

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»
Зав. кафедри теоретичної та
прикладної системотехніки
_____ д.т.н., проф. С. І. Шматков
«__» _____ 2023 р

Пояснювальна записка

до кваліфікаційної роботи
бакалавра

на тему: «Моделювання експертних систем на основі сервісно-орієнтованого
підходу та cloud computing»

Захищено на засіданні
Атестаційної комісії № __
протокол № __ від __.12.2023 р.
Оцінка ____ / ____
Голова Атестаційної комісії

(підпис)

Скоб О. Ю.
(прізвище та ініціали)

Виконав:
студент 4 курсу, групи КІ- 41
Галузь знань: 12 – Інформаційні
технології
Спеціальність: 123 – «Комп'ютерна
інженерія»
Титаренко Владислав Сергійович



(підпис)

Керівник:
старший викладач
Павлов Анатолій Миколайович



Рецензент:
доцент кафедри безпеки інформаційних
систем і технології к.т.н.
Колованова Євгенія Павлівна

(підпис)

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і трьох додатків. Загальний обсяг роботи складає 79 сторінки, із яких 55 сторінок основної частини з 24 рисунками, 3 таблицями, 12 найменуваннями списку використаних джерел та чотирма додатками.

Метою кваліфікаційної роботи є розробка сервісу щодо розширення знань у різноманітних сферах за допомогою експертних систем на мові програмування JESS.

Об'єкт дослідження – процеси моделювання систем технічного забезпечення (СТЗ) на основі використання експертних систем, які будуть побудовані на базі сервісно-орієнтованого підходу та cloud computing з метою поліпшення процесу оцінки стану системи.

Предмет дослідження – модель експертних систем на основі сервісно-орієнтованого підходу та cloud computing.

Завдання, яке вирішується в кваліфікаційній роботі полягає в тому, щоб скориставшись існуючими програмними засобами та сервісно-орієнтованим підходом та cloud computing, розробити сервіс з визначення очних захворювань.

Область застосування – розробка веб-додатків. Розроблений програмний продукт може використовуватися в сфері медицини.

Ключові слова: хмарні обчислення, cloud computing, JESS, Google App Engine, реалізація сервісу.

ABSTRACT

The explanatory note to the bachelor's thesis consists of an introduction, three chapters, conclusions, a list of references and three appendices. The total volume of the work is 79 pages, including 55 pages of the main part with 24 figures, 3 tables, 12 references and four appendices.

The purpose of the qualification work is to develop a service to expand knowledge in various fields using expert systems in the JESS programming language.

The object of research is the processes of modeling technical support systems (TSS) based on the use of expert systems, which will be built on the basis of a service-oriented approach and cloud computing in order to improve the process of assessing the state of the system.

The subject of research is a model of expert systems based on a service-oriented approach and cloud computing.

The task that is solved in the qualification work is to use existing software tools and a service-oriented approach and cloud computing to develop a service for determining eye diseases.

The field of application is web application development. The developed software product can be used in the field of medicine.

Keywords: cloud computing, cloud computing, JESS, Google App Engine, service implementation.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ РОЗВИТКУ «ХМАРНИХ ТЕХНОЛОГІЙ». МОДЕЛІ І СЕРВІСИ.....	9
1.1 Головні чинники розвитку хмарних технологій.....	9
1.2 Сучасний cloud computing	16
1.3 Моделі та сервіси хмарних технологій.....	17
Висновки за розділом 1	30
РОЗДІЛ 2. ДОСЛІДЖЕННЯ ЕКСПЕРТНИХ СИСТЕМ.....	31
2.1 Визначення експертних систем та їх властивості.....	31
2.2 Класифікації експертних систем.....	35
2.3 Базові функції експертної системи	39
2.4 Подання та організація знань в експертних системах.....	41
Висновки за розділом 2.....	42
РОЗДІЛ 3. ПОБУДОВА МОДЕЛІ ТА РЕАЛІЗАЦІЯ СЕРВІСУ ЩОДО РОЗШИРЕННЯ ЗНАНЬ У РІЗНОМАНІТНИХ СФЕРАХ ЗА ДОПОМОГОЮ ЕКСПЕРТНИХ СИСТЕМ.....	43
3.1 Побудова моделі експертних систем на основі сервісно- орієнтованого підходу.....	43
3.2 Формулювання, принцип роботи та синтаксис мови JESS	44
3.3 Призначення й основні функції Google App Engine.....	47
3.4 Розробка та аналіз програмного сервісу	49
3.4.1 Режим уведення знань експерта	49
3.4.2 Режим консультації користувача	53

3.4.3 Адміністрування сервісу.....	55
Висновки за розділом 3.....	58
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДОДАТКИ.....	61

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

LAN	–	локальна обчислювальна мережа;
WAN	–	глобальна комп'ютерна мережа;
CC	–	cloud computing;
IaaS	–	Infrastructure-as-a-service;
PaaS	–	Platform-as-a-service;
SaaS	–	Software-as-a-service;
HaaS	–	Hardware-as-a-Service;
EC	–	експертна система;
JESS	–	Java Expert System Shell.
GAE	–	Google App Engine.

ВСТУП

Розробка сервісу для вирішення складних завдань, які важко розв'язати експерту-людині, і який може досягати результатів, які не поступаються за якістю і ефективністю до результатів, які надає експерт, є актуальною проблемою. Для реалізації такого сервісу потрібні режим введення знань експерта в базу знань та режим консультації користувачів. Однак, кількість користувачів та розмір бази знань ще невідомі, так само як і кількість обчислювальних ресурсів, необхідних для роботи сервісу. Використання технології хмарних обчислень може допомогти вирішити цю проблему.

Актуальність роботи. Сервісно-орієнтований підхід та cloud computing дозволяють збільшувати доступність експертних систем для широкого кола користувачів, знижувати витрати на розробку та підтримку, а також підвищувати їхню ефективність та надійність. Моделювання експертних систем на основі цих технологій є важливим напрямом розвитку інформаційних технологій, особливо з урахуванням широкого застосування штучного інтелекту в різних галузях, таких як медицина, фінанси, виробництво тощо. Тому, дослідження та розробка ефективних та доступних рішень у цій галузі має велике значення.

Метою дослідження є створення сервісу для розробки та використання експертних систем, який базується на сервісно-орієнтованому підході та використовує cloud computing. Розроблений сервіс буде використовувати методи Web 2.0, що дозволить користувачам збільшувати якість та кількість експертних систем, що в свою чергу підвищить якість рішень та зробить їх більш точними.

Об'єкт дослідження – процеси моделювання систем технічного забезпечення (СТЗ) на основі використання експертних систем, які будуть побудовані на базі сервісно-орієнтованого підходу та cloud computing з метою поліпшення процесу оцінки стану системи.

Предмет дослідження – модель експертних систем на основі сервісно-орієнтованого підходу та cloud computing.

Методи дослідження: методи системного аналізу, евристичні методи, методи аналізу інформативності та методи оцінки якості класифікації.

Завдання дослідження

1. Дослідити основні концепції сервісно-орієнтованого підходу та cloud computing і їх застосування в IT-індустрії.
2. Розглянути поняття експертних систем та їх використання в різних сферах діяльності.
3. Розробити сервіс за допомогою експертних систем на основі сервісно-орієнтованого підходу та cloud computing і провести його тестування.

РОЗДІЛ 1

АНАЛІЗ ПОНЯТТЯ «ХМАРНІ ТЕХНОЛОГІЇ»

1.1 Головні чинники розвитку хмарних технологій

Хмарні технології роками розвивалися з постійними досягненнями в галузі обчислювальної техніки. Хмарні технології - втілення розвитку та об'єднання існуючих обчислювальних стилів, таких як ґрид-обчислення та утилітарні обчислення. Деякі сліди ґрид-обчислень та утилітарних обчислень можна знайти у випадках використання хмарних технологій. Однак хмарні технології відрізняються від попередніх технологій поєднання новітніх технічних розробок і нових бізнес-моделей, що створює значну комерційну цінність у нових сценаріях використання[1].

Початок можемо побачити ще з того часу, коли віддалений доступ до системи розподілених обчислень був нереальним. Шлях до хмарних обчислень прокладали десятиліттями досліджень, зокрема в області паралельних і розподілених обчислень. Ретельна дискусія про розвиток хмарних обчислень ніколи не може оминати увагою безперервні інновації в галузі електронного та обчислювального обладнання. У міру того, як розвивалися апаратні технології, розвивалося і програмне забезпечення. Крім того, з розвитком комунікаційних протоколів, технологія мережевого зв'язку, а також Інтернет також відіграли життєво важливу роль у цьому процесі.

Приблизно в 1970-х роках перші дні комерційного використання обчислювальної техніки почалися з мейнфреймів. Мейнфрейми були великими суперкомп'ютерами, котрі потребували потужного охолодження, а вся система була дуже дорогою. Ця проблема вирішилася з появою концепції розподіленого використання ресурсів. Це була основна мета, яку спеціалісти тривалий час намагалися досягти. Дана розробка відкрила можливість спільного використання одних і тих самих ресурсів комп'ютера декількома користувачами та відпала потреба в декількох мейнфреймах. Мейнфрейм-

системи були доступні для декількох офісів за допомогою неінтелектуальних терміналів, які передавали дані до сервера, але не мали ніякої обчислювальної потужності.

В умовах централізованої обробки даних паралельні запити, які надходили з різних терміналів, частіше за все не могли бути оброблені одночасно. Пізніше з'явилися недорогі мікропроцесори та завдяки ним спеціалісти розробили менші обчислювальні пристрої. Дані пристрої стали називати інтелектуальними терміналами, пізніше відомими як клієнтські системи, і вони брали участь у частковому виконанні програм разом з серверами.

За часів мейнфреймів кілька обчислювальних терміналів встановлювали зв'язок через сервер, але вже в 1980-х роках було досягнуто успіху та налагодження взаємодії між системами, що розташовані поруч на невеликій відстані. Збільшення швидкості передачі даних було головною проблемою при створенні мережевого зв'язку між віддаленими комп'ютерами. Рішенням стали винаходи локальних (LAN) та глобальних (WAN) мереж. Оскільки комп'ютери, розміщені на відстані, могли безпосередньо взаємодіяти один з одним, не звертаючись до сервера, режим обчислень перейшов від централізованої мережевої архітектури епохи клієнт-сервер. Це стало початком однорангових мереж (P2P), де кожен комп'ютер міг грати роль як клієнта, так і сервера.

До 1980-х років продуктивність комп'ютерів залежала від швидкості процесорів. Винахід паралельних обчислень дозволив використовувати декілька процесорів одночасно. Розподілені обчислення дозволили об'єднати декілька обчислювальних систем в потужну обчислювальну систему. Для оптимальної продуктивності розподіленої обробки була потрібна вища швидкість мережевого зв'язку.

Наступним кроком еволюції в галузі обчислювальної техніки була поява концепції кластеризації. Обчислювальні кластери складаються з декількох вузлів (комп'ютерів), з'єднаних мережею, і які виконують схожі

завдання. Таким чином, виконання завдання може бути швидшим, оскільки воно може бути розподілене і паралельно виконуватися на декількох машинах всередині кластера. Всі вузли кластера разом створюють враження єдиної системи.

Кластерна архітектура призводить до більш потужних і надійних обчислювальних систем завдяки створенню пулів ресурсів, але вона викликає занепокоєння через свою залежність від голови кластера. Продуктивність такої системи значною мірою залежала від ефективності та доступності кластерного центру. Існування кластерного вузла також підвищує ймовірність виникнення єдиної точки відмови. Але виникали проблеми, які потребували вирішення. Спеціалісти дійшли до того, що кожен вузол, що належить до кластера, повинен мати однаковий пріоритет. Було потрібно, щоб вони всі могли виконувати схожі функції, і жодному з них не потрібно було призначати роль «голови» серед них. Нова архітектура була представлена на початку 1990-х років та отримала назву «сітка». Приклад даної архітектури показано на рисунку 1.1.

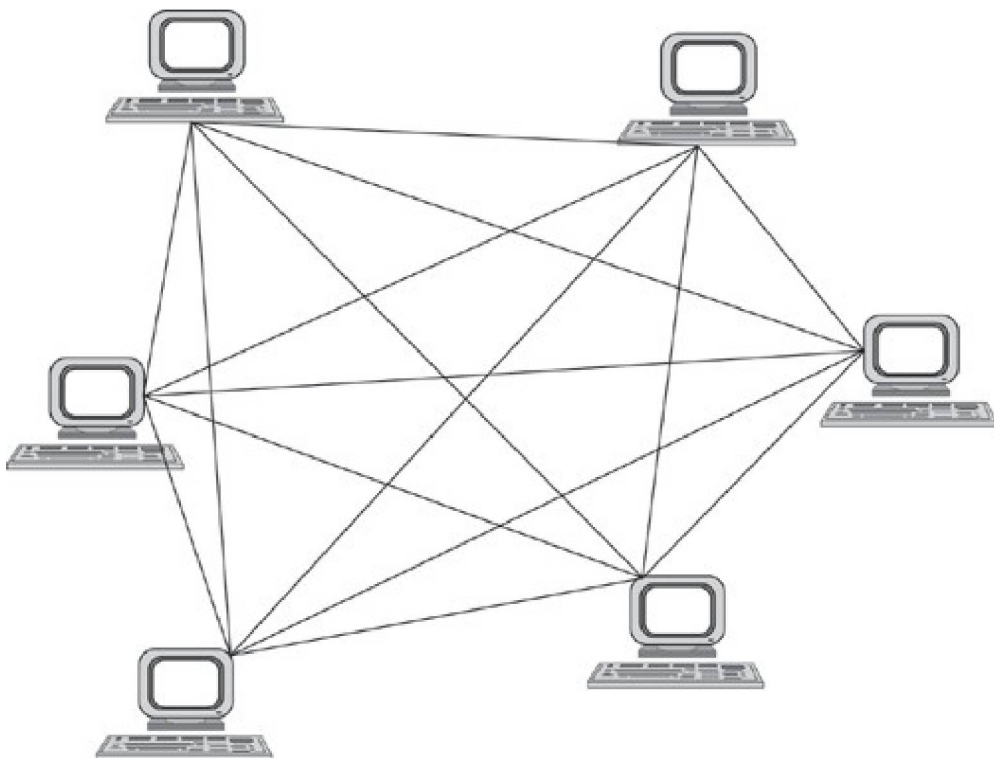


Рис 1.1. Сіткова модель обчислень.

Аналогію цієї концепції можна провести з електромережею, до якої підключені споживачі можуть отримати регулярне електропостачання. Було обґрунтовано, що абонентам достатньо підключитися до обчислювальної мережі в цілому (а не до якогось конкретного вузла мережі), щоб користуватися обчислювальними ресурсами, подібно до того, як споживачі підключаються до електромережі.

За концепцією кластеризації слідом з'явилась концепція грид-обчислень, яка запровадила ідею децентралізації управління в розподіленому обчислювальному середовищі. Головною перевагою грид-обчислень було те, що обчислювальне середовище тепер може бути побудоване з гетерогенних обчислювальних систем, тобто систем з різноманітними апаратними конфігураціями.

З'явилась нова проблема: перед створенням глобальної розподіленої обчислювальної мережі було узгодження між системами різних адміністративних доменів, зокрема управління ресурсами та безпекою. Популярність грид-обчислення зросла, коли корпоративні додатки використовували його модель, і спеціалісти спрямували свої зусилля на уніфіковану реалізацію безпеки та управління ресурсами. Зараз розподілене обчислювальне середовище гетерогенних систем можна розподіляти по різних географічних областях, використовуючи для взаємодії інтернет-мережу. ПО, що використовується в грид-обчисленнях, спеціально розроблено для роботи в таких розподілених середовищах.

Таблиця 1.1

Порівняння кластерних та грид-обчислень

Характеристика	Кластерні обчислення	Грид-обчислення
Керування системою	Централізоване	Децентралізоване
Керування користувачами	Централізоване	Децентралізоване
Створення пулу ресурсів	Так	Так
Тип вузла ресурсу	Гомогенний	Гетерогенний
Планування використання ресурсів	Централізоване	Децентралізоване
Тип мережевого з'єднання	LAN	LAN/MAN/WAN
Архітектура системи	Образ єдиної системи	Автономні, самостійні вузли
Здатність до відмовостійкості	Низька	Висока

Винахід концепції грид-обчислень став важливим кроком на шляху до майбутньої ери обчислень. Грид-обчислення іноді називають прямим попередником хмарних обчислень. Але однією з головних переваг хмарних обчислень є можливість масштабування в реальному часі. На відміну від грид-технологій, обчислювальні ресурси в хмарі можна додавати в реальному часі, щоб задовольнити попит на обчислення. Це стало можливим, оскільки віртуалізація ресурсів обмежує додатки від прямого доступу до фізичних ресурсів.

Поетапний розвиток до ери хмарних обчислень від раннього періоду розвитку обчислень показано на рисунку 1.2.

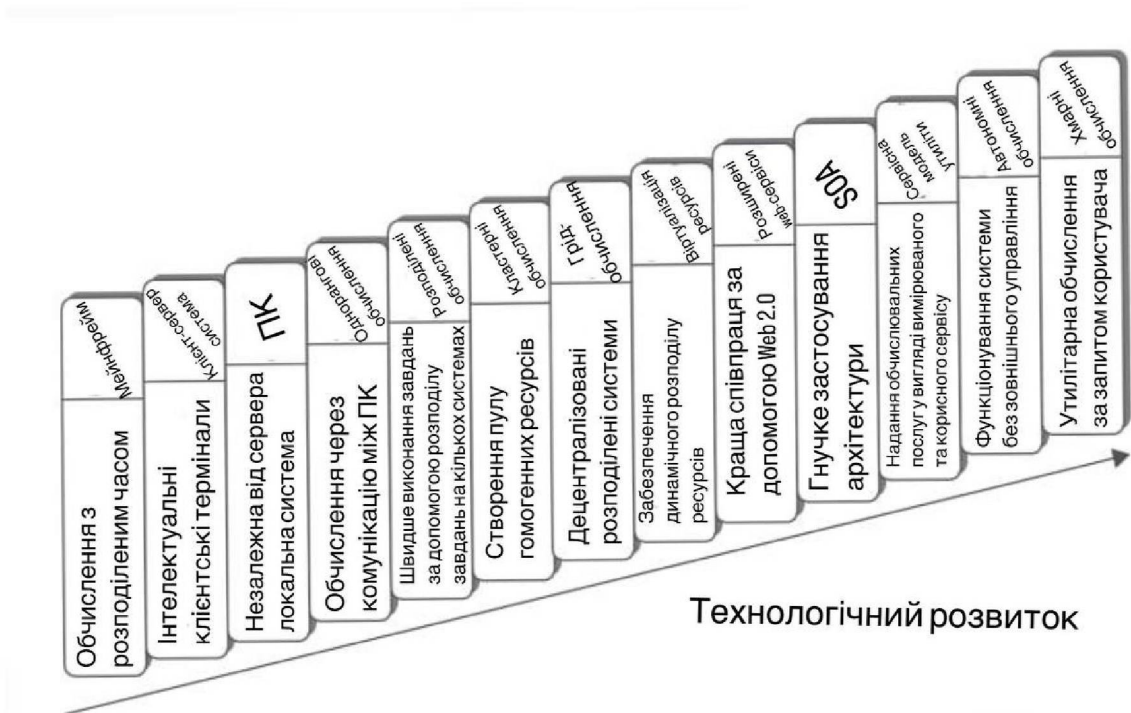


Рис.1.2. Технологічний прогрес на шляху до зрілості хмарних обчислень.

Еволюція хмарних обчислень була результатом, заснованим на технологічному прогресу таких галузях, як обчислювальна техніка, апаратне забезпечення, розподілені обчислення, комунікаційні технології, веб-технології, архітектура додатків і т.д., які однаково сприяли цьому процесу. Рисунок 1.3 підсумовує, як різні технології були об'єднані та сприяли появі хмарних технологій.



Рис.1.3. Конвергенція технологій для розвитку хмарних обчислень.

Слід зазначити, що віртуалізація ресурсів дозволила відокремити програмні системи від фізичних апаратних пристроїв. Таким чином, обчислювальні системи, до яких мають доступ користувачі, стали переносимими на інші набори апаратних пристроїв. Це робить систему більш відмовостійкою і вирішує проблему нестачі ресурсів, оскільки функція розділення дає можливість додавати більше ресурсів у пул ресурсів. Такий тип системи, яка може зростати відповідно до вимог, називається масштабована система[2].

Хмарні обчислення з'явилися в результаті еволюції і принесли революцію в галузі обчислень.

1.2 Сучасний cloud computing

На даний момент існує 5 популярних напрямків розвитку cloud CC, на які варто було б звернути увагу в 2023 році. Хмарні обчислення стали більш актуальними за часів пандемії через те, що багато хто почав працювати з дому у віртуальному середовищі. Нижче будуть представлені ключові тенденції CC, над якими потрібно замислитися.

По-перше, треба зробити фокус на сталості задля більшого впровадження CC. Компаніям усього світу прийдеться виділити час для того, щоб розібратися з питанням сталого розвитку. Вони розуміють, що зменшене споживання енергії з комп'ютерів, інфраструктурних послуг та сховищ, допоможе їм просуватися вперед до сталого розвитку. Багато технологічних гігантів працюють над хмарними технологіями, які дозволять їм досягнути нульового рівня викидів вуглецю.

По-друге, ключову роль CC відіграватиме у штучному інтелекті. Вони використовуються від пошукових запитів у Google до фільтрів у соціальних мережах. Розвиток штучного інтелекту та машинного навчання в найближчі роки приведе до випуску нових продуктів та послуг, і хмарні обчислення продовжать відігравати важливу роль у наданні цих послуг та будівництві необхідної інфраструктури.

По-третє, CC розвиватиметься для розробки нових сценаріїв використання. Дані, які надали Gartner, показують, що за весь 2023 рік витрати на CC наблизяться до цифри в 500 мільярдів доларів. CC залишатимуться міцною основою майже для кожної цифрової послуги. За допомогою більш швидкісних обчислювальних мереж, таких як Wi-Fi 6E і 5G, можна побачити різні формати даних, які передаються в потоковому режимі. CC доповненої та віртуальної реальності призведуть до інновацій у різних галузях, що змусить компанії ще більше перейти на хмарні сервіси.

По-четверте, набуде популярності «функції як послуга», також їх називають «безсерверним». Це можна зрозуміти так, що компанії зможуть

обирати сервіс, який зможе розширити можливості серверу, коли їх додатки цього вимагають, а не інвестувати у великі сервери.

По-п'яте, гібридні хмарні середовища зростатимуть. Зазвичай компанії мали на вибір міграцію до приватної або публічної хмари. Проте зараз компаніям дають змогу експериментувати з гібридною хмарою, що поєднує в собі приватну та публічну. Ті дані, які треба швидко доставити користувачам і до яких часто звертаються, можна зберігати в публічних хмарах, а критично важливу інформацію краще зберігати в приватних хмарах, використовуючи власні додатки. Використання декількох сервісів від різних вендорів може ускладнювати роботу. Гібридна хмара допоможе спростити складну природу та оптимізувати роботу користувачів.

1.3 Моделі та сервіси хмарних технологій

Хмарні сервіси можна організувати або розгорнути кількома способами. Вибір способу розгортання залежить від вимог організації-споживача. Модель розгортання описує корисність хмари, а також визначає межі доступу до неї. Модель також вказує на відносне розташування хмари по відношенню до розташування організації-споживача.

Виділяють три поширені моделі:

- Публічна;
- Приватна;
- Гібридна;

Публічна хмара - модель розгортання хмари, яка забезпечує найширший доступ до споживачів серед усіх хмарних розгортань. Кожен, хто підписується на неї, отримує відкритий доступ до цього хмарного ресурсу. Споживачем може бути як окремих користувач, так і група людей, що представляють споживачем може бути як окремих користувач, так і група людей, що представляє якусь організацію або підприємство. Публічну хмару також називають зовнішньою хмарою, оскільки з точки зору фізичного

розташування вона залишається зовнішнім або off-premises, а споживачі можуть отримати до неї віддалений доступ.

Розгортання публічної хмари сприяє мультиоренді в найвищому ступені. Один і той самий фізичний обчислювальний ресурс можна використовувати спільно з кількома непов'язаними між собою споживачами. Це забезпечує великі переваги, оскільки стає можливим для одного постачальника хмарних послуг обслуговувати велику кількість споживачів. Коли велика кількість споживачів, розкиданих по всьому світу, спільно використовують ресурси з дата-центру одного постачальника, це автоматично підвищує коефіцієнт використання ресурсів і зменшує витрати постачальника на надання послуг. Таким чином, для споживачів ключовою перевагою використання публічної хмари є її фінансова вигода.

З іншого боку, провайдери публічних хмар отримують вигоду від масштабів своєї діяльності. Будучи великими за обсягом і бізнесом, вони можуть дозволити собі найсучасніші технології та кваліфікованих людей. Це забезпечує кращу якість послуг. Завдяки цій моделі споживачі можуть отримати доступ до потенційно кращого сервісу за нижчою ціною. Оскільки різні споживачі (з різних частин світу) мають різні вимоги до робочого навантаження протягом дня, тижня, місяця або року, хмарний провайдер завжди може ефективно підтримувати навантаження під час високого попиту (який зазвичай створюється частиною його споживачів у будь-який конкретний момент).

Розгортання приватної хмари не передбачає відкритого доступу для всіх. Вона призначена в основному для організаційного і доступ до розгортання приватної хмари обмежений для широкої публіки. Приватна хмара також називають внутрішньою хмарою, оскільки вона створюється для внутрішніх цілей організації. У той час як публічні хмари однаково корисні як для окремих користувачів, так і для організацій, приватні хмари як правило, слугують лише для цілей організацій. Для систем з високим рівнем

безпеки та критично важливих систем, таких як систем оборонних організацій, приватна хмара є рекомендованим підходом.

В той час як публічна хмара не може фізично знаходитися в будь-якому місці (фізична межа) споживача, приватні хмари можуть розташовуватися або в приміщенні організації-споживача (on-premises), або зовні (off-premises) у будь-якому нейтральному місці. On-premises приватні хмари фізично знаходяться під фізично розташовуються на території організації-споживача, а також в межах мережі. Off-premises приватні хмари знаходяться за межами мережі організації, але залишається під контролем або наглядом організації-споживача.

Приватна хмара може бути створена і управлятися самою організацією-споживачем або вона (споживач) може передати цю відповідальність іншому постачальнику обчислювальних послуг. Рисунки та зображено локальні та зовнішні приватні хмари відповідно.



Рис. 1.6. On-premises приватна хмара.

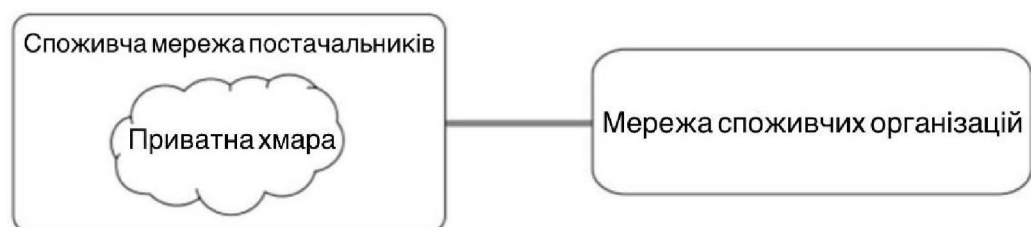


Рис. 1.7. Off-premises приватна хмара.

Однією з основних відмінностей приватної хмари від публічної є те, що будь-яка приватна хмара має відносини зі споживачем "один до одного", тоді

як публічна хмара має відносини "один до одного" зі споживачем. Це означає, що ресурси приватної хмари залишаються призначеними лише для однієї організації-споживача тільки для однієї організації-споживача і не можуть бути надані іншим. Таким чином, особливості мультиоренди (де орендарі є зовнішніми непов'язаними організаціями) не застосовуються в приватній хмарі, як це відбувається в публічній хмарі. Але така ізоляція ізоляція забезпечує конфіденційність і створює більш безпечне обчислювальне середовище. Однак, це не обов'язково означає, що публічна хмара недостатньо безпечна[2].

Інша відмінність виникає через здатність споживача контролювати хмару.

Споживачі не мають контролю над загальнодоступним хмарним середовищем. Але у випадку з приватною хмарою споживачі можуть скористатися більшістю переваг хмарних обчислень і все ще можуть контроль над середовищем. Для споживачів вартість використання приватної хмари є вищою, ніж публічної хмари, оскільки тут ресурси залишаються виділеними для конкретної організації.

Приватна хмара не підтримує багатокористувацьку оренду, а отже, не може запропонувати багато з точки зору ефективності. Але вона забезпечує більшу безпеку та конфіденційність даних.

Таблиця 1.2

Порівняння приватної хмари та публічної хмари

Приватна хмара	Публічна хмара
Це може бути обох типів - on-premises та off- premises.	Не може бути on-premises розгортання публічної хмари.
Дані приватної хмари можуть бути доставлена приватною мережею.	Дані можуть бути доставлений тільки через загальнодоступну мережу.

Продовження таблиці 1.2

Він не підтримує функцію мультиоренди для не пов'язаних між собою та зовнішніх орендарів.	Це демонструє можливість багатокористувацької роботи завдяки його повній функціональності.
Ресурси призначені для ексклюзивного використання одним споживача (як правило, організації).	Ресурси діляться між кількома споживачами.
Доступ до приватної хмари має обмежена кількість людей.	Доступ є для всіх бажаючих.
Це для організаційного використання.	Це може використовуватись як організацією, так і користувачем.
Споживачі відіграють важливу роль у управлінні хмарою.	Управління цілком залежить від відповідальності провайдера.
Споживачі мають більшу контроль над середовищем.	Споживачі мають дуже мало контролю або ж жодного контролю.
Це забезпечує більшу впевненість щодо безпеки даних, оскільки вони залишаються під контролем безпекової межі організації-користувача.	Розгортання публічної хмари часто викликає занепокоєння щодо безпеки та конфіденційності даних.
Вартість обчислень є вищою порівняно з публічним хмарним сервісом.	Це більш економічно, оскільки кілька не пов'язаних між собою споживачів (орендарів) користуються однією інфраструктурою.

Закінчення таблиці 1.2

Це неідеальний сценарій для просування філософії "плати по мірі використання".	Це ідеальна модель для практики філософії "плати за користування".
Ресурс часто може простоювати, що призводить до марнотратства ресурсу.	Використання ресурсів є оптимальним завдяки наявності більшої кількості споживачів.

Гібридна хмара - це поєднання приватної з публічною хмарою разом. Ця модель розгортання допомагає компаніям скористатися перевагами приватної або загальнодоступної хмари, зберігаючи критичні програми та дані. Водночас вона забезпечує економічну вигоду, зберігаючи спільні дані та додатки в публічній хмарі. На рисунку 4.9 показано гібридну хмарну модель, що поєднує публічну хмару з локальною приватною хмарою.



Рис.1.8. Гібридна хмарна модель.

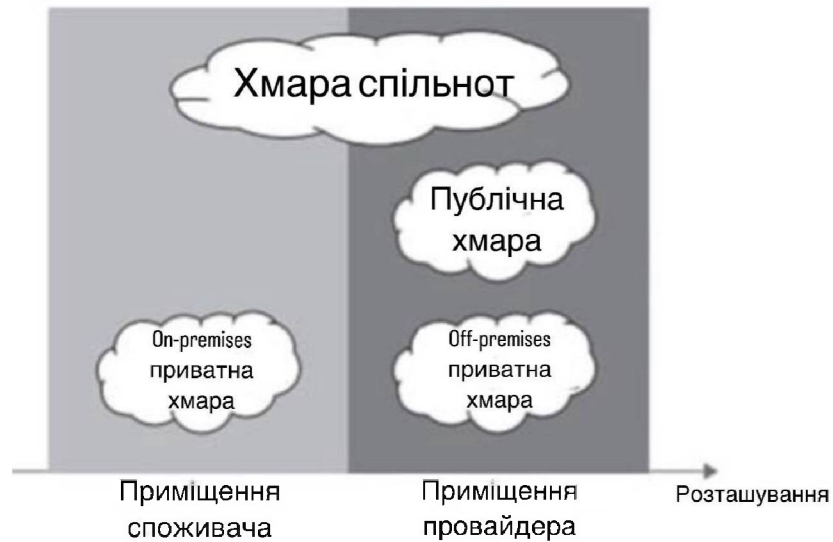


Рис.1.9. Фізичне розташування хмарних розгортань.

Три категорії комп'ютерних послуг, які люди споживають з часів традиційних

обчислень:

- Послуги інфраструктури;
- Служба платформи;
- Сервіс програмних додатків.

Хмарні обчислення говорять про надання цих засобів споживачам як обчислювальних послуг через мережу/інтернет. Перевага для споживачів полягає в тому, що вони можуть користуватися цими послугами через Інтернет у будь-який час і в необхідному обсязі, перебуваючи у своєму власному місці, з мінімальними витратами. Їм потрібно лише мати простий і відповідний пристрій доступу (наприклад, ПК, ноутбук, планшет, мобільний телефон тощо) для доступу до цих послуг. За допомогою цих простих пристроїв будь-хто може отримати доступ до будь-якої обчислювальної інфраструктури, платформи або програмного додатку на основі оплати за фактичне використання[3].

Хмарні обчислення пропонують обчислювальну інфраструктуру, платформу та додатки, що надаються "як послуга". Ці послуги вважаються первинними послугами хмарних обчислень і називаються:

- Інфраструктура як послуга (IaaS);
- Платформа як послуга (PaaS);
- Програмне забезпечення як послуга (SaaS)[4].

IaaS, PaaS і SaaS стосуються надання обчислювальних послуг споживачам через мережу.

Хмарні обчислення надають доступ до обчислювальних ресурсів у віртуальному середовищі, яке в народі відомому як "хмара". IaaS надає віртуалізовані апаратні (не фізичні, а імітовані програмні) ресурси споживачам, відомі як віртуальні ресурси. Вона надає можливість віддаленого використання віртуальних процесорів, пам'яті, сховищ даних та мережевих ресурсів.

Ці віртуальні ресурси можна використовувати так само, як і фізичні (апаратні) ресурси для побудови будь-якої обчислювальної установки (наприклад, віртуальної машини або віртуальної мережі).

З цієї причини IaaS також називають Hardware-as-a-Service (HaaS).

Споживачам більше не потрібно керувати або контролювати базову обчислювальну інфраструктуру, яку вони споживають як IaaS. IaaS - це найнижчий рівень сервісної моделі хмарних обчислень. Стек компонентів IaaS показано на рисунку 1.10. Це обчислювальне рішення, в якому складнощі та витрати на управління базовим обладнанням передаються на аутсорсинг деяким постачальникам хмарних послуг.

Тут значення аутсорсингу відрізняється від його звичайного тлумачення. У традиційній системі аутсорсинг обчислювального обладнання означає, що якась інша сторона буде надавати та управляти апаратними ресурсами відповідно до вимог користувача. Користувачі мають прямий доступ до цього обладнання і можуть використовувати ці ресурси, встановлюючи на них необхідне програмне забезпечення. У моделі хмарних обчислень IaaS провайдер організовує та управляє апаратними ресурсами для користувачів, але користувачі не мають прямого доступу до цих апаратних ресурсів. Апаратні ресурси представлені як імітовані програмні компоненти

(реалізовані за допомогою техніки віртуалізації ресурсів) і надаються споживачам через Інтернет (або мережу) за допомогою веб-сервісів. Споживачі можуть використовувати ці симульовані компоненти так само, як реальні апаратні пристрої, і можуть побудувати обчислювальну систему з необхідним процесором, пам'яттю та обчислювальну систему з необхідними процесорами, пам'яттю, накопичувачами та мережевими пристроями.

Апаратні ресурси в хмарних обчисленнях не є безпосередньо доступними для користувачів. Модель IaaS надає всі апаратні компоненти у віртуальному режимі: віртуальний процесор, віртуальне сховище, віртуальні віртуальний процесор, віртуальна пам'ять, віртуальні комутатори тощо.

Таким чином, імітований (або віртуальний) апаратний компонент, що надається відповідно до вимог споживача є унікальною особливістю моделі IaaS. Споживачі можуть отримати доступ до цих віртуальних апаратних ресурсів на вимогу і в будь-який час з будь-якої точки мережі. Вони можуть створювати комп'ютери (віртуальні комп'ютери), використовуючи ці віртуальні (або віртуалізовані) апаратні компоненти, і навіть можуть встановлювати на них операційні системи та інше програмне забезпечення.



Рис.1.10. Стек IaaS-компонентів.

Найбільші постачальники обчислювальної техніки, такі як Amazon, Google, GoGrid, RackSpace, надають послуги IaaS. Всі ці постачальники пропонують віртуалізовані апаратні ресурси різних типів. Крім того, що вони пропонують компонентів ресурсів окремо для побудови будь-якої обчислювальної системи, постачальники IaaS, як правило, пропонують споживачам віртуальні машини, створені на замовлення (з цих віртуальних компонентів). Наприклад, Amazon EC2 та Google Compute Engine є популярними серверними середовищами. Споживачі можуть встановити ОС і почати працювати на цих серверах. Крім віртуальної машини, сховище даних сховище є дуже поширеною пропозицією IaaS. Amazon S3 - популярний сервіс зберігання даних, доступний як IaaS.

Хмарні провайдери надають налаштовані та готові віртуальні машини за допомогою пропозицій IaaS, які це порожні комп'ютери (навіть без ОС), зібрані з віртуальних компонентів.

Якщо ж говорити про PaaS, у комп'ютерних технологіях платформа означає базову систему, на якій можуть бути встановлені (а також розроблені) програмні додатки. Обчислювальна платформа складається з апаратних ресурсів, операційної системи, проміжне програмне забезпечення (за потреби) та бібліотеки часу виконання. Прикладні програми також встановлюються на цю платформу.

Розробка та розгортання додатків у традиційних обчисленнях вимагає участі користувачів участі користувачів в управлінні апаратним забезпеченням, операційною системою, проміжним програмним забезпеченням, веб-серверами та іншими компонентами. Наприклад, користувачі повинні встановити відповідний фреймворк (наприклад, J2EE, .NET) перед тим, як перед початком роботи на будь-якій платформі додатків. З іншого боку, послуга PaaS звільняє користувачів від усіх цих від

усіх цих проблем і надає споживачам готову платформу через мережу/Інтернет.

Стек компонентів PaaS, крім того, забезпечує розробку та розгортання додатків (розробку та розгортання) платформу над стеком компонентів IaaS. Постачальник PaaS не тільки надає повністю кероване середовище для розробки та розгортання додатків, але також дбає про управління ресурсами нижнього рівня (рівень інфраструктури). Як показано на рисунку 1.11, PaaS постачається з інтегрованими в неї можливостями IaaS.



Рис.1.11. Стек PaaS-компонентів.

Таким чином, PaaS створюється шляхом додавання додаткових шарів програмного забезпечення над IaaS. З використанням PaaS спрощується спільна розробка додатків, коли кілька користувачів можуть працювати з різних географічних місць. PaaS також знижує загальну вартість володіння (TCO), оскільки обчислювальна платформа стає доступною на умовах оренди.

На ринку є багато пропозицій PaaS. Google App Engine, Microsoft Azure Platform, GoGrid Cloud Center, Force.com є дуже популярними серед них. PaaS з відкритим кодом також доступні на ринку. Хмарна ливарня - одна з таких, яку розробляє VMware.

Однією з проблем моделі PaaS є те, що вона прив'язує розроблені додатки до платформи. Це спричиняє проблеми з переносимістю. Наприклад, додаток, розроблений на Google App Engine на будь-якій мові програмування (що підтримується Google PaaS), використовує API Google, а отже його не можна запустити на PaaS-платформі інших постачальників. Ця проблема з переносимістю через прив'язку до постачальника.

Програмне забезпечення як послуга (SaaS) - це спосіб надання додатків як послуги через мережу/Інтернет. Інтернет, які користувачі можуть безпосередньо споживати без необхідності встановлювати або налаштовувати програми. У традиційних комп'ютерних технологіях споживачі повинні були платити не лише за ліцензію на програмне забезпечення, але й витратити значну частину свого бюджету на створення інфраструктури та платформи, на якій буде працювати додаток. SaaS усуває цю проблему і обіцяє простіший та дешевший спосіб використання програми.

SaaS розміщується постачальниками SaaS і надається споживачам через мережу/Інтернет. На відміну від традиційних упакованих додатків, які користувачі встановлюють на власному комп'ютері, SaaS постачальники SaaS запускають їх у своїх дата-центрах. Клієнтам не потрібно купувати ліцензії на програмне забезпечення або будь-які додаткові обчислювальні ресурси для підтримки програми і можуть отримати доступ до додатків за певну орендну плату на основі використання. Додатки SaaS іноді називають веб-програмами, або хостинговим програмним забезпеченням.

SaaS будується шляхом додавання шарів над стеком компонентів PaaS (рис. 1.12). Це можливість використання додатків, які адмініструються та надаються постачальником послуг через хмарну інфраструктуру. У моделі SaaS усім керує постачальник, включно з модернізацією або оновленням програми; навіть дані та дії, з якими працює додаток, також управляються (зберігання в базі даних, файлах тощо) за допомогою SaaS. Користувачі

можуть отримати доступ до додатків через інтерфейс тонкого клієнта (зазвичай браузер) з будь-якого місця.



Рис.1.12. Стек SaaS-компонентів.

SaaS був найбільш примітивним серед різних типів хмарних сервісів. SaaS почав набирати популярність на початку поточного десятиліття, пропонуючи рішення як для звичайних користувачів, так і для підприємств. Електронна пошта - це один із поширених прикладів SaaS, яким користується кожен. Пакет CRM (управління взаємовідносинами з клієнтами) від Salesforce.com набув популярності серед підприємств з початку 2000-х років. SAP (Systems, Application and Products), як постачальник рішень для планування ресурсів підприємства (ERP), вийшла на ринки SaaS CRM та ERP зі своїм рішенням "Business ByDesign". Компанія Oracle запустила свою CRM SaaS "На вимогу". Сьогодні на ринку також є багато популярних пропозицій SaaS для звичайних користувачів наприклад, GoogleApps, Microsoft Office 365 та інші.

Висновки за розділом 1

Було розглянуто основні поняття, пов'язані з хмарними технологіями, їх типи та переваги в порівнянні з традиційними рішеннями, що базуються на локальній інфраструктурі. Було досліджено такі поняття, як платформи, сервіси та моделі хмарного обчислення, а також їх характеристики, переваги та недоліки.

РОЗДІЛ 2

ДОСЛІДЖЕННЯ ЕКСПЕРТНИХ СИСТЕМ

2.1 Визначення експертних систем та їх властивості

Досвід розробки та експлуатації комп'ютерних обчислювальних систем різного призначення довів, що їх ефективність залежить не тільки від якості вбудованих алгоритмів, але й від обсягу знань, які вони використовують під час роботи. Ця концепція призвела до розробки спеціалізованих програмних систем, які можуть вважатися "експертами" у певній предметній галузі - експертні системи. Можемо дати визначення, що експертні системи являють собою інтелектуальні програмні засоби, які, внаслідок інтерактивної взаємодії з користувачем, здатні отримувати, акумулювати та коригувати знання у певній предметній області, виводити нові знання, розв'язувати на основі цих знань практичні задачі та пояснювати спосіб їх розв'язання[6].

ЕС займаються з середини 70-х років ХХ ст. в межах вивчення штучного інтелекту, тому цей напрям є молодим в науці. Через виникнення змін у використанні програмних засобів ШІ та в технологіях розробки даних напрям почав активно розвиватися. Серед них найбільш важливими є:

- ізоляція логічного виведення від залежної частини предметної області (бази знань).
- покращення взаємодії між користувачем та комп'ютерною програмою[7].

ЕС мають важливу властивість - орієнтацію на процеси накопичення та організації знань. Для вирішення практичних завдань у певній предметній області експертні системи базуються на високоякісних знаннях висококваліфікованих фахівців-експертів, які зосереджені в базах знань експертних систем. Ці системи виступають в ролі помічників-консультантів для осіб, які займаються управлінням або обслуговуванням складних технічних, соціальних або природних систем, в тому числі електроенергетичних систем.

ЕС можна використовувати для розв'язання різних типових завдань, таких як інтерпретація, діагностика, прогнозування, моніторинг і багато інших. Хоча ці завдання можна розв'язати за допомогою традиційних алгоритмічних програм, ЕС мають свої суттєві особливості, які відрізняють їх від звичайних програм:

1. ЕС моделюють механізм мислення людини для вирішення проблем у певній галузі, а не її фізичну природу. Це не означає повного відтворення психофізичної моделі мислення, але основна увага приділяється методології вирішення проблем, яку використовують експерти в цій галузі.
2. ЕС використовують знання, накопичені висококваліфікованими експертами в певній предметній області. Ці знання зберігаються окремо від програмного коду, що дає змогу створювати гнучкі експертні системи, які можна використовувати для розв'язання різних прикладних задач.
3. Для вирішення проблем використовуються евристичні методи, які ґрунтуються на знаннях і досвіді людей, накопичених у вирішенні подібних проблем. Ці методи не гарантують точного результату, але дозволяють знайти прийнятне рішення, навіть коли вихідна інформація обмежена. Такі методи є важливим інструментом для вирішення складних проблем, де не завжди працюють точні математичні моделі.

Але не завжди доцільно використовувати ЕС для вирішення тих чи інших задач, бо на сьогодні є й інші варіанти розв'язання. Наведу таблицю порівнянь, яка розкриває тему доцільного використання ЕС.

Таблиця 2.1

Доцільне використання ЕС

Доцільно	Не доцільно
Існування способів розв'язання задач, які базуються на досвіді та здогадках, разом із тим, що вони можуть бути вирішені без жорстких алгоритмів.	Можливість вирішення задач за допомогою ефективних алгоритмів.
Наявність експертних знань у предметній області	Відсутність експертних знань або обмежений доступ до них.
Ці завдання призначені для оцінки чогось або когось.	Задачі мають характер розрахунків.
Вирішення завдань в умовах, коли вихідна інформація не є визначеною.	Наявність точних даних та алгоритмів.
Можливість вирішення задач за допомогою формальних методів міркування.	Для вирішення задач використовують методи на основі алгоритмів.
Знання статичні	Знання динамічні

На сьогодні ЕС не мають такого широкого застосування, як очікувалося на початку їх використання, і це має свої об'єктивні причини:

1. Більшість ЕС вимагають спеціального апаратного забезпечення, і їх перенесення на універсальні платформи знижує продуктивність.
2. Великі ЕС незручні в управлінні, і керування ними може вимагати декількох операторів, що робить їх адаптацію до конкретних ситуацій та організацій складною.
3. Для вирішення практичних задач їх не використовують через повільність деяких ЕС.
4. Впровадження ЕС зазвичай потребує організаційних змін.

Проте, використання ЕС має також свої переваги, до яких можна віднести наступне.

1. ЕС - це такі програми, які дозволяють зберігати знання висококваліфікованих фахівців після їх виходу на пенсію, зміни роботи або смерті.
2. Ці системи дозволяють збирати та використовувати знання багатьох експертів у предметній області, що значно підвищує ефективність розв'язання практичних задач.
3. Робота ЕС не залежить від психофізичного та емоційного стану людини-експерта, що забезпечує підвищену надійність та якість отримання розв'язку.
4. Для того, щоб отримувати доступ до експертних знань, можна використовувати будь-які наявні комп'ютерні та апаратні засоби, що робить інформацію більш доступною.
5. Використання ЕС дозволяє знизити витрати на отримання експертних знань під час розв'язання практичних задач.
6. ЕС можуть пояснювати шлях отримання результату та використані під час розв'язання задачі експертні знання.
7. Такі системи можуть використовуватися як інтелектуальна навчальна програма для діагностики знань учнів та навчання на прикладах розв'язання складних задач предметної області.
8. ЕС можна використовувати для організації доступу до баз даних з використанням алгоритмів штучного інтелекту, що підвищує ефективність роботи з базами даних.

Застосування ЕС у різних областях людської діяльності дозволяє знизити витрати на отримання та зберігання експертних знань, покращити якість прийнятих рішень та забезпечити їх надійність.

Наприклад, в енергетиці ЕС використовуються для діагностики та прогнозування відмов обладнання, а також для оптимізації експлуатації електростанцій та енергосистем. В медицині вони можуть допомогти при

діагностиці та плануванні лікування різних захворювань, а також при виборі оптимальних доз лікарських засобів.

У високомолекулярній хімії ЕС можуть допомогти при проектуванні нових полімерів та матеріалів з певними властивостями. В інших галузях вони можуть використовуватися для розв'язання складних фінансових та економічних задач, для проектування та планування виробництва, для управління логістичними системами та багатьох інших завдань.

У будь-якій сфері застосування ЕС важливо пам'ятати, що вони не замінять повністю експертів, а слугують допоміжним інструментом для прийняття рішень. Крім того, для ефективної роботи ЕС необхідно мати якісні та достовірні дані, на яких вона буде базуватись для роботи з базами даних.

2.2 Класифікації експертних систем

На даний момент існують різноманітні класифікації ЕС, які представлені на рис. 2.1. ЕС - це програмні комплекси, що мають широкий спектр застосування і можуть бути класифіковані за різними ознаками.

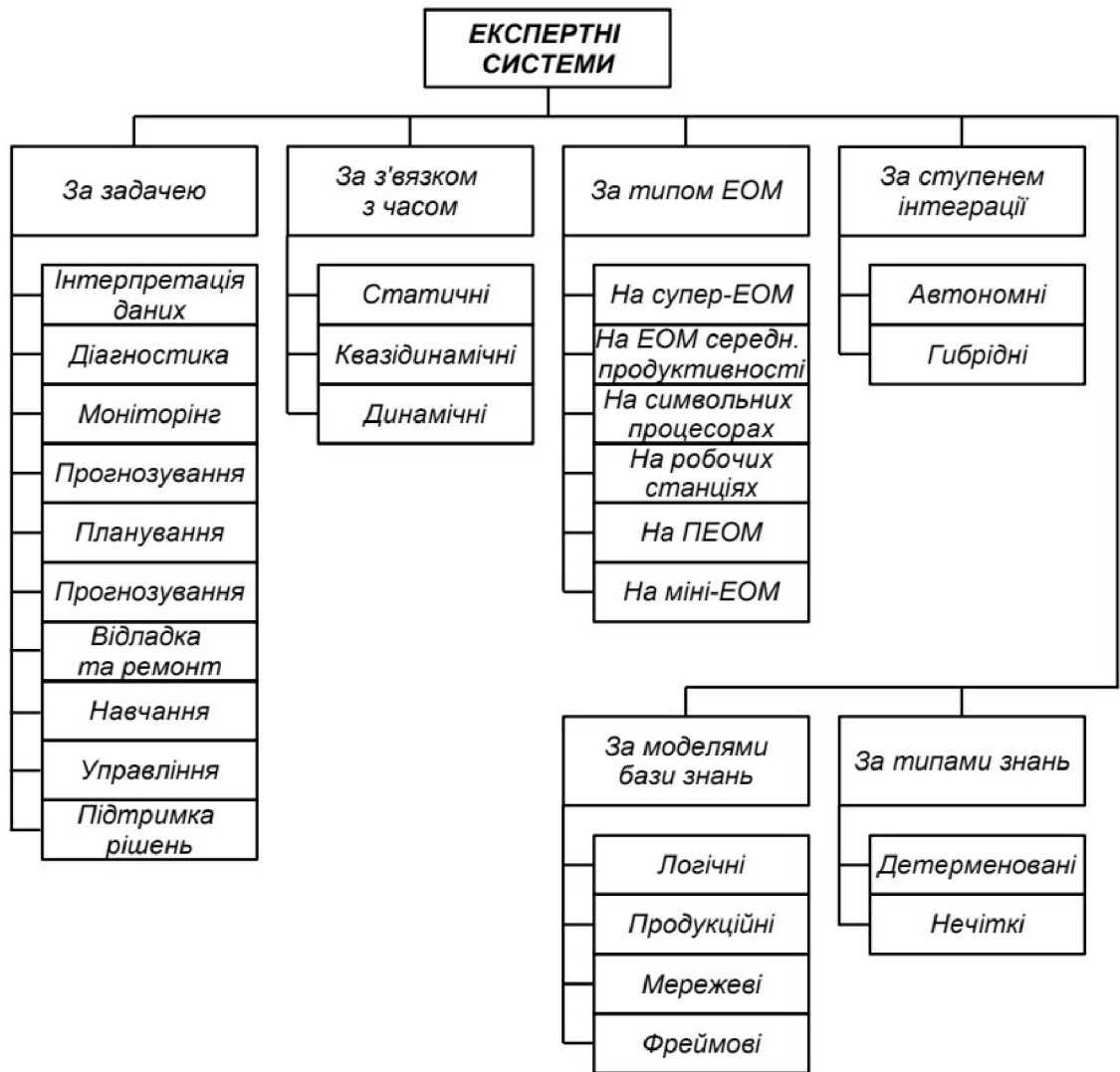


Рис. 2.1. Класифікація експертних систем.

Аналізуючи рис. 2.1, можемо побачити 6 класифікацій, які будуть розглянуті нижче більш детально.

Класифікація за типом задачі. ЕС спрямовані на вирішення задач, які є складними для розв'язання за допомогою звичайних математичних методів. Серед типових задач, для вирішення яких використовують ЕС, можна відзначити:

- інтерпретація даних – отримання корисної інформації з первинних даних;
- діагностика – виявлення дефектів та їх чинників у системі;

- моніторинг – забезпечення безперебійної інтерпретації даних в умовах реального часу й сповіщає, якщо контрольовані параметри виходять за допустимі межі;
- прогнозування – передбачення вигодних наслідків, спираючись на події минулого та теперішнього;
- планування – встановлення порядку дій, які необхідні на досягнення попередньо визначених цілей;
- проектування – встановлення конфігурації технічної системи з урахуванням обмежень;
- налагодження та ремонт – встановлення порядку дій з метою забезпечення потрібних режимів роботи системи;
- навчання – передбачення аналізу та корекцію знань та вмінь учнів з метою досягнення більш ефективного навчального процесу;
- управління – процес формування ключових впливів, що контролюють роботу непростих технічних систем;
- підтримка прийняття рішень – забезпечення особи, що здійснює прийняття рішень, важливою інформацією й рекомендаціями з метою спрощення цього процесу.

Класифікація за зв'язком з реальним часом:

- статичні ЕС засновані на передбаченні незмінності часу та моделі статичної предметної області;
- динамічні ЕС працюють в режимі реального часу з датчиками об'єктів та неперервною інтерпретацією даних, які надходять в систему;
- квазідинамічні ЕС реагують на змінну ситуації в залежності від таймеру, який вбудований в систему;

Класифікація за типом ЕОМ. Ефективність взаємодії експертних систем з апаратною частиною комп'ютерів сильно впливає на їх продуктивність, що робить їх унікальними програмними продуктами. Розрізняють кілька типів експертних систем:

- ЕС, що працюють на суперкомп'ютерах (CRAY, CONVEX тощо), використовуються для вирішення унікальних та стратегічно важливих задач;
- ЕС на середньопродуктивних ЕОМ (mainframe) призначені для вирішення різноманітних завдань;
- ЕС на символічних процесорах та робочих станціях (SUN, Silicon Graphics, APOLLO) використовуються для вирішення завдань з опрацювання зображень та відео;
- ЕС на персональних комп'ютерах (IBM PC, MAC) є доступними та зручними в роботі інструментами для вирішення різних задач;
- ЕС на міні- та суперміні-ЕОМ (VAX, micro-VAX) використовуються для розв'язання завдань, які не потребують високої продуктивності.

Класифікація за ступенем інтеграції з іншими програмами:

- автономні ЕС працюють незалежно від користувача і призначені для вирішення конкретних експертних завдань без використання традиційних методів обробки даних. Вони працюють у режимі консультацій з користувачем для розв'язання специфічних експертних задач;
- гібридні ЕС – це програмне забезпечення, яке комбінує стандартні прикладні програми та інструменти для роботи зі знаннями. Воно може бути інтелектуальним доповненням до прикладних програм або інтегрованим середовищем для вирішення складних завдань з елементами експертного знання. Розробка таких систем є складною задачею, оскільки вона включає компонування різних програм та методологій, що може викликати як теоретичні, так і практичні труднощі;

Класифікація за моделями баз знань:

- логічні ЕС використовують бази знань та машини логічного виведення, які базуються на формальній логіці та зазвичай використовують числове предикатне числення;
- продукційні ЕС використовують правила типу "Якщо ... То ..." у своїй роботі та базуються на апараті продукцій;
- мережеві ЕС для зберігання та обробки знань використовують апарат семантичних мереж;
- фреймові ЕС для роботи зі знаннями використовують фрейми - об'єктно-орієнтовані інформаційні структури;

Класифікація за видами знань:

- ЕС із детермінованими знаннями вимагають точного визначення всієї вихідної інформації та розв'язання практичних задач за жорсткими правилами;
- ЕС із невизначеними знаннями дозволяють розв'язувати практичні задачі, незважаючи на неповноту, недостовірність, невизначеність або нечіткість вихідної інформації;

2.3 Базові функції експертної системи

До основних функцій ЕС відносять:

- 1) отримання знань;
- 2) представлення знань;
- 3) управління процесом пошуку рішень;
- 4) пояснення прийнятого рішення[8].

Отримання знань - це процес, коли експерти з певної області діляться своїм досвідом та знаннями з іншими людьми, щоб ті змогли використовувати ці знання у експертних системах. Найпоширеніший спосіб передачі знань - це спілкування між експертами з проектування систем та експертами з певної області. Проте, цей процес є складним та неефективним, тому що він може зайняти багато часу та ресурсів. Це обумовлено низкою причин. Експерти з конкретних галузей зазвичай використовують свій

власний спеціалізований жаргон, що ускладнює передачу корисної інформації іншим людям без такого ж досвіду. Крім того, знання експертів у специфічних областях не завжди можна чітко сформулювати в термінах математичної теорії або детермінованої моделі. Це може потребувати використання евристичних правил, які можуть бути складнішими, ніж традиційні логічні моделі. Для розв'язання проблем у цих областях експертам необхідно знати, які види інформації потрібні для формування міркувань, які джерела інформації надійні, і як можна поділити складну проблему на більш прості задачі. Експертний аналіз потрібно вміщати в широкий контекст, що містить багато фактів, які для сторонньої людини можуть не бути очевидними. Визначити певну кількість та природу знань, які необхідні для вирішення певної задачі, може бути досить складно.

В цілому, теорія витягнення знань описує різноманітні завдання, які залежать від психологічних, мовних та філософських аспектів і т.д.

Формалізація знань - це спроба зберегти інформацію так, щоб вона була доступна для подальшої обробки комп'ютерами. У цьому процесі використовуються методи, які схожі на ті, що людина використовує для зберігання та роботи з інформацією у своїй голові. Однак у формалізації знань більше уваги приділяється логічній, а не біологічній стороні процесу. Головною метою є розробка методів для формального подання великих обсягів корисної інформації, яку можна далі обробляти комп'ютером. Основні вимоги до таких методів полягають у логічній адекватності, евристичній потужності та природній нотації.

Управління процесом пошуку рішень з використанням штучного інтелекту, зокрема ЕС, передбачає застосування логічного виведення. Логічне виведення - це процес отримання нових знань з вихідних фактів за заданими правилами. Існують різні види логічного виведення, такі як дедукція, індукція, інтуїція, евристика, метод спроб та помилок, абдукція, судження за аналогією, судження за замовченням та немонотонні судження.

При створенні ЕС важливо враховувати, яким чином здійснюється доступ до знань та як вони використовуються під час пошуку рішень.

Здатність ЕС пояснювати методику прийняття рішення інколи називають прозорістю системи. Прозорість ЕС означає, наскільки легко персоналу розуміти, як саме програма приймає рішення. Важливо, щоб різні люди, які працюють з системою, могли перевірити її коректність та правильність роботи на різних рівнях. Наприклад, користувачі хочуть бачити, що їхні запити опрацьовуються правильно, інженери зі знань переконуються в тому, що знання правильно застосовуються, експерти з предметної області можуть відстежувати хід міркувань, а програмісти повинні мати можливість відлагоджувати систему на вищому рівні, ніж просто окремі функції програмного коду. Крім того, менеджер системи повинен мати можливість переконатися, що рішення, які пропонує експертна система, обґрунтовані та надійні.

2.4 Подання та організація знань в експертних системах

Однією з головних задач під час створення ЕС є визначення того, які знання необхідно включити до неї. Також важливо вирішити, як саме ці знання будуть представлені. Ці питання пов'язані між собою, і обраний спосіб представлення може бути неефективним для вираження деяких знань[9].

Питання "як представляти" знання можна розділити на дві незалежні задачі: організацію знань та їх представлення у формалізмі. Організація знань є важливою складовою будь-якої мови представлення, і її розв'язання має бути здійснене незалежно від використовуваного формалізму.

Питання, які вирішуються під час представлення знань, включають визначення того, які знання необхідно включити, організацію та представлення знань. Склад знань ЕС визначається проблемним середовищем, архітектурою системи, потребами користувачів та мовою спілкування.

Для функціонування статичної ЕС необхідні знання про процес розв'язання задачі, знання про мову спілкування, знання про способи подання та модифікації знань, а також знання, що підтримують структурні та керівні знання.

Предметні знання містять дані про предметну галузь та способи перетворення цих даних під час розв'язання поставлених завдань. Знання про подання та управління є метазнаннями щодо предметних знань.

Висновки за розділом 2

У даному розділі було розглянуто експертні системи та було виявлено, що їх використання є дуже важливим у багатьох галузях, де необхідно приймати рішення на основі складних аналізів даних та експертного досвіду.

РОЗДІЛ 3

ПОБУДОВА МОДЕЛІ ТА РЕАЛІЗАЦІЯ СЕРВІСУ ЩОДО РОЗШИРЕННЯ ЗНАНЬ У РІЗНОМАНІТНИХ СФЕРАХ ЗА ДОПОМОГОЮ ЕКСПЕРТНИХ СИСТЕМ

3.1. Побудова моделі експертних систем на основі сервісно-орієнтованого підходу.

При побудові моделі експертних систем на основі сервісно-орієнтованого підходу, насамперед, необхідно визначитись з архітектурою експертної системи.

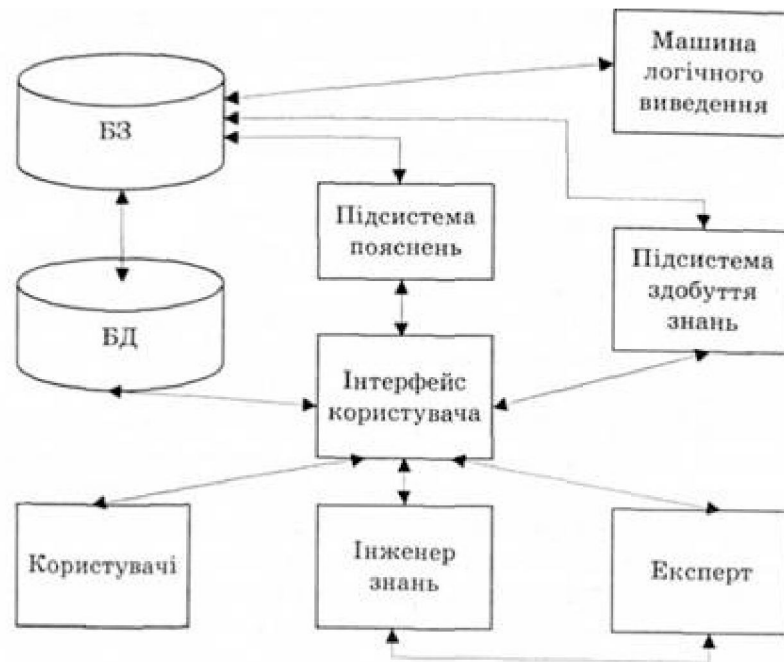


Рис. 3.1. Архітектура експертної системи

Тобто, архітекутра експертної системи містить:

- БЗ - це ядро експертної системи, є сукупністю знань у формі зрозумілої експертів і користувачеві мови, призначеної для зберігання експертних знань, які використовуються при рішенні задач ЕС;
- БД, призначеної для зберігання фактів або гіпотез, які є проміжними рішеннями або результатом спілкування системи із зовнішнім середовищем;

- машини логічного виведення - механізму, що моделює хід міркувань експерта, оперуючи знаннями та даними з метою отримання нових даних зі знань та інших даних, що містяться в робочій пам'яті;
- інтерфейсу користувача, призначеного для ведення діалогу з користувачами для отримання фактів, необхідних для процесу міркування;
- підсистеми пояснень, що дає користувачеві можливість розуміти процес отримання результату;
- підсистеми здобуття знань, призначеної для коригування і поповнення БЗ. У простому випадку це інтелектуальний редактор БЗ, який надає ІЗ можливість створення БД у діалоговому режимі.

У складніших БС - засоби здобуття знань з БД, що містить неструктурований текст, графічну інформацію тощо.

База знань ЕС - це сукупність відомостей Про, для якої розробляється БС. Для функціонування ЕС її БЗ має бути наповнена знаннями. З цією метою запрошують експертів - висококваліфікованих спеціалістів у тій галузі, для якої розробляється ЕС. їх завдання - формально описати усі свої знання, потрібні для функціонування ЕС.

У БЗ містяться знання двох типів:

- загальновідомі факти, явища, закономірності цієї Про;
- набір емпіричних правил, відповідно до яких спеціалісти приймають рішення за умов невизначеності, неповноти та суперечливості інформації.

3.2. Формулювання, принцип роботи та синтаксис мови JESS

JESS - це програмний продукт, який включає двигун правил та скриптове середовище, написані мовою Java Ернестом Фрідманом-Хіллом в лабораторіях Sandia National у Ліверморі, Каліфорнія. Цей інструмент

дозволяє створювати Java-додатки з використанням знань, які подаються у вигляді декларативних правил, та виводу результатів[10].

JESS є одним з найшвидших та має невеликий розмір. Його скриптова мова дає доступ до всього Java API, а повнофункціональне середовище розробки базується на платформі Eclipse. JESS має унікальні функції, такі як зворотний логічний вивід, запити до робочої пам'яті та можливість керування та виведення Java об'єктів. Крім того, JESS використовує розширену версію алгоритму Rete для обробки правил, який забезпечує швидкодію та ефективність[11].

Застосування JESS дозволяє створювати експертні системи та бази знань, що автоматично приймають рішення на основі заданих правил і фактів. Без компіляції Java-коду можна створювати та керувати Java-об'єктами, що спрощує процес розробки та прискорює випуск додатків, заснованих на JESS.

В цілому, JESS є потужним інструментом для створення експертних систем та баз знань на основі правил, який може значно прискорити процес розробки та поліпшити результати.

Для повного розуміння синтаксису мови програмування JESS потрібно розібратися з основними синтаксичними одиницями мови:

- символи є фундаментальним поняттям мови. Вони подібні до ідентифікаторів в інших мовах. Символи можуть містити літери, цифри та певні знаки. Символ не може починатися з цифри, але він може починатися з одного з наступних символів (деякі з них мають особливе значення). Існують спеціальні символи, які JESS інтерпретує по-особливому: нуль, TRUE, FALSE. Символи є чутливими до регістру;
- числа. JESS використовує java-функції `parseInt(java.lang.String)`, `parseLong(java.lang.String)` та `parseDouble(java.lang.String)` для перетворення цілих, довгих та значень з плаваючою комою;

- рядки. Символьні рядки в JESS беруться в лапки. Для відокремлення спеціальних символів у рядку можна використовувати зворотну косу риску;
- списки є основною структурою даних в JESS. Списки завжди складаються з пари охоплюючих дужок та одного або більше символів, чисел, рядків або інших списків у цих дужках;
- виклики функцій. Як і в Lisp, весь код в JESS (керуючі структури, присвоєння, виклики процедур) має форму виклику функції. Деякі функції мають назви, що нагадують оператори Java, і мають схожу функціональність. Виклик функції в JESS - це простий список, заголовком якого є ім'я функції;
- змінні. Ідентифікатори, які починаються зі знаку питання, є змінними JESS. Ім'я змінної може складатися з будь-якої комбінації символів, цифр і знаків, крім крапки. Змінні можуть посилатися на символи, числа, рядки або списки. Ви можете присвоїти значення змінній за допомогою функції `bind`;
- визначення функцій. Щоб визначити функцію, використовуйте конструкцію `defunction`. Ім'я функції має бути символом. Кожен параметр повинен бути змінною. Коментар повинен бути укладений в лапки. Функція може мати будь-яку кількість виразів. Специфікатор `return` вказує на значення, що повертається функцією;
- Шаблони. Кожен факт має шаблон. Факт отримує ім'я та список слотів з шаблону. Таким чином, шаблон в JESS схожий на клас в Java. Шаблони створюються автоматично, якщо ви не створюєте їх вручну. Шаблон створюється за допомогою конструкції `deftemplate`. Оголошення шаблону включає ім'я, рядок документації, умову розширення, список оголошень і список визначень слотів;
- визначення правил. Правила в JESS нагадують оператори `if-then` у процедурних мовах, але використовуються по-іншому. Якщо оператори "if-then" виконуються в певний час і в певному порядку,

залежно від того, як їх написав програміст, то правила JESS виконуються, коли виконується їхня ліва (умовна) частина.

3.3 Призначення й основні функції Google App Engine

Google App Engine (GAE) - це хмарна платформа, що дозволяє розробникам створювати та розгортати веб-додатки без необхідності управління інфраструктурою. За допомогою GAE розробники можуть швидко та ефективно створювати додатки та масштабувати їх залежно від потреб користувачів. Розглянемо призначення GAE та його переваги для розробки веб-додатків[12].

Перш за все, GAE дозволяє розробникам створювати веб-додатки на хмарній платформі, що зменшує необхідність у власних серверах та інфраструктурі. Замість цього, розробники можуть сконцентруватись на створенні функціоналу своїх додатків та залишати управління хмарною інфраструктурою на GAE.

Друге призначення GAE - це можливість розгортання веб-додатків на масштабованій інфраструктурі. GAE автоматично масштабує ресурси в залежності від потреб користувачів, що дозволяє забезпечити високу продуктивність додатків навіть при великому навантаженні.

Третє призначення GAE - це підтримка багатьох мов програмування. GAE підтримує такі мови програмування, як Java, Python, Node.js та інші, що дозволяє розробникам використовувати свої улюблені мови та інструменти розробки для створення веб-додатків.

Четверте призначення GAE - це високий рівень безпеки. GAE має вбудований механізм обмеження доступу та контролювання прав доступу до ресурсів, що дозволяє захистити додатки від несанкціонованого доступу. Крім того, GAE надає додаткові можливості забезпечення безпеки, такі як захист від DDOS-атак та перевірка безпеки на рівні коду.

П'яте призначення GAE - це інтеграція з іншими сервісами Google. GAE дозволяє розробникам інтегрувати свої додатки з іншими сервісами

Google, такими як Google Cloud Storage, Google Cloud SQL та інші. Це дозволяє розробникам легко розширювати функціональність своїх додатків та отримувати доступ до інших корисних сервісів Google.

Останнє призначення GAE - це ефективне використання ресурсів. GAE дозволяє розробникам використовувати лише ті ресурси, які необхідні для роботи їх додатків, що забезпечує ефективне використання ресурсів та зниження витрат на хмарну інфраструктуру.

Основна функціональність GAE включає:

- Розгортання та масштабування додатків: GAE дозволяє розробникам швидко та легко розгорнути свої додатки на серверах Google, забезпечуючи масштабування додатків за потребою. Розробники можуть встановлювати обмеження на кількість запитів, кількість ресурсів та час виконання для кожного запиту. GAE автоматично масштабує додатки відповідно до цих обмежень, забезпечуючи високу доступність та швидкість.
- Сховище даних: GAE надає можливість зберігати дані у вигляді об'єктів в базі даних, або використовувати NoSQL рішення з Google Cloud Datastore. Google Cloud Datastore - це розподілена NoSQL база даних, яка забезпечує високу доступність та швидкість зберігання даних. Розробники можуть використовувати Google Cloud Datastore для зберігання та отримання даних у своїх додатках.
- Керування ресурсами: GAE дозволяє розробникам керувати ресурсами, включаючи CPU, пам'ять та мережеві ресурси, щоб забезпечити оптимальну продуктивність своїх додатків. Розробники можуть встановлювати обмеження на кількість ресурсів, які використовує кожен запущений екземпляр додатку, а також використовувати автоматичне масштабування для динамічного збільшення або зменшення кількості екземплярів додатку в залежності від потреб користувачів.

- **Безпека:** GAE забезпечує високий рівень безпеки для додатків, розгорнутих на його платформі. Всі додатки автоматично захищені від багатьох типів атак, таких як кросс-сайтові скрипти, внедрення SQL та інші. Крім того, GAE включає в себе механізми автоматичного виявлення та блокування потенційно шкідливих дій, що можуть становити загрозу для додатків.
- **Швидкість:** GAE надає швидкість та продуктивність, необхідні для роботи сучасних веб-додатків. Платформа забезпечує ряд оптимізацій, таких як кешування та автоматична оптимізація запитів до баз даних, що дозволяє покращити швидкість роботи додатків. Крім того, GAE надає можливість використовувати CDN-сервіс, що дозволяє розподіляти контент між серверами з метою зменшення навантаження на сервер додатку та покращення швидкості завантаження контенту.
- **Сумісність:** GAE підтримує різні мови програмування та фреймворки, такі як Python, Java, PHP та інші. Це дозволяє розробникам використовувати свої улюблені інструменти та мови програмування для створення додатків на платформі GAE.

Загалом, GAE є потужним та універсальним інструментом для розробки та розгортання веб-додатків. Він забезпечує розробникам легку інтеграцію з іншими сервісами Google, масштабованість додатків, продуктивність та ефективне використання ресурсів. Завдяки цим функціям GAE дозволяє розробникам швидко створювати та розгортати веб-додатки на хмарній платформі, не хвилюючись про управління інфраструктурою.

Окрім цього, GAE забезпечує високий рівень безпеки для розгорнутих додатків. Він включає в себе вбудований механізм обмеження доступу та контролювання прав доступу до ресурсів, що дозволяє захистити додатки від несанкціонованого доступу.

3.4 Розробка та аналіз програмного сервісу

3.4.1 Режим уведення знань експерта

Сервіс, який було реалізовано, має дві частини: Expert view для введення знань і перегляду вже наявних експертних систем зареєстрованим користувачем, та створення нових систем. Процес створення нових ЕС включає кілька етапів. На першому етапі експерт вибирає назву системи і визначає набір ключових слів, за якими користувачі зможуть її знайти. Назва ЕС має бути унікальною і відображати предметну галузь, у якій експерт компетентний. На наступних етапах будуть додані додаткові ключові слова, які будуть пов'язані з системою. Початковий етап створення ЕС представлено на рис. 3.2.

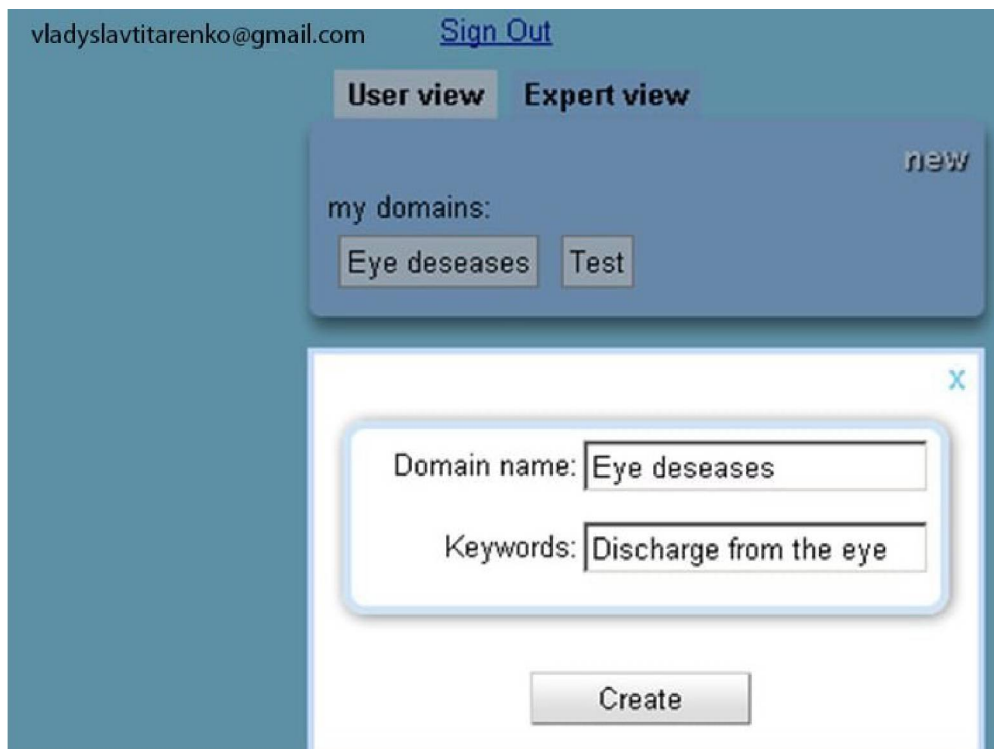


Рис. 3.2. Початковий етап створення ЕС

Основний етап полягає у створенні всіх правил виведення експертної системи за допомогою інтуїтивно зрозумілого інтерфейсу. Для цього використовуються такі елементи:

- вікно з підказкою, що містить список фактів, необхідних для спрацьовування поточного правила (ліва частина продукційного правила). Це вікно допомагає експерту зрозуміти, який висновок необхідно зробити,

маючи представлений список фактів;

- вікно попереднього перегляду, що показує, який вигляд матиме поточна ітерація консультації під час режиму консультації користувачів з урахуванням html-форматування тексту;

- Панель управління експерта, яка включає вікно введення та вікно вибору варіантів відповіді. Вікно введення дозволяє експерту вводити рядок консультації з html-форматуванням тексту. Вікно вибору варіантів відповіді визначає, як користувач може відповісти на поточний рядок консультації експерта. Це може бути стандартний варіант відповіді типу "так" або "ні", або експерт може створити свої варіанти. Якщо обрано стандартний варіант, користувач побачить пару кнопок із відповідною назвою. У разі створення своїх варіантів, буде представлений список, що випадає. На панелі також розміщені кнопки "Preview" для попереднього перегляду і "Next" для переходу до складання наступного правила. Кнопка "Back" дає змогу повернутися до попереднього правила, якщо експерту необхідно його відредагувати.

На рис. 3.3 показано процес створення правил за допомогою представленого інтерфейсу.

Rule No.14 Eye deseases

▼ tooltip

Choose one of the problem areas listed below -
Discharge from the eye

Are you suffer from eyes red ? - yes

Are you suffer from eyes swollen eyelids ? - no

Are you suffer from watery discharge, seasonal allergies, itching eyes ? - no

Expert question:

Are you suffer from fever and sore throat, red rash, begins on face then body, clear discharge from eye ?

Yes No

Possible answers

Yes/No
 Custom variants
 None

Preview
Next

<< Back

- 1: Discharge from the eye
- 2: Bulging Eye
- 3: Double Vision
- 4: Drooping Eyelid

+ -

Finish

переходять до листових вузлів. На рис. 3.5 показано дерево, де вузли q - це рядки консультації, а дуги v - варіанти відповідей користувача. Факти подано у вигляді шаблонів із двома слотами: рядком консультації та відповіддю. Правила містять у лівій частині список таких фактів і в правій частині виведення нового рядка консультації з можливими варіантами відповідей. Якщо правило не виводить можливих варіантів відповідей, то його можна представити у вигляді листового вузла дерева. Таке правило також виводить шаблон зі слотами like і dislike для визначення релевантності рішення експерта. Для цього обчислюють коефіцієнт, який являє собою відношення числа позитивних оцінок до негативних, і якщо результат менший за одиницю, то вживають заходів щодо коригування цього рішення.

Перший рівень дерева, що є коренем, містить правило, яке автоматично спрацьовує без наявності фактів у лівій частині. Потім правила можуть бути складені двома способами: в ширину та в глибину. У випадку з першим, всі правила поточного рівня заповнюються спочатку, після чого відбувається перехід на наступний рівень. У другому випадку завжди спочатку заповнюється найглибший вузол на поточній позиції, тоді як створення правил переходить на найглибший рівень дерева, де вузли не мають дочірніх елементів. На рис. 3.5 вузли нумеруються так, якби вони були заповнені в ширину.

Якщо однакові рядки консультації зустрічаються на різних рівнях правил, то це не є проблемою. Якщо експерт не знає, яку консультацію використати для певного набору фактів, то він може залишити її порожньою, і це правило буде позначено для подальшого редагування.

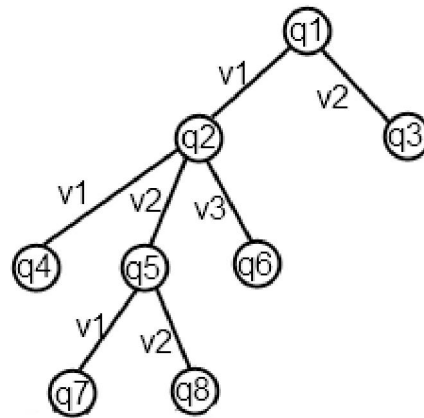


Рис. 3.5. Подання правил в ЕС

Для тестирования сервису було використано фрагмент експертної системи для визначення захворювань очей. При створенні експертної системи було сформовано код з синтаксисом мови JESS. Згенерований текст програми можна знайти у додатку Г.

3.4.2 Режим консультації користувача

У сервисі, призначеному для консультації користувачів (User view), є можливість використовувати експертні системи, які вже існують у базі даних. Для цього користувач вводить ключові слова в поле пошуку, що описують його проблему. Якщо в базі даних є експертні системи, які можуть допомогти користувачеві, вони будуть виведені. Після вибору експертної системи починається процес консультації, як показано на рис. 3.6. Кожному користувачеві виділяється окрема робоча пам'ять для введення фактів.

Після завершення консультації та отримання остаточного рішення від експерта, користувач може зробити оцінку цього рішення. Якщо оцінка виявилася незадовільною, або експерт пропустив якусь частину процесу, то користувачеві будуть доступні дві опції редагування (див. рис. 3.7). Перша опція дає змогу відредагувати виведення рядка консультації поточного правила, і при цьому коефіцієнт оцінювання буде скинуто. Друга опція дозволяє зареєстрованому користувачеві, який має свої експертні системи, доповнити цю частину експертної системи своїми даними. У цьому разі

поточне нерелевантне правило буде замінено першим (кореневим) правилом доданої експертної системи. Крім того, для коректної роботи правил, що вставляються, будуть додані необхідні факти до їхньої лівої частини. Коефіцієнти, що відображають релевантність правил у доданій експертній системі, збержуться. Однак, ці дані не є обов'язковими і можуть бути використані тільки тоді, коли користувач вважає себе достатньо компетентним у цій галузі.

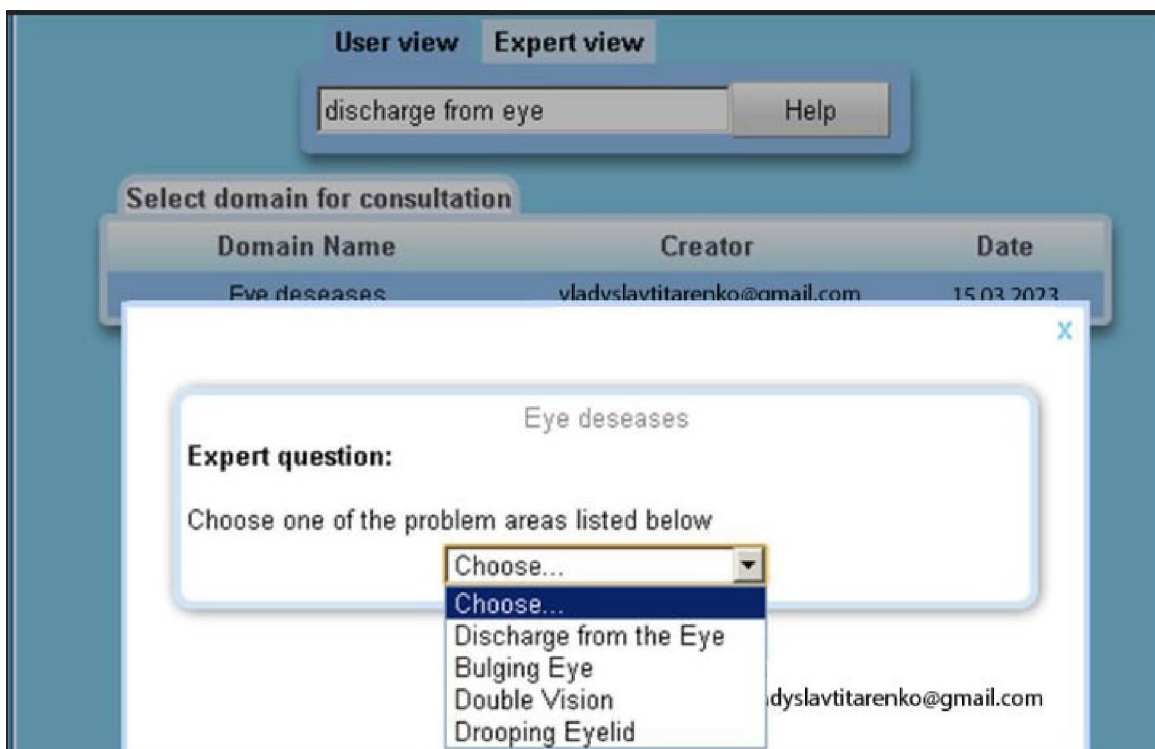


Рис. 3.6. Консультація користувачів

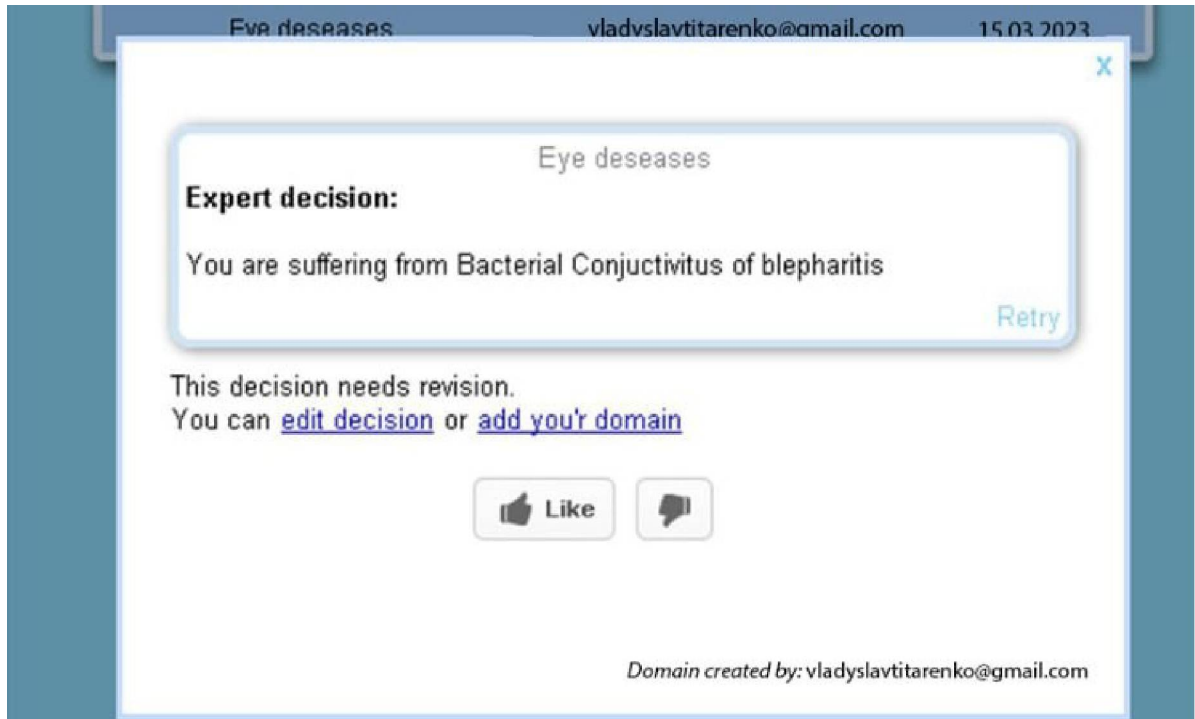


Рис. 3.7. Оцінка рішень експерта

3.4.3 Адміністрування сервісу

Окремий сервіс - консоль адміністрування GAE, який дає змогу відстежувати створювані сервіси.

На головному екрані ви можете побачити графік, який показує, як змінюються різні параметри (як-от запити, помилки та використання ресурсів процесора) залежно від часу (подивіться на малюнок 3.8). Ви можете вибрати часовий інтервал від останніх 30 хвилин роботи сервісу до 30 днів. Нижче ви знайдете інформацію про статус виділених для сервісу квот (подивіться на малюнок 3.9). Квоти оновлюються кожні 24 години. У розділі "Версії" ви можете керувати версіями сервісу.

Розділ "Data" надає можливість керувати даними, які були додані в стандартну базу даних GAE Datastore або Blob. Для цього використовуються підрозділи "Datastore Viewer" і "Blob Viewer" відповідно, як показано на рис. 3.10. У Datastore Viewer можна здійснювати запити до бази даних за допомогою синтаксису GQL, а також редагувати і видаляти отримані результати. Крім того, у цьому розділі можна створювати нові сутності



Instances ⓘ

App Engine Release	Total number of instances	Average QPS ^{**}	Average Latency [*]
Billing Status: Free - Settings			
Resource	Usage		
Frontend Instance Hours	<div style="width: 0%;"></div>	0%	0.00 of 28.00 Instance Hours
Backend Instance Hours	<div style="width: 0%;"></div>	0%	0.00 of 9.00 Instance Hours
Datastore Stored Data	<div style="width: 0%;"></div>	0%	0.00 of 1.00 GBytes
Logs Stored Data	<div style="width: 0%;"></div>	0%	0.00 of 1.00 GBytes
Task Queue Stored Task Bytes	<div style="width: 0%;"></div>	0%	0.00 of 0.49 GBytes
Blobstore Stored Data	<div style="width: 0%;"></div>	0%	0.00 of 5.00 GBytes
Code and Static File Storage	<div style="width: 3%;"></div>	3%	0.03 of 1.00 GBytes
Datastore Write Operations	<div style="width: 0%;"></div>	0%	0.00 of 0.05 Million Ops
Datastore Read Operations	<div style="width: 0%;"></div>	0%	0.00 of 0.05 Million Ops
Datastore Small Operations	<div style="width: 0%;"></div>	0%	0.00 of 0.05 Million Ops
Outgoing Bandwidth	<div style="width: 0%;"></div>	0%	0.00 of 1.00 GBytes
Recipients Emailed	<div style="width: 0%;"></div>	0%	0 of 100
Stanzas Sent	<div style="width: 0%;"></div>	0%	0 of 10,000
Channels Created	<div style="width: 0%;"></div>	0%	0 of 100
Logs Read Bandwidth	<div style="width: 0%;"></div>	0%	0.00 of 0.10 GBytes

Query
Create

By kind: Domain kind: set at 0:00:49 ago Number of Co

[Options](#)

By GQL:

SELECT * FROM Domain

[Learn more about GQL syntax.](#)

Run Query

Domain Entities

< Prev 20
1-2
Next 20 >

<input type="checkbox"/> ID/Name	context	creator	date
<input type="checkbox"/> name=Eye diseases	(import com.cleanproject.project.client.model.Consultation)(import com.cleanproject.project.client.model.Fact)(import com.cleanproject.project.client.model.Ratio)(deftemplate Consultation (declare (from-class Consultation)))(deftemplate Fact (decl...	vladyaslavtitarenko@gmail.com	2023-03-15 14:24:17.627000
<input type="checkbox"/> name=Test	(import com.cleanproject.project.client.model.Consultation)(import com.cleanproject.project.client.model.Fact)(import com.cleanproject.project.client.model.Ratio)(deftemplate Consultation (declare (from-class Consultation)))(deftemplate Fact (decl...	vladyaslavtitarenko@gmail.com	2023-03-15 15:24:17.656000

Delete
Flush Memcache

Performance

Frontend Instance Class: F4_1G (2400MHz, 1024MB)

Adjusting your Frontend Instance Class also consumes frontend instance hours at a rate of 1 hour per instance hour of an F1. [Learn more.](#)

Max Idle Instances: (Automatic)

Adjusting your Max Idle Instances slider allows you to control the number of idle instances available to the default version of your application at any given time. Idle instances are pre-loaded with your application code, so when a new instance is needed, it can serve traffic immediately. You will not be charged for idle instances over the specified maximum. A smaller number of idle instances means your application costs less to run, but may encounter more startup latency during load spikes. [Learn more.](#)

Min
1
50
100
500
Automatic
Max

Min Pending Latency: (Automatic)

The Pending Latency slider controls how long requests spend in the pending queue before being served by an instance of the default version of your application. If the minimum pending latency is high App Engine will allow requests to wait rather than start new instances to process them. This can reduce the number of instance hours your application uses, but can result in more user-visible latency. [Learn more.](#)

Min
Automatic
500ms
1s
7.5s
15s
Max

Висновки за розділом 3

У даному розділі була розглянута розробка програмної реалізації сервісу, який дозволяє розширювати знання в експертних системах з визначення очних захворювань .

ВИСНОВКИ

Завдяки технології хмарних обчислень, інформаційні технології перетворюються на послугу, що вигідно всім учасникам. Компанії можуть скоротити витрати на ІТ і сконцентруватися на розвитку свого бізнесу, переводячи капітальні витрати в операційні. Це також дає малим і середнім компаніям доступ до корпоративних ІТ-технологій. Ринок ІТ і телекомунікацій стимулює розвиток нових бізнес-моделей і стартапів, що сприятливо впливає на економіку країни та суспільство загалом. Хоча існує багато інформації про технологію хмарних обчислень, на ринку спостерігається значна поінформованість щодо PaaS. Більшість компаній впроваджують хмарні технології, щоб запускати інноваційні додатки, такі як програмне забезпечення як послуга (SaaS), мобільні додатки, інтернет-торгівля та соціальні служби. Це робиться різними мовами і допомагає знизити витрати і поліпшити операційну готовність, функціональність і час безвідмовної роботи. Розроблений сервіс на основі технології хмарних обчислень дає змогу створювати, зберігати, розширювати і використовувати експертні системи. Сервіс орієнтований на методику Web 2.0, що забезпечує зростання якості та кількості розроблених експертних систем зі збільшенням кількості користувачів. Це призводить до розширення баз знань і підвищення ймовірності успішного розв'язання складних проблем користувачами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bhowmik, Sandeep. "Cloud Computing." New Delhi, India: Tata McGraw-Hill Education, 2012. Print.
2. Ahson, S.A., & Ilyas, M. (Eds.). Cloud Computing and Software Services: Theory and Techniques. Boca Raton, FL: CRC Press, 2011.
3. Erl, Thomas; Mahmood, Zaigham; Puttini, Ricardo. Cloud Computing: Concepts, Technology & Architecture. Prentice Hall, 2013.
4. Kavis, M. J. and Skilton, M. D. Architecting the Cloud: Design Decisions for Cloud Computing Service Models. John Wiley & Sons, Inc, 2014.
5. Luger G. F. Artificial Intelligence. Structures and Strategies for Complex Problem.
6. Кацадзе Т. Л. Експертні системи, прийняття рішень в енергетиці / Т. Л. Кацадзе. - Київ: Наукова думка, 2004. - 344 с.
7. Solving / George F. Luger. – Pearson Education Limited, 2005. – 903 p.
8. Месюра В. І., Яровий А. А., Арсенюк І. Р. "Експертні системи". Київ: Видавничий дім "Інтерсервіс", 2008. 256 с.
9. Giarratano, J. C., & Riley, G. Expert Systems: Principles and Programming. Sixth Edition. Boston, MA: Cengage Learning, 2018.
10. Friedman-Hill, E. JESS in Action: Java Rule-Based Systems. Greenwich, CT: Manning Publications Co, 2003.
11. Едвардсон, Е. Jess Rules Engine, Second Edition. O'Reilly Media, 2015.
12. Dan Sanderson, Google App Engine Team. Google App Engine: Building Web Applications. O'Reilly Media, 2010.

13.

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) бакалавр
Галузь знань: 12 – Інформаційні технології.
Спеціальність 123 – Комп'ютерна інженерія.

ЗАТВЕРДЖУЮ

**Завідувач кафедри теоретичної
та прикладної системотехніки
д.т.н., проф. Шматков С. І.**



«17» листопада 2022 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Титаренко Владислава Сергійовича
(прізвище, ім'я, по батькові студента)

1. Тема роботи **«Моделювання експертних систем на основі сервісно-орієнтованого підходу та cloud computing»**
керівник роботи Павлов Анатолій Миколайович, старший викладач
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «23»_05_2023_року №_4101-5/895_

2. Строк подання студентом роботи 26 травня 2023
3. Перелік питань, які потрібно розробити
1. Аналіз головних факторів розвитку хмарних технологій.
 2. Cloud computing у даний час.
 3. Аналіз експертних систем.
 4. Принципи роботи JESS.
 5. Google App Engine (основні функції та призначення).

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Аналіз предметної області та пошук літератури	Листопад 2022
2	Аналіз головних факторів розвитку хмарних технологій та розгляд cloud computing	Листопад 2022- Грудень 2022
3	Аналіз експертних систем : призначення, структура та найбільш поширені експертні системи	Січень 2022
4	Реалізація сервісу щодо розширення знань у різноманітних сферах за допомогою експертних систем	Січень 2023
5	Оформлення пояснювальної записки	Травень 2023
6	Перед захист кваліфікаційної роботи	Травень 2023
7	Представлення кваліфікаційної роботи керівнику та рецензенту	Травень 2023

5. Дата видачі завдання 1 листопада 2022

Студент

Титаренко В. С.

ініціали, прізвище



підпис

Керівник роботи

Павлов А. М.

ініціали, прізвище



підпис

Додаток Б

Затверджую

« » 2023 р.

**Технічне завдання
на розробку програмного виробу «Модель експертних систем на
основі сервісно-орієнтованого підходу та cloud computing»**

1.	Введення	1.1. Назва: Модель експертних систем на основі сервісно-орієнтованого підходу та cloud computing. 1.2. Галузь застосування: Інформаційні технології
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп'ютерна інженерія 2.2. Завдання на кваліфікаційну роботу бакалавра № <u> 4101-5/895 </u> від « <u> 23 </u> » <u> 05 </u> 2023 (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3.	Призначення розробки	3.1. Мета розробки: створення сервісу для розробки та використання експертних систем, який базується на сервісно-орієнтованому підході та використовує cloud computing. 3.2. Призначення розробки надає можливість використання облака для збереження та обробки даних. Передбачає використання сучасних технологій та підходів для розробки ефективних експертних систем, які здатні підвищити швидкість прийняття рішень, зменшити кількість необхідних датчиків та підвищити точність діагностування. 3.3. Вихідні дані розробки: сервіс експертної системи з визначення очних захворювань
4.	Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: підвищення швидкості прийняття рішень та точності діагностування шляхом визначення найбільш інформативних параметрів стану систем та зменшення кількості необхідних датчиків. 4.2. Вимоги до надійності: забезпечення безперебійної роботи програмного виробу при будь-яких вимогах користувача в рамках призначення виробу . 4.3.Вимоги до умов експлуатації: немає 4.4. Вимоги до складу і параметрів технічних засобів: для виконання програми повинен підходити ПК із

		<p>будь-якою операційною системою сімейства Windows, Linux/Unix, Mac OS X, OS/2, Amiga. Крім того, для роботи потрібний інтерпретатор мови програмування.</p> <p>4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 10, підтримка мови програмування, підтримка різних платформ.</p> <p>4.6. Вимоги до маркування та упаковки: вимоги до маркування та упакування не представляються.</p> <p>4.7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються.</p> <p>4.8. Спеціальні вимоги: спеціальні вимоги до програмного виробу не пред'являються.</p>	
5.	Вимоги до програмної документації	<p>Програмною документацією до виробу «Метод аналізу інформативності змінних стану при діагностиці систем з використанням інформаційних критеріїв» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Методику розрахунку інформативності змінних стану (у вигляді глав 3.3 та 3.4 пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Опис виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)</p>	
6.	Вимоги до техніко-економічних показників	<p>Програмною документацією до виробу «Моделювання експертних систем на основі сервісно-орієнтованого підходу та cloud computing» вважати:</p> <p>1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).</p> <p>2) Опис програмного виробу (представити в Розділі 2 пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Джерела базової інформації.</p>	
7.	Стадії і етапи розробки	Дата	Назва етапу
		від 25 листопада 2022 до 16 грудня 2022	Аналіз предметної області та пошук літератури.
		від 25 листопада 2022 до 25 грудня 2023	Аналіз головних факторів розвитку хмарних технологій та розгляд cloud computing.
			Аналіз експертних систем : призначення, структура та

		<p>від 26 грудня 2023 до 26 лютого 2023</p> <p>від 27 лютого 2023 до 2 квітня 2023</p> <p>від 3 квітня 2023 до 10 травня 2023</p> <p>від 11 травня 2023 до 1 червня 2023</p>	<p>найбільш поширені експертні системи.</p> <p>Реалізація сервісу щодо розширення знань у різноманітних сферах за допомогою експертних систем.</p> <p>Оформлення пояснювальної записки.</p> <p>Представлення кваліфікаційного проєкту керівнику кваліфікаційної роботи та рецензенту.</p>
8.	Порядок контролю і приймання програмного продукту (моделі)	<ol style="list-style-type: none"> 1. Перевірку ходу розробки програми виконувати раз в 3 тижні. 2. захист розробленої моделі провести на засіданні Атестаційної комісії. 3. Пояснювальну записку подати на паперових носіях в 1 примірнику і в електронному вигляді в 1 примірнику на CD-R компакт-диску. 	

Виконавець
студент групи КІ- 41
Титаренко В. С.



Замовник
старший викладач
Павлов А. М.



Додаток В

Програма і методика випробувань програмного виробу

«Модель експертних систем на основі сервісно-орієнтованого підходу та cloud computing»

1 Об'єкт випробувань

1. Назва програмного виробу : «Модель експертних систем на основі сервісно-орієнтованого підходу та cloud computing»
2. Галузь застосування : Інформаційні технології
3. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності функціональності програмної реалізації системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення**1. Підстави для проведення випробувань**

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

2. Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

- 4.1. Вимоги до функціональних характеристик: підвищення швидкості прийняття рішень та точності діагностування шляхом визначення найбільш

інформативних параметрів стану систем та зменшення кількості необхідних датчиків

4.2. Вимоги до надійності: забезпечення безперебійної роботи програмного виробу при будь-яких вимогах користувача в рамках призначення виробу .

4.3. Вимоги до умов експлуатації: немає.

4.4. Вимоги до складу і параметрів технічних засобів: для виконання програми повинен підходити ПК із будь-якою операційною системою сімейства Windows, Linux/Unix, Mac OS X, OS/2, Amiga. Крім того, для роботи потрібний інтерпретатор мови програмування.

4.5. Вимоги до інформаційної та програмної сумісності: підтримка ОС Linux або Windows 10, підтримка мови програмування, підтримка різних платформ.

4.6. Вимоги до маркування та упаковки: вимоги до маркування та упаковки не представляються.

4.7. Вимоги до транспортування і зберігання: вимоги до транспортування та зберігання не представляються.

4.8. Спеціальні вимоги: спеціальні вимоги до програмного виробу не пред'являються.

Спеціальні вимоги (не пред'являються).

5. Вимоги до програмної документації

Документацією до виробу «Модель оцінки ризиків при розробці коригуючих рішень в процесі управління IT-проектом» вважати:

- 1) Документація по мові програмування та додаткові мануали.
- 2) Програму і методику випробувань розробленої програми (представити як Додаток В до пояснювальної записки до кваліфікаційної роботи).
- 3) Опис програмного виробу (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи).
- 4) Джерела базової інформації.

програмні засоби: інтерпретатор мови програмування.

6.2 Порядок проведення випробувань

Це правило випробування складається з трьох етапів:

Тест 1

1. Перевірка виконання програми;
2. Створення правил ЕС;
3. Отримання відповіді серверу про успішне створення об'єкту.

The screenshot shows a web browser window with a title bar that reads "Expert System - Eye diseases". The main content area is titled "Rule No.14 Eye diseases". On the left, there is a section labeled "Icd10" with a dropdown arrow. Below it, the text says "Choose one of the problem areas listed below - Discharge from the eye". Three radio button options are listed: "Are you suffer from eyes red ? - yes", "Are you suffer from eyes swollen eyelids ? - no", and "Are you suffer from watery discharge, seasonal allergies, itching eyes ? - no". On the right, a box titled "Expert question:" contains the text "Are you suffer from fever and sore throat, red rash, begins on face then body, clear discharge from eye ?" with "Yes" and "No" buttons below it. At the bottom, there is a rich text editor with a toolbar containing icons for bold, italic, underline, text color, background color, bulleted list, numbered list, link, unlink, indent, outdent, undo, redo, and print. The text area of the editor contains the same expert question. To the right of the editor is a "Possible answers" panel with three radio button options: "Yes/No" (which is selected), "Custom variants", and "None". Below these options are "Preview" and "Next" buttons. At the bottom right of the entire interface is a "<< Back" button.

Рис. В.1 Тест 1

Тест 2

1. Перевірка виконання програми
2. Консультація користувачів;
3. Отримання результатів.
- 4) Якщо перевірки на першому та другому етапах виконано успішно, то виріб вважається таким, що пройшов випробування.

Для проведення випробувань пропонується тест 1, тест 2 та тест 3.

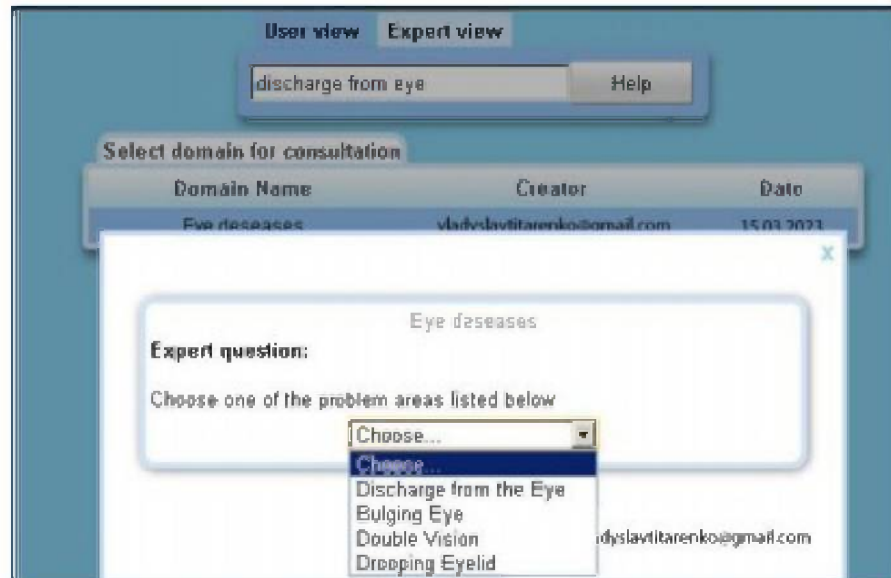


Рис. В.2 Тест 2

Тест 3

4. Перевірка виконання програми
5. Огляд Datastore;
6. Отримання результату.

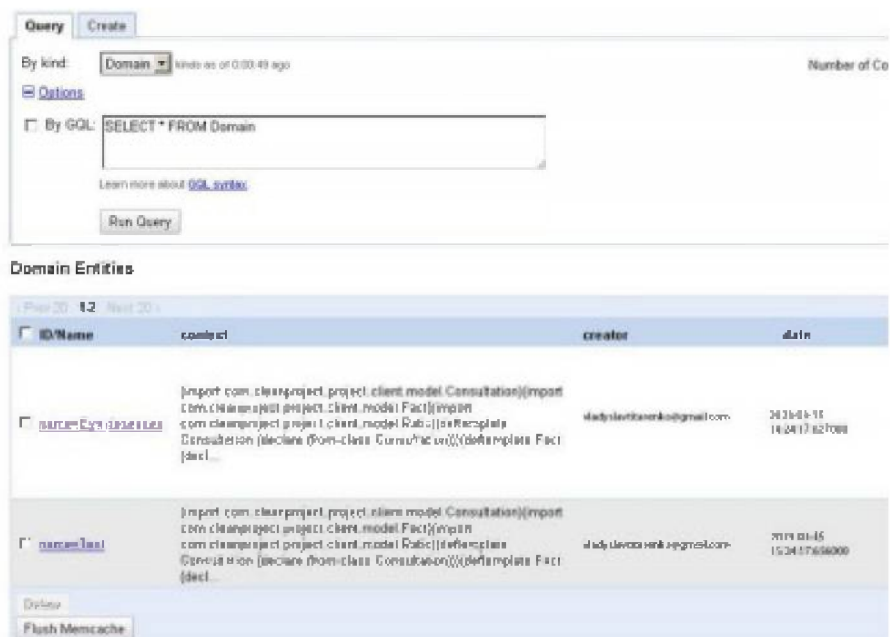


Рис. В.3 Тест 3

Тест вважається пройденим, якщо відбуваються вказані операції і їх відображення у програмному продукті.

Висновки: тест 1 успішно пройшов випробування, тест 2 успішно пройшов випробування і тест 3 успішно пройшов випробування. Випробування пройшло успішно.

Виконавець: студент групи KI-41, Титаренко В. С.

Додаток Г

JESS-код експертної системи з визначення очних захворювань «Моделювання експертних систем на основі сервісно-орієнтованого підходу та cloud computing»

```
(import com.cleanproject.project.client.model.Consultation)
(import com.cleanproject.project.client.model.Fact)
(import com.cleanproject.project.client.model.Ratio)
(deftemplate Consultation (declare (from-class Consultation)))
(deftemplate Fact (declare (from-class Fact)))
(deftemplate Ratio (declare (from-class Ratio)))
(defrule Rule0(Fact(question «»)(answer «»))=>(bind ?1 (new
java.util.ArrayList))(call ?1 add «Discharge from the Eye»)(call
?1 add «Bulging Eye»)(call ?1 add «Double Vision»)(call ?1 add
«Drooping Eyelid»)(add(new Consultation Rule0 «Choose one of the
problem areas listed below» ?1)))
(defrule Rule1(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))=>(bind ?1 (new
java.util.ArrayList))(call ?1 add «yes»)(call ?1 add
«no»)(add(new Consultation Rule1 «Are you suffer from eyes red
?» ?1)))
(defrule Rule2(Fact(question «Choose one of the problem areas
listed below»)(answer «Bulging Eye»))=>(bind ?1 (new
java.util.ArrayList))(add(new Consultation Rule2 «» ?1))(add(new
Ratio 1 1)))
(defrule Rule3(Fact(question «Choose one of the problem areas
listed below»)(answer «Double Vision»))=>(bind ?1 (new
java.util.ArrayList))(add(new Consultation Rule3 «» ?1))(add(new
Ratio 1 1)))
(defrule Rule4(Fact(question «Choose one of the problem areas
listed below»)(answer «Drooping Eyelid»))=>(bind ?1 (new
java.util.ArrayList))(add(new Consultation Rule4 «» ?1))(add(new
Ratio 1 1)))
(defrule Rule5(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «yes»))=>(bind ?1 (new
java.util.ArrayList))(call ?1 add «yes»)(call ?1 add
«no»)(add(new Consultation Rule5 «Are you suffer from eyes
swollen eyelids ?» ?1)))
(defrule Rule6(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))=>(bind ?1 (new
java.util.ArrayList))(call ?1 add «yes»)(call ?1 add
«no»)(add(new Consultation Rule6 «Are you suffer from swelling
of eyelids, tongue and face?» ?1)))
(defrule Rule7(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «yes»))(Fact(question
```

```

«Are you suffer from eyes swollen eyelids ?»)(answer
«yes»))=>(bind ?1 (new java.util.ArrayList))(call ?1 add
«yes»)(call ?1 add «no»)(add(new Consultation Rule7 «Are you
suffer from pus like discharge and crusting of eyelids on
awakening ?» ?1)))
(defrule Rule8(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «yes»))(Fact(question
«Are you suffer from eyes swollen eyelids ?»)(answer
«no»))=>(bind ?1 (new java.util.ArrayList))(call ?1 add
«yes»)(call ?1 add «no»)(add(new Consultation Rule8 «Are you
suffer from watery discharge, seasonal allergies, itching eyes
?» ?1)))
(defrule Rule9(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))(Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «yes»))=>(bind ?1 (new java.util.ArrayList))(call
?1 add «yes»)(call ?1 add «no»)(add(new Consultation Rule9 «Are
you suffer from sweating, use of new medication, insect bite,
new foods, rapid heart beat ?» ?1)))
(defrule Rule10(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))(Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»))=>(bind ?1 (new java.util.ArrayList))(call
?1 add «yes»)(call ?1 add «no»)(add(new Consultation Rule10 «Are
you suffer from eyes red with watery discharge ?» ?1)))
(defrule Rule11(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «yes»))(Fact(question
«Are you suffer from eyes swollen eyelids ?»)(answer
«yes»))(Fact(question «Are you suffer from pus like discharge
and crusting of eyelids on awakening ?»)(answer «yes»))=>(bind
?1 (new java.util.ArrayList))(add(new Consultation Rule11 «You
are suffering from Bacterial Conjunctivitus of blepharitis»
?1))(add(new Ratio 1 2)))
(defrule Rule12(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «yes»))(Fact(question
«Are you suffer from eyes swollen eyelids ?»)(answer
«yes»))(Fact(question «Are you suffer from pus like discharge
and crusting of eyelids on awakening ?»)(answer «no»))=>(bind ?1
(new java.util.ArrayList))(call ?1 add «yes»)(call ?1 add
«no»)(add(new Consultation Rule12 «Are you suffer from sticky,
crusty eyelids, red itchy eyelids, loss of eyelashes ?» ?1)))
(defrule Rule13(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «yes»))(Fact(question
«Are you suffer from eyes swollen eyelids ?»)(answer
«no»))(Fact(question «Are you suffer from watery discharge,
seasonal allergies, itching eyes ?»)(answer «yes»))=>(bind ?1

```

```

(new java.util.ArrayList)) (add(new Consultation Rule13 «You are
suffering from Allergic Conjunctivitus OR viral conjunctivitis»
?1)) (add(new Ratio 1 1)))
(defrule Rule14 (Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»)) (Fact(question
«Are you suffer from eyes red ?»)(answer «yes»)) (Fact(question
«Are you suffer from eyes swollen eyelids ?»)(answer
«no»)) (Fact(question «Are you suffer from watery discharge,
seasonal allergies, itching eyes ?»)(answer «no»))=>(bind ?1
(new java.util.ArrayList)) (call ?1 add «yes») (call ?1 add
«no») (add(new Consultation Rule14 «Are you suffer from fever and
sore throat, red rash, begins on face then body, clear discharge
from eye ?» ?1)))
(defrule Rule15 (Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»)) (Fact(question
«Are you suffer from eyes red ?»)(answer «no»)) (Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «yes»)) (Fact(question «Are you suffer from
sweating, use of new medication, insect bite, new foods, rapid
heart beat ?»)(answer «yes»))=>(bind ?1 (new
java.util.ArrayList)) (add(new Consultation Rule15 «You are
suffering from Allergy OR Anaphylaxis» ?1)) (add(new Ratio 1 1)))
(defrule Rule16 (Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»)) (Fact(question
«Are you suffer from eyes red ?»)(answer «no»)) (Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «yes»)) (Fact(question «Are you suffer from
sweating, use of new medication, insect bite, new foods, rapid
heart beat ?»)(answer «no»))=>(bind ?1 (new
java.util.ArrayList)) (add(new Consultation Rule16 «You should be
sure for you symptoms .. Try again...» ?1)) (add(new Ratio 1 1)))
(defrule Rule17 (Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»)) (Fact(question
«Are you suffer from eyes red ?»)(answer «no»)) (Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»)) (Fact(question «Are you suffer from eyes
red with watery discharge ?»)(answer «yes»))=>(bind ?1 (new
java.util.ArrayList)) (call ?1 add «yes») (call ?1 add
«no») (add(new Consultation Rule17 «Are you suffer from headache
centered around eye, headaches usually lasting 15-30 minutes,
more common in men, nasal discharge ?» ?1)))
(defrule Rule18 (Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»)) (Fact(question
«Are you suffer from eyes red ?»)(answer «no»)) (Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»)) (Fact(question «Are you suffer from eyes
red with watery discharge ?»)(answer «no»))=>(bind ?1 (new
java.util.ArrayList)) (call ?1 add «yes») (call ?1 add
«no») (add(new Consultation Rule18 «Are you suffer from blurred
vision ?» ?1)))
(defrule Rule19 (Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»)) (Fact(question

```

```

«Are you suffer from eyes red ?»)(answer «yes»))(Fact(question
«Are you suffer from eyes swollen eyelids ?»)(answer
«yes»))(Fact(question «Are you suffer from pus like discharge
and crusting of eyelids on awakening ?»)(answer
«no»))(Fact(question «Are you suffer from sticky, crusty
eyelids, red itchy eyelids, loss of eyelashes ?»)(answer
«yes»))=>(bind ?l (new java.util.ArrayList))(add(new
Consultation Rule19 «You are suffering from Blepharitis OR
Conjunctivitis OR Stye.» ?l))(add(new Ratio 1 1)))
(defrule Rule20(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «yes»))(Fact(question
«Are you suffer from eyes swollen eyelids ?»)(answer
«yes»))(Fact(question «Are you suffer from pus like discharge
and crusting of eyelids on awakening ?»)(answer
«no»))(Fact(question «Are you suffer from sticky, crusty
eyelids, red itchy eyelids, loss of eyelashes ?»)(answer
«no»))=>(bind ?l (new java.util.ArrayList))(call ?l add
«yes»)(call ?l add «no»)(add(new Consultation Rule20 «Are you
suffer from watery discharge, fever and sore throat, swollen
glands in neck ?» ?l)))
(defrule Rule21(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «yes»))(Fact(question
«Are you suffer from eyes swollen eyelids ?»)(answer
«no»))(Fact(question «Are you suffer from watery discharge,
seasonal allergies, itching eyes ?»)(answer «no»))(Fact(question
«Are you suffer from fever and sore throat, red rash, begins on
face then body, clear discharge from eye ?»)(answer
«yes»))=>(bind ?l (new java.util.ArrayList))(add(new
Consultation Rule21 «You are suffering from Measles OR Viral
Conjunctivitis.» ?l))(add(new Ratio 1 1)))
(defrule Rule22(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «yes»))(Fact(question
«Are you suffer from eyes swollen eyelids ?»)(answer
«no»))(Fact(question «Are you suffer from watery discharge,
seasonal allergies, itching eyes ?»)(answer «no»))(Fact(question
«Are you suffer from fever and sore throat, red rash, begins on
face then body, clear discharge from eye ?»)(answer
«no»))=>(bind ?l (new java.util.ArrayList))(add(new Consultation
Rule22 «You should be sure for your symptoms.. Try again...»
?l))(add(new Ratio 1 1)))
(defrule Rule23(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))(Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»))(Fact(question «Are you suffer from eyes
red with watery discharge ?»)(answer «yes»))(Fact(question «Are
you suffer from headache centered around eye, headaches usually
lasting 15-30 minutes, more common in men, nasal discharge
?»)(answer «yes»))=>(bind ?l (new java.util.ArrayList))(add(new

```

```

Consultation Rule23 «You are suffering from Cluster Headache OR
Migraine Headache OR Allergic Rhinitis.» ?1)) (add(new Ratio 1
1)))
(defrule Rule24 (Fact(question «Choose one of the problem areas
listed below»)) (answer «Discharge from the Eye»)) (Fact(question
«Are you suffer from eyes red ?»)) (answer «no»)) (Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)) (answer «no»)) (Fact(question «Are you suffer from eyes
red with watery discharge ?»)) (answer «yes»)) (Fact(question «Are
you suffer from headache centered around eye, headaches usually
lasting 15-30 minutes, more common in men, nasal discharge
?»)) (answer «no»)) => (bind ?1 (new java.util.ArrayList)) (call ?1
add «yes») (call ?1 add «no») (add(new Consultation Rule24 «Are
you suffer from more common in elderly people, may develop pus
like drainage from eye?» ?1)))
(defrule Rule25 (Fact(question «Choose one of the problem areas
listed below»)) (answer «Discharge from the Eye»)) (Fact(question
«Are you suffer from eyes red ?»)) (answer «no»)) (Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)) (answer «no»)) (Fact(question «Are you suffer from eyes
red with watery discharge ?»)) (answer «no»)) (Fact(question «Are
you suffer from blurred vision ?»)) (answer «yes»)) => (bind ?1 (new
java.util.ArrayList)) (call ?1 add «yes») (call ?1 add
«no») (add(new Consultation Rule25 «Are you suffer from pain and
redness, excessive tearing, sensitivity to light, feeling of
foreign body in eye ?» ?1)))
(defrule Rule26 (Fact(question «Choose one of the problem areas
listed below»)) (answer «Discharge from the Eye»)) (Fact(question
«Are you suffer from eyes red ?»)) (answer «no»)) (Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)) (answer «no»)) (Fact(question «Are you suffer from eyes
red with watery discharge ?»)) (answer «no»)) (Fact(question «Are
you suffer from blurred vision ?»)) (answer «no»)) => (bind ?1 (new
java.util.ArrayList)) (call ?1 add «yes») (call ?1 add
«no») (add(new Consultation Rule26 «Are you suffer from weakness
of one side of the face, recent viral illness, drooping of lower
eyelid, difficulty closing eyelid, discharge from the eye ?»
?1)))
(defrule Rule27 (Fact(question «Choose one of the problem areas
listed below»)) (answer «Discharge from the Eye»)) (Fact(question
«Are you suffer from eyes red ?»)) (answer «yes»)) (Fact(question
«Are you suffer from eyes swollen eyelids ?»)) (answer
«yes»)) (Fact(question «Are you suffer from pus like discharge
and crusting of eyelids on awakening ?»)) (answer
«no»)) (Fact(question «Are you suffer from sticky, crusty
eyelids, red itchy eyelids, loss of eyelashes ?»)) (answer
«no»)) (Fact(question «Are you suffer from watery discharge,
fever and sore throat, swollen glands in neck ?»)) (answer
«yes»)) => (bind ?1 (new java.util.ArrayList)) (add(new
Consultation Rule27 «You are suffering from Viral Conjunctivitis
OR allergic conjunctivitis» ?1)) (add(new Ratio 1 1)))

```

```

(defrule Rule28 (Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «yes»))(Fact(question
«Are you suffer from eyes swollen eyelids ?»)(answer
«yes»))(Fact(question «Are you suffer from pus like discharge
and crusting of eyelids on awakening ?»)(answer
«no»))(Fact(question «Are you suffer from sticky, crusty
eyelids, red itchy eyelids, loss of eyelashes ?»)(answer
«no»))(Fact(question «Are you suffer from watery discharge,
fever and sore throat, swollen glands in neck ?»)(answer
«no»))=>(bind ?l (new java.util.ArrayList))(add(new Consultation
Rule28 «You should be sure for your symptoms.. Try again...»
?l))(add(new Ratio 1 1)))
(defrule Rule29 (Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))(Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»))(Fact(question «Are you suffer from eyes
red with watery discharge ?»)(answer «yes»))(Fact(question «Are
you suffer from headache centered around eye, headaches usually
lasting 15-30 minutes, more common in men, nasal discharge
?»)(answer «no»))(Fact(question «Are you suffer from more common
in elderly people, may develop pus like drainage from
eye?»)(answer «yes»))=>(bind ?l (new
java.util.ArrayList))(add(new Consultation Rule29 «You are
suffering from Dacryocystitis OR Styte OR Conjunctivitis»
?l))(add(new Ratio 1 1)))
(defrule Rule30 (Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))(Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»))(Fact(question «Are you suffer from eyes
red with watery discharge ?»)(answer «yes»))(Fact(question «Are
you suffer from headache centered around eye, headaches usually
lasting 15-30 minutes, more common in men, nasal discharge
?»)(answer «no»))(Fact(question «Are you suffer from more common
in elderly people, may develop pus like drainage from
eye?»)(answer «no»))=>(bind ?l (new
java.util.ArrayList))(add(new Consultation Rule30 «You should be
sure for your symptoms.. Try again...» ?l))(add(new Ratio 1 1)))
(defrule Rule31 (Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))(Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»))(Fact(question «Are you suffer from eyes
red with watery discharge ?»)(answer «no»))(Fact(question «Are
you suffer from blurred vision ?»)(answer «yes»))(Fact(question
«Are you suffer from pain and redness, excessive tearing,
sensitivity to light, feeling of foreign body in eye ?»)(answer
«yes»))=>(bind ?l (new java.util.ArrayList))(add(new
Consultation Rule31 «You are suffering from Corneal Abrasion OR
Corneal Ulcer.» ?l))(add(new Ratio 1 1)))

```

```

(defrule Rule32(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))(Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»))(Fact(question «Are you suffer from eyes
red with watery discharge ?»)(answer «no»))(Fact(question «Are
you suffer from blurred vision ?»)(answer «yes»))(Fact(question
«Are you suffer from pain and redness, excessive tearing,
sensitivity to light, feeling of foreign body in eye ?»)(answer
«no»))=>(bind ?l (new java.util.ArrayList))(call ?l add
«yes»)(call ?l add «no»)(add(new Consultation Rule32 «Are you
suffer from eyes red lump on the edge of eyelid, lump may drain
pus, painful lump ?» ?l)))
(defrule Rule33(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))(Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»))(Fact(question «Are you suffer from eyes
red with watery discharge ?»)(answer «no»))(Fact(question «Are
you suffer from blurred vision ?»)(answer «no»))(Fact(question
«Are you suffer from weakness of one side of the face, recent
viral illness, drooping of lower eyelid, difficulty closing
eyelid, discharge from the eye ?»)(answer «yes»))=>(bind ?l (new
java.util.ArrayList))(add(new Consultation Rule33 «You are
suffering from Bell's Palsy OR Multiple Sclerosis OR Stroke OR
Acoustic Neuroma...» ?l))(add(new Ratio 1 1)))
(defrule Rule34(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))(Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»))(Fact(question «Are you suffer from eyes
red with watery discharge ?»)(answer «no»))(Fact(question «Are
you suffer from blurred vision ?»)(answer «no»))(Fact(question
«Are you suffer from weakness of one side of the face, recent
viral illness, drooping of lower eyelid, difficulty closing
eyelid, discharge from the eye ?»)(answer «no»))=>(bind ?l (new
java.util.ArrayList))(add(new Consultation Rule34 «You should be
sure for your symptoms.. Try again...» ?l))(add(new Ratio 1 1)))
(defrule Rule35(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))(Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»))(Fact(question «Are you suffer from eyes
red with watery discharge ?»)(answer «no»))(Fact(question «Are
you suffer from blurred vision ?»)(answer «yes»))(Fact(question
«Are you suffer from pain and redness, excessive tearing,
sensitivity to light, feeling of foreign body in eye ?»)(answer
«no»))(Fact(question «Are you suffer from eyes red lump on the
edge of eyelid, lump may drain pus, painful lump ?»)(answer
«yes»))=>(bind ?l (new java.util.ArrayList))(add(new
Consultation Rule35 «You are suffering from Styte OR
Chalazion...» ?l))(add(new Ratio 1 1)))

```

```

(defrule Rule36(Fact(question «Choose one of the problem areas
listed below»)(answer «Discharge from the Eye»))(Fact(question
«Are you suffer from eyes red ?»)(answer «no»))(Fact(question
«Are you suffer from swelling of eyelids, tongue and
face?»)(answer «no»))(Fact(question «Are you suffer from eyes
red with watery discharge ?»)(answer «no»))(Fact(question «Are
you suffer from blurred vision ?»)(answer «yes»))(Fact(question
«Are you suffer from pain and redness, excessive tearing,
sensitivity to light, feeling of foreign body in eye ?»)(answer
«no»))(Fact(question «Are you suffer from eyes red lump on the
edge of eyelid, lump may drain pus, painful lump ?»)(answer
«no»))=>(bind ?l (new java.util.ArrayList))(add(new Consultation
Rule36 «You should be sure for your symptoms.. Try again...»
?l))(add(new Ratio 1 1))

```