SBNZ - CityExplorer

**Author:** Arpad Varga Šomođi SW-35/2016

**Project name:** CityExplorer

An application to discover various activites you can do in and around of Novi Sad

**Motivation**

The various restrictions in our lives because of the Covid-19 pandemic forced us into our homes and restricts our ability to travel, socialise and to discover new experiences. When the pandemic calms down a lot of people will want to go out, however, many won’t want to travel abroad for adventures until it’s fully safe, so they’ll be more interested in discovering recreational opportunities in their own hometown.

**Problem overview**

The application solves the problem of finding the best activities in Novi Sad that are in line with the occasion, needs and capabilities of users. When searching for ‘activities in Novi Sad’ on Google Search, the result are mostly articles for tourist which are based on the writers own taste, so you need to read through and research a lot to find something new that interests you. Existing solutions, like TripAdvisor[1] bases their recommendation on the number of and average of ratings from all users, or impressions from the places the user has already visited and suggest similar ones without taking into account the users specific needs.

**Methodology**

3 main type of users: Administrator, Registered user and basic(unregistered) user.

Funcionalities by roles are the following:

Basic user:

* view all activities
* search by name and description
* view activity profile
* register/login

Registered user:

* get a recommended activity/place by preferences
* rate activities
* view profile
* everything that the basic user can do

Administrator:

* add new activity/place
* view profile
* view detailed statistics about activities ( ratings distribution)
* view reports about: 1. Most and least recommended activities 2. Activities ranged by rating average(example: min:2, max:3.5) 3. Most satisfied users 4. Least satiesfied users

**Expected input**:

* Price ( Free | Cheap | Moderate | Expensive ) - multi choice
* Companions ( Alone | Partner | Family | Colleagues | Friends )
* Number of people (not showing if ’Alone” is chosen at Companions)
* Transportation ( By foot | Auto | Bus )
* Special occasion ( Yes | No )
* Theme ( Nature | Adrenaline | History | Sport)

Expected output je place/activity which best suits the users expectations.

Model of the most important characteristics of the activities are the following

* Location ( City Center | Suburbs | Outside the city )
* Keywords ( Romantic | Chill | Adventure | TeamBuilding | Luxury | Family Friendly | Educational | Nature | Sport | History | Adrenaline )
* Space ( Small | Medium | Large )
* Outdoor (Yes | No)
* Other: Bus station nearby, parking , reservation available, program for children, wifi, tv

Example of a recommendation:

Inputs: Moderate|Expensive price, with colleagues, 5 people, has car, not a special occasion and searching for Adrenaline

Output: Go-karting at “Ns Carting”

Technologies used: Spring Boot backend, Angular frontend, Drools for rule based system and JUnit for testing rules.

**Examples of rules**

Rules have different priorities:

* Saliance 100
* Saliance 80
* Saliance 60
* If user chose he’s going with Family, the rule determines the keyword Family friendly
  + Family Friendly keyword activates the following characteristic : program for children
    - Points are added to activities that meet the needs of the previous rules
* If user goes with partner and it is a special occasion, the rule will determine a romantic activity is needed
  + - Giving points for activities with Romantic and Luxury keywords
* If user goes with colleagues and it is a special occasion, activities with the keyword TeamBuilding are preferred
  + After activation of rule for teambuilding, activates the rule to add characteristic reservation available
    - Giving extra point for activities with the preferred characteristics
* If user goes with colleagues without a special occasion, activities with keyword “Luxury” are preferred
  + - Points are given to luxurious activities
* If user goes alone or with friends without a special occasion, activities with keyword “Chill” are preferred
  + Chill keyword activates the following characteristic : wifi and tv
    - Points are added to activities that meet the needs of the previous rules
* If current date is spring or summer, outdoor activities are more diserable
  + - Points are added to activities which are outdoors
* If user as transportation chose ‘By foot’, activities in the City center are preferred
  + - Points are given for activities in the city
    - A few points taken for activities in suburbs
    - More points taken for activities outside of the city
* If user as transportation chose ‘Bus’, activities close to Bus stations are preferred
  + - Points are given to those activities
* If user has a Car, more desirable are activities with parking nearby
  + - Points are given to those activities
* When a bigger group is coming (more than 5 people), Medium or Large space is needed
  + - Point are added for larger activities
* Based on the user’s entries for price and theme, points are added (more than in previous cases because these are important factors) to activities with these characteristics
* If an activity was already recommended to the user (exists in users recommendation history) a bigger number of points is deducted, because the point is to discover NEW things

**Queries** will be used to get the average rating of activities, the total number of recommendations or the rating distribution, the result wil be taken from QueryResults.

**Templates** will be used when admin searches all activities with a given rating in between a given interval (minRating, maxRating)

**Complex event processing examples:**

1. Every time a user tries to log in with the wrong credentials a BadCredentialsEvent is created. If the same user tries to log in with the wrong password 10 times in 3 minutes a SuspicousLoginEvent tis activated and a warning is passed to the admin at the console. If the user tries to log in 10 times in a minute, the user is is deactivated.

2. Every time the user rates something an event is created.If the average rating of an activity is below 3 and a minimum of 4 people rated that activity, BadRatingAlarm is created and a warning is shown that the rating for that activity is very bad.

3. If a user gives five or more 1 star ratings in a minute, it means he may try to sabotage the site with constant negative review, so a ReviewBombAlarm is created and a warning is written on the console

**Comparison with examples for grades:**

*These rules are for demonstration only, they are not intended to use correct Drools syntax(for now)*

forward-chaining example:

1.Rule : “Going with family”

saliance 100

when

$e: ExpectationsDTO( companions == “Family”) ;

$r: ActivityRequirements();

then

add keyword Family friendly to the requirements ($r)

2.Rule “Family friendly activity”

saliance 80

when

$r: ActivityRequirements( keywords contains Keywords.FAMILY\_FRIENDLY);

then

add to features of $r Features.CHILDRENS\_PROGRAMS

3. Rule - Calculating score for each activity based on the requirements

4. ***This rule’s RHS only executes once, the best activity with biggest saliance(score)***

global Activity best;

declare FoundBest

end

Rule “Find best activities”

saliance - dinamically set by finding the score for each activity

saliance $score

when

$requirements: ActivityRequirements();

$service: ActivityService();

$activity: $Activity($name:name, $score: score);

not FoundBest()

then

set best activity, insert FoundBest, etc.

(+1) rule “reset activity scores” with saliance -100 -> get everything back to 0 at last

*For grade 9:* - reports for admin – find least satisfied users

rule “User in the last 2 months giving at least 4 bad ratings for at least 8 recommendations”

when

$user: RegisteredUser( $history : recommendedActivities, recommendedActivities.size >= 8)

Accumulate all distinct activities from history, collecting to a list those which the user in the last 2 months gave a worse rating then 2.5, and counting if list size >=4

then

add user to dissatisfied user, write to console, etc...

For grade 10: -CEP and queries

rule “10 unsuccesful login in 1 minute from a user”

when

cought 10 BadCredentialsEvent over window:time(1m)

then

write warning to console, block user

**Literature**:

* Tripadvisor – Attractions in Novi Sad: <https://www.tripadvisor.com/Attractions-g295380-Activities-Novi_Sad_Vojvodina.html>