**Nume:**

Lucaci Cristiana – baza cunostinte
Mihalachi Razvan - Constantin - inferenta
Mihutescu Monica - interfata

**Echipa** : M12_2

## Proiect SE - Amon

### Descrierea aplicatiei

Aceasta aplicatie din domeniul culinar are ca scop facilitarea unei alegeri mai bune a restaurantului in care sa manance in functie de zona din Timisoara, de rating, de specificul restaurantului, de meniu si pretul mancarii.

### Functionalitate

Cand un utilizator intra in aplicatie va avea la dispozitie mai multe alegeri pe care trebuie sa le faca pentru a i se recomanda cea mai buna varianta de restaurant in functie de preferinte.

Pentru alegerea zonei, va fi afisata o lista cu zonele orasului, iar utilizatorul va avea posibilitatea sa aleaga o singura varianta.

In urmatorul pas utilizatorul va putea sa isi aleaga atat specificul restaurantului cat si felurile de mancare prin selectarea a mai multor optiuni preferate.

Ratingul si preturile vor putea fi stabilite prin definirea unui interval.

### Tinta

Aplicatia se adreseaza turistilor care aleg ca destinatie orasul Timisoara si sunt in cautarea unui loc in care sa manance.

### Tehnologii:

Metoda: Forward Chaining
Java, JavaFX, JSON

### Baza de cunostiinte:

- Setul de date de intrare /* citire din interfata*/
  - **initial_premises**([travels_by_train, is_in_a_hurry, is_visiting_the_center, walking_by_the_river, wants_romantic_sitting, is_shopping, wanting_a_business_environment, eating_with_friends_and_family, , is_vegetarian, is_smoking, likes_traditional_food, likes_italian_food,

likes_chinese_food, likes_greek_food, likes_american_food, looking_for_smaller_prices]).

- Setul de premise

  //reguli intermediare pentru concluzii partiale

  - rule([travels_by_train, is_in_a_hurry], near_train_station).
  - rule([walking_by_the_river, is_in_a_hurry], near_bega).
  - rule([is_shopping, is_in_a_hurry], near_mall).
  - rule(is_visiting_the_center, is_in_a_hurry], near_city_center).

  //reguli de scop, reguli finale

  - rule([near_train_station, eating_with_friends_and_family, likes_traditional_food, likes_italian_food, looking_for_smaller_prices, is_smoking], Thalia).
  - rule([near_train_station, wants_romantic_sitting, is_vegetarian, likes_italian_food], Casa_Iosefin).
  - rule([near_train_station, wanting_a_business_environment, likes_traditional_food, is_smoking], Trattoriaccia).
  - rule([near_city_center, wanting_a_business_environment, is_vegetarian, likes_italian_food], Locanda_Del_Corso).
  - rule([near_city_center, eating_with_friends_and_family, likes_greek_food, looking_for_smaller_prices], Pepper_Steak_Shake).
  - rule([near_city_center, wants_romantic_sitting, is_vegetarian, likes_american_food], Neata_Omlette_Bistro).
  - rule([near_mall, eating_with_friends_and_family, likes_traditional_food, is_smoking], Manufaktura).
  - rule([near_mall, wanting_a_business_environment, ], Fabrika)
- Setul de concluzii
  - restaurant(Thalia)
  - restaurant(Casa_Iosefin)
  - restaurant(Trattoriaccia)
  - restaurant(Locanda_Del_Corso)
  - restaurant(Pepper_Steak_Shake)
  - restaurant(Neata_Omlette_Bistro)
  - restaurant(Manufaktura)
  - restaurant(Fabrika)
  - restaurant(Riviere_Brasserie)
  - restaurant(Gradina_Banateana)

**Zone restaurante**:

- Langa gara ( near_train_station )
    - restaurant(Thalia)
    - restaurant(Casa_Iosefin)

      restaurant(Trattoriaccia)
- Centrul orasului ( near_city_center )
    - restaurant(Locanda_Del_Corso)
    - restaurant(Pepper_Steak_Shake)
    - restaurant(Neata_Omlette_Bistro)
- Langa mall ( near_mall )
    - restaurant(Manufaktura)
    - restaurant(Fabrika)
- Pe malul Begai ( near_bega )
    - restaurant(Riviere_Brasserie)
    - restaurant(Gradina_Banateana)

**Exemplu Scenariu**:

- When a user opens the app he is prompted with a welcome message ("Hello! Are you visiting Timisoara and don't know where to eat?"). The button "Get Help" will redirect him to another window view
- After clicking the button, the message *"In order to give you a suggestion where to eat, we first need to know a little more about your visit to Timisoara and your preferences."* will appear. Now the user must answer a set of questions:

    - Are you in a hurry? Yes No
    - Right now, you are:

      at the train station in Timisoara ○

      just finishing a shopping session at the mall ○

      visiting the city center ○

      walking by the Bega River ○

      none of the above ○
    - Are you going to have:

      a romantic date ○

      a business meeting ○

a friends and family meal ○

- You like:

  Romanian traditional dishes □

  Italian food □

  Chinese food □

  Greek food □

  American food □

- Are you a vegetarian? Yes No
- Are you smoking? Yes No
- Are you looking for smaller prices? Yes No

- After answering all the questions, the application will suggest a restaurant

○ - single choice

□ - multiple choice

**Code Inference** :

- model :
  - KnowledgeBase ( List<Rule> rules , List<String> fact )
  - Rule ( List<String> premises , String head )

- parser :
  - KB.json ( aici sunt descrise regulile care au formatul , o lista de premise si o concluzie )
  - Data (void getData() : sunt preluate datele din fisierul KB urmand dupa , pentru fiecare informatie sa se creeze un obiect Rule , iar la final respectivul obiect este introdus in KnowledgeBase la parametrul rules )

- Inference :
  - Conclusion ( List<String> allConclusions , void initialise() : se parcurge lista de “rules” din KnowladgeBase , iar daca concluzie incepe cu litera mare inseamna ca respectiva concluzie este una finala si nu una partiala , urmand ca apoi sa fie adaugata in lista “allConclusions” )
  - ForwardChaining ( String result , void generateConclusion() : initial , se initializeaza un vector “count” de tip int cu numarul de premise pentru

fiecare regula , acesta are scopul de a controla daca o premisa se regaseste intr-o regula sau nu . Dupa , dorim ca pentru fiecare premisa sa aflam la cate reguli se potriveste, odata gasita la o regula k , scadem cu –1 la count[k] . Astfel , daca o regula k are count[k] = 0 , inseamna ca au fost parcursi fiecare premisa din regula . In final se verifica daca concluzia la o regula este una finala sau partiala , daca este finala ne oprim , altfel , concluzia partiala este adaugat in lista de premise , urmand sa parcurgem din nou ciclul de verificare .

**Concluzie:**

Una dintre problemele cu care ne-am confruntat a fost selectarea si deselectarea raspunsurilor la intrebari si baza de date care trebuie sa aiba optiuni pentru toate variantele de selectare. Ne-am limitat la un numar destul de mic de restaurante si intrebari.

O imbunatatire ar putea fi salvarea optiunilor initiale selectate de utilizator pe care sa le poata modifica in loc sa completeze din nou tot formularul.