

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”

Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор

Олексій СМІРНОВ

“ ___ ” _____ 2022 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему

**“Дослідження та програмна реалізація хмарної
рекомендаційної системи обрання контенту з застосуванням
технологій штучного інтелекту”**

Виконав здобувач вищої освіти

II курсу, групи КН-21 М (1,4)

ОПП «Комп’ютерні науки»

спеціальності 122 «Комп’ютерні науки»

_____ Подкопаєв Д. М.

« ___ » _____ 2022р.

Керівник проекту

кандидат фізико-математичних наук, доцент

_____ Наталія ЯКИМЕНКО

« ___ » _____ 2022р.

Рецензент _____

м. Кропивницький

Центральноукраїнський національний технічний університет
Факультет *Механіко-технологічний*
Кафедра *Кібербезпеки та програмного забезпечення*
Рівень вищої освіти *магістр*
Галузь знань *12* "Інформаційні технології"
Спеціальність *122* "Комп'ютерні науки"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2022 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Подкопаєву Дмитру Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація хмарної рекомендаційної системи обрання контенту з застосуванням технологій штучного інтелекту*

2. Керівник роботи *Якименко Наталія Миколаївна, канд. фіз.-мат. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 18-13 від 17.08.2022 року

3. Строк подання студентом роботи до захисту *10.12.2022 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація хмарної рекомендаційної системи обрання контенту з застосуванням технологій штучного інтелекту*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна *1 аркуш*

Структурна схема системи *1 аркуш*

Функціональна схема системи *1 аркуш*

Діаграма процесів *1 аркуш*

Блок-схема алгоритму роботи додатку *4 аркуша*

Показники економічної ефективності *1 аркуш*

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---------------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| Економічний | Савеленко Г.В. | 05.10.2022 | 14.11.2022 |
| Охорона праці | Оришака О.В. | 06.10.2022 | 16.11.2022 |
| | | | |

7. Дата видачі завдання « 6 » вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти | Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти | Примітка |
|-------|---|---|----------|
| 1. | Аналіз існуючих систем | 10.10.2022 р. | |
| 2. | Постановка задачі, оформлення ТЗ | 15.10.2022 р. | |
| 3. | Розробка моделі компонента | 20.10.2022 р. | |
| 4. | Розробка структур даних | 25.10.2022 р. | |
| 5. | Розробка алгоритмів зв'язку та відображення | 30.10.2022 р. | |
| 6. | Програмування алгоритмів | 10.11.2022 р. | |
| 7. | Розрахунок економічної ефективності | 13.11.2022 р. | |
| 8. | Розрахунки з охорони праці та техніки безпеки | 15.11.2022 р. | |
| 9. | Оформлення ПЗ | 17.11.2022 р. | |
| 10. | Попередній захист роботи | 10.12.2022 р. | |
| | | | |

Дата видачі завдання
« 6 » вересня 2022 р.

Підпис керівника

Якименко Н.М.
(прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2022 р.

Підпис здобувача

Подкопаєв Д.М.
(прізвище та ініціали)

АНОТАЦІЯ

Подкопаєв Д.М. Дослідження та програмна реалізація хмарної рекомендаційної системи обрання контенту з застосуванням технологій штучного інтелекту. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2022.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначене для створення рекомендацій на базі хмарної рекомендаційної системи.

Метою розробки є дослідження та програмна реалізація рекомендаційної системи, яка базується на вмісті контенту та працює на неявних для користувача даних медіа-файлів.

Об'єктом дослідження є процес створення рекомендацій на базі контентної фільтрації.

Предметом дослідження є методи побудови рекомендацій на базі контентної фільтрації.

Методи дослідження базуються на методах Big Data, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація хмарної рекомендаційної системи обрання контенту з застосуванням технологій штучного інтелекту.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на PC з ОС 7/8/10/11. Linux. Mac OS. та мобільних системах Android, IOS та HarmonyOS .

Програму розроблено в середовищі Visyal Studio Code 1.73.1.

Ключові слова: комп'ютерні науки, системи рекомендацій.

ABSTRACT

Podkopaev D.M. Research and software implementation of cloud recommendation system for content selection using artificial intelligence technologies. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.

In this final qualification work for the second (master's) level of higher education, software has been developed that is designed to create recommendations based on a cloud recommendation system.

The purpose of the development is the research and software implementation of a recommendation system that is based on content and works on implicit data of media files for the user.

The object of research is the process of creating recommendations based on content filtering.

The subject of research is the methods of building recommendations based on content filtering

Research methods are based on Big Data methods, methods of mathematical statistics, software development methods.

The result of the work is the software implementation of a cloud-based content recommendation system using artificial intelligence technologies.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions on how to work with the software are given

The program can be used on PC with OS 7/8/10/11. Linux. Mac OS. and mobile systems Android, IOS and HarmonyOS..

The program is developed in the environment Visual Studio Code 1.73.1.

Keywords: computer science, recommendation systems.

ЗМІСТ

| | |
|--|----|
| ПЕРЕЛІК СКОРОЧЕНЬ, СИМВОЛІВ ТА СПЕЦІАЛЬНИХ ТЕРМІНІВ | 4 |
| ВСТУП..... | 5 |
| 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ | 8 |
| 1.1 Призначення системи..... | 8 |
| 1.2 Область застосування..... | 9 |
| 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ | 11 |
| 2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень по профілю теми бакалаврської дипломної роботи роботи | 11 |
| 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування | 21 |
| 2.3 Розгорнута постановка завдання | 25 |
| 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ | 27 |
| 3.1 Опис функціонування системи | 27 |
| 3.2 Розробка структурної схеми..... | 33 |
| 3.3 Розробка функціональної схеми | 36 |
| 3.4 Розробка діаграми процесів..... | 37 |
| 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.... | 39 |
| 4.1 Розробка блок–схем та опис алгоритмів функціонування системи..... | 39 |
| 4.2 Захист розробленого програмного забезпечення..... | 45 |
| 5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ | 47 |
| 6 НАУКОВА НОВИЗНА | 49 |
| 7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ..... | 50 |
| 7.1 Техніко економічне обґрунтування теми магістерської роботи..... | 50 |

| | | | | | | | | |
|-----------|----------------|----------|--------|------|--|----------------------------|-------|---------|
| | | | | | | ВКРМ -122.22.0011.00.00.ПЗ | | |
| Вим | Арк. | № докум. | Підпис | Дата | | | | |
| Розробив | Подкопасв Д.М. | | | | Дослідження та програмна реалізація хмарної рекомендаційної системи обрання контенту з застосуванням технологій штучного інтелекту | Літ. | Аркуш | Аркушів |
| Перевірів | Якименко Н.М. | | | | | М | 2 | 92 |
| Н. Контр. | Гермак В.С. | | | | ЦНТУ КІ-21М-1,4 | | | |
| Затв. | Смірнов О.А. | | | | | | | |

| | |
|--|----|
| 7.2 Розрахунок трудомісткості розробки програмної продукції..... | 52 |
| 7.3 Визначення чисельності виконавців і планового фонду зарплати..... | 54 |
| 7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника..... | 58 |
| 7.5 Визначення собівартості розробки та ціни програмної продукції..... | 63 |
| 7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції..... | 67 |
| 7.7 Визначення експлуатаційних витрат..... | 68 |
| 7.8 Визначення економічної ефективності програмної продукції..... | 69 |
| 7.9 Висновок..... | 71 |
| 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ..... | 72 |
| 8.1 Вступ..... | 72 |
| 8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером..... | 73 |
| 8.3 Дослідження та аналіз санітарно-гігієнічних умов праці на робочому місці програміста..... | 74 |
| 8.4 Розробка заходів з умов поліпшення охорони праці..... | 77 |
| 8.5 Розрахункова частина..... | 79 |
| 8.6 Висновок..... | 86 |
| 9 ОСНОВНІ ВИСНОВКИ..... | 87 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 88 |

| | | | | |
|------|------|-------------|--------|------|
| | | | | |
| Вим. | Арк. | № документа | Підпис | Дата |

**ПЕРЕЛІК СКОРОЧЕНЬ, СИМВОЛІВ ТА СПЕЦІАЛЬНИХ
ТЕРМІНІВ**

ПЗ – програмне забезпечення

AI – штучний інтелект

РС – рекомендаційні системи

ML – машинне навчання

API – прикладний програмний інтерфейс

SaaS – програмний продукт

ID – ідентифікатор

ВКРМ -122.22.0011.00.00.ПЗ

| | | | | | | |
|-----|-----|----------|--|--------|--|------|
| | | | | | | Арк. |
| Вим | Арк | № докум. | | Підпис | | 4 |

ВСТУП

В сучасному світі з вільним доступом до будь-якого контенту, такого як: фільми, музика, книги, відео контент та зображення складно обійтись без рекомендаційних систем, які допомагають відсортувати контент за низкою показників, що буде корисним і цікавим певній ланці людей.

Рекомендаційні системи (РС) – це комплекс алгоритмів, програм і послуг, завдання яких полягає в прогнозуванні, які об'єкти, товари або послуги найбільше зацікавлять користувачів, спираючись на наявну інформацію про користувача.

Контент, який ми споживаємо, впливає на те як ми мислимо, на формування наших смаків, звичок тощо. Тому розробка рекомендаційних систем дозволяє нам створити набори контенту для задоволення потреб, або для керування цілими групами людей. Знаючи, що цікавить людей, їх звички, інтереси і пристрасті, можна побудувати нову систему цінностей.

«Ми те – що ми дивимося» - саме так системи ідентифікують нас за низкою показників: вік, стать, регіон проживання, політичні погляди, фінансова спроможність і т.д.

Компанії вкладають мільйони власних коштів, щоб проаналізувати своїх користувачів, створивши для них продукт і збираючи метрику їх приладів. Кожний наш прилад, який дає можливість вийти в інтернет і отримати контент, має низку ідентифікаторів, де зібрана детальна інформація про те, що ми дивимося, як ми це робимо, в який час та скільки часу це займає.

Найпростіший приклад це рекламний ідентифікатор, який аналізує дані нашого пошуку в глобальній мережі. Який товар ми переглядаємо, на яких маркетплейсах ми зосереджуємо свою увагу. Потім ці дані нам явно підсуюються в рекламних вставках та банерах, на різних сайтах та додатках, які

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 5 |

базують рекламу на цьому ідентифікаторі. Як показує практика, це всі додатки, які встановлено на нашому телефоні.

Отже рекомендаційні системи, це чудовий інструмент як для користувачів, оскільки вони отримують контент, який шукають. Так і для компаній, оскільки це допомагає проводити моніторинг ринку. Залишаючись актуальним для швидкоплинного ринку та динамічної аудиторії.

Метою даної роботи є програмна реалізація рекомендаційної системи, яка базується на вмісті контенту та працює на неявних для користувача даних медіа-файлів і здатна надавати релевантні рекомендації.

Предметом дослідження є сучасні програмні рішення та технології для систем рекомендацій медіа-контенту, які забезпечують потреби користувача.

Розробка будь-якої рекомендаційної системи потребує великої кількості спеціалістів та даних. Збором та дослідженням даних займаються спеціалісти науки про дані (Data science), що об'єднує в собі різні галузі аналізу даних, таких як статистика, класифікація, добування даних, передбачувальна аналітика, кластеризація та машинне навчання.

Збір та аналіз даних користувачів дозволяє передбачати дії користувачів і динамічно реагувати на них. Підставляючи все новий і новий контент. За такою схемою працюють більшість сервісів для перегляду відео контенту, таких як YouTube, TikTok, Instagram Reels, Snapchat Spotlight.

Збираючи дані, про те скільки часу ми дивимось відео, якщо це коротке відео то рахується кількість повторних переглядів, загальна зацікавленість тематикою відео, яка підкріплюється реакцією користувача у вигляді коментаря, натискання на кнопку «Подобається» або поширення в інших соціальних мережах. В результаті ці дані піднімають ці відео в рекомендації для користувачів з максимальним збігом інтересів, роблячи його більш релевантним для показу.

Об'єктом дослідження є поєднання явних і неявних методів збору даних.

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 6 |

Явним методом вважаємо оцінку, яку встановлює користувач в наявній системі та спостереження за списком який створюю сам користувач, даними з яких можна корелювати його наступні рекомендації.

Неявним же збором буде спостереження за оглядом користувача в різних системах, дані з протоколів поведінки онлайн та відстежування вмісту приладу користувача.

Порівнявши однотипні дані від різних людей або груп людей, можемо сформулювати граф інтересів та за його показниками обчислити список рекомендацій для конкретної групи або конкретного користувача.

Також об'єктом дослідження виступає сам контент, як і користувачів він містить в собі купу даних за якими ми можемо його дослідити та розподілити за явними і неявними параметрами.

Новизною даної роботи є підбір контенту на основі його настрою, який містить неявні для користувача параметри але чітко співпадає зі смаками користувача. Настрої мають велику градацію параметрів, визначивши певні характеристики метаданих можна виділити їх залежність і як вони відкликаються у користувача.

Люди підбирають собі контент за настроєм або цим контентом формують його. Взявши це за основу, розробляється рекомендаційна система, яка бере вже відомий користувачу контент і на його основі формує список подібного. Оскільки система не звертає уваги на попередні дії користувача та його загальні дані вона кожного разу буде створювати нові рекомендації. Це в перспективі дозволить розширити кругозір користувачеві та підвищити його зацікавленість.

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | <i>ВКРМ -122.22.0011.00.00.ПЗ</i> | Арк. |
| Вим | Арк | № докум. | | Підпис | | 7 |

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Задачею моєї магістерської роботи було дослідити та програмно реалізувати хмарну рекомендаційну систему обрання контенту з застосуванням технологій штучного інтелекту. Що в моїй роботі дозволить користувачу отримати контент на базі його настрою або настрою обраного ним музичного твору.

1.1 Призначення системи

Перш за все, будь-яка рекомендаційна система призначена для рекомендацій, таких як, рекомендації товарів, медіа-контенту та послуг або сервісів.

Поширення рекомендаційних систем пов'язано з надмірною кількістю інформації в веб мережі. Кожного дня з'являється сотні відео, аудіо та інших цифрових товарів. Заплутати в кількості яких дуже легко, але ще складніше знайти щось для себе. На допомогу приходять рекомендаційні системи, які аналізують поведінку користувача, його потреби та запити. В результаті чого з легкістю можуть передбачити бажання користувача, або задовільнити його пошукові запити.

Варто відзначити, що рекомендаційні системи це не тільки чудовий механізм для користувача але і чудовий ринковий інструмент для компаній. Важко знайти компанію, яка відмовиться від постійного притоку нових користувачів, так і утримання вже існуючих в їх базі.

Збір даних та машинне навчання, невід'ємна частина рекомендаційних систем. Вони є основою таких систем.

Основними призначеннями системи є залучення користувача до споживання нової інформації на певній платформі та пропонування товарів

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 8 |

повсякденного використання на основі історії користувачів, що несе за собою мету підвищення середнього чеку і загальної лояльності клієнта до компанії.

Для вирішення цих задач існують такі підходи, як колаборативна, контентна, гібридна фільтрація та SVD.

Колаборативна фільтрація – метод прогнозу, який використовує відомі переваги оцінок групи користувачів для прогнозування невідомих переваг нового користувача. Схемою цього підходу є певна таблиця користувачів, які розподілені за схожістю, та на її основі прогнозуються результати для інших груп.

Контентна фільтрація базується на моделі самого об'єкту, оцінки якого будуть прогнозуватися. Кожний об'єкт отримує математичну модель з конкретних характеристик об'єкту. Рекомендації будуються на порівнянні поточного об'єкта та базою всіх об'єктів заданої категорії.

Гібридна фільтрація поєднує обидва підходи, та є найпопулярнішою для побудови рекомендаційних систем для веб ресурсів.

SVD алгоритм, який позбавлений загальних проблем попередніх підходів. Він являє собою гібридний алгоритм методу сингулярного розкладу, який створили саме для покращення результатів звичайних алгоритмів.

Призначення нової системи базується на контентній фільтрації, незалежно від користувача. Систему можна використовувати на різних платформах, оскільки вона шукає подібності незважаючи на явні характеристики такі, як жанр і виконавець. Тому буде цікаві варіанти подібностей до отриманого медіафайлу.

1.2 Області застосування

Області застосування найрізноманітніші, від пошуку фільмів, музики, ритейлу, електронної комерції до фільтрації контенту соціальних мереж.

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 9 |

У комплексі рекомендаційні системи – це великий клас моделей для практичної допомоги кожному класу бізнесу. Своєчасні рекомендації та користування правильним каналом зв'язку допомагає бізнесу продавати більше і якісніше.

Підвищення лояльності клієнта до компанії або соціальної мережі, породжують більшу зацікавленість новинками які вони можуть пропонувати своїм користувачам, знаходячи правильний відклик від аудиторії. Що призводить до збільшення споживання контенту та товарів та породжуватиме ріст прибутків.

Реклама - також чудове поле для використання рекомендаційних систем, коли певна група людей буде отримувати персоналізовані рекламні матеріали і ненав'язливі push-повідомлення. Збільшуючи тим самим відвідування постійними користувачами і зменшує відтік клієнтів.

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | <i>ВКРМ -122.22.0011.00.00.ПЗ</i> | Арк. |
| Вим | Арк | № докум. | | Підпис | | 10 |

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень при дослідженні та програмній реалізації хмарної рекомендаційної системи з застосуванням технологій штучного інтелекту

Рекомендаційні системи на кожному кроці, завжди поруч з нами. Далі розглянемо найпопулярніші варіанти, де ми зустрічаємось з РС в нашому житті.

Gboard

Gboard – віртуальна клавіатура, розроблена компанією Google у 2016 році для операційних систем Android та IOS. Об'єднавши в собі функціонал автодоповнення, прогнозування тексту та пошук від Google який видає прогнозовані відповіді являє собою чудовий приклад використання рекомендаційних систем в загальному вжитку. Адже кожного дня ми користуємося нею, в пошуку, в спілкуванні в соціальних мережах та в нотуванні заміток. Використовуючи зібрані дані від стилю ведення і популярності вживаних слів, дуже добре рекомендує текст.

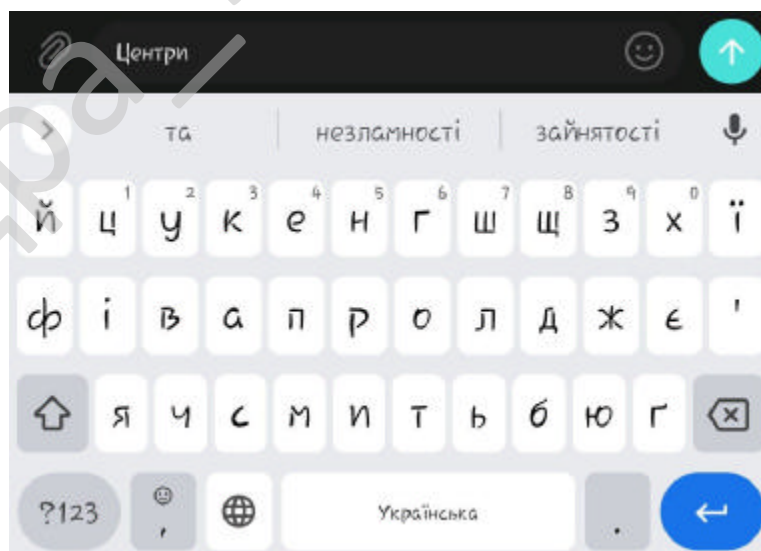


Рисунок 2.1 – Приклад рекомендації, створеної сервісом Gboard

| | | | | | | |
|-----|-----|----------|--|--------|----------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 11 |

Алло

Алло – це Українська компанія заснована в 1998 році, включає в себе онлайн та класичний ритейл, маркетплейс та дистриб'юцію електроніки. Розглянемо саме її онлайн ритейл. Для вироблення рекомендацій використовує збір персональних даних відповідно до «Положення про збір персональних даних» з яким можна ознайомитися в них на сайті або в мобільному додатку, аналізуючи взаємодії користувача з платформою, від замовлень до участі в акціях. Збір даних використовується для визначення характеристик платформи та пропозицій, які матимуть найбільшу зацікавленість у користувачів. Використовують так само файли cookie що допомагає зібрати інформацію про кількість і частоту запитів, а також відсіяти користувачів або пристрої, які порушують правила компанії. Для регулярної зацікавленості використовують push повідомлення, в яких вже відображаються пропозиції по товарам які можуть зацікавити користувача, на базі його попередніх запитів. Також може використовувати сповіщення в месенджерах тримаючи постійний зв'язок з клієнтом

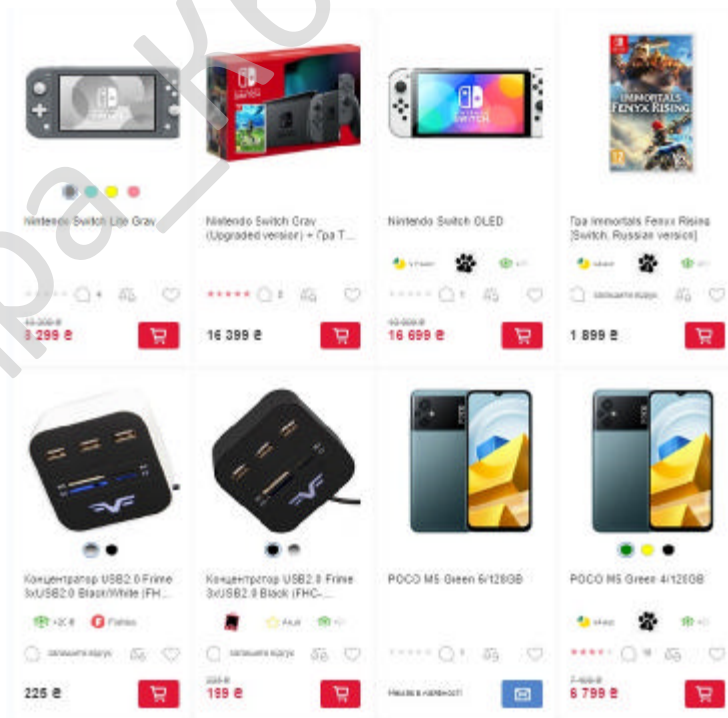


Рисунок 2.2 – Приклад рекомендації, створеної сервісом Алло

| | | | | | | |
|-----|-----|----------|--|--------|----------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 12 |

Rozetka

Rozetka – прямиий конкурент попередньої компанії та найбільший онлайн-ритейлер в Україні з 2005 року. Навіть мають звучний слоган з описом своєї рекомендаційної системи «Щоразу що треба». За своєю роботою мають максимальну схожість, оскільки використовують однакову систему збору персональних даних відповідно до «Положення про збір персональних даних» з яким можна ознайомитись в їх додатку або на сайті. Також присутні інструменти cookie та push повідомлення і не забувають про інтеграцію в месенджери.

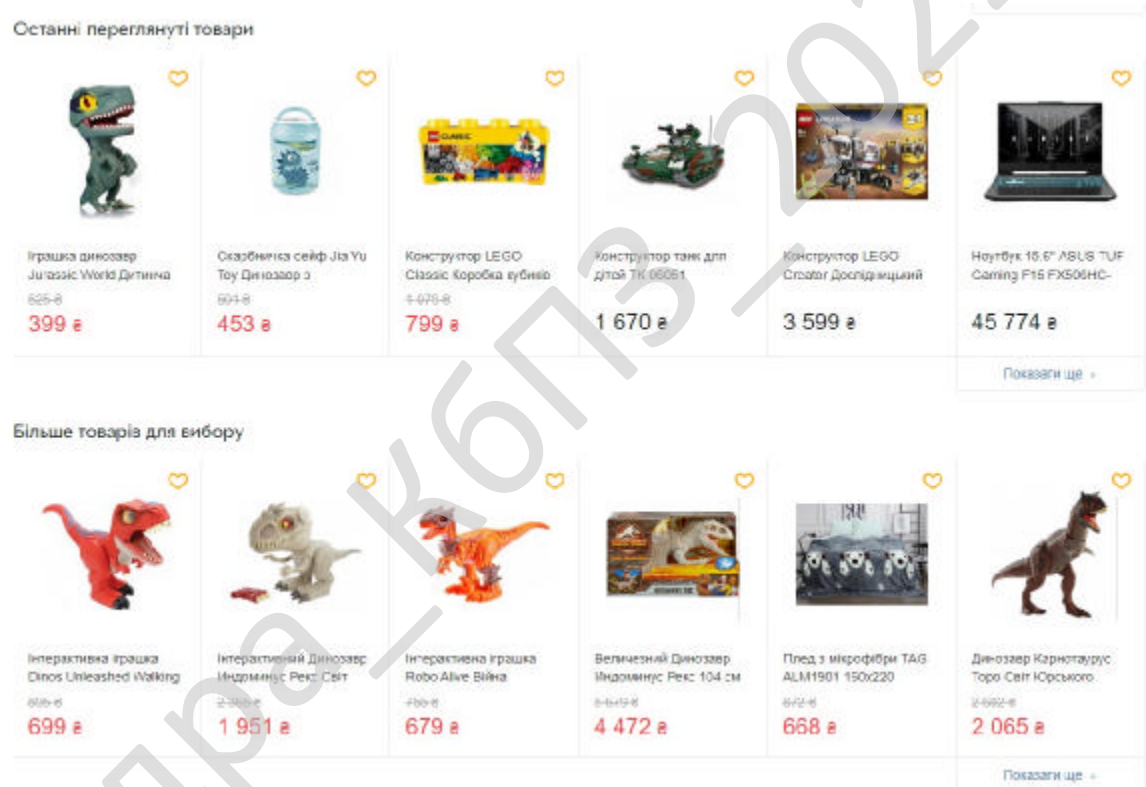


Рисунок 2.3 – Приклад рекомендації, створеної сервісом Rozetka

Розглянемо ще два приклади але вже міжнародних сервісів електронної комерції.

Amazon

Amazon – це перший сервіс електронної комерції, де було застосовано рекомендаційну систему. Сервіс використовує рекомендації на основі контенту. Ґрунтуючись на тому, що клієнти переглядали, купували та які оцінки з

відгуками залишали після цих операцій. Система охоплює всі дані які може зібрати, навіть ті які не вміє використовувати за рахунок чого забезпечує постійні 35% продажі.

Опираючись на обраний клієнтом товар Amazon рекомендує інші товари, придбані іншими користувачами, на основі матриці схожості з попередньою покупкою. В компанії є запатентований підхід під назвою Item-To-Item Collaborative Filtering (коллаборативна фільтрація від елемента до елемента). Успішно реалізуючи її на своїй платформі та так само успішно впроваджує в компаніях партнерах.

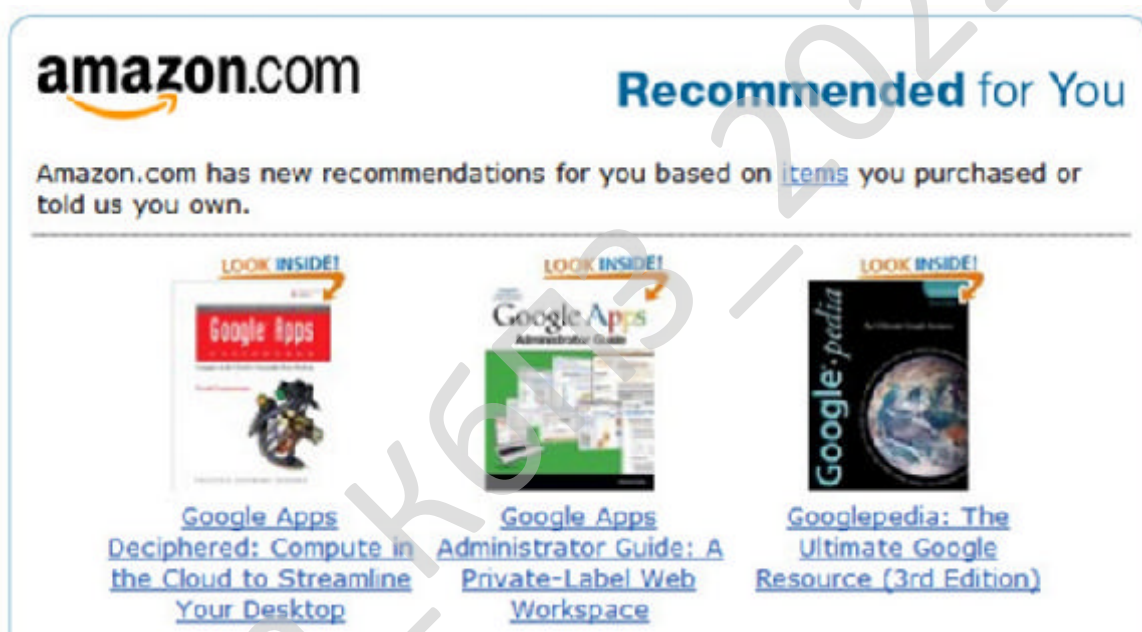


Рисунок 2.4 – Приклад рекомендації, створеної сервісом Amazon

eBay

eBay – торговий гігант та експериментатор з інструментами ML та методами AI, що впроваджений на всіх рівнях компанії. Розумне ранжування пошукової видачі та влучна реклама яка базується на їх алгоритмах приносить компанії додаткові продажі в близькі до 1 млрд. доларів кожного кварталу.

Їх система аналізує, все що продається на сайті, найбільшу ймовірність зацікавлення користувачів товаром і на кожному етапі стежить щоб рекомендації не дублювались від пошуку і до самої продажі товару.

Найважливішим в їх підході є те, щоб рекомендації викликали інтерес і змушували додати і їх в корзину.

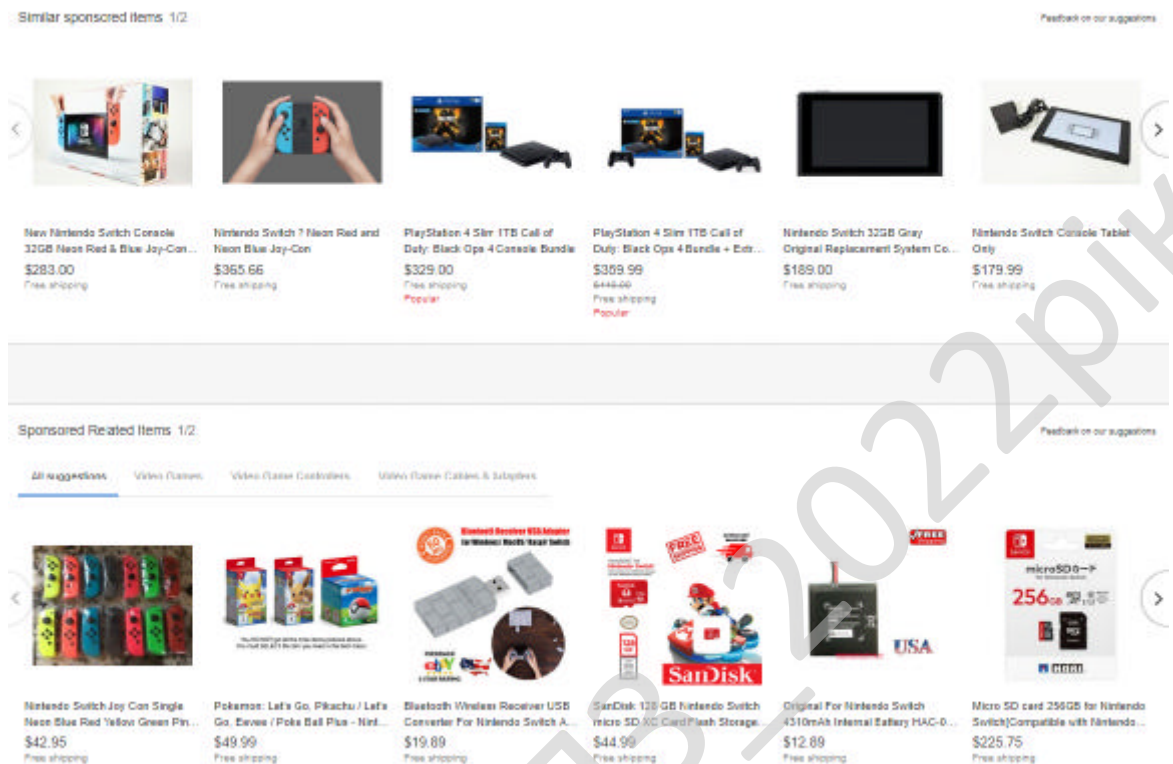


Рисунок 2.5 – Приклад рекомендації, створеної сервісом eBay

Музична сфера також не обійшлась без рекомендаційних систем. Розглянемо приклади двох найпопулярніших сервісів для прослуховування потокової музики.

Apple Music

Apple Music - музичний сервіс потокового прослуховування музики, з назви зрозуміло що розробила його компанія Apple. Користувачі можуть самостійно обирати музикальні твори, які бажають слухати на своїй приладі.

Сервіс пропонує підказки та рекомендації, що базуються на вподобаннях користувача. При першому старті сервіс запропонує скоригувати музичні переваги за жанрами та виконавцями, що відобразиться на щотижневих добірках. Також цей список потім можна корегувати в будь-який момент. Перші дні або навіть тижні сервіс розгойдується в добірках, це проблема холодного старту що проблемою кожної рекомендаційної системи на її старті, та потім з

зібраний плейлист, будуть грати подібні пісні до вже прослуханих. Кожного дня сервіс робить підбірки за жанрами які подобаються користувачу називаючи це «Мікси дня» а для постійних користувачів добірки за рік, та сезони добірки з пісень які прослуховувались на повторі в певний сезон.

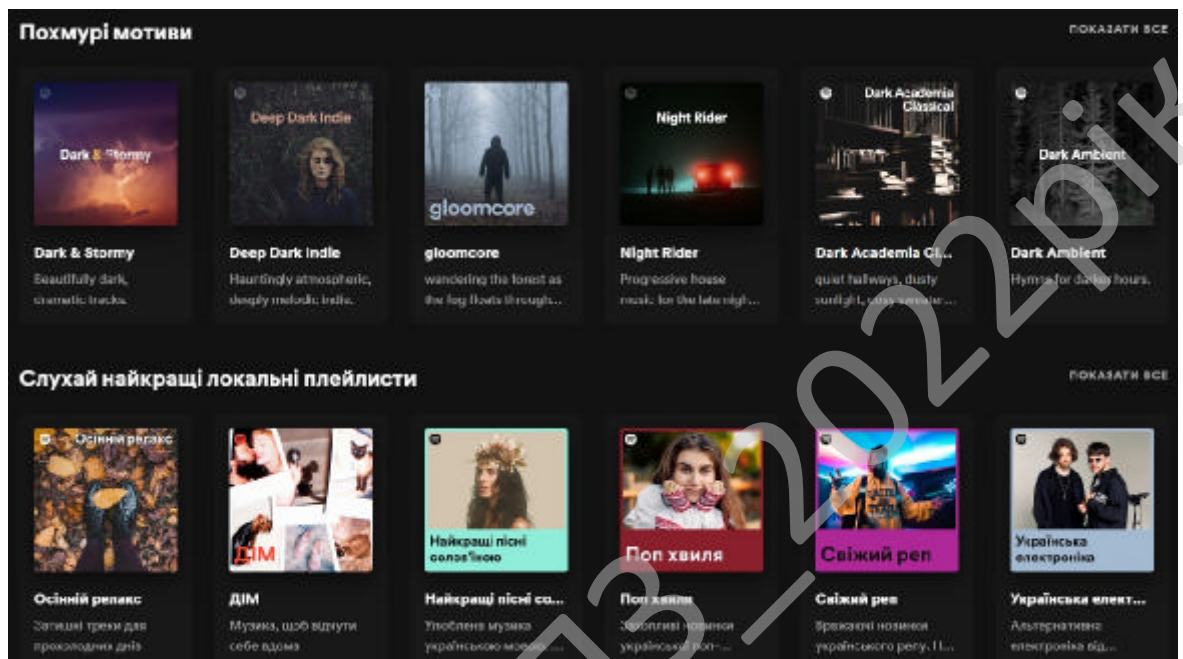


Рисунок 2.7 – Приклад рекомендації, створеної сервісом Spotify

Від музики плавно перейдемо до відео-контенту.

Youtube

Youtube – найпопулярніший відеохостинг з 2005 року, який постійно змінюється і модернізується, що зробило його найпопулярнішим на фоні всіх його конкурентів.

Кожну хвилину Youtube генерує 300 годин нового відео контенту, база його користувачів неймовірно велика нарахує до декількох мільярдів користувачів одночасно по всьому світі. Для обробки такої кількості даних компанія використовує можливості AI, щоб покращувати сервіс і спрощувати деякі операції.

Його система складається з двох основних частин. Перша – це нейрона мережа, яка займається генерацією кандидатів, на основі вже переглянутих

Netflix аналізує кожний клік, свайп, затримку на іконці своїх 100 мільйонів користувачів на шляху для отримання унікальної системи деталізації переваг свого сервісу.

Також компанія вмiло використовує свій контент, вибираючи найбільш цікаві сцени з моментів фiльмiв та серiалiв, орієтуючись на користувача, для тих хто полюбляє екшен, будуть пропонуватись відповідні кадри навіть з фiльмiв де це не є головним лейтмотивом.



Рисунок 2.9 – Приклад рекомендації, створеної сервісом Netflix

Розглянемо також більш вузький приклад сфери використання:

LinkedIn

LinkedIn – бізнес-орієнтована соціальна мережа, для пошуку і встановлення ділових контактів. Має базу в понад 774 мільйони користувачів з 150 галузей по всьому світі.

Пропонує своїм користувачам рекомендації щодо знайомих людей, відповідні до його спеціальності компанії так вакансії. Для формування рекомендацій використовується колаборативна система фільтрації, яка була заснована на технології Apache Hadoop.

Apache Hadoop – вільна програмна платформа розподіленого зберігання.

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 19 |

Ці проблеми можна вирішити різними підходами, та це забирає велику кількість часу.

В моїй системі підібравши правильні параметри можна пришвидшити математичні операції. Використання сучасних API та сервісу Kaggle вирішує проблему створення моделі, оскільки завжди можна отримувати актуальні данні від сервісу та скористуватися вже зібраними моделями. Вже зібрані моделі завжди можна об'єднати для збільшення загальної моделі

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для програмної реалізації рекомендаційної системи. Була обрана мова програмування Python і інтерактивне середовище розробки Visual Studio Code.

Python – мова програмування, націлена на збільшення продуктивності. Головними її перевагами є можливість написання проекту будь-якої складності і направленості. Маючи при цьому низький поріг для входження. За рахунок простоти написання та читання коду, є однією із найзручніших сучасних мов програмування.

Популярність цієї мови програмування яскраво розкривається в її ком'юніті, яке породжує велику кількість модулів, що дозволяє скоротити час на розробку та підледіти цікаві алгоритми вирішення різноманітних задач.

Та при всій простоті та чудовості є недоліки, це порівняно невисока швидкість виконання Python програми. Але цей недолік легко вирішити за допомогою модулів і підбору нових алгоритмів, що пришвидшують компіляцію та час виконання задач.

Розглянемо деякі модулі використання в цьому проекті.

Модуль Telebot – оболонка над запитами до TelegramBotApi, використовується для спрощення і мінімізації написання коду.

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 21 |

В класі TeleBot розташовані всі методи API, щоб слідувати загальним угодам про імена Python, вони перейменовані. Наприклад: sendMessage - send_message. editMessageText edit - _message_text.

Модуль Request – бібліотека Python, яка виконує HTTP запити. Починаючи від параметрів в URL-адресах до відправки користувацьких заголовків і перевірки SSL.

TelegramBotApi являє собою HTTP інтерфейс для роботи з ботами в Telegram. Кожен бот – це спеціально створений акаунт для автоматичного оброблення та відправлення повідомлень.

Документація TelegramBotApi виділяє два максимально протилежних способи отримання оновлень:

- 1) Періодичні запити;
- 2) Встановлення веб-хуків.

Вхідні оновлення зберігаються до того моменту, поки сервер не обробить його, але не більше ніж 24 години. У відповідь отримує об'єкт Update, який серіалізовано в JSON, незалежно від способу отримання оновлень.

Перший і найбільш простий варіант полягає в періодичному опитуванні серверів Telegram на наявність нової інформації. Відкривається з'єднання на нетривалий час і всі оновлення відразу відправляються боту, все це здійснюється через зв'язок Long Polling. Цей спосіб є простим, але не дуже надійним.

Веб-хуки працюють трохи інакше: якщо в чат проходить повідомлення, то Telegram самостійно повідомляє про це, в цьому і полягає робота веб-хуку. Тепер відпадає необхідність періодично опитувати сервери, тим самим, зникає причина помилок пошукових роботів. Однак за цю можливість, потрібно платити необхідністю використання повноцінного веб-сервера на той пристрій, на якому і планується запускати пошуковий бот.

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | <i>ВКРМ -122.22.0011.00.00.ПЗ</i> | Арк. |
| Вим | Арк | № докум. | | Підпис | | 22 |

Pandas, як вказано на їх офіційному сайті це швидкий, потужний, гнучкий і простий у використанні інструмент аналізу та маніпулювання даними з відкритим кодом, побудований на основі мови програмування Python.

Основними рисами цього модулю є:

- 1) Проста обробка відсутніх даних у форматі з плаваючою комою,
- 2) Мінливість розміру (вставлення і видалення) елементів з DataFrame,
- 3) Автоматичне та явне вирівнювання даних,
- 4) Інтелектуальна нарізка на основі етикеток,
- 5) Інтуїтивне об'єднання даних в набори,
- 6) Надійні інструменти ІО для завантаження і обробки плоских файлів таких як .csv .

Spotipy – це легка бібліотека Python для веб-API Spotify. За допомогою якого отримуємо повний доступ до всіх музичних даних, наданих платформою Spotify.

Numpy є основним пакетом для обчислення масивів за допомогою Python. Використовується як ефективний багатовимірний контейнер загальних даних. Довільні типи даних можуть бути визначені. Це дозволяє numpy легко та швидко інтегруватися з широкою різноманітністю баз даних.

Visual Studio Code – це зрозумілий, зручний, багатофункціональний редактор коду, який має кросплатформову модель.

IDE підтримує всі популярні мови програмування та велике ком'юніті, які розробляють даткові модулі та візуальні надбудови, що покращує користувацький досвід роботи.

Включає в себе вбудований «дебагер», для зручності розробки і відлагодження коду на кожному етапі. Також має підтримку внутрішніх юніт тестів, для перевірки працездатності розробленого ПЗ. Підтримка інструментів для роботи з Git-репозиторіями, підкреслення синтаксису за стилем обраним користувачем, засоби для рефакторингу та IntelliSense (технологія автодоповнення, яка підкаже конструкцію за першими літерами).

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 23 |

Має зручний інтерфейс з візуалізацією типів файлів, які використовуються в проєкті, має надбудови терміналу для внутрішньої роботи в системі та слідкування за виконанням коду. Підтримка широкої лінійки мов програмування дозволяє комбінувати мови в одному проєкті.

Мови що підтримуються:

- 1) JavaScript;
- 2) TypeScript;
- 3) Node.js
- 4) C++;
- 5) C#;
- 6) Java;
- 7) PHP;
- 8) Go;
- 9) Python;
- 10) .NET;
- 11) Unity.

Telegram – кросплатформовий месенджер, який має гнучкий функціонал для створення ботів в своїй структурі. З повноцінною документацією і відкритим кодом. Дозволяє розробити різні структури і проєкти на базі ботів (віртуальних користувачів).

Має гнучку систему налаштування, кожен користувач може обрати стиль і розміщення даних в месенджері, що робить з нього чудовий інструмент, як для спілкування так і виконання різноманітних задач.

Kaggle – платформа змагань з аналітики так передбачувального моделювання. Містить в собі собарні моделі для прогнозування та опису даних, що пропонує компанія або користувачі.

Дуже зручний інструмент для отримання датасетів, які не так легко зібрати, через певні обмеження платформ з великими користувацькими базами.

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 24 |

Сервіс ділиться даними в доступному, непатентованому форматі, якщо це можливо. Платформа підтримує відкриті доступні формати даних з якими можна працювати використовуючи різноманітні інструменти.

Підтримувані типи файлів:

- 1) CSV;
- 2) JSON;
- 3) BigQuery;
- 4) PNG;
- 5) HDF5.

На базі своєї платформи підтримують блокноти (схоже на Jupiter або Google Colab).

2.3 Розгорнута постановка завдання

Проаналізувавши поставлену задачу про програмну реалізацію рекомендаційної системи, я дійшов до висновку, що цікавим в розробці буде Telegram бот. Який буде за допомогою розробленої рекомендаційної системи видавати користувачеві фільми та подібні рекомендації на базі обраної ним музичної композиції.

Тому для реалізації потрібно виконати наступні завдання:

- 1) Провести дослідження актуальних систем та наукових досліджень по цій тематиці проаналізувавши з якими проблема можна зіткнутися при створенні такої системи і можливі варіанти вирішення цих проблем;
- 2) Визначити переваги та недоліки існуючих систем, для обрання напрямку проекту. Аналіз існуючих системи дає чудову статистику за різними напрямки в яких можливе використання подібних систем;
- 3) Протестувати та обрати інструменти розробки. Підбір дійсно корисних інструментів вирішує базові проблеми та забезпечує

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 25 |

безперебійну розробку. Чим краще буде обране програмне середовище, тим простіше буде вирішувати і шукати додаткову інформацію за питаннями які постійно виникають в такій сфері.;

- 4) Розробити структурну та функціональну схему проекту які зможуть підтвердити роботу проекту і пришвидшити його розробку;
- 5) Розробити алгоритм роботи ПЗ. Найголовніша частина будь-якого проекту, оскільки на його базі стає зрозуміло який функціонал можливо розробити виходячи з обмежень системи на якій він буде будуватися. Та також буде корисним при розширенні проекту та залучені нових фахівців;
- 6) Зібрати датасети інформації з якою буде працювати система, адже без них система не зможе проводити рекомендації.
- 7) Розробити просту систему користувацького спілкування з системою, щоб у користувачів не виникали питання як взаємодіяти з готовою системою

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Рекомендаційна система, яка буде базуватись в чат-боті, повинна отримувати і реагувати на запити користувача у месенджері Telegram. Чат-бот повинен мати зручний інтерфейс без зайвих надбудов. З першого старту користувач повинен одразу зрозуміти, яким чином він отримає інформацію. Створюювана система оснащена новітніми сервісами з яких і буде черпати інформацію. З аналізу задач та огляду методів програмного забезпечення можна сформулювати вимоги до системи:

- 1) Оброблювати отриману інформацію (різного типу);
- 2) Вести діалог з користувачем.

Основою рекомендаційної системи була обрана система заснована на контентній фільтрації (content-based filtering). Вона використовує схожість у продуктах, службах або функціях вмісту а також інформацію, накопичену про користувача для надання рекомендацій.

За допомогою цієї системи ми будемо намагатися передбачити, що зможе сподобатися користувачеві, виходячи з його активності.

В нашому випадку об'єктами метаданих буде дата випуску, жанр або лейб, з яким була опублікована пісня. Також ми отримуємо абстрактні музичні особливості з самого аудіо, а саме: ключ, темп та абстрактні математичні дані MFCC.

Для отримання аудіоданих використаємо API Spotify. Отримавши набір даних нам потрібно його налаштувати. Для цього застосуємо валентність та збудження для рекомендацій з використанням векторних відстаней.

```
genres = spotify.recommendation_genre_seeds()
```

```
data_dict = {
```

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 27 |

```

    "id": [],
    "genre": [],
    "track_name": [],
    "artist_name": [],
    "valence": [],
    "energy": []
  }

```

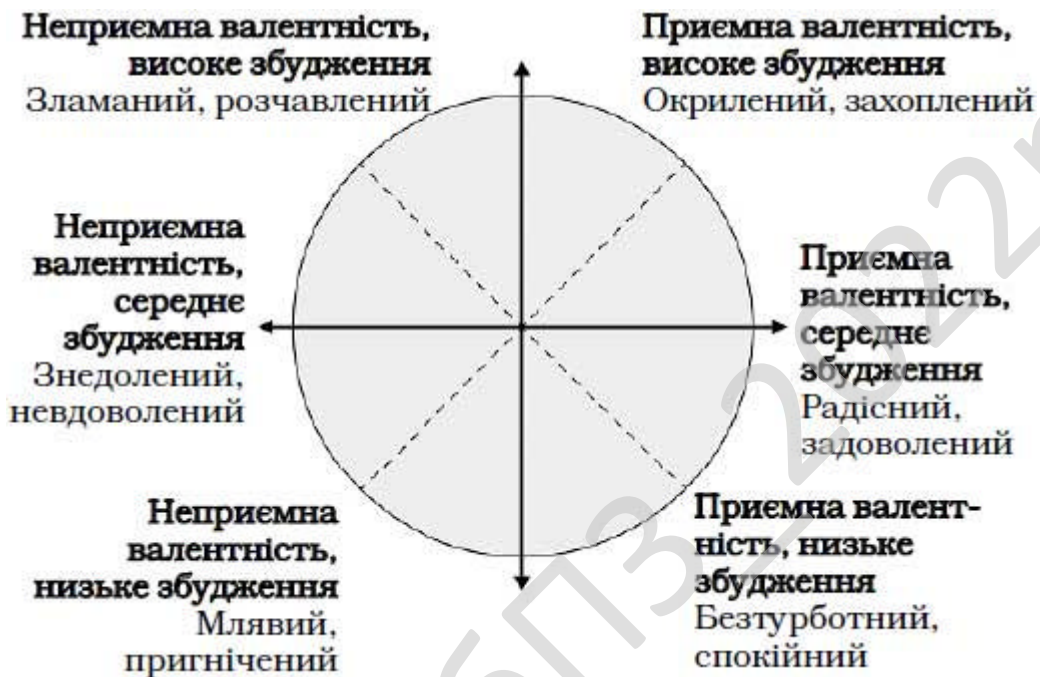


Рисунок 3.1 – площина «валентність - збудження»

Площина «Валентність збудження» одна із основних психологічних моделей структури емоцій. Модель описує, скільки існує незалежних вимірів емоцій. Складаючі з двох компонентів - збудження (активність або млявість) та валентності (позитив або негатив).

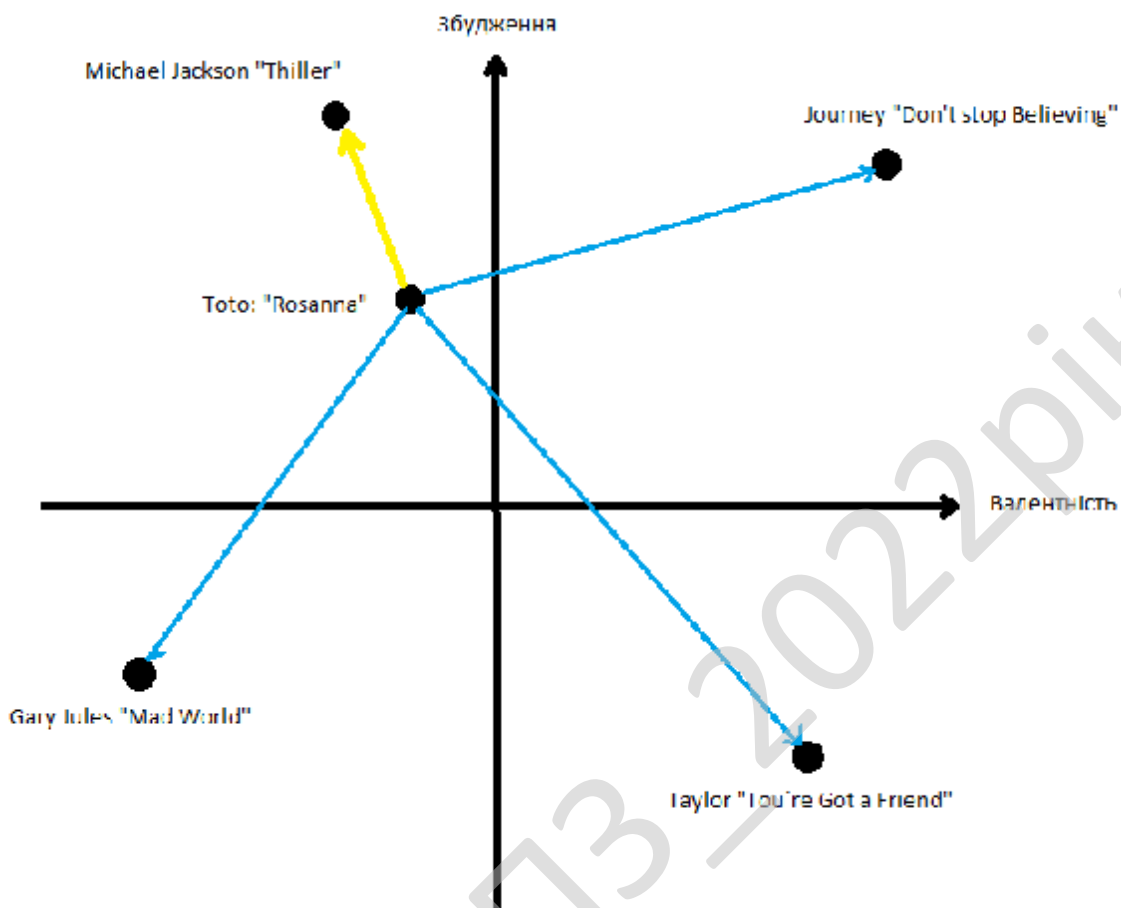


Рисунок 3.2 – приклад використання «Валентності і збудження»

На зображенні (Рисунок 3.2) зображено приклад використання «Валентності і збудження» на якому помітно, що емоційний профіль композиції «Thriller» Майкла Джексона наближений до композиції «Rosanna» у виконанні колективу Toto. Тому композиція «Thriller» підходить для рекомендації невідмінну від інших представлених на зображенні.

Математично лінії, що з'єднують всі композиції є векторами і ми можемо виміряти їх довжину. Довжина двовимірного вектора (a,b) визначається як $\sqrt{a^2+b^2}$. Уявимо що пісня має 0.7 «валентності» і 1 «енергію», отримаємо координати (0.7, 1). Тоді відстань від початку координат (0,0) дорівнюватиме $\sqrt{0.7^2+1^2}$, тобто становитиме 1,22. Далі ми знаходимо всі вектори, що з'єднують задану колію з іншими треками, застосувати формулу і взяти

найменшу вектор з найменшою довжиною. Виходячи з спостережень отримаємо, що заснована на настрої «відстань» між двома треками $t1$ і $t2$ дорівнює нормі $(t2-t1)$.

Обчисливши відстань між вхідною доріжкою та всіма іншими піснями в наборі даних, відсортуємо контрольні доріжки від найменшої до найбільшої відстані і отримаємо n найменш віддалених треків.

```
def recommend(track_id, ref_df, spotify, n_recs = 5):
    features = spotify.track_audio_features(track_id)
    vector = np.array([features.valence, features.energy])
    ref_df["distances"] = ref_df["mood_vec"].apply(lambda x: norm(vector-
np.array(x)))
    df_sorted = ref_df.sort_values(by = "distances", ascending = True)
    df_sorted = ref_df_sorted[ref_df_sorted["id"] != track_id]
    return ref_df_sorted.iloc[:n_recs]
```

Не маючи інформації про нового користувача, система опирається на опис внутрішніх даних своєї бібліотеки. На початку роботи з користувачем проводиться опитування, з якого ми виділяємо основні критерії користувача. Прогнавши їх по базі, знаходимо найбільш схожі за описом елементи. В роботі з цією системою користувач також може надіслати посилання на пісню в сервісі Spotify, система порівняє ідентифікатор та низку параметрів пісні з існуючим датафреймом і на базі векторних підрахунків зможе підібрати певну кількість рекомендацій базуючись на отриманій інформації.

| | id | name |
|--------|------------------------|---|
| 285519 | 0B97zHGjID7nCiLFxnJqIP | Elas Fican Loucas |
| 349287 | 26254MDXFmUHClid1pInxG | Prima Iristeza |
| 556553 | 1yWpXFA9RXG0xsIS4dDerP | Lai-Lai-Lai |
| 83742 | 3JK09Qk7o9ZzL20gqYY04K | Esta Loca - Supernova 2010 Revision |
| 56043 | 4JUzi3tbIAmHi8AIGAce5H | We Can't Be Beaten |
| 242213 | 0nOqKydhOJHPGm3zd2SAej | Keep The Fire Burning - Original 12-Inch Versions |
| 531435 | 3fpVYSLnxkadjYwQPTyo4H | Laal Ghaghra |
| 535310 | 0CQEaNH5cJA2ofbcOFu6Al | Tetovaci |
| 551005 | 2OK7UOMnUDohhjJHcB6Hpy | MACHINE |
| 350399 | 4KtgjTY7cnRmWmzMm628V6 | Romance de la luna gitana y el sol poeta |
| 62385 | 0P0srEkEjzah5iM0j5mbPS | I Want a Woman |
| 224068 | 5LZjo9oExGUz1k5K4I7i7Y | סיריחובה ודקרי |
| 269158 | 4imMjh9dojhsVG0sHd5Byc | Prima Donna - 2006 Remaster |
| 301567 | 2GNsDkUXo7qnkdw1g3hMDx | The Wall |
| 534621 | 6W6T4H5854GGrEoqILosVI | Indiáni II |

Рисунок 3.3 Приклад виведення пісень схожих на введену.

Для рекомендування фільмів використовується загальнодоступний набір даних MovieLens-25M. Він складається з 62 000 фільмів з 25 млн. рейтингів і тегами від 163000 користувачів. Також використовується IMDbPY для отримання інформації про фільм з IMDb сюди входять жанри, ключові слова і посилання на фільми.

Створення моделі системи рекомендацій щодо фільму

Використаємо алгоритм обчислення подібності.

```
knn = NearestNeighbors(metric='cosine', algorithm='brute', n_neighbors=20,
n_jobs=-1)
```

```
knn.fit(csr_data)
```

Принцип роботи рекомендаційної функції:

На першому етапі перевіряємо, чи є введення імені фільму в базу даних, щоб знайти подібні фільми та відсортувати їх на основі відстані подібності та вивести задану кількість найкращих фільмів їх відстанями від вхідного фільму.

```
def get_movie_recommendation(movie_name):
    n_movies_to_reccomend = 10
    movie_list = movies[movies['title'].str.contains(movie_name)]
    if len(movie_list):
        movie_idx= movie_list.iloc[0]['movieId']
        movie_idx = final_dataset[final_dataset['movieId'] ==
movie_idx].index[0]
```

```

        distances , indices =
knn.kneighbors(csr_data[movie_idx],n_neighbors=n_movies_to_reccomend+1)
        rec_movie_indices =
sorted(list(zip(indices.squeeze().tolist(),distances.squeeze().tolist())),key=la
mbda x: x[1])[:0:-1])
        recommend_frame = []
        for val in rec_movie_indices:
            movie_idx = final_dataset.iloc[val[0]]['movieId']
            idx = movies[movies['movieId'] == movie_idx].index

recommend_frame.append({'Title':movies.iloc[idx]['title'].values[0],'Distance':v
al[1]})

        df =
pd.DataFrame(recommend_frame,index=range(1,n_movies_to_reccomend+1))
        return df
    else:
        return "No movies found. Please check your input"

```

Для тесту проведемо пошук подібних фільмів до фільму «Iron man»

```
get_movie_recommendation('Iron Man')
```

В результаті отримаємо добірку фільмів схожих за жанром та настроєм на заданий фільм.

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 32 |

| | Title | Distance |
|----|--------------------------------|----------|
| 1 | Up (2009) | 0.368857 |
| 2 | Guardians of the Galaxy (2014) | 0.368758 |
| 3 | Watchmen (2009) | 0.368558 |
| 4 | Star Trek (2009) | 0.366029 |
| 5 | Batman Begins (2005) | 0.362759 |
| 6 | Avatar (2009) | 0.310893 |
| 7 | Iron Man 2 (2010) | 0.307492 |
| 8 | WALL-E (2008) | 0.298138 |
| 9 | Dark Knight, The (2008) | 0.285835 |
| 10 | Avengers, The (2012) | 0.285319 |

Рисунок 3.4 Приклад отриманих фільмів

Розглянемо також роботу бота.

Головна функція бота – відправляти і отримувати повідомлення. І те, інше можна робити з усіма видами повідомлень (фото і відео, файли, опитування, голосові повідомлення і т.д.).

Бот може писати в приватні повідомлення лише тим, хто його активував, тобто зареєструвався в його зоні роботи. Користувач може заблокувати бота, і тоді бот не зможе з ним спілкуватись.

Боти не можуть писати один одному. Бота можна додати до групи чи каналу лише з правами адміністратора. Надавши йому права адміністратора в обраній групі ми дозволимо йому спілкуватись зі всіма членами групи.

Кожне повідомлення в Telegram має свій id. Це відноситься і до системних повідомлень (користувач доєднався, змінилась назва і т.д.).

Через Telegram API боти можуть отримувати за запитом повідомлення в будь-якому чаті за їх id.

ID повідомлень в особистих повідомленнях і звичайних групах працюють по іншому. Там нумерація наскрізна: id повідомлень унікальне для кожного користувача. Тобто перше повідомлення користувача нумеруються як 1, друге як 2 і т.д.

Для взаємодії з ботом можна включити його інлайн режим. Інлайн-режим (Inline mode) - це спеціальний режим роботи бота, за допомогою якого користувач може використовувати бота в усіх чатах.

Виглядає це так: користувач вводить юзернейм бота в поле для вводу. Після посилання на бота можна зробити запит до 256 символів. З'явиться інтерактивне меню з результатами. Вибираючи результат, користувач відправляє повідомлення боту.

Активація цього режиму проводиться через вбудований в Telegram менеджер ботів BotFather, в налаштування якого можна обрати плейсхолдер взамін стандартного тексту. Результати відображаються в двох варіантах: сіткою або вертикальним списком.

Коли користувач викликає інлайн-режим, бот не може отримати жодної інформації про контекст, окрім даних користувача.

3.2 Розробка структурної схеми

Структурна схема (Рис 3.5) розбита на три блоки, розглянемо їх структуру знизу вгору.

Розглянемо перший блок роботи основного модуля. Він включає в себе елементи спілкування з системою у вигляді чат боту. В цьому елементі користувач вводить запити та отримує результати рекомендації на основі його вводу. Також поряд з ним реалізована взаємодія зі сторонніми API, які використовуються в проекті, там описана загальна логіка обробки зв'язку з

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 34 |

сервісами, реєстрація в них на основі токенів та секретних ключів. Так само розглянемо блок внутрішньої реалізації, де розміщується реалізація рекомендаційної системи для фільмів та музики, обробка датафреймів, їх структуризація та операції по виділенню і порівнянню елементів що вниз зберігаються.

Середній блок роботи основного модуля. Цей блок містить реалізацію логіки керування системою та побудови запитів на сервер, де знаходиться. Являє собою ключовий елемент у зв'язуванні основних функцій ПЗ та серверу, який обробляє та повертає результати. В схемі зображено два виклики чат боту, оскільки реалізовано два варіанти його виклику: через основний чат з ботом та через виклик його функціоналу через кодове ім'я в будь-якому іншому чаті.

В ньому реалізовані елементи комутації та умови спрацювання системи.

Верхній блок відповідає за зв'язок, сервер Telegram виступає у ролі організатора зв'язку. Він обробляє інформацію, яку вводить користувач, передає її основному (середньому) блоку і очікує, коли отримає зворотне повідомлення, щоб передати її користувачеві.

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | <i>ВКРМ -122.22.0011.00.00.ПЗ</i> | Арк. |
| Вим | Арк | № докум. | | Підпис | | 35 |

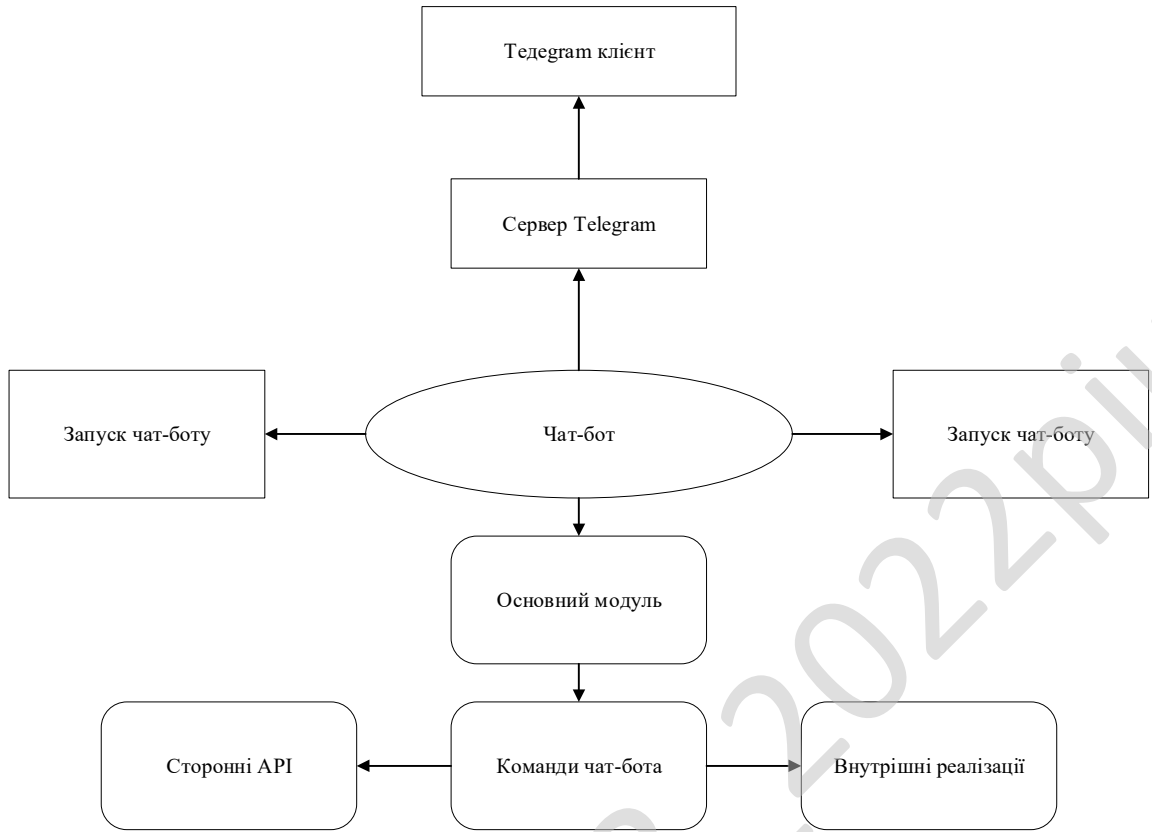


Рисунок 3.5 - Структурна схема системи.

3.3 Розробка функціональної схеми

Розробка функціональної схеми (рис 3.7) розбита на блоки взаємодії.

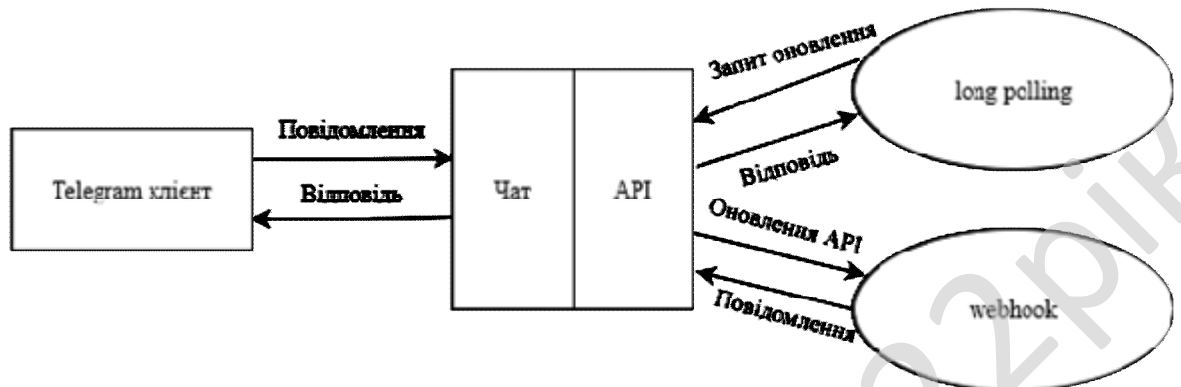


Рисунок 3.7 - Функціональна схема

Першим блоком виступає клієнт Telegram, який може бути встановлений на будь-якому пристрої яким користується клієнт. Або це може бути його веб версія, що збільшує кількість можливостей у користувача, скористатися нашою системою.

Другий блок описує в собі взаємодію користувача з ботом, в цьому блоці описується взаємодія внутрішніх модулів та сторонніх сервісів в отриманні інформації та виведенні результату. Наприклад користувач запитав у бота, чи може він знайти подібну пісню до тієї, що надіслав користувач. Бот отримує команду знайти подібні пісні, отримує посилання на пісню, виділяє з нього основні дані, які далі передає рекомендаційній системі, вона в свою чергу звертається до API Spotify, отримує дані про цей трек та потім порівнює його зі своїм набором даних за низкою показників, які використовуються в цьому проекті. Далі користувач отримує відповідь, в якій містяться прямі посилання на ці пісні в сервісі Spotify.

Останній блок - взаємодія бота з сервером та API Telegram. Він складається з методів отримання і надсилання повідомлень через Telegram. Для цього ми використовуємо декілька механізмів (технологій).

Long Polling – технологія довгих запитів. Сервер отримує запит, але не поспішає на нього відповісти. Його відповідь буде проведена, після того як відбудеться якась подія на сервері, надійде нове вхідне повідомлення або закінчиться заданий час очікування запиту.

Це зручний механізм, для систем з затримкою на виконання підрахунків. Він чекає поки запит буде оброблений і повертається з відповіддю, але це може займати багато часу. В результаті помилки клієнт отримає повідомлення «заглушку», яка спрацює коли закінчиться дозволений час запиту.

Webhook – механізм сповіщення системи про події. Він вирішує недоліки постійних запитів, які можуть не працювати, якщо на сервері варто робити затримку на отримання нових повідомлень. Коли відбувається подія Webhook запитує адресу з параметрами і передає її в тілі POST запиту JSON де найчастіше вже будуть нові або оновлені дані.

API (Application Programming Interface) – набір готових класів, функцій і структур, що дозволяють швидко обмінюватись або тільки приймати інформацію напряду з серверу потрібного нам сервісу. Зазвичай реалізуються самими сервісами, для підтримки свого продукту і взаємодії зі сторонніми розробниками.

3.4 Розробка діаграми процесів

Діаграма процесів системи (рис 3.8) зображає взаємодію процесів системи та порядок дій, що відбувається під час роботи системи. Початком роботи системи вважаємо запит користувача, адже без нього система не почне працювати і не матиме даних для аналізу. Кожний елемент системи матиме зв'язок між собою, щоб проаналізувати запит користувача та вивести йому рекомендації.

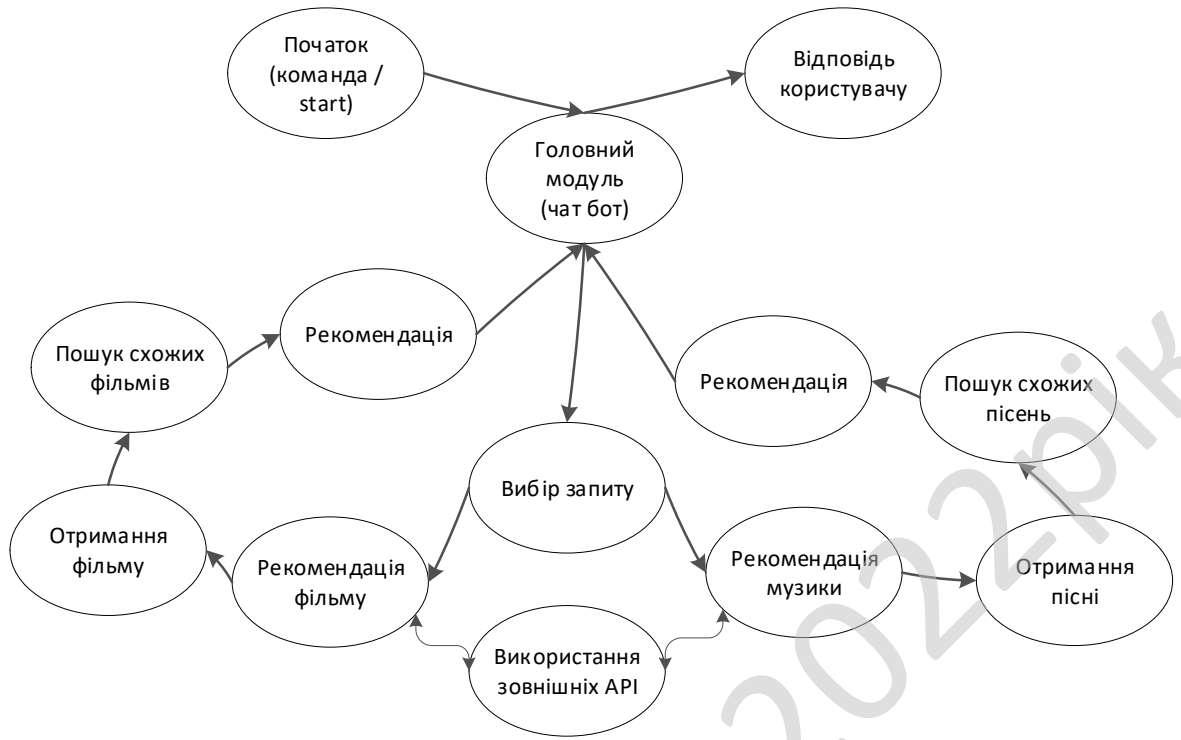


Рисунок 3.8 - Діаграма процесів

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ ТА ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

Перейдемо до розробки алгоритмів системи та програмування рішення поставленої задачі.

4.1 Розробка блок-схем та опис алгоритмів функціонування системи

При розробці блок-схем були виділені декілька основних стадій роботи програми, а саме:

1) Ініціалізація

Первинна стадія роботи боту, цей етап включає в себе всі етапи налаштувань підключення та першого запиту користувача.

2) Блок роботи основної програми

Обробка запитів користувача, підключення зовнішніх сервісів за допомогою API, обробка датафреймів та принципи побудови рекомендацій для користувача.

3) Завершення роботи

Користувач отримав рекомендацію і пішов її реалізовувати, тобто, переглядати фільм або слухати підбірку пісень виведених системою на базі його запиту. Система далі залишається в стані очікування наступних запитів, або завершення діалогу командою стоп.

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 40 |

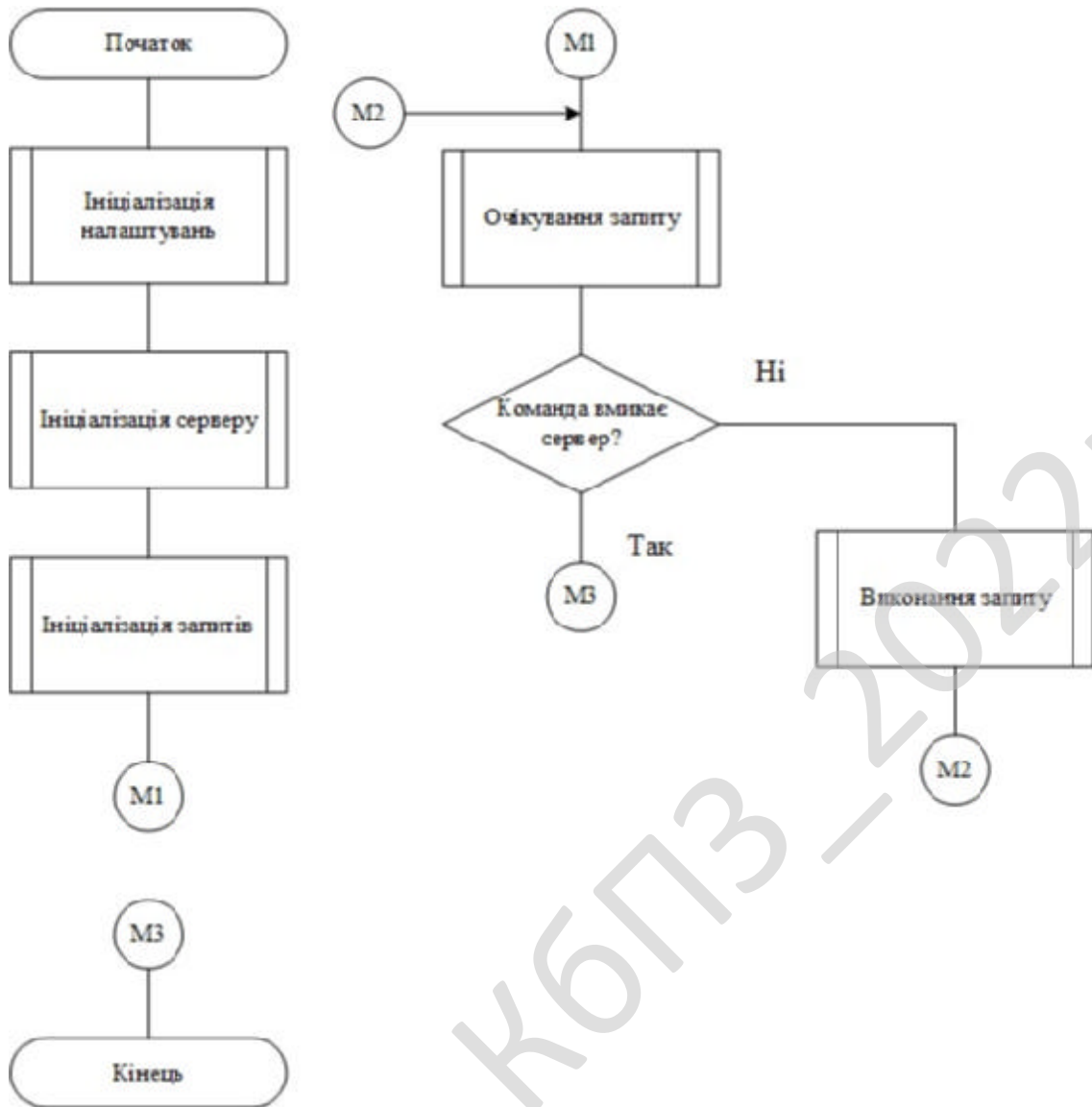


Рисунок 4.1 Блок схема логіки запуску та виконання запиту

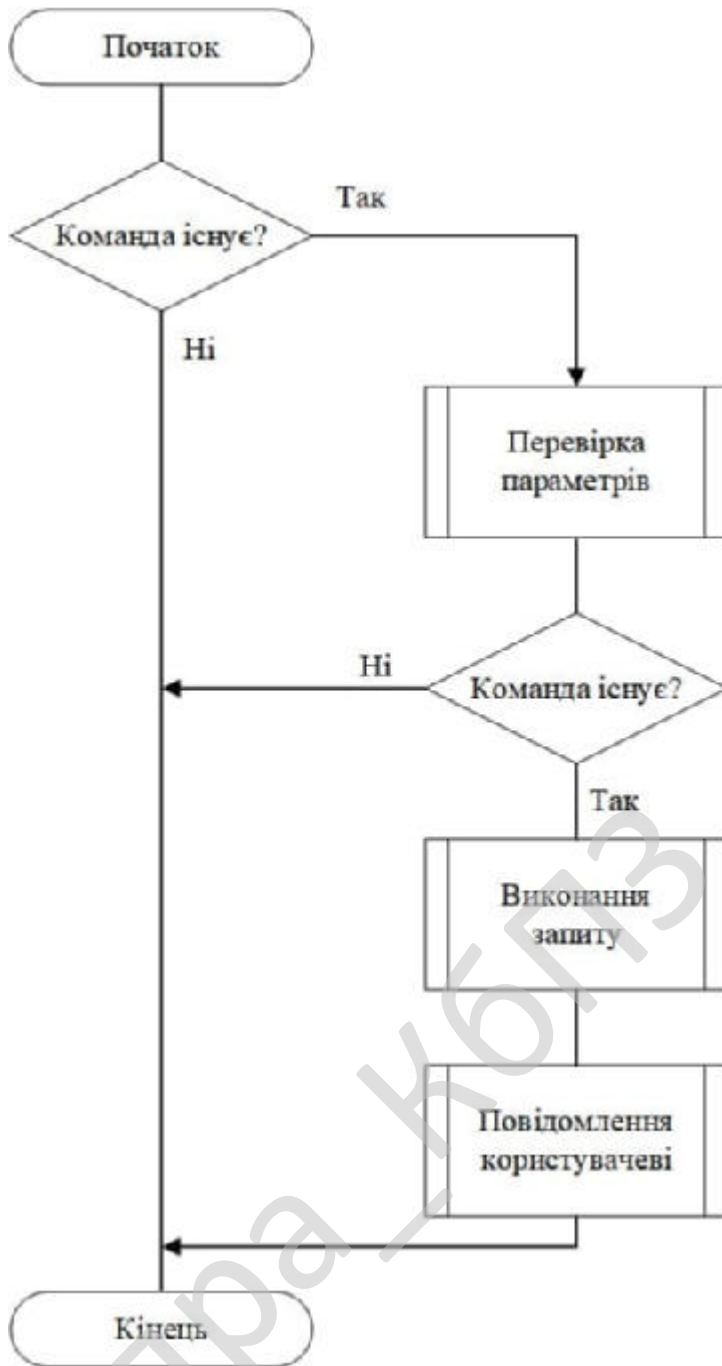


Рисунок 4.2 – Блок-схема обробки команд

Розглянемо алгоритм роботи основної програми та звернень до серверу. Всі команди перевіряються на існування. Якщо команда пуста, то буде відповідь, що команда недійсна. Кожен запит має певні параметри, які передаються до інших модулів. Якщо параметри не проходять перевірку, система повертає повідомлення про відмову і запропонує перевірити введені дані. Якщо виконати команду з невірними параметрами, є можливість зламати

Розглянемо алгоритм роботи модуля рекомендації, що був представлений на блок-схемі (Рисунок 4.3). За допомогою схеми ми можемо детально розглянути роботу цієї частини програмного забезпечення, а саме нашої рекомендаційної системи.

Розглянемо роботу системи в комплексі з попередніми схемами (Рисунок 4.1 та 4.2). Користувач, своєю командою /start запускає роботу чат-боту. Ця команда формується ще на реєстрації самого бота в системі Telegram через BotFather, вона є стандартною та виводить текст який ми задаємо. В нього можна помістити базову інформацію про систему. А саме, як називається система, функціональність системи. В нашому випадку це рекомендація музики і фільмів, яка працює по фільтрації контенту на основі вмісту.

Розглянемо сам алгоритм рекомендації. Оскільки ми нічого не знаємо про користувача, окрім його ідентифікатора в Telegram, система йому нічого порекомендувати не може. Тому була розроблена логіка, що користувач, шукає щось подібне до того, що він вже знає.

Користувач обирає команду /music і відправляє посилання на знайому йому композицію в Spotify. Система, отримавши його посилання, перевіряє чи посилання, чи воно працює. Якщо ні повертає повідомлення, що посилання неактивне, або посилання активне і система починає проводити аналіз.

Оскільки в кожному посиланні на музику в Spotify вже вказаний ідентифікатор на композицію, система виділяє його з основного повідомлення і отримує ідентифікатор, з яким вже звертається до API Spotify і отримує метадані пісні. Отримавши дані, система виділяє два параметри з метаданих, це валентність і збудження (Рисунок 3.1 зображає їх залежність). З отриманих даних формуємо вектори і переглядаємо їх в нашій збірній базі та шукаємо найближчих сусідів. Потім отриману таблицю ми обрізаємо до n заданих рекомендацій та повертаємо користувачеві у вигляді посилання на них у Spotify.

Рекомендація фільмів викликається командою /movie. Система запитає ім'я фільму. Та отримавши його почне пошук рекомендації. Блок-схема її логіки зображена нижче.

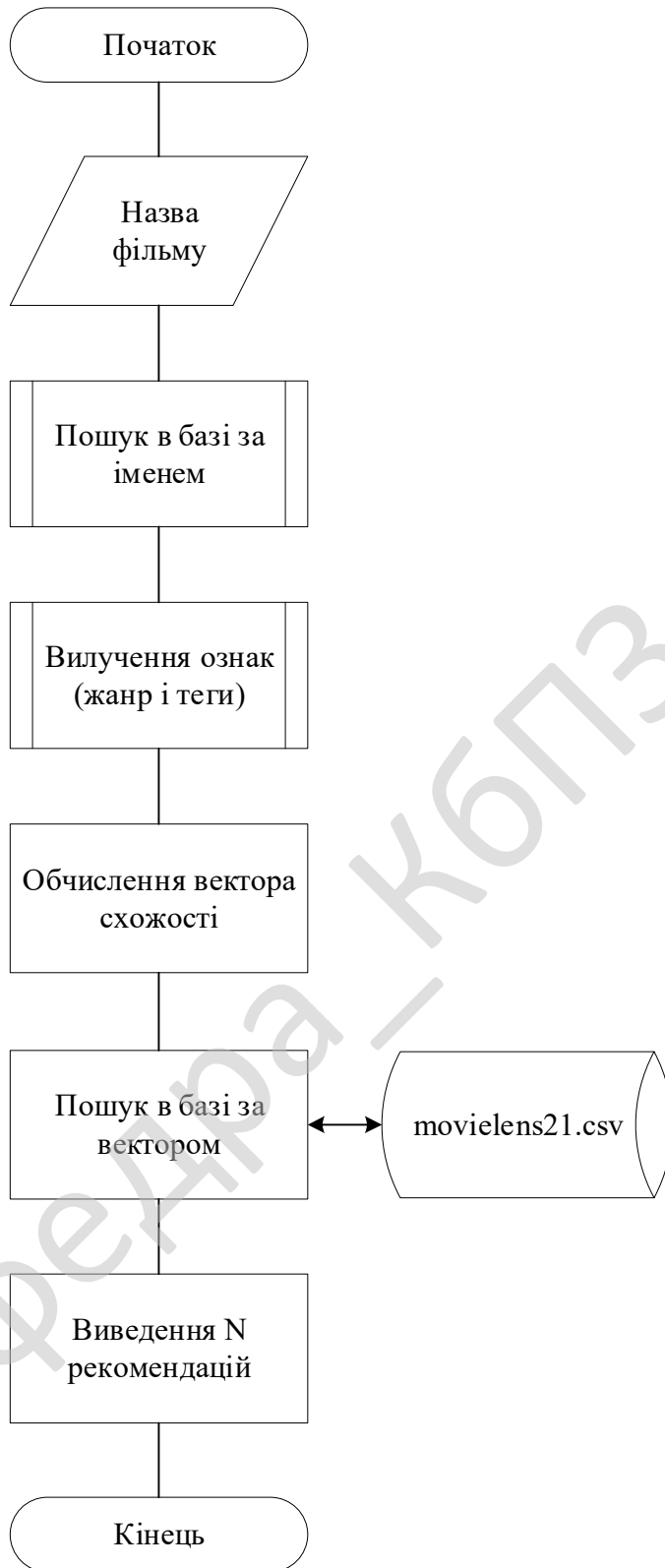


Рисунок 4.4 Блок-схема роботи рекомендації фільмів

Як можна помітити, схема роботи рекомендації фільмів і музики відрізняється лише у виглядом введення даних та базою пошуку. Оскільки обидві базуються на фільтрації контенту на основі вмісту.

4.2 Захист розробленого програмного забезпечення

Захист інформації програмного забезпечення рекомендаційної системи реалізовується провайдером на базі якого буде розміщена наша система. Оскільки система має хмарну реалізацію.

Тобто On-Cloud – програми, які вже розгорнуті на віддаленому сервері або розроблені з урахуванням такої можливості.

Основними перевагами такого рішення є те, що не потрібно мати власний сервер. Усіма технічними питанням буде займатися провайдер.

Також хмарна реалізація дає можливість швидкого автоматизованого розширення або скорочення інфраструктури, що дозволить динамічно реагувати в залежності від навантаженості та потреб системи. Користуйтеся і платіть тільки за те, що потрібно (Pay-As-You-Go).

Безперебійна робота інфраструктури і забезпечення аварійного відновлення це безумовна перевага такого рішення.

Вибір провайдера – це один із основних етапів перенесення проекту в хмару. Розглянемо фактори на які потрібно звернути увагу при виборі провайдера:

1) Відмовостійкість – це безперебійний доступ до інфраструктури та збереження наших даних у разі несправностей. Надійність дата-центру визначається міжнародною класифікацією ANSI/TIA-942.

Стандарт ранжує дата-центри за 4-ма класами – Tier I, II, III та IV. Чим вищий клас, тим більше систем продубльовано на випадок відмов обладнання.

Рекомендуються Tier III та IV.

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 46 |

2) Захист даних – це критично важливий фактор. Тому при виборі провайдера звертаємо увагу на внутрішні протоколи. Міжнародна сертифікація система безпеки – добрий знак для вибору. Звертаємо увагу на наявність сертифікату ISO/IES 27001.

Стандарт регулює політику безпеки, контроль за кваліфікацією співробітників, системи моніторингу, регулярність внутрішніх аудитів та інше.

3) Додаткові послуги. Гарний хмарний провайдер полегшує роботу. Чудовий знак, якщо додаткові послуги надаються за моделлю XaaS – «Що завгодно як сервіс». Прикладом таких послуг є налаштування захисту від DDoS-атак, налаштування VPN тунелів для безпечний підключень та самостійне керування DNS записами.

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | <i>ВКРМ -122.22.0011.00.00.ПЗ</i> | Арк. |
| Вим | Арк | № докум. | | Підпис | | 47 |

5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Першим етапом впровадження системи буде розміщення системи на базі хмарної PaaS-платформі Heroku.

Розміщення на Heroku схоже з роботою розподіленої системи контролю версій (Git). Існує три способи розгортання:

- 1) Heroku Git і Container Registry використовуючи Heroku CLI;
- 2) Підключення аккаунту Github з автоматичним розгортанням;
- 3) Підключення аккаунту Dropbox з автоматичним розгортанням

Для розміщення цього проекту був обраний перший спосіб через Heroku CLI. Розгортання системи потребує наступних дій:

- 1) Реєстрація на платформі Heroku;
- 2) Локальне встановлення Heroku CLI;
- 3) Через CMD або термінал (Windows 10/11) виконується вхід через команду Heroku login (login);
- 4) Обрати розташування нашого проекту або виконувати попередні дії вже в директорії проекту
- 5) Створюємо файл Procfile, який вказує основний файл для запуску на сервері;
- 6) Створюємо файл requirements.txt, який містить в собі всі завантажувальні компоненти проекту.
- 7) Відправити зафіксовану версію системи на віддалений хмарний сервер командою git push Heroku master

В результаті виконання послідовних команд отримаємо у вікні CMD або терміналу інформацію про стан проекту та режим доступу до нього.

Наступним кроком відкриваємо клієнт Telegram на будь-якому приладі де він встановлений в пошуку вводимо ім'я боту @Restom_bot відкриваємо

| | | | | | | |
|-----|-----|----------|--|--------|----------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 48 |

його та починаємо роботу командою /start. На цьому етапі впровадження нашої системи завершено, можна користуватися системою на всіх приладах де встановлено Telegram та є доступ до глобальної мережі.

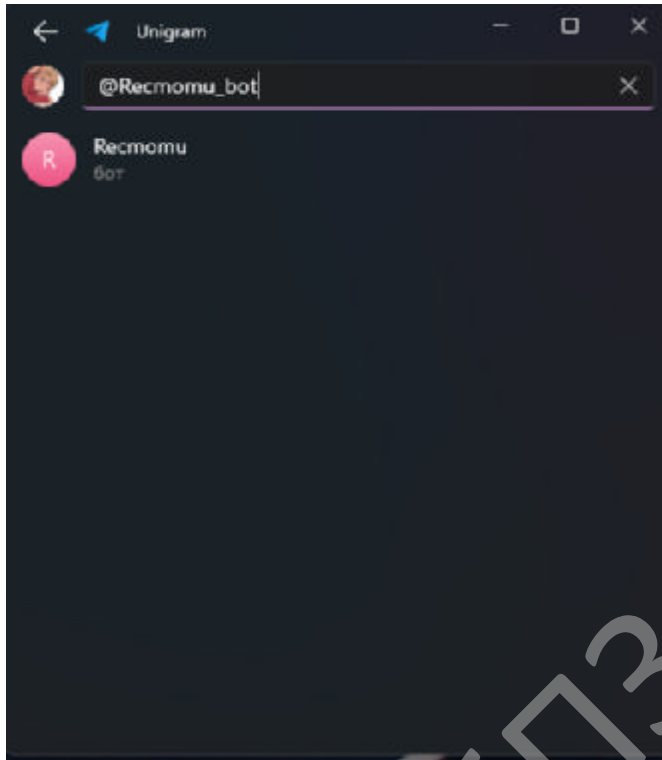


Рисунок 5.1 Пошук бота

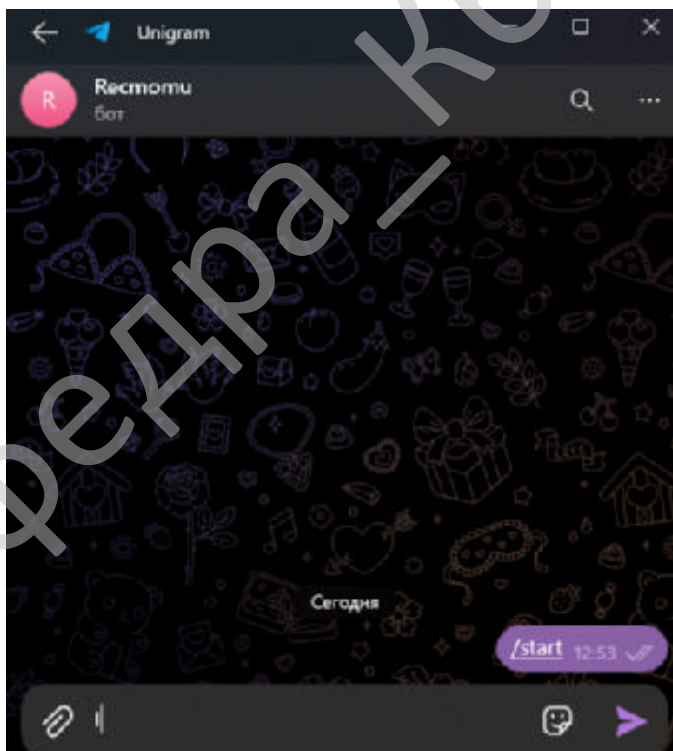


Рисунок 5.2 Початок роботи

| | | | | | | |
|-----|-----|----------|--|--------|----------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 49 |

6 НАУКОВА НОВИЗНА

Метою роботи було дослідження та програмна реалізація хмарної рекомендаційної системи обрання контенту з застосуванням технологій штучного інтелекту.

В результаті цієї роботи було досліджено та протестовано сучасні рекомендаційні системи на практиці. Розглянуто логіку їх роботи і як-вона динамічність реакції на дії користувача.

Вся отримана інформація була використана для побудови своєї рекомендаційної системи яка базується на фільтрації контенту на основі вмісту.

Для створення проекту за основу був обраний метод фільтрації контенту на основі вмісту, оскільки він позбавлений головних проблем своїх конкурентів. Цей метод дозволяє опрацьовувати великі масиви даних за різними параметрами. Саме ця його особливість підштовхнула на розробку хмарної рекомендаційної системи.

Система базується на платформі сучасного кросплатформового месенджера Telegram, що дозволить охопити велику кількість користувачів та надати їм швидкий доступ до системи.

Аналіз розподілення медіа-файлів за їх настройми на основі площини «збудження та валентності» дозволяє знайти подібне в різних стилях та жанрах, що дозволить користувачеві отримати новий досвід в його медіапросторі. Обравши за початкові координати медіа-файл обраний користувачем, система буде відштовхуватись від смаку користувача та його настрою, це облегшує підрахунки, оскільки не потрібно передбачати настрої користувача. Такий підхід дозволить підтримувати увагу користувача та впливати на його настрої.

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | <i>ВКРМ -122.22.0011.00.00.ПЗ</i> | Арк. |
| Вим | Арк | № докум. | | Підпис | | 50 |

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми магістерської роботи

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи рекомендації даних.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

| Показники | Позначення | Характеристика або величина |
|--|------------|-----------------------------|
| 1 | 2 | 3 |
| 1. Кількість розроблених програм період, шт. | N | 1 |
| 2. Кількість екземплярів програм, шт. | Ne | 45 |
| 3. Запланований термін розробки, днів | Fpq | 24 (1 місяць) |
| 4. Група задачі підсистеми управління (1-6) | – | 1 |
| 5. Ступінь новизни задачі (А, Б, В, Г) | – | Г |
| 6. Складність алгоритму (1, 2, 3) | – | 2 |

Продовження таблиці 7.1

| 1 | 2 | 3 |
|--|---|---|
| 7. Кількість макетів вхідної інформації | – | 3 |
| 8. Кількість форм вихідної інформації. | – | 4 |
| 9. Мова програмування (1-6) | – | 1 |
| 10. Попередній досвід (1-6) | – | 4 |
| 11. Гнучкість проекту ПП (1-6) | – | 2 |
| 12. Детальність проекту ПП (1-6) | – | 3 |
| 13. Рівень спрацьованості колективу (1-6) | – | 3 |
| 14. Ступінь вимірності процесів (1-6) | – | 3 |
| 15. Необхідна надійність програмного забезпечення (1-6) | – | 5 |
| 16. Розмір бази даних (порівняно з розміром програми) (1-6) | – | 1 |
| 17. Складність кінцевого програмного продукту (1-6) | – | 3 |
| 18. Необхідний рівень забезпечення повторного використання (1-6) | – | 3 |
| 19. Документованість відповідно до планованого життєвого циклу (1-6) | – | 3 |
| 20. Вимоги до швидкодії ПП (1-6) | – | 4 |
| 21. Обмеження на розміри основного сховища даних (1-6) | – | 2 |
| 22. Різноманітність використовуваних обчислювальних платформ (1-6) | – | 1 |
| 23. Професійний рівень аналітиків (1-6) | – | 4 |
| 24. Професійний рівень програмістів (1-6) | – | 2 |
| 25. Постійність складу команди розробників (1-6) | – | 4 |
| 26. Досвід розробки додатків (1-6) | – | 1 |
| 27. Досвід роботи з обчислювальною платформою (1-6) | – | 2 |

Продовження таблиці 7.1

| 1 | 2 | 3 |
|---|-----|--------|
| 28. Досвід роботи з мовою і інструментами середовища розробки (1-6) | – | 1 |
| 29. Досвід роботи з програмними інструментами розробки (1-6) | – | 2 |
| 30. Розробка ПЗ для декількох серверів одночасно (1-6) | – | 1 |
| 31. Вимоги до дотримання встановленого графіка робіт (1-6) | – | 1 |
| 32. Вартість ПЗ у розробника (НМА), грн. | – | 250000 |
| 33. Норматив додаткової зарплати, % : | Нд | 10 |
| 34. Норматив відрахувань у соціальні фонди, % | Нс | 22 |
| 35. Норматив загальногосподарських витрат, % | Нг | 15 |
| 36. Норматив витрат на освоєння нових мов програмування, % | Нп | 15 |
| 37. Рівень рентабельності програмної продукції, % | Ре | 50 |
| 38. Ставка податку на додану вартість, % | Ндв | 20 |

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де: A – коефіцієнт Боема, $A = 2,45$;

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 53 |

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(1,62 + 4,86 + 2,53 + 2,97 + 2,73) = 1,024.$$

$$T_{ном} = 2,45 \cdot 0,6^{1,024} = 1,4 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де: $\prod V_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 1,4 \cdot (1,39 \cdot 0,93 \cdot 1 \cdot 1 \cdot 1 \cdot 1,11 \cdot 1 \cdot 0,87 \cdot 0,83 \cdot 1,16 \cdot 0,92 \cdot 1,22 \cdot 1,12 \cdot 1,22 \cdot 1,12 \cdot 1,25 \cdot 1,29) = 3,4 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкість програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33+0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4);

S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 2,75 \cdot 3,4^{0,33+0,2(1,024-1,01)} \cdot 82 = 148 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 54 |

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

| Стадії розробки | Трудомісткість за типовими нормами та розрахунками | |
|-------------------|--|-----------|
| | Величина, люд/дні | Підстава |
| Технічне завдання | 4 | Д5 |
| Ескізний проект | 5 | Д6 |
| Технічний проект | 9 | Д7 |
| Робочий проект | 148 | Ф 7.1-7.4 |
| Впровадження | 13 | Д13 |
| Всього | 179 | – |

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$\mathcal{U} = \frac{T_{nz}N}{F_{pq} - H_{se}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

T_{nz} – трудомісткість розробки програмного забезпечення люд-дні.

$$\mathcal{U} = \frac{179 \cdot 1}{24 \cdot 3} = 8,5 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

| Найменування обладнання | Профілактичне обслуговування | | | |
|--------------------------------------|------------------------------|----------------------|--------------------|---------------------|
| | Кількість хв. на один. обл. | Кількість обладнання | Затрати часу в хв. | Затрати часу в год. |
| Системний блок ПК | 385 | 15 | 5775 | 96 |
| Монітор | 160 | 15 | 2400 | 40 |
| Клавіатура | 140 | 15 | 2100 | 35 |
| Маніпулятор «мишка» | 30 | 15 | 450 | 7 |
| Принтер матричний | 185 | 2 | 370 | 6 |
| Принтер лазерний | 355 | 1 | 355 | 6 |
| Принтер струминний | 300 | 3 | 900 | 15 |
| Сканер | 155 | 2 | 310 | 5 |
| Концентратор–маршрутизатор | 155 | 3 | 465 | 8 |
| Кабельні господарства ЛВС на 1 м. п. | 2,5 | 120 | 300 | 5 |
| Кабельне господарство електромережі | 48 | 60 | 2880 | 48 |
| Копіювальний апарат | 285 | 2 | 570 | 9 |
| Усього за рік: | | | 3 _ч | 280 |

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2}, \quad (7.6)$$

$$\Phi_{op}^c = \frac{280 \cdot 1}{1,2} = 233,3 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора
ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 233,3 / (24 \cdot 8) = 1,4 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

| Посада | Вид роботи | Час | К-ть штатних одиниць |
|--|--|-----|----------------------|
| Адміністратор загальної мережі, аналітик | Адміністрування локальної мережі, VPN, PPPoE, Wi-Fi, маршрутизатора Cisco | 2 | 0.5 |
| | Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS) | 0.5 | |
| | Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ | 0.5 | |
| | Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет | 1 | |
| Всього | | 4 | |

Продовження таблиці 7.4

| Посада | Вид роботи | Час | К-ть штатних одиниць |
|---------------------|---|-----|----------------------|
| Продакт-менеджер | Презентації нової продукції, пошук каналів збуту | 2 | 0.5 |
| | Підтримка постійних клієнтів | 1 | |
| | Оформлення договорів, ведення тендерів | 0.5 | |
| | Контроль взаєморозрахунків з постачальниками | 0.5 | |
| Всього | | 4 | |
| Інженер верстальник | Розробка та верстка макетів рекламної продукції та технічної документації | 1 | 0.5 |
| | Верстка друкованих видань | 1 | |
| | Додрукова підготовка макетів | 1 | |
| | Розміщення графіки і контенту на Інтернет сторінках | 1 | |
| Всього | | 4 | |

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

| Посада | Кількість ставок | Середньомісячний оклад, грн. | Всього за період розробки, грн. |
|---------------------------|------------------|------------------------------|---------------------------------|
| Керівник (ІТ-менеджер) | 1 | 15000 | 15000 |
| Продакт-менеджер | 0.5 | 10000 | 5000 |
| Інженер-програміст | 8.5 | 12000 | 102000 |
| Інженер-електронщик | 1 | 8500 | 8500 |
| Інженер-системотехнік | 0,25 | 8500 | 2125 |
| Адміністратор мережі | 0.5 | 9000 | 4500 |
| Системний програміст | 0,25 | 9000 | 2250 |
| Всього за період розробки | $R_{cn} = 12$ | - | $\Phi_{роб} = 139375$ |

Розрахуємо середньоденну зарплату одного виконавця:

$$Z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$Z_{cd} = \frac{139375}{12 \cdot 24} = 484 \text{ грн}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

$$B_{yd} = R_{cn}^1 S_y C_{пл}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 12 робочих місць;

S_y – питома площа на одне робоче місце, m^2 ;

$C_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 250...1600 у.о./ m^2 . Враховуючи, що курс у.о. складає 37 приймаємо для розрахунку вартість одного метра квадратного рівною 8500 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{yd} = 12 \cdot 8 \cdot 6560 = 816000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 816000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

| | | | | | | |
|-----|-----|----------|--|--------|----------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 59 |

$$I_{не} = R_{сн}^1 \cdot Ц_{м}, \quad (7.10)$$

де: $Ц_{м}$ – ціна меблів для одного робочого місця, грн.

$$I_{не} = 12 \cdot 3500 = 42000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Brain за 28.10.22 – джерело <https://brain.com.ua>.

Таблиця 7.6 – Специфікація

| Найменування комплектуючої або обладнання | Тип | Оптова ціна |
|---|--|-------------|
| Персональний комп'ютер | | 36999 |
| Системний блок | Asus 3D textured mATX S500MC Reddot Winner 2021 | - |
| Процесор | б'ядер Intel Core i5-11400 (2.6 - 4.4 ГГц) | - |
| Системна плата | Asus B560, Чіпсет Intel B560, 2 x USB 3.2, 2 x USB 2.0, 3 x Audio, 1 x Microphone, 4 x USB 2.0, 2 x PS/2, 1 x HDMI, 3 x Display port, 1 x RJ45, Realtek ALC887, 10/100/1000 Мбіт/с | - |
| Відеокарта | Asus GeForce RTX 3060 Phoenix V2 12 ГБ GDDR6 | - |
| Жорсткий диск | SSD 1000 GB | - |
| Оперативна пам'ять | SK Hynix 16ГБ DDR4-3200 (2 x 8ГБ) | - |
| DVD-привод | Не комплектується | - |
| Корпус | Asus 3D textured mATX S500MC Reddot Winner 2021 | - |

Продовження таблиці 7.6

| Найменування комплектуючої або обладнання | Тип | Оптова ціна |
|---|---|-------------|
| Кардрідер внутрішній | USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black | 220 |
| інше | Клавіатура, мишка | Подарунок |
| Монітор | Монитор 24" BenQ GL2480 Black (9H.LHXLB.QBE/9H.LHXLB.VBE). Діагональ дисплея 24" Максимальное разрешение дисплея 1920 x 1080 Время реакции матрицы 1 мс (GtG) Яркость дисплея 250 кд/м2 Тип матрицы TN Контрастность дисплея 1000:1 Особенности Flicker-Free | 3675 |
| Принтер лазерний | Canon i-SENSYS LBP6030W | 2700 |
| Принтер струминний | Epson Stylus Photo P50 (C11CA45341) + USB cable | 5500 |
| Копіювальний апарат | Canon i-SENSYS MF217W with Wi-Fi | 5965 |

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | БКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 61 |

Таблиця 7.7 – Балансова вартість обчислювальної техніки

| Найменування обчислювальної техніки | Кількість, шт. | Ціна за одиницю, грн. | Витрати на транспортування, монтаж та випробовування. | Загальна вартість, грн. |
|-------------------------------------|----------------|-----------------------|---|-------------------------|
| Персональні комп'ютери | 12 | 36999 | 16420,5 | 460480 |
| Принтер лаз. | 2 | 2700 | 540 | 5940 |
| Принтер струм. | 1 | 5500 | 550 | 6050 |
| Копіюв. апарат | 1 | 5965 | 596,5 | 6561,5 |
| Всього | – | – | – | 479031 |

ВКРМ -122.22.0011.00.00.ПЗ

| | | | | |
|-----|-----|----------|--------|------|
| Вим | Арк | № докум. | Підпис | Арк. |
| | | | | 62 |

Витрати на транспорт, монтаж та випробування прийняті в межах до 10% від оптової ціни.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

| Групи та види основних фондів | Балансова вартість, грн. | Амортизація | |
|-------------------------------|--------------------------|-------------|--------------------|
| | | Норма, % | Відрахування, грн. |
| 1 | 2 | 3 | 4 |
| Група 3 | | | |
| 1. Будівлі | 816000 | - | - |
| 2. Передавальні пристрої | 81600 | - | - |
| Всього по групі | 897600 | 5 | 44880 |
| Група 4 | | | |
| 3. Обчислювальна техніка | 479031 | - | - |
| Всього по групі | 479031 | 50 | 239515 |
| 4. Нематеріальні активи | 250000 | 50 | 125000 |
| Група 5, 6 | | | |
| 5. Вимірювальні пристрої | 9031 | 25 | 2257,75 |
| 6. Транспортні засоби | 339200 | 20 | 67840 |
| 7. Господарський інвентар | 42000 | 25 | 10500 |
| Всього по групі | 390 231 | - | 80 597 |
| Разом | $K_p = 2016862$ | | $A_p = 489992$ |

Примітка: вартість автомобіля RENAULT Megane BOSE взята по даним з автосалону «RST», джерело <https://rst.ua> складає 339200 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 484 \cdot 180 / 45 = 1936 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 1936 \cdot 10 \cdot 0,01 = 193,6 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 37\%$ від суми основної та додаткової зарплати:

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 37 (1936 + 193,6) = 468,5 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д.) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 1936 \cdot 15 \cdot 0,01 = 290,4 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.;

Z_{M2} – вартість запам'ятовуючих пристроїв, грн.;

Z_{M3} – вартість фарби, картриджей, тонеру, грн.;

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | БКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 64 |

N_e – кількість екземплярів програм, шт.

Згідно виданих викладачем норм приймаємо 0,5 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 265$ грн., визначаємо вартість паперу за період розробки $N_m = 1$ міс:

$$Z_{M1} = C_n \cdot N_m. \quad (7.16)$$

$$Z_{M1} = 265 \cdot 1 \cdot 0,5 = 132,5 \text{ грн.}$$

Згідно виданих викладачем норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості, що дорівнює 45 екземплярів програм та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 30 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 30 грн./шт.

$$Z_{M2} = 45 \cdot 30 + 30 = 1380 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_z, \quad (7.18)$$

де: C_z – вартість розхідних матеріалів друкуючих пристроїв: картридж для CANON LBP-3010 Black Canon 712 – 574 грн.; картридж для EPSON STYLUS PHOTO R390 – 558 грн.; картридж для CANON IR-1022A – LJ Q2612A Cart. HP LJ 1010/1012/1015/3015/3020/3030 (2500 стр.) – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (132,5 + 1380 + 1702) / 45 = 71,4 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 1936 \cdot 15 \cdot 0,01 = 290,4 \text{ грн.}$$

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 65 |

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 45$ прим.):

$$A_m = \frac{A_p \cdot N_{\text{міс}}}{N_e \cdot 12}, \quad (7.20)$$

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 489992 \cdot 1 / (45 \cdot 12) = 907,3 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 1936 + 193,6 + 468,5 + 290,4 + 71,4 + 290,4 + 907,3 = 4157,6 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 30%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 30 \cdot 4157,6 = 10247,2 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9.

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 66 |

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

| Найменування статей витрат | Позначення | Величина, грн. |
|--|------------|----------------|
| 1 | 2 | 3 |
| 1. Основна зарплата виконавців | Z_o | 1936 |
| 2. Додаткова зарплата виконавців | Z_d | 193,6 |
| 3. Відрахування на соціальні потреби | C_{oc} | 468,5 |
| 4. Загальногосподарські витрати | G_{ocn} | 290 |
| 5. Витрати на матеріали | Z_M | 71,4 |
| 6. Освоєння нових операційних систем, мов програмування | O_n | 290,4 |
| 1 | 2 | 3 |
| 7. Амортизація основних фондів | A_m | 907,3 |
| 8. Повна собівартість програмного забезпечення | C_n | 4157,6 |
| 9. Плановий прибуток | P_p | 10247 |
| 10.Ціна підприємства $C_n = C_n + P_p$ | C_n | 14404,6 |
| 11.Податок на додану вартість $ПДВ = 0.01 \cdot H_{дв} \cdot C_n$ | $ПДВ$ | 2880,92 |
| 12.Відпускна ціна програмної продукції $C = C_n + ПДВ$ | C | 17285,5 |

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.9.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

| Найменування капітальних вкладень | Сума за варіантами, грн. | |
|-----------------------------------|--------------------------|---------|
| | Базовий | Новий |
| Вартість програмної продукції | – | 17285,5 |
| Всього капітальних витрат | – | 17285,5 |

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

| Найменування статей витрат | Позначення | Сума витрат за варіантами, грн. | |
|------------------------------|------------|---------------------------------|-------|
| | | Базовий | Новий |
| 1. Витрати на обслуговування | Z_p | 28211 | 17632 |
| 2. Витрати на електроенергію | $Z_{ел}$ | 7600 | 7220 |
| 3. Витрати на амортизацію | $Z_{ам}$ | 0 | 630 |
| Всього витрат за рік | I | 35811 | 24852 |

Витрати на обслуговування системи:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування за рік, год.;

Z_2 – заробітна плата обслуговуючого персоналу, грн/год.

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 240 годин на рік до 150 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 26 \cdot 1,1 \cdot 1,37 \cdot 3 = 28211 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 150 \cdot 26 \cdot 1,1 \cdot 1,37 \cdot 3 = 17632 \text{ грн}.$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 69 |

$$Z_{ел} = П_{ел} \cdot T_p \cdot Ц_{ел}. \quad (7.24)$$

$$Z_{ел баз} = 7 \cdot 0,15 \cdot 240 \cdot 1,67 = 421 \text{ грн.}$$

$$Z_{ел нов} = 7 \cdot 0,15 \cdot 150 \cdot 1,67 = 263 \text{ грн.}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

| Групи основних фондів | Норма амортизації % | Балансова вартість, грн., за варіантами | | Сума відрахувань, грн., за варіантами | |
|-----------------------|------------------------|---|---------|---------------------------------------|--------|
| | | Базовий | Новий | Базовий | Новий |
| Програмна продукція | 50 | – | 17285,5 | – | 8642,7 |
| Всього відрахувань | - | – | 17285,5 | – | 8642,7 |

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (C_n - C_n) \cdot N_e - E_n \cdot K_p, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.

$$E_e = (14404,6 - 4157,6) \cdot 45 - 0,15 \cdot 2016862 \cdot 1/12 = 435904,2$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p}{(C_n - C_n) \cdot N_e}, \quad (7.26)$$

$$T_e = 2016862 / (((14404,6 - 4157,6) \cdot 45 \cdot 12) / 1) = 0,3 \text{ років.}$$

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\delta} - I_n) - E_n(K_n - K_{\delta}), \quad (7.27)$$

де: I_{δ} , I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

K_{δ} , K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (35811 - 24852) - 0,5 \cdot 17285 = 2316,5 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\delta}}{I_{\delta} - I_n}, \quad (7.28)$$

$$T_{cn} = 17285 / (35811 - 24852) = 1,5 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

| Найменування показників | Одиниця виміру | Величина |
|---|----------------|----------|
| 1. Кількість екземплярів програми | Прим. | 45 |
| 2. Повна собівартість розробленої програми | Грн. | 4157,6 |
| 3. Ціна розробленої програми | Грн. | 14404,6 |
| 4. Плановий прибуток від реалізації розробленої програми | Грн. | 10247,2 |
| 5. Рентабельність програмної продукції | % | 30 |
| 6. Об'єм додаткових капітальних вкладень у виробника програмної продукції | Грн. | 2016862 |
| 7. Загальний прибуток від реалізації програмної продукції | Грн. | 461124 |

Продовження таблиці 7.13

| Найменування показників | Одиниця виміру | Величина |
|---|----------------|----------|
| 8. Величина економічного ефекту при виготовлені програмної продукції | Грн. | 435904,2 |
| 9. Період окупності додаткових капітальних вкладень у виробника програмної продукції | Років | 0,3 |
| 10. Об'єм додаткових капітальних вкладень у споживача програмної продукції | Грн. | 17285,5 |
| 11. Величина економічного ефекту у користувача програмної продукції | Грн. | 2316,5 |
| 12. Період окупності додаткових капітальних вкладень у користувача програмної продукції | Років | 1,5 |

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 72 |

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Законом України “Про охорону праці” [3] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

Програмісти у процесі роботи отримують негативний вплив на органи зору, а також мають значну розумову напругу і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [5], та «Вимоги щодо безпеки

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | БКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 73 |

та захисту здоров'я працівників під час роботи з екранними пристроями»
НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення негативного впливу комп'ютера на організм людини визначимо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Шкідливими факторами при роботі з персональним комп'ютером є неіонізуюче випромінювання промислової частоти, збільшене нервово-емоційне навантаження на оператора, збільшення навантаження на органи зору та дрібні стереостатичні рухи кінцівок.

Ці фактори можуть викликати у працівника певні розлади здоров'я, зокрема підвищення артеріального тиску, кон'юктивіти, тендовагініти та інші захворювання.

Комп'ютер, як і будь-який електричний прилад, особливо при його неправильному підключенні, може бути джерелом ураження оператора електричним струмом. Саме тому всі працівники, які працюють з персональним комп'ютером, повинні мати першу (або другу) групу допуску з електробезпеки.

Через наявність зазначених факторів працівники, які працюють з персональними комп'ютерами, підлягають попередньому та періодичному медичному огляду згідно з пунктом 6.2.3 додатку 4 до наказу Міністерства охорони здоров'я України «Про затвердження Порядку проведення медичних оглядів працівників певних категорій» від 21 травня 2007 року № 246.

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 74 |

8.3 Дослідження та аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 – Геометричні розміри приміщення

| Найменування | Значення, м |
|--------------|-------------|
| Ширина | 5,4 |
| Довжина | 6 |
| Висота | 2,75 |

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого*

| Геометрична характеристика | Одиниця виміру | Нормативне значення* | Фактичне значення |
|----------------------------|----------------|----------------------|-------------------|
| Площа, S | м ² | не менше 6.0 | 8,1 |
| Обсяг, V | м ³ | не менше 20.0 | 24,3 |

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

У зазначеному приміщенні працюють 4 людей. За даними, які наведено у табл. 8.1, та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста не відповідають нормативним вимогам ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [63], але відповідають нормативним вимогам Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [63] та НПАОП 0.00-1.28-10 «Правила

охорони праці під час експлуатації електронно-обчислювальних машин»). Таним чином можна зробити висновок, що санітарно-гігієнічні умови праці на робочому місці програміста відповідають вимогам.

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно

Постанови № 42 від 01.12.1999 Головного державного санітарного лікаря України, робота, виконувана в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря в приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 76 |

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

| Пора року | Оптимальні для Іа | | | Фактичні | | |
|-----------|-------------------|--------------|------------------------|-----------------|--------------|------------------------|
| | Температура, °С | Вологість, % | Швидкість повітря, м/с | Температура, °С | Вологість, % | Швидкість повітря, м/с |
| Холодна | 22-24 | 40-60 | 0,1 | 22-23 | 40-55 | 0,1 |
| Тепла | 23-25 | 50-70 | 0,1 | 24-25 | 50-65 | 0,11 |

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року, а в літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер HP 1100, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

З 2019 року діють Державні будівельні норми України “Природне і штучне освітлення” – ДБН В.2.5-28:2018 [59], у яких прописані вимоги до використання всіх освітлювальних приладів, у т.ч. світлодіодних.

Працю працівника, який постійно працює за комп’ютером, згідно ДБН В.2.5-28:2018 [59], можна віднести до роботи з малою точністю (найменший розмір об’єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об’єкта розрізнення з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об’єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і

розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 Лк. [59], Крім того все поле зору повинне бути освітлено достатньо рівномірно - ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення, яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

З метою належного правового забезпечення необхідно розширити та доповнити перелік основних професій комп'ютерної галузі у національному класифікаторі ДК-003-2010, а також підготувати відповідний випуск у кваліфікаційному довіднику посад фахівців ІТ-індустрії, що сприятиме вирішенню питань їх соціального захисту, пенсійного забезпечення, атестації робочих місць основних професій за умовами праці на предмет подальших певних видів пільг та компенсацій за важкі шкідливі і небезпечні умови праці.

Важливим напрямом стосовно визначення професійної придатності фахівців з інформаційних технологій є проведення психофізіологічної експертизи відповідно до 5 статті Закону України «Про охорону праці». Робота з комп'ютерами нового покоління характеризується певним психофізіологічними перенавантаженнями, втому зорового аналізатора, гіпокінезією, відсутність диференційованих норм праці при роботі з новою комп'ютерною технікою в залежності від віку, статі, категорії зорової роботи, режимів праці і відпочинку (протягом робочого дня, тижня, щорічного режиму відпусток).

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | БКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 78 |

Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців ІТ-індустрії і стандартів підприємств, центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів.

Особлива роль з точки зору збереження та відновлення здоров'я працюючих в комп'ютерній галузі належить попереднім та періодичним наглядам з подальшої психофізіологічної експертизи і встановленням професійної придатності при роботі з комп'ютерами нового покоління, який супроводжується виникненням певних факторів професійного ризику електротравматизму при їх ремонті та обслуговуванні. В цьому зв'язку необхідне запровадження експертизи на предмет безпечної експлуатації ПЕОМ, тобто офіційне підтвердження фактичних параметрів електробезпеки, їх відповідності вимогам нормативної документації фахівців, які проводять таку експертизу повинні пройти навчання і перевірку знань відповідно до вимог ДНАОП 0.00-8.20-99. За результатами експертизи повинні прийматися рішення про відповідність ПЕОМ нормам безпеки, терміни чергової експертизи, оформлюються протоколи вимірювань і випробувань, проведені у разі потреби розрахунки та експертний висновок.

Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців ІТ-індустрії.

Зарубіжний досвід охорони праці при використанні новітніх інформаційних технологій та сучасного комп'ютерного обладнання передбачає з метою попередження наслідків монотонної праці, підвищення рівня рухової активності і покращення розумової працездатності фахівців ІТ-індустрії під час технологічних перерв участь у спеціальних облаштованих приміщеннях необхідним спортивним інвентарем та різними тренажерами

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | <i>ВКРМ -122.22.0011.00.00.ПЗ</i> | Арк. |
| Вим | Арк | № докум. | | Підпис | | 79 |

відповідних фізичних вправ, індивідуальних тренінгових завдань відповідно до віку, статі та категорії зорової роботи. Такий підхід дозволяє зняти надлишкове психофізіологічне перевантаження, підвищити працездатність центральної нервової системи, попередити перевтому зорового аналізатора. Показана ефективність проведення різноманітних за своєю спрямованістю вправ робітників цієї галузі (приблизно на 5-30%).

Всі наведені заходи щодо вдосконалення охорони праці фахівців ІТ-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці підприємства.

Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

8.5 Розрахункова частина

Занулення електричних установок виконується навмисним з'єднанням корпусів електричних установок із захисним нульовим проводом за допомогою занулюючих провідників.

Розрахунок занулення складається з трьох частин:

1. Розрахунок на відключаючу спроможність;
2. Визначення максимальної напруги на корпусі обладнання відносно землі при замиканні фази на корпус;
3. Розрахунок робочого і повторного заземлювачів

Початкові данні:

1. Потужність електродвигуна, який підлягає зануленню : $P = 20$ кВт.
2. Кількість електродвигунів: $m = 10$.
- Потужність освітлювальних приладів : $P_o = 30$ кВт.
3. Довжина магістрального кабеля: $LM=100$ м.
4. Довжина розгалудження (від розподільчого щита до електродвигуна)

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 80 |

9 ОСНОВНІ ВИСНОВКИ

Після проведення всіх досліджень в якості мови програмування була обрана мова Python та інтерактивне середовище розробки Visual Studio Code, було розглянуто та створені функціональні та структурні схеми і по зроблена діаграма процесів рекомендаційної системи на базі чат боту.

Опис функціональної схеми, мав в собі концепцію програмного забезпечення для рекомендаційної системи. Розглянуті основні частини обмінну даними між клієнтом та системою.

Були розглянуті та реалізовані приклади рекомендацій на основі контентної фільтрації. Розглянуті методи підрахунку схожості як для музичних творів так і для фільмів. Використовуючи набори даних для формування рекомендацій змогли вивести рекомендації базуючись на введеному запиті.

Робота системи є універсальною і її можна використовувати і для інших продуктів, які не були розглянуті в цій реалізації. Метод дає системний підхід для вирішення подібних задач, але для кожної потрібні будуть лише деякі логічні зміни, щоб система підходила до нової бази та методів пошуку її компонентів. Для цього були обрані інструменти які дозволяють проводити оцінку і багатокритеріальні порівняння різних типів даних.

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | <i>ВКРМ -122.22.0011.00.00.ПЗ</i> | Арк. |
| Вим | Арк | № докум. | | Підпис | | 88 |

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Big Data Частина 3: Spark. [Електронний ресурс] 05.10.2015. – Режим доступу: <https://habr.com/ru/company/dca/blog/268277/>
2. Linden G., Smith B., and York J. (2003). Item-to-Item Collaborative Filtering. Los Alamitos, CA, USA: IEEE Internet Computing. p76–80
3. Мазурік О. Ю. Покращення результатів роботи рекомендаційних систем за допомогою алгоритму SVD / О. Ю. Мазурік // International scientific journal. – 2015. – № 9. – С. 61-64.
4. Рекомендаційні системи [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Рекомендаційна_система
5. Інтелектуальні технології Data Mining і Text Mining [Електронний ресурс]. — Режим доступу: https://pidruchniki.com/1623021247786/informatika/i_ntelektualni_tehnologiyi_data_mining_text_mining
6. Добування даних [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Добування_даних
7. Francesco R., Lior R., Bracha S., Paul B. (2011). Recommender System Handbook. P.845
8. Є.В. Мелешко, С.Г. Семенов, В.Д. Хох Дослідження методів побудови рекомендаційних систем в мережі інтернет [Електронний ресурс] – Режим доступу: <http://journals.nupp.edu.ua/sunz/article/view/949/796>
9. Є.В. Мелешко. Методи оцінки якості роботи рекомендаційних систем [Електронний ресурс] – Режим доступу: <http://journals.nupp.edu.ua/sunz/article/view/1271/1071>
10. Є.В. Мелешко. Проблеми сучасних рекомендаційних систем та методи їх рішення [Електронний ресурс] – Режим доступу: <http://journals.nupp.edu.ua/sunz/article/view/1211/1019>

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 89 |

11. Колаборативна фільтрація [Електронний ресурс]. — Режим доступу: <https://habr.com/ru/post/150399/>
12. Як вирватися з інформаційної бульбашки [Електронний ресурс]. — Режим доступу: <https://wz.lviv.ua/blogs/388693-yak-vyrvatysia-z-informatsiinoi-bulbashky>
13. Точність та повнота [Електронний ресурс]. — Режим доступу: https://uk.wikipedia.org/wiki/Точність_та_повнота
14. Pyrogram [Электронный ресурс] : Pyrogram Documents. – Режим доступу: <https://docs.pyrogram.org/faq#what-is-pyrogram>
15. FastCrypto [Электронный ресурс] : Pyrogram Documents. – Режим доступу: <https://docs.pyrogram.org/topics/tgcrypto>
16. Welcome to Python Telegram Bot's documentation [Електронний ресурс] - Режим доступу: <https://python-telegram-bot.readthedocs.io/en/stable>
17. Learn to build your first bot in Telegram with Python n [Електронний ресурс] - Режим доступу: <https://medium.freecodecamp.org/learn-to-build-your-first-bot-in-telegram-with-python-4c99526765e4>
18. Telegram FAQ [Електронний ресурс]. – Режим доступу: <https://telegram.org/faq>
19. Python Requests [Електронний ресурс]. – Режим доступу: <https://2.python-requests.org/en/master/>
20. Черга повідомлень [Електронний ресурс]. – Режим доступу: <https://aws.amazon.com/ru/message-queue/>
21. TelegramBotAPI [Електронний ресурс] - Режим доступу: <https://core.telegram.org/bots/api>
22. Огляд протоколу HTTP [Електронний ресурс]. Режим доступу: <https://developer.mozilla.org/ru/docs/Web/HTTP/Overview>
23. JSON [Електронний ресурс]. –Режим доступу: <https://www.json.org/json-ru.html>

| | | | | | | |
|-----|-----|----------|--------|--|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | Підпис | | | 90 |

46. 58. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>

47. Охорона праці. Ч. 2. Занулення : метод. вказ. до викон. розрахунків з викор. персон. ЕОМ IBM–сумісного типу / [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака, А. Е. Солових, С. Е. Катеринич] ; Мін-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. - 2–ге вид., перероб. та доп. - Кропивницький : ЦНТУ, 2019. - 27 с. Режим доступу до ресурсу: [CUNTUR: Охорона праці. Ч. 2. Занулення \(kntu.kr.ua\)](http://CUNTUR:Охорона праці. Ч. 2. Занулення (kntu.kr.ua))

48. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99>

49. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : “Колос”, 1973. – 238 с.

50. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення 23.10.22).

| | | | | | | |
|-----|-----|----------|--|--------|-----------------------------------|------|
| | | | | | ВКРМ -122.22.0011.00.00.ПЗ | Арк. |
| Вим | Арк | № докум. | | Підпис | | 93 |

Додаток А
(обов'язковий)

Технічне завдання

Зміст

| | |
|---|---|
| 1 Найменування та область застосування..... | 2 |
| 2 Підстава для розробки..... | 2 |
| 3 Мета та призначення розробки..... | 2 |
| 4 Джерела розробки..... | 2 |
| 5 Технічні вимоги..... | 2 |
| 5.1 Вміст проекту..... | 2 |
| 5.2 Показники призначення..... | 3 |
| 5.3 Вимоги до функціональних характеристик..... | 3 |
| 5.4 Вимоги до архітектури..... | 3 |
| 5.5 Вимоги до надійності..... | 3 |
| 5.6 Умови експлуатації..... | 4 |
| 5.7 Вимоги до складу та параметрів технічних засобів..... | 4 |
| 5.8 Вимоги до інформаційної і програмної сумісності..... | 4 |
| 5.8.1 Обладнання..... | 4 |
| 5.8.2 Мова програмування..... | 4 |
| 5.8.3 Вхідні дані..... | 5 |
| 5.8.4 Вихідні дані..... | 5 |
| 6 Вимоги до програмної документації..... | 5 |
| 7 Економічні вимоги..... | 5 |
| 8 Вимоги щодо охорони праці..... | 5 |
| 9 Перелік документів, що розробляються..... | 6 |
| 10 Етапи розробки..... | 6 |
| 11 Порядок контролю та приймання..... | 6 |

| | | | | | | | | |
|-----------|-----------------|-------------|--------|------|---|------|-------|---------|
| | | | | | ВКРМ-122.22.0011.00.00.ТЗ | | | |
| Вим. | Арк. | № документа | Підпис | Дата | | | | |
| Розробив | Подкопачев Д.М. | | | | Дослідження та програмна реалізація хмарної рекомендаційної системи обрання контенту з застосуванням технологій штучного інтелекту | Літ. | Аркуш | Аркушів |
| Перевірів | Якименко Н.М. | | | | | М | 1 | 6 |
| Н. Контр. | Гермак В.С. | | | | ЦНТУ КН-21М-1,4 | | | |
| Затв. | Смірнов О.А. | | | | | | | |

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію хмарної рекомендаційної системи обрання контенту з застосуванням технологій штучного інтелекту.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 19-13 від 17.08.2022 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація хмарної рекомендаційної системи обрання контенту з застосуванням технологій штучного інтелекту.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

| | | | | | | |
|------|------|-------------|--------|------|---------------------------|------|
| | | | | | ВКРМ-122.22.0011.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата | | 2 |

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- хмарної рекомендаційної системи обрання контенту з застосуванням технологій штучного інтелекту;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

| | | | | | | |
|------|------|-------------|--------|------|----------------------------------|------|
| | | | | | ВКРМ-122.22.0011.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата | | 3 |

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Python.

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

| | | | | | | |
|------|------|-------------|--------|------|----------------------------------|------|
| | | | | | ВКРМ-122.22.0011.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата | | 2 |

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2022 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута пожежна безпека.

9 Перелік документів, що розробляються

- | | |
|--|---------------|
| – Наукова новизна | – 1 аркуш. |
| – Структурна схема системи | – 1 аркуш. |
| – Функціональна схема системи | – 1 аркуш. |
| – Діаграма процесів | – 1 аркуш. |
| – Блок-схема алгоритму роботи програми | – 3 аркуша. |
| – Показники економічної ефективності | – 1 аркуш. |
| – Пояснювальна записка | – 92 аркушів. |

| | | | | | | |
|------|------|-------------|--------|------|----------------------------------|------|
| | | | | | ВКРМ-122.22.0011.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата | | 5 |

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2022 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 22.12.2022 р.

| | | | | | | |
|------|------|-------------|--------|------|----------------------------------|------|
| | | | | | ВКРМ-122.22.0011.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата | | 6 |

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи
за другим (магістерським) рівнем вищої освіти
_____ Якименко Н.М.

*Дослідження та програмна реалізація хмарної рекомендаційної системи
обрання контенту з застосуванням технологій штучного інтелекту*

Лістинг програми

Код документу 12

Носій: DVD-диск

Загальна кількість аркушів: 13

Літера: РП

Кропивницький – 2022 року

Лістинг momutbot.py

```

import telebot
from telebot import types
import random
import subprocess
import tempfile
import os
import json
import urllib.request
from keyboa import keyboa_maker
import pyqrcode
from PIL import Image
from pyzbar import pyzbar
from PIL import Image
import requests
import speech_recognition as sr
import musicR as mr
import movie_r as movr

r = sr.Recognizer()
bot = telebot.TeleBot('1457018782:AAG-GKQglqynBLx7X2UqWuyiXMZpsBpbZ9c')
response = requests.get(url).json()

def findat(msg):
    # from a list of texts, it finds the one with the '@' sign
    for i in msg:
        if '@' in i:
            return i

@bot.message_handler(commands=['start']) # welcome message handler
def send_welcome(message):
    bot.reply_to(message, "Вітаю, {0.first_name}!\nЯ - {1.first_name},  
Momut. Спробуй мене. Я можу запропонувати рекомендації по музичним трекам та  
фільмам, для цього обери одну із команд !".format(message.from_user,  
bot.get_me()))

@bot.message_handler(commands=['help']) # help message handler
def send_welcome(message):
    bot.reply_to(message, "Якщо виникли якісь питання звертайтеся за адресою  
devonoset@gmail.com")

@bot.message_handler(commands=['recom_music'])
def rm(message):
    send = bot.send_message(message.chat.id, "Надішли мені посилання на твою  
композицію в Spotify")
    bot.register_next_step_handler(send, recmu)
def recmu(message):
    bot.send_message(message.chat.id, 'Провожу аналіз твого поіслання  
' + message.text)
    data = message.text
    rm = rec.create(data)
    mr=rm.recomendet((track_id, ref_df, sp, n_recs = 5))
    qmu = open('nrecs')
    bot.send_photo(message.chat.id, qmu)

@bot.message_handler(commands=['recom_movie'])
def rm(message):
    send = bot.send_message(message.chat.id, "Надішли мені назву фільму для  
аналізу")
    bot.register_next_step_handler(send, recmu)
def recmu(message):

```

```

    bot.send_message(message.chat.id, 'Провожу аналіз
'.format(qrs=message.text))
    data = message.text
    mvr = movrec.create(data)
    mor= mvr.find_similar(title)
    m = open(most_similar)
    bot.send_photo(message.chat.id, m)

@bot.message_handler(commands=['creat_qr'])
def crQR(message):
    send = bot.send_message(message.chat.id, "Створити QR код для створенно
листу рекомендацій ")
    bot.register_next_step_handler(send, qrs)
def qrs(message):
    bot.send_message(message.chat.id, 'Створюю код за твоїм списком
рекомендацій '.format(qrs=message.text))
    data = message.text
    qr = pyqrcode.create(data)
    qr.png("yourcode.png", scale=6)
    q = open('yourcode.png', 'rb')
    bot.send_photo(message.chat.id, q)

@bot.message_handler(commands=['decode_qr'])
def dqr(message):
    bot.send_message(message.chat.id, "Надішли мені qr-код")

@bot.message_handler(content_types=['photo'])
def photo(message):
    fileID = message.photo[-1].file_id
    file_info = bot.get_file(fileID)
    downloaded_file = bot.download_file(file_info.file_path)

    with open("image.jpg", 'wb') as new_file:
        new_file.write(downloaded_file)
    dqr = open('image.jpg', 'rb')
    img = Image.open(dqr)
    decoded = pyzbar.decode(img)
    data = decoded[0].data
    bot.send_message(message.chat.id, data)

@bot.inline_handler(func=lambda query: len(query.query) > 0)
def query_text(query):
    r = types.InlineQueryResultArticle(
        id='1',
        title=query.query,
        input_message_content=types.InputTextMessageContent(
            message_text=
            "https://open.spotify.com/track/"+query.query+"?si=72a0e97138f241f2"),
            #
            url="https://open.spotify.com/track/"+query.query+"?si=72a0e97138f241f2"),
        )

    bot.answer_inline_query(query.id, [r])

bot.polling(none_stop=True)

```

Лістинг musicR.py

```

import pandas as pd
import random
import auth
import numpy as np
from numpy.linalg import norm

#read dataset from which was created above
df = pd.read_csv("tracks.csv", low_memory=False)

#lets combine valence and energy
df["mood_vec"] = df[["valence", "energy"]].values.tolist()

sp = auth.auth()

def recommend(track_id, ref_df, sp, n_recs = 5):

    # Crawl valence and arousal of given track from spotify api
    track_features = sp.track_audio_features(track_id)
    track_moodvec = np.array([track_features.valence, track_features.energy])

    # Compute distances to all reference tracks
    ref_df["distances"] = ref_df["mood_vec"].apply(lambda x:
norm(track_moodvec-np.array(x)))
    # Sort distances from lowest to highest
    ref_df_sorted = ref_df.sort_values(by = "distances", ascending = True)
    # If the input track is in the reference set, it will have a distance of
0, but should not be recommendet
    ref_df_sorted = ref_df_sorted[ref_df_sorted["id"] != track_id]

    # Return n recommendations
    return ref_df_sorted.iloc[:n_recs]

track = "6PwjJ58I4t7Mae9xfZ9l9v"
print(recommend(track_id = track, ref_df = df, sp = sp, n_recs = 15))

```

Лістинг movieR.py

```

import numpy as np
import pandas as pd
import torch
import torch.nn as nn
import torch.nn.parallel
import torch.optim as optim
import torch.utils.data
from torch.autograd import Variable

# Importing the dataset
movies = pd.read_csv('ml-1m/movies.dat', sep = '::', header = None, engine =
'python', encoding = 'latin-1')
users = pd.read_csv('ml-1m/users.dat', sep = '::', header = None, engine =
'python', encoding = 'latin-1')
ratings = pd.read_csv('ml-1m/ratings.dat', sep = '::', header = None, engine
= 'python', encoding = 'latin-1')

# Preparing the training set and the test set
training_set = pd.read_csv('ml-100k/ul.base', delimiter = '\t')
training_set = np.array(training_set, dtype = 'int')
test_set = pd.read_csv('ml-100k/ul.test', delimiter = '\t')
test_set = np.array(test_set, dtype = 'int')

# Getting the number of users and movies
nb_users = int(max(max(training_set[:,0]), max(test_set[:,0])))
nb_movies = int(max(max(training_set[:,1]), max(test_set[:,1])))

# Converting the data into an array with users in lines and movies in columns
def convert(data):
    new_data = []
    for id_users in range(1, nb_users + 1):
        id_movies = data[:,1][data[:,0] == id_users]
        id_ratings = data[:,2][data[:,0] == id_users]
        ratings = np.zeros(nb_movies)
        ratings[id_movies - 1] = id_ratings
        new_data.append(list(ratings))
    return new_data
training_set = convert(training_set)
test_set = convert(test_set)

# Converting the data into Torch tensors
training_set = torch.FloatTensor(training_set)
test_set = torch.FloatTensor(test_set)

# Creating the architecture of the Neural Network
class SAE(nn.Module):
    def __init__(self, ):
        super(SAE, self).__init__()
        self.fc1 = nn.Linear(nb_movies, 20)
        self.fc2 = nn.Linear(20, 10)
        self.fc3 = nn.Linear(10, 20)
        self.fc4 = nn.Linear(20, nb_movies)
        self.activation = nn.Sigmoid()
    def forward(self, x):
        x = self.activation(self.fc1(x))
        x = self.activation(self.fc2(x))
        x = self.activation(self.fc3(x))
        x = self.fc4(x)
        return x

```

```

sae = SAE()
criterion = nn.MSELoss()
optimizer = optim.RMSprop(sae.parameters(), lr = 0.01, weight_decay = 0.5)

nb_epoch = 256
for epoch in range(1, nb_epoch + 1):
    train_loss = 0
    s = 0.
    for id_user in range(nb_users):
        input = Variable(training_set[id_user]).unsqueeze(0)
        target = input.clone()
        if torch.sum(target.data > 0) > 0:
            output = sae(input)
            target.require_grad = False
            output[target == 0] = 0
            loss = criterion(output, target)
            mean_corrector = nb_movies/float(torch.sum(target.data > 0) + 1e-
            loss.backward()
            train_loss += np.sqrt(loss.data[0]*mean_corrector)
            s += 1.
            optimizer.step()

test_loss = 0
s = 0.
for id_user in range(nb_users):
    input = Variable(training_set[id_user]).unsqueeze(0)
    target = Variable(test_set[id_user]).unsqueeze(0)
    if torch.sum(target.data > 0) > 0:
        output = sae(input)
        target.require_grad = False
        output[target == 0] = 0
        loss = criterion(output, target)
        mean_corrector = nb_movies/float(torch.sum(target.data > 0) + 1e-10)
        test_loss += np.sqrt(loss.data[0]*mean_corrector)
        s += 1.

```

Лістинг datarecon.py

```
import sys
import auth
import pandas as pd
import time
from tqdm import tqdm

sp = auth.auth()

genres = sp.recommendation_genre_seeds()

n_recs = 100

data_dict = {"id":[], "genre":[], "track_name":[], "artist_name":[],
             "valence":[], "energy":[]}

for g in tqdm(genres):

    recs = sp.recommendations(genres = [g], limit = n_recs)

    recs = eval(recs.json().replace("null", "-999").replace("false",
"False").replace("true", "True"))["tracks"]

    for track in recs:

        data_dict["id"].append(track["id"])
        data_dict["genre"].append(g)

        track_meta = sp.track(track["id"])
        data_dict["track_name"].append(track_meta.name)
        data_dict["artist_name"].append(track_meta.album.artists[0].name)

        track_features = sp.track_audio_features(track["id"])
        data_dict["valence"].append(track_features.valence)
        data_dict["energy"].append(track_features.energy)

        time.sleep(0.2)

df = pd.DataFrame(data_dict)

df.drop_duplicates(subset = "", keep = "first", inplace = True)
df.to_csv("valence_arousal_dataset.csv", index = False)
```

Лістинг webhook.py

```

import datetime
import time
import traceback

import pytz
from flask import request
from werkzeug.exceptions import HTTPException

from setup import *

# ----- server boot time -----
boot_time = time.time()
boot_date = datetime.datetime.now(tz=pytz.timezone("Europe/Moscow"))

# ----- Exception handler -----
@app.errorhandler(Exception)
def handle_exception(e):
    if isinstance(e, HTTPException):
        return e

    logger.error(traceback.format_exc())
    return "Oops", 500

# ----- status webpage -----
@app.route('/')
def status():
    password = request.args.get("password")
    if password != ADMIN_PASSWORD:
        logger.info('Status page loaded without password')
        return "<h1>Access denied!</h1>", 403

    return f'<h1>This is telegram bot server, ' \
        f'<a href="https://github.com/otter18/telegram-bot-  

template">templated</a> by ' \
        f'<a href="https://github.com/otter18">@otter18</a></h1>' \
        f'<p>Server uptime: {datetime.timedelta(seconds=time.time() -  

boot_time)}</p>' \
        f'<p>Server last boot at {boot_date}'

# ----- webhook -----
@app.route('/') + WEBHOOK_TOKEN, methods=['POST'])
def getMessage():
    # temp = request.stream.read().decode("utf-8")
    temp = request.get_data().decode("utf-8")
    temp = telebot.types.Update.de_json(temp)
    logger.debug('New message received. raw: %s', temp)
    bot.process_new_updates([temp])
    return "!", 200

@app.route("/set_webhook")
def webhook_on():
    password = request.args.get("password")
    if password != ADMIN_PASSWORD:
        logger.info('Set_webhook page loaded without password')
        return "<h1>Access denied!</h1>", 403

```

```

bot.remove_webhook()
url = 'https://' + os.environ.get('HOST') + '/' + WEBHOOK_TOKEN
bot.set_webhook(url=url)
logger.info(f'Webhook is ON! Url: %s', url)
return "<h1>WebHook is ON!</h1>", 200

@app.route("/remove_webhook")
def webhook_off():
    password = request.args.get("password")
    if password != ADMIN_PASSWORD:
        logger.info('Remove_webhook page loaded without password')
        return "<h1>Access denied!</h1>", 403

    bot.remove_webhook()
    logger.info('WebHook is OFF!')
    return "<h1>WebHook is OFF!</h1>", 200
bot = telebot.TeleBot(os.environ.get('BOT_TOKEN'))

# ----- flask app -----
app = Flask(__name__)

# ----- logging -----
logger = logging.getLogger("tg-bot-template")

alpha_logger = logging.getLogger()
alpha_logger.setLevel(logging.INFO)

app.logger.setLevel(logging.ERROR)
telebot.logger.setLevel(logging.ERROR)

users = [int(os.environ.get("ADMIN_ID"))]

if os.environ.get("LOG_BOT_TOKEN", '') != '':
    tg_logger.setup(alpha_logger, token=os.environ.get("LOG_BOT_TOKEN"),
users=users)
    tg_logger.setup(app.logger, token=os.environ.get("LOG_BOT_TOKEN"),
users=users)
    tg_logger.setup(telebot.logger, token=os.environ.get("LOG_BOT_TOKEN"),
users=users)

# ----- webhook -----
ADMIN_PASSWORD = os.environ.get('ADMIN_PASSWORD')
WEBHOOK_TOKEN = os.environ.get('WEBHOOK_TOKEN')

```

Лістинг setup.py

```

from flask import Flask, request
from telebot import types

# ----- uptime var -----
boot_time = time.time()
boot_date = datetime.datetime.now(tz=pytz.timezone("Europe/Moscow"))

# ----- flask config -----
ADMIN_PASSWORD = os.environ.get('ADMIN_PASSWORD')
app = Flask(__name__)

# ----- bot config -----
WEBHOOK_TOKEN = os.environ.get('WEBHOOK_TOKEN')
BOT_TOKEN = os.environ.get('BOT_TOKEN')
bot = telebot.TeleBot(BOT_TOKEN)

# ----- log -----
users = [int(os.environ.get("ADMIN_ID"))]

alpha_logger = logging.getLogger()
alpha_logger.setLevel(logging.INFO)
tg_logger.setup(alpha_logger, token=os.environ.get(
    "LOG_BOT_TOKEN"), users=users)
tg_logger.setup(app.logger, token=os.environ.get("LOG_BOT_TOKEN"),
users=users)

logger = logging.getLogger("tg-bot-template")

# ----- status webpage -----
@app.route('/')
def status():
    password = request.args.get("password")
    if password != ADMIN_PASSWORD:
        logger.info('Status page loaded without password')
        return "<h1>Access denied</h1>", 403

    return f'<h1>This is telegram bot server, ' \
        f'<a href="https://github.com/otter18/telegram-bot-  

template">templated</a> by ' \
        f'<a href="https://github.com/otter18">@otter18</a></h1>' \
        f'<p>Server uptime: {datetime.timedelta(seconds=time.time() -  

boot_time)}</p>' \
        f'<p>Server last boot at {boot_date}'

# ----- webhook -----
@app.route('/') + WEBHOOK_TOKEN, methods=['POST'])
def getMessage():
    temp = request.stream.read().decode("utf-8")
    temp = telebot.types.Update.de_json(temp)
    logger.debug('New message received. raw: %s', temp)
    bot.process_new_updates([temp])
    return "!", 200

@app.route("/set_webhook")
def webhook_on():

```

```
password = request.args.get("password")
if password != ADMIN_PASSWORD:
    logger.info('Set_webhook page loaded without password')
    return "<h1>Access denied!</h1>", 403

bot.remove_webhook()
url = 'https://' + os.environ.get('HOST') + '/' + WEBHOOK_TOKEN
bot.set_webhook(url=url)
logger.info(f'Webhook is ON! Url: %s', url)
return "<h1>WebHook is ON!</h1>", 200
```

Кафедра _ КБПЗ _ 2022 рік

Лістинг encodeMovie.py

```

import numpy as np
import pandas as pd
import torch
import torch.nn as nn
import torch.nn.parallel
import torch.optim as optim
import torch.utils.data
from torch.autograd import Variable

# Importing the dataset
movies = pd.read_csv('ml-1m/movies.dat', sep = '::', header = None, engine =
'python', encoding = 'latin-1')
users = pd.read_csv('ml-1m/users.dat', sep = '::', header = None, engine =
'python', encoding = 'latin-1')
ratings = pd.read_csv('ml-1m/ratings.dat', sep = '::', header = None, engine
= 'python', encoding = 'latin-1')

# Preparing the training set and the test set
training_set = pd.read_csv('ml-100k/ul.base', delimiter = '\t')
training_set = np.array(training_set, dtype = 'int')
test_set = pd.read_csv('ml-100k/ul.test', delimiter = '\t')
test_set = np.array(test_set, dtype = 'int')

# Getting the number of users and movies
nb_users = int(max(max(training_set[:,0]), max(test_set[:,0])))
nb_movies = int(max(max(training_set[:,1]), max(test_set[:,1])))

# Converting the data into an array with users in lines and movies in columns
def convert(data):
    new_data = []
    for id_users in range(1, nb_users + 1):
        id_movies = data[:,1][data[:,0] == id_users]
        id_ratings = data[:,2][data[:,0] == id_users]
        ratings = np.zeros(nb_movies)
        ratings[id_movies - 1] = id_ratings
        new_data.append(list(ratings))
    return new_data
training_set = convert(training_set)
test_set = convert(test_set)

# Converting the data into Torch tensors
training_set = torch.FloatTensor(training_set)
test_set = torch.FloatTensor(test_set)

# Creating the architecture of the Neural Network
class SAE(nn.Module):
    def __init__(self, ):
        super(SAE, self).__init__()
        self.fc1 = nn.Linear(nb_movies, 20)
        self.fc2 = nn.Linear(20, 10)
        self.fc3 = nn.Linear(10, 20)
        self.fc4 = nn.Linear(20, nb_movies)
        self.activation = nn.Sigmoid()
    def forward(self, x):
        x = self.activation(self.fc1(x))
        x = self.activation(self.fc2(x))
        x = self.activation(self.fc3(x))
        x = self.fc4(x)
        return x

```

```

sae = SAE()
criterion = nn.MSELoss()
optimizer = optim.RMSprop(sae.parameters(), lr = 0.01, weight_decay = 0.5)

nb_epoch = 256
for epoch in range(1, nb_epoch + 1):
    train_loss = 0
    s = 0.
    for id_user in range(nb_users):
        input = Variable(training_set[id_user]).unsqueeze(0)
        target = input.clone()
        if torch.sum(target.data > 0) > 0:
            output = sae(input)
            target.require_grad = False
            output[target == 0] = 0
            loss = criterion(output, target)
            mean_corrector = nb_movies/float(torch.sum(target.data > 0) + 1e-
10)

            loss.backward()
            train_loss += np.sqrt(loss.data[0]*mean_corrector)
            s += 1.
            optimizer.step()
    print('epoch: '+str(epoch)+' loss: '+str(train_loss/s))

test_loss = 0
s = 0.
for id_user in range(nb_users):
    input = Variable(training_set[id_user]).unsqueeze(0)
    target = Variable(test_set[id_user]).unsqueeze(0)
    if torch.sum(target.data > 0) > 0:
        output = sae(input)
        target.require_grad = False
        output[target == 0] = 0
        loss = criterion(output, target)
        mean_corrector = nb_movies/float(torch.sum(target.data > 0) + 1e-10)
        test_loss += np.sqrt(loss.data[0]*mean_corrector)
        s += 1.
print('test loss: '+str(test_loss/s))

```

Лістинг auth.py

```
import tekore as tk

def auth():
    client_id = "183b7f0b2fde45b28aa72155df8c4882"
    secret_key = "4b387b62cdbf498e94e81b4b4927399d"
    token = tk.request_client_token(client_id, secret_key)
    return tk.Spotify(token)
```

Кафедра КБПЗ 2022 рік