

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія для прийняття рішень щодо відповідності вибору подарунка до вподобань працівника»

Виконала: студентка 2-го курсу, групи
ІКН-22м спеціальності 122

«Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Панасюк Н. А.

(прізвище та ініціали)

Керівник: к.т.н., доц кафедри КН

Барабан С. В.

(прізвище та ініціали)

« 07 » 12 2023 р.

Опонент:

Хмільчук М. С.

(прізвище та ініціали)

« 07 » 12 2023 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А. А.

(прізвище та ініціали)

« 08 » 12 2023 р.

Вінниця ВНТУ - 2023 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо-професійна програма – Системи штучного інтелекту

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
д.т.н., проф. Яровий А.А.

(підпис)

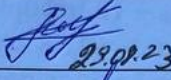
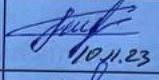
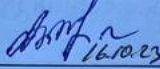
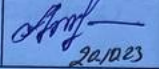
« 29 » 08 2023 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Панасюк Надії Андріївни
(прізвище, ім'я, по батькові)

1. Тема роботи «Інформаційна технологія для прийняття рішень щодо відповідності вибору подарунка до вподобань працівника»
керівник роботи к.т.н., Барабан Сергій Володимирович.
затверджені наказом вищого навчального закладу від «18» 09 2023 року № 247
2. Строк подання студентом роботи 13. 11. 2023 року
3. Вихідні дані до роботи: мова програмування – об'єктно-орієнтована; мінімальна кількість питань до користувача – 10; мінімальна кількість записів в базі даних – 30; мінімальна швидкість доступу до даних – 120 мкс; тип інтерфейсу користувача інтуїтивно-зрозумілий.
4. Зміст текстової частини: вступ, обґрунтування доцільності розробки інформаційної технології для прийняття рішень щодо відповідності вибору подарунка до вподобань працівника, моделювання інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника, структурна організація та особливості програмної реалізації інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника, економічна частина, висновки, список використаних джерел, додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): структурна схема інформаційної технології; загальна структурна схема; діаграма варіантів використання; схема загального алгоритму функціонування інформаційної технології; схема алгоритму роботи інформаційної технології для ролі працівника; схема формування запитань; UML-діаграма класів; приклад роботи програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціалита посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3 Спеціальна частина	Барабан С. В. к.т.н., доц. каф. КН	 29.09.23	 10.11.23
4 Економічна частина	Адлер О. О. к. т. н., доц. каф. ЕПВМ	 16.10.23	 20.10.23


7. Дата видачі завдання 29.08 2023 року

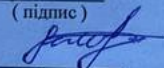
КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Обґрунтування доцільності розробки інформаційної технології для прийняття рішень щодо відповідності вибору подарунка до вподобань працівника	07.09.23 - 07.09.23	Розділ 1
2	Моделювання інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника	08.09.23 - 24.09.23	Розділ 2
3	Структурна організація та особливості програмної реалізації інформаційної технології	25.09.23 - 15.10.23	Розділ 3
4	Підготовка економічної частини	16.10.23 - 20.10.23	Розділ 4
5	Апробація та/або впровадження результатів дослідження	21.10.23 - 05.11.23	Тези доповідей
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23 - 10.11.23	Пояснювальна записка, графічний матеріал, презентація

Студент

Керівник роботи


(підпис)


(підпис)

Панасюк Н.А.

Барабан С. В.

АНОТАЦІЯ

Панасюк Н. А. Інформаційна технологія надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника. Магістерська кваліфікаційна робота зі спеціальності 122 – Комп'ютерні науки, освітня програма «Системи штучного інтелекту». Вінниця: ВНТУ, 2023. 122 с.

На укр. мові. Бібліогр.: 34 назв; рис.: 20; табл. 9.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника. Виконано аналіз сучасних аналогів, які використовуються для надання рекомендації стосовно вибору подарунків відповідно до вподобань людини, наведено коротку порівняльну характеристику знайдених програм. Проведено обґрунтування вибору методу для надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника та описано математичну модель на основі поєднання методів аналізу ієрархій та нечіткого дерева рішень. Розроблено алгоритм прийняття рішень щодо відповідності вибору подарунка до вподобань працівника та відповідне програмне забезпечення мовою програмування Java. Аналіз роботи програмного забезпечення показав підвищення ефективності надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.

У розділі економічної частини здійснено оцінювання комерційного потенціалу розробки інформаційної технології прогнозування обсягу закупівлі квітів, проведено оцінювання комерційного потенціалу розробки, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, розраховано період окупності.

Ілюстративна частина складається з 7 плакатів із результатами моделювання.

Ключові слова: інформаційна технологія, прийняття рішень, надання рекомендацій, подарунки.

ABSTRACT

Panasiuk N. A. Information technology for providing recommendations for decision-making regarding the suitability of the choice of gift to the employee's preferences. Master's work in specialty 122 «Artificial intelligence systems». Vinnytsia: VNTU, 2023. 122 c.

In Ukrainian language. Bibliogr.: 34 titles; Figures: 20; Table 9.

This master's qualification work is devoted to the the development of information technology for providing recommendations for decision-making regarding the conformity of the choice of gift to the employee's preferences. An analysis of modern analogues, which are used to provide recommendations regarding the choice of gifts according to a person's preferences, is performed, and a brief comparative description of the found programs is given. The justification of the choice of the method for providing recommendations for decision-making regarding the correspondence of the choice of a gift to the employee's experiences is carried out, and a mathematical model based on a combination of methods of analysis of hierarchies and a fuzzy decision tree is described. A decision-making algorithm for matching the choice of a gift to the employee's experiences and the corresponding software in the Java programming language were developed. Analysis of the software's performance showed an increase in the effectiveness of providing recommendations for decision-making on the suitability of the choice of gift to the employee's preferences.

In the section of the economic part, an assessment of the commercial potential of the development of information technology for forecasting the volume of flower purchases was carried out, an assessment of the commercial potential of the development was carried out, the costs of carrying out scientific work and the implementation of the results were forecast, and the payback period was calculated..

The graphic part consists of 8 posters with modelling results.

Keywords: information technology, decision-making, recommendations, gifts.

ЗМІСТ

ВСТУП	4
1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ З ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДПОВІДНОСТІ ВИБОРУ ПОДАРУНКА ДО ВПОДОБАНЬ ПРАЦІВНИКА..	8
1.1 Дослідження особливостей використання програмних засобів для вибору подарунку.....	8
1.2 Обґрунтування доцільності технології надання рекомендацій з прийняття рішень вирішення проблеми вибору подарунка.....	12
1.3 Існуючі моделі та методи, що застосовуються для надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника	14
1.4 Аналіз сучасних продуктів для вибору подарунка відповідно до вподобань отримувача.....	22
1.5 Постановка задачі дослідження.....	30
1.6 Висновок до розділу 1	31
2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ З ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДПОВІДНОСТІ ВИБОРУ ПОДАРУНКА ДО ВПОДОБАНЬ ПРАЦІВНИКА	32
2.1 Обґрунтування вибору методу надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника	32
2.2 Розробка математичної моделі надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника	34
2.3 Проектування структури інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника	38
2.4 Розробка алгоритму функціонування інформаційної технології.....	43
2.5 Обґрунтування вибору бази даних для зберігання даних для інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.....	48

2.6 Висновок до розділу 2	51
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ З ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДПОВІДНОСТІ ВИБОРУ ПОДАРУНКА ДО ВПОДОБАНЬ ПРАЦІВНИКА	53
3.1 Обґрунтування вибору мови та середовища програмування.....	53
3.2 Обґрунтування вибору бібліотек та фреймворків для інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.....	56
3.3 Опис процесу програмної розробки.....	60
3.4 Опис класів і функцій інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника	64
3.5 Тестування та дослідження результатів роботи інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.....	66
3.6 Висновок до розділу 3	74
4 ЕКОНОМІЧНА ЧАСТИНА	75
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки	75
4.2 Розрахунок витрат на здійснення науково-дослідної роботи	76
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	83
4.4 Висновок до розділу 4	87
ВИСНОВКИ	88
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	90
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	94
Додаток Б (обов'язковий) Лістинг програми.....	95
Додаток В (обов'язковий) Ілюстративна частина	111
Додаток Г (довідниковий) Інструкція користувача.....	119

ВСТУП

Актуальність теми. Прийняття рішень щодо відповідності вибору подарунку до вподобань працівника має високу актуальність, оскільки правильно обраний подарунок може виявитися не тільки приємним сюрпризом, але й вказівкою на те, що колеги звертають увагу на інтереси та особистість працівника. Подарунок, який враховує вподобання отримувача, може створити позитивну атмосферу на роботі, підвищити мотивацію та підтримати гармонійні взаємини в колективі.

Наведемо деякі фактори, які підкреслюють актуальність правильного вибору подарунку для працівників. Мотивація та визнання: правильно обраний подарунок може бути проявом визнання індивідуальних досягнень та внеску працівника. Побудова позитивних взаємин: вибір подарунка, що відповідає вподобанням працівника, може зміцнити зв'язок між вами як роботодавцем або керівником і співробітником. Збереження морального духу: відповідний подарунок може підняти настрій та покращити настрої працівника. Підвищення ретенції персоналу: це може позитивно вплинути на задоволеність працівника своєю роботою та зменшити ймовірність його переходу до іншої компанії. Демонстрація відповідального роботодавця: враховуючи вподобання працівників при виборі подарунків, ви показуєте свою відповідальність та професіоналізм як роботодавець. Отже, актуальність правильного вибору подарунку до вподобань працівника важлива для створення сприятливої робочої атмосфери, мотивації працівників та підтримки продуктивних взаємин.

Розробка програмного рішення дозволяє автоматизувати процес вибору подарунків, що займає значну кількість часу і зусиль, особливо якщо є велика команда або багато працівників. Програма може здійснювати аналіз та рекомендації швидше і ефективніше, звільняючи людину від рутинних завдань. Розробка програмного рішення також допоможе забезпечити більш точні рекомендації, що підвищує ймовірність вибору подарунку, який сподобається та зробить працівника щасливим. Так як кожен працівник унікальний, розробка

програмного рішення дозволить створити персоналізовані рекомендації на основі його вподобань. Це створить враження індивідуального підходу та підсилює зв'язок між працівником і організацією. Впровадження інформаційної технології для прийняття рішень щодо відповідності вибору подарунку може стати конкурентною перевагою компанії. Це допоможе покращити репутацію як роботодавця, збільшити лояльність працівників та залучити нових талановитих фахівців. Зважаючи на ці переваги, розробка інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунку до вподобань працівника є актуальною та цікавою ідеєю для впровадження в організації.

Висновок із висловленого раніше полягає в тому, що тема досліджень у сфері прийняття рішень щодо відповідності вибору подарунку індивідуальним вподобанням працівника є надзвичайно актуальною. Системи підтримки прийняття рішень знаходяться в стадії інтенсивного розвитку, що свідчить про постійне підвищення якості їхньої роботи. Таким чином, для проведення дослідження слід аналізувати найкращі практики, моделі та методи з метою підвищення ефективності систем, що рекомендують варіанти подарунків, відповідно до власних вподобань працівників..

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження є підвищення ефективності надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника шляхом поєднання методів аналізу ієрархій та нечіткого дерева рішень.

Для досягнення поставленої мети необхідно розв'язати наступні **задачі:**

- провести аналіз програм-аналогів та обґрунтувати доцільності розробки інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника;
- розробити структуру інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника;
- розробити математичну модель надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника;
- здійснити програмну реалізацію інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника;
- провести тестування програми та проаналізувати отримані результати;
- економічно обґрунтувати доцільність розробки інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.

Об’єкт дослідження – це процес надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.

Предмет дослідження – програмні засоби надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.

Методи дослідження. У роботі використано такі методи наукових досліджень: метод системного аналізу для аналізу структури інформаційної системи; метод нечіткого дерева рішень; методи класифікації; методи експертних оцінок; методи математичної статистики; методи об’єктно-орієнтованого програмування для автоматизації розрахунків.

Наукова новизна одержаних результатів полягає в наступному: удосконалено інформаційну технологію для прийняття рішень щодо відповідності вибору подарунка до вподобань користувача, яка відрізняється від існуючих удосконаленим методом, який поєднує в собі алгоритми нечіткого дерева рішень та алгоритм на основі методу аналізу ієрархій для надання

рекомендацій, що дозволяє максимально задовольнити потреба користувача та забезпечує високу ефективність прийняття рішень щодо відповідності вибору подарунку до вподобань працівника.

Практичне значення одержаних результатів полягає у такому:

1. Удосконалено алгоритм прийняття рішень щодо відповідності вибору подарунка до вподобань користувача.
2. Здійснено програмну реалізацію інформаційної технології для прийняття рішень щодо відповідності вибору подарунка до вподобань користувача.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок здобувача. Результати даної магістерської кваліфікаційної роботи отримані самостійно. В публікації у співавторстві здобувачу належить дослідження перспектив інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань користувача [1, 2].

Апробація результатів магістерської кваліфікаційної роботи. Результати досліджень було апробовано на LI Науково-технічній конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2023) м. Вінниці у 2023 р. [2].

Публікації магістерської кваліфікаційної роботи. За основними результатами досліджень опубліковано тези доповіді на конференції [1, 2].

1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ З ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДПОВІДНОСТІ ВИБОРУ ПОДАРУНКА ДО ВПОДОБАНЬ ПРАЦІВНИКА

1.1 Дослідження особливостей використання програмних засобів для вибору подарунку

Дарування подарунків завжди було надзвичайно особистим заняттям. Хоча ця діяльність є ознакою вдячності серед людей, це також надзвичайно виснажлива та прискіплива робота.

Зіткнувшись з величезним вибором, споживачі часто стикаються з проблемою вибору правильного подарунку. В результаті з'явилася персоналізація як спосіб допомогти користувачам знайти унікальні та продумані подарунки, які ідеально підходять для будь-якої події.

Програмне рішення для прийняття рішень щодо відповідності вибору подарунку до вподобань працівника може бути цікавим інструментом, який допоможе автоматизувати і спростити цей процес. При розробці такого рішення можна виділити наступні кроки [3]:

Збір інформації про працівника. Розробка програми може починатися зі збору інформації про вподобання та інтереси працівника. Це може включати заповнення анкети або опитувальника, де працівник може вказати свої хобі, захоплення, уподобання щодо подарунків.

Створення бази даних. Наступним кроком буде створення бази даних, де будуть зберігатися інформація про працівників, їх вподобання і можливі подарунки. База даних може містити інформацію про категорії подарунків, попередні вибори працівників та їх реакцію на отримані подарунки.

Аналіз та зважування вподобань. Програмне рішення може використовувати алгоритми аналізу та зважування, щоб визначити, які вподобання працівника є найбільш вагомими та відповідними. Наприклад, можна

присвоїти вагу різним категоріям подарунків, щоб врахувати їх важливість для конкретного працівника.

Рекомендації щодо подарунків. На основі зібраної інформації і вагових коефіцієнтів програма може надати рекомендації щодо подарунків, які найкраще відповідають вподобанням конкретного працівника. Рекомендації можуть бути зроблені на основі попередніх виборів і реакцій інших працівників зі схожими інтересами [4].

Персоналізація та адаптація. Кожен працівник унікальний, тому програма може включати можливості персоналізації рекомендацій. Наприклад, ви можете дати можливість працівнику налаштувати вагові коефіцієнти різних категорій або вказати конкретні предмети, які він бажає отримати.

Відстеження результатів. Важливо відстежувати результати рекомендацій і реакції працівників на отримані подарунки. Це дозволить вдосконалити систему рекомендацій і забезпечити більш точні результати у майбутньому.

Забезпечення конфіденційності. При розробці програмного рішення для прийняття рішень щодо відповідності вибору подарунку, важливо врахувати питання конфіденційності даних працівників. Забезпечення захисту персональної інформації та дотримання відповідних норм та політик є обов'язковим.







Це лише загальні кроки при розробці програмного рішення для прийняття рішень щодо відповідності вибору подарунку до вподобань працівника. Конкретні деталі і функції програми будуть залежати від специфічної ситуації та потреб конкретної організації.

Організації активно застосовують методи штучного інтелекту для надання персоналізованих подарунків своїм працівникам, зорієнтованих на їх індивідуальні вподобання та потреби. Це означає, що системи рекомендацій використовують різні техніки, такі як аналіз соціальних даних та доступ до Інтернету, для надання спеціальних пропозицій [5].

Автоматичні рекомендації в інтернет-магазинах допомагають знаходити ідеальний продукт, спираючись на аналіз попередніх виборів користувача. Наприклад, розділ «рекомендації на основі ваших переглядів» в інтернет-магазині

Rozetka може містити товари, що відповідають потребам та вподобанням користувача (рис. 1.1).

Реклама ⓘ

 <p>Набір Убтан Hillary ASAI UBTAN 100 г + 739 ₴ 369 ₴</p>	 <p>Набір шовкового догляду за тілом з дерев'яним 2-278 ₴ 1 595 ₴</p>	 <p>Комплексний догляд: Очищуюча пінка. 2-733 ₴ 1 366 ₴</p>	 <p>Набір для літнього догляду Hillary Perfect 2-979 ₴ 1 192 ₴</p>	 <p>Набір для літнього догляду Hillary Perfect 3-341 ₴ 1 671 ₴</p>	 <p>Набір для літнього догляду Hilary Perfect 2-147 ₴ 1 074 ₴</p>
--	---	---	---	--	---

Показати ще ↗

Рекомендації на основі ваших переглядів




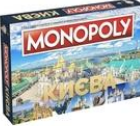


 <p>Настільна гра Strateg Монополія світу 479 ₴ 309 ₴</p>	 <p>Настільна гра Strateg Прапори світу 319 ₴ 189 ₴</p>	 <p>Настільна гра Strateg Монополія (велика) 249 ₴ 149 ₴</p>	 <p>Настільна гра Rozum Монополія Знамениті 1 599 ₴</p>	 <p>Настільна гра Strateg Монополія по-Одеськи 479 ₴ 279 ₴</p>	 <p>Настільна гра Strateg Мафія українською 153 ₴ 99 ₴</p>
---	--	--	--	---	---

Рисунок 1.1 – Рекомендації інтернет-магазину Rozetka

Інтернет-магазин працює подібно до підходу Netflix або Spotify, що дозволяє кожному користувачеві отримати індивідуальні рекомендації [6].

Технологія спільної фільтрації на основі штучного інтелекту представляє собою один із ефективних підходів у системах рекомендацій. Цей метод дозволяє автоматично передбачати та фільтрувати інтереси користувача, використовуючи інформацію про їхні вподобання та смак, яку система отримує від співпраці з багатьма іншими користувачами. Заснована на взаємодії та обміні даними, спільна фільтрація надає рекомендації, зокрема, на основі попередніх виборів користувачів, таких як натискання, перегляди, покупки, вподобання чи оцінки.

Основна ідея за цією технологією полягає в гіпотезі, що якщо один користувач, наприклад, Джейн, має схожі думки з іншим користувачем, скажімо,

Джо, щодо конкретного фільму чи продукту (означеного як А), то існує велика ймовірність, що їхні вподобання схожі і щодо іншого фільму чи продукту (позначеного як Б). Це вірогідності обчислюються на підставі подібності вподобань та думок між користувачами.

Наприклад, якщо Джейн і Джо обидва високо оцінили фільм А, то на основі спільної фільтрації система рекомендацій може передбачити, що їхні вподобання узгоджені і вірогідно Джейн також сподобається фільм Б, який отримав високу оцінку від Джо. Це підтверджує ідею, що схожість у вподобаннях між користувачами може служити основою для точних та персоналізованих рекомендацій у системах штучного інтелекту.

Найефективнішим способом збільшення продажів та покращення утримання клієнтів є поєднання кількох методів фільтрації. Окрім спільної фільтрації, використовуються фільтрація на основі вмісту та гібридні рекомендаційні системи. Фільтрація на основі вмісту рекомендує продукти на основі їхніх характеристик та ключових слів, тоді як гібридні системи поєднують обидві підходи.

Прогнози, що надходять від технології фільтрації на основі вмісту, є унікальними для кожної людини та продукту, засновані на характеристиках продукту, даних користувача та інформації, зібраної від багатьох інших користувачів.

Вражаючим прикладом гібридної системи рекомендацій слугує вже згаданий сервіс Netflix. Компанія не лише стежить та агрегує дані від користувачів щодо фільмів, але і враховує характеристики фільмів, жанри, визначальних акторів та інші релевантні аспекти. Здійснюючи аналіз споживчих звичок, Netflix пропонує індивідуальні рекомендації фільмів, об'єднуючи інформацію про схожість смаків інших користувачів з даними про спільні характеристики фільмів [7].

Для прикладу, якщо користувачі з подібними вподобаннями сприймають певний фільм позитивно і високо оцінюють його, Netflix може рекомендувати інші фільми, що мають спільні компоненти, такі як той самий жанр, визначальний

актор чи сюжетні елементи. Цей підхід дозволяє платформі пропонувати персоналізовані рекомендації, що ґрунтуються на поєднанні індивідуальних виборів користувача та колективної інформації, зібраної від інших учасників платформи. Такий синтез даних дозволяє Netflix забезпечувати точні та цікаві рекомендації для свого різноманітного аудиторійного спектру.

В реальному сценарії електронної комерції, користувач, наприклад, шукаючи шейкер для коктейлів, може отримати індивідуальні рекомендації про інші інструменти для приготування коктейлів, які відповідають його вибору. Така гібридна система об'єднує фільтрацію на основі вмісту та спільну фільтрацію, забезпечуючи ефективні рекомендації на великих масштабах та запобігаючи перевантаженню інформації для користувача. Використання штучного інтелекту в цьому контексті вирішує завдання підвищення ефективності та забезпечує унікальний та персоналізований досвід для кожного клієнта.

1.2 Обґрунтування доцільності технології надання рекомендацій з прийняття рішень вирішення проблеми вибору подарунка

Використання технології прийняття рішень допомагає зробити процес вибору подарунка більш систематичним, ефективним та особистим. Вона дозволяє забезпечити відповідність між подарунком і отримувачем, зробити приємний враження і підвищити якість взаємин.

Технологія прийняття рішень дозволяє обробити велику кількість даних. При виборі подарунка для великої кількості отримувачів може бути складно розглянути всі варіанти і зробити оптимальний вибір. Використання технології дозволяє ефективно обробляти великі обсяги даних, аналізувати їх і надавати рекомендації на основі цих даних. Технологія прийняття рішень може використовувати алгоритми та штучний інтелект для врахування індивідуальних вподобань отримувачів. Вона аналізує інформацію про їхній смак, інтереси, хобі та інші фактори, що допомагають зробити більш точний і персоналізований вибір подарунка [8].

Використання технології прийняття рішень дозволяє підвищити задоволення отримувача від отриманого подарунка. Вона допомагає забезпечити відповідність між вибраним подарунком і вподобаннями отримувача, забезпечуючи більш особистий і цінний досвід. Технологія прийняття рішень допомагає ефективно використовувати ресурси, такі як час і кошти. Вона дозволяє швидко знайти найкращий варіант подарунка, уникнути невдалих виборів і забезпечити оптимальний результат. Використання технології дозволяє покращити користувацький досвід при виборі подарунка [9]. Вона може пропонувати інтерактивні інтерфейси, персоналізовані рекомендації та зручні інструменти для пошуку і вибору подарунків.

Інформаційна технологія надання рекомендацій з прийняття рішень для вибору подарунка відображає собою систему, яка має на меті допомогти користувачеві у прийнятті оптимального рішення щодо вибору подарунка, який відповідає його вподобанням.

Нехай існує множина вхідних даних $A = \{(a_i, v_i, p_i)\}$ де a_i представляє собою конкретний атрибут, v_i – відповідне значення цього атрибуту, p_i – фактор впевненості в тому, що наскільки відповідне значення v_i відповідає атрибуту a_i . охоплюють всі різноманітні характеристики, які визначають аспекти вибору, такі як вік, стать, ціна, хобі та інші. Значення фактору впевненості p_i лежить в межах від 0 до 1 і вказує на ступінь впевненості відповідності між конкретним атрибутом та його значенням.

Множина результатів $B = (m_i, q_i)$, де m_i визначає рекомендований подарунок, q_i вказує на ступінь впевненості у відповідності цього рекомендованого подарунку. Кожне таке поєднання подарунка та ступеня впевненості також знаходиться в межах від 0 до 1.

Завданням є розробка методу, який створить відображення $f: A \rightarrow B$, що найкраще враховує вподобання працівника. Це означає, що для кожного набору вхідних атрибутів, представленого множиною A , метод має надавати найоптимальніший рекомендований подарунок разом із відповідним ступенем впевненості, враховуючи індивідуальні вподобання та особливості працівника.

Внаслідок проведених досліджень у визначеній сфері виникла потреба в створенні інформаційної технології, яка дозволить користувачам більш ефективно приймати рішення щодо вибору подарунка, враховуючи вподобання конкретного працівника.

Основні завдання такої системи включають обробку вхідних даних, встановлення взаємозв'язків між різними атрибутами та надання індивідуалізованих рекомендацій. Під час обробки вхідних даних система аналізує різні параметри, такі як вік, стать, цінові уподобання та інші, для створення повноцінного профілю користувача. Визначення залежностей передбачає аналіз взаємодії між різними факторами, щоб точніше врахувати вподобання та особливості кожного працівника. Надання рекомендацій полягає у висуненні індивідуально адаптованих порад щодо вибору подарунка, що відповідає унікальним уподобанням кожного користувача.

1.3 Існуючі моделі та методи, що застосовуються для надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника

У сучасному інформаційному середовищі виникла необхідність дослідження та вдосконалення моделей та методів, спрямованих на надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка індивідуальним вподобанням працівника. Тому необхідно провести огляд існуючих підходів та методологій у сфері систем підтримки прийняття рішень, які стосуються вибору подарунків.

В галузі рекомендаційних систем існують різноманітні моделі та методи, які успішно застосовуються для аналізу та рекомендаційної пропозиції товарів. Однак, враховуючи особливості задачі вибору подарунків, важливо вивчити та адаптувати ці підходи для врахування особистих вподобань працівника.

Доцільно розглянути наступні методи, що можуть ефективно виконувати завдання прийняття рішень щодо відповідності вибору подарунка до вподобань працівника: метод аналізу ієрархій, метод ELECTRE, FuzzyID3.

Метод аналізу ієрархій є популярним інструментом для структурування та аналізу складних рішень. В контексті вибору подарунка може бути використаний для ранжування різних альтернатив відповідно до важливості різних критеріїв. Підходить для врахування суб'єктивних вподобань працівника та призначення їм ваг у процесі прийняття рішення.

Метод аналізу ієрархії (MAI) [10] є багатокритеріальним методом прийняття рішень, розробленим Томасом Сааті. В основі цього методу лежить ідея розкладання складного рішення на ієрархічні рівні, що спрощує процес прийняття рішень. MAI використовує парний порівняльний аналіз для визначення важливості та взаємозв'язку критеріїв, а також альтернатив.

Процес включає в себе створення матриці порівнянь, де для кожної пари критеріїв або альтернатив визначається їхнє відносне значення. На основі цієї матриці розраховуються ваги для кожного елементу ієрархії. Після цього проводиться синтез ваг та ранжування альтернатив.

MAI дозволяє враховувати як об'єктивні, так і суб'єктивні критерії, а також уникає надмірної складності завдань прийняття рішень за рахунок їх розбиття на менші, легше управляючі частини. Цей метод застосовується в різних галузях, включаючи управління, економіку, технічні науки та інші. Цей алгоритм дозволяє систематизувати та узгодити різні критерії та альтернативи, роблячи процес прийняття рішень більш об'єктивним та зрозумілим. Структура методу показана на рисунку 1.2.

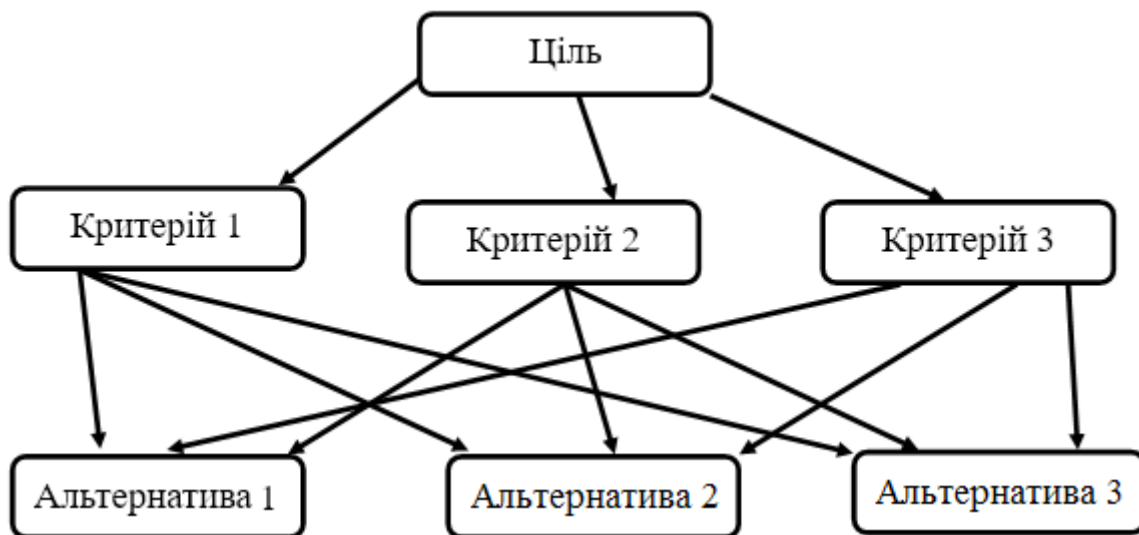


Рисунок 1.2 – Структурна схема методу аналізу ієрархій

Алгоритм складається з наступних кроків:

1. Формулювання проблеми: визначення мети прийняття рішення та ідентифікація альтернатив;
2. Побудова ієрархії: розбиття проблеми на ієрархічні рівні, включаючи критерії та альтернативи;
3. Парний порівняльний аналіз: створення матриці порівнянь для кожного рівня ієрархії. Для кожної пари елементів визначається їхнє відносне значення за допомогою шкали порівнянь (наприклад, від «рівнозначні» до «дуже суттєво кращий»);
4. Обчислення ваг: на основі отриманих матриць визначаються ваги для кожного елементу на кожному рівні ієрархії. Обчислення ваг використовується для побудови синтетичної матриці ваг;
5. Синтез ваг: обчислення загальної ваги для кожної альтернативи на верхньому рівні ієрархії;
6. Ранжування альтернатив: ранжування в порядку спадання їхніх загальних ваг;
7. Аналіз стійкості: перевірка стійкості отриманих результатів шляхом аналізу чутливості до змін вхідних даних.

Кожну матрицю необхідно перевірити на узгодженість за формулою 1.1:

$$CI = \frac{\lambda_{max} - m}{m - 1}, \quad (1.1)$$

де m – порядок матриці попарних порівнянь; λ_{max} – найбільше власне число матриці попарних порівнянь.

Множину альтернатив визначають одним або кількома з наступних алгоритмів:

- метод найближчого сусіда;
- K-means метод;
- метод SVD;
- метод гібридної фільтрації;
- коефіцієнт кореляції Пірсона;
- метод SVD;
- TF-IDF;
- Apriori.

Метод ELECTRE (ELimination Et Choix Traduisant la REalite) є методом, спрямованим на вирішення багатокритеріальних завдань прийняття рішень. Використовується для порівняння альтернатив та визначення їхнього рівня відповідності критеріям. Може бути застосований для врахування різноманітних критеріїв, таких як ціна, стиль, вік та інші, у виборі ідеального подарунка.

Метод ELECTRE складається з наступних етапів [11]:

1. Альтернатив за критеріями, для кожного критерію визначається ваговий коефіцієнт w для кожного критерію і встановлюються пороги індексів узгодження.

2. Розбиття множини критеріїв. Множина K розбивається на три підмножини: перевага першої альтернативи, рівні оцінки та перевага другої альтернативи.

3. Визначення відносної важливості кожної множини. Розрахунок відносної важливості кожної з цих множин за допомогою формули 1.2.

$$P_{x_1x_2}^* = \sum_{i|K_i \in K^*(x_1x_2)} w_i, \text{ де } * \in \{+, =, -\}; \quad (1.2)$$

4. Обчислення індексу узгодження $f(P_{x_1x_2}^+, P_{x_1x_2}^-, P_{x_1x_2}^-)$ для порівняння з пороговим значенням C :

$$f(P_{x_1x_2}^+, P_{x_1x_2}^-, P_{x_1x_2}^-) = \begin{cases} \frac{P_{x_1x_2}^+ + P_{x_1x_2}^-}{\sum_{i=1}^n w_i}, & \text{якщо } 0,5 \leq C \leq 1 \\ \frac{P_{x_1x_2}^+}{P_{x_1x_2}^-}, & \text{якщо } C \geq 1 \end{cases} \quad (1.3)$$

Якщо виконується умова (потрібна) $f(P_{x_1x_2}^+, P_{x_1x_2}^-, P_{x_1x_2}^-) > C$, то перша перевага має переваги порівняно з другою.

5. Обчислення індексу неузгодження $d_{x_1x_2}$:

$$d_{x_1x_2} = \begin{cases} 0, & \text{якщо } K^-(x_1x_2) = \emptyset \\ \frac{\max_{i|K_i \in K^-(x_1x_2)} \{w_i * |x_1^i - x_2^i|\}}{\max_{i=1, \bar{n}} \{w_i * |x_1^i - x_2^i|\}}, & \text{якщо } K^-(x_1x_2) \neq \emptyset \end{cases} \quad (1.4)$$

Відбувається визначення індексу неузгодження для визначення ступеня розбіжності між альтернативами. Якщо виконується умова (достатня): $d_{x_1x_2} < d$, то перша альтернатива має переваги порівняно з другою.

6. Порівняння індексів та прийняття рішення. Обираються пари альтернатив, для яких виконуються відповідні умови, і порівнюються значення їх індексів.

7. Визначення ядра відношення: виділення альтернатив, для яких проводяться порівняння, і обрані альтернативи між собою не порівнюються.

Цей метод дозволяє систематизувати та оцінити альтернативи, враховуючи важливість кожного критерію та ступінь узгодження/неузгодження між альтернативами.

Зображення етапів методу представлено на рисунку 1.3.

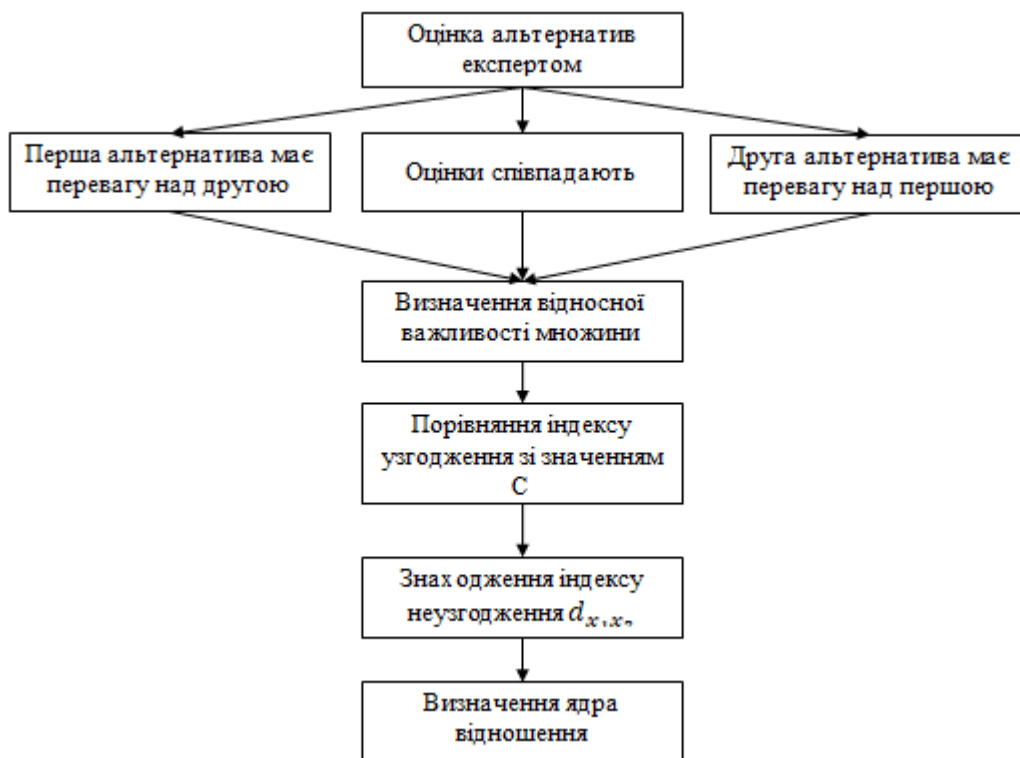


Рисунок 1.3 – Етапи методу ELECTRE

Fuzzy ID3 – це розширення класичного алгоритму ID3 (Iterative Dichotomiser 3), який використовується для побудови дерев рішень в галузі машинного навчання. Алгоритм ID3 використовується для навчання моделей класифікації на основі вхідних даних та вибору оптимальних рішень на кожному кроці [12]. FuzzyID3 використовує нечітку логіку для класифікації об'єктів. Застосування нечітких множин дозволяє враховувати неоднозначність та вагомість критеріїв в процесі вибору подарунка. Метод може допомогти врахувати нечіткі або нетривіальні вподобання працівника.

Основні етапи Fuzzy ID3 можуть включати:

1. Визначення нечітких правил. Врахування ступенів належності для визначення того, наскільки елементи даних відповідають різним класам.
2. Розгалуження дерева рішень. Побудова дерева рішень, де кожен вузол відображає правило прийняття рішення, а гілки представляють різні можливості.
3. Підтримка нечіткості в умовах. Врахування нечіткості при визначенні умов для рішень на кожному рівні дерева.

4. Критерії розділення. Використання нечітких критеріїв для визначення, які атрибути краще розділяють дані на класи.

5. Оптимізація для роботи з нечіткими даними. Адаптація оптимізацій та методів для роботи з нечіткими масивами даних.

Застосування нечіткої логіки дозволяє Fuzzy ID3 більш гнучко враховувати невизначеність та неоднозначність у вхідних даних, що може бути корисним у ситуаціях, де точні значення важко визначити [13].

Побудова нечіткого дерева рішень для прийняття рішень щодо відповідності вибору подарунку до вподобань працівника може бути виконана за допомогою алгоритму, який враховує нечіткість, або невизначеність, у вхідних даних та правилах прийняття рішень. Наведемо загальний опис алгоритму:

Крок 1: Збір даних. Необхідно зібрати інформацію про вподобання працівника та характеристики подарунків (атрибути). Визначити ступені належності працівника до різних варіантів подарунків на основі його вподобань.

Крок 2: Визначення атрибутів. Визначити атрибути, які впливають на відповідність вибору подарунку до вподобань працівника (наприклад, ціна, тип, колір, функціональність).

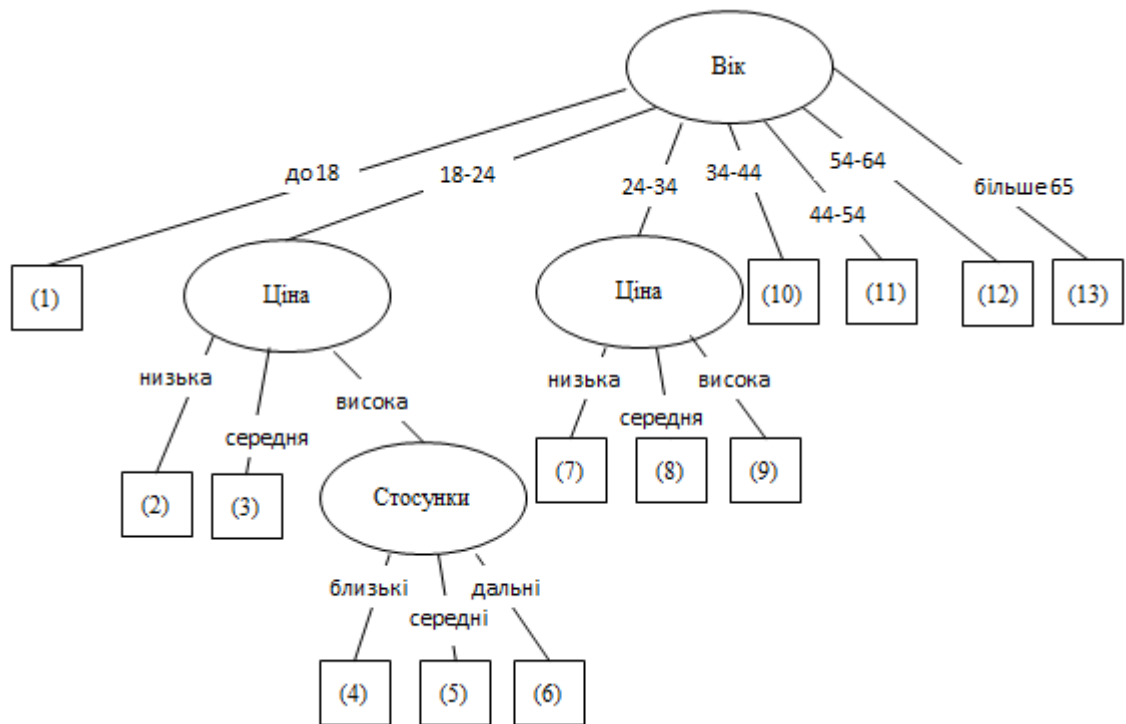
Крок 3: Розгалуження дерева. Визначити критерії розділення для кожного вузла дерева на основі атрибутів. Враховувати нечіткість у визначенні умов розділення, де можуть бути нечіткі категорії або ступені належності.

Крок 4: Побудова правил прийняття рішень. Для кожного листка дерева формувати правило прийняття рішення, враховуючи вподобання працівника та ступень належності до конкретних категорій подарунків.

Крок 5: Визначення ступенів впевненості. Присвоїти ступені впевненості для кожного правила прийняття рішення, враховуючи ступені належності та інші фактори.

Крок 6: Використання дерева для рекомендацій. Використовувати побудоване дерево для прийняття рекомендацій щодо подарунків на основі вподобань працівника.

Приклад нечіткого дерева прийняття рішень для вибору подарунку наведено на рисунку 1.4.



*

Рисунок 1.4 – Приклад нечіткого дерева прийняття рішень

Однією з переваг нечіткого дерева рішень є його здатність ефективно моделювати та обробляти нечіткі або невизначені дані. Така система може працювати з концепціями, які не мають чітко визначених меж або категорій, що є важливим у випадках, коли вхідні дані можуть бути суб'єктивними або неоднозначними.

Іншою перевагою є можливість врахування ступенів належності та невизначеності у вираженні відносності атрибутів. Наприклад, якщо вподобання працівника до певної категорії подарунків не є абсолютно чітким, нечітке дерево рішень може врахувати цей фактор та надати більш гнучкі рекомендації.

Крім того, нечіткі дерева рішень можуть автоматично враховувати важливість кожного атрибуту та його вплив на кінцеве рішення, що сприяє покращенню точності та адаптивності системи.

Розглянувши наведені методи, можна прийти до висновку, що найбільш оптимальним підходом для прийняття рішень щодо відповідності вибору подарунку до вподобань працівника є комбінація методів аналізу ієрархій та Fuzzy ID3. Комбінація цих методів може створити комплексну систему, яка дозволить більш точно та гнучко враховувати усі аспекти при виборі подарунку, забезпечуючи при цьому ефективні та збалансовані рекомендації.

1.4 Аналіз сучасних продуктів для вибору подарунка відповідно до вподобань отримувача

Існує кілька програм та онлайн-інструментів, які можуть допомогти в підборі подарунку відповідно до вподобань людини. Наприклад:

GiftAdvisor [14]: Ця програма використовує штучний інтелект для рекомендацій подарунків. Вона аналізує вподобання, інтереси та характеристики отримувача, такі як вік, стать і т. д., а потім пропонує варіанти подарунків, які йому можуть сподобатися. Можна вибирати за категоріями або за конкретними характеристиками.

Gift Wizard [15]: Ця програма пропонує рекомендації подарунків на основі даних, які ви надаєте про отримувача. Ви можете вказати його вік, стать, інтереси, бюджет і багато іншого, і програма запропонує вам варіанти подарунків, які підходять до цих критеріїв.

Amazon Gift Finder [16]: Це інструмент, який можна знайти на веб-сайті Amazon. Він дозволяє вам ввести характеристики отримувача, такі як вік, стать, інтереси, і програма пропонує варіанти подарунків, доступних на Amazon, які можуть бути йому цікаві.

Etsy Gift Finder [17]: Цей інструмент можна знайти на веб-сайті Etsy. Він дозволяє вам ввести вподобання, стиль або інтереси отримувача, а програма пропонує рекомендації подарунків, які відповідають цим критеріям. Etsy спеціалізується на ручній роботі та унікальних товарах, що робить його особливо вигідним для пошуку оригінальних подарунків.

Однак наведені інструменти не адаптовані до користувачів з України, мають платні версії, а також підбір відбувається на основі вибору користувача, що в контексті вибору подарунків для працівників не є надійним джерелом.

Застосунок під назвою «Знайти подарунок» може стати важливим інструментом у виборі подарунків, якщо особа, яка робить вибір, добре знайома з інтересами конкретного отримувача [18]. Це додаток, який допомагає вирішити питання пошуку подарунку, надаючи оптимальні варіанти, які враховують вартість і інтереси отримувача.

Користувачеві пропонується вибрати стать особи, для якої призначений подарунок, а також вибрати категорію, таку як техніка та електроніка, товари для дому, засоби для краси та спорту і т. д., і визначити ціновий діапазон (рисунок 1.5). Потім додаток генерує відповідні варіанти. У випадку, якщо варіант не влаштовує, можна скористатися кнопкою «Повторити пошук», і додаток запропонує нові ідеї. Якщо ж користувач задоволений обраним варіантом, він може натискати «Шукати в інтернеті». Додаток надасть перелік магазинів і веб-сайтів, де можна придбати обраний товар.

Наведений застосунок «Знайти подарунок» виявляється корисним інструментом для полегшення вибору подарунків, проте варто враховувати кілька важливих аспектів. Ефективність додатку значно залежить від глибини знань користувача про інтереси та уподобання отримувача. Якщо ці знання обмежені, існує ризик неправильного вибору подарунку. Існує ймовірність, що обраний подарунок може вже бути у власності отримувача, що може вплинути на оригінальність та прийняття подарунка. Розгляд можливостей додавання додаткових функцій, таких як опції персоналізації чи деталізації критеріїв вибору, може підвищити точність та задоволення користувачів, а можливість додавання регіональних аспектів, таких як популярні магазини або знижки, може зробити додаток більш адаптованим до конкретних умов користувачів

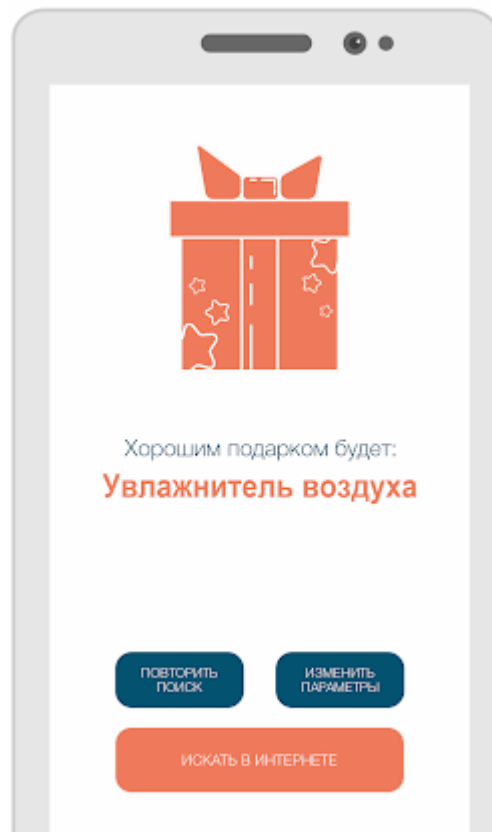


Рисунок 1.5 – Приклад роботи додатку «Знайти подарунок»

Наступним помічником у прийнятті рішень щодо відповідності вибору подарунка до вподобань працівника розглянемо «Giftenizer», який діє на кшталт соціальної мережі (рисунок 1.6), створений для того, щоб спростити та зробити цікавішим процес обміну та вибору подарунків. Він надає можливість ділитися з іншими користувачами тими речами, які вас зацікавили та які ви хочете отримати в подарунок на майбутні свята. Додаток дозволяє користувачам вести окремі списки бажань для різних заходів чи свят, що спрощує організацію та вибір подарунків. Також, додаток дозволяє зручно забронювати та придбати бажаний подарунок в рекомендованих інтернет-магазинах.

Застосунок створює можливість вибору ідеального подарунка відповідно до вподобань та здійснення покупки вчасно. Гарантує сюрприз та захист від спойлерів для того, хто формує свій список бажань.

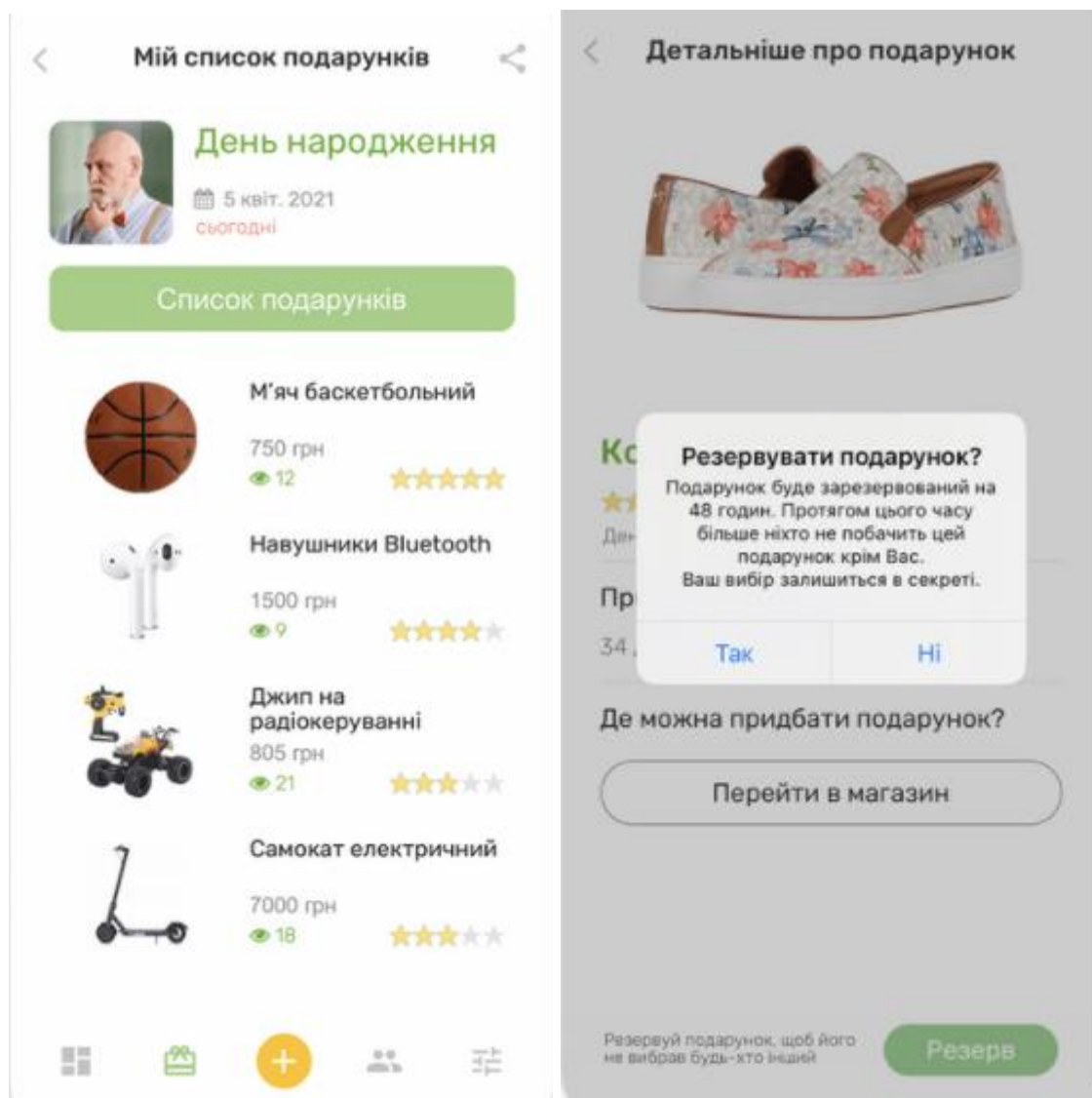


Рисунок 1.6 – Приклад роботи додатку «Giftenizer»

«Giftenizer» є інноваційним рішенням для ефективного управління процесом вибору та обміну подарунками. Завдяки його ключовим функціям, він створює зручний простір для спільного планування та обговорення подарунків між друзями, колегами та родичами. Користувачі можуть легко додавати друзів до свого списку та обмінюватися ідеями для подарунків, отримувати сповіщення про майбутні події та уникати дублювання, позначаючи та резервуючи вибрані подарунки. Забезпечуючи приватність, додаток гарантує, що лише вибрані особи мають доступ до особистих списків бажань користувачів. Зокрема, можливість

створення різних списків бажань для різних подій робить «Giftenizer» універсальним інструментом для керування подарунковим процесом [19].

Додаток пропонує унікальну можливість не лише вибрати бажаний подарунок відповідно до особистих вподобань, але й забронювати та придбати його безпосередньо в рекомендованих інтернет-магазинах. Крім того, «Giftenizer» забезпечує ефект сюрпризу, дозволяючи користувачам, які резервують пункт в списку, зберігати в секреті інформацію про вже зарезервовані подарунки від користувача, що створив список. Усе, що потрібно для використання додатка – це наявність смартфона та доступ до Інтернету.

«NU SHO» – це інтернет-магазин, який пропонує зручний інструмент для підбору ідеального подарунка, враховуючи індивідуальні вподобання отримувача [20]. Сервіс дозволяє прискорити процес вибору за допомогою кількох рівнів фільтрів, а генератор невідповідних товарів надає варіанти, які найкраще підходять для конкретної особи. Підбір подарунка можна здійснити за кілька хвилин, відповідаючи на п'ять простих запитань. Крім того, на сайті існує можливість надихнутися доступними товарами або навіть замовити виготовлення унікальних речей за допомогою конструктора сайту (рис. 1.7).

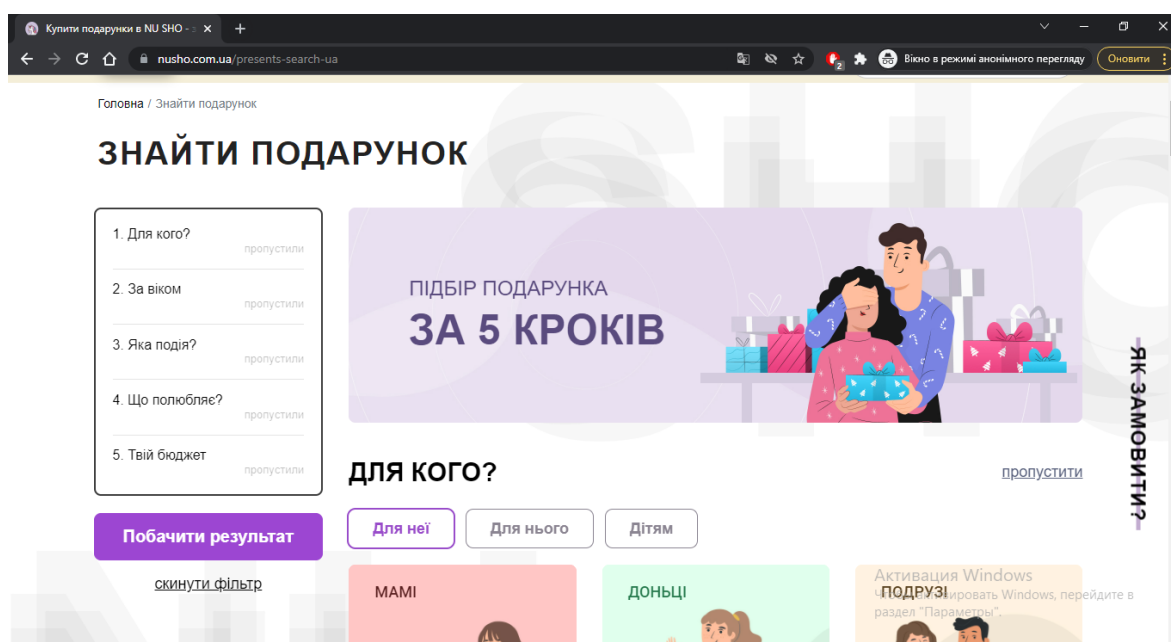


Рисунок 1.7 – Інтерфейс сайту «NU SHO»

Сайт «NU SHO» вигідно виділяється своїм підходом до вибору подарунків, пропонуючи ефективні засоби фільтрації та генератор невідповідних товарів. Відмінною рисою є можливість надихнутися доступними товарами та створити оригінальні речі за допомогою конструктора на сайті. Загалом, сайт «NU SHO» виглядає як практичний та креативний інструмент для вибору подарунків, що враховує індивідуальні смаки та вподобання отримувача.

Зручним рішенням для вибору ідеального подарунка відповідно до індивідуальних уподобань може стати Findodo, ізраїльський стартап, що використовує штучний інтелект для допомоги у святкових покупках [21].

Користувачам пропонується пройти опитування, яке визначає оптимальний список подарунків (рисунок 1.8). Початково програма містила 120 запитань, але з часом вдалося оптимізувати процес, скоротивши їхню кількість до 21, при цьому зберігши точність підбору.



Рисунок 1.8 – Інтерфейс стартапу «Findodo»

Один із недоліків програми полягає в тому, що вона навчалася на американських користувачах, тож точність її роботи для українських користувачів може бути обмеженою. Однак розробники взяли на себе зобов'язання перед виходом на новий ринок здійснити навчання для місцевих користувачів і адаптувати алгоритм анкетування до їхніх потреб. Крім того, вони активно працюють над удосконаленням алгоритмів та розширенням переліку доступних магазинів і подарунків. Компанія виявляє зацікавленість не лише у виборі персоналізованих подарунків, але й у співпраці з брендами, щоб збирати інформацію про їхні продукти та допомагати їм краще розуміти потреби своїх клієнтів.

Ще однією цікавою можливістю для отримання допомоги при виборі подарунків є використання сайту «GiftHub» [22]. Цей інноваційний ресурс використовує автоматизований підхід до вибору подарунків, аналізуючи профілі користувачів у соціальній мережі Instagram (рисунок 1.9).

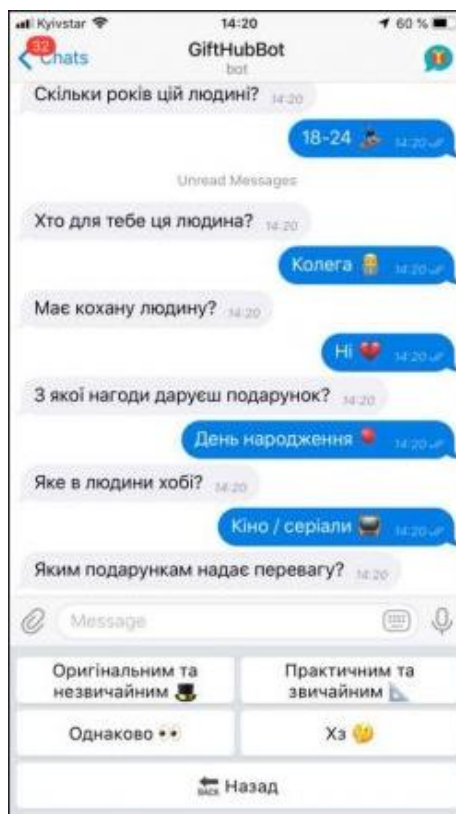


Рисунок 1.9 – Інтерфейс стартапу «GiftHub»

GiftHub враховує активність облікового запису та рекомендує категорії продуктів, такі як телефони та аксесуари, настільні ігри чи прикраси. Отримати обрані подарунки можна на популярних платформах, таких як Amazon або в магазині Sanatty. За кожну успішну покупку користувача GiftHub отримує відсоток від угоди.

Початково GiftHub стартував як локальний проєкт в Україні через Telegram-бот, який аналізував акаунти Instagram. Проте, з метою спрощення процесу, розробники вивели його на більший ринок. Користувачам тепер простіше вибрати ідеальний подарунок, вказавши деякі критерії у діалоговому вікні, такі як особа, якій призначено подарунок, і її хобі. Зараз команда активно розвиває проєкт, зосереджуючись на розширенні бази доступних подарунків та вдосконаленні алгоритму відбору.

Подальші плани команди включають вдосконалення розпізнавання об'єктів на фотографіях, щоб іскусствений інтелект міг надавати кращі рекомендації. Наприклад, якщо алгоритм виявить на фото особу з електрогітарою, він запропонує набір звукознімачів, струн, чохла та інших аксесуарів.

Отже, при аналізі аналогів було виявлено ряд значущих недоліків. Більшість програм характеризуються недостатньою точністю у виборі подарунків та обмеженою кількістю запитань про отримувача. Наприклад, додаток «Знайти подарунок» та сайт «NU SHO» обмежені лише п'ятьма запитаннями, що може вплинути на їхню ефективність. Крім того, важливим недоліком програм «Friendsbox» та «Giftenizer» є необхідність залучення отримувача до процесу вибору подарунка через завантаження додатка. Це, хоча і може покращити точність підбору, але не є вдалим способом прийняття рішень щодо відповідності вибору подарунку до вподобань працівника.

Серед аналізованих програмних рішень «Findodo» та бот «GiftHub» можна визнати найбільш схожими за характеристиками із розроблюваною інформаційною технологією для прийняття рішень щодо відповідності вибору

подарунка до вподобань користувача. Проте, як зазначалося раніше, аналог «Findodo» вивчав користувачів у Сполучених Штатах та взаємодіє з американськими інтернет-магазинами. Ще одним значущим недоліком є його орієнтованість на широкий аудиторію, тоді як розроблюване програмне забезпечення повинно бути націлене на працівників та їхніх колег. Отже, необхідно створити інформаційну технологію, яка враховує всі недоліки аналогів, зокрема розробивши питання анкети, спрямовані на працівників, і використати переваги використання ботів для Telegram.

1.5 Постановка задачі дослідження

Задача цієї роботи полягає у розробці інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунку до вподобань працівника.

Вимоги до працівника:

- варіанти відповідей на питання;
- надання уточнюючих запитань
- завершення сесії (вихід).

Вимоги до адміністратора:

- можливість редагування інформації;
- можливість перегляду рекомендацій;
- можливість додавання нової інформації (питань);
- можливість видалення інформації;

Вимоги до серверу:

- зберігання даних користувача через зв'язок з базою даних;
- визначення типу користувача;
- видача необхідних даних по запиту від адміністратора.

Вхідними даними роботи є наступні питання:

- 1) Вкажіть статтю.
- 2) Вкажіть вік.

- 3) Вкажіть посаду.
- 4) Маєте кохану людину?
- 5) Маєте дітей?
- 6) Яке ваше хобі?
- 7) Який саме вид спорту?
- 8) Що любите читати?
- 9) Подобаються гаджети?
- 10) Подобається активний відпочинок?
- 11) Подобається купувати цікаві речі для дому?
- 12) Подобається все планувати/записувати?
- 13) Маєте домашніх улюбленців?
- 14) Кого саме?

Вихідними даними є рекомендації для користувача щодо вибору подарунку, на основі аналізу його відповідей на поставлені питання.

Отже, потрібно розробити інформаційну технологію, в якій буде реалізовано алгоритм підбору подарунку відповідно до вподобань працівника.

1.6 Висновок до розділу 1

У даному розділі було досліджено особливості використання програмних засобів для вибору подарунку. Також обґрунтовано доцільність технології прийняття рішень для вибору подарунку. Було проведено аналіз сучасних програм-аналогів, які використовуються для підбору подарунків. Було наведено короткий опис основних функцій, які виконують дані програми. Також було досліджено методи, що можуть бути використані для надання рекомендацій щодо вибору подарунку відповідно до вподобань користувача, для чого запропоновано використати поєднання методів аналізу ієрархій та Fuzzy ID3. Здійснено постановку задачі.

2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ З ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДПОВІДНОСТІ ВИБОРУ ПОДАРУНКА ДО ВПОДОБАНЬ ПРАЦІВНИКА

2.1 Обґрунтування вибору методу надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника

Метод нечіткого дерева рішень (Fuzzy Decision Tree) – це аналітичний метод, який поєднує в собі два основних підходи до прийняття рішень: дерево рішень і нечітку логіку. Він використовується для моделювання та прийняття рішень в умовах, коли дані або критерії не завжди чітко визначені і мають ступінь нечіткості або неоднозначності.

Наведемо основні характеристики методу нечіткого дерева рішень. Схоже на традиційні дерева рішень, метод нечіткого дерева рішень розглядає різні альтернативи та критерії для прийняття рішень. Вирішальні правила представлені у вигляді дерева з гілками і вузлами, де кожен вузол представляє питання або критерій, а гілки вказують на можливі варіанти рішень. Нечітка логіка використовується для обробки нечітких або неоднозначних даних. Кожен критерій або альтернатива може мати ступінь приналежності до різних класів чи категорій, що дозволяє враховувати нечіткість в оцінках [23].

У методі нечіткого дерева рішень рішення приймаються на основі набору правил, які враховують усі різні критерії та ступені приналежності. Ці правила формалізують логічні відношення між критеріями та варіантами рішень. Однією з основних переваг методу є здатність моделювати нечіткість і неоднозначність в прийнятті рішень. Він дозволяє враховувати реальні умови, де чітко визначені значення не завжди існують.

Метод нечіткого дерева рішень може бути застосований у різних галузях, включаючи фінанси, логістику, медицину, управління ризиками, прийняття рішень в умовах невизначеності, а також в інших сферах, де важливо враховувати нечіткі критерії та нечіткі дані.

Метод нечіткого дерева рішень допомагає вирішувати завдання прийняття рішень в умовах нечіткості і неоднозначності, що дозволяє покращити якість прийнятих рішень та зробити їх більш обґрунтованими.

Метод нечіткого дерева рішень може полегшити складний процес вибору подарунків та покращити якість прийнятих рішень, сприяючи покращенню відносин у колективі і підвищенню задоволення працівників.

Вибір подарунка, який ідеально відповідає вподобанням працівника, може бути складним завданням через різноманітність можливих варіантів і індивідуальність смаків. Метод нечіткого дерева рішень дозволяє структурувати цей процес і враховувати нечіткі або неоднозначні фактори. Кожна людина унікальна, і метод нечіткого дерева рішень дозволяє створити персоналізовані рекомендації, які враховують індивідуальні вподобання кожного працівника. Це поліпшує якість рекомендацій та може позитивно вплинути на відносини в колективі.

Вибір подарунка може залежати від різних факторів, таких як бюджет, тип події, відносини між співробітниками тощо. Метод нечіткого дерева рішень може об'єднати ці фактори та зробити більш складне прийняття рішень більш системним і обґрунтованим. Метод нечіткого дерева рішень дозволяє додавати нові критерії та розширювати базу знань, що полегшує адаптацію до змін в умовах та вподобаннях співробітників.

Метод нечіткого дерева рішень дозволяє додавати нові критерії та розширювати базу знань, що полегшує адаптацію до змін в умовах та вподобаннях співробітників.

Метод аналізу ієрархії обрано для розробки інформаційної технології завдяки тому, що він дозволяє створювати структуровані та багаторівневі моделі, що відображають різні аспекти вподобань працівника. Наприклад, різні категорії подарунків, бюджет, особисті інтереси, інші фактори, що впливають на вибір. Також важливим є те, що МАІ дозволяє визначити вагомість кожного фактора вподобань працівника, допомагаючи точніше враховувати та вибирати аспекти, які мають найбільший вплив на кінцевий вибір подарунка; надає чіткі та легкі для

розуміння результати, представлені відповідно до вагомості та ієрархії факторів, що полегшує процес прийняття рішень та взаємодію користувача з системою.

МАІ може працювати як з кількісними, так і з якісними даними, що важливо при роботі з інформацією про вподобання та стилі користувача, а також допомагає структурувати та систематизувати процес прийняття рішень, що полегшує аналіз та вибір оптимальних варіантів.

Правильно обрані подарунки можуть позитивно вплинути на задоволення працівників і підвищити їхню мотивацію. Використання методу нечіткого дерева рішень спільно з методом аналізу ієрархії допомагає досягти цієї мети, оскільки він допомагає вибирати більш відповідні подарунки.

2.2 Розробка математичної моделі надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника

Поєднання алгоритмів нечіткого дерева рішень та методу аналізу ієрархій може бути виражено в математичній моделі для надання рекомендацій з прийняття рішень щодо відповідності вибору подарунку до вподобань працівника. Загальний опис такої моделі виглядатиме наступним чином.

Нехай:

C – множина критеріїв, що визначають вибір подарунку;

A – множина альтернатив, тобто варіанти подарунків;

W – множина ваг або важливостей, призначених критеріям та альтернативам.

Таким чином, для кожного критерію $c \in C$ і кожної альтернативи $a \in A$, маємо вагу $w_{c,a} \in W$.

Алгоритм ID3 є одним з ранніх алгоритмів для побудови рішучих дерев рішень [24]. Він використовується для класифікації та прийняття рішень на основі визначених критеріїв. Отож, загальний опис алгоритму ID3 виглядатиме наступним чином:

1. Визначення вхідних даних. Набір даних, що складається з ознак та відповідних класів (міток).
2. Вибір критерію для розбиття. Визначення інформаційного виграшу для кожної ознаки. Інформаційний виграш вимірює, наскільки добре певна ознака розбиває дані на класи. Вибір ознаки з найбільшим інформаційним виграшем для розбиття.
3. Розбиття даних. Розбиття набору даних на підмножини на основі вибраної ознаки.
4. Рекурсивний процес. Повторення процесу для кожної отриманої підмножини даних. Зупинка рекурсії, якщо виконуються певні критерії, такі як досягнення максимальної глибини дерева або поділ даних на підмножини, які є однорідними за класом.
5. Побудова дерева. Створення вузлів дерева для кожної обраної ознаки. Листям дерева стають класи (мітки), до яких належать всі об'єкти в даному вузлі, або, якщо виконані критерії зупинки, визначається класифікація об'єктів у відповідному вузлі.

Алгоритм ID3 використовує ентропію та інформаційний виграш для прийняття рішень про те, як розбити дані та як будувати дерево рішень. Він старається максимізувати інформаційний виграш на кожному кроці, щоб кожне розділення призвело до більш однорідних підмножин даних. Функція інформаційної ентропії обчислюється за формулою:

$$H(S) = - \sum_{i=1}^n p_i \cdot \log_2(p_i) \quad (2.1)$$

де p_i – фактор приналежності даних класу K_i в множині S :

$$p_i = \frac{|S^{K_i}|}{|S|} \quad (2.2)$$

$$E(A_i, S) = \sum_{j=1}^m (p_{ij} \cdot H(S_{F_{ij}})) \quad (2.3)$$

де p_{ij} – фактор приналежності нечіткого набору j серед всіх можливих значень i :

$$p_{ij} = \frac{|S_{F_{ij}}|}{\sum_{j=1}^m |S_{F_{ij}}|} \quad (2.4)$$

Далі відбувається поділ множини S на нечіткі множини S_1, S_2, \dots, S_m відповідно до обраного максимізованого атрибута, де значення ступеню впевненості обчислюється як добуток ступеня належності множини S на до значення $F_{max,j}$ для A_{max} в S . При цьому максимізований атрибут A_{max} виключається з подальшого розгляду. Під час вибору подарунку слід враховувати, що деякі атрибути можуть вимагати використання двовимірного масиву як елементів нечітких підмножин, що може викликати деякі труднощі. Наприклад, якщо розглядається атрибут «вік», то необхідно перетворити матрицю належності цього атрибуту віку до окремого числа – ступеня впевненості μ_a . Для досягнення цього використовується зведення матриці:

$$\mu_{a_1} = \frac{1}{q} \sum_i^n \sum_j^n v_{ij} f_{ij}^{(a_1)} \in [0; 1] \quad (2.5)$$

де $V = ||v_{ij}||$ – матриця віку,

$F_{a_1} = |f_{ij}^{(a_1)}|$ – матриця типізації віку.

q в даному випадку буде дорівнювати кількості комірок матриці віку.

Після проведення наведених розрахунків проводиться генерація нових вузлів t_1, t_2, \dots, t_m для нечітких множин S_1, S_2, \dots, S_m та присвоєння ребру, яке з'єднає вузли, що розглядаються, значення нечіткої множини $F_{max,j}$.

Після цього відбувається заміна S на $S_j (j = 1, 2, \dots, m)$ та повернення до циклу, коли розглядається поточний вузол t .

Значення параметрів θ_n та θ_r вибираються експериментально. Кожен листок нечіткого дерева рішень містить інформацію про ступінь приналежності вподобань працівника та тип подарунку.

Згідно методу аналізу ієрархій необхідно створити ієрархічну структуру для критеріїв та альтернатив. Далі потрібно визначити матриці попарних порівнянь для критеріїв та альтернатив.

Нехай N – множина альтернатив або критеріїв, тобто можливих подарунків та їх характеристик. Тоді A – матриця попарних порівнянь для альтернатив або критеріїв.

Далі проводиться створення шкали для визначення відносної важливості пар альтернатив або критеріїв, вводяться ваги або коефіцієнти важливості, які представляють важливість одного елемента в порівнянні з іншим.

Після цього відбувається створення матриці попарних порівнянь. Кожен елемент матриці $A_{i,j}$ вказує, наскільки важливішою є i -та альтернатива чи критерій порівняно з j -тою [25]. Далі відбувається нормалізація матриці для отримання індексу узгодженості та вектора ваг, проводиться розрахунок ваг для альтернатив чи критеріїв на основі вектора ваг та інших факторів (якщо необхідно). Враховуючи ваги та важливість кожної альтернативи чи критерію, можна прийняти рішення щодо відповідності вибору подарунку вподобанням працівника.

Обчислення ваг критеріїв опишемо наступною формулою:

$$W_c = \frac{\text{сума по рядку } i \text{ в матриці}}{\text{сума всіх елементів матриці}}; \quad (2.6)$$

Тоді як обчислення ваг альтернатив буде описано:

$$W_a = \frac{\text{сума по рядку } i \text{ в матриці}}{\text{сума всіх елементів матриці}}; \quad (2.7)$$

Зважені суми для кожної альтернативи обчислюють як:

$$S_a = \sum_{c \in C} (W_c \cdot \mu_{c,a}(x)) \cdot W_a \quad (2.8)$$

Альтернатива з максимальною зваженою сумою буде рекомендованою. Такий підхід дозволяє поєднати нечітке моделювання вподобань працівника з аналізом ієрархій для визначення вагового коефіцієнта кожного аспекту та альтернативи та дозволяє допомогти системі рекомендацій адаптувати свої пропозиції відповідно до індивідуальних вподобань користувача на основі побудованого рішучого дерева та важливості критеріїв, визначеної методом аналізу ієрархій.

2.3 Проектування структури інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника

Створення програмного забезпечення для надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка вподобанням працівника може включати в себе різні компоненти та етапи. Важливим етапом є збір інформації про вподобання, хобі, анкетні дані. Це може бути реалізовано за допомогою процесу анкетування працівника, де він відповідає на підготовлені питання. Аналіз характеристик для рекомендацій, наприклад, бюджет, тип подарунка, особливі вподобання впливає на пошук відповідного подарунка.

Наступний етап включає обробку та аналіз інформації про працівника та параметрів подарунка. За використання комбінації алгоритмів ID3 та MAI враховуються індивідуальні вподобання працівників до рекомендацій для надання рекомендацій. Після цього відбувається відображення списку рекомендованих подарунків для користувача.

Структурну схему програмного забезпечення інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника наведено на рисунку 2.1.

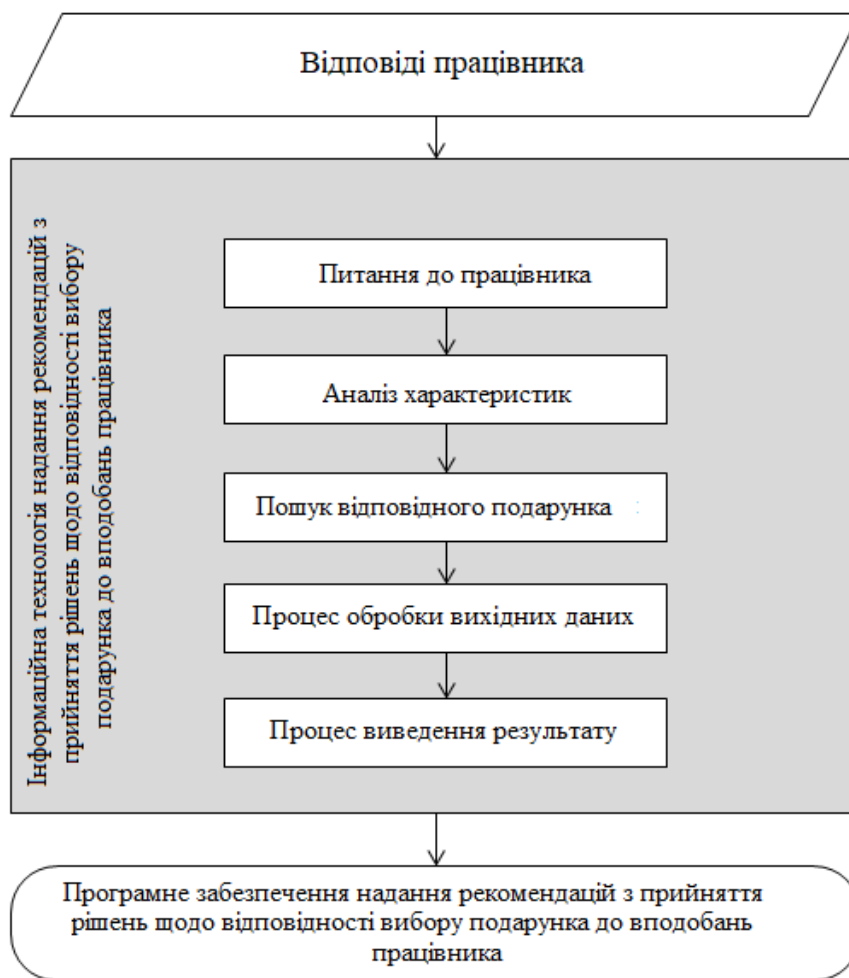


Рисунок 2.1 – Структурна схема інформаційної технології

Для зручності використання програмного забезпечення надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника було вирішено скористатися можливостями створення чот-боту для платформи Telegram. Цей підхід має кілька переваг порівняно з аналогами, таких як зручність і зрозумілість використання та доступність для користувачів Telegram. Чат-боти в Telegram – це програмні інструменти, які реалізують функціонал, визначений у чаті, отримуючи та оброблюючи повідомлення та команди від користувача та надаючи відповіді у вигляді обробленої інформації. Іншими словами, чат-боти – це засоби виконання завдань, які можна формалізувати у вигляді команд та обробити.

Структура інформаційної технології надання рекомендацій з прийняття рішень щодо вибору подарунка до вподобань працівника може включати кілька

ключових компонентів та процесів. Ось загальна структура, яку можна використовувати для такого рішення:

1. Взаємодія з користувачем (користувачами). Цей етап включає в себе взаємодію з користувачем за допомогою телеграм-чат-бота. Користувачі будуть вводити свої вподобання та іншу інформацію, пов'язану з вибором подарунка.

2. Збір і обробка даних. Збір даних включає в себе отримання від користувачів інформації про їхні вподобання, які можуть бути відображені у профілі користувача. Обробка даних може включати в себе аналіз інформації про вподобання, групування і класифікацію даних.

3. Система рекомендацій. Розробка алгоритмів рекомендацій, які враховують вподобання користувачів і можуть рекомендувати подарунки, які їм сподобаються. Це може включати алгоритми машинного навчання, які аналізують історію вибору подарунків та вподобань користувача.

4. База даних. Зберігання інформації про користувачів, їхні вподобання та історію вибору подарунків у базі даних.

5. Інтеграція з Telegram API. Розробка інтеграції з Telegram API для отримання та надсилання повідомлень користувачам.

6. Інтерфейс користувача. Створення інтерфейсу чат-бота для взаємодії з користувачами, відображення рекомендацій і можливість користувачів вибирати подарунки.

7. Інтерфейс адміністратора. Розробка інтерфейсу для адміністраторів, де вони можуть керувати користувачами, перевіряти статистику та додавати нові подарунки до системи.

8. Система безпеки. Забезпечення захисту конфіденційної інформації користувачів та інших даних.

9. Система логування і аналізу. Логування дій користувачів та аналіз статистики для покращення алгоритмів рекомендацій.

10. Тестування і вдосконалення. Проведення тестування чат-бота та системи рекомендацій і внесення виправлень та покращень на основі результатів тестування.

11. Розгортання і підтримка. Розгортання чат-бота на сервері та надання підтримки для користувачів.

Загалом, структура інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника може бути складною, але це проект, який може бути дуже корисним для роботодавців та працівників, які шукають подарунки.

На рисунку 2.2 наведено загальну структурну схему програмного забезпечення.

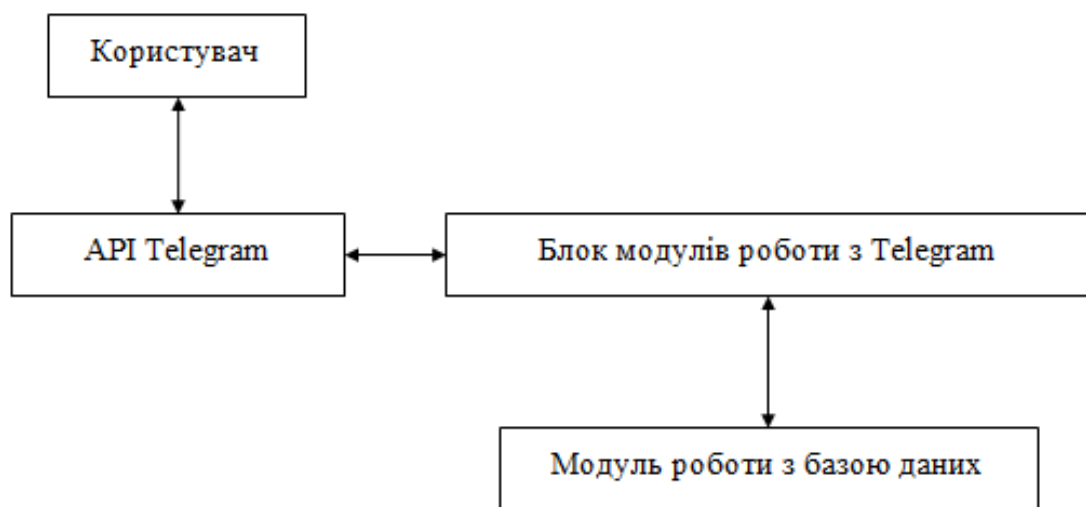


Рисунок 2.2 – Загальна структурна схема

Запропонована структура ілюструє взаємодію користувача з програмним забезпеченням і включає блок модулів, які працюють у середовищі Telegram. Ці модулі відповідають за введення персональних даних, прийняття рішення про вибір подарунка та взаємодію з модулем бази даних..

Модуль бази даних встановлює зв'язок з базою даних, яка містить дані про подарунки та їх характеристики. У цьому модулі реалізовано SQL-запити для обробки даних.

Для забезпечення коректної роботи програмного забезпечення була розроблена методика вибору подарунка на основі вподобань користувача на основі нечіткого дерева рішень за допомогою Telegram Bot API. Цей API надає такі функції:

1. Передача інформації з Telegram на сервер і навпаки.
2. Надання можливості введення особистих даних (вік, стать, посада, наявність коханої людини та дітей).
3. Збір інформації про вподобання співробітників.
4. Прийняття рішення про вибір подарунка відповідно до вподобань опитаного співробітника.
5. Виведення та зберігання результатів на сервері.

Інформаційна технологія надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника матиме два види користувачів: адміністратор і працівник.

Діаграма варіантів використання наведена на рисунку 2.3.

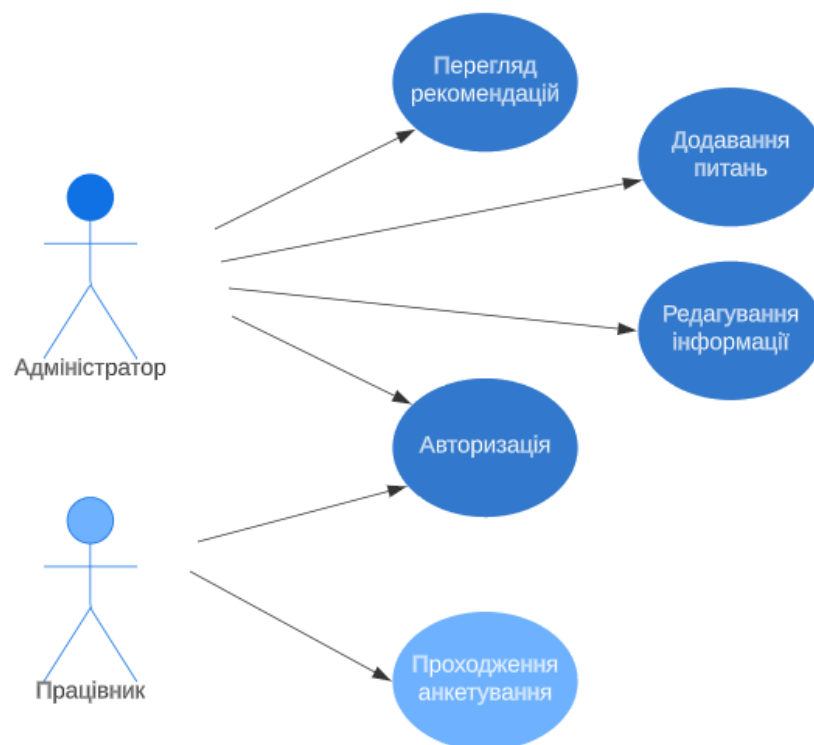


Рисунок 2.3 – Діаграма варіантів використання
Адміністратор матиме можливість виконувати наступні дії:

- можливість редагування інформації;
- можливість перегляду рекомендацій;
- можливість додавання нової інформації (питань);

Користувач матиме обмежені авторизацією можливості, для нього реалізовано тільки можливість авторизації та проходження анкетування.

Отже, визначено види користувачів інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника, на основі чого було створено діаграму варіантів використання інформаційної технології.

2.4 Розробка алгоритму функціонування інформаційної технології

Графічне представлення загального алгоритму функціонування інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника зображено на рисунку 2.4.

Даний алгоритм складається з наступних етапів:

1. Початок роботи програми. Користувач авторизується, після чого програма визначає його тип.
2. Перевірка прав адміністратора. Якщо користувач не має прав адміністратора, відбувається перехід до пункту 7.
3. Обробка відповідей працівників, які пройшли анкетування.
4. Інтелектуальний підбір подарунків на основі алгоритму нечіткого дерева рішень та методу аналізу ієрархій.
5. Формування результату.
6. Виведення результату. Рекомендація з прийняття рішень щодо подарунка, відповідного вподобанням працівника. Закінчення діалогу.
7. Початок анкетування працівника, що немає прав адміністратора.
8. Поступове задання основних питань.
9. Перевірка на необхідність додаткових питань. Якщо немає такої потреби – перехід до пункту 11.
10. Задання додаткових питань.

11. Завершення анкетування. Бот повідомляє про завершення анкетування та завершує діалог.

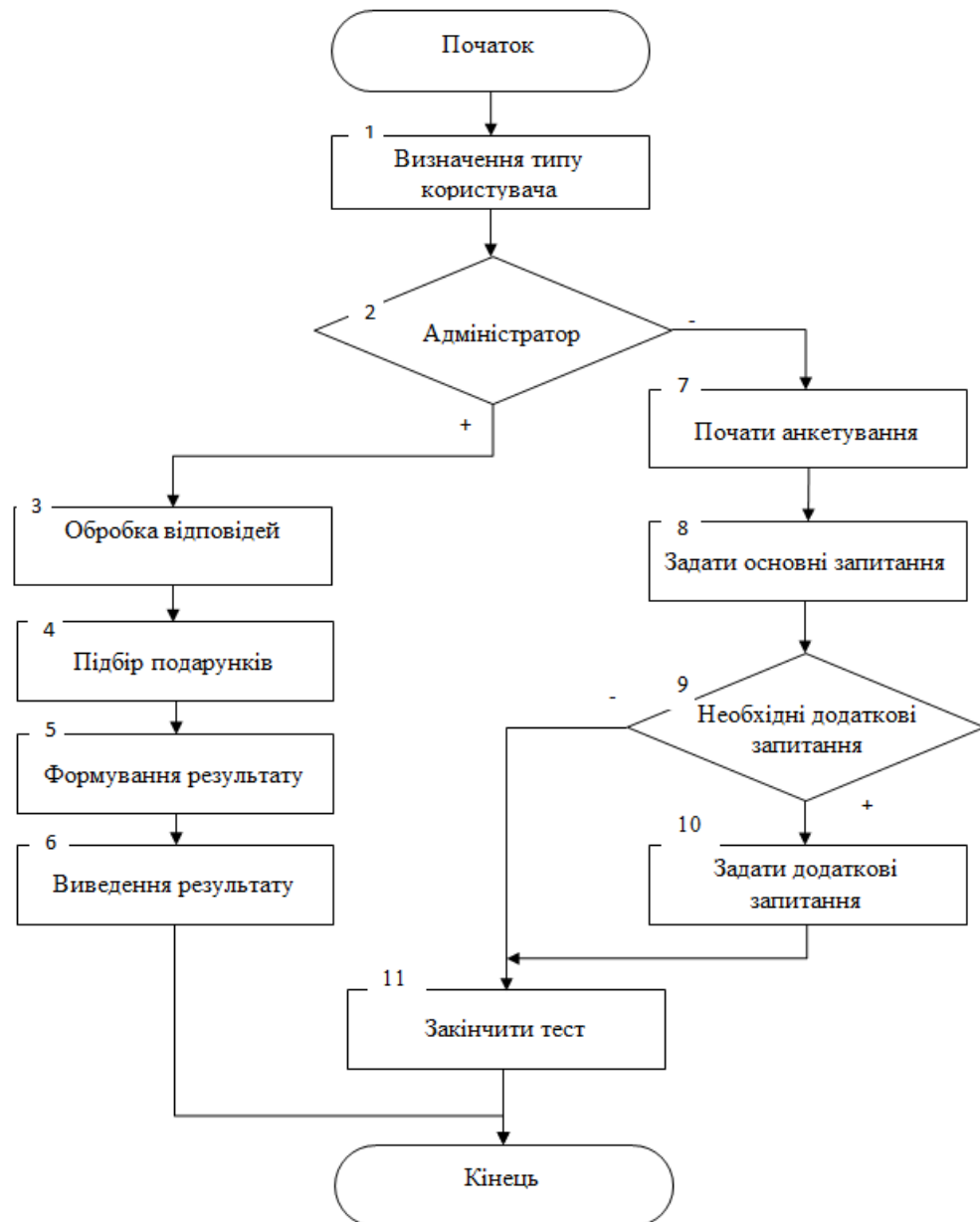


Рисунок 2.4 – Схема загального алгоритму функціонування інформаційної технології

Як було сказано раніше, користувачі матимуть два види доступу: адміністратор і працівник. На рисунку 2.5 наведено алгоритм роботи інформаційної технології для ролі працівника.

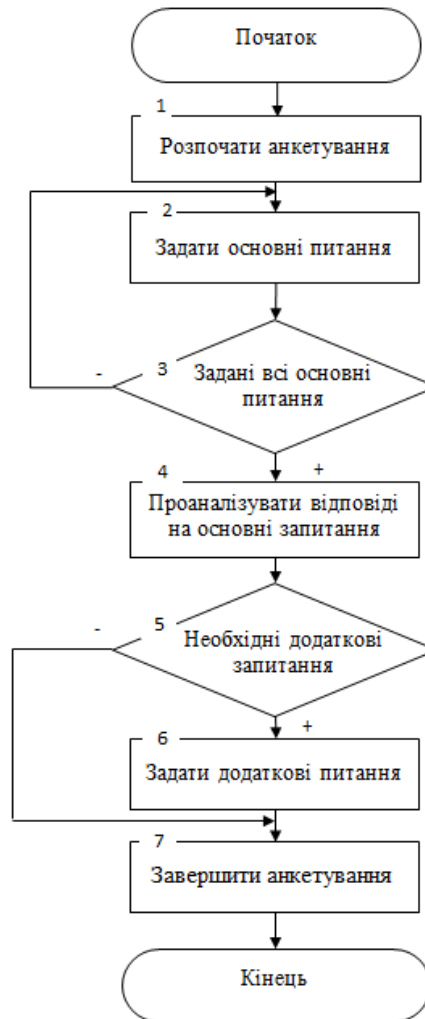


Рисунок 2.5 – Схема алгоритму роботи інформаційної технології для ролі працівника

Даний алгоритм складається з наступних кроків:

1. Початок роботи програми. Розпочинається процес анкетування працівника.
2. Програма починає задавати основні питання працівнику.
3. Цикл опитування. Система проводить цикл опитування, збираючи відповіді на особисті дані та вподобання працівника.
4. Система аналізує отримані відповіді на опитування.
5. Перевірка на необхідність додаткових запитань. Система визначає, чи необхідно задати уточнюючі запитання.

6. Процес покрокового опитування працівника для прийняття рішень щодо відповідності вибору подарунку його вподобанням.

7. Аналіз відповідей на додаткові запитання. Система аналізує відповіді на додаткові запитання.

8. Виведення результатів анкетування. Результати анкетування виводяться у вигляді рекомендацій щодо вибору подарунку, який відповідає вподобанням працівника.

Вибірка параметрів для формування запитань враховує індивідуальність кожного працівника та орієнтується на різноманітні аспекти, що можуть впливати на його вподобання та інтереси. Формування запитань для працівників впроваджується через вивчення важливих аспектів їхнього індивідуального стилю та інтересів. Процес починається із визначення стандартних параметрів, але перед початком пошуку ідеального подарунка варто враховувати переваги, які колеги можуть цінувати. Основна мета – забезпечити усвідомлений та індивідуальний вибір, спрямований на задоволення потреб та вражень отримувача.

Ділова активність та обов'язки на роботі визначаються як ключові елементи для визначення відповідних подарунків. Це може включати предмети, корисні на робочому місці, або декор для створення комфортного робочого середовища. Важливо також врахувати, що персоналізований подарунок повинен відображати індивідуальність отримувача, і тому формулюються запитання, спрямовані на вивчення його особистих параметрів та вподобань.

Хоча особисті параметри, такі як вік, стать, та сімейний стан, можуть впливати на вибір, для колег часто є характерною незалежність від цих факторів. Тим не менш, анкета, яка спрямована на підбір подарунка для колег, часто розпочинається запитаннями щодо статі та віку. Це відображає усвідомлення того, що інтереси можуть відрізнятися в залежності від вікової групи та статі, допомагаючи створити персоналізовані та унікальні рекомендації для працівників у різних життєвих етапах. Формування запитів користувачів показано на рисунку 2.6.

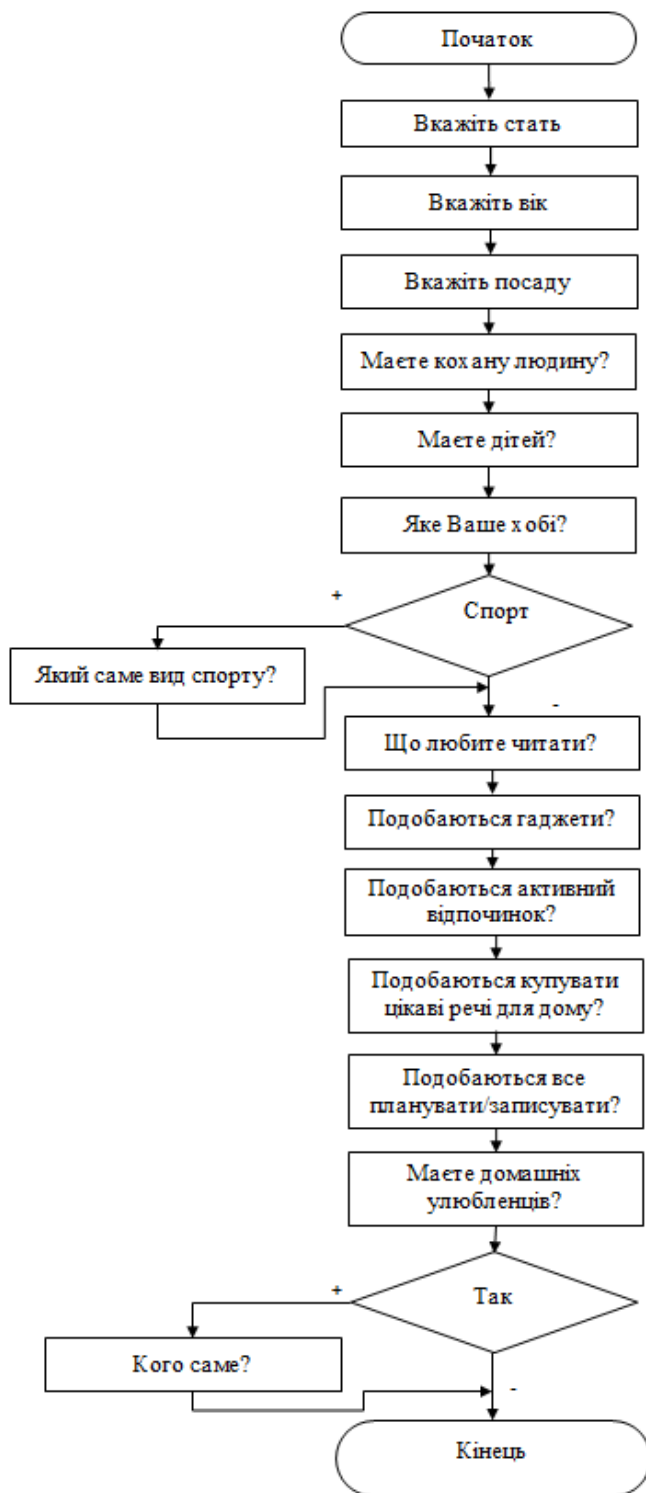


Рисунок 2.6 – Схема формування запитань

Серйозна увага до інтересів та захоплень колеги є важливим етапом у процесі вибору подарунка, оскільки це може значно покращити якість вибору та зробити його більш збалансованим і вдалим. Врахування хобі та вподобань

отримувача глибше перевіряється, сприяючи створенню подарунків, які не лише вражають зовнішнім виглядом, а й виражають індивідуальність та увагу до унікальних рис отримувача.

Подарунки, обрані відповідно до хобі чи захоплень, мають потужний ефект особистого зв'язку. Такі подарунки відзначаються вищою ступенем персоналізації, що робить їх особливо цінними для отримувача. Це свідчить про те, що ви вкладаєте зусилля у вивчення індивідуальних характеристик і стаєте ближче до серця отримувача через вибір подарунка, який відзначається увагою до його особистих захоплень і інтересів.

Підсумовуючи, врахування інтересів та захоплень колеги допомагає підняти вибір подарунка на вищий рівень, створюючи не тільки приємний сюрприз, але й підкреслюючи ваше бажання подарувати щось особливе та унікальне.

2.5 Обґрунтування вибору бази даних для зберігання даних для інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника

Вибір бази даних для зберігання даних інформаційній технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника – це важливий крок у розробці проекту.

В базі даних має зберігатися інформація про користувачів, їхні вподобання, історію вибору подарунків, можливі подарунки та інші відомості. База даних повинна підтримувати ці дані та забезпечувати зручний доступ до них.

Для реалізації інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника вибрано реляційну базу даних.

Реляційна база даних – це тип бази даних, яка зберігає та надає доступ до точок даних, пов'язаних одна з одною. Реляційні бази даних базуються на реляційній моделі, інтуїтивно зрозумілому, простому способі представлення

даних у таблицях. У реляційній базі даних кожен рядок у таблиці є записом з унікальним ідентифікатором, який називається ключем. Стівпці таблиці містять атрибути даних, і кожен запис зазвичай має значення для кожного атрибута, що полегшує встановлення зв'язків між точками даних.

Реляційна база даних є типом бази даних, яка використовує модель даних, відому як «реляційна модель». Ця модель базується на використанні таблиць, де дані організовані в рядки і стівпці. Відносини між даними визначаються за допомогою ключів [26]. Реляційна модель означає, що логічні структури даних – таблиці даних, подання та індекси – відокремлені від фізичних структур зберігання. Це розділення означає, що адміністратори баз даних можуть керувати фізичним зберіганням даних, не впливаючи на доступ до цих даних як до логічної структури. Наприклад, перейменування файлу бази даних не перейменовує таблиці, що зберігаються в ньому.

Розрізнення між логічним і фізичним також стосується операцій бази даних, які є чітко визначеними діями, які дозволяють програмам маніпулювати даними та структурами бази даних. Логічні операції дозволяють додатку вказувати потрібний вміст, а фізичні операції визначають спосіб доступу до даних і виконання завдання.

Щоб забезпечити постійну точність і доступність даних, реляційні бази даних дотримуються певних правил цілісності. Наприклад, правило цілісності може вказувати, що повторювані рядки не допускаються в таблиці, щоб виключити можливість потрапляння помилкової інформації в базу даних.

Реляційна модель даних забезпечувала стандартний спосіб представлення та запиту даних, який міг використовувати будь-який додаток. З самого початку розробники визнали, що головною перевагою моделі реляційної бази даних є використання таблиць, які є інтуїтивно зрозумілим, ефективним і гнучким способом зберігання структурованої інформації та доступу до неї.

Згодом з'явилася ще одна сильна сторона реляційної моделі, коли розробники почали використовувати мову структурованих запитів (SQL) для запису та запиту даних у базі даних. Протягом багатьох років SQL широко

використовувався як мова для запитів до бази даних. Базуючись на реляційній алгебрі, SQL забезпечує внутрішню послідовну математичну мову, яка полегшує покращення продуктивності всіх запитів до бази даних. Для порівняння, інші підходи повинні визначати окремі запити.

SQL (Structured Query Language): SQL є стандартною мовою для роботи з реляційними базами даних. Вона використовується для створення, читання, оновлення та видалення даних. SQL включає такі команди, як SELECT для отримання даних, INSERT для вставки нових записів, UPDATE для оновлення існуючих даних, та DELETE для видалення записів.

SQL – це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також управління базами даних [27]. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення, і видалення даних, використовуючи систему управління і адміністративні функції. SQL також включає CLI (Call Level Interface) для доступу і управління базами даних дистанційно.

MySQL - це одна з найпопулярніших реляційних систем управління базами даних (RDBMS). Наведемо деякі з основних переваг MySQL:

- Відкриті джерела та безкоштовність. Це робить його доступним для багатьох проектів та організацій.
- Добре документований. MySQL має багато документації, яка допомагає розробникам розуміти його функції та використовувати його в їхніх проектах.
- Висока продуктивність. MySQL відомий своєю високою продуктивністю та швидкодією. Він може обробляти велику кількість одночасних підключень і запитів.
- Підтримка транзакцій. MySQL підтримує транзакції, що дозволяє вам виконувати групу операцій як атомарну одиницю, забезпечуючи цілісність даних.
- Реплікація. MySQL підтримує реплікацію, що дозволяє створити декілька копій бази даних для покращення доступності та надійності.

- Широкий спектр опцій зберігання. MySQL підтримує різні типи таблиць, включаючи InnoDB, MyISAM, та інші, що дає можливість вибирати оптимальний тип для вашого використання.
- Активна спільнота та підтримка. Існує велика та активна спільнота користувачів і розробників, що дозволяє швидко отримувати підтримку та розв'язувати проблеми.
- Підтримка багатьох операційних систем. MySQL доступний для багатьох операційних систем, включаючи Windows, Linux, macOS та інші.
- Сумісність з багатьма мовами програмування. MySQL підтримує багато мов програмування, включаючи PHP, Python, Java, та інші.
- Безпека. MySQL має різні механізми безпеки, такі як автентифікація, авторизація та шифрування даних.
- Масштабованість. MySQL може бути використаний як для невеликих проєктів, так і для великих систем з великою кількістю даних. Він підтримує різні методи масштабування.
- Інструменти для адміністрування. Існують різні інструменти для адміністрування та моніторингу MySQL, такі як phpMyAdmin, MySQL Workbench і багато інших.

Зазначені вище переваги роблять MySQL популярним вибором для багатьох додатків та проєктів, де потрібна реляційна база даних.

До недоліків SQL можна віднести: складність, відхилення від стандартів та складність роботи з ієрархічними структурами.

2.6 Висновок до розділу 2

Для розробки інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунку до вподобань працівника необхідно застосовувати методи, що базуються на ймовірнісних методах, зокрема, метод нечіткого дерева рішень та метод аналізу ієрархій, комбінація яких повністю відповідає поставленим вимогам. Згідно специфіки задачі було запропоновано математичну модель для даної інформаційної технології. Побудована модель

може допомогти системі рекомендацій адаптувати свої пропозиції відповідно до індивідуальних вподобань користувача на основі побудованого нечіткого дерева рішень та важливості критеріїв, визначених методом аналізу ієрархій.

Відповідно до вищесказаного було спроектовано структуру інформаційної технології, розроблено схему загального алгоритму функціонування системи, схему алгоритму роботи інформаційної технології для ролі працівника та схему формування запитань, наведено діаграму варіантів використання інформаційної технології. Також обґрунтовано вибір бази даних для інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунку до вподобань працівника.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДАННЯ РЕКОМЕНДАЦІЙ З ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ВІДПОВІДНОСТІ ВИБОРУ ПОДАРУНКА ДО ВПОДОБАНЬ ПРАЦІВНИКА

3.1 Обґрунтування вибору мови та середовища програмування

Мови програмування Java і Python є популярними виборами для створення чат-ботів в Telegram. Обидві мови мають свої переваги, і вибір залежатиме від уподобань та досвіду розробника, а також конкретних вимог проекту.

Java – це мова програмування, яку розробники використовують для створення різноманітних програм для комп'ютерів, смартфонів, планшетів та інших розумних пристроїв [28]. Вона є однією з найпопулярніших мов програмування у світі завдяки своїй платформонезалежності та широкому використанню в різних галузях індустрії. Особливість Java полягає в тому, що програми, написані на цій мові, можуть працювати на будь-якому комп'ютерному пристрої з різними операційними системами, і це можливо без необхідності перекомпілювати їхній вихідний код.

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня зі суворою динамічною типізацією [29]. Python відзначається простим та читабельним синтаксисом, що робить його ідеальним для початківців у програмуванні і зрілими розробниками. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, аспектно-орієнтована та функціональна.

Наведемо короткий огляд переваг кожної мови для створення Telegram чат-ботів:

Python:

Простота та читабельність. Python відомий своєю простотою синтаксису, що робить його дуже доступним для початківців. Це допомагає швидко створювати прості чат-боти.

Багатий вибір бібліотек. Python має багатий вибір бібліотек для створення чат-ботів, включаючи "python-telegram-bot." Ці бібліотеки спрощують взаємодію з Telegram API.

Інтерактивність. Python ідеально підходить для створення інтерактивних чат-ботів, оскільки він дозволяє швидко відповідати на повідомлення користувачів.

Швидка розробка. Python дозволяє розробникам розробляти прототипи та експериментувати швидше завдяки своїй простоті.

Java:

Вища продуктивність. Java зазвичай має вищу продуктивність в порівнянні з Python, особливо при обробці великої кількості даних або одночасних запитів. Це важливо для чат-ботів з великою кількістю користувачів.

Багатопоточність. Java має потужну підтримку багатопоточності, що дозволяє боту обробляти багато одночасних запитів в Telegram без затримок.

Безпека. Java має вбудовані засоби для захисту від атак та вразливостей, що дозволяє розробникам створювати безпечні додатки, включаючи чат-боти.

Багатофункціональність. Java має багатий екосистему бібліотек і фреймворків, що спрощують розробку чат-ботів. Для взаємодії з Telegram API, ви можете використовувати бібліотеки, такі як "TelegramBots," які спрощують процес.

Отже, для розробки програмного забезпечення, яке надає рекомендації з прийняття рішень щодо вибору подарунка до вподобань працівника у вигляді Telegram чат-бота, мова програмування Java може бути відмінним вибором. Java є потужною та популярною мовою програмування, яка має багатий набір бібліотек і фреймворків, що допомагають в розробці чат-ботів та інтерактивних додатків.

Розглядаючи можливість використання мов програмування Java або Python для створення чат-бота в Telegram, варто врахувати наступні переваги Java в порівнянні з Python: вища продуктивність, багатопоточність, більше контролю над ресурсами, безпека, багатофункціональність, інтеграція з іншими технологіями.

IntelliJ IDEA – це інтегроване середовище розробки (IDE) для мови програмування Java і інших технологій [30]. Воно розроблене компанією JetBrains і вважається одним із найпопулярніших та потужних інструментів для розробки на Java. Ось деякі ключові особливості IntelliJ IDEA:

Розширена підтримка Java. IntelliJ IDEA надає високий рівень підтримки мови Java, включаючи інтелектуальне автодоповнення, аналіз коду, рефакторинг і інші інструменти, які полегшують розробку.

Підтримка інших мов та технологій. IntelliJ IDEA не обмежується лише Java. Воно підтримує багато інших мов програмування, таких як Kotlin, Scala, Groovy, а також технології для веб-розробки, мобільної розробки та інше.

Автоматичний рефакторинг. IntelliJ IDEA має широкий набір інструментів для автоматичного рефакторингу коду, що допомагає покращити якість і читабельність вашого коду.

Підтримка фреймворків. IntelliJ IDEA включає в себе підтримку багатьох популярних фреймворків, таких як Spring, Hibernate, Android SDK, і багато інших.

Інтегрована система збудування та розгортання. Ви можете здійснювати збірку та розгортання своїх проектів прямо з середовища розробки, що спрощує робочий процес.

Версійний контроль. IntelliJ IDEA підтримує різні системи версійного контролю, такі як Git, Subversion, Mercurial і багато інших.

Плагіни. Ви можете розширити функціональність IntelliJ IDEA за допомогою плагінів. JetBrains і розробники створили широкий спектр плагінів для різних мов і фреймворків.

Спільнота і підтримка. IntelliJ IDEA має велику і активну спільноту користувачів і надає високоякісну технічну підтримку.

IntelliJ IDEA доступний у безкоштовній версії (Community) та платній версії (Ultimate) з додатковими функціями для розробки підприємницьких додатків та інтеграції з різними технологіями. Для розробки чат-ботів в Java або інших проєктів, IntelliJ IDEA може бути відмінним інструментом розробки з багатим набором функцій.

Отже, завдяки перерахованим вище перевагам, для розробки інформаційної технології надання рекомендацій з прийняття рішення щодо відповідності вибору подарунку до вподобань працівника було вибрано об'єктно-орієнтовану мову програмування Java та середовище розробки IntelliJ IDEA.

3.2 Обґрунтування вибору бібліотек та фреймворків для інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунку до вподобань працівника

Для створення Telegram чат-бота для надання рекомендацій щодо відповідності вибору подарунку до вподобань працівника з використанням мови програмування Java, необхідно вибрати певні бібліотеки і фреймворки.

Такими бібліотеками можуть бути TelegramBots, Spring Framework, JPA, Maven, NLP та інші.

Бібліотека TelegramBots є популярною Java бібліотекою для взаємодії з Telegram API. Вона надає простий інтерфейс для створення та керування чат-ботами в Telegram. Цю бібліотеку використовують для створення чат-бота та обробки повідомлень від користувачів.

Основними компонентами бібліотеки є:

- TelegramLongPollingBot – основний клас для створення Telegram бота. Він реалізує інтерфейс `org.telegram.telegrambots.bots.TelegramLongPollingBot` та надає методи для обробки вхідних повідомлень;
- Update – клас, який представляє вхідне повідомлення від Telegram API. Об'єкт класу Update передається в методи бота для подальшої обробки;

- `SendMessage` – клас для визначення повідомлення, яке буде відправлене користувачу. Містить методи для налаштування тексту, клавіатури та інших властивостей повідомлення;
- `ReplyKeyboardMarkup` – клас для створення клавіатури в Telegram боті. Дозволяє визначити кнопки та їх розміщення;
- `BotApiMethod` – абстрактний клас, який є базовим для всіх методів виклику Telegram API. Використовується для визначення різних дій, таких як відправка повідомлення, оновлення клавіатури тощо.

Maven і Gradle – це інструменти для автоматизації збирання та управління залежностями проекту. Вони допоможуть керувати залежностями проекту та легко встановлювати необхідні бібліотеки.

Maven використовує концепцію «об'єктно-орієнтованого проектування», де проект розглядається як структурований набір залежностей та конфігурацій. Maven визначає стандартний життєвий цикл збірки проекту, який включає такі етапи, як компіляція, тестування, упаковка та розгортання. Це спрощує процес збірки та розгортання проектів. Maven підтримує використання плагінів, які надають різноманітні функції під час збірки проекту. Він використовує центральний репозиторій Maven для зберігання та вибіркового завантаження бібліотек та залежностей.

Бібліотеки для аналізу даних. Якщо планується аналізувати дані про вподобання працівників для надання рекомендацій, можливо, знадобиться бібліотека для обробки даних, така як Apache Spark або Apache Flink.

Для створення системи рекомендацій можна використовувати бібліотеки, такі як Apache Mahout або LensKit, які надають можливість розробити алгоритми рекомендацій для чат-бота.

Існує кілька фреймворків для створення веб-сервісів, але серед них одним з найпопулярніших для мови програмування Java є Spring Framework [31]. Spring Framework визнаний одним з найбільш використовуваних фреймворків для розробки на Java. У своїй роботі Spring застосовується для керування бізнес-логікою чат-бота, інтеграції з базами даних та створення RESTful API для

взаємодії з ботом. Також підтримуються об'єктно-орієнтована технологія відображення та інтерфейс JPA.

Spring Framework є одним з найбільш популярних та використовуваних фреймворків для розробки програмного забезпечення на мові програмування Java. Його основна мета – полегшити розробку великих та складних програмних систем, враховуючи принципи інверсії управління та введення відображення.

Ключовим концептом у Spring є інверсія управління. Замість того, щоб програма сама створювала об'єкти, Spring надає контейнер, який управляє створенням та управлінням об'єктами. Це полегшує тестування та робить систему менш залежною від конкретних класів.

Spring надає альтернативу важким EJB-компонентам, дозволяючи розробникам створювати легкі, прості в управлінні об'єкти. Spring надає підтримку для декларативного та програмного управління транзакціями, що полегшує взаємодію з базою даних та іншими ресурсами.

Для спрощення доступу до даних в реляційних та нереляційних базах даних використовується компонент Spring Data, а для реалізації функцій аутентифікації та авторизації – Spring Security. Spring Boot – це розширення Spring, яке надає конвенції для швидкого старту розробки та робить ряд задач, таких як налаштування вбудованого сервера, дуже простими.

JPA (Java Persistence API) або Hibernate. Якщо чат-бот потребує доступу до бази даних для зберігання інформації про вподобання працівників і подарунки, знадобиться рішення ORM (Object-Relational Mapping). JPA або Hibernate допоможуть зробити це легше та ефективніше.

Далі розглянемо Spring Boot Starter. Starters у Spring Boot – це, так би мовити, збірники зручних конфігураційних налаштувань, які дозволяють легко інтегрувати необхідні залежності у ваш проект. Вони представляють собою зручний і централізований інструментарій, що об'єднує всі необхідні бібліотеки та технології Spring, щоб ви могли віддавати більше часу написанню функціоналу свого додатка, а не витрачати його на конфігурацію.

Припустімо, вам потрібно використовувати Spring разом із JPA для зв'язку з базою даних. Включаючи в проект залежність `spring-boot-starter-data-jpa`, ви автоматично включаєте всі необхідні компоненти для роботи з базою даних з урахуванням усіх нюансів конфігурації. Вам не потрібно самостійно вибирати та додавати кожен окрему залежність; Стартери організують це для вас [32]. Слід зазначити, що лаунчери, створені сторонніми розробниками, мають свої особливості. Їм не дозволяється починати з Spring Boot, оскільки це зарезервовано для офіційних артефактів Spring Boot. Натомість їм зазвичай дають назву конкретного проекту. Наприклад, якщо сторонній розробник створює стартовий проект під назвою `thirdpartyproject`, має сенс назвати його `partyproject-spring-boot-starter`.

Таким чином, використання Spring Boot Starter робить процес налаштування та включення необхідних бібліотек приємнішим та ефективнішим, генеруючи функціональний код вашої програми швидше.

Далі розглянемо модуль під назвою `spring-boot-devtools`. Spring Boot DevTools – це додатковий модуль, який надає ряд інструментів для полегшення розробки додатків на базі Spring Boot. Ці інструменти дозволяють покращити ефективність та швидкість розробки, забезпечуючи ряд зручностей.

Основна ідея DevTools – це підтримка гарячого перезавантаження (Hot Swapping) вашого додатка під час розробки. Гаряче перезавантаження дозволяє вам вносити зміни до вашого коду і переглядати їх в реальному часі без перезапуску всього додатка. Це суттєво зменшує час, який ви витрачаєте на очікування перезапуску, і полегшує процес виправлення помилок та експериментів з кодом.

Додатково, DevTools також надає інші корисні функції, такі як: автоматичне перезавантаження браузера (LiveReload), що може автоматично перезавантажувати сторінку браузера після кожної зміни; підтримка теплового перезапуску (LiveRestart) що дозволяє автоматично перезапускати додаток, зберігаючи при цьому стан додатка. Це допомагає уникнути повного перезапуску, який може бути витратним з точки зору часу; виявлення змін класів у режимі

реального часу, тобто можливість автоматично виявляти зміни у кодї та виконувати гаряче перезавантаження тільки тих частин, які зазнали змін, замість повного перезапуску; підтримка віддаленої розробки (Remote Development), що дозволяє використовувати DevTools для віддаленої розробки, коли додаток виконується на віддаленому сервері.

Використання DevTools може значно покращити продуктивність розробників, зменшуючи час, необхідний для здійснення змін та тестування коду.

Наступною розглянемо бібліотеку Lombok. Lombok – це бібліотека для мови програмування Java, яка допомагає розробникам зменшити кількість шаблонного коду, який потрібно писати. Основна ідея Lombok полягає в автоматизації створення стандартних методів та конструкторів, що дозволяє розробникам фокусуватися на бізнес-логіці, а не на писанні рутинного коду. [33].

Основні функції та анотації Lombok:

- @Getter і @Setter: Анотації для автоматичного створення методів-геттерів і сеттерів для полів класу.
- @NoArgsConstructor, @AllArgsConstructor, @RequiredArgsConstructor: Анотації для автоматичного створення конструкторів без параметрів, з параметрами або лише з обов'язковими параметрами.
- @ToString: Генерує стандартний метод toString() для класу.
- @Builder: Дозволяє використовувати паттерн Builder для створення об'єктів класу.

Ці анотації дозволяють уникнути багато рутинного коду, який часто вводиться в Java-проектах, і підвищують читабельність та обслуговуваність коду. Lombok генерує цей код автоматично на етапі компіляції. Однак, важливо враховувати, що Lombok впливає на процес компіляції, тому розробники повинні коректно налаштовувати свої інструменти розробки..

3.3 Опис процесу програмної розробки

За допомогою бібліотеки TelegramBots створюється чат бот для інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.

Далі необхідно створити клас Bot, успадкувати його від класу TelegramLongPollingBot, перевизначивши його методи:

```
package com.example.bot;
import com.example.processor.UpdateProcessor;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
import org.telegram.telegrambots.bots.TelegramLongPollingBot;
import org.telegram.telegrambots.meta.api.objects.Update;

@Component
public class TelegramBot extends TelegramLongPollingBot {

    @Value("${telegram.bot.username}")
    private String username;
    @Value("${telegram.bot.token}")
    private String token;

    private UpdateProcessor updateProcessor;

    @Autowired
    public void setUpdateProcessor(UpdateProcessor updateProcessor) {
        this.updateProcessor = updateProcessor;
    }

    @Override
    public String getBotUsername() {
        return username;
    }

    @Override
    public String getBotToken() {
        return token;
    }

    @Override
    public void onUpdateReceived(Update update) {
        updateProcessor.process(update);
    }
}
```

Метод «getBotUsername()» повертає ім'я бота, вказане під час реєстрації, а метод «getBotToken()» повертає token бота для зв'язку з сервером Telegram.

Працює це все таким чином: при запуску програми, вона починає раз на п кількість секунд відправляти на сервер Telegram GET запит на наступний URL: <https://api.telegram.org/BotToken/getMe>, де BotToken – токен розробленого бота, отримуючи у відповідь JSON, де знаходяться всі повідомлення. Кожне таке повідомлення обробляється бібліотекою та надходить у метод «OnUpdateReceived(Update update)» об'єктом Update. Саме з ним і відбувається взаємодія.

Існує можливість надсилати боту повідомлення, використовуючи символ «/», якщо розробник передбачив можливість використання команд, наприклад з допомогою повідомлення «/start» можна розпочати діалог. Однак зручніше для цього використовувати кнопки. Є два види кнопок: ті, що з'являються під полем введення, «ReplyKeyboardMarkup» та кнопки, які знаходяться безпосередньо під повідомленням, до якого прив'язані, «InlineKeyboardMarkup».

При розробці інформаційної технології було використано «ReplyKeyboardMarkup» – шаблони повідомлень. Наведений нижче код демонструє створення клавіатури з кнопкою «Почати».

```
private void createAuthorizationWindow(Long chatId) {
    ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
    KeyboardRow row2 = new KeyboardRow();
    row2.add(KeyboardButton.builder()
        .text("Почати")
        .requestContact(true)
        .build());
    markup.setKeyboard(List.of(row2));
    markup.setResizeKeyboard(true);
    SendMessage sendMessage = SendMessage.builder()
        .chatId(String.valueOf(chatId))
        .replyMarkup(markup)
        .build();
    messageSender.sendMessage(sendMessage);
}
```

В даному класі є метод «.setResizeKeyboard()», що в становлює параметр «resizeKeyboard». Якщо «resizeKeyboard» встановлено в «true», клавіатура буде змінювати свій розмір в залежності від розміру екрану користувача. Якщо

«resizeKeyboard» встановлено в «false», клавіатура буде фіксованого розміру. Також є метод «.setKeyboard()», який встановлює рядки кнопок для клавіатури. Кожен «KeyboardRow» представляє собою рядок кнопок. Можна додавати кнопки до «KeyboardRow» за допомогою методу «add(KeyboardButton keyboardButton)».

Для отримання даних з об'єкта «Update» потрібно виконати «update.getCallbackQuery()», цей метод повертає «CallbackQuery», з якого можна отримати дані, передані боту. Якщо є повідомлення, слід відправити на обробку новий потік повідомлення, якщо є «CallbackQuery», відправляємо його на обробку у відповідний потік. На «CallbackQuery» можна надсилати відповідь. Кожен об'єкт у Telegram має свій id. Для відправки відповіді на певний «CallbackQuery» потрібно знати лише його id, який ми отримаємо з відповідного об'єкта.

Для анкетування працівників та перегляду результатів адміністратором необхідно реалізувати логіку авторизації. Для цього потрібно створити базу даних або інший механізм для збереження інформації про авторизованих користувачів, таку як ідентифікатори чатів або інші дані, які можна використовувати для перевірки авторизації. Наступний фрагмент реалізує перевірку користувачів на наявність в користувача прав адміністратора.

```
public class UserService {
    private final UserCache userCache;
    private static final List<String> ADMINS = List.of("Compass");
    public boolean isNewUser(Long chatId) {
        return isNull(userCache.getBy(chatId));
    }
    public void addUser() {
        userCache.add(User.builder().build());
    }
    public boolean isNotAuthUser(Long chatId) {
        return isNull(userCache.getBy(chatId).getName());
    }
    public void authorizeUser(Message message) {
        Contact contact = message.getContact();
        String name = getName(contact);
        User user = userCache.getBy(message.getChatId());
        user.setName(name);
        if (isAdmin(name)){
            user.setAdmin(true);
        }
    }
}
```

```

private boolean isAdmin(String name) {
    return ADMINS.contains(name);
}
private String getName(Contact contact) {
    StringBuilder sb = new StringBuilder();
    if (contact.getFirstName() != null
        && !contact.getFirstName().isEmpty()) {
        sb.append(contact.getFirstName());
    }
    if (contact.getLastName() != null
        && !contact.getLastName().isEmpty()) {
        sb.append(contact.getLastName());
    }
    return sb.toString();
}
public boolean isUserAdmin(Message message) {
    return userCache.getBy(message.getChatId()).isAdmin();
}
}

```

Де «isAdmin» – метод, який перевіряє, чи є користувач авторизованим, шляхом перевірки його ідентифікатора в базі даних.

3.4 Опис класів і функцій інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника

Метою розробки UML-діаграми класів інформаційної технології з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника є візуалізація класів, наведення взаємозв'язків між класами, опис методів та функцій кожного класу, визначення атрибутів та реалізація методів та логіки. На рисунку 3.1 зображено загальну UML-діаграму класів інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.

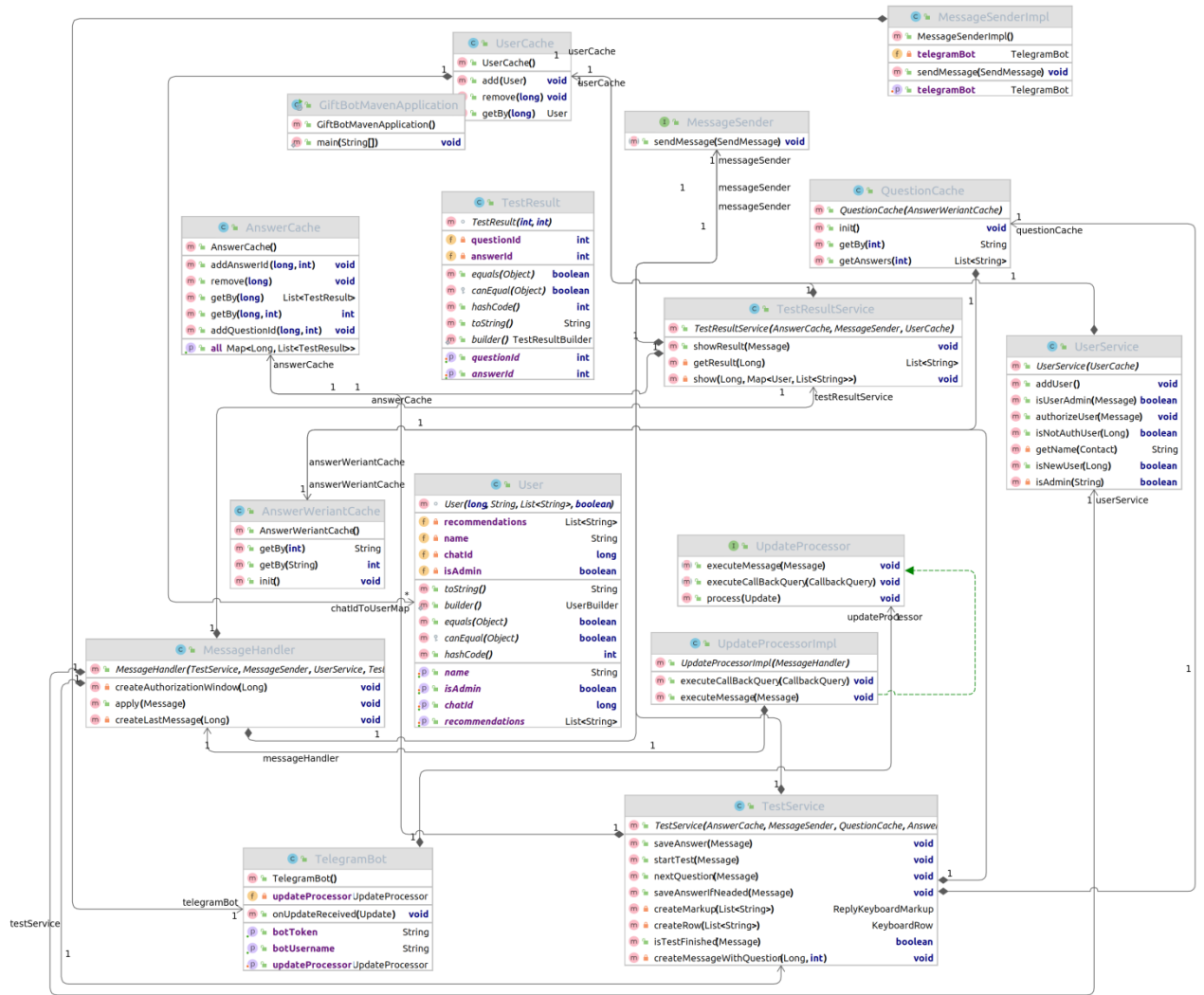


Рисунок 3.1 – Загальна UML-діаграма класів

Діаграма класів – це вид діаграми в мові моделювання UML, яка відображає структуру та взаємодію класів в системі програмного забезпечення. Діаграма класів визначає класи, атрибути, методи та взаємозв'язки між класами.

Розглянемо класи, розроблені для інформаційної технології з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника

Клас `UserCache` зберігає інформацію про користувачів, щоб уникнути частого доступу до бази даних.

Клас `TestResultBuilder` використовується для створення об'єктів результату тестування, а клас `TestResult` представляє результати тестування для конкретного користувача.

Клас AnswerCache зберігає варіанти відповідей для певних питань, а AnswerVariantCache зберігає варіанти відповідей для використання в тестуванні. QuestionCache зберігає питання для тестування.

Клас MessageHandler обробляє вхідні повідомлення від користувачів.

Клас GiftBotMavenApplication є основним класом додатка, що містить точку входу для запуску додатка.

UserBuilder використовується для створення об'єкта користувача, а UserService надає сервіси для роботи з об'єктами користувачів.

Отже, у рамках даного підрозділу було здійснено реалізацію інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника, розроблено загальну UML-діаграму класів та наведено огляд основних класів проекту.

3.5 Тестування та дослідження результатів роботи інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника

Тестування програмного забезпечення – це процес перевірки та підтвердження того, чи програмне забезпечення або програма не мають помилок, чи відповідає технічним вимогам у тому вигляді, в якому вони були спроектовані та розроблені. Процес тестування програмного забезпечення має на меті не тільки виявити помилки в існуючому програмному забезпеченні, але й знайти заходи щодо покращення програмного забезпечення з точки зору ефективності, точності та зручності використання.

Тестування програмного забезпечення можна розділити на два етапи:

Верифікація: відноситься до ряду завдань, які гарантують, що програмне забезпечення правильно реалізує певну функцію. Це означає: «Чи правильно ми створюємо продукт?».

Валідація: це серія різних завдань, які гарантують, що створене програмне забезпечення відповідає вимогам замовника. Це означає: «Чи створюємо ми правильний продукт?».

Тестування програмного забезпечення є важливим, оскільки будь-які помилки можна виявити на ранній стадії та виправити до того, як програмне забезпечення буде доставлено. Тестування здатне покращити якість програмного забезпечення: тестування програмного забезпечення виявляє помилки в програмному забезпеченні, а їх виправлення покращує якість програмного забезпечення.

Тестування програмного забезпечення забезпечує надійність, безпеку та високу продуктивність, що призводить до економії часу, коштів і задоволеності клієнтів. Допомагає з масштабованістю: тести типу нефункціонального тестування програмного забезпечення допомагають виявити проблеми з масштабованістю та момент, коли програма може припинити роботу. Після запуску програми буде дуже важко відстежувати та виправляти проблеми, оскільки виконання цієї дії коштуватиме більше часу. Тому під час розробки програмного забезпечення краще проводити тестування програмного забезпечення через регулярні проміжки часу.

Тестування чат-бота необхідно проводити за наступними кейсами:

- швидкість відклику на команду;
- правильна обробка повідомлень від користувача;
- коректне відображення діалогу;
- швидкість відображення відповіді після її написання користувачем;
- обробка натискання на клавішу відправки повідомлення;
- оцінка коректності відповідей.

Тестування програмного забезпечення становить ключовий етап у забезпеченні якості та надійності продукту для користувачів та клієнтів. Цей процес надає об'єктивну та незалежну інформацію щодо якості програмного забезпечення та його потенційних ризиків виникнення помилок.

Тести можуть бути виконані навіть на частково завершеному кодї, дозволяючи виявити та виправити проблеми на ранніх етапах розробки. Тим самим процес розробки часто визначає, коли і як проводиться тестування. Наприклад, у поетапному процесі, тестування зазвичай відбувається після визначення системних вимог та їхнього впровадження у програми тестування. У гнучкому програмному забезпеченні розробка та тестування можуть відбуватися одночасно відповідно до змінюваних вимог [34].

Розроблене програмне забезпечення було піддане інтенсивному тестуванню, що підтвердило його коректність та надійність. Було виконано 100 запусків додатку, піддано його різноманітним сценаріям використання, що надало змогу об'єктивно оцінити його ефективність.

Результати тестування програмного забезпечення наведено в таблиці 3.1. Ця таблиця включає характеристики, оцінювання яких надає можливість зробити висновок щодо відповідності програмного забезпечення вимогам.

Таблиця 3.1 – Результати тестування програмного забезпечення

Характеристика	Значення характеристики
Швидкодія	0,4 с
Час виконання складних запитів до бази даних	< 2с
Навантаження (запитів за хвилину)	100
Використання пам'яті	1024 мб

За результатами тестування, наведеними в таблиці 3.1 можна зробити висновок, що бот забезпечує швидку реакцію на запити користувачів. Вимірювання часу виконання складних запитів до бази даних показало, що середній час виконання запиту не перевищує 2 секунди, що забезпечує ефективну роботи з базою даних під час великого обсягу інформації. Бот стійко витримує 100 запитів за хвилину без втрати продуктивності, що означає здатність програмного забезпечення працювати при великому обсязі даних. Використання ресурсів

знаходиться на прийнятному рівні, не перевищує встановлені ліміти, що означає оптимальне використання ресурсів пристрою.

Після того, як користувач запустить телеграм-бота, перший крок – розпочати взаємодію з ним. Бот, який отримав назву Gift Helper, очікує на повідомлення користувача. Для початку спілкування достатньо натискати кнопку «Start», яку вам запропонує бот при переході до нього в Telegram (рисунок 3.2).



Рисунок 3.2 – Вкладка з початком роботи з ботом

Після цього бот висилає повідомлення «Почнемо після авторизації». Тут впроваджена система авторизації, яка визначає права адміністратора. Якщо користувач не має прав адміністратора, йому одразу запропонується пройти анкетування, яке ініціює бот (рисунок 3.3). Учаснику надаються варіанти відповідей у вигляді кнопок з альтернативами.

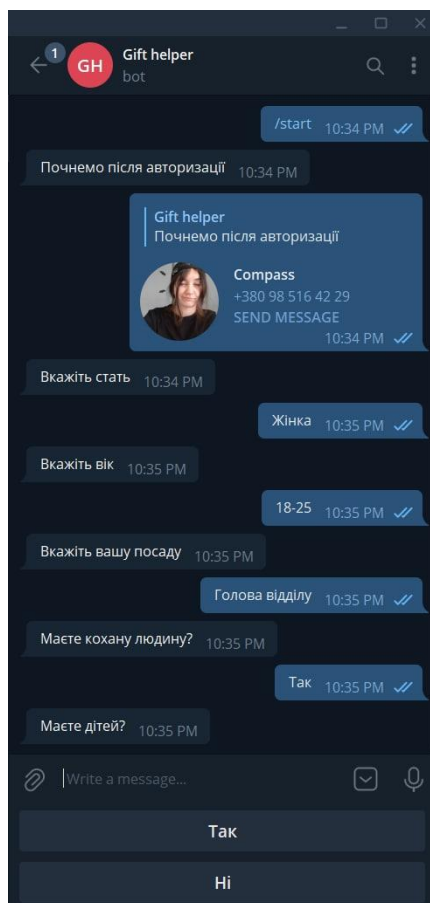


Рисунок 3.3 – Вкладка з початком анкетування працівника

Анкетування виглядає як тест з основними та додатковими питаннями, спеціально розробленими для отримання повної інформації про вподобання працівника. Ці дані важливі для прийняття рішення щодо того, наскільки вибір подарунка відповідає особистим уподобанням працівника (рисунок 3.4).

Після того, як бот отримає всі відповіді на поставлені питання, «Gift helper» висилає повідомлення «Супер, дуже дякуємо за відповіді!». Після цього можна вважати, що діалог із ботом завершено. Бот обробляє відповіді працівника та формує результат, який може бути виведений за запитом адміністратора.

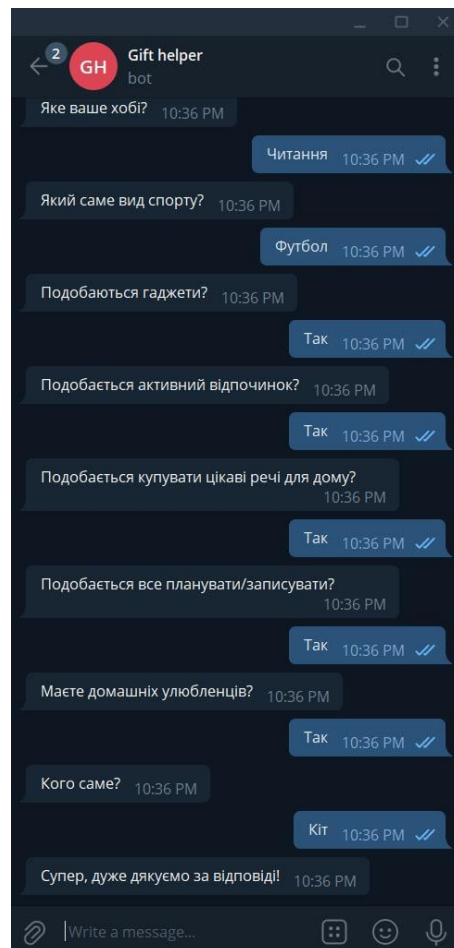


Рисунок 3.4 – Вкладка з продовженням опитування працівника

У випадку, якщо при перевірці прав користувача виявлено, що він має адміністраторські права, він отримує доступ до перегляду відповідей інших користувачів. Лише адміністратор може ініціювати процес підбору подарунка для конкретного працівника, а результат виводиться тільки для нього. Інформація про подарунки відображається через повідомлення, відправлене ботом, і містить список подарунків, впорядкованих у порядку спадання відповідності вподобань працівника. Тобто найбільш відповідний подарунок з'являється на початку списку (рисунок 3.5).

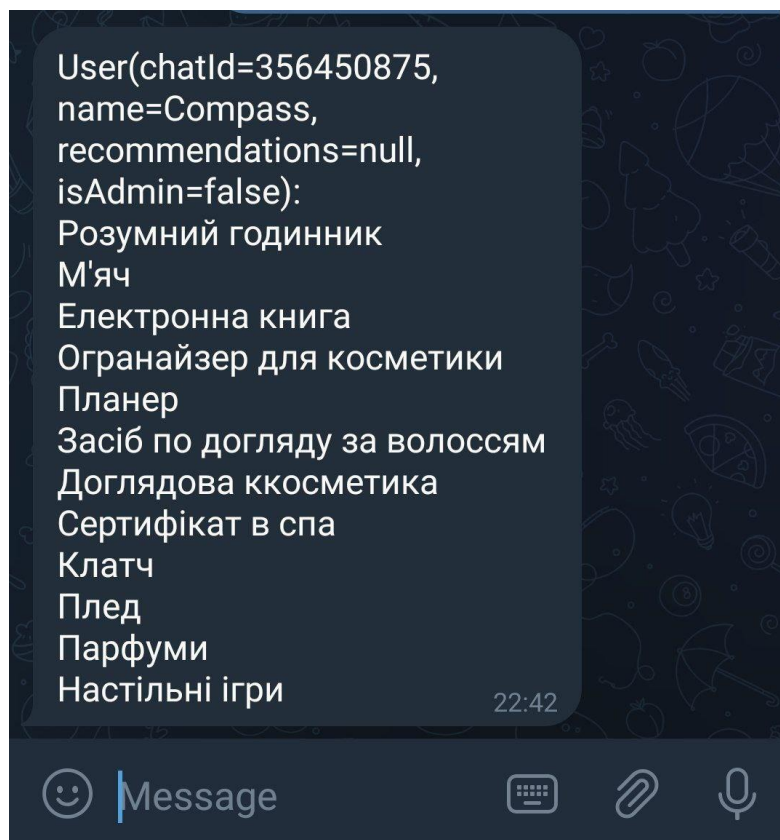


Рисунок 3.5 – Вкладка з результатом надання рекомендацій

Після проведення тестування можна зазначити, що програма відповідає вимогам та демонструє адекватну реакцію на відповіді користувача. Результати свідчать про коректну роботу додатку, яка підтверджується під час 100 запусків, в ході яких було оцінено його функціональні можливості.

У порівнянні з аналогами, які мають обмежену точність підбору і обмежену кількість запитань, розроблений бот «Gift helper» вирізняється збільшеною кількістю питань, покликаною покращити точність підбору. Таким чином, порівняно із «Знайти подарунок» та «NU SHO», кількість питань у «Gift helper» майже вдвічі більша.

Аналог «Findodo» з більш ніж 8 питань має обмеження, орієнтоване на американських користувачів та партнерство з американськими магазинами. З іншого боку, «Gift helper» спрямований на українських працівників та їхніх колег, що усуває недоліки «Findodo».

Додатково, ефективність прийняття рішень щодо відповідності вибору подарунка до вподобань працівника була оцінена через опитування 100 респондентів. Кожен учасник пройшов анкетування за допомогою розробленого додатку та аналогу «Findodo», отримавши рекомендацію щодо найбільш підходящого подарунку. Респондентам було запропоновано відповісти на запитання: «Чи хотіли б ви отримати такий подарунок?» та виставити оцінку від 1 до 10 щодо відповідності вибору їхнім вподобанням.. Результати тестування записані в таблиці 3.2.

Таблиця 3.2 – Оцінювання результатів респондентами

Оцінка відповідності вибору подарунка	Кількість респондентів «Gift helper»	Кількість респондентів аналогу «Findodo»
1	2	2
2	2	2
3	5	6
4	5	6
5	1	5
6	14	14
7	7	18
8	14	15
9	27	20
10	23	12

Відповідно до інформації, наведеній в таблиці 3.2 можна вирахувати точність роботи програмного забезпечення для прийняття рішень щодо відповідності вибору подарунка до вподобань працівника та аналогічного рішення «Findodo». Підрахувавши оцінки для кожного додатку визначимо, що сумарна кількість балів у розробленого додатку 764, натомість в аналогу 703. Отже, розроблений додаток має точність 76,4%, а аналог – 70,3%, що означає покращення точності надання рекомендації щодо вибору подарунку на 6,1%.

3.6 Висновок до розділу 3

В даному розділі було проведено реалізацію інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника мовою програмування Java, що є потужною та популярною мовою програмування, яка має багатий набір бібліотек і фреймворків, що допомагають в розробці чат-ботів та інтерактивних додатків, завдяки її високій продуктивності, багатопоточності, збільшенню контролю над ресурсами, безпеці, багатofункціональності та інтеграції з іншими технологіями. Розробка проведена у середовищі IntelliJ IDEA, яке характеризується чітким і лаконічним інтерфейсом та інтеграцією з автоматизованими системами складання та наявністю інструментів для тестових програм.

Обґрунтовано вибір бібліотек та фреймворків, за допомогою яких проводилась реалізація інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.

Було розроблено UML-діаграму класів інформаційної технології, наведено огляд та приклад використання бібліотек та фреймворків, а також програмно реалізовано чат-бот під назвою «Gift helper».

Тестування ефективності надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника відбувалось за допомогою опитування групи респондентів, які оцінювали ефективність підбору подарунку з допомогою розробленого програмного забезпечення. Порівняно з аналогом точність надання рекомендації зросла на 6,1%, що підтвердило коректність роботи додатку.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту залучаємо 3-х незалежних експертів, якими є провідні викладачі випускової або спорідненої кафедри.

Результати оцінювання науково-технічного рівня розробки зводимо до таблиці 1.

Таблиця 1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу засобу поляриметричного аналізу оптично активних рідни

Критерії	Експерти		
	Барабан С.В	Озеранський В.С	Арсенюк І.Р
	Бали, виставлені експертами		
Технічна здійсненність концепції	2	2	2
Ринкові переваги (наявність аналогів)	3	3	3
Ринкові переваги (ціна продукту)	4	2	3
Ринкові переваги (технічні властивості)	3	2	2
Ринкові переваги (експлуатаційні витрати)	2	2	2
Ринкові перспективи (розмір ринку)	3	3	2
Ринкові перспективи (конкуренція)	2	2	3
Практична здійсненність (наявність фахівців)	2	3	2

Продовження таблиці 4.1

Практична здійсненність (наявність фінансів)	3	3	3
Практична здійсненність (необхідність нових матеріалів)	3	3	2
Практична здійсненність (термін реалізації)	3	3	2
Практична здійсненність (розробка документів)	2	2	2
Сума балів	32	30	28
Середньоарифметична сума балів, СБ	30		

За результатами розрахунків, наведених в таблиці 1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника – середній.

4.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати на оплату праці. Належать витрати на виплату основної та додаткової заробітної плати керівникам, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Основна заробітна плата дослідників. Витрати на основну заробітну плату дослідників (Z_0) розраховують відповідно до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p},$$

де k – кількість посад дослідників, залучених до процесу дослідження; M_{ni} – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.; T_p – число робочих днів в місяці; приблизно $T_p = (21 \dots 23)$ дні; t_i – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 2.

Таблиця 2 – Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	15000	714	20	14280
Розробник	30000	1428	50	71400
Всього:				85680

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт розраховують за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i,$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год; t_i – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_C}{T_p \cdot t_{3M}},$$

де M_m – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), у 2023 році $M_m=6700$ грн; K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду; K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати; T_p – середня кількість робочих днів в місяці, приблизно $T_p = 21 \dots 23$ дні; $t_{зм}$ – тривалість зміни, год.

Таблиця 3 – Витрати на заробітну плату робітників

Найменування робіт	Трудовісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Дослідження предметної області	120	5	54,2	1,36	6504
Розробка архітектурних рішень та постановка задач	80	6	57,8	1,45	4624
Розробка програмного забезпечення	280	6	57,8	1,45	16184
Тестування програмного забезпечення	80	3	47,1	1,18	3768
Всього					31080

Додаткова заробітна плата. Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = 0,1 \cdot (Z_o + Z_p) = 0,1 \cdot (85680 + 31080) = 11676 \text{ грн.}$$

Відрахування на соціальні заходи. Нарахування на заробітну плату $H_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$\begin{aligned} H_{зп} &= \beta \cdot (З_о + З_р + З_д) = \\ &= 0,22 \cdot (85680 + 31080 + 11676) = 28256 \text{ грн.} \end{aligned}$$

де $З_о$ – основна заробітна плата розробників, грн.; $З_р$ – основна заробітна плата робітників, грн.; $З_д$ – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Програмне забезпечення. До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10...12% від вартості програмного забезпечення. Балансову вартість програмного забезпечення розраховують за формулою:

$$V_{\text{прг}} = \sum_1^k C_{\text{іпрг}} \cdot C_{\text{прг.і}} \cdot K_i,$$

де $C_{\text{іпрг}}$ – ціна придбання програмного забезпечення i -го виду, грн.; $C_{\text{прг.і}}$ – кількість одиниць програмного забезпечення відповідного виду, шт.; K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного забезпечення, $K_i = (1,1 \dots 1,12)$; k – кількість видів програмного забезпечення.

Таблиця 4 – Витрати на придбання програмного забезпечення

Найменування програмного забезпечення	Ціна за одиницю, грн.	Витрачено	Вартість програмного забезпечення, грн.
Intellij Idea	6700	1	6700
MySQL Workbench	9800	1	9800

Продовження таблиці 4.4

Всього, з врахуванням коефіцієнта інсталяції та налагодження	18150
--	-------

Амортизація обладнання. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування A в цілому бути розраховані за формулою:

$$A = \frac{Ц_6}{T_в} \cdot \frac{t}{12'}$$

де $Ц_6$ – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн.; t – термін використання основного фонду, місяці; $T_в$ – термін корисного використання основного фонду, роки.

Таблиця 5 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук HP Pavilion 15-eh1108ua	24999	5	2	833
Монітор Samsung Curved	5400	5	2	180
Всього	1013			

Витрати на електроенергію для науково-виробничих цілей. Витрати на силову електроенергію V_e , якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

Таблиця 6 – Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Ноутбук HP Pavilion 15-eh1108ua	0.075	500
Монітор Samsung Curved	0.03	300
Освітлення	0.04	200

$$\begin{aligned}
 B_e &= \sum \frac{W_i \cdot t_i \cdot C_e \cdot K_{впі}}{ККД} \\
 &= \frac{0,075 \cdot 500 \cdot 7,5 \cdot 0,95}{0,98} + \frac{0,03 \cdot 300 \cdot 7,5 \cdot 0,95}{0,98} \\
 &\quad + \frac{0,04 \cdot 200 \cdot 7,5 \cdot 0,95}{0,98} = 396 \text{ грн.},
 \end{aligned}$$

W_i – встановлена потужність обладнання, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год.; C_e – вартість 1 кВт електроенергії, грн.; $K_{впі}$ – коефіцієнт використання потужності; ККД – коефіцієнт корисної дії обладнання.

Інші витрати. До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_B = (Z_o + Z_p) \cdot \frac{H_{iB}}{100\%} = (85680 + 31080) \cdot \frac{87}{100} = 101581 \text{ грн.},$$

де H_{iB} – норма нарахування за статтею «Інші витрати».

Накладні (загальнопромислові) витрати. До статті «Накладні (загальнопромислові) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку

(перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$V_{\text{нзв}} = (Z_o + Z_p) \cdot \frac{N_{\text{нзв}}}{100\%} = (85680 + 31080) \cdot \frac{147}{100} = 171637 \text{ грн.},$$

де $N_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення науково-дослідної роботи. Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$V_{\text{заг}} = Z_o + Z_p + Z_{\text{дод}} + Z_n + V_{\text{прг}} + A_{\text{обл}} + V_e + \\ + I_v + V_{\text{нзв}} = 85680 + 31080 + 11676 + 28256 + 18150 + 1013 + 396 + \\ 101581 + 171637 = 439469 \text{ грн.}$$

Загальні витрати. Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{V_{\text{заг}}}{\eta} = \frac{439469}{0,9} = 488299 \text{ грн.},$$

де η – коефіцієнт, що характеризує етап виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на стадії впровадження, то $\eta=0,9$.

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

В даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати: ΔN – збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу; N – кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки; C_0 – вартість послуги у році до впровадження інформаційної системи; $\pm\Delta C_0$ – зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta\Pi$ для кожного із років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta\Pi = (\pm\Delta C_0 \cdot N + C_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right),$$

де $\pm\Delta C$ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки); $\pm\Delta C_0$ може мати як додатне, так і від'ємне значення (від'ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни); N – основний кількісний показник,

який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки; C_0 – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році; C_6 – основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів; ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки); λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,8333$; ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho = 0,2 \dots 0,5$; ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$.

Очікуваний термін життєвого циклу розробки 1 рік, тому:

$$\begin{aligned} \Delta\P &= ((1800 - 1200) \cdot 5000 - (5000 - 5000) \cdot 1200) \cdot 0,8333 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) \\ &= 633600 \text{ грн.} \end{aligned}$$

Далі розраховують приведену вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\P_i}{(1 + \tau)^t} = \frac{633600}{(1 + 0,1)^1} = 576000 \text{ грн.,}$$

де $\Delta\P$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.; T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо $T=1$ рік); τ – ставка дисконтування, за яку можна взяти щорічний прогнозований

рівень інфляції в країні, $\tau = 0,05 \dots 0,15$; t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{інв}} \cdot ЗВ = 1 \cdot 488299 = 488299 \text{ грн.}$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}} = 1 \dots 5$, але може бути і більшим; $ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV = 576000 - 488299 = 87701 \text{ грн.,}$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.; PV – теперішня вартість початкових інвестицій, грн.

Оскільки $E_{\text{абс}} > 0$, то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність E_v або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність,

нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій E_B , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою:

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} = \sqrt{1 + \frac{87701}{488299}} = 1,08,$$

де $T_{ж}$ – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування τ_{\min} , тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій τ_{\min} визначається за формулою:

$$\tau_{\min} = d + f = 0,9 + 0,05 = 0,95,$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,9 \dots 0,12$; f – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина $f = 0,05 \dots 0,5$, але може бути і значно вищою.

Оскільки $E_B = 1,08 > \tau_{\min} = 0,95$, то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

Далі розраховуємо період окупності інвестицій T_o , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_o = \frac{1}{E_B} = \frac{1}{1,08} = 0,93 \text{ року.}$$

Оскільки $T_o=0,93 < 1 \dots 3$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

4.4 Висновок до розділу 4

В четвертому розділі було проведено оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, результати представлені в таблиці 1. Також було здійснено розрахунок витрат на здійснення науково-дослідної роботи, та розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором. В результаті проведених розрахунків та досліджень можна сказати, що науково-технічна розробка, виконана в межах МКР має комерційну привабливість для інвесторів та клієнтів.

ВИСНОВКИ

Під час виконання магістерської кваліфікаційної роботи було реалізовано інформаційну технологію для прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.

Обґрунтовано доцільність розробки інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника. Було проведено аналіз сучасних програм-аналогів, які використовуються для підбору подарунків та наведено короткий опис основних функцій, які виконують дані програми. Виявлено недоліки, що характеризуються недостатньою точністю у виборі подарунків та обмеженою кількістю запитань про отримувача.

Було розроблено структуру інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника та спроектовано схему алгоритму роботи розроблюваної програми.

Також було досліджено методи, що можуть бути використані для надання рекомендацій щодо вибору подарунку відповідно до вподобань працівника, для чого запропоновано використати комбінацію методів на основі нечіткого дерева рішень та методу аналізу ієрархій. На основі цих методів запропоновано математичну модель надання рекомендацій щодо вибору подарунку відповідно до вподобань працівника, що дозволяє ефективно підібрати подарунок.

Здійснено програмну реалізацію інформаційної технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника. Обґрунтовано вибір мови програмування. Розробка інформаційної технології велася на об'єктно-орієнтованій мові програмування Java в середовищі розробки IntelliJ IDEA. Розглянуто бібліотеки та фреймворки, які доцільно використовувати в інформаційній технології надання рекомендацій з прийняття рішень щодо відповідності вибору подарунка до вподобань працівника.

Також було проведено оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності. Було здійснено розрахунок витрат на здійснення науково-дослідної роботи, та розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором. В результаті проведених розрахунків та досліджень можна сказати, що науково-технічна розробка, виконана в межах МКР має комерційну привабливість для інвесторів та клієнтів, оскільки період окупності інвестицій $T_o = 0.93$ року

Тестування ефективності надання рекомендацій з прийняття рішень щодо відповідності вибору подарунку до вподобань працівника показало підвищення точності надання рекомендації на 6,1%, що означає доведення поставленої мети.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Панасюк Н. А., Белзецький Р. С. Програмне забезпечення для прийняття рішень щодо відповідності вибору подарунка до вподобань працівника. LI Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/15717/13184> (дата звернення: 23.09.2023).
2. Панасюк Н. А. Інформаційна технологія для прийняття рішень щодо відповідності вибору подарунка до вподобань працівника. LII Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/schedConf/presentations> (дата звернення: 23.09.2023).
3. Які поради варто використовувати при виборі подарунків? URL: https://lb.ua/tech/2021/09/06/493186_yaki_poradi_varto_vikoristovuvati.html (дата звернення: 23.09.2023).
4. Подарунок – гарний спосіб висловити вдячність співробітникам, клієнтам чи партнерам URL: <https://obukhiv.info/news/podarunok-garnii-sposib-visloviti-vdyachnist-spivrobotnikam-klientam-chi-partneram/> (дата звернення: 23.09.2023).
5. Литвак Р. Б. Вибір алгоритмічного забезпечення підтримки процесу сегментації користувачів системи та позиціонування товарів на прикладі книжкового ринку // Інформатика та обчислювальна техніка-IOT-2018 : матеріали наук.-практ. конф. Київ, 2018. С. 186-191.
6. Як бізнес може використовувати штучний інтелект URL: <https://www.epravda.com.ua/columns/2023/05/8/699875/> (дата звернення: 23.09.2023).
7. Дудченко, Н. В. (2019). Рекомендаційні системи для відеоконтенту: аналіз та вибір підходу. Вісник Київського національного університету імені Тараса Шевченка, (32), 30-35.

8. Системи і методи підтримки прийняття рішень: навч. посіб./ Бідюк П. І., Тимощук О. Л., Коваленко А. Є., Коршевніук Л. О. Київ : КПІ ім. Ігоря Сікорського, 2022. 610 с.
9. Знахуренко В.П. Система надання рекомендацій щодо вибору подарунків. Сучасні інформаційні технології : збірник тез та доповідей IV Всеукраїнської заочної науково-практичної конференції. Київ, 2013. С 115.
10. Волошин, О. Ф., Мащенко С. О. Моделі та методи прийняття рішень : навч. посіб. для студ. вищ. навч. закл. 2-ге вид., перероб. і доп. Київ : Видавничополіграфічний центр "Київський університет", 2010. 336 с.
11. Cendrowska J. PRISM: An algorithm for inducing modular rules. International Journal of ManMachine Studies 1987. Vol. 27, № 4. P. 349-370.
12. Umano M., Okamoto H., Hatono I. Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. IEEE World Congress on Computational Intelligence. Proceedings of the Third IEEE Conference, 1994 Vol.3. P. 2113 - 2118.
13. Олійник А.О. Побудова асоціативних правил на основі інтелектуального стохастичного пошуку // Математичні машини і системи. Київ, 2015. № 4. С. 45-56.
14. GiftAdvisor URL: <https://www.giftadvisor.com/> (дата звернення: 01.10.2023).
15. Gift Wizard URL: <https://www.giftwizard.ai/> (дата звернення: 01.10.2023).
16. Amazon Gift Finder URL: <https://www.amazon.co.uk/gcx/Gifts-for-everyone/gfhz/> (дата звернення: 01.10.2023).
17. Etsy Gift Finder URL: https://www.etsy.com/market/gift_finder (дата звернення: 01.10.2023).
18. Знайти подарунок. URL: <https://play.google.com/store/apps/details?id=by.idei.jarkie.giftgenerator3> (дата звернення: 01.10.2023).
19. Додаток для вибору подарунків Giftenizer. URL: <https://giftenizer.com/> (дата звернення: 01.10.2023).

20. Додаток для вибору подарунків NU SHO. URL: <https://nusho.com.ua/> (дата звернення: 01.10.2023).
21. Додаток для вибору подарунків Findodo. URL: <https://www.findodo.com/> (дата звернення: 01.10.2023).
22. Стартап GiftHub URL: <https://pingvin.pro/gadgets/news-gadgets/gifthub-ukrayinska-komanda-stvoryla-servis-z-pidboru-podarunkiv.html> (дата звернення: 01.10.2023).
23. Циганков О. В. Основи проектування систем штучного інтелекту : конспект лекцій. Одеса : ОНПУ, 2006. 195 с.
24. Сорока П. М., Іларіонов О. Є. Вирішення проблем прийняття рішень в умовах невизначеності за допомогою дерева рішень // Вісник національного технічного університету ХПІ. Механіко-технологічні системи та комплекси. Харків, 2016. № 50 (1222). С. 106-111.
25. Сокур Д. В. Нечітке дерево розв'язків як метод автоматичного аналізу даних // Наукові розробки молоді на сучасному етапі: тези доповідей XVII Всеукраїнської наукової конференції молодих вчених та студентів (м. Київ, 26-27 квітня 2018 р.) / відпов. за випуск Блохін О. Л. Київ, 2018. С. 493-494.
26. Добролюбова М. В. Програмування баз даних: конспект лекцій. Київ; КПІ ім. Ігоря Сікорського, 2021. 275 с.
27. Погромська Г. С. Побудова запитів на мові SQL: навч. посіб. Миколаїв: МНУ ім. В. О. Сухомлинського, 2014. 133 с.
28. Кадомський К. К., Ніколюк П. К. Java. Теорія і практика: навч. посіб. Вінниця, ДОННУ, 2019. 197 с.
29. Мокін Б. І., Мокін В. Б., Мокін О. Б. Навчальний посібник для опанування студентами способів розв'язання задач з функціонального аналізу мовою Python, частина 1 : навч. посіб. Вінниця : ВНТУ, 2022. 124 с.
30. IntelliJ Idea – середовище розробки для Java. URL: <https://ualinux.com/uk/ubuntu-apps-programming/idea-ic> (дата звернення: 23.05.2022).
31. Spring Framework. URL: <https://spring.io/projects/spring-framework> (дата звернення: 23.05.2022).

32. Spring Boot. URL: <https://spring.io/projects/spring-boot> (дата звернення: 23.05.2022).
33. Project Lombok. URL: <https://projectlombok.org/> (дата звернення: 23.05.2022).
34. Авраменко А. С., Авраменко В. С., Костенюк Г. В. Тестування програмного забезпечення : навч. посіб. Черкаси : ЧНУ ім. Богдана Хмельницького, 2017. 284 с.

Додаток А (обов'язковий)
Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень

Назва роботи: Інформаційна технологія для прийняття рішень щодо відповідності вибору подарунка до вподобань працівника

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФПТА
(кафедра, факультет)

Показники звіту подібності Unicheck


Оригінальність 93,42% Схожість 6,58%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.


Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи



Панасюк Н.А.

Керівник роботи



Барабан С.В.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

Клас MessageHandler

```

package com.example.bot;

import com.example.messagesender.MessageSender;
import com.example.service.TestService;
import com.example.service.TestResultService;
import com.example.service.UserService;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Component;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Message;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.ReplyKeyboardMarkup;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.buttons.KeyboardButton;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.buttons.KeyboardRow;

import java.util.List;

@Component
@RequiredArgsConstructor
public class MessageHandler {

    private final TestService testService;
    private final MessageSender messageSender;
    private final UserService userService;
    private final TestResultService testResultService;

    public void apply(Message message) {
        if (userService.isNewUser(message.getChatId())){
            userService.addUser();
            createAuthorizationWindow(message.getChatId());
            return;
        }
        if (userService.isNotAuthUser(message.getChatId())){
            userService.authorizeUser(message);
            if (userService.isUserAdmin(message)){
                testResultService.showResult(message);
                // testResultService.cleanCash();
            }
            return;
        }
        testService.startTest(message);
        return;
    }
    if (testService.isTestFinished(message)){
        testService.saveAnswerIfNeaded(message);
        createLastMessage(message.getChatId());
        return;
    }
}

```

```

        testService.saveAnswer(message);
        testService.nextQuestion(message);
    }

    private void createLastMessage(Long chatId) {
        SendMessage sendMessage = SendMessage.builder()
            .text("Супер, дуже дякуємо за відповіді!")
            .chatId(String.valueOf(chatId))
            .build();
        messageSender.sendMessage(sendMessage);
    }

    private void createAuthorizationWindow(Long chatId) {
        ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
        KeyboardRow row2 = new KeyboardRow();
        row2.add(KeyboardButton.builder()
            .text("Почати")
            .requestContact(true)
            .build());
        markup.setKeyboard(List.of(row2));
        markup.setResizeKeyboard(true);
        SendMessage sendMessage = SendMessage.builder()
            .chatId(String.valueOf(chatId))
            .replyMarkup(markup)
            .build();
        messageSender.sendMessage(sendMessage);
    }
}

```

Клас TelegramBot

```

package com.example.bot;

import com.example.processor.UpdateProcessor;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
import org.telegram.telegrambots.bots.TelegramLongPollingBot;
import org.telegram.telegrambots.meta.api.objects.Update;

@Component
public class TelegramBot extends TelegramLongPollingBot {

    @Value("${telegram.bot.username}")
    private String username;
    @Value("${telegram.bot.token}")
    private String token;

    private UpdateProcessor updateProcessor;

    @Autowired
    public void setUpdateProcessor(UpdateProcessor updateProcessor) {

```

```

    this.updateProcessor = updateProcessor;
}

@Override
public String getBotUsername() {
    return username;
}

@Override
public String getBotToken() {
    return token;
}

@Override
public void onUpdateReceived(Update update) {
    updateProcessor.process(update);
}
}

```

Класс TestService

```

package com.example.service;

import com.example.cache.AnswerCache;
import com.example.cache.AnswerWeriantCache;
import com.example.cache.QuestionCache;
import com.example.entity.TestResult;
import com.example.messagesender.MessageSender;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Message;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.ReplyKeyboardMarkup;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.buttons.KeyboardRow;

import java.util.List;

@Service
@RequiredArgsConstructor
public class TestService {

    private final AnswerCache answerCache;
    private final MessageSender messageSender;
    private final QuestionCache questionCache;
    private final AnswerWeriantCache answerWeriantCache;

    public void startTest(Message message) {
        questionCache.init();
        answerCache.addQuestionId(message.getChatId(), 1);
        createMessageWithQuestion(message.getChatId(), 1);
    }
}

```

```

public void saveAnswer(Message message) {
    String text = message.getText();
    int answerId = answerWeriantCache.getBy(text);
    answerCache.addAnswerId(message.getChatId(), answerId);
}

public void saveAnswerIfNeaded(Message message) {
    String text = message.getText();
    List<TestResult> testResults = answerCache.getBy(message.getChatId());
    TestResult lastAnswer = testResults.get(testResults.size());
    if (lastAnswer.getAnswerId() == 0) {
        int answerId = answerWeriantCache.getBy(text);
        answerCache.addAnswerId(message.getChatId(), answerId);
    }
}

public boolean isTestFinished(Message message) {
    List<TestResult> testResults = answerCache.getBy(message.getChatId());
    TestResult lastAnswer = testResults.get(testResults.size());
    if (lastAnswer.getQuestionId() == 14) {
        return true;
    }
    if (lastAnswer.getQuestionId() == 13 && lastAnswer.getAnswerId() == 12) {
        return true;
    }
    return false;
}

private void createMessageWithQuestion(Long chatId, int questionId) {
    String questionText = questionCache.getBy(questionId);
    List<String> answers = questionCache.getAnswers(questionId);

    ReplyKeyboardMarkup markup = createMarkup(answers);

    SendMessage sendMessage = SendMessage.builder()
        .text(questionText)
        .chatId(String.valueOf(chatId))
        .replyMarkup(markup)
        .build();
    messageSender.sendMessage(sendMessage);
}

private ReplyKeyboardMarkup createMarkup(List<String> answers) {
    ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
    if (answers.size() == 2) {
        KeyboardRow row = createRow(answers.subList(0, 1));
        KeyboardRow row2 = createRow(answers.subList(1, 2));
        markup.setKeyboard(List.of(row, row2));
    }
    if (answers.size() == 3) {
        KeyboardRow row = createRow(answers.subList(0, 1));
        KeyboardRow row2 = createRow(answers.subList(1, 2));
    }
}

```

```

        KeyboardRow row3 = createRow(answers.subList(2, 3));
        markup.setKeyboard(List.of(row, row2, row3));
    }
    if (answers.size() == 4) {
        KeyboardRow row = createRow(answers.subList(0, 2));
        KeyboardRow row2 = createRow(answers.subList(2, 4));
        markup.setKeyboard(List.of(row, row2));
    }
    if (answers.size() == 5) {
        KeyboardRow row = createRow(answers.subList(0, 2));
        KeyboardRow row2 = createRow(answers.subList(2, 4));
        KeyboardRow row3 = createRow(answers.subList(4, 5));
        markup.setKeyboard(List.of(row, row2, row3));
    }
    if (answers.size() == 6) {
        KeyboardRow row = createRow(answers.subList(0, 2));
        KeyboardRow row2 = createRow(answers.subList(2, 4));
        KeyboardRow row3 = createRow(answers.subList(4, 6));
        markup.setKeyboard(List.of(row, row2, row3));
    }

    markup.setResizeKeyboard(true);
    return markup;
}

private KeyboardRow createRow(List<String> answers) {
    KeyboardRow row = new KeyboardRow();
    // List<KeyboardButton> keyboardButtons = answers.stream()
    //     .map(s -> KeyboardButton.builder()
    //         .text(s)
    //         .build())
    //     .collect(Collectors.toList());
    row.addAll(answers);
    return row;
}

public void nextQuestion(Message message) {
    long chatId = message.getChatId();
    List<TestResult> testResults = answerCache.getBy(message.getChatId());
    TestResult lastAnswer = testResults.get(testResults.size());
    int lastQId = lastAnswer.getQuestionId();
    if (lastQId == 5 && lastAnswer.getAnswerId() == 13) {
        answerCache.addQuestionId(message.getChatId(), 6);
        createMessageWithQuestion(chatId, 6);
        return;
    }
    if (lastQId < 13) {
        answerCache.addQuestionId(message.getChatId(), lastQId + 1);
        createMessageWithQuestion(chatId, lastQId + 1);
        return;
    }
}

```

```

    if (lastQId == 13 && lastAnswer.getAnswerId() == 11) {
        answerCache.addQuestionId(message.getChatId(), 14);
        createMessageWithQuestion(chatId, 14);
        return;
    }
}
}
}

```

Класс UserService

```

package com.example.service;

import com.example.cache.UserCache;
import com.example.entity.User;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.telegram.telegrambots.meta.api.objects.Contact;
import org.telegram.telegrambots.meta.api.objects.Message;

import java.util.List;

import static java.util.Objects.isNull;

@Service
@RequiredArgsConstructor
public class UserService {

    private final UserCache userCache;

    private static final List<String> ADMINS = List.of("Alex");

    public boolean isNewUser(Long chatId) {
        return isNull(userCache.getBy(chatId));
    }

    public void addUser() {
        userCache.add(User.builder().build());
    }

    public boolean isNotAuthUser(Long chatId) {
        return isNull(userCache.getBy(chatId).getName());
    }

    public void authorizeUser(Message message) {
        Contact contact = message.getContact();
        String name = getName(contact);
        User user = userCache.getBy(message.getChatId());
        user.setName(name);
        if (isAdmin(name)){
            user.setAdmin(true);
        }
    }
}

```



```

    }

    private boolean isAdmin(String name) {
        return ADMINS.contains(name);
    }

    private String getName(Contact contact) {
        StringBuilder sb = new StringBuilder();
        if (contact.getFirstName() != null
            && !contact.getFirstName().isEmpty()) {
            sb.append(contact.getFirstName());
        }
        if (contact.getLastName() != null
            && !contact.getLastName().isEmpty()) {
            sb.append(contact.getLastName());
        }
        return sb.toString();
    }

    public boolean isUserAdmin(Message message) {
        return userCache.getBy(message.getChatId()).isAdmin();
    }
}

```

Клас TestResultService

```

package com.example.service;

import com.example.cache.AnswerCache;
import com.example.cache.UserCache;
import com.example.entity.TestResult;
import com.example.entity.User;
import com.example.messagesender.MessageSender;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Message;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import java.util.stream.Collectors;

@Service
@RequiredArgsConstructor
public class TestResultService {

    private final AnswerCache answerCache;
    private final MessageSender messageSender;
    private final UserCache userCache;

    private final String LINE_SEPARATOR = System.lineSeparator();

```

```

public void showResult(Message message) {
    Map<Long, List<TestResult>> all = answerCache.getAll();
    Long chatId = message.getChatId();
    Map<User, List<String>> collect = all.keySet()
        .stream()
        .collect(Collectors.toMap(id -> userCache.getBy(id),
            id -> getResult(chatId)));

    show(chatId, collect);
}

private List<String> getResult(Long chatId) {

    List<String> gifts = new ArrayList<>();
    if (answerCache.getBy(chatId, 9) == 11 && (answerCache.getBy(chatId, 2) == 3 ||
answerCache.getBy(chatId, 2) == 4)) {
        gifts.add("Розумний годинник");
    }
    if (answerCache.getBy(chatId, 9) == 11 && (answerCache.getBy(chatId, 2) == 5 ||
answerCache.getBy(chatId, 2) == 6)) {
        gifts.add("Приставка для розумного холодильника");
    }
    if (answerCache.getBy(chatId, 6) == 13) {
        gifts.add("Теммобілізна");
        gifts.add("Кросівки");
        gifts.add("Спортивна форма");
    }
    if (answerCache.getBy(chatId, 7) == 22) {
        gifts.add("Фітнес браслет");
        gifts.add("Бутылка для води");
    }
    if (answerCache.getBy(chatId, 7) == 18 || answerCache.getBy(chatId, 7) == 19 ||
answerCache.getBy(chatId, 7) == 21) {
        gifts.add("М'яч");
    }
    if (answerCache.getBy(chatId, 6) == 14) {
        gifts.add("Електронна книга");
    }
    if (answerCache.getBy(chatId, 8) == 26 && answerCache.getBy(chatId, 1) == 1) {
        gifts.add("Книжна 'Життя у позику'");
    }
    if (answerCache.getBy(chatId, 8) == 26 && answerCache.getBy(chatId, 1) == 2) {
        gifts.add("Книжна 'До зустрічі з тобою'");
        gifts.add("Книжна 'Зоряний пил'");
    }
    if (answerCache.getBy(chatId, 8) == 24 && answerCache.getBy(chatId, 1) == 1) {
        gifts.add("Книжна 'Убивство в Східному експресі'");
    }
    if (answerCache.getBy(chatId, 8) == 24 && answerCache.getBy(chatId, 1) == 2) {

```

```

    gifts.add("Книжна 'Бентежна кров'");
    gifts.add("Книжна 'Ребекка'");
}
if (answerCache.getBy(chatId, 8) == 23 && answerCache.getBy(chatId, 1) == 1) {
    gifts.add("Книжна 'Сліди на дорозі'");
    gifts.add("Книжна 'Війна як внутрішнє переживання'");
}
if (answerCache.getBy(chatId, 8) == 23 && answerCache.getBy(chatId, 1) == 2) {
    gifts.add("Книжна 'Щоденник книгаря'");
}
if (answerCache.getBy(chatId, 7) == 22) {
    gifts.add("Фітнес браслет");
    gifts.add("Бутылка для води");
}
if (answerCache.getBy(chatId, 6) == 15 && answerCache.getBy(chatId, 1) == 1) {
    gifts.add("Похідний рюкзак на 100л.");
}
if (answerCache.getBy(chatId, 6) == 15 && answerCache.getBy(chatId, 1) == 2) {
    gifts.add("Похідний рюкзак на 80.");
}
if (answerCache.getBy(chatId, 6) == 15 && answerCache.getBy(chatId, 4) == 11) {
    if (answerCache.getBy(chatId, 5) == 11) {
        gifts.add("3-х місна палатка");
    } else {
        gifts.add("2-х місна палатка");
    }
}
if (answerCache.getBy(chatId, 6) == 15 && answerCache.getBy(chatId, 4) == 12) {
    gifts.add("1 місна палатка");
}
if (answerCache.getBy(chatId, 6) == 15 && answerCache.getBy(chatId, 2) == 3) {
    gifts.add("КаріMAT");
}
if (answerCache.getBy(chatId, 6) == 15 && answerCache.getBy(chatId, 2) == 4) {
    gifts.add("Спальнік");
}
if (answerCache.getBy(chatId, 6) == 15 && answerCache.getBy(chatId, 2) == 5) {
    gifts.add("Термос");
}
if (answerCache.getBy(chatId, 6) == 15 && answerCache.getBy(chatId, 2) == 6) {
    gifts.add("Палиці гірськолижні");
}
if (answerCache.getBy(chatId, 11) == 11 && answerCache.getBy(chatId, 1) == 1) {
    gifts.add("Набір інструментів");
}
if (answerCache.getBy(chatId, 11) == 11 && answerCache.getBy(chatId, 1) == 2) {
    gifts.add("Огранайзер для косметики");
}
if (answerCache.getBy(chatId, 12) == 11 && answerCache.getBy(chatId, 6) == 15) {
    gifts.add("Настінна карта подорожей");
}
if (answerCache.getBy(chatId, 12) == 11) {

```

```

    gifts.add("Планер");
}
if (answerCache.getBy(chatId, 14) == 28) {
    gifts.add("Поводок");
}
if (answerCache.getBy(chatId, 14) == 27) {
    gifts.add("Переноска");
}

if (answerCache.getBy(chatId, 1) == 2) {
    gifts.add("Засіб по догляду за волоссям");
    gifts.add("Доглядова косметика");
    gifts.add("Сертифікат в спа");
}
if (answerCache.getBy(chatId, 1) == 2 && (answerCache.getBy(chatId, 2) == 3 ||
answerCache.getBy(chatId, 2) == 4)) {
    gifts.add("Клатч");
}
if (answerCache.getBy(chatId, 1) == 2 && (answerCache.getBy(chatId, 2) == 5 ||
answerCache.getBy(chatId, 2) == 6)) {
    gifts.add("Сумка");
}
if (answerCache.getBy(chatId, 1) == 1) {
    gifts.add("Електрична бритва");
    gifts.add("Гаманець");
}

    gifts.add("Плед");
    gifts.add("Парфуми");
    gifts.add("Настільні ігри");
    return gifts;
}

private void show(Long chatId, Map<User, List<String>> collect) {
    StringBuilder sb = new StringBuilder();
    collect.keySet()
        .forEach(k -> {
            sb.append(k).append(":").append(LINE_SEPARATOR);
            List<String> gifts = collect.get(k);
            gifts.forEach(g -> sb.append(g).append(LINE_SEPARATOR));
        });
    SendMessage sendMessage = SendMessage.builder()
        .text(sb.toString())
        .chatId(String.valueOf(chatId))
        .build();
    messageSender.sendMessage(sendMessage);
}
}

```

Клас MessageSenderImpl

```

package com.example.messagesender;

import com.example.bot.TelegramBot;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.exceptions.TelegramApiException;

@Service
public class MessageSenderImpl implements MessageSender {

    private TelegramBot telegramBot;

    @Autowired
    public void setTelegramBot(TelegramBot telegramBot) {
        this.telegramBot = telegramBot;
    }

    @Override
    public void sendMessage(SendMessage sendMessage) {
        try {
            telegramBot.execute(sendMessage);
        } catch (TelegramApiException e) {
            throw new RuntimeException(e);
        }
    }
}

```

Клас GiftBotMavenApplication

```

package com.example.giftbotmaven;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class GiftBotMavenApplication {

    public static void main(String[] args) {
        SpringApplication.run(GiftBotMavenApplication.class, args);
    }

}

```

Клас TestResult

```

package com.example.entity;

import lombok.Builder;
import lombok.Data;

@Data

```

```

@Builder
public class TestResult {

    private int questionId;
    private int answerId;
}

```

Класс User

```

package com.example.entity;

import lombok.Builder;
import lombok.Data;

import java.util.List;

@Data
@Builder
public class User {

    private long chatId;
    private String name;
    private List<String> recommendations;
    private boolean isAdmin;
}

```

Класс AnswerCache

```

package com.example.cache;

import com.example.entity.TestResult;
import org.springframework.stereotype.Component;

import javax.inject.Singleton;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

@Component
@Singleton
public class AnswerCache {

    private Map<Long, List<TestResult>> chatIdToTestResultsTest = new HashMap<>();

    public void addQuestionId(long chatId, int questionId) {
        TestResult build = TestResult.builder().questionId(questionId).build();
        chatIdToTestResultsTest.get(chatId).add(build);
    }

    public void addAnswerId(long chatId, int answerId) {
        TestResult
        last
        chatIdToTestResultsTest.get(chatId).get(chatIdToTestResultsTest.get(chatId).size());
    }
}

```

```

    last.setAnswerId(answerId);
}

public int getBy(long chatId, int questionId){
    return chatIdToTestResultsTest.get(chatId).stream()
        .filter(tr -> tr.getQuestionId() == questionId)
        .findFirst()
        .get()
        .getAnswerId();
}

public List<TestResult> getBy(long chatId){
    return chatIdToTestResultsTest.get(chatId);
}

public void remove(long chatId){
    chatIdToTestResultsTest.remove(chatId);
}

public Map<Long, List<TestResult>> getAll(){
    return chatIdToTestResultsTest;
}
}

```

Клас AnswerWeriantCache

```

package com.example.cache;

import org.springframework.stereotype.Component;

import javax.inject.Singleton;
import java.util.HashMap;
import java.util.Map;

@Component
@Singleton
public class AnswerWeriantCache {

    private Map<Integer, String> answerMap = new HashMap<>();

    public void init(){
        if (!answerMap.isEmpty()){
            return;
        }
        answerMap.put(1, "Чоловік");
        answerMap.put(2, "Жінка");
        answerMap.put(3, "18-25");
        answerMap.put(4, "25-35");
        answerMap.put(5, "35-50");
        answerMap.put(6, "50+");
        answerMap.put(7, "Молодший спеціаліст");
    }
}

```

```

    answerMap.put(8, "Спеціаліст");
    answerMap.put(9, "Старший сеціаліст");
    answerMap.put(10, "Голова відділу");
    answerMap.put(11, "Так");
    answerMap.put(12, "Ні");
    answerMap.put(13, "Спорт");
    answerMap.put(14, "Читання");
    answerMap.put(15, "Туризм");
    answerMap.put(16, "Рукоділля");
    answerMap.put(17, "Інше");
    answerMap.put(18, "Футбол");
    answerMap.put(19, "Волейбол");
    answerMap.put(20, "Перегони");
    answerMap.put(21, "Баскетбол");
    answerMap.put(22, "Біг");
    answerMap.put(23, "Документальна література");
    answerMap.put(24, "Детективи");
    answerMap.put(25, "");
    answerMap.put(26, "Любовні романи");
    answerMap.put(27, "Кіт");
    answerMap.put(28, "Собака");
}

public String getBy(int id){
    return answerMap.get(id);
}

public int getBy(String test){
    return answerMap.keySet().stream()
        .filter(k -> answerMap.get(k).equalsIgnoreCase(test))
        .findFirst().get();
}
}

```

Клас QuestionCache

```

package com.example.cache;

import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Component;

import javax.inject.Singleton;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

@Component
@Singleton
@RequiredArgsConstructor
public class QuestionCache {

    private final AnswerWeriantCache answerWeriantCache;

```



```

private Map<Integer, String> questionMap = new HashMap<>();

public void init(){
    if (!questionMap.isEmpty()){
        return;
    }
    answerWeriantCache.init();
    questionMap.put(1, "Вкажіть статть");
    questionMap.put(2, "Вкажіть вік");
    questionMap.put(3, "Вкажіть вашу посаду");
    questionMap.put(4, "Маєте кохану людину?");
    questionMap.put(5, "Маєте дітей?");
    questionMap.put(6, "Яке ваше хобі?");
    questionMap.put(7, "Який саме вид спорту?");
    questionMap.put(8, "Що любите читати?");
    questionMap.put(9, "Подобаються гаджети?");
    questionMap.put(10, "Подобається активний відпочинок?");
    questionMap.put(11, "Подобається купувати цікаві речі для дому?");
    questionMap.put(12, "Подобається все планувати/записувати?");
    questionMap.put(13, "Маєте домашніх улюбленців?");
    questionMap.put(14, "Кого саме?");
}

public String getBy(int id){
    return questionMap.get(id);
}

public List<String> getAnswers(int qId){
    if (qId == 1){
        return List.of(answerWeriantCache.getBy(1), answerWeriantCache.getBy(2));
    }
    if (qId == 2){
        return List.of(answerWeriantCache.getBy(3), answerWeriantCache.getBy(4),
answerWeriantCache.getBy(5), answerWeriantCache.getBy(6));
    }
    if (qId == 3){
        return List.of(answerWeriantCache.getBy(7), answerWeriantCache.getBy(8),
answerWeriantCache.getBy(9), answerWeriantCache.getBy(10));
    }
    if (qId == 6){
        return List.of(answerWeriantCache.getBy(13), answerWeriantCache.getBy(14),
answerWeriantCache.getBy(15), answerWeriantCache.getBy(16),
answerWeriantCache.getBy(17));
    }
    if (qId == 7){
        return List.of(answerWeriantCache.getBy(18), answerWeriantCache.getBy(19),
answerWeriantCache.getBy(20), answerWeriantCache.getBy(21),
answerWeriantCache.getBy(22), answerWeriantCache.getBy(17));
    }
    if (qId == 8){

```

```

        return List.of(answerWeriantCache.getBy(23), answerWeriantCache.getBy(24),
answerWeriantCache.getBy(25), answerWeriantCache.getBy(26));
    }
    if (qId == 14){
        return List.of(answerWeriantCache.getBy(27), answerWeriantCache.getBy(28),
answerWeriantCache.getBy(17));
    }
    return List.of(answerWeriantCache.getBy(11), answerWeriantCache.getBy(12));
}
}
}

```

Клас UserCache

```

package com.example.cache;

import com.example.entity.User;
import org.springframework.stereotype.Component;

import javax.inject.Singleton;
import java.util.HashMap;
import java.util.Map;

@Component
@Singleton
public class UserCache {

    private Map<Long, User> chatIdToUserMap = new HashMap<>();

    public void add(User user){
        chatIdToUserMap.put(user.getChatId(), user);
    }

    public void remove(long chatId){
        chatIdToUserMap.remove(chatId);
    }

    public User getBy(long chatId){
        return chatIdToUserMap.get(chatId);
    }
}

```

Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

**«Інформаційна технологія для прийняття рішень щодо
відповідності вибору подарунка до вподобань
працівника»**

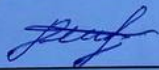
Виконала: студентка 2-го курсу, групи
1КН-22м
Спеціальності 122 – «Комп'ютерні
науки»



Панасюк Н. А.

(прізвище та ініціали)

Керівник: д.т.н., доц. каф. КН



Барабан С. В.

(прізвище та ініціали)

«07» 12 2023 р.

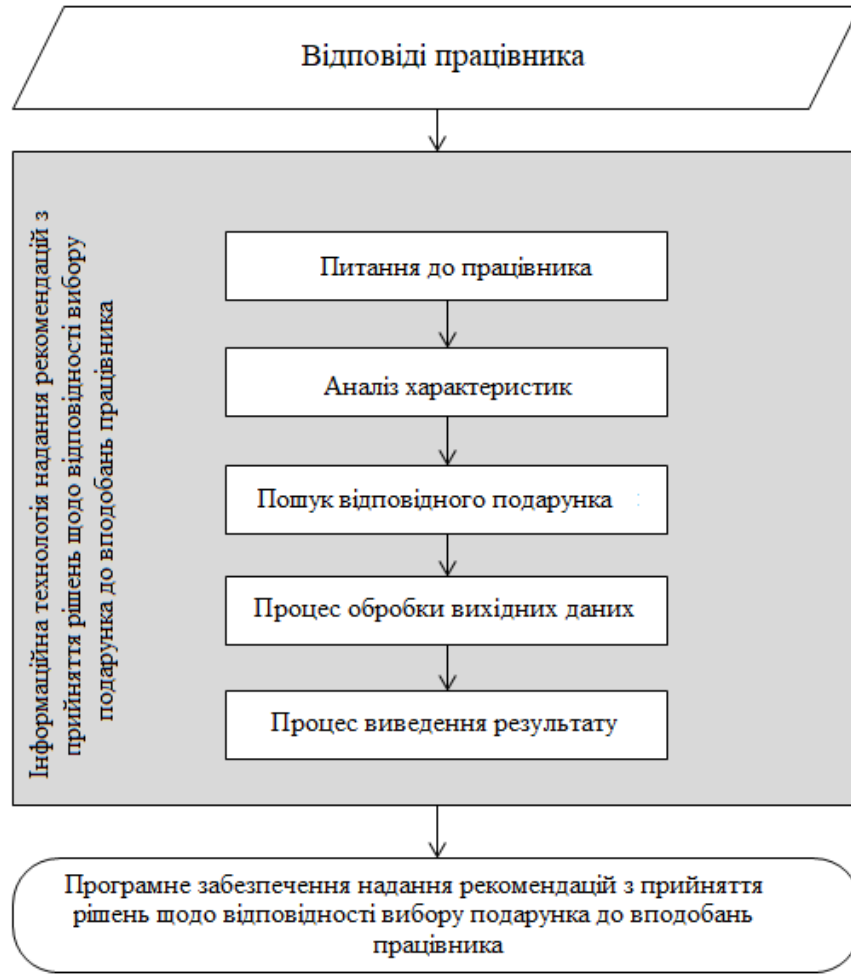


Рисунок В.1 – Структурна схема інформаційної технології

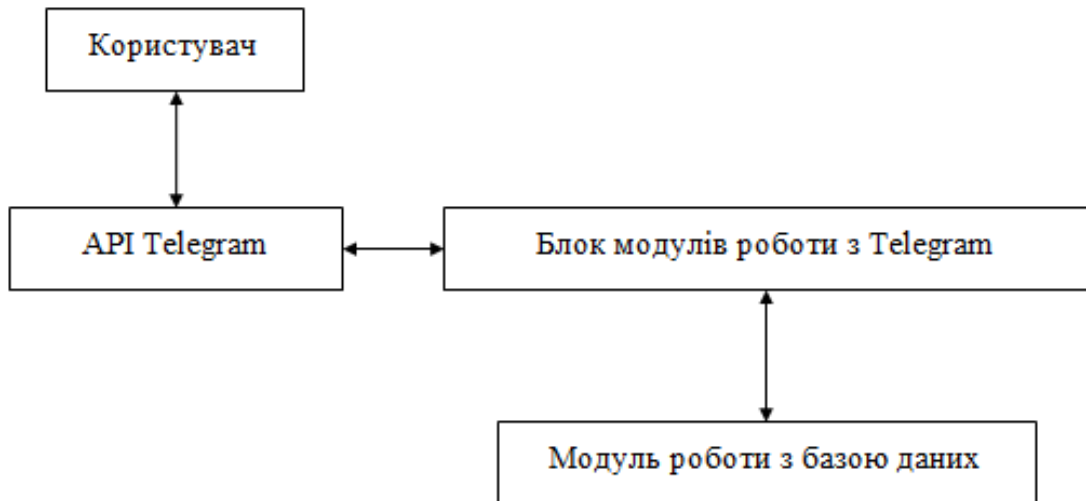


Рисунок В.2 – Загальна структурна схема

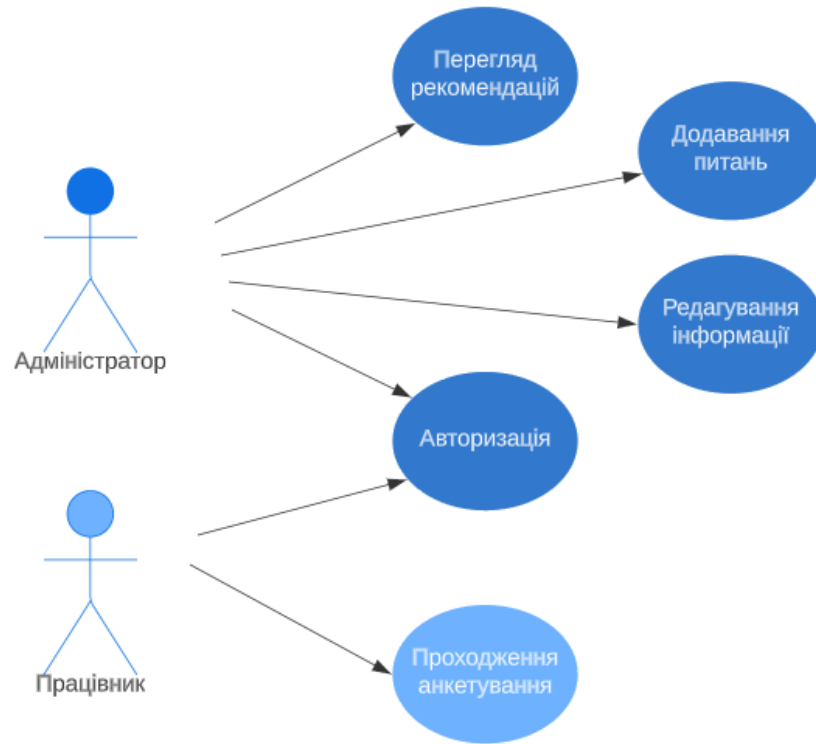


Рисунок В.3 – Діаграма варіантів використання

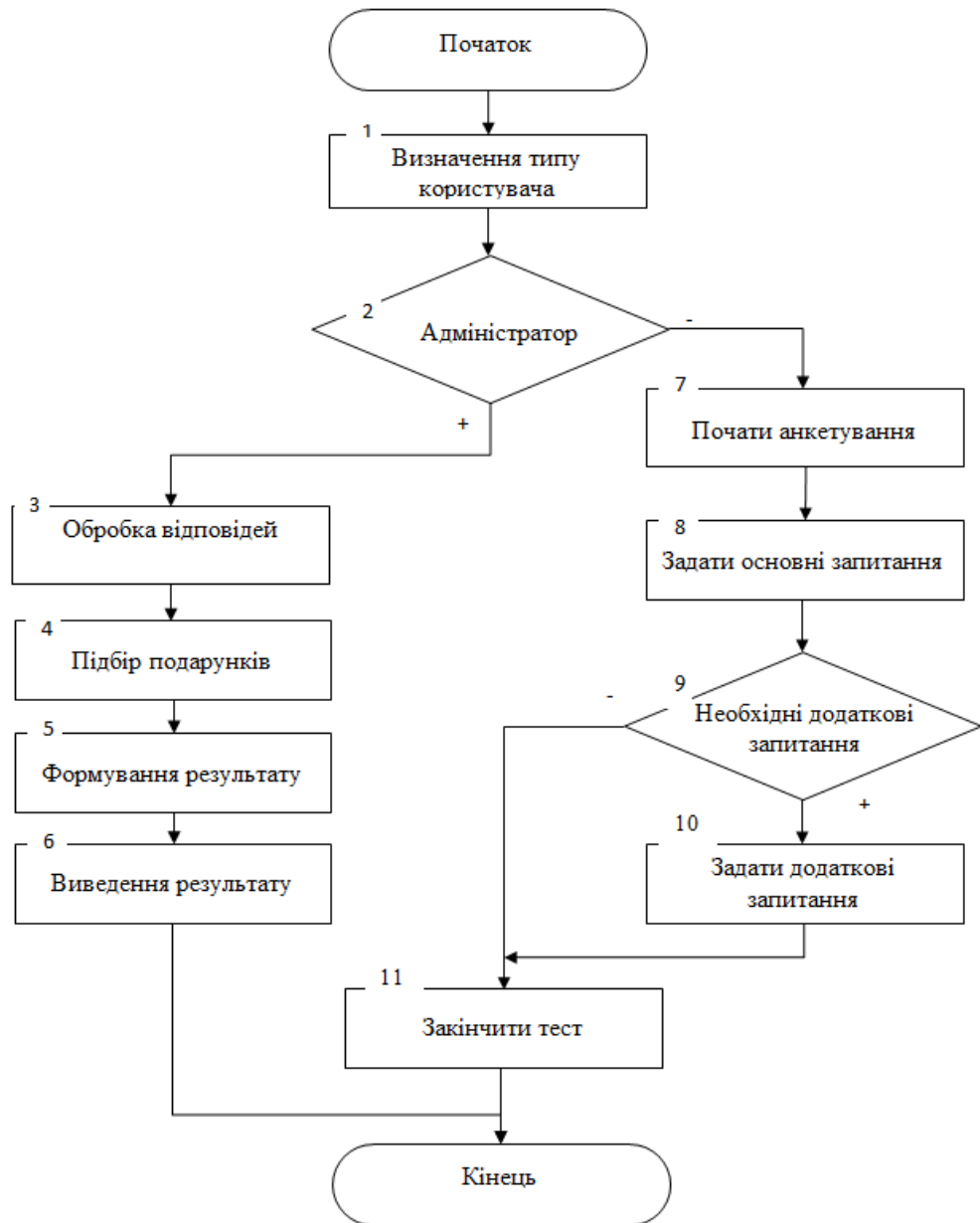


Рисунок В.4 – Схема загального алгоритму функціонування інформаційної технології

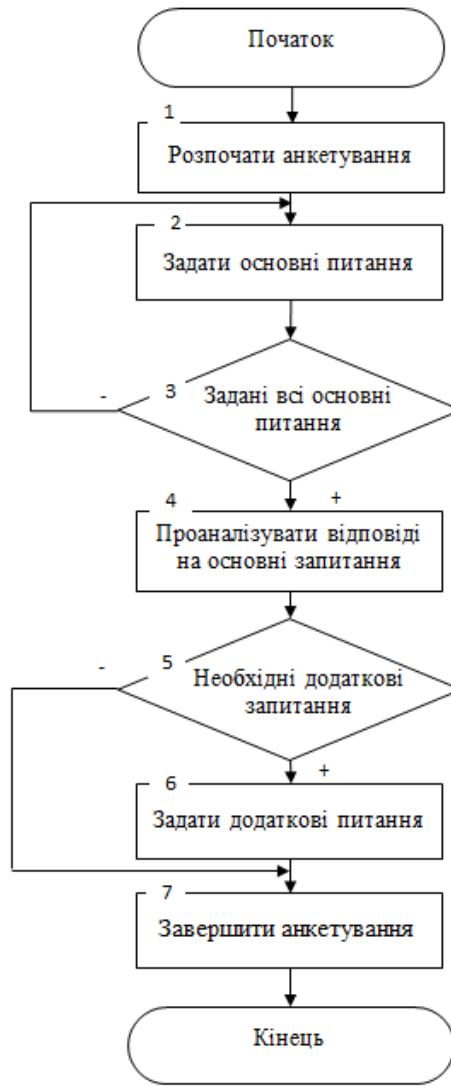


Рисунок В.5 – Схема алгоритму роботи інформаційної технології для ролі працівника

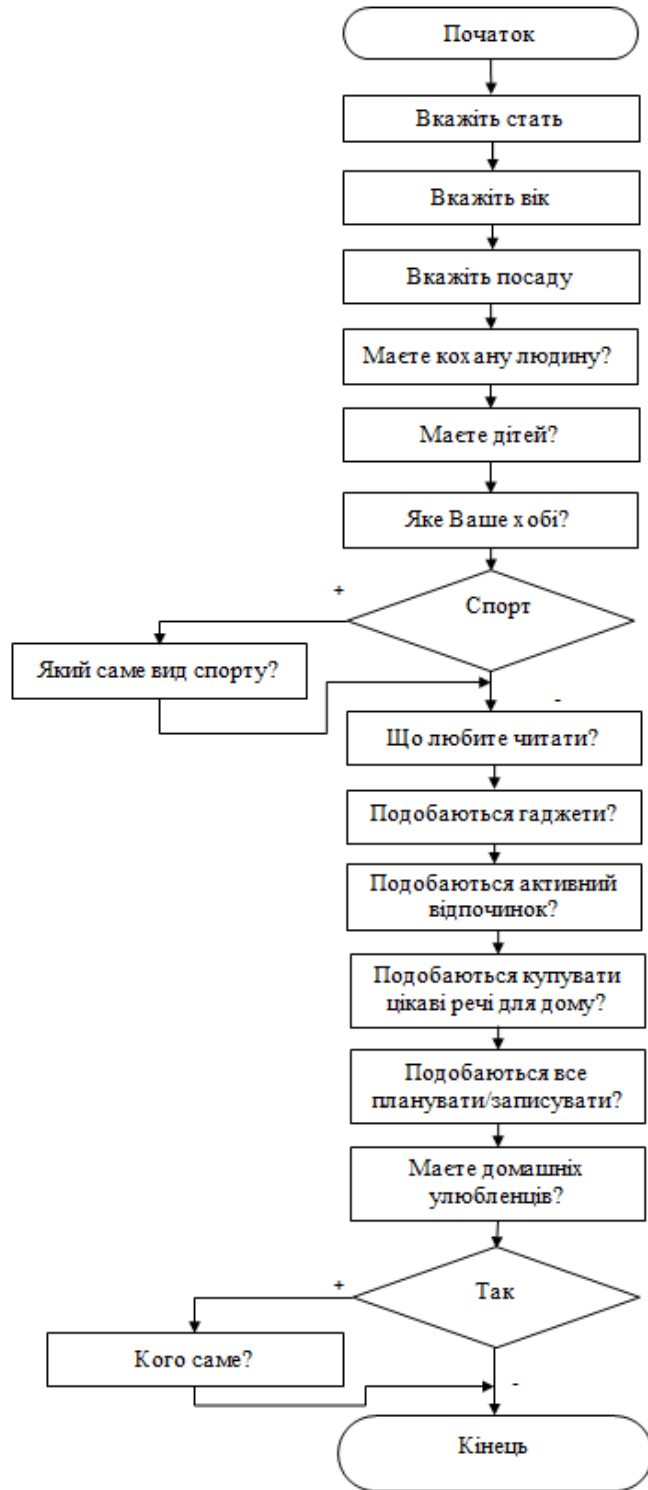


Рисунок В.6 – Схема формування запитань

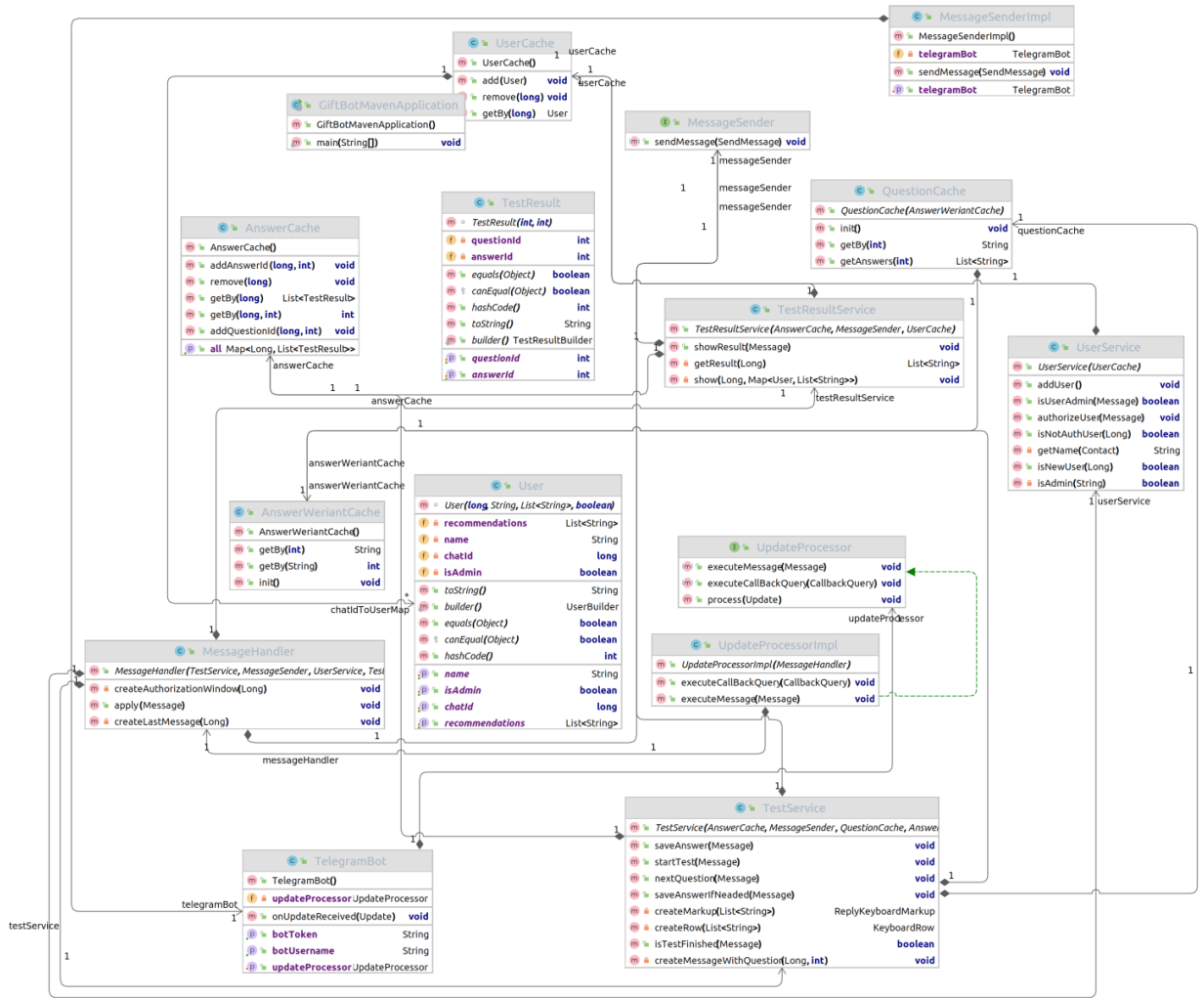


Рисунок В.7 – UML-діаграма класів

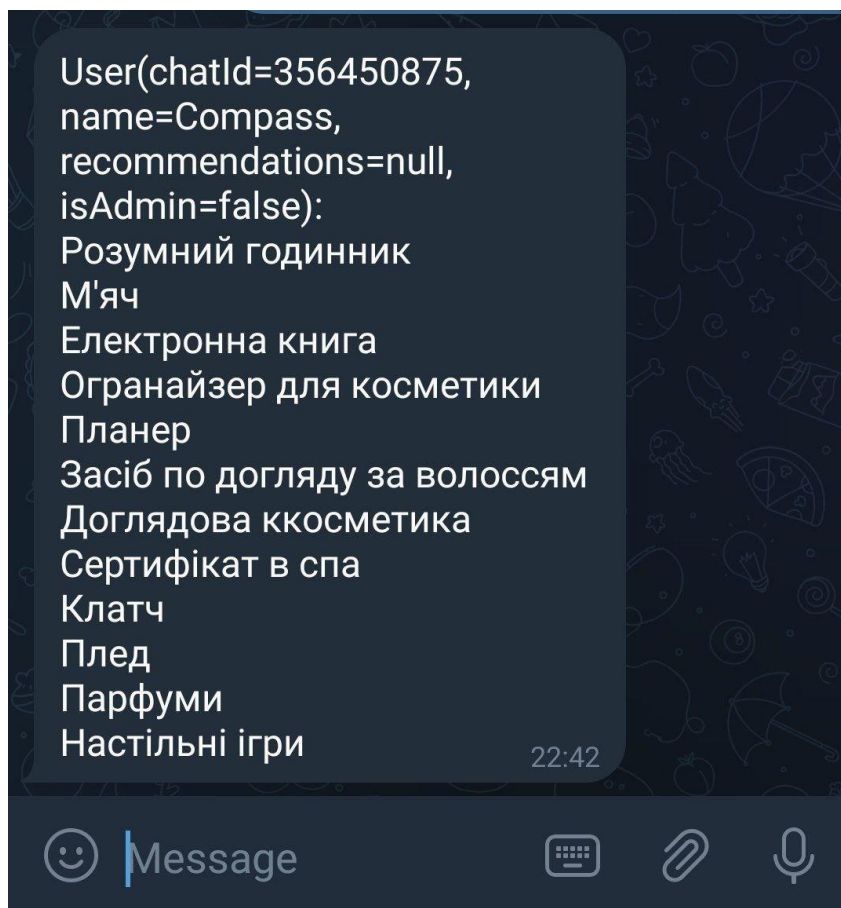


Рисунок В.8 – Приклад роботи програми

Додаток Г (довідниковий)

Інструкція користувача

Після запуску телеграм боту необхідно розпочати роботу з ним. Розроблений бот називається Gift helper. Після переходу до боту в телеграм необхідно розпочати спілкування, натиснувши пропоновану кнопку «Start» (рисунок Г.1).



Рисунок Г.1 – Початок роботи з ботом

Далі бот надсилає повідомлення «Почнемо після авторизації». Необхідно пройти авторизацію, для виявлення прав адміністратора. Користувач, що не має прав адміністратора, одразу проходить опитування ботом (рисунок Г.2). Користувачу пропонуються варіанти відповіді у вигляді кнопок з альтернативами.

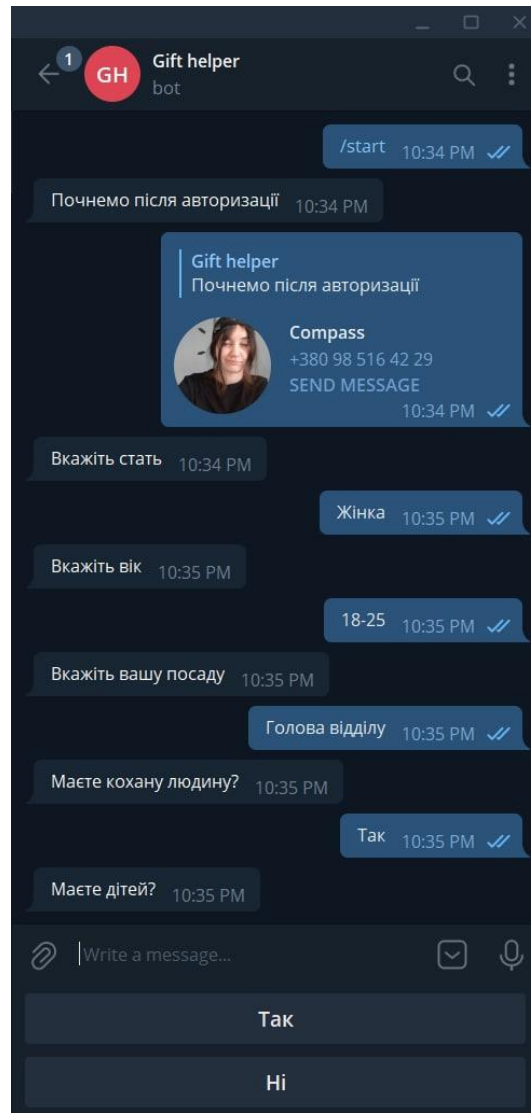


Рисунок Г.2 – Вкладка з початком анкетування працівника

Анкетування здійснюється у вигляді тесту з основними та додатковими запитаннями, необхідними для прийняття рішення щодо відповідності вибору подарунку до вподобань працівника (рисунок Г.3).

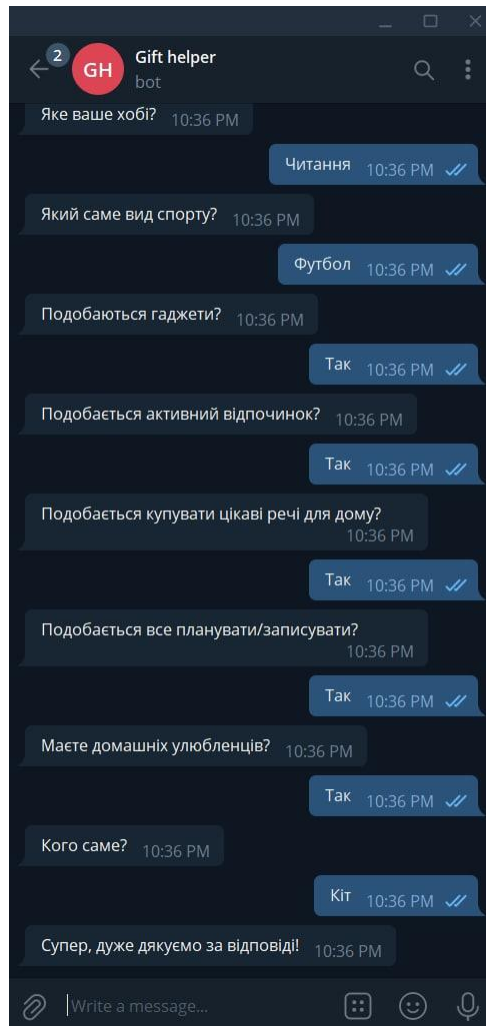


Рисунок Г.3 – Вкладка з продовженням опитування працівника

Після отримання ботом «Gift helper» надсилає повідомлення «Супер, дуже дякуємо за відповіді!», після чого діалог з ботом можна вважати завершеним. Бот опрацює відповіді працівника, сформує результат, що може бути виведений адміністратору.

Якщо авторизований користувач має права адміністратора, він має права для перегляду відповідей користувача. Саме для адміністратора відбуватиметься підбір подарунку конкретному користувачу та виведення результату. Результат виводиться у вигляді повідомлення зі списком подарунків, розташованих в порядку спадання відповідності до вподобань працівника. Тобто найкращий подарунок буде розташований на початку (рисунок Г.4)

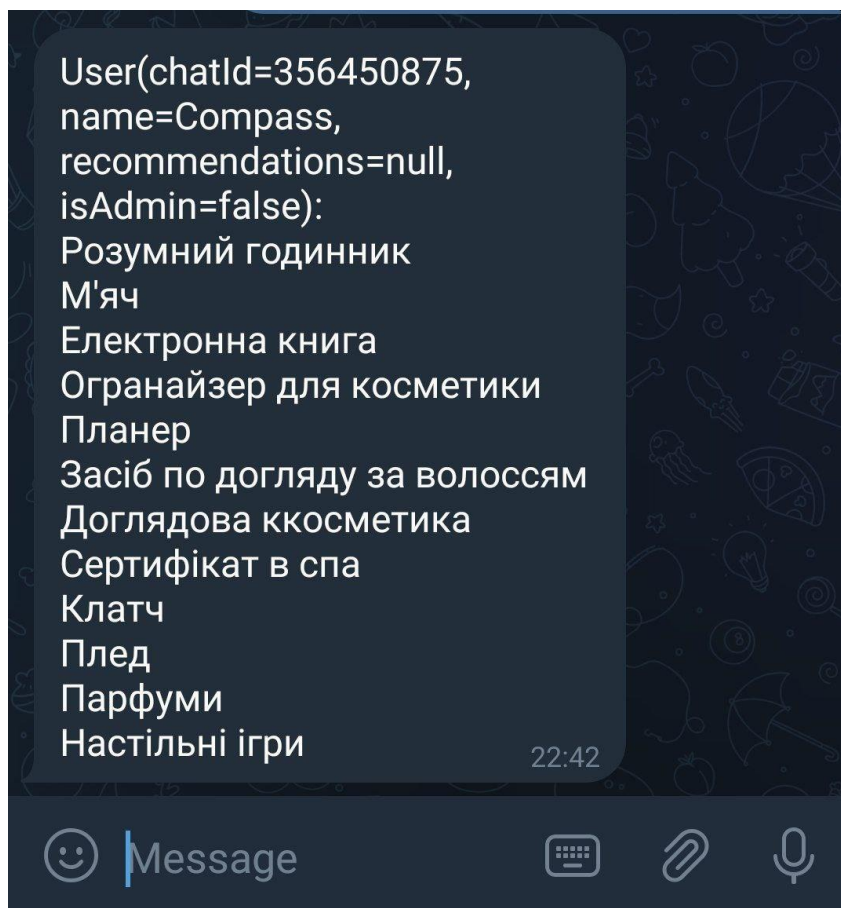


Рисунок Г.4 – Вкладка з результатом прийняття рішень