumam.

**3. Pielieto vismaz vienu no minētajām datu struktūrām;**

SR: 2.4.14. Izmanto dažādas datu struktūras (t. sk. masīvi, kopas, ieraksti, steks, rinda, saraksts, koks, grafs, datne) un ar tiem saistītos pamatalgoritmus.

SR: 2.4.19. Veido dotā uzdevuma (problēmas) risinājumu, izmantojot gatavus algoritmus un/vai pielāgojot vai kombinējot tos, un/vai izstrādājot jaunus algoritmus. Izprot un skaidro dažādu algoritmu darbību, pielāgo tos dažādām nestandarta situācijām, ja nepieciešams, veidojot jaunas datu struktūras.

**4. Ir realizēta lietotāja piekļuves vai datu aizsardzība.**

SR: 3.1.2. Izmanto kriptogrāfijas metodes konkrētā uzdevuma risinājumā.

**Programmēšanas labās prakses principu snieguma līmeņu apraksts**

Labās prakses principi ir:

* katru atsevišķu priekšrakstu raksta jaunā rindā;
* koda loģiskās daļas (piemēram, zarošanos, ciklu, masīvu) savstarpēji atdala ar tukšu rindu;
* lieto atkāpes, lai vizualizētu priekšrakstu vai struktūru iekļaušanu citās struktūrās;
* izvairās no garām koda rindām. Lieto pārnesi jaunā rindā, atvieglojot koda lasīšanu;
* mainīgo, funkciju u. c. nosaukumus veido jēgpilnus, atvieglojot koda uztveri (piemēram, perimetru apzīmējot nevis ar “a”, bet “perim” vai “perimetru”);
* ar komentāriem skaidro programmatūras koda loģiskās daļas (piemēram, zarošanos, ciklu, masīvu), to lomu programmatūrā (piemēram “Datu izvade”).