CINEMARESERVE

Informační systém pro kino

BI-SI1

Tým C

Návrh architektury - technologie

- Python Flask
- React
- PostgreSQL
- SQLAlchemy

Návrh architektury - diagram balíčků

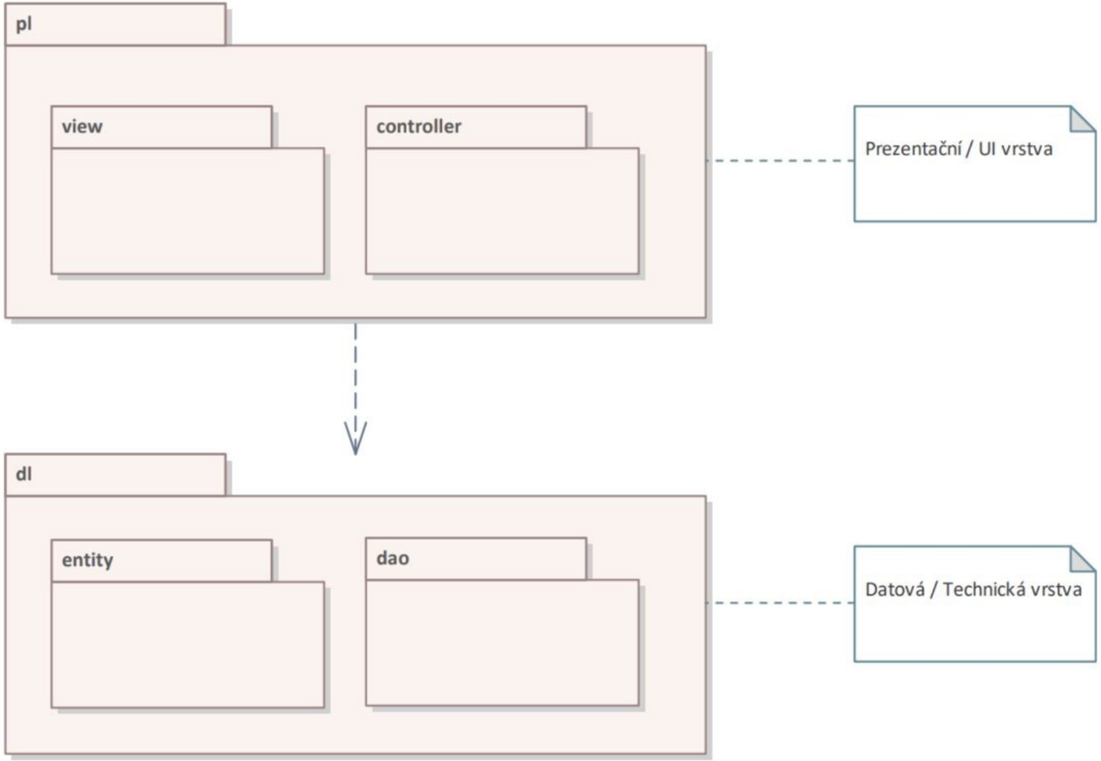
- pl
    - view

      = prezentace dat uživateli
    - controller

      = zpracovávání vstupů
- dl
    - entity

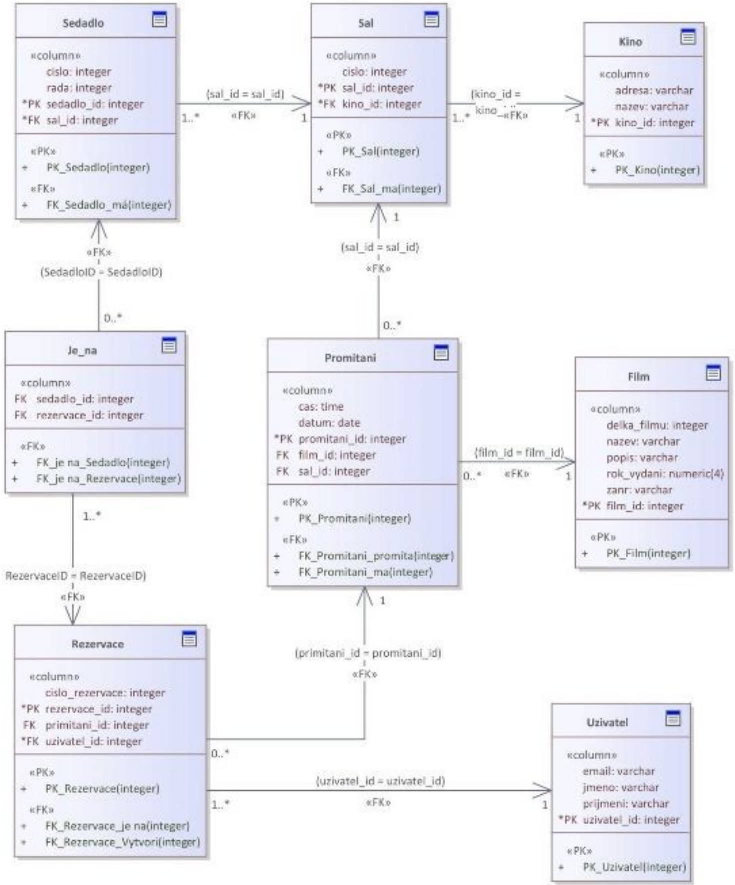
      = definice datových objektů
    - dao

      = rozhraní k načítání/ukládání dat



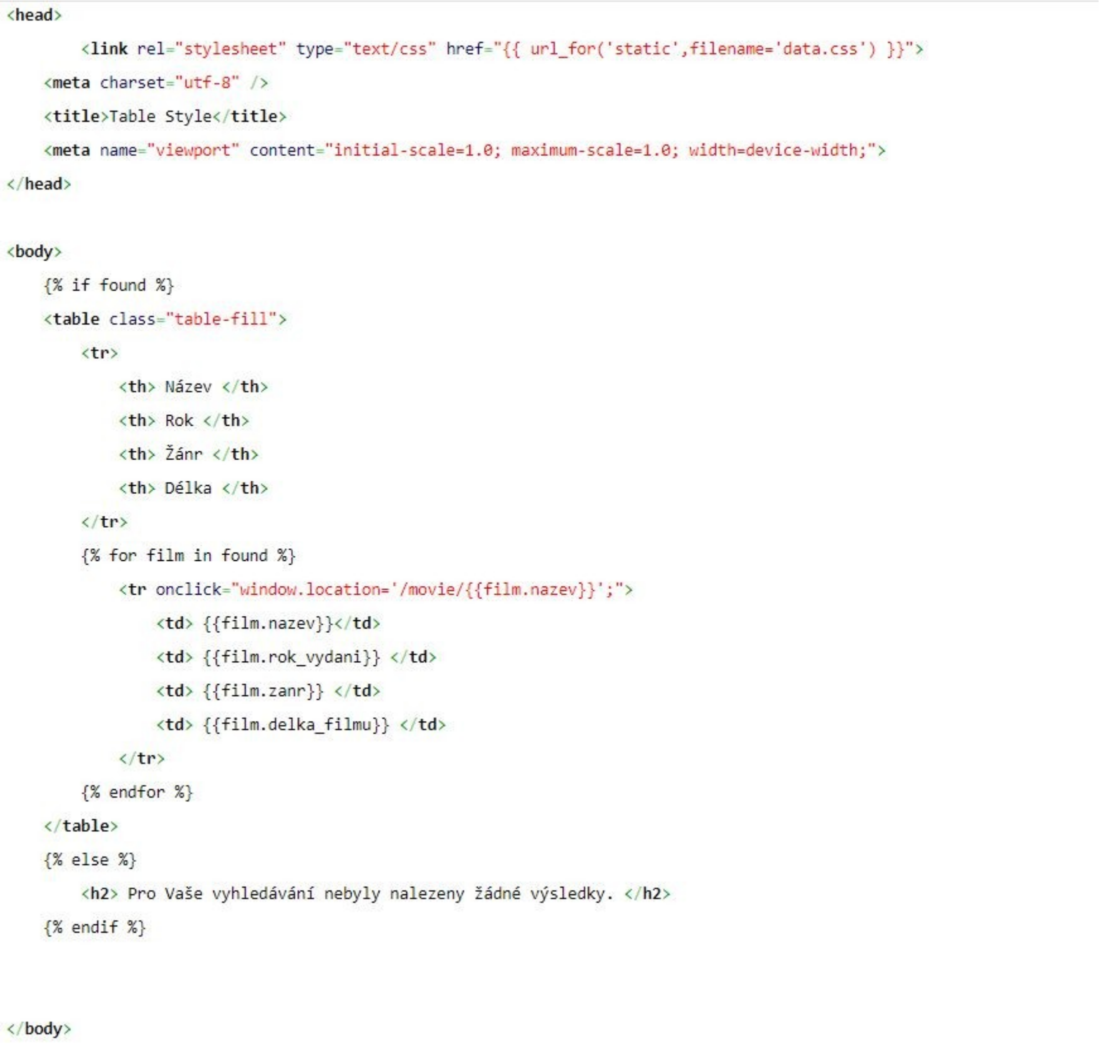
Databázový model

- Promítání a rezervace

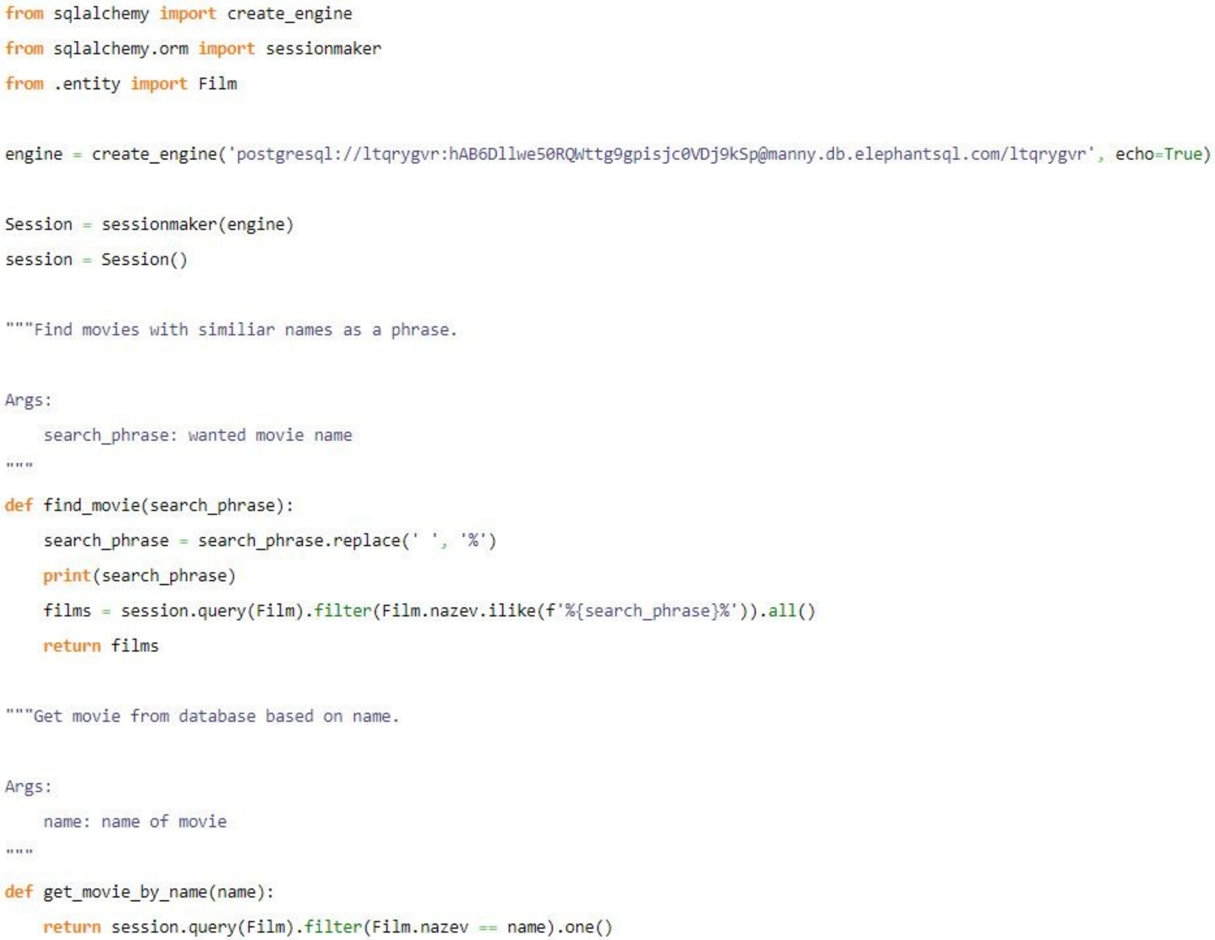


CINEMARESERVE

Ukázka kódu



```
<head>
        <link rel="stylesheet" type="text/css" href="{{ url_for('static',filename='data.css') }}">
    <meta charset="utf-8" />
    <title>Table Style</title>
    <meta name="viewport" content="initial-scale=1.0; maximum-scale=1.0; width=device-width;">
</head>

<body>
    {% if found %}
    <table class="table-fill">
        <tr>
            <th> Název </th>
            <th> Rok </th>
            <th> Žánr </th>
            <th> Délka </th>
        </tr>
        {% for film in found %}
            <tr onclick="window.location='/movie/{{film.nazev}}';">
                <td> {{film.nazev}}</td>
                <td> {{film.rok_vydani}} </td>
                <td> {{film.zanr}} </td>
                <td> {{film.delka_filmu}} </td>
            </tr>
        {% endfor %}
    </table>
    {% else %}
        <h2> Pro Vaše vyhledávání nebyly nalezeny žádné výsledky. </h2>
    {% endif %}


</body>
```



```python
from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
from .entity import Film

engine = create_engine('postgresql://ltqrygvr:hAB6Dllwe50RQWttg9gpisjc0VDj9kSp@manny.db.elephantsql.com/ltqrygvr', echo=True)

Session = sessionmaker(engine)
session = Session()

"""Find movies with similiar names as a phrase.

Args:
    search_phrase: wanted movie name
"""
def find_movie(search_phrase):
    search_phrase = search_phrase.replace(' ', '%')
    print(search_phrase)
    films = session.query(Film).filter(Film.nazev.ilike(f'%{search_phrase}%')).all()
    return films

"""Get movie from database based on name.

Args:
    name: name of movie
"""
def get_movie_by_name(name):
    return session.query(Film).filter(Film.nazev == name).one()
```

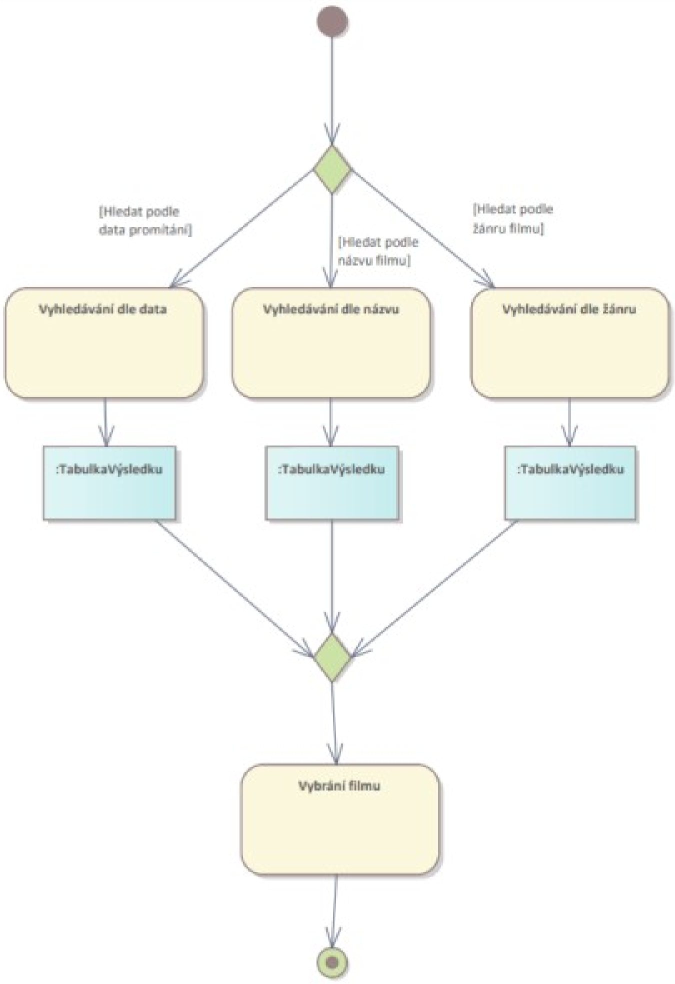
CINEMARESERVE

CINEMARESERVE

```python
# coding: utf-8
from sqlalchemy import Column, Date, ForeignKey, Integer, Numeric, String, Table, Time
from sqlalchemy.orm import relationship
from sqlalchemy.ext.declarative import declarative_base

Base = declarative_base()
metadata = Base.metadata


class Film(Base):
    __tablename__ = 'Film'

    delka_filmu = Column(Integer)
    nazev = Column(String)
    popis = Column(String)
    rok_vydani = Column(Numeric(4, 0))
    zanr = Column(String)
    film_id = Column(Integer, primary_key=True)

class Kino(Base):
    __tablename__ = 'Kino'

    adresa = Column(String)
    nazev = Column(String)
    kino_id = Column(Integer, primary_key=True)


class Uzivatel(Base):
    __tablename__ = 'Uzivatel'

    email = Column(String)
    jmeno = Column(String)
    prijmeni = Column(String)
    uzivatel_id = Column(Integer, primary_key=True)
```

```python
class Sal(Base):
    __tablename__ = 'Sal'

    cislo = Column(Integer)
    sal_id = Column(Integer, primary_key=True)
    kino_id = Column(ForeignKey('Kino.kino_id'), nullable=False)

    kino = relationship('Kino')


class Promitani(Base):
    __tablename__ = 'Promitani'

    cas = Column(Time)
    datum = Column(Date)
    promitani_id = Column(Integer, primary_key=True)
    film_id = Column(ForeignKey('Film.film_id'))
    sal_id = Column(ForeignKey('Sal.sal_id'))

    film = relationship('Film')
    sal = relationship('Sal')
```

```python
class Sedadlo(Base):
    __tablename__ = 'Sedadlo'

    cislo = Column(Integer)
    rada = Column(Integer)
    sedadlo_id = Column(Integer, primary_key=True)
    sal_id = Column(ForeignKey('Sal.sal_id'), nullable=False)

    sal = relationship('Sal')


class Rezervace(Base):
    __tablename__ = 'Rezervace'

    cislo_rezervace = Column(Integer)
    rezervace_id = Column(Integer, primary_key=True)
    primitani_id = Column(ForeignKey('Promitani.promitani_id'))
    uzivatel_id = Column(ForeignKey('Uzivatel.uzivatel_id'), nullable=False)

    primitani = relationship('Promitani')
    uzivatel = relationship('Uzivatel')
    sedadlos = relationship('Sedadlo', secondary='Je_na')


t_Je_na = Table(
    'Je_na', metadata,
    Column('sedadlo_id', ForeignKey('Sedadlo.sedadlo_id')),
    Column('rezervace_id', ForeignKey('Rezervace.rezervace_id'))
)
```



CINEMARESERVE

CINEMARESERVE

Děkujeme za pozornost.
Dotazy?