**Міністерство освіти і науки України**

**Чернівецький національний університет**

**імені Юрія Федьковича**

***Звіт***

про виконання лабораторної роботи № 3

з курсу «Інтелектуальні системи прийняття рішень»

Тема:

**Створення продукційної моделі експертної системи**

Виконав:

студент 541 групи

Попелюк Михайло

Чернівці 2022

**Мета роботи**: Проектування експертних систем в заданій предметній області

за допомогою продукційної моделі і представлення правил у вигляді дерева.

**Завдання**:

1. Розглянути приклади експертних систем, запропонованих викладачем.
2. У предметній області розробити експертну систему за допомогою продукційної моделі. База знань повинна обов'язково включати кілька рівнів міркувань (тобто остаточні висновки не повинні безпосередньо випливати з комбінацій вхідних даних, обов'язкова присутність проміжних висновків) і демонструвати деяку інтелектуальність в прийнятті рішень. Орієнтовна кількість правил «якщо – то» - близько 30.
3. Для розробленого набору правил побудувати діаграму залежностей у вигляді дерева.
4. Оформити звіт по роботі:
5. Титульний лист,
6. Короткий опис предметної області і задачі, яка розв’язується (який результат роботи системи, що є вхідними даними),
7. Розроблений набір правил експертної системи,
8. Докладні висновки.

**Хід роботи**

Я розглянув приклади експертних систем, запропонованих викладачем і вирішив реалізувати систему під номером 10 «Визначення місця відпочинку для майбутньої відпустки».

Дана ЕС повинна допомогти користувачу обрати місце наступної відпустки виходячи із вхідних даних. Також, під час роботи програми є можливість показати список всіх варіантів відпустки у рейтинговому порядку та їх додступність, базуючись на відповідях.

Оскільки база знань повинна обов'язково включати кілька рівнів міркувань, то було реалізовано механізм, де певні питання та фінальні результати можуть бути добавлені чи виключені базуючись на попередніх відповідях. Також, кожен фінальний результат має відносну оцінку відповіді на кожне питання, за якими і будується рейтинг.

Опишемо набір опцій базуючись на яких буде утворюватись рейтинг для кожного фінального результату.

Набір опцій:

* Активний чи лінивий відпочинок
* Довгий чи короткий відпочинок (кількість днів)
* Дорогий, дешевий чи середньої вартості
* Чи комфортно добиратись на авто/кораблем/літаком
* Чи є бажання екстремального відпочинку
* Море чи гори
* Чи важлива безпека під час відпочинку
* Клімат (холодний/помірний/теплий)
* Чи важливий комфорт
* Всередині країни чи за кордоном
* Та ін...

Результатом програми буде назва місця відпочи нку, що рекомендується даною ЕС. Приклади виводу:

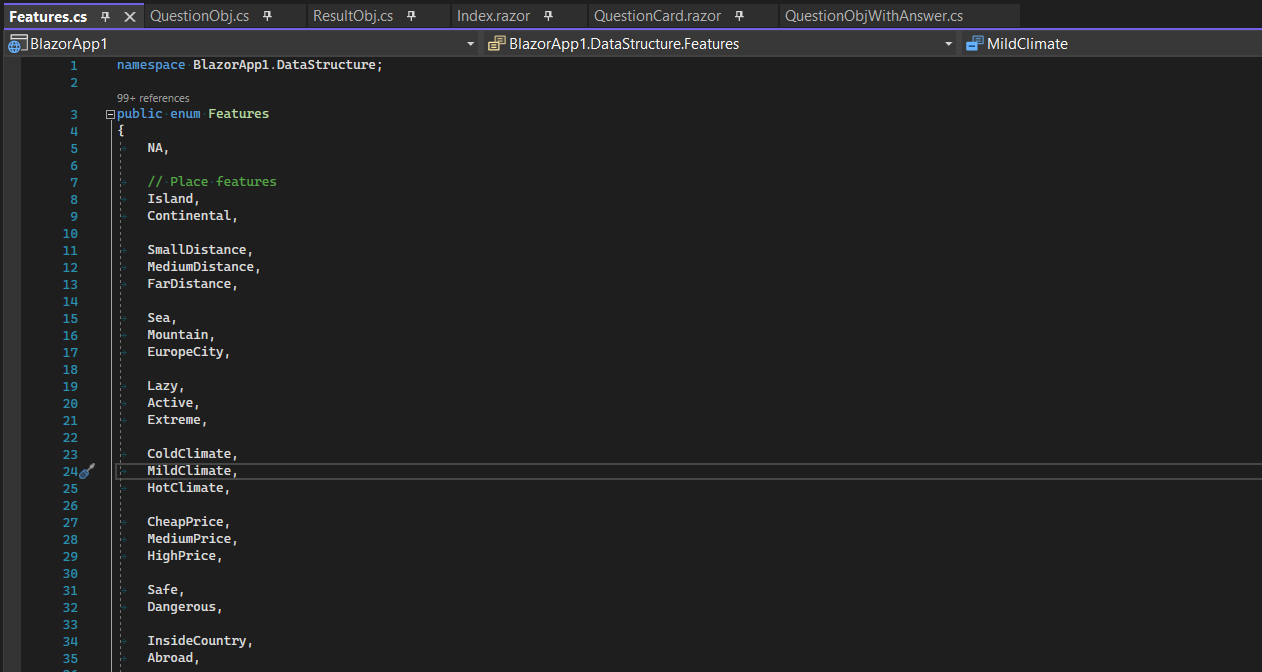
* Найкращий варіант для вас { варіант відпочинку 1 }. Також можете розглянути варіант { варіант відпочинку 2 } – у випадку, якщо є кілька розумних варіантів, то обереться два найкращі.
* Найкращий варіант для вас { варіант відпочинку } – якщо лише один варіант

Тепер опишемо правила, якими буде користуватись система:

1. ЯКЩО на момент перевірки чи завершення програми є кілька доступних варіантів, ТО першим виводиться той, який має найбільшу кількість балів, а другим – наступний за ним у рейтингу (як додаткова рекомендація).
2. ЯКЩО на момент перевірки чи завершення програми є різниця між першим та другим варіантом у рейтингу більше 10 одиниць, ТО виводиться тільки перший варіант відпочинку.
3. ЯКЩО користувач відповідає на питання, ТО певна особливість (залежно від питання) буде добавлена до загального списку особливостей користувача.
4. ЯКЩО користувач відповідає на питання, ТО отримана особливість опрацьовується усіма питаннями та фінальними варіантами.
5. ЯКЩО фінальний варіант має певний бал для особливості, ТО цей бал буде додано у рейтинг фінального варіанту.
6. ЯКЩО фінальний варіант не має оцінки для особливості, ТО до рейтингу буде додано 0 балів.
7. ЯКЩО особливість є затребуваною для фінального варіанту проте користувач її не обрав, ТО цей фінальний варіант не включається до списку доступних.
8. ЯКЩО особливість є такою, що виключає фінальний варіант та користувач її обрав, ТО цей фінальний варіант виключається зі списку доступних.
9. ЯКЩО фінальний варіант не є доступним, ТО система при виводі інформації не може його рекомендувати.
10. ЯКЩО особливість є затребуваною для певного наступного питання, проте користувач її не обрав, ТО це питання не буде показуватись користувачу.
11. ЯКЩО кількість днів для відпустки менше 5, ТО це коротка відпустка.
12. ЯКЩО кількість днів для відпустки більше чи рівне 5, ТО це довга відпустка.
13. ЯКЩО користувач обирає активну відпустку, ТО він зможе вибрати екстремальну відпустку.
14. ЯКЩО користувач обирає екстримальну відпустку, ТО варіант «Еверест» йому доступний.
15. ЯЩО користувач обирає «Море», ТО варіант «Еверест» йому не доступний.
16. ЯКЩО користувач обирає «Гори», ТО йому доступні варіанти «Буковель».
17. ЯКЩО користувач не може летіти літаком, ТО йому не доступні острови.
18. ЯКЩО користувачу не доступні острови, ТО йому не доступний варіант «Майорка».
19. ЯКЩО користувач має морську хворобу, ТО йому не доступні варіанти з «Круїзом».
20. ЯКЩО у користувача бюджет менше 500, ТО це дешева відпустка.
21. ЯКЩО у користувача бюджет між 500 та 1500, ТО це відпустка середньої вартості.
22. ЯКЩО у корстувача бюджет більше 1500, ТО це дорога відпустка.
23. ЯКЩО користувач може виїхати за кордон, ТО всі варіанти поза межами країни оцінюються в 10 балів.
24. ЯКЩО користувач не може виїхати за кордон, ТО всі варіанти всередині країни оцінюються в 10 балів.
25. ЯКЩО користувач молодого віку, ТО варіант «Буковель (катання на лижах/борді)» оцінюється в 10 балів.
26. ЯКЩО користувач старшого віку, ТО варіант «Круїз на лайнері» оцінюється в 10 балів.
27. ЯКЩО користувачу подобаються острови, ТО варіанти «Шрі Ланка», «Ісландія», «Азори» та «Майорка» оцінюються в 10 балів.
28. ЯКЩО користувачу подобаються європейські міста, ТО варіанти «Київ», «Львів» та «Відень» отримують 10 балів.
29. ЯКЩО користувач обирає холодний клімат, ТО варіанти «Буковель», «Ісландія» та «Еверест» отримують 10 балів.
30. ЯКЩО користувач обирає теплий клімат, ТО варіанти «Дубаї» та «Шрі Ланка» отримують 10 балів.
31. ЯКЩО більше нема питань, ТО користувач отримує результат.

Для реалізації було вирішено використовувати мову програмування C# та бібліотеку Blazor. Початкове налаштування було показано в лабораторній роботі №1.

Розглянемо список особливостей.



*Рис. 1 Частина зі списку особливостей.*

На основі цього списку будуться правила доступності питань та фінальних варіантів відповідей, а також рейтингова оцінка для кожного фінального варіанту.

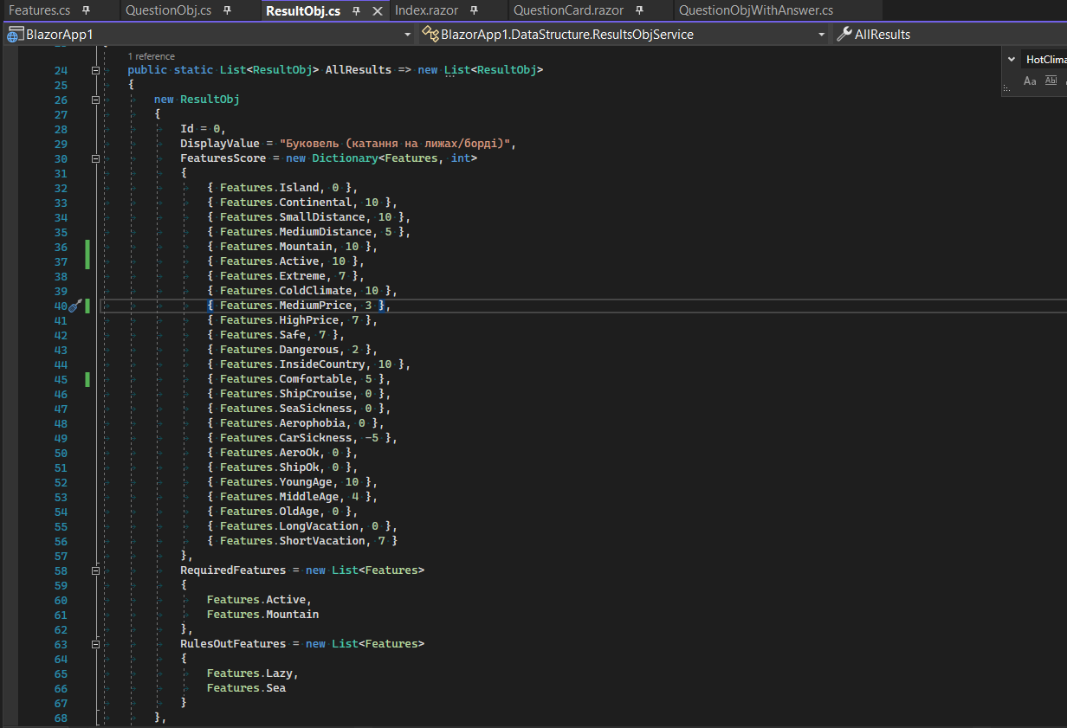
Тепер розглянемо модель об’єкта питання та одне реалізоване питання.

Text

Description automatically generated  
*Рис. 2. Модель питання та реалізовані питання.*

Тут можемо бачити, що питання про екстрим буде доступне тільки коли користувач обере варіант активного відпочинку. Також,у разі позитивної відповіді на це питання поверне користувачу особливість «Екстрим».

Тепер розлянемо як реалізовані фінальні варіанти відповідей.



*Рис. 3. Реалізація фінальних відповідей.*

Отже кожен варіант має список оцінок для особливостей, список особливостей які потрібні, щоб цей варіант був доступний та список особливостей які виключають цей варіант з результатів.

Перейдемо до коду, що реалізовує взаємодію елементів програми.

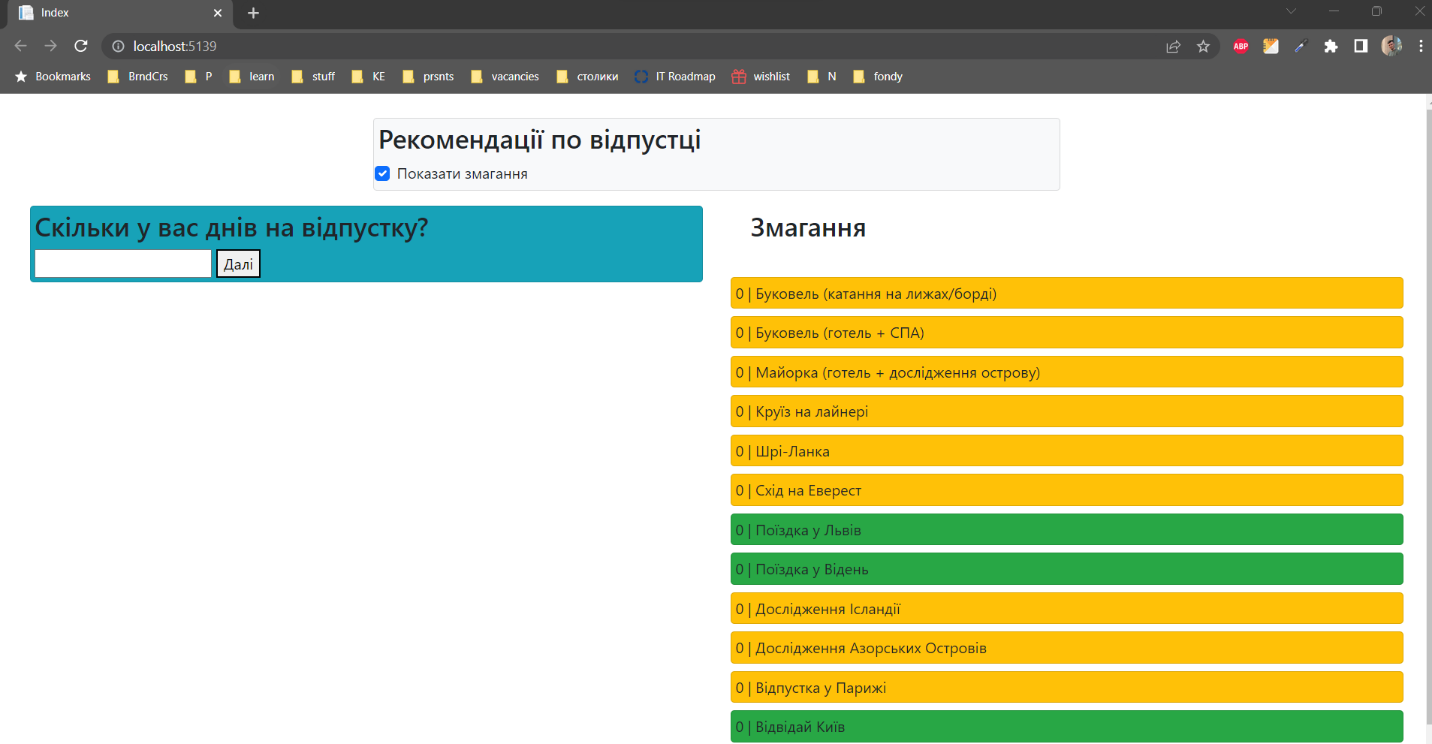
private async Task OnAnswered(QuestionObjWithAnswer value {

var q = AllQuestions.First(x => x.Id == value.QuestionId);  
 var featuredResult = q.GetFeatureResult(value.Answer);  
 Features.Add(featuredResult);  
 var answeredQIds = QuestionWithAnswers.Select(x => x.QuestionId).ToList();  
 QuestionObj? nextQ = null;  
 foreach (var newQ in AllQuestions.Where(x => answeredQIds.Contains(x.Id) == false)) {  
 if (!newQ.RequiredFeatures.Any()){  
 nextQ = newQ;  
 break;}  
 if (newQ.RequiredFeatures.Intersect(Features).Any()) {  
 nextQ = newQ; break; }

}  
 // calculate results  
 foreach (var res in AllResults) {  
 ResultsRace[res.Id] += res.FeaturesScore.ContainsKey(featuredResult) ? res.FeaturesScore[featuredResult] : 0; }  
 if (nextQ == null) {  
 // no more questions, show result  
 // rule out results and check if requirements met  
 foreach (var possRes in ResultsRace.OrderByDescending(x => x.Value)) {  
 var result = AllResults.First(x => x.Id == possRes.Key);  
 if (result.RequiredFeatures.Any() && result.RequiredFeatures.All(x => Features.Contains(x)) == false) { continue; }  
 if (result.RulesOutFeatures.Any() && result.RulesOutFeatures.Any(x => Features.Contains(x))) { continue; }  
 if (FirstBest == null) {  
 FirstBest = result;   
 continue; }  
 if (SecondBest == null) {  
 SecondBest = result;   
 showSecondBest = ResultsRace[FirstBest.Id] - possRes.Value <= 10;  
 break; } }  
 return; }  
 QuestionWithAnswers.Add(new QuestionObjWithAnswer { QuestionId = nextQ.Id });}

**Тестування**

Перейдемо до тестування системи. Запустимо застосунок та бачимо початковий екран.



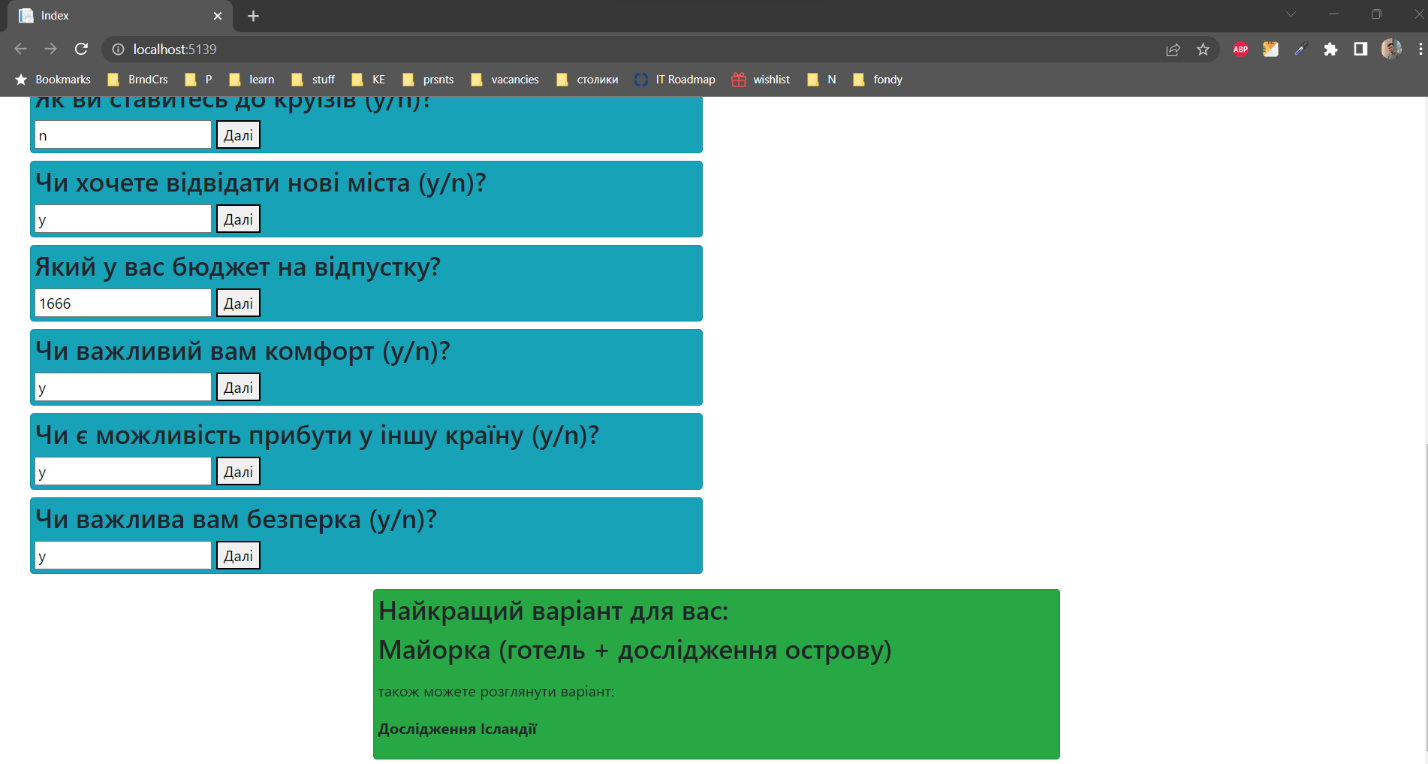
*Рис. 4. Початковий екран.*

Тепер дамовідповідь на декілька запитань. Бачимо наступний екран.

Graphical user interface, application

Description automatically generated  
*Рис. 5. Екран після декількох відповідей.*

Як бачимо – при виборі активного відпочинку ми отримали питання про екстремальний відпочинок. Давши згоду на екстремальний відпочинок, варіант «Еверест» став доступний як фінальнний варіант. Тепер дамо відповідь на питання що залишились.



*Рис. 6. Фінальний екран з пропозицією відпустки.*

Як бачимо, не зважаючи на те, що була вибрана екстремальна відпуста, по сумі балів рейтингу переміг варіант «Майорка».

A picture containing bar chart

Description automatically generated  
*Рис. 7. Фінальний рейтинг.*

Також бачимо, що варіанти «Буковель» були відсіяні так як було вибрано «Море», а не «Гори».

Бачимо очікуваний результат, тому можна дійти висновку, що наша експертна система працює відповідно до створених правил.

Весь код можна переглянути у GitHub репозиторії за посиланням у джерелах.[1]

**Висновок.** У ході даної лабораторної роботи мені вдалось розробити експертну систему за допомогою продукційної моделі. Також розроблено набір із 31 правила, якими керується система. Дана база знань включає кілька рівнів міркувань – це реалізовано тим, що кожне питання та фінальний варіант може бути показаний чи прихований за набором особливостей. Це дозволяє будувати систему будь-якої складності.

**Джерела**

1. GitHub project: <https://github.com/mpopeljuk/uni-decision_making-lab_3>