

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Дослідження основних підходів до побудови
Web-систем рекомендаційного типу
(тема)

Виконав:
студент 2 курсу, групи СШМ-18-1
Степанова А.О.
(прізвище, ініціали)

Спеціальність 122 – Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного
інтелекту (СШІ)
(повна назва освітньої програми)

Керівник проф. Рябова Н.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2019 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 122 – Комп'ютерні науки _____
(код і повна назва)

Тип програми освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системи штучного інтелекту (СШІ) _____
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
« ____ » _____ 2019 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові _____ Степановій Альоні Олександрівні _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Дослідження основних підходів до побудови Web-систем рекомендаційного типу _____

затверджена наказом по університету від 04 листопада 2019 р. № 1623Ст

2. Термін подання студентом роботи до екзаменаційної комісії 19 ____ 12 ____ 2019 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо розробки та дослідження рекомендаційних систем _____

4. Перелік питань, що потрібно опрацювати в роботі Вступ, 1 Дослідження принципів побудови рекомендаційних систем, 1.1 Основні поняття і історія виникнення рекомендаційних систем, 1.2 Аналіз основних підходів до вироблення рекомендацій, 1.2.1 Колаборативна фільтрація, 1.2.2 Методи, що аналізують зміст об'єктів, 1.2.3 Методи, засновані на знаннях, 1.2.4 Гібридні методи, 2 Аналіз предметної області і постановка задач дослідження, 2.1 UX-дизайн: головні положення, правила та етапи створення, 2.2 Класифікація сайтів по типу, за призначенням та за структурою, 2.3 Постановка задач дослідження, 3 Розробка системи рекомендаційного типу для прототипування інтерфейсів, 3.1 Існуючі системи прототипування веб-інтерфейсів, 3.2 Побудова діаграми прецедентів для опису функціоналу системи, 3.3 Розробка базового функціоналу РС для прототипування інтерфейсів, 4 Аналіз підходів до тестування usability інтерфейсів, Висновки, Перелік посилань, Додаток А, Додаток Б _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Рисунок 1 – Матриця оцінок, Рисунок 2 – Загальна структура методу колаборативної фільтрації, Рисунок 3 – Класифікація методів колаборативної фільтрації, Рисунок 4 – UX-компоненти, Рисунок 5 – Айсберг UX/UI, Рисунок 6 – Прототип сайту-візитки, Рисунок 7 – Приклад корпоративного сайту, Рисунок 8 – Веб-інтерфейс сайту ХНУРЕ, Рисунок 9 – Веб-інтерфейс порталу Яндекс.Метрика, Рисунок 10 – Приклад інтернет-магазину, Рисунок 11 – Картка товару в інтернет-магазині, Рисунок 12 – Корзина в інтернет-магазині, Рисунок 13 – Техніка AIDA, Рисунок 14 – Блок Attention з конкретною дією, Рисунок 15 – Фрагмент блоку Interest, Рисунок 16 – Фрагмент блоку Desire, Рисунок 17 – Фрагмент блоку Action, Рисунок 18 – Фрагмент блоку Confidence, Рисунок 19 – Форум графічних та UX/UI дизайнерів, Рисунок 20 – Сайт-блог, Рисунок 21 – Сервіс для генерації релевантних ключів, Рисунок 22 – Інтерфейс інструменту Axure, Рисунок 23 – Інтерфейс інструменту Origami, Рисунок 24 – Інтерфейс інструменту Proto.io, Рисунок 25 – Інтерфейс інструменту Vectr, Рисунок 26 – Інтерфейс інструменту InVision, Рисунок 27 – Діаграма прецедентів системи

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Рябова Н.В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	04.11.19	виконано
2	Аналіз предметної області	05.11.19-10.11.19	виконано
3	Постановка завдання та узгодження з керівником	11.11.19-13.11.19	виконано
4	Дослідження методів до побудови рекомендаційних систем	14.11.19-25.11.19	виконано
5	Аналіз підходів до тестування інтерфейсу	26.11.19-30.11.19	виконано
6	Розробка прототипу системи	1.12.19-5.12.19	виконано
7	Написання пояснювальної записки	6.12.19-12.12.19	виконано
8	Попередній захист	16.12.19	виконано
9	Захист перед ЕК	19.12.19	

Дата видачі завдання 04 листопада 2019 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Рябова Н.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 85 с., 36 рис., 9 табл., 2 дод., 23 джерела.

ДИЗАЙН ІНТЕРФЕЙСІВ, ДІАГРАМА ПРЕЦЕДЕНТІВ,
КЛАСТЕРИЗАЦІЯ, КОЕФІЦІЄНТ ДАЙСА, КОЛАБОРАТИВНА
ФІЛЬТРАЦІЯ, КОНТЕНТНА ФІЛЬТРАЦІЯ, МАШИННЕ НАВЧАННЯ,
РЕКОМЕНДАЦІЙНІ СИСТЕМИ, СИНГУЛЯРНИЙ РОЗКЛАД МАТРИЦІ,
ШТУЧНИЙ ІНТЕЛЕКТ, USABILITY ТЕСТУВАННЯ

Об'єкт дослідження – інтелектуальні веб-системи.

Предмет дослідження – рекомендаційні системи.

Мета роботи – провести аналіз існуючих методів до побудови інтелектуальних веб-систем рекомендаційного типу.

Методи дослідження – колаборативна фільтрація, контентна фільтрація, сингулярний розклад матриці, кластерування.

РЕФЕРАТ

Пояснительная записка: 85 с., 36 рис., 9 табл., 2 прил., 23 источника.

ДИАГРАММА ПРЕЦЕДЕНТОВ, ДИЗАЙН ИНТЕРФЕЙСОВ, КЛАСТЕРИЗАЦИЯ, КОЛЛАБОРАТИВНАЯ ФИЛЬТРАЦИЯ, КОНТЕНТНАЯ ФИЛЬТРАЦИЯ, КОЭФФИЦИЕНТ ДАЙСА, МАШИННОЕ ОБУЧЕНИЕ, РЕКОМЕНДАТЕЛЬНЫЕ СИСТЕМЫ, СИНГУЛЯРНОЕ РАЗЛОЖЕНИЕ МАТРИЦЫ, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, USABILITY ТЕСТИРОВАНИЕ

Объект исследования – интеллектуальные веб-системы.

Предмет исследования – рекомендательные системы.

Цель работы – провести анализ существующих методов к построению интеллектуальных веб-систем рекомендательного типа.

Методы исследования – коллаборативная фильтрация, контентная фильтрация, сингулярное разложение матрицы, кластеризация.

ABSTRACT

Explanatory note: 85 p., 36 fig., 9 tabl., 2 ann., 23 sources.

ARTIFICIAL INTELLIGENCE, CLUSTERIZATION, CONTENT-BASED FILTERING, COLLABORATIVE FILTERING, DICE COEFFICIENT, INTERFACE DESIGN, MACHINE LEARNING, PRECEDENCE DIAGRAM, RECOMMENDATION SYSTEMS, SINGULAR VALUE DECOMPOSITION, USABILITY TESTING

Object of study – intelligent web systems.

Subject of study – recommendation systems.

Purpose of work – to analyze existing methods for constructing advisory-type intelligent web systems.

Research methods – collaborative filtering, content filtering, singular matrix decomposition, clustering.

СОДЕРЖАНИЕ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	...8
Вступ
Ошибка! Закладка не определена.	
1 Дослідження принципів побудови рекомендаційних систем 11
1.1 Основні поняття і історія виникнення рекомендаційних систем	11
1.2 Аналіз основних підходів до вироблення рекомендацій.....	14
1.2.1 Колаборативна фільтрація.....	15
1.2.1 Методи, що аналізують вміст об'єктів	23
1.2.3 Методи, які базуються на знаннях.....	26
1.2.4 Гібридні методи	31
2 Аналіз предметної області та постановка завдання 34
2.1 UX-дизайн: головні положення, правила та етапи створення	34
2.2 Класифікація сайтів за типом, призначенням та структурою	42
2.3 Постановка задач дослідження	58
3 Розробка системи рекомендаційного типу для прототипування інтерфейсів.....	59
3.1 Існуючі системи прототипування веб-інтерфейсів.....	59
3.2 Побудова діаграми прецедентів для опису функціоналу системи	62
3.3 Розробка базового функціоналу РС для прототипування інтерфейсів	65
4 Аналіз основних підходів до тестування usability інтерфейсів 71
Висновки 76
Перелік джерел посилання
Ошибка! Закладка не определена.	
Додаток А
Ошибка! Закладка не определена.	
Додаток Б
Ошибка! Закладка не определена.	

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних;

СР – системи рекомендацій;

РС – рекомендаційні системи;

ЗМІ – засоби масового інформування;

ЧПУ – числове програмне керування;

AMP – Accelerated Mobile Pages – оптимізація мобільних сторінок;

B2B – сектор ринку, який працює не на кінцевого споживача, а на інший бізнес;

B2C – форма електронної торгівлі, метою якої є прямі продажі для споживача;

RMSE – Root Mean Square Error – середньоквадратична похибка;

SEO – Search Engine Optimization – пошукова оптимізація;

UI – User Interface Design – дизайн інтерфейсів;

UX – User Experience Design – досвід користувача;

WWW – World Wide Web – всемережжя.

ВСТУП

За тридцять років після появи World Wide Web значно зросли обсяги інформації, з якими нам доводиться працювати щодня. Люди повинні орієнтуватися серед надзвичайно великої кількості доступних альтернатив, коли роблять який-небудь вибір в ролі споживачів. З іншого боку виступають власники інтернет-магазинів і різних сервісів: вони зацікавлені в персональній рекламі і рекомендаціях кожному конкретному користувачеві, тому що такий підхід суттєво збільшить прибуток їхніх компаній. В результаті чого і були впроваджені рекомендаційні системи, мета яких – допомогти людині здійснювати всі ці дії найбільш швидко і ефективно.

На сьогоднішній час рекомендаційні системи вже мають досить широке застосування. В першу чергу, вони активно використовуються в інтернет-комерції. Сервіси збирають інформацію про уподобання користувачів і пропонують їм товари або послуги, які можуть їх зацікавити. Яскравими прикладами є Amazon, eBay, iTunes та інші. Інше важливе застосування даних систем – допомога користувачам у виборі книг, музики, подкастів та іншого контенту. Наприклад: Яндекс.Дзен – рекомендаційна стрічка контенту і платформа для блогерів, яка автоматично підлаштовується під інтереси користувача.

Існує безліч підходів до побудови таких систем, але всі вони мають свої переваги і недоліки. У наш час інформаційного хаосу рекомендаційні системи це вже не просто зручний інструмент, а необхідність. Саме тому дослідження в цій галузі залишаються актуальними і, більш того, зростають з кожним днем.

В рамках цієї роботи пропонується застосувати рекомендаційні системи в якості допомоги фахівцям області дизайну інтерфейсів, а також звичайним користувачам, яким потрібен прототип власного майбутнього веб-сайту або чітке технічне завдання для дизайнера.

Головні завдання, які треба вирішити:

- розглянути основні підходи до розробки інтелектуальних веб-систем рекомендаційного типу;
- описати процес роботи і обов'язки дизайнера;
- з'ясувати, які існують проблеми при прототипуванні інтерфейсів;
- виділити основні типи веб-ресурсів та чим вони відрізняються;
- розробити прототип системи, що видає рекомендації щодо покрокового проектування інтерфейсу в залежності від заданих критеріїв;
- проаналізувати підходи до тестування usability інтерфейсів.

1 ДОСЛІДЖЕННЯ ПРИНЦИПІВ ПОБУДОВИ РЕКОМЕНДАЦІЙНИХ СИСТЕМ

1.1 Основні поняття і історія виникнення рекомендаційних систем

Зараз ми сприймаємо складні самонавчаючі комп'ютерні програми як даність, використовуючи для їх опису такі терміни, як Big Data, штучний інтелект і рекомендаційні системи. Однак, починалося все з машинного навчання. У цій галузі науки були сформульовані математичні підходи та описані алгоритми, які до теперішнього часу лежать в основі всіх рішень. Наприклад, наївний баєсів класифікатор або штучна нейронна мережа.

Власне, рекомендаційні системи – це програми, головна мета яких полягає у формуванні рекомендацій різних об'єктів для користувачів на основі їх переваг. Будь-яку рекомендаційну систему: можна описати наступним чином: предмет рекомендації, мета рекомендації, контекст рекомендації, джерело рекомендації, ступінь персоналізації, прозорість, формат рекомендації та алгоритми.

Перший етап розвитку рекомендаційних систем – колаборативна фільтрація. Перший цифровий відеомагнітофон, який намагався врахувати переваги користувачів, щоб автоматично записувати відповідні їм передачі, з'явився в кінці дев'яностих. Пристрій називався TiVo і продавався кінцевим користувачам такими брендами, як Sony і Philips. У той час добре структурованих описів для всіх транслуючих телепередач не було, і потужності пристроїв для складних обчислень теж не вистачало, тому в основу системи рекомендацій був покладений алгоритм колаборативної фільтрації.

Колаборативна фільтрація ґрунтується виключно на схожості поведінки користувачів і не використовує ніяких описів телепередач і фільмів або іншої семантики. Якщо на двох пристроях TiVo в двох різних

домогосподарствах глядачі зазвичай вибирали одні і ті ж самі програми, а потім перший користувач починав регулярно дивитися будь-яку нову телепередачу, то цю телепередачу рекомендували і другому користувачеві. Всі пристрої TiVo вночі відправляли дані про послідовність кліків на центральний сервер, потім рекомендаційний алгоритм обчислював персональні рекомендації і вся ця інформація завантажувалася на пристрої. Така організація процесу дозволяла, по-перше, порівнювати і знаходити глядачів зі схожими смаками, а по-друге, не перевантажувати відеомагнітофони TiVo з їх обмеженими обчислювальними можливостями.

Незважаючи на те, що рішення не працювало в реальному часі і іноді неправильно визначало інтереси користувача, простота підходу швидко зробила його популярним у галузі. Чемпіоном застосування стала компанія Netflix, яка залучила команду розробників, які налаштовували параметри рекомендаційного движка колаборативної фільтрації, тим самим досягаючи з року в рік все кращих результатів. Компанія тоді давала в прокат не цифрові копії фільмів, а розсилала VHS-касети і DVD. Для неї було дуже важливо підвищити якість рекомендацій, тому що чим краще Netflix рекомендує своїм користувачам фільми, тим більше вони беруть потім в прокат. Відповідно, зростає і прибуток компанії.

Рекомендаційні системи на той момент існували близько 20 років, проте справжній підйом стався завдяки компанії Netflix, яка в надії врятувати ситуацію за рахунок колективного розуму в 2006 році організувала конкурс Netflix Prize. Вони виклали у відкритий доступ зібрані дані: близько ста мільйонів оцінок за п'ятибальною шкалою з зазначенням ID користувачів, які брали в цьому участь. Учасники змагання повинні були якомога краще передбачати, яку оцінку поставить певному фільму той чи інший користувач. Якість передбачення вимірювалася за допомогою метричних показників RMSE, тобто середньо-квадратичне відхилення. У Netflix вже був алгоритм, який передбачав оцінки з якістю 0.9514 по даним метричним показникам. Завданням було покращити

прогнозування хоча б на 10% – до 0.8563. Кращі команди розробників протягом трьох років домоглися цього результату, але запропоновані рішення використовувати на практиці було не можна – навіть невелике покращення точності вимагало абсолютно неадекватних операційних витрат. Колаборативна фільтрація, як основний метод рекомендацій, зайшла в глухий кут.

Другий етап розвитку рекомендаційних систем – контентна фільтрація. До процесу розвитку нових підходів до вирішення завдання долучилися нові компанії, зокрема Jinni і Argico, які зрозуміли, що якісна інформація про рекомендований контент могла б служити серйозною основою для тонкого пристосування до інтересів глядачів. Ця нова хвиля досліджень в основному використовувала класифікацію Баеса, яка дозволяла зрозуміти і оцінити, чому користувач віддає перевагу тому чи іншому контенту. У Netflix відразу оцінили перспективи нового напрямку семантичного аналізу описів і знову вирішили зайнятися власною розробкою. Як і Jinni, вони стали створювати опис і класифікатори для фільмів свого каталогу. Цей підхід мав низку проблем, пов'язаних з масштабуванням, оскільки він доцільний і виправданий, тільки якщо каталог відео відносно малий і збільшується повільно. Решта компаній, що займаються семантичними рекомендаційними рішеннями, сфокусувалися на класифікації контенту на основі метаданих, які існували на ринку. Netflix знову відзначився. Його каталог жанрів швидко виріс з початкових 560 варіантів до 93 116, за даними на липень 2014 року, серед яких були, наприклад, такі як «емоційні драми 80-х, які отримали високу оцінку кінокритиків», куди до каталогу Netflix потрапляло лише два фільми. І це знову було занадто – категорії, яким відповідала незначна кількість об'єктів, не були придатні для аналізу і ставали незначними для класифікатора Баеса.

Третій етап розвитку рекомендаційних систем – аналіз контексту. Тим часом розвиток хмарних обчислень надав компаніям, які розробляють

рекомендації, чималі можливості для проведення натурних тестів. На їх основі можна були отримати чітку інформацію про те, в якому вигляді люди хочуть отримувати рекомендації, як вони реагують і який зворотний зв'язок готові надавати. Виявилось, що вимоги користувачів жоден конкретний рекомендаційний алгоритм задовольнити не може. Їм була потрібна комбінація редакторських рекомендацій, фільтрації відповідно контенту на основі персональних переваг, рекомендацій спільноти людей з подібними смаками, і щоб в цих рекомендаціях були вдалі знахідки, а підбір контенту відповідав миттєвому настрою і типу використовуваного пристрою. При цьому користувач повинен розуміти, що він сам може вплинути на результат. Складні налаштування пропонувати не потрібно. Наприклад, співвідношення між редакторськими рекомендаціями і фільтрацією відповідно контенту визначає оператор, але якщо поруч з рекомендованим фільмом зазначено, що даний фільм вам радять подивитися бо вам подобається актор в головній ролі, а це хибне свідчення, то потрібно мати можливість відразу це зазначити і негайно отримати інші рекомендації. Таким чином, після періоду вивчення кліків користувача і семантичних рекомендацій прийшов новий етап, в якому основну роль відігравав контекст: миттєві наміри, інтереси та оточення користувача.

1.2 Аналіз основних підходів до вироблення рекомендацій

Основне завдання системи рекомендацій – маючи дані про об'єкти і користувачів, отримати список об'єктів, найбільш цікавих для конкретного користувача. Залежно від того, які дані використовуються для розрахунку рекомендацій, системи діляться на три великі класи: методи колаборативної фільтрації; методи, які аналізують зміст об'єктів і методи, які базуються на знаннях.

Методи колаборативної фільтрації припускають, що інтереси

користувачів представлені оцінками, які вони надають об'єктам після взаємодії з ними. Основна ідея цих методів полягає в порівнянні між собою інтересів різних користувачів або об'єктів на основі цих оцінок, при цьому ніякої додаткової інформації про самих користувачів і об'єкти не використовується.

Методи другого класу, навпаки, використовують зміст об'єктів для отримання рекомендацій. Ці методи працюють в тих випадках, коли зміст об'єктів подано у вигляді текстів. Вони добре підходять для рекомендації книг. Також, їх можна використовувати для порівняння назв, описів та іншої текстової інформації.

Методи, засновані на знаннях, вимагають від користувача описати свої вимоги до потрібних йому об'єктів, а потім шукають з використанням своєї бази знань об'єкти, що задовольняють поставлені вимоги.

1.2.1 Колаборативна фільтрація

В рамках цього підходу ми будемо рекомендації для кожного користувача на підставі оцінок схожих користувачів. Для складання якісних рекомендацій використовуються методи явного або неявного збору даних:

- при явному зборі від користувача необхідно отримати заповнену анкету для виявлення переваг. Недолік методу в тому, що досить складно змусити користувача поставити оцінку;

- при неявному зборі протоколюються дії користувача: що користувач подивився, який товар доданий в кошик, що прокоментував, яку покупку зробив. Складання рейтингів відбувається автоматично. Недолік методу – невизначеність: якщо користувач подивився товар, невідомо, чи сподобався він йому чи ні; якщо користувач не купив товар – знову ж таки невідомо, чим було зумовлене таке рішення.

Чим більше оцінок, тим вище точність рекомендацій. Розглянемо

алгоритм User-based, заснований на порівнянні користувачів між собою. Нехай деякі користувачі оцінили деякі об'єкти, і нехай оцінка – це натуральне число від 1 до 5. Тоді всі оцінки можна зобразити у вигляді матриці (рисунок 1.1).

		<i>Об'єкти</i>												
		1	2	...	i	...	m							
<i>Користувачі</i>	1	5	3		1	2								
	2		2				4							
	:			5										
	u	3	4		2	1								
	:					4								
	n			3	2									
		<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">a</td> <td style="border: 1px solid black; text-align: center;">3</td> <td style="border: 1px solid black; text-align: center;">5</td> <td style="border: 1px solid black;"></td> <td style="border: 1px solid black; text-align: center;">?</td> <td style="border: 1px solid black;"></td> <td style="border: 1px solid black; text-align: center;">1</td> </tr> </table>						a	3	5		?		1
a	3	5		?		1								

Рисунок 1.1 – Матриця оцінок

Нехай є користувач а. Наше завдання – передбачити, яку б оцінку поставив цей користувач об'єкту і. Будемо розглядати тільки користувача а і тих користувачів, які оцінили об'єкт і. Алгоритм включає в себе наступні кроки:

- для кожного користувача u обчислимо, наскільки його інтереси збігаються з інтересами користувача а;
- після цього виберемо множину користувачів, найбільш близьких до а;
- передбачимо оцінку на основі оцінок об'єкта і сусідами з попереднього кроку.

Перший крок. Кожному користувачеві в матриці оцінок відповідає

рядок, тому будемо обчислювати близькість векторів користувачів. Існує безліч способів, один з яких – порахувати косинус між цими векторами:

$$sim(u, a) = \frac{\sum_{i=1}^m r_{a,i} \cdot r_{u,i}}{\sqrt{\sum_{i=1}^m r_{a,i}^2} \cdot \sqrt{\sum_{i=1}^m r_{u,i}^2}}, \quad (1.1)$$

де $sim(u, a)$ – міра схожості користувачів a і u ;

$r_{u,i}$ – значення матриці оцінок: u – рядок, i – стовпець.

$sim(u, a)$ приймає значення з відрізка $[0, 1]$. Якщо користувач не вказав оцінку для якогось об'єкта, відповідне значення матриці дорівнює 0.

Існує інша міра близькості між векторами – коефіцієнт кореляції Пірсона:

$$sim(u, a) = \frac{\sum_{i \in I} (r_{a,i} - \bar{r}_a) \cdot (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in I} (r_{a,i} - \bar{r}_a)^2} \cdot \sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2}}, \quad (1.2)$$

де I – множина об'єктів, які оцінив як користувач a , так і користувач u ;

\bar{r}_a і \bar{r}_u – середні з оцінок користувачів a і u , відповідно.

У чисельнику підраховується добуток відхилень оцінок двох користувачів від середніх значень для одного об'єкта. Знаменник необхідний для того, щоб величина приймала значення з відрізка $[-1, 1]$. Чим краще збігаються інтереси, тим ближче значення близькості до 1 користувачів протилежні. Якщо коефіцієнт кореляції Пірсона негативний, то інтереси користувачів протилежні. Коефіцієнт кореляції Пірсона буде залишатися високим, якщо всі оцінки порівнюваних користувачів відрізняються на постійну величину. Косинус в цьому випадку видасть менше значення. Однак, якщо оцінки двох користувачів відрізняються пропорційно, то косинус буде більше, а кореляція – менше. В обох випадках оцінки схожі, тому вибір метричних показників залежить від

завдання.

Другий крок. Тепер потрібно вибрати множину K найбільш схожих на a користувачів. Можна вибрати всіх, але так як користувачів з несхожими інтересами досить багато, то вони будуть мати негативний вплив на точність передбачення оцінки для об'єкта i . Крім того, кількість користувачів впливає на обсяг обчислень на третьому кроці. Одне з можливих рішень – встановити поріг міри близькості, обчисленої на першому кроці. Користувачі з мірою близькості, що перевищує поріг, увійдуть в множину K . Решта – ні. Найчастіше вибирається ціла константа k , а потім всі користувачі упорядковуються відповідно зі зменшенням міри близькості. У підсумку, в множину K входять k користувачів, найбільш близьких до a .

Третій крок. Маючи множину K близьких користувачів потрібно обчислити оцінку, яку поставив би користувач a об'єкту i . Нагадаємо, що розглядаються тільки ті користувачі, які оцінили об'єкт i . Потрібна оцінка обчислюється за формулою:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u \in K} (r_{u,i} - \bar{r}_u) \times \text{sim}(a, u)}{\sum_{u \in K} |\text{sim}(a, u)|}, \quad (1.3)$$

де $p_{a,i}$ – це пророкуюча оцінка користувача a для об'єкта i .

За основу береться його середня оцінка \bar{r}_a , а потім додається середнє відхилення оцінки інших користувачів з множини K для об'єкта i від їх середньої оцінки. Чим ближче користувач u до користувача a , згідно міри близькості $\text{sim}(a, u)$, тим сильніше його внесок в прогнозування оцінки. Таким чином, описаний алгоритм прогнозує оцінки для об'єктів, які поточний користувач ще не оцінив. Для формування рекомендації для даного користувача, досить передбачити оцінки для всіх неоцінених об'єктів і вибрати об'єкти з найкращо передбаченою оцінкою.

Преваги User-based алгоритму: є більш універсальним підходом;

часто дає кращі результати; проста реалізація і інтерпретація.

Недоліки User-based алгоритму: проблема холодного старту, тобто виникає проблема видачі релевантних рекомендацій для нових користувачів або об'єктів; розріджена матриця рейтингів; яка не масштабується.

Існує аналогічний підхід, але замість користувачів ми будемо порівнювати об'єкти. Даний алгоритм називається Item-based і включає в себе наступні етапи:

- для кожного об'єкта j обчислимо, наскільки він схожий на об'єкт i , для якого передбачається оцінка;
- виберемо множину об'єктів, найбільш близьких до i ;
- передбачимо оцінку на основі оцінок, обраних на другому кроці об'єктів користувачем a .

$$sim(i, j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i) \cdot (r_{u,j} - \bar{r}_j)}{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2 \cdot \sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}, \quad (1.4)$$

де U – множина U користувачів, які оцінили об'єкти i та j ;

\bar{r}_i – середня оцінка об'єкта i .

Другий крок аналогічний другому кроку алгоритму User-based. Відмінність полягає в тому, що K – множина близьких до i об'єктів.

На третьому кроці рахується зважена середня оцінка схожих об'єктів, отримана користувачем a :

$$P_{a,i} = \frac{\sum_{j \in K} r_{a,j} \times sim(i, j)}{\sum_{j \in K} (r_{a,i} - r_a)^2 \cdot \sum_{i \in I} |sim(i, j)|}, \quad (1.5)$$

Переваги Item-based алгоритму: інтереси користувачів часто змінюються, а середня оцінка об'єкта залишається постійною, тому при порівнянні об'єктів досить поррахувати міру близькості кожного з кожним i

зберегти в пам'яті. В частому оновленні цих даних немає необхідності, а міри можна порахувати навіть коли в системі немає активних користувачів. Також, зниження розмірності дозволяє вирішити проблему розрідженості даних.

Недоліки Item-based алгоритму: низький рівень інтерпретації моделі через приховані чинники.

У сучасних системах рекомендацій число користувачів і об'єктів постійно зростає, тому виникає необхідність використовувати розподілену архітектуру системи. Відразу виникає проблема зберігання даних: було б логічно схожих користувачів або схожі об'єкти зберігати поруч. Для її вирішення можна застосувати алгоритми кластеризації для користувачів або об'єктів, використовуючи міру близькості між ними. Розбиття користувачів або об'єктів на кластери дає ще одну перевагу: на першому кроці потрібний об'єкт достатньо порівняти з об'єктами зі свого кластера, що значно скорочує кількість обчислень.

Описані вище методи колаборативної фільтрації називаються методами, заснованими на роботі з пам'яттю, так як необхідно зберігати матрицю оцінок. Існує інша група алгоритмів – методи, засновані на моделях. Ідея полягає в тому, що маючи матрицю оцінок можна побудувати модель, яка буде менше за обсягом займаної пам'яті. Використовуючи отриману модель, можна буде обчислювати рекомендації. Прикладами таких моделей є алгоритм Басса, при якому початкова задача зводиться до задачі класифікації; а також Марковські моделі, які дозволяють враховувати не тільки схожість об'єктів, але і їх цінність.

Наприклад: ви створюєте веб-сайт, щоб пропонувати його відвідувачам рекомендації щодо подкастів. На основі інформації від багатьох користувачів, які підписуються на різні теми, ви можете створити групу по перевагам. Далі ви ідентифікуєте найпопулярніші теми серед тих, які слухають учасники цієї групи, а потім рекомендуєте їх конкретному

користувачеві, який ще не підписаний на них і не слухав подкастів на ці теми.

У таблиці 1.1 рядки відповідають набору тем подкастів, а стовпці – користувачам. Осередок на перетині рядка теми подкасту і стовпця користувача містить кількість подкастів, прослуханих цим користувачем на дану тему. Кластеризація користувачів на такій основі дозволяє виділити два кластери, кожен з яких містить по два користувача. Цю процедуру можна виконати використовуючи, наприклад, алгоритм найближчого сусіда. Власне, потрібно звернути увагу на схожість переваг у членів кожного кластера. Кластер 1 утворюють учасники з іменами Володимир і Марина, кожен з яких прослухав по кілька подкастів на тему «Самооцінка» і тему «Дисципліна». Кластер 2 утворюють учасники з іменами Іра і Юра, кожен з яких прослухав по кілька подкастів на тему «Емоційний інтелект» і тему «Мінімалізм».

Таблиця 1.1 – Приклад колаборативної фільтрації

	Володимир, кластер 1	Іра, кластер 2	Марина, кластер 1	Юра, кластер 2
Самооцінка	13	3	11	–
Книги	10	–	–	3
Дисципліна	6	1	9	–
Емоційний інтелект	–	6	–	9
Мінімалізм	–	7	1	8

Тепер можна ідентифікувати певні відмінності в рамках кожного

кластера і сформувані значущі рекомендації. У кластері 1 Володимир прослухав 10 подкастів на тему «Книги», а Марина не прослухала жодного такого подкасту; Марина прослухала один подкаст на тему «Мінімалізм», а Володимир не прослухав жодного такого подкасту. Таким чином, Марині можна порекомендувати подкаст на тему «Книги». Для Володимира неможливо сформувані ніяких рекомендацій, оскільки невелика різниця між ним і Мариною за кількістю прослуховування подкастів на тему «Мінімалізм» з великою ймовірністю буде відфільтрована. У кластері 2 Юра прослухав 3 подкасти на тему «Книги», а Іра не прослухала жодного такого подкасту. Іра прослухала 3 подкасти на тему «Самооцінка», а Юра не прослухав жодного такого подкасту. Таким чином, для учасників кластера 2 можна сформувані наступні дві рекомендації: Юрі рекомендується подкаст на тему «Самооцінка», а Ірі – подкаст на тему «Книги».

Для узагальнення отриманої інформації на рисунку 1.2 і рисунку 1.3 представлені схеми [1].



Рисунок 1.2 – Загальна структура методу колаборативної фільтрації

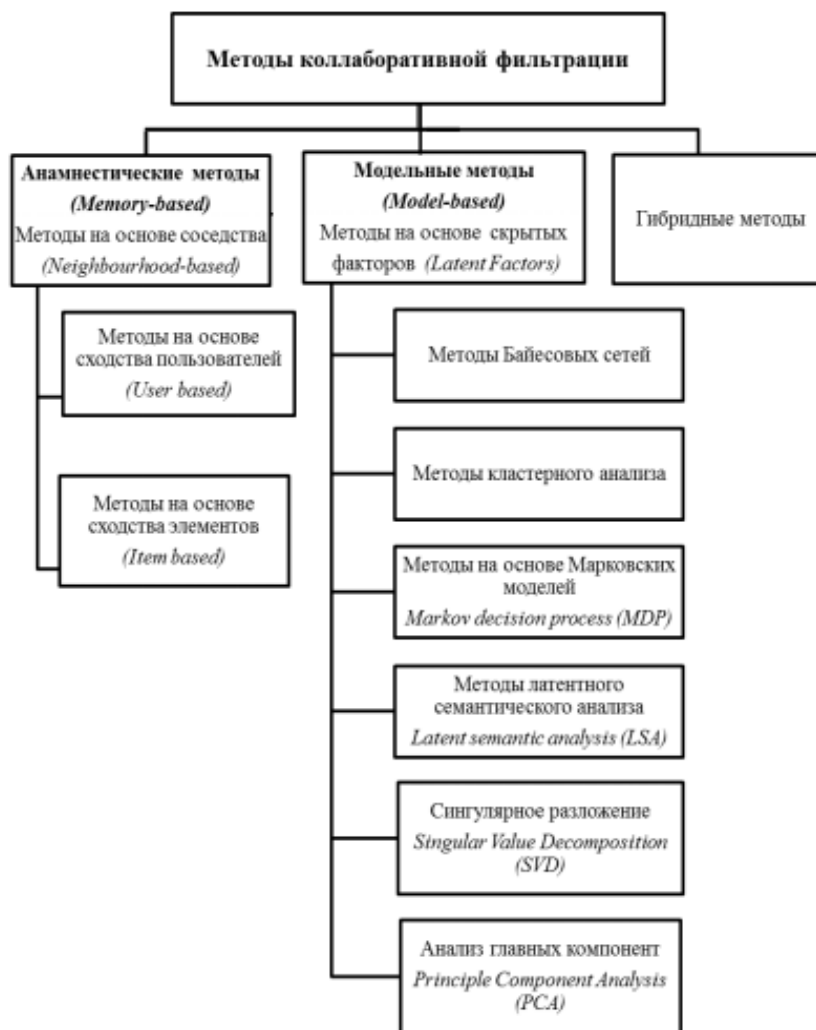


Рисунок 1.3 – Класифікація методів колаборативної фільтрації

1.2.1 Методи, що аналізують вміст об'єктів

У випадках, коли колаборативна фільтрація не дає результату, доводиться використовувати інформацію про об'єкти. В рамках цього підходу ми знову працюємо з множиною користувачів і множиною об'єктів, однак сам підхід до побудови рекомендацій сильно відрізняється. Нам потрібно уявити кожен об'єкт з множини і у вигляді вектора ознак, які залежать від предметної області, в якій ми працюємо. Скажімо, якщо ми працюємо з рекомендаційною системою для кіно, то нам потрібні ознаки,

які дозволяють описати кожен фільм. Припустимо, про кожен фільм ми швидше за все знаємо: жанр, режисера, рік випуску та акторський склад. Саме ці чинники і можуть бути ознаками.

Тепер нам потрібно описати користувача, адже про нього ми теж знаємо достатню кількість інформації. Як мінімум, ми розуміємо як наші користувачі взаємодіяли з рекомендаційною системою. Наприклад, вийшло продовження якогось фільму, яке додається в систему. Виникає питання: кому його рекомендувати. Колаборативна фільтрація відповідь не дасть через проблему «холодного старту», але є сенс рекомендувати новий фільм тим, хто подивився першу частину фільму. Відповідно, для кожного користувача ми можемо розрахувати ознаки, пов'язані з його поведінкою. Наприклад: улюблена категорія продуктів, улюблені фільми, улюблені жанри та інші. Також, для кожної пари користувач-об'єкт ми можемо розрахувати ознаки, що описують їх взаємодію. Наприклад: чи переглядав раніше користувач цей продукт, як часто він купував такі продукти, дата останньої покупки і частота покупок з цієї ж категорії. Власне, ми можемо обчислити три групи ознак: ознаки, що описують товар; ознаки, що описують користувача і парні ознаки.

Тепер ми можемо зібрати навчальну вибірку, що описує кожну пару користувач-об'єкт. Єдине питання, яке залишилося – що буде виступати в ролі цільової функції, тобто що ми будемо прогнозувати. Давайте будемо прогнозувати ті самі оцінки, які користувачі дають об'єктам. Таким чином, отримавши навчальну вибірку, ми отримали можливість побудувати модель класифікації або регресії, яка наблизитиме оцінки користувачів. Припущення, яке ми робимо за ознаками, які ми виділили раніше – за ознаками об'єктів, за ознаками користувачів і за парними ознаками – ми зможемо оновити переваги користувачів. Далі, застосовуючи отриману модель до невідомих пар користувач-об'єкт, ми будемо наблизати ті оцінки, які користувач поставив би цьому об'єкту і на основі таких оновлених оцінок – будувати рекомендації.

Розглянемо конкретний приклад. Кожен об'єкт в будь-якій системі рекомендацій має назву, а також іншу текстову інформацію, яку можна використовувати. У таблиці 1.2 представлений приклад доступної інформації про фільми на інтернет-сервісі.

Таблиця 1.2 – Інформація про фільми на онлайн-ресурсі

Назва	Жанр	Режисер	Кінокомпанія	Рейтинг	Ключові слова
Одного разу в Америці	кримінал, драма	Серджіо Леоне	The Ladd Company, Rafran Cinematografica	8,7 / 10	кримінал, дружба, гроші драма
Їж , молися, кохай	драма, мелодрама	Райан Мерфі	Plan B Entertainment	8,2 / 10	драма, мелодрама, любов, медитація
Ірландець	драма	Мартін Скорсезе	Tribeca Productions	9,5 / 10	історичний, кримінал, драма

Користувачі системи мають свій профіль. Там вони вказують інформацію про себе і свої інтереси: улюблені жанри, улюблених режисерів та інше. Маючи ці дані, а також інформацію про фільми з таблиці 1.2 можна зробити висновки – чи цікавий конкретний фільм користувачеві. В цьому випадку, міра схожості інтересів користувача і фільму приймає два можливих значення: 1 – якщо жанр фільму є в профілі користувача, 0 – в іншому випадку.

Для більш точних рекомендацій використовується інший підхід: знаючи інформацію про фільми, переглянуті користувачем, йому можна порадити схожі фільми. Схожість вимірюється за допомогою порівняння даних з таблиці 1.2. Наприклад: якщо $keywords(F_i)$ – множина ключових слів фільму F_i , то порівняти його з фільмом F_j можна наступним чином:

$$dice(F_i, F_j) = \frac{|keywords(F_i) \cap keywords(F_j)|}{|keywords(F_i) \cup keywords(F_j)|}, \quad (1.6)$$

За допомогою цієї формули можна розрахувати коефіцієнт Дайса, який приймає значення з відрізка $[0,1]$ і показує наскільки близькі дві множини.

1.2.3 Методи, які базуються на знаннях

В рамках даного підходу ми знову представляємо об'єкти у вигляді векторів ознак, а далі на основі знань про те як ці об'єкти співвідносяться з інтересами користувача – будуємо базу знань, яка за допомогою правил ці співвідношення описує. Потім, на основі переваг користувачів, знову оцінюється корисність об'єктів за цими правилами і на підставі цієї корисності будується рекомендація.

Методи колаборативної фільтрації і аналізу вмісту добре працюють, коли людина досить часто купує товари або слухає музику. Однак, якщо мова йде про більш рідкісні випадки, то такі методи не допоможуть, тому що система не має потрібної кількості інформації. Для вирішення такого завдання і використовуються методи, що базуються на знаннях. Вони діляться на дві групи: використання жорстких обмежень та вибір близьких об'єктів.

Ідея обох методів наступна: користувачі формують свої вимоги до товару, а система намагається знайти потрібний товар. У першому випадку

шукаються і рекомендуються тільки ті об'єкти, які точно відповідають всім вимогам користувача. У другому випадку шукаються об'єкти з характеристиками, близькими до вимог.

Розглянемо конкретний приклад. У таблиці 1.3 наведені камери та їхні характеристики.

Таблиця 1.3 – Фотокамери в інтернет-магазині

Номер	Ціна	Мегапікселі	Оптичний зум	Розмір екрану	Відео	Звук	Захист від пилу і води
1	148	8.0	4x	2.5	Ні	Ні	Так
2	182	8.0	5x	2.7	Так	Так	Ні
3	189	8.0	10x	2.5	Так	Так	Ні
4	196	10.0	12x	2.7	Так	Ні	Так
5	151	7.1	3x	3.0	Так	Так	Ні
6	199	9.0	3x	3.0	Так	Так	Ні
7	259	10.0	3x	3.0	Так	Так	Ні
8	279	9.1	10x	3.0	Так	Так	Так

Завдання системи – обрати камери, що задовольняють запитам користувача, які не завжди виражаються множиною значень параметрів. Вони можуть виглядати наступним чином: камера ціною не більше 200, не менше 8 мегапікселів, з підтримкою відео. Досить часто користувача цікавлять деякі функціональні особливості, тоді його запиту приймають такий вигляд: камера для спортивної зйомки або камера для друку великих

фотографій. Виходячи з цього, виділяються два типи властивостей: властивості товару і властивості, призначені для користувача. У нашому випадку властивостями товару є: ціна (0 ... 1000), мегапікселі (3.0 ... 12.0), оптичний зум (4x ... 12x), дисплей (2.5 ... 3.0), відео (так, ні), звук (так, ні), водостійкість (так, ні). Призначені для користувача властивості: максимальна ціна (0 ... 1000), використання (цифрове, друк великих фото, друк маленьких фото), фотографії (спортивні, пейзажі, портрети, макрозйомка). У своєму запиті користувач може вказати максимальну ціну, якого розміру він буде друкувати фотографії, що він буде фотографувати. Крім знань про товари, в базі знань присутні обмеження.

Вони можуть являти собою дозволені значення властивостей користувача, наприклад: вимоги = друк великих фотографій, максимальна ціна $\geq 200 \Rightarrow$ система не дозволить користувачеві вказати мінімальну ціну менше 200, якщо він хоче друкувати великі фотографії.

Інший вид обмежень дозволяє перетворити властивості користувача в властивості об'єктів: вимоги = друк великих фото, мегапікселі $\geq 5x \Rightarrow$ якщо користувач хоче великі фотографії, то не слід йому рекомендувати камери, менше ніж з п'ятьма мегапікселями. Власне, завдання системи – вибрати об'єкти, що задовольняють запитам користувача, за допомогою знань про об'єкти і обмеження.

Розглянемо першу групу методів, що базуються на знаннях – використання жорстких обмежень. Ідея методу проста: знайти такі об'єкти в базі даних, які повністю задовольняють вимогам користувача. Однак, пошук об'єктів не є основною проблемою. Головне завдання – отримання вимог користувача. Потрібно визначити як з ним взаємодіяти, які питання і в якій послідовності ставити, а які питання можна не ставити. Таким чином, основне завдання – отримати систему, зручну для користувача. Для того, щоб обмежити кількість поставлених користувачеві питань, використовуються значення типових параметрів. Вони діляться на три групи:

- статичні замовчування. Приклад: вимоги = друк великих фото. Більшість користувачів вимагають можливості друку великих фотографій, тому можна встановити дане значення параметра вимог за замовчуванням;

- залежні замовчування. Приклад: вимоги = друк маленьких фото, максимальна ціна = 300. Значить, якщо користувач вказав, що будуть друкуватися маленькі фотографії, то його можна не питати про максимальну ціну і вказати її рівною 300;

- похідні умовчання. Якщо перші два типи замовчувань визначаються заздалегідь, то похідні умовчання обчислюються з використанням логів взаємодії системи з користувачами. Користувачі вказують не всі необхідні параметри, а тільки ті, які вважають найважливішими. Логи взаємодії попередніх користувачів з системою використовуються також для прогнозування наступного параметра, який буде цікавий користувачеві;

Таблиця 1.4 містить інформацію про те, в якій послідовності користувачі встановлювали атрибути.

Таблиця 1.4 – Послідовність відповідей користувачів на питання системи

Номер	1 атрибут	2 атрибут	3 атрибут	4 атрибут	...
1	ціна	оптичний зум	мегапікселі	відео	...
2	ціна	оптичний зум	мегапікселі	відео	...
3	ціна	мегапікселі	оптичний зум	розмір дисплея	...
4	мегапікселі	ціна	оптичний зум	розмір дисплея	...
5	мегапікселі	ціна	розмір дисплея	оптичний зум	...

Параметри передбачаються з урахуванням їх популярності. Перший параметр що передбачається – найпопулярніший перший параметр серед попередніх користувачів. Якщо користувач вказав в якості першого параметра ціну, а в якості другого – оптичний зум, то шукається найпопулярніший третій атрибут в рядках з номерами 1 і 2. Якщо користувач вказав занадто жорсткі вимоги, то може виявитися, що таких об'єктів немає в системі. В цьому випадку система послідовно і автоматично послаблює обмеження користувача на об'єкти.

Розглянемо другу групу методів, заснованих на знаннях – вибір близьких об'єктів. На відміну від методів, які використовують жорсткі обмеження, методи даного класу рекомендують об'єкти, які мають найтісніший контакт з вимогами користувачів. В цьому випадку можлива рекомендація об'єктів, які не цілком задовольняють вимоги. Міра близькості між вимогами і властивостями обчислюється за формулою:

$$\text{similarity}(p, REQ) = \frac{\sum_{r \in REQ} w_r \cdot \text{sim}(p, r)}{\sum_{r \in REQ} w_r}, \quad (1.7)$$

де p – об'єкт;

REQ – множина вимог користувача;

r – вимога;

w_r – вага даної вимоги;

$\text{sim}(p, r)$ – міра близькості об'єкта p до вимоги r .

Чисельні властивості для користувача поділяються на три групи:

- «Більше – краще»: чим більше мегапікселів у камері, тим вона вважається краще;
- «Менше – краще»: чим ціна нижче, тим краще для користувача;
- «Ближче – краще»: користувачеві потрібен монітор з певною діагоналлю, більше або менше заданого параметра – гірше.

1.2.4 Гібридні методи

Власне, були розглянуті три основні види систем рекомендацій. Кожен з них має свої плюси і мінуси. Узагальнимо висновки в таблиці 1.5.

Таблиця 1.5 – Переваги та недоліки методів рекомендацій

Метод	Переваги	Недоліки
колаборативної фільтрації	досить мати тільки дані про рейтинги	проблема "холодного старту" при додаванні нових користувачів або об'єктів потрібно багато даних про різні користувачів
аналізу змісту	досить мати профіль поточного користувача і вміст об'єктів	проблема «холодного старту» для нових користувачів працюють переважно тільки з текстовими даними
заснований на знаннях	не обов'язково мати багато даних про покупки	необхідність використання бази знань

Великі системи рекомендацій, які використовують різні дані про об'єкти, користувачів і зв'язки між ними, використовують гібридні методи. Вони є об'єднанням різних підходів і алгоритмів: колаборативної фільтрації, аналізу змісту, аналізу соціальних зв'язків і зв'язків між об'єктами та інші. Виділяється три типи гібридних систем: з монолітною

організацією, з паралельною організацією і з конвеєрною організацією. Схеми кожного типу представлені на рисунках 1.4 – 1.6 [1].



Рисунок 1.4 – Монолитна організація гібридної СР

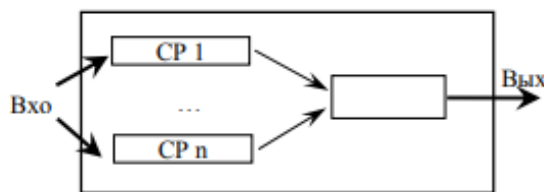


Рисунок 1.5 – Паралельна організація гібридної СР

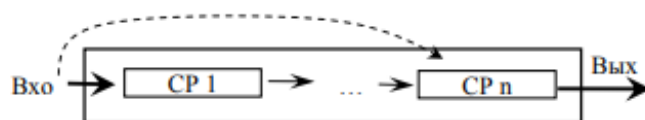


Рисунок 1.6 – Конвеєрна організація гібридної СР

На відміну від паралельної і конвеєрної організацій гібридних систем, системи з монолітною організацією мають один модуль, що включає в себе різні методи обробки неявних даних. Наприклад: якщо алгоритм колаборативної фільтрації рекомендує користувачеві кілька фільмів, то серед них можна вибрати найбільш відповідний, за допомогою порівняння жанрів цих фільмів з популярними жанрами користувача. Тобто якщо користувач віддає перевагу фантастиці, то серед фільмів двох

жанрів – історія і фантастика – система запропонує другий варіант.

Наступна схема організації називається паралельною і представлена на рисунку 1.5. Вихідні дані надходять на кожен модуль системи. Після того, як всі підсистеми видадуть відповідь, починає працювати агрегуючий модуль. Є три основні підходи для вибору множини найбільш релевантних рекомендацій. Перший підхід – змішати всі рекомендації і видати їх користувачеві:

$$rec(u) = \bigcup_{k=1}^n rec_k(u), \quad (1.8)$$

де $rec(u)$ – результат роботи «агрегатора» для користувача u ;

$rec_k(u)$ – результат роботи k -го модуля системи для користувача u .

Другий підхід – для кожного об'єкта підрахувати суму рейтингів, отриманих паралельними модулями і вивести об'єкти з найбільш високим рейтингом:

$$r(u, i) = \sum_{k=1}^n \beta_k \cdot r_k(u, i), \quad (1.9)$$

де $r(u, i)$ – потрібна передбачена оцінка об'єкта і користувачем u ;

$r_k(u, i)$ – оцінка, отримана k -м модулем;

β_k – вагові коефіцієнти.

Третій підхід – вибрати одну відповідну систему рекомендацій, реалізовану в одному з модулів для кожної конкретної ситуації. Наприклад: якщо модуль колаборативної фільтрації не дає відповідь через брак даних, то система вибирає метод аналізу змісту. Схема гібридної системи з конвеєрною організацією представлена на рисунку 1.6. Завдання рекомендацій ділиться на етапи, які виконуються послідовно. Результат роботи чергового етапу використовується в наступних.

2 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ

2.1 UX-дизайн: головні положення, правила та етапи створення

Ідея повної заміни дизайнера алгоритмом звучить сучасно, але вона є слабоперспективною. Дизайнер допомагає команді перетворити сиру ідею в цілісний інтерфейс з продуманою логікою роботи, інформаційною архітектурою та візуальним стилем, які вирішують бізнес-завдання і посилюють бренд. Під час роботи він приймає величезну кількість рішень, причому більшість з них неможливо описати чіткими процедурами. Крім того, вимоги замовника бувають хибними та суперечливими, і тут не обійтися без прийняття рішення людиною. Весь цей процес більше ніж підбір правильного шаблону і розфарбування його відповідно сучасній стилістиці. Машина не наділена емпатією, що є основою роботи дизайнера, але якщо говорити про творче партнерство, коли дизайнер в парі з алгоритмами вирішує завдання – тут багато перспектив і хороших прикладів.

Власне, існує два поняття: User Experience і User Interface. Розберемо їх визначення та виділимо основні відмінності. У класичному розумінні, User Experience означає досвід взаємодії та включає в себе різні UX-компоненти: інформаційну архітектуру, проектування взаємодії, графічний дизайн і контент (рисунок 2.1) [2].

Він має на увазі комплексний підхід до взаємодії користувача з інтерфейсом. Мета дизайну взаємодії – викликати вдячність у клієнта за доступність і зручність при взаємодії з продуктом. Це означає, що UX-дизайнери найчастіше орієнтуються на почуття, які викликає продукт, і працюють над покращенням досвіду, який отримують користувачі при роботі з ним. Головне завдання UX-дизайнера – зробити продукт логічним, зрозумілим та інтуїтивним [16].

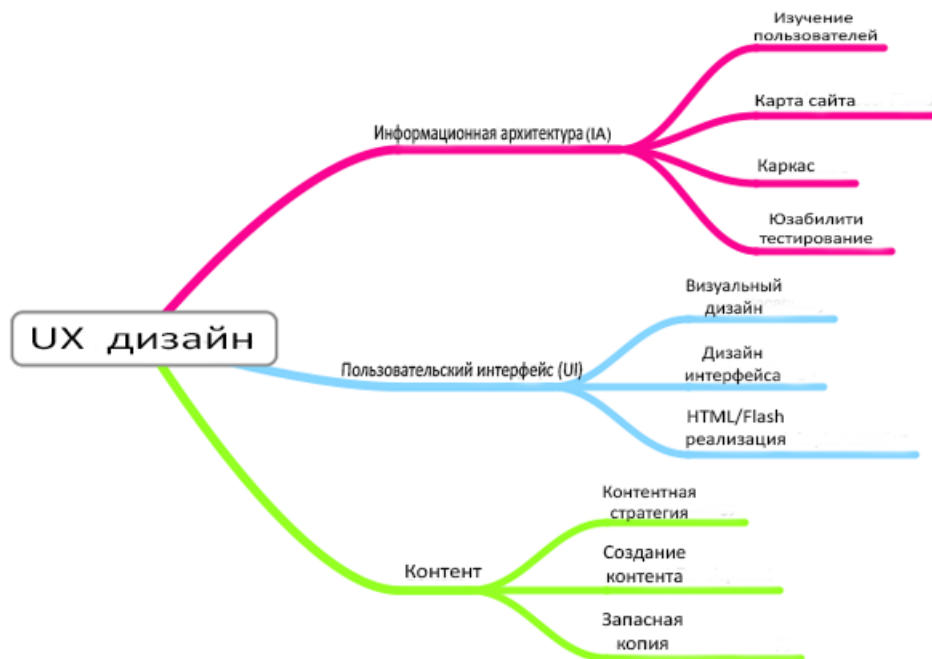


Рисунок 2.1 – UX-компоненты

Основні питання, які вирішуються UX дизайном:

- постановка цілей і завдань;
- підбір відповідних UX-інструментів для реалізації цілей;
- розробка продукту, максимально зручного і легкого в сприйнятті цільовою аудиторією;
- аналіз кінцевого результату – чи відповідає продукт очікуванням замовника і наскільки високий рівень сатисфакції користувачів.

User Interface – це більш вузьке поняття, яке включає в себе певний набір графічно оформлених технічних елементів. Його завдання – допомогти користувачеві організувати взаємодію з сайтом. На сьогоднішній момент існують деякі правила UI-дизайну:

- організованість елементів інтерфейсу. Це означає, що вони повинні бути логічно структуровані і взаємопов'язані;
- угруповання елементів інтерфейсу. Має на увазі об'єднання в

групи логічно пов'язаних елементів;

- вирівнювання елементів інтерфейсу. Погано вирівняний інтерфейс не може бути зручний;
- єдиний стиль елементів інтерфейсу. Стильове оформлення грає не останню роль, адже саме воно залишається в пам'яті користувача;
- наявність вільного простору. Це дозволяє розмежовувати інформаційні блоки, зосереджуючи увагу на чомусь одному [17].

Розроблений за всіма правилами призначений для користувача інтерфейс значно підвищує ефективність ресурсу і дає йому конкурентні переваги. Для ілюстрації вищесказаного проведемо аналогію: коли ми дивимося на айсберг, то бачимо лише його вершину. У цифровому світі вершиною айсберга є UI, а вся інша його частина – UX (рисунок 2.2) [3].

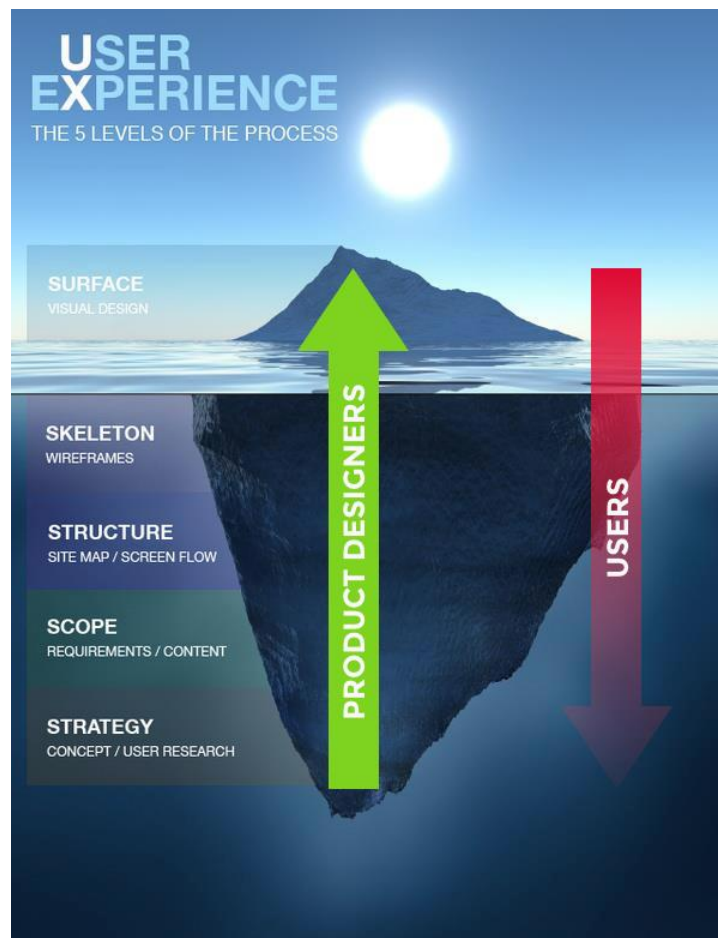


Рисунок 2.2 – Айсберг UX/UI

Саме тому правильний UX непомітний. Він заснований на досвіді взаємодії користувачів, на ментальній моделі і на конкретних законах. Розглянемо деякі з них.

Закон Фіттса: час досягнення мети прямо пропорційний дистанції до мети і обернено пропорційний розміру цілі. Математично цей закон можна описати таким чином:

$$T = \log_2 \left(\frac{D}{W} + 1 \right), \quad (2.1)$$

де T – час досягнення мети;

D – дистанція до мети;

W – розмір мети.

Головним чином це означає, що витрачений час на досягнення мети є функцією відстані і розміру цілі. На перший погляд це здається очевидним: чим далі ми від мети і чим менше вона за розміром, тим більше часу буде потрібно для позиціонування. Однак, точна математична характеристика вражає, так як включає в себе логарифмічну функцію, а це означає, що взаємозв'язок між розміром і часом реакції така, що невелике збільшення розмірів малих об'єктів дозволяє їх легше виділяти, тоді як невелика зміна розмірів великих об'єктів вже не має значення. Те ж саме стосується і дистанції до мети [18].

Закон Хіка: чим більше перед нами об'єктів, тим більше часу знадобиться на вибір одного з них. Математично цей закон можна описати таким чином:

$$T = a + b \cdot \log_2(n + 1), \quad (2.2)$$

де T – загальний час реакції;

a і b – константи, які описують індивідуальні особливості сприйняття, такі як затримка перед виконанням завдання і індивідуальний коефіцієнт швидкості прийняття рішення;

n – кількість рівнозначних альтернативних варіантів, з яких потрібно вибрати.

При проектуванні інтерфейсів, закон Хіка допомагає визначити оптимальну кількість об'єктів в однорідному масиві – наприклад, у меню. Зазвичай він застосовується в зв'язці з законом Фітса. Закон Хіка також тісно пов'язаний з іншими принципами сприйняття і психологічними особливостями прийняття рішень. Його можна однаково ефективно розглядати в контексті теорії близькості та правила 7 ± 2 , а також інших моделей поведінки користувача на сайті [19].

Теорія близькості: розташовані поруч об'єкти сприймаються користувачем як пов'язані між собою. Так як нас оточує величезна кількість різних об'єктів, мозок навчився захищатися від перевантаження шляхом автоматичного угруповання тих з них, які знаходяться поруч один з одним. Так простіше аналізувати і обробляти інформацію та працює на рівні інстинктів.

Закон Міллера: людина може сприймати число об'єктів 7 ± 2 . В іншому випадку, мозок відмовиться сприймати всю інформацію.

Закон Якоба: користувачі проводять більшу кількість часу на інших сайтах. Відповідно, вони вважають за краще, щоб ваш сайт працював так само, як і сайти, які вони вже знають.

Закон прегнантності: люди будуть сприймати і інтерпретувати неоднозначні або складні зображення в найпростішій формі, тому що ця інтерпретація вимагає від нас найменше когнітивних зусиль.

Закон Паркінсона: будь-яке завдання займає весь відведений на нього час. Згідно з цим законом, обмеження в часі, просторі або бюджеті призводять до збільшення продуктивності та ефективності.

Ефект серійного розташування: користувачі найкраще запам'ятовують перший і останній об'єкт в серії [20].

Ефект Фон Ресторфф: при наявності великої кількості схожих об'єктів найімовірніше запам'ятається той, що відрізняється від інших.

Зазвичай для досягнення мети фахівці з UX проводять спостереження за поведінкою користувача. Такі методи носять поетапний характер, а результати кожного комплексу аналізуються, щоб сформувати і поліпшити взаємодію. Розглянемо етапи створення UX-дизайну:

- формулювання стратегії UX. На цьому етапі UX-дизайнер аналізує головну мету, статус и створює опис проекту;

- аналіз. Якщо кінцевий продукт вже існує, UX-дизайнер проводить комплексну оцінку його поточного стану. На основі аналізу він робить звіт про виявлені проблеми і рекомендує необхідні правки. Потім проводяться дослідження конкурентів. Розглянемо особливості і інструменти такого аналізу:

- а) створення образу типового клієнта

Персона – це вигаданий персонаж, який представляє певну аудиторію користувачів, для яких призначений продукт. Створення персон для проекту включає інтерпретацію різних аналітичних даних, проведення консультацій, тестування поведінки користувачів та інших опитувань серед невеликих груп типових користувачів. Головною метою створення персон є визначення способу перспективного користувача і формування шаблонів на його основі.

- б) розкадрування і сценарії

Сценарій – це хронологічний порядок типового дня певної персоні, що описує, як додаток або сайт використовується в їх повсякденному житті. Розкадрування – це спосіб опису коли, чому і як люди використовують продукт. Це може бути чорнова замальовка, щоб пояснити ідеї і надати їм форму, або більш пророблений комікс, щоб забезпечити ефективне залучення аудиторії [21];

- каркасне моделювання. Багато UX-дизайнерів починають новий проект з моделювання каркаса сайту, так як це простий базовий етап. Каркасна модель – це спрощений зразок інтерфейсу сайту або програми, інструмент, який часто пов'язаний з роботою UX-дизайнера;

- прототипування. Після розробки статичних каркасних моделей, наступним кроком буде створення клікабельного прототипу на їх основі. Такі прототипи тестують за допомогою внутрішньої команди, клієнта або фокус-групи;

- тестування прототипів на рівні каркаса. На цьому етапі можуть проводитися різні види тестів, наприклад, докладне внутрішнє або зовнішнє тестування з використанням різних інструментів і технік. В результаті створюється список необхідних доопрацювань і змін. Етапи 3-5 можуть повторюватися кілька разів до досягнення необхідного результату;

- UI-дизайн. Незважаючи на те, що ця частина роботи виконується UI-дизайнерами, вона повинна бути включена в загальний процес створення інтерфейсу продукту. На цьому етапі дизайнери створюють стилі, а також оцінюють естетичність і функціональність кінцевого продукту;

- тестування прототипів. На цьому етапі тестують прототип майже готового продукту. Етапи 6 та 7 можуть циклічно повторюватися. Іноді етапи 4 та 5 виключаються з циклу розробки, а тестування проводиться на етапі «Тестування прототипів». Таке тестування може включати наступні підетапи:

а) А/В тестування

Спеціаліст по UX може провести дослідження і протестувати ефективність та якість UX-графічних елементів інтерфейсу. Вони перевіряють різні гіпотези, наприклад, привабливіє блакитна кнопка більше уваги, ніж жовта. Також вони створюють і досліджують різні версії дизайну і визначають, яка з них забезпечує кращу взаємодію з користувачем.

б) тестування через користувача

Тестування через користувача це процес демонстрації продукту користувачам, при якому необхідно виконати дії, визначені заздалегідь, і висловити свою думку з приводу тієї чи іншої функції. Кількість ітерацій тестування і чисельність користувачів, які беруть у ньому участь, залежить від бюджету проекту і тимчасових обмежень [22];

- етап передачі проекту. На цьому етапі всі документи проекту передаються розробникам;

- візуальний контроль якості. Головна мета візуального контролю якості – вдосконалення продукту. Важливо перевіряти можливі недоліки товару після того, як команда розробників реалізувала UI. На цьому етапі оцінюють зовнішній вигляд продукту і перевіряють можливі дефекти, неточності елементів інтерфейсу або недоліки локалізації;

- запуск продукту. Робота UX-дизайнера не припиняється після запуску продукту. Він аналізує відгуки реальних користувачів, щоб визначити проблеми, які не були виявлені раніше. У деяких випадках можливий перезапуск проекту з другого етапу. Процес тестування, дизайну, розробки та аналізу відгуків користувачів може повторюватися кілька разів поспіль.

Власне, перейдемо до проблем, з якими стикається фахівець. Головна проблема це, звичайно ж, величезна кількість інформації і альтернатив для вирішення тієї чи іншої графічної задачі. Тренди постійно змінюються. Важливо структурувати всі ці дані і вибрати саме те, що зараз актуально і що буде показувати результати належного рівня в практичному застосуванні. Також, система штучного інтелекту може значно прискорити процес виконання проекту, підказати нові ідеї для реалізації і, в цілому, позбавити UX-дизайнера від деяких обов'язків.

Сучасні технології дозволяють проектувати системи штучного інтелекту для дизайну. За допомогою машинного навчання, штучний інтелект «розуміє» правила праці у сфері творчості і застосовує їх. Однак,

щоб отримати максимальну вигоду з цього, потрібно чітко розуміти сильні сторони і обмеження штучного інтелекту. Автоматизація в дизайні може розвантажити дизайнерів і дати їм можливість зосередитися на завданнях, які здатна вирішити тільки людина, тобто технологія дає можливість делегувати. Більшу частину UX-дизайну складають правила і методи, тобто речі, які можна запрограмувати. Наприклад, корзина на сайті завжди знаходиться в правому верхньому куті. Такі технології – безсумнівний прорив, адже якщо звільнити фахівців від виконання більш простих завдань, залишивши їх штучному інтелекту, вони зможуть сконцентруватися на творчих завданнях і стануть більш продуктивними, а якість роботи збільшити в рази [23].

2.2 Класифікація сайтів за типом, призначенням та структурою

На сьогоднішній день існує велика кількість веб-сайтів. Вони класифікуються залежно від своїх функціональних можливостей, виконуваних завдань, призначення та інших характеристик. Звичайно, будь-яка класифікація в деякій мірі умовна, так як сайти різних видів часто можуть мати деяку схожість. Більш того, на практиці можна зустріти комбіновані веб-сайти. Однак, всі вони мають ключові відмінності і певні правила створення. Що логічно використовувати для одного типу сайту - може абсолютно не підійти для іншого, і це потрібно розрізняти. Також, можна виділити як обов'язкові блоки, так і другорядні. Перед тим як перейти до постановки завдань дослідження, розглянемо основні типи веб-сайтів і їх структуру.

Сайт візитка. З технічної точки зору сайт-візитка нічим не відрізняється від великих корпоративних сайтів або інтернет-магазинів. Він повинен бути SEO-оптимізований, мати правильну структуру, зручну і зрозумілу навігацію, стильний дизайн, що запам'ятовується з його адаптивної версією для охоплення мобільних користувачів, а також мати

високу швидкість завантаження. Однак з точки зору наповнення і функціоналу сайту, він все ж відрізняється від інших видів. В першу чергу, своєю лаконічністю – сайт-візитка, як правило, не передбачає велику кількість статей і текстів, а також має значно менший обсяг функціоналу. Головне правило: мінімум функціоналу і максимум ефективності. Отже, на даному типі сайту обов'язково повинні бути: інформація про компанію, інформація про надані товари і послуги, контакти. Другорядні блоки: привабливий фон, портфолію або галерея, інформація про команду, відгуки клієнтів, нагороди компанії, переваги перед конкурентами, заклик до дії, форма зворотного зв'язку. Елементарна структура сайту-візитки представлена на рисунку 2.3.

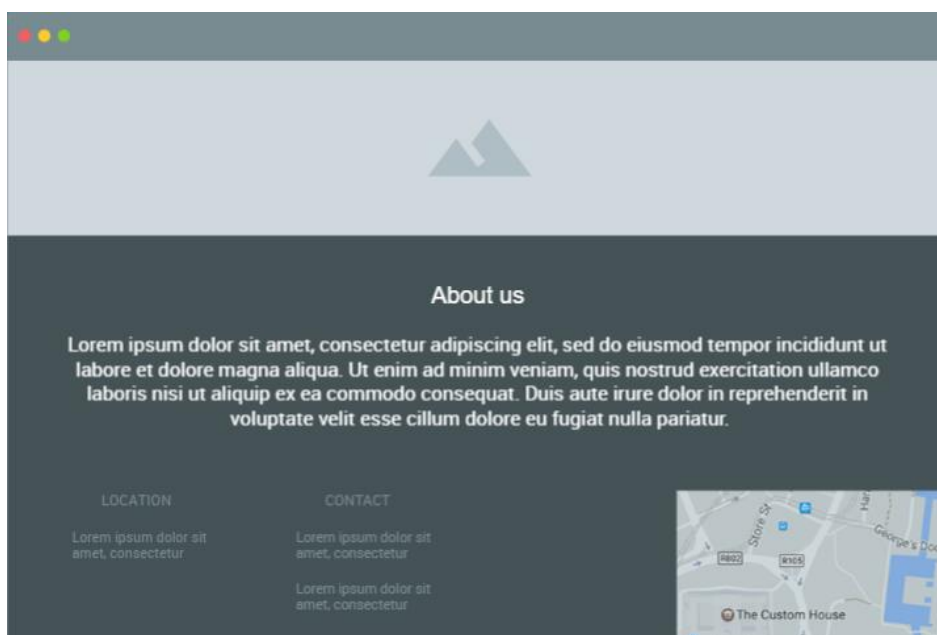


Рисунок 2.3 – Прототип сайту-візитки

Корпоративний сайт. Сайт даного типу це якісна і повноцінна презентація компанії в Інтернеті. Його основна мета – залучити нових клієнтів або партнерів, розвинути бізнес шляхом розкриття переваг компанії, докладної розповіді про її діяльність, послуги або товарів. Обов'язкові блоки: інформація про компанію, послуги, що надаються,

переваги перед конкурентами, портфолію, контакти. Другорядні блоки: відгуки клієнтів, інформація про команду, нагороди, заклик до дії, партнери, форма зворотного зв'язку. На рисунку 2.4 представлений приклад багатофункціонального і об'ємного корпоративного сайту елітного ріелтора [4]:



Рисунок 2.4 – Приклад корпоративного сайту

Сайт-портал. У класичному розумінні, сайт даного типу має багатокomпонентну розгалужену структуру, скомпоновану з функціонально самодостатніх сайтів. Концепція веб-порталів – надання максимальної кількості інтернет-сервісів в одному місці для залучення найбільшого числа користувачів. Вони можуть допомагати користувачам в пошуку потрібної інформації, представляючи інформацію з різних джерел в однаковому вигляді.

Існує два види порталів:

а) горизонтальний портал – портал, що охоплює безліч тематик, що представляє набір сервісів і орієнтований на максимально широку аудиторію – на максимальне охоплення її інтересів. Такі портали, як правило, поєднують в собі різноманітні функції, пропонують

різноплановий контент і різні сервіси;

б) вертикальний портал – портал вузької тематичної спрямованості, що надає різні сервіси для користувачів мережі за певними інтересами і орієнтований на повне охоплення певної тематики або області діяльності.

Кожен портал є унікальний. Якщо говорити про блоги або сайти з новинами, то можна знайти кілька таких ресурсів. Скрізь будуть різні новини, різні статті, різний дизайн, але загальний скелет один і той же – місце для публікації якихось матеріалів і, можливо, деякого спілкування між користувачами. Може бути і додатковий функціонал, але не великий. Портали же в цьому плані дуже сильно відрізняються, тому що кожен з них по-своєму унікальний. Мало того, що у такого проекту унікальний дизайн, у нього ще й унікальний функціонал. Яскравими прикладами є Яндекс.Метрика, Google Analytics, Google AdSense. Всі вони чимось схожі, але функціонал кожного з цих сервісів сильно відрізняється один від одного, на відміну від інших типів сайтів. Відповідно, не можна виділити основні і другорядні блоки.

Однак, порталами часто називають себе великі сайти з розгалуженою внутрішньою структурою і великою кількістю посилань, але якщо більша частина цих посилань – внутрішні, які ведуть на іншу сторінку цього ж сайту, називати такий сайт порталом не зовсім правильно. Однак дане твердження носить суб'єктивний характер. Наприклад, офіційний сайт Харківського національного університету радіоелектроніки потрапив в список переможців Національної інтернет-премії України «WEB AWARDS UA», ставши кращим в категорії «Портали та каталоги». Виходячи з класичного опису сайту-порталу, логічніше було б віднести даний ресурс до категорії корпоративних веб-сайтів, але це питання залишається суб'єктивним. На рисунку 2.5 представлений веб-інтерфейс даного сайту.

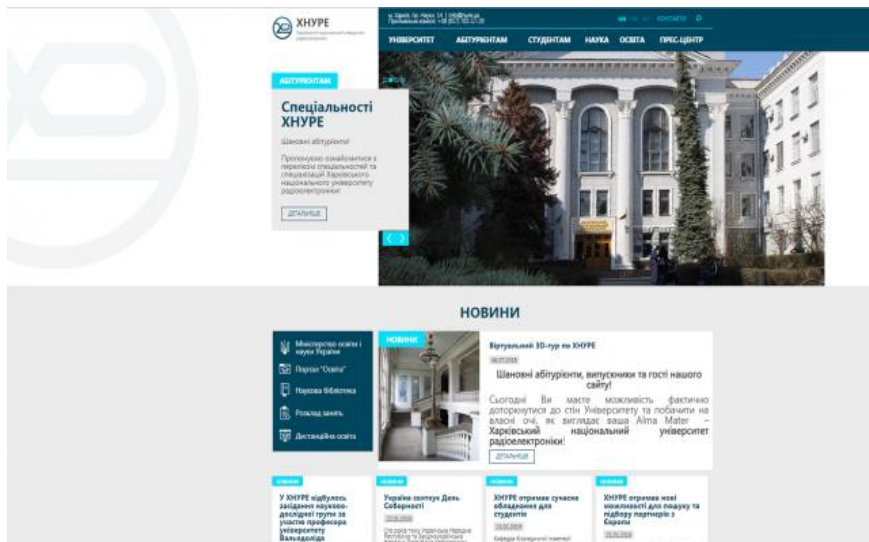


Рисунок 2.5 – Веб-інтерфейс сайту ХНУРЕ

Сайт-сервіс. Сайти даного типу призначені для вирішення прикладних утилітарних завдань. Яскравим прикладом є Яндекс.Маркет – сервіс пошуку товарів і порівняння їх за цінами (рисунок 2.6) [5].

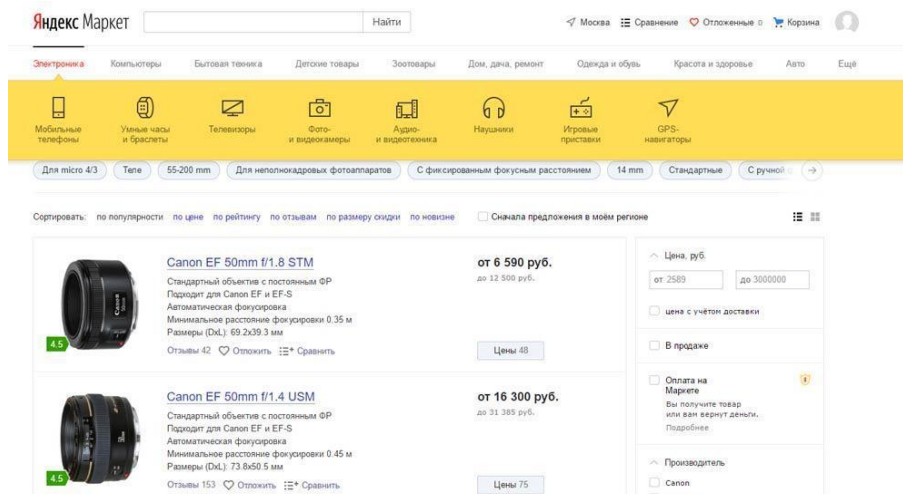


Рисунок 2.6 – Веб-інтерфейс порталу Яндекс.Метрика

Вони відрізняються лаконічністю і мінімалістичністю у всьому, включаючи основну функцію. Це чудовий і часто єдиний можливий варіант зробити корисний для клієнтів інструмент, виконаний в витриманій палітрі кольорів, і інтуїтивно зрозумілий кожному. На запропонованому сайті все підпорядковано одній меті: зробити пошук товарів зручним і зрозумілим (рисунок 2.6).

Інтернет магазин. Сайти даного типу можна розглядати як з технологічної позиції, так і з позиції маркетингових комунікацій. Інтернет-магазин – це інтерактивний веб-сайт, що рекламує товар або послугу, приймає замовлення на покупку, пропонує користувачеві вибір варіанту розрахунку, способу отримання замовлення і виписує рахунок на оплату. З технічної точки зору, інтернет-магазин можна визначити як багатофункціональний програмний модуль, вбудований в веб-сайт і покликаний забезпечити продаж продукції фірми через Інтернет. На сайті інтернет-магазину зазвичай представлений докладний каталог товарів з цінами, на основі якого користувач формує своє замовлення. Логіка інтернет-магазину вибудовується за принципами реального торгового залу.

У будь-якому інтернет-магазині неодмінною умовою є наявність структурованого каталогу товарів (аналог полиць в магазині), системи пошуку за ключовими словами (аналог секцій в торговому залі), кошики замовлень (аналог реальної кошика), різноманітності платіжних систем, вибір способів і служб доставки. Інтернет-магазини класифікують за видами продажів, способам оплати і за типом і кількістю товарних категорій. Інтернет-магазини в залежності від експонованих товарів в каталозі діляться на інтернет-супермаркети, інтернет-вітрини, автоматичні магазини, онлайн-аукціони. Розрізняють інтернет-магазини розроблені спеціально тільки для B2B або B2C сфери. Зазвичай це багатосторінковий сайт, який включає в себе головну сторінку, кошик, картку товару та інші. Всі ці сторінки представлені на рисунках 2.7 [6], 2.8 та 2.9.

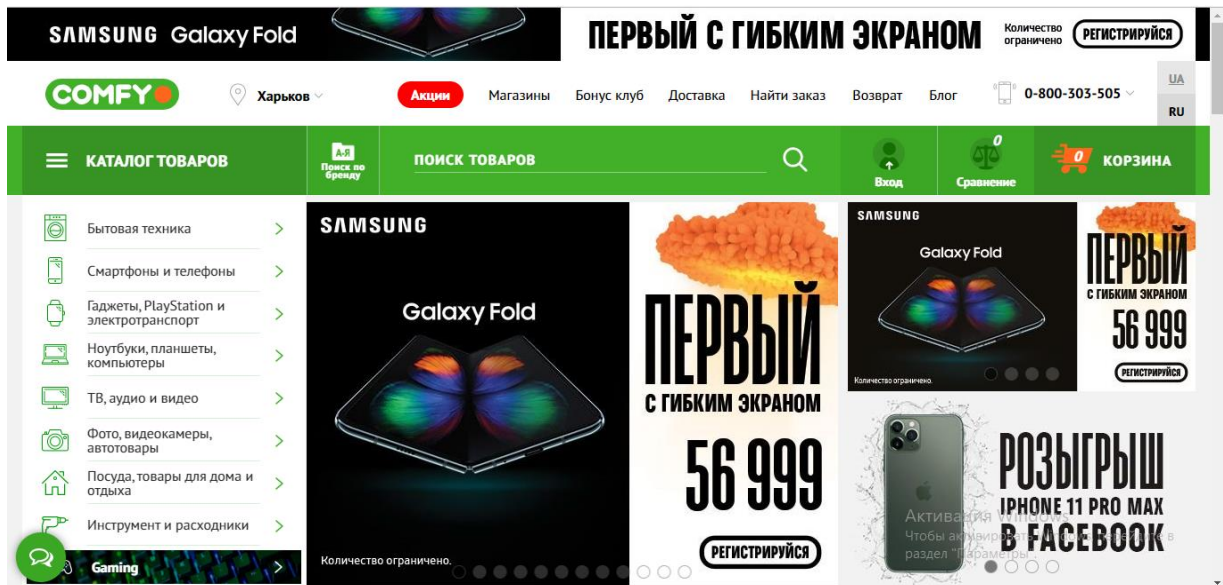


Рисунок 2.7 – Пример интернет-магазина

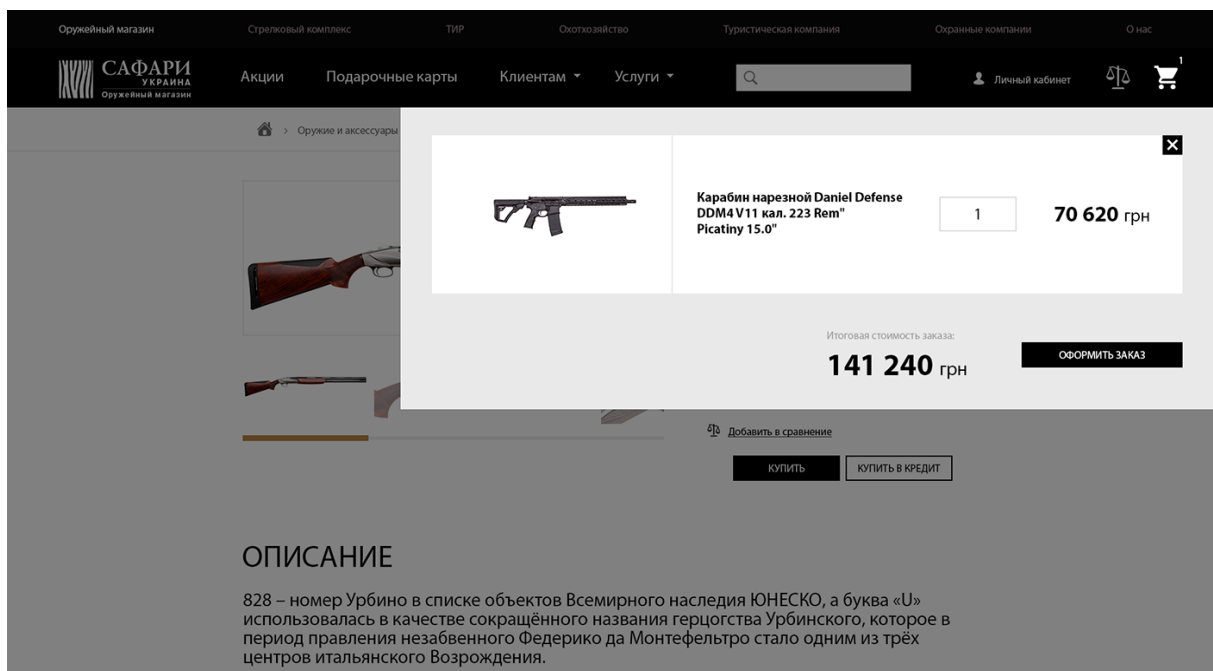


Рисунок 2.8 – Корзина в интернет-магазине


Соружный магазин Стрелковый комплекс ТРФ Охотхозяйства Туристическая компания Охотничьи компании Охот

САФАРИ
УКРАИНА
Охотничьи магазины

Акции Подарочные карты Клиентам Услуги

Вход Личный кабинет

Оружие и аксессуары > Стрелковое оружие > Оружие гладкоствольное > Двухствольное переломное > Ружья охотничьи > Benelli 828U кал.12 28" (345 мм)



РУЖЬЕ ОХОТНИЧЬЕ BENELLI 828U SILVER КАЛ. 12 28" (345 MM)

калибр	выстрелок	вес	длина ствола
12	76	2,98 кг	710 мм

Производитель: Benelli
Страна производитель: Италия
Модель: 828U Black
Материал ложа: дерево
Полностью приспособленное ложа с мужской Уложка: пластиковый кейс

★★★★★ 105 530 грн

Добавить в сравнение

купить купить в кредит

ОПИСАНИЕ


828 – номер Урбино в списке объектов Всемирного наследия ЮНЕСКО, а буква «U» использовалась в качестве сокращённого названия герцогства Урбинского, которое в период правления незабвенного Федерико да Монтефельтро стало одним из трёх центров итальянского Возрождения.

Таким образом, название «828U» объединяет в себе культурное наследие прошлого и технологические достижения наших дней.

Исполнение «Black» основано на строгой эстетике спортивных ружей, подчеркнутой стремительными очертаниями чёрной колодки, гармонично сочетающимися с линиями ложа.

Ружье оснащено системой дифференцированного поглощения импульса отдачи «Progressive Comfort», состоящей из установленного в полости приклада амортизатора, сменных затыльников и гребней и комплекта проставок (5 для регулировки погиба от 45 до 65 мм с шагом 5 мм, 4 для регулировки отвода на ± 6 , ± 3 , $\pm 6 \frac{1}{2}$, $\pm 3 \frac{1}{2}$), позволяющих выбрать сорок вариантов положения приклада.

ОБЗОР
РУЖЬЕ BENELLI 828U






Benelli Black:
Ружья, которое все хотят
Высочайшее качество, прогрессивная система поглощения отдачи, регулируемый затылковый упор Benelli. Позволяет вносить вклад от нашего опыта на создание уникального изделия...

Benelli Cup 2019:
Какие будут следующие призы?
Спортивный приз будет разыгран между двумя лучшими командами в турнире на Кубок Benelli, который традиционно проводится чемпионатом Украины по ВОО. Официальный представитель Benelli.

Интерактивное ружье Benelli Montefeltro 12 76
Montefeltro – легендарное имя семейства охотничьих ружей Benelli, выходящее в настоящее время.

Benelli Raffaello Via Diamant
Дизайнерский, уникальный и инновационный охотничий инструмент. Создан в Украине с душой и талантом, который формирует уникальные черты своего производства. Уникальный и легендарный дизайн Raffaello Via Diamant – это настоящее искусство.

СОПУТСТВУЮЩИЕ ТОВАРЫ

 <p>Кейс для хранения Benelli</p> <p>950 грн</p> <p>купить</p>	 <p>Сухая тефлоновая смазка Resoil 150 мл</p> <p>150 грн</p> <p>купить</p>	 <p>Набор для чистки ружья кал. 12</p> <p>199 грн</p> <p>купить</p>
--	--	--

УЗНАВАЙТЕ ПЕРВЫМИ О НОВИНКАХ И АКЦИЯХ

Введите e-mail ПОДПИСАТЬСЯ

Instagram Facebook YouTube

Контактная информация
(097) 500-64-44
(093) 501-64-44
(044) 501-64-44
shop@safari-ukraina.com

Режим работы:
понедельник-суббота с 10:00 до 18:00
воскресенье - выходной




Рисунок 2.9 – Карточка товару в интернет-магазині

Посадкова сторінка / landing page. Сайти даного типу збудовані по певній структурі під назвою AIDA. Їх метою є спонукати користувача вчинити будь-яку активну дію. Техніка AIDA визначає черговість вибудовування секцій (рисунок 2.10) [7].

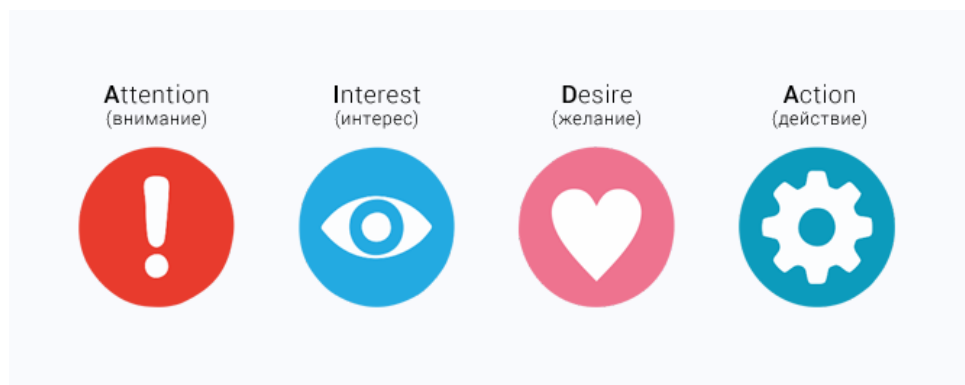


Рисунок 2.10 – Техніка AIDA

Мета першого екрану посадкової сторінки (рисунок 2.11) – захопити увагу відвідувача. Існує правило 5 секунд, яке свідчить, що приблизно стільки часу потрібно відвідувачеві сайту, щоб зрозуміти, чи хоче він залишитися на сторінці або піти. Саме тому обов'язковим вважається чіткий і зрозумілий заголовок, заклик і саме цільова дія.

Щоб написати заголовок, який буде продавати можна скористатися методикою 4U:

- Usefulness (корисність): яку користь ви надасте вашому клієнтові;
- Ultra-specificity (специфічність): наскільки це вигідніше;
- Urgency (терміновість): за який проміжок часу клієнт це отримає;
- Uniqueness (унікальність): унікальність вашої вигоди в порівнянні

з аналогами.

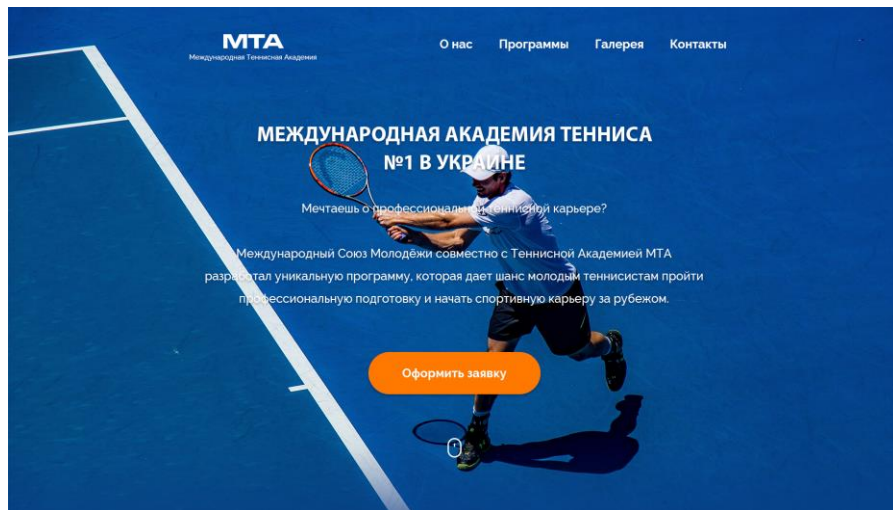


Рисунок 2.11 – Блок Attention з конкретною дією

Після того як людина зрозуміла, що вона знаходиться там, де треба, потрібно викликати інтерес (рисунок 2.12). Це можна зробити за допомогою докладної інформації про вашу пропозицію, яке включає в себе зображення, графіку, інфографіку та відео.

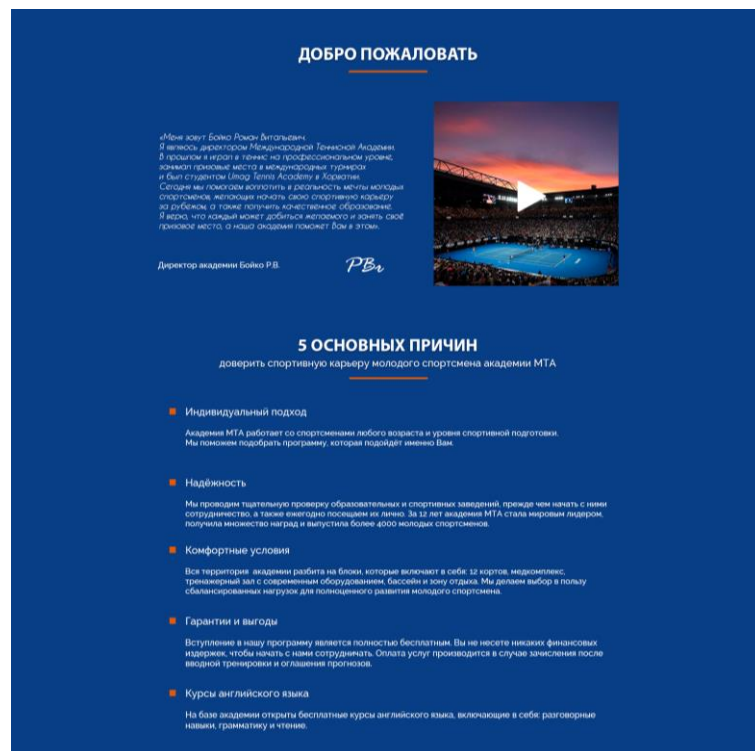


Рисунок 2.12 – Фрагмент блока Interest

В блоці D (рисунок 2.13) необхідно викликати бажання у користувача до того, щоб він зробив цільової дію. Це можуть бути акції, подарунки, обмеження пропозиції за допомогою лічильника або штучного дефіциту продукту. Також тут можна розташувати другорядний заклик до дії. Наприклад завантажити програму якоїсь події.



Рисунок 2.13 – Фрагмент блока Desire

В кінці посадкової сторінки завжди розміщено заклик до дії з простим процесом отримання результату. (рисунок 2.14).

The image shows a dark blue background with the title 'ЗАЯВКА' in white. Below the title is a short paragraph in white text: 'Узнайте, может ли Ваш ребенок тренироваться и учиться за границей за счет спортивной стипендии, которую выделяют на развитие молодежного спорта за границей'. Below the text are two white input fields with rounded corners. The first field is labeled 'Ваше имя*' and the second is labeled 'Номер телефона*'. Below the input fields is an orange button with the text 'Свяжитесь со мной'.

Рисунок 2.14 – Фрагмент блока Action

Однак, до даної техніки варто додати дуже важливий елемент, який впливає на конверсію сайтів, це С – confidence (довіра). Мета цього блоку заспокоїти відвідувача і завоювати його довіру шляхом розміщення

відгуків клієнтів, нагород і спонсорів. Фрагмент данного блока представлений на рисунку 2.15.

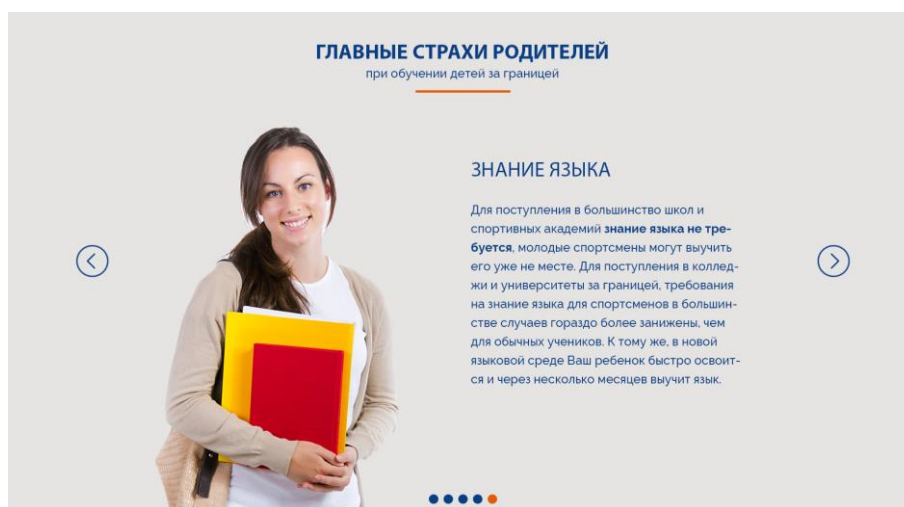


Рисунок 2.15 – Фрагмент блока Action

Форум. Даний тип сайтів передбачає платформу для спілкування між різними користувачами на одну або на декілька тем, що залежить від спеціалізації форуму. Суть його роботи полягає в створенні користувачами своїх тем з їх подальшим обговоренням, шляхом розміщення повідомлень всередині цих тем. Окремо взята тема, по суті, являє собою тематичну гостьову книгу. Користувачі можуть коментувати заявлену тему, задавати питання по ній і отримувати відповіді, а також самі відповідати на питання інших користувачів форуму і давати їм поради. Усередині теми також можуть влаштовуватися опитування. Питання і відповіді зберігаються в базі даних форуму, і в подальшому можуть бути корисні як учасникам форуму, так і будь-яким користувачам мережі Інтернет, які можуть зайти на форум, знаючи адресу сайту, або отримати його від пошукових систем за допомогою пошуку інформації.

Тематика форумів може бути найрізноманітнішою, охоплюючою всі сфери життя, і визначається або адміністрацією, або залежить від контингенту учасників. При цьому форуми можуть бути як спеціалізовані,

так і вузькоспеціалізовані. Теми однієї тематики об'єднуються, для зручності в відповідні розділи і, таким чином, найпоширеніша ієрархія веб-форуму: Розділи → Теми → пости. Існують форуми можуть як автономно, так і бути частиною веб-порталів. Приклад даного типу сайтів представлений на рисунку 2.16 [8].

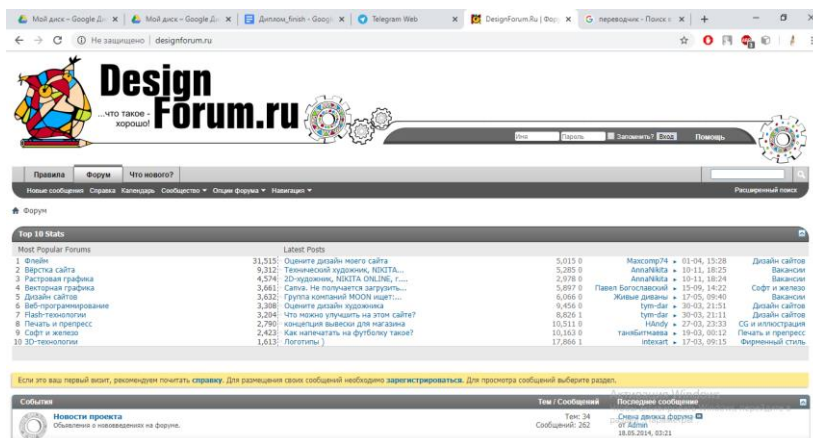


Рисунок 2.16 – Форум графічних і UX / UI дизайнерів

Блог. Даний тип сайтів передбачає, що основний вміст це записи, які регулярно додаються і які можуть містити текст, зображення або мультимедіа. Вони схожі з форумами, однак основна відмінність - те, що блог веде одна людина (рисунок 2.17) [9].

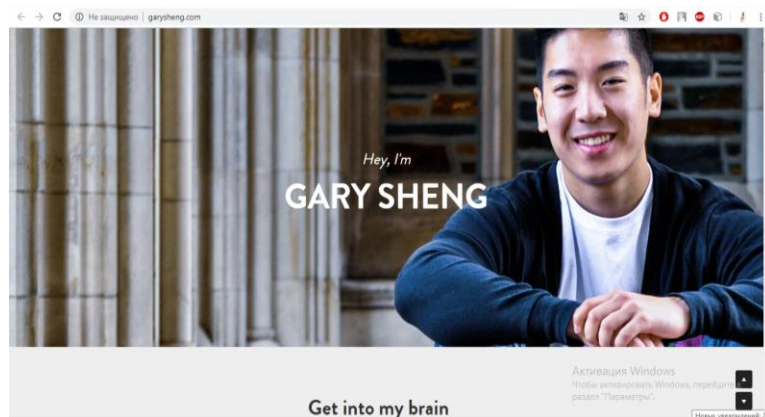


Рисунок 2.17 – Сайт-блог

Сайти з новинами включають в себе стрічку різнорідних новин бізнесу, політики, спорту та інші. Інакше кажучи, це аналог ЗМІ у всесвітній павутині. Основні формати розміщення інформації: банери, текстово-графічні елементи і PR статті. Сайти даного типу кожен день борються за місце в топі пошукової видачі, щоб залучити якомога більше відвідувачів. Для цього ринку характерний дуже високий рівень конкуренції, і щоб випередити суперників недостатньо просто наповнювати сайт свіжим і релевантним контентом. Ресурс також необхідно оптимізувати відповідно до рекомендацій пошукових систем. В першу чергу, це дозволить зробити сайт зручним для відвідувачів і, як наслідок, підняти його в результатах пошуку. На сайтах з новинами щодня виникають десятки нових сторінок, і якщо не налагодити систему навігації, користувачі будуть просто йти, не зумівши знайти потрібну інформацію в нескінченному потоці статей. Це спричинить неминуче погіршення поведінкових факторів, зростання показника відмов, а також зниження сайту в ранжируванні.

Щоб налаштувати навігацію, яка робить сайт зручним для користувача, потрібно подбати про такі аспекти UX:

- Структура сайту повинна бути простою і зрозумілою як для користувачів, так і для пошукових роботів пошукових систем. Вона будується на базі семантичного ядра: групи запитів зумовлюють категорії, підкатегорії і теги. Правильна структура сайту повинна відповідати наступним вимогам: чітка ієрархія розділів і підрозділів; можливість масштабування, щоб при додаванні нових сторінок структура сайту не змінювалася; рівень вкладеності сторінок не повинен бути більше трьох; URL сторінок повинні відповідати принципу ЧПУ.

- Теги та категорії групують пости і полегшують пошук потрібного контенту відвідувачам. Їх можна створювати в ієрархічному порядку, додаючи підкатегорії. При роботі варто врахувати такі нюанси:

- а) створювати нову категорію необхідно, тільки якщо ви

впевнені, що будете наповнювати її контентом;

б) не варто поміщати один матеріал в кілька категорій, щоб уникнути появи дублів;

в) при впровадженні підкатегорій необхідно налаштувати «хлібні крихти» для поліпшення навігації усередині сайту;

г) не потрібно робити багато категорій. Ідеальної кількості не існує, так як все залежить від тематики, але велика кількість категорій суттєво ускладнить навігацію для користувача.

«Хлібні крихти» – це навігаційний ланцюжок, що відображає шлях від головної до поточної сторінки. Найчастіше вона розташовується у верхній частині, під хедером і горизонтальним меню.

При впровадженні хлібних крихт варто дотримуватися наступних рекомендацій:

а) розміщувати в одному місці на всіх сторінках, щоб не збивати з пантелику відвідувачів;

б) оформляти таким чином, щоб вони візуально відрізнялися від основного тексту на сторінці, але і не були занадто дрібними;

в) сторінка не повинна посилатися на саму себе, тобто остання ланка в ланцюжку має бути на рівень вище поточної сторінки. Зокрема, хлібні крихти не повинні розташовуватися на головній сторінці - циклічні посилання шкодять оптимізації сайту;

г) прописувати шлях від початку до кінця, не загубивши жоден з елементів.

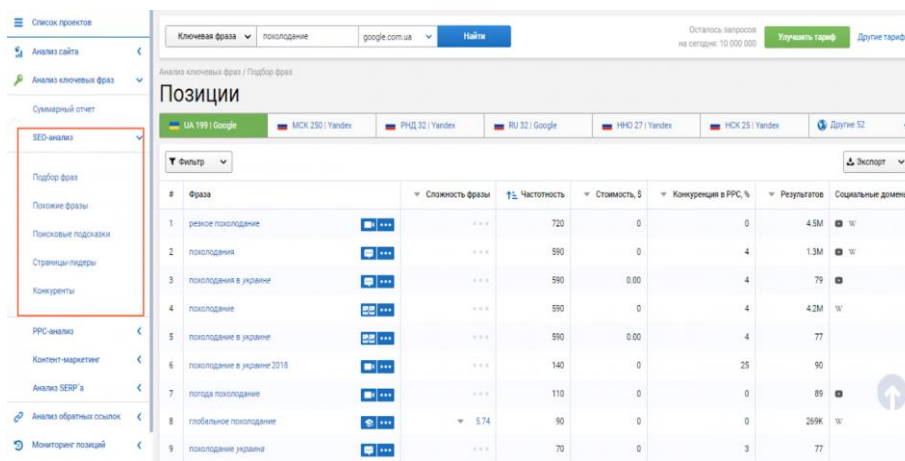
Панель пошуку необхідна великим сайтам: за її допомогою користувачі зможуть моментально знайти потрібний матеріал.

Карта сайту. Для ресурсу з новинами швидкість індексації та появи статей у видачі має вирішальне значення. Для прискорення цього процесу, створюють спеціальну карту сайту. Щоб забезпечити повне охоплення контенту пошуковими роботами Google, створюється XML Sitemap для новинного пошуку з динамічної генерацією посилань і постійним

оновленням. Карта буде служити навігатором для бота Googlebot-News, і з її допомогою він буде швидше індексувати нові сторінки. При створенні файлу Sitemap для сайтів з новинами потрібно враховувати наступні рекомендації:

- а) в файлі необхідно розміщувати сторінки, що були опубліковані за останні два дні;
- б) в один файл можна помістити не більше 1000 URL;
- в) карту необхідно оновлювати у міру публікації постів;
- г) після генерації карти сайту надішліть її на розгляд пошукової системи через панель вебмайстрів Google.

Пошук і впровадження ключових слів. Для новинних порталів потрібен швидкий пошук ключових слів, адже в цій сфері неможливо заздалегідь скласти перелік запитів для просування тієї чи іншої статті. Щоб швидко підібрати релевантні ключі існують різні сервіси. Досить ввести основний запит і вибрати потрібні ключові слова. Приклад такого сервісу представлений на рисунку 2.18 [10].



The screenshot shows a web interface for keyword analysis. At the top, there is a search bar with the text 'ключевая фраза' and 'поисковик' set to 'google.com.ua'. Below this is a table titled 'Позиции' (Positions) with columns for 'Фраза' (Phrase), 'Сложность фразы' (Phrase complexity), 'Частотность' (Frequency), 'Стоимость, \$' (Cost), 'Конкуренция в РРС, %' (Competition in PPC), 'Результатов' (Results), and 'Социальные домены' (Social domains). The table lists 9 phrases related to 'поисковик' (search engine) and 'погода' (weather).

#	Фраза	Сложность фразы	Частотность	Стоимость, \$	Конкуренция в РРС, %	Результатов	Социальные домены
1	реклама поисковик	...	720	0	0	4.5M	W
2	поисковик	...	590	0	4	1.3M	W
3	поисковик в украине	...	590	0.00	4	79	W
4	поисковик	...	590	0	4	4.2M	W
5	поисковик в украине	...	590	0.00	4	77	W
6	поисковик в украине 2018	...	140	0	25	90	W
7	погода поисковик	...	110	0	0	89	W
8	глобальное поисковик	...	90	0	0	269K	W
9	поисковик украина	...	70	0	3	77	W

Рисунок 2.18 – Сервіс для генерації релевантних ключів

Для успішного просування, ключові слова необхідно розміщувати органічно і без переспама. Вони повинні міститися не тільки в текстовому блоці, але і в інших структурних елементах.

Адаптація сайту під різні пристрої. Зараз адаптація під мобільні пристрої необхідна кожному сайту, але для новинних порталів це особливо важливо, адже користувачі частіше читають новини з телефонів. Найпоширеніші методи оптимізації під смартфони: впровадження адаптивного дизайну і застосування технології AMP.

2.3 Постановка задач дослідження

Метою даної атестаційної роботи є дослідження основних принципів побудови систем рекомендаційного типу. Виходячи з поставленої мети, необхідно вирішити такі завдання:

- розглянути основні визначення, що стосуються систем рекомендаційного типу;
- провести порівняльний аналіз існуючих підходів до розробки РС;
- розглянути методи вироблення рекомендацій і використовуються алгоритми;
- вивчити області застосування РС;
- провести аналіз предметної області, а саме поняття UX-дизайн і роботи фахівців даної сфери;
- виявити основні проблеми, з якими стикається користувач при створенні прототипів;
- розглянути наявні системи прототипування інтерфейсів;
- провести аналіз цільової аудиторії;
- розробити прототип системи, яка підтримує покрокове проектування структури призначеного для користувача інтерфейсу у вигляді веб-ресурсу;
- детально описати функціонал системи, що розробляється і які проблеми вона вирішує;
- провести аналіз методик тестування usability;
- визначитися з методикою тестування розробленого продукту.

3 РОЗРОБКА СИСТЕМИ РЕКОМЕНДІЙНОГО ТИПУ ДЛЯ ПРОТОТИПУВАННЯ ІНТЕРФЕЙСІВ

3.1 Існуючі системи прототипування веб-інтерфейсів

Проаналізуємо найпопулярніші існуючі системи прототипування, щоб оцінити їх переваги та недоліки, а в подальшому порівняти з власним продуктом. Для зручності зведемо інформацію в таблицю 3.1 (Додаток А), в якій порівняємо сервіси за такими параметрами:

- мова – мова інтерфейсу;
- безкоштовний тариф - наявність безкоштовного тарифу або пробної версії;
- найдешевший тариф – вартість найдешевшого тарифу або ліцензії;
- особливості – відмінні «фішки» інструменту;
- недоліки – які є мінуси і проблеми у використанні;
- інтеграції – можливі інтеграції з іншими сервісами;
- завантажувані додатки / веб-версія – деякі з сервісів вдають із себе завантажувані програми, деякі працюють прямо в браузері, а деякі доступні в двох варіантах. У таблиці 3.1 вказано наявність онлайн-версії / викачуваного додатки і підтримувані ОС для кожного інструменту;
- мобільний додаток – наявність мобільного застосування для попереднього перегляду прототипів;
- техпідтримка – можливі способи зв'язку з технічною підтримкою сервісу.

Розглянемо кожний інструмент для прототипування інтерфейсів.

Axure – найбільший і найвідоміший додаток для прототипування, дозволяє створювати високоякісні прототипи сайтів і мобільних додатків (рисунки 3.1) [11]. Володіє широким функціоналом з великим набором інструментів. Вимагає установки софту на ПК, онлайн-версія недоступна. Працює з Windows і macOS.

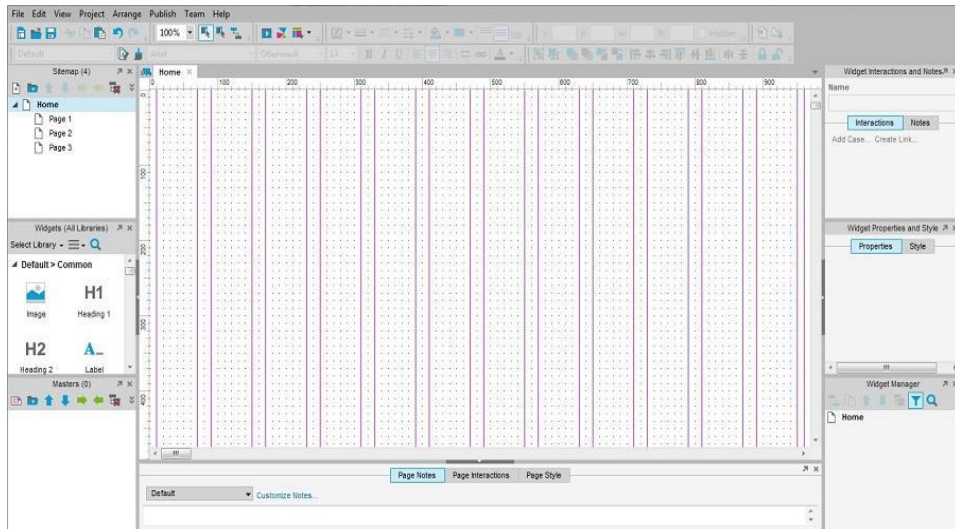


Рисунок 3.1 – Інтерфейс інструменту Axure

Origami Studio – повністю безкоштовний зручний додаток від Facebook (рисунок 3.2) [12]. Працює з macOS. Синхронізується зі Sketch. У додатку є спільнота на Facebook, де користувачі можуть ділитися своїми прототипами, обговорювати їх створення і допомагати один одному.

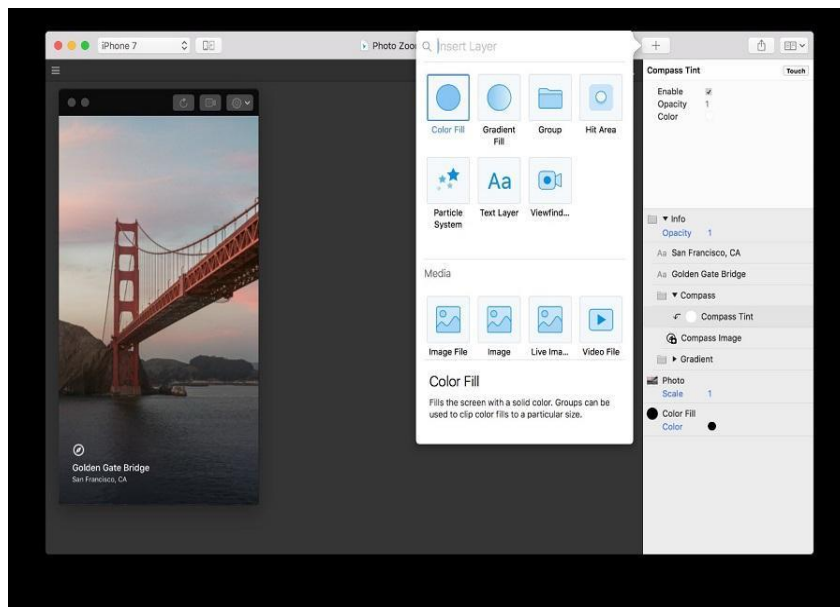


Рисунок 3.2 – Інтерфейс інструменту Origami Studio

У Proto.io зручний і досить простий інтерфейс (рисунок 3.3) [13]. Як і у більшості сервісів, у нього є можливість колективної роботи і спільного внесення правок. На будь-якому моменті роботи прототип можна протестувати як на ПК, так і на мобільних пристроях.

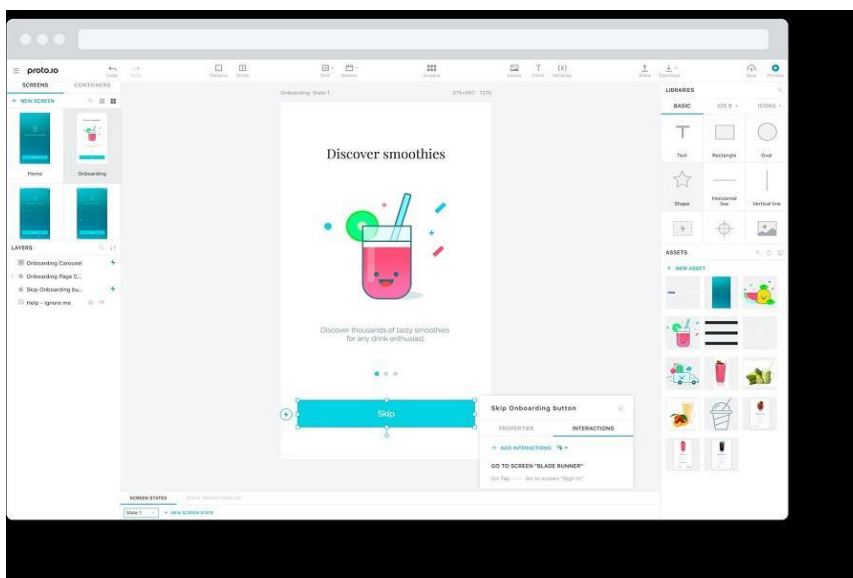


Рисунок 3.3 – Інтерфейс інструменту Proto.io

З Vestr можна працювати як онлайн, так і встановивши додаток на ПК (рисунок 3.4) [14].

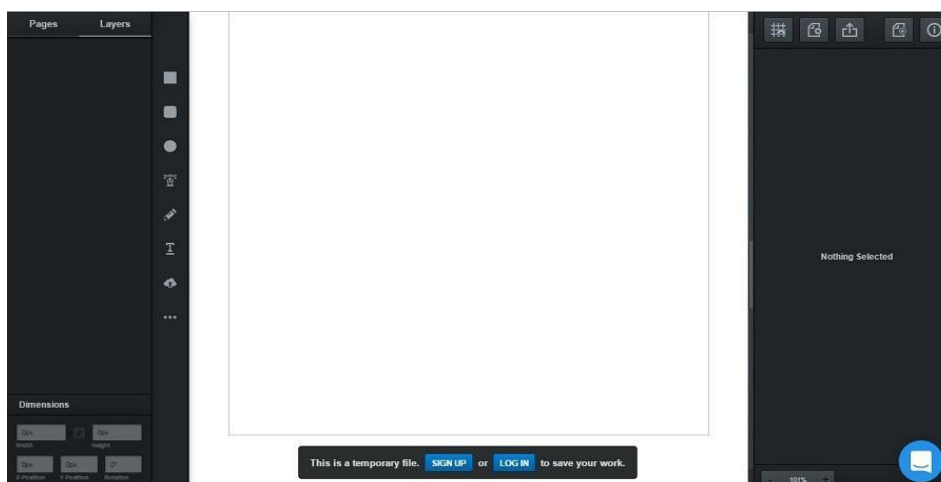


Рисунок 3.4 – Інтерфейс інструменту Vestr

При цьому обидві версії синхронізуються одна з одною, даючи можливість працювати над одним прототипом як онлайн, так і з комп'ютера. Кожному створюваному проекту присвоюється URL, який можна відправити колегам або клієнту для внесення коментарів і правок.

InVision – популярний інструмент для створення прототипів (рисунок 3.5) [15]. З особливостей можна виділити зручний інтерфейс, можливість організувати командну роботу.

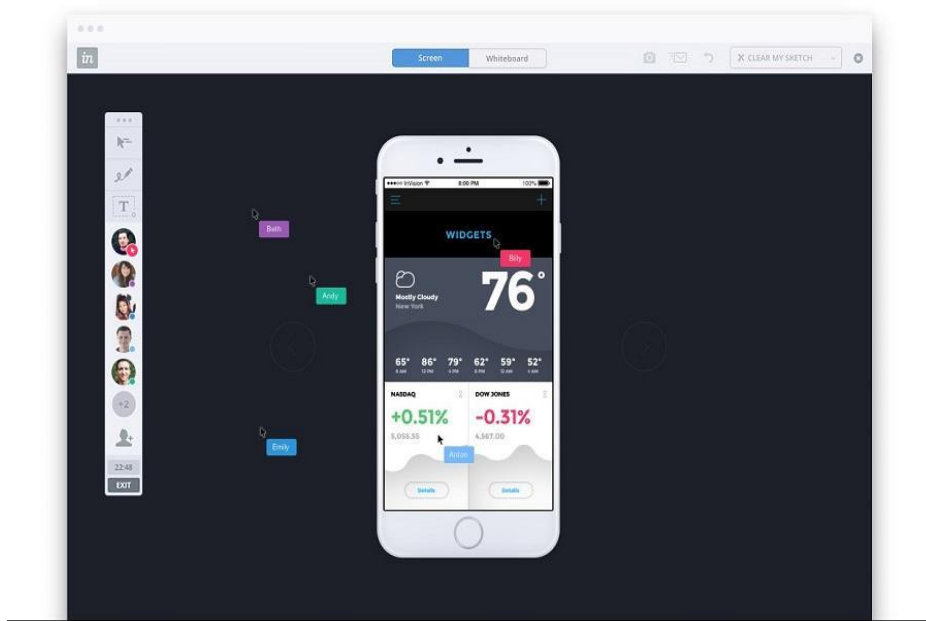


Рисунок 3.5 – Інтерфейс інструменту InVision

3.2 Побудова діаграми прецедентів для опису функціоналу системи

Діаграма прецедентів, також звана діаграмою варіантів використання, відображає відносини між акторами і прецедентами і дає уявлення про функціональну поведінку системи. На рисунку 3.6 представлена діаграма, що описує типи користувачів і їх взаємодію з стрічкою прототипів.

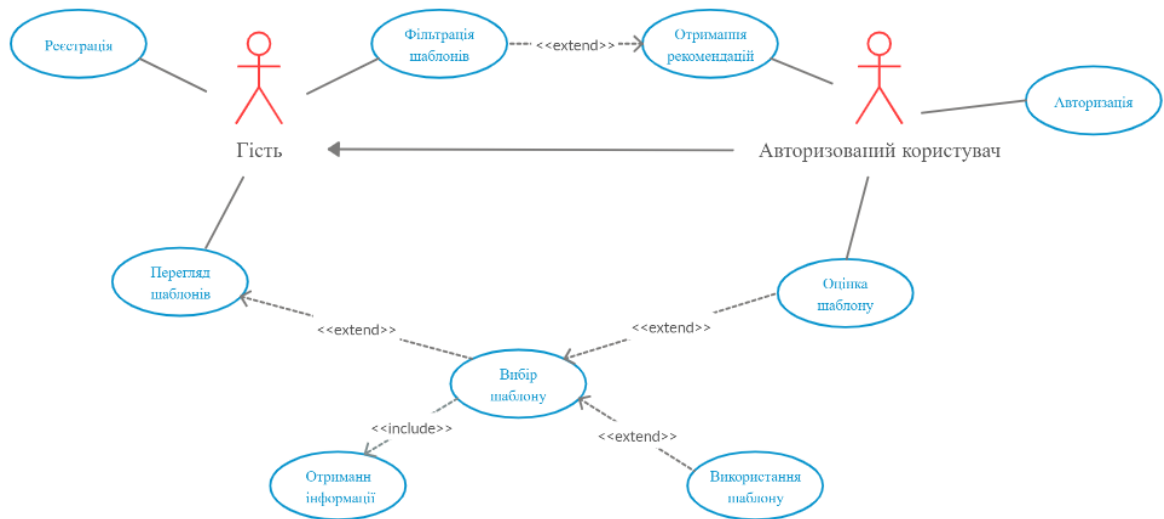


Рисунок 3.6 – Діаграма прецедентів системи

Далі наводиться сценарій для кожного прецеденту.

Use-case: «Реєстрація». Для отримання доступу до розширеного функціоналу сайту користувач може зареєструватися на ньому, вказавши необхідні відомості про себе.

Головна послідовність: користувач вводить ім'я, логін, адресу пошти і пароль, для перевірки коректності адреси пошти та пароля повторює їх введення в формі на відповідній сторінці і підтверджує введення. Система переводить користувача на сторінку з інформацією про профіль.

Альтернативна послідовність 1: введення вже існуючих логіна або пошти. Система видає повідомлення про те, що дані, пошта зайняті, і пропонує повторити введення.

Альтернативна послідовність 2: введення некоректного адреси пошти. Система видає повідомлення про те, що адреса електронної пошти введений помилково, і пропонує повторити введення.

Альтернативна послідовність 3: введення короткого пароля. Система видає повідомлення про те, що пароль менше чотирьох символів, і пропонує повторити введення.

Use-case: «Введення логіна і пароля». Якщо у відвідувача вже є профіль на сайті, то він може авторизуватися на ньому, вказавши свої логін і пароль.

Головна послідовність: користувач вводить логін і пароль у формі на сторінці авторизації і підтверджує введення. Система переводить користувача на сторінку з особистими проектами.

Альтернативна послідовність: невірний введення логіна та пароля. Система видає повідомлення про те, що логін або пароль введені невірно, і пропонує повторити введення.

Use-case: «Перегляд шаблонів». На головній сторінці сайту представлений каталог всіх шаблонів, що існують в базі даних. Користувач може переглядати їх, прокручуючи коліщатко мишки або використовуючи смугу прокрутки.

Use-case: «Фільтрація шаблонів». Якщо у відвідувача сайту немає бажання переглядати всі існуючі шаблони, він може виставити необхідні фільтри і переглянути тільки ті шаблони, які їм задовольняють. Також, є можливість сортувати прототипи за новизною і за коефіцієнтом використання.

Головна послідовність: на головній сторінці сайту в спеціальній формі користувач за бажанням вибирає вид шаблону, тип сайту і один або кілька тематичних тегів, потім підтверджує введення. Після цього сторінка перезавантажується і виводяться шаблони, відповідні виставленим фільтрам.

Альтернативна послідовність: в базі даних відсутні шаблони, що задовольняють виставленим фільтрам. Система виводить повідомлення про те, що шукані шаблони не існують.

Use-case: «Вибір шаблону». При перегляді шаблонів відвідувач сайту може вибрати зацікавив його шаблон, клікнувши на нього лівою кнопкою миші, і тим самим викликати появу спливаючого вікна з інформацією про вибраний шаблоні.

Use-case: «Отримання інформації про шаблон». У спливаючому вікні може бути присутнім інформація про вид та тип шаблону, коефіцієнт використання його іншими користувачами і тематичні теги.

Головна послідовність: користувач вибирає шаблон і автоматично спливає вікно, що містить більш детальну інформацію.

Use-case: «Використання шаблону». Крім перегляду шаблонів, користувачі також можуть їх редагувати або створювати власні, використовуючи конструктор з модулем рекомендацій. За рахунок цього буде поповнятися наявна БД. Однак, потрібно надати користувачеві можливість вибору, якщо він не хоче розміщувати свій проект у відкритому доступі.

Головна послідовність: користувач натискає на кнопку «Використовувати шаблон» і автоматично переходить в конструктор прототипів.

Use-case: «Оцінка шаблонів». Крім коефіцієнта використання шаблону, під кожним з них буде показуватися рейтинг. Він заснований на оцінках авторизованих користувачів сайту.

Головна послідовність: користувач виставляє оцінку шаблоном від одного до п'яти за допомогою вибору кількості зірок, розташованих під кожним прототипом поруч з коефіцієнтом використання.

Альтернативна послідовність: користувач не авторизований. Система виводить повідомлення про неможливість голосування і пропонує відвідувачу пройти авторизацію.

3.3 Розробка базового функціоналу РС для прототипування інтерфейсів

В якості основного підходу до побудови рекомендаційної системи були обрані методи, засновані на знаннях, а саме – використання жорстких обмежень. В рамках цього підходу послідовність дій така: користувачі

формулюють свої вимоги до прототипу, система намагається знайти підходящі шаблони в БЗ і видати їх в якості рекомендованих. Шукаються і рекомендуються тільки ті шаблони, які точно відповідають всім вимогам користувача.

В якості другорядних були обрані методи, які аналізують вміст об'єктів. На них заснована робота стрічки прототипів і пошук ключових слів для більш високої точності рекомендацій. За допомогою формули 1.6 розраховується коефіцієнт Дайса, який приймає значення з відрізка $[0,1]$ і показує наскільки близькі два вектора характеристик. На рисунках 4.1 – 4.10 представлений інтерфейс розробленого прототипу.

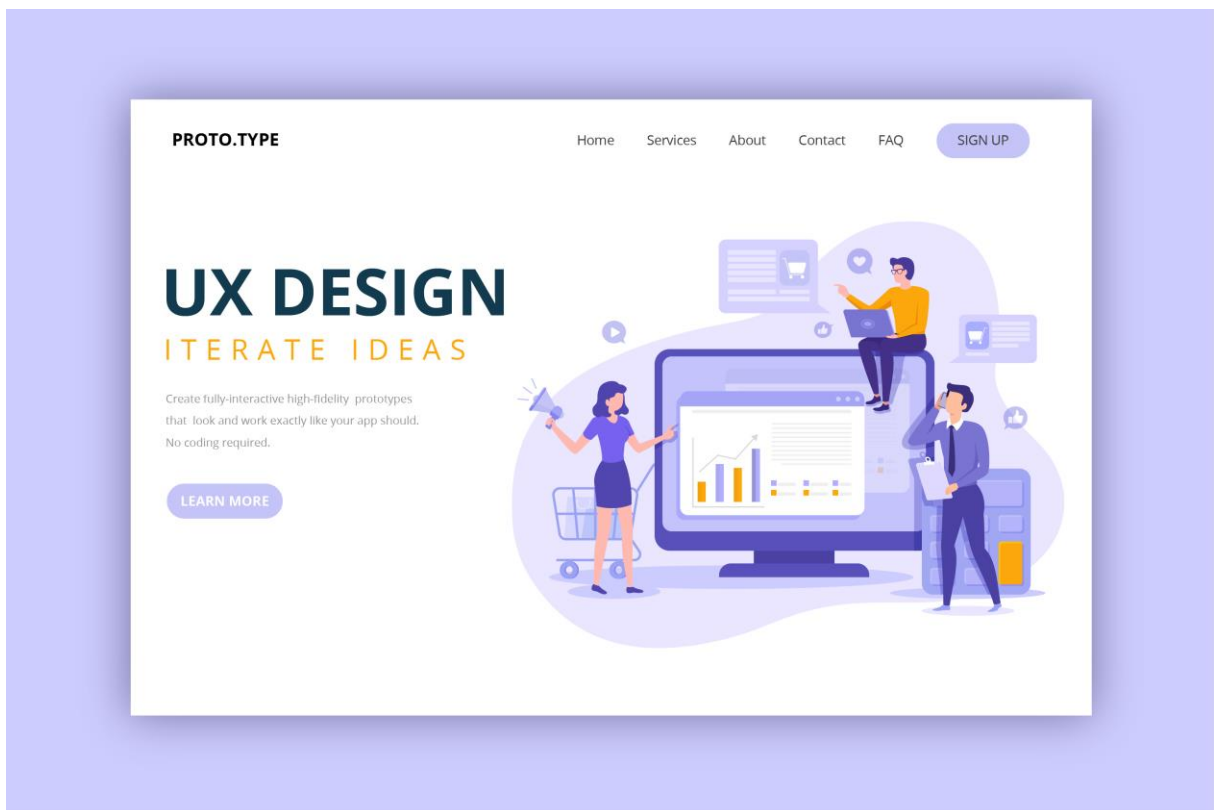


Рисунок 4.1 – Початкова сторінка

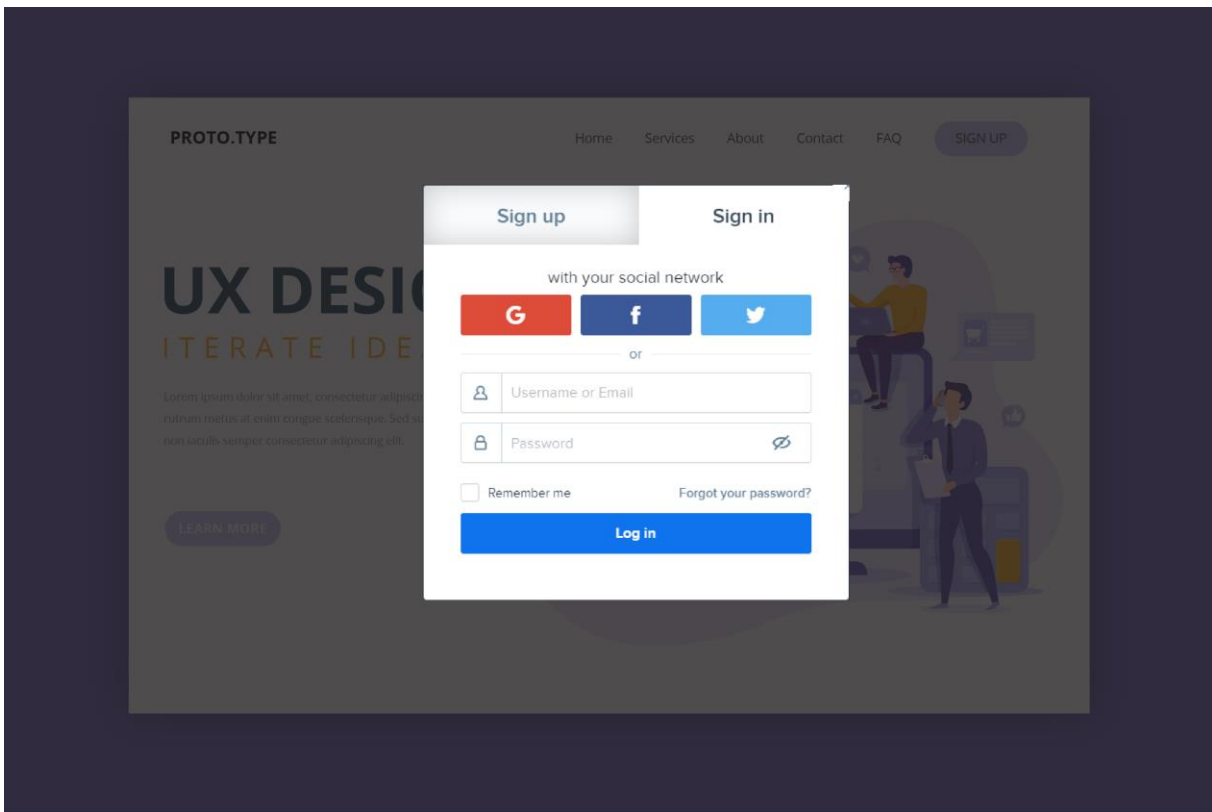


Рисунок 4.2 – Спливаюче вікно «Вхід/Реєстрація»

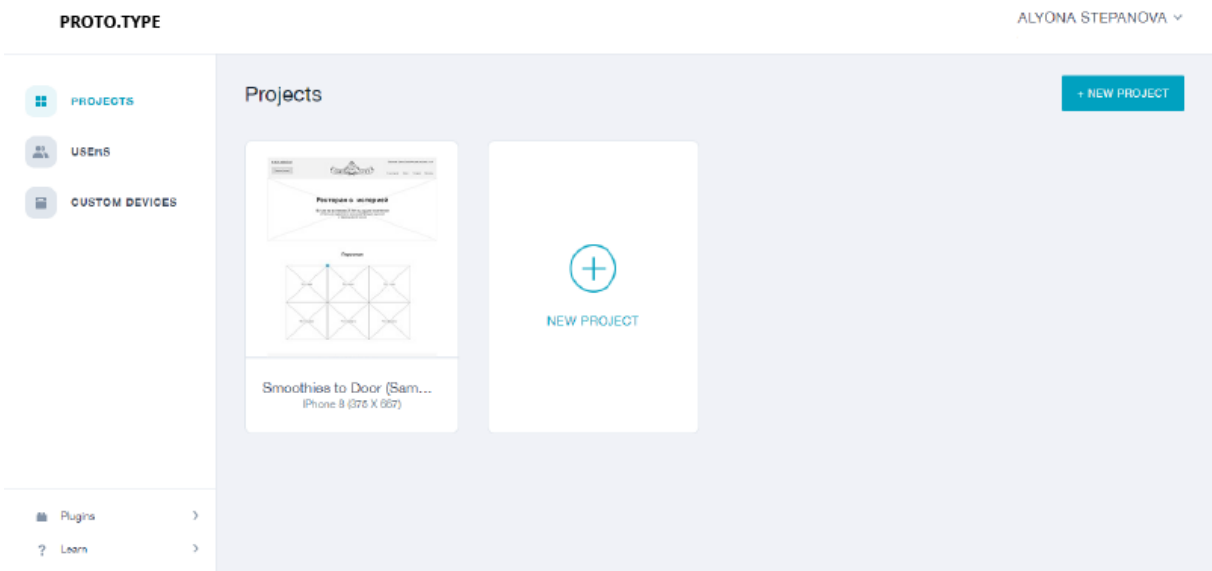


Рисунок 4.3 – Веб-сторінка «Особистий кабінет»

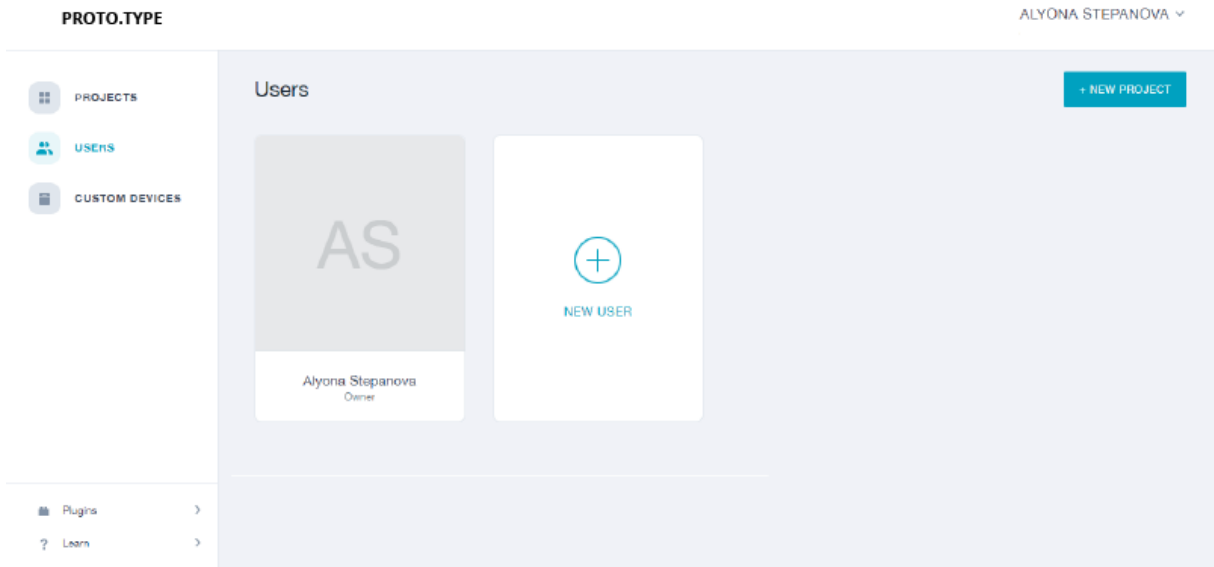


Рисунок 4.4 – Вкладка Users в особистому кабінеті

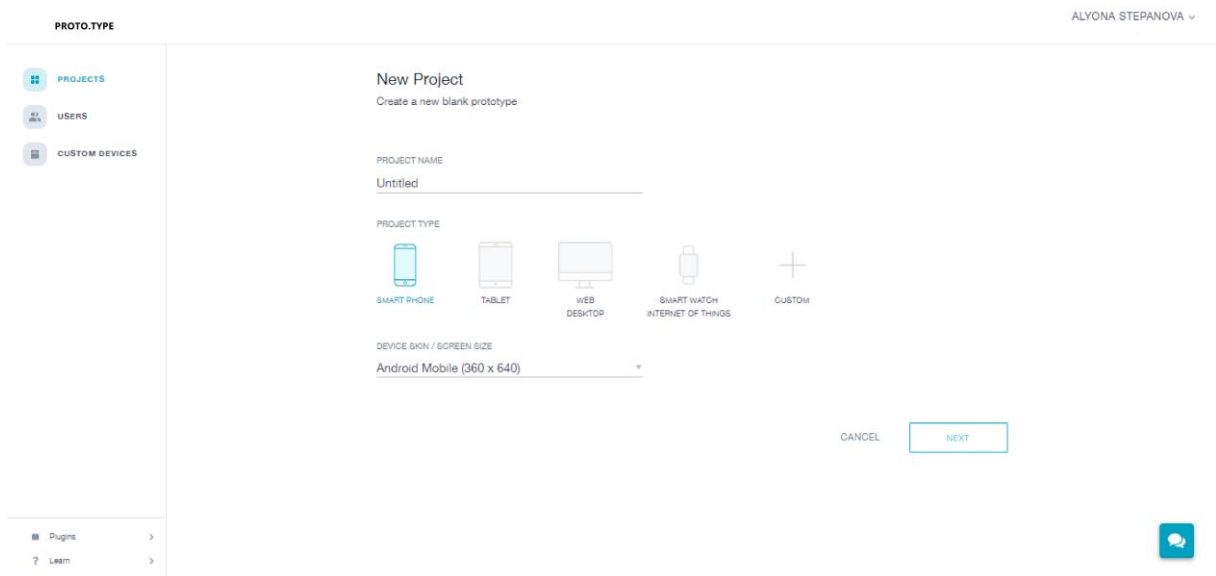


Рисунок 4.5 – Веб-сторінка «Вибір виду сайту»

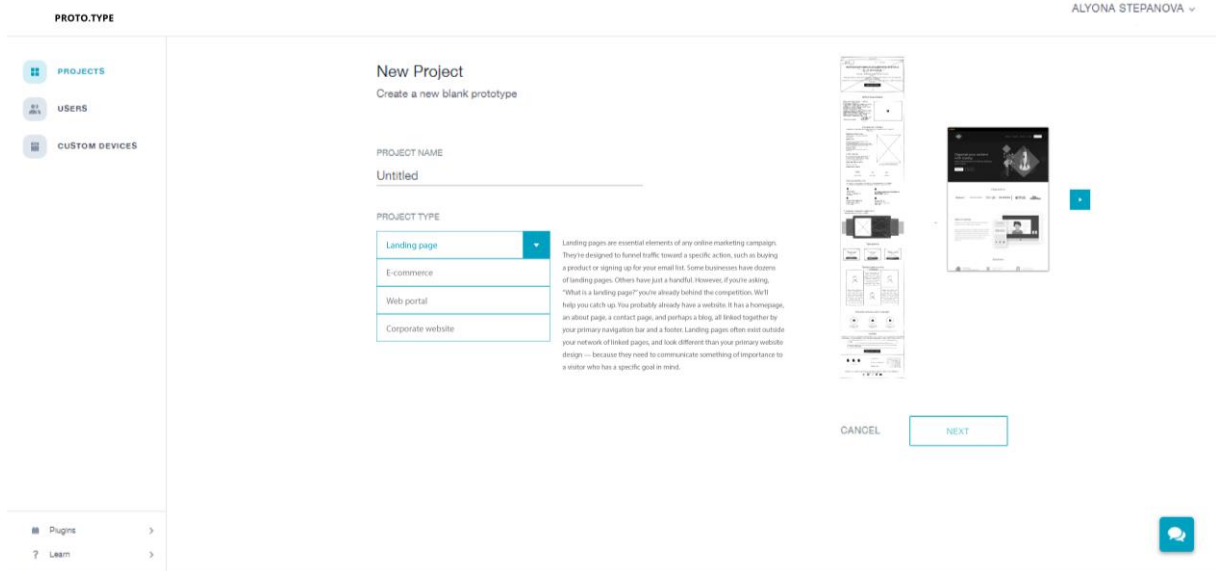


Рисунок 4.6 – Веб-сторінка «Вибір типу сайту»



Рисунок 4.7 – Фрагмент веб-сторінки «Вибір тегів»

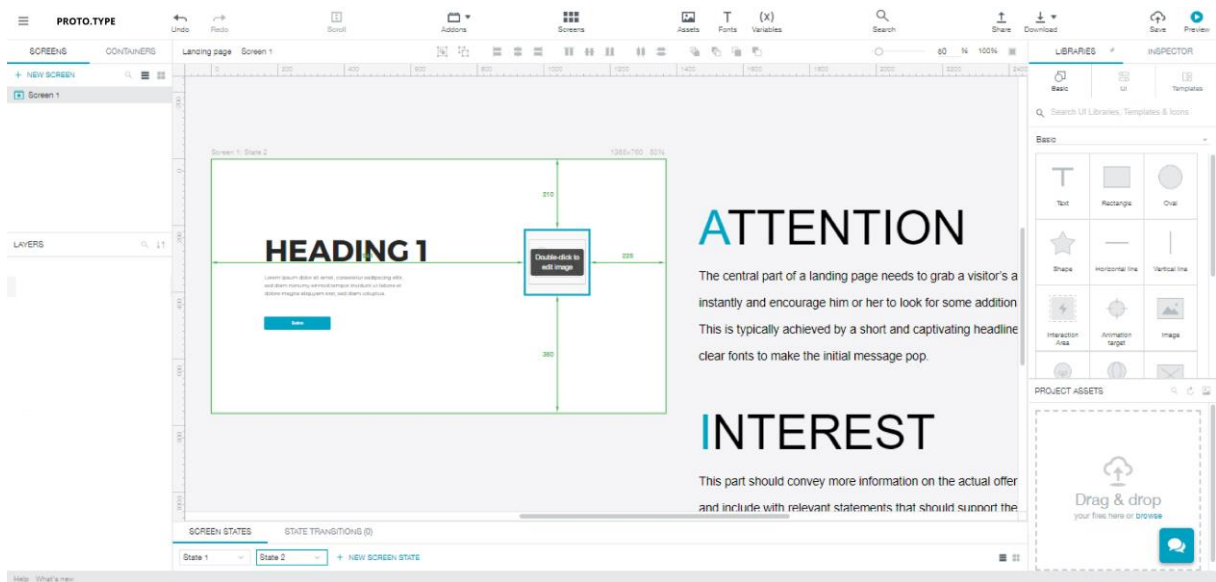


Рисунок 4.8 – Веб-сторінка конструктора прототипів

Розроблена система може допомогти UX-дизайнеру, а також звичайному користувачеві, в пошуку натхнення і розробці прототипів інтерфейсів. Алгоритм покрокового проектування і доступний опис кожного елемента дозволить зробити його правильним і якісним. Вироблення рекомендацій дозволяє підлаштовувати стрічку контенту і конструктор під запити користувача.

За допомогою методу фокус-груп було проведено тестування. Усі помилки виправлено. Створений прототип можна передати UI-дизайнеру для подальшої обробки.

4 АНАЛІЗ ОСНОВНИХ ПІДХОДІВ ДО ТЕСТУВАННЯ USABILITY ІНТЕРФЕЙСІВ

Правильно розроблена методологія тестування usability дає можливість отримати корисну інформацію та допомагає рухатися в правильному напрямку. Останні роки підходи до юзабіліті-аудитів сильно еволюціонували, але існує чотири основні методи: віддалене немодероване тестування, віддалене модероване, лабораторне модероване і фокус-групи. Всі вони переслідують одну і ту ж мету – оцінка зручності і зрозумілості елементів інтерфейсу сайту або програми, відрізняючись переважно форматом збору зворотного зв'язку від користувачів [16].

Віддалене немодероване юзабіліті-тестування полягає в створенні сценарію тестування, який виконується тестувальниками на ПК або смартфоні в зручній для них час і без участі модератора. Також, даний метод називається «думки вголос», і є одним з найбільш популярних в світі. Він відмінно підходить для задач, пов'язаних з перевіркою конверсій або збором відвертих думок про продукт. Найчастіше віддалені немодеровані тести використовуються для готових сайтів або додатків, в тому числі і конкурентів. Переваги та недоліки даного методу представлені в таблиці 4.1.

Таблиця 4.1 – Переваги і недоліки віддаленого немодерованого юзабіліті-тестування

Переваги	Недоліки
1	2
Даний спосіб дозволяє оперативно перевірити певну гіпотезу або отримати зворотній зв'язок від більшої кількості людей. Крім того, такі тести зазвичай значно менш витратні.	Статичність набору завдань не дає можливості змінити напрямок дослідження і уточнити деталі.

Продовження таблиці 4.1

1	2
<p>Віддалений формат взаємодії дозволяє дізнатися думку не тільки локальних користувачів, що підвищує якість оцінки і мінімізує локальну похибку отриманих результатів.</p>	<p>Відсутність модератора може утруднити збір зворотного зв'язку для прототипів або продуктів з неповним функціоналом, тому що у користувачів напевно виникне маса труднощів в роботі.</p>
<p>Особистий фактор і анонімність дозволяють отримати більш відвертий "нефільтрований" фідбек, а також учасники тестування знаходяться у звичному для себе середовищі. Отже, вони не відчують тиску з боку спостерігача, тому зазвичай поведуться максимально природно, що дозволяє тим самим знайти глибинні інсайти.</p>	<p>Оскільки респонденти в змозі самостійно брати участь у тестуванні і проходять їх день у день, вони напевно мають більш високий рівень комп'ютерної грамотності в порівнянні з середньостатистичним представником цільової аудиторії.</p>

Віддалене модероване тестування проводиться в форматі інтерв'ю за допомогою відео-конференції з метою виявлення проблемних моментів і очікувань аудиторії, а також збору невербальних сигналів. Зазвичай спілкування з респондентами ведеться вільному форматі, хоча завжди краще мати заготовлений заздалегідь набір завдань і питань. Найчастіше такі тести використовуються для тестування прототипів або бета-версій продуктів. Переваги та недоліки даного методу представлені в таблиці 4.2.

Таблиця 4.2 – Переваги та недоліки віддаленого модерowanego юзабіліті-тестування

Переваги	Недоліки
<p>Під час інтерв'ю можна задати уточнюючі питання або по міміці респондента зрозуміти його стан, відношення до продукту або його складових.</p>	<p>Оскільки респонденти в змозі самостійно брати участь у тестуванні і проходять їх день у день, вони напевно мають більш високий рівень комп'ютерної грамотності в порівнянні з середньостатистичним представником цільової аудиторії.</p>
<p>Відсутність витрат на поїздки та оренду або використання спеціального приміщення і техніки дозволяє значно здешевити проект, при цьому швидко отримати результати</p>	<p>Людський фактор під час спілкування респондента і модератора тестування завжди може відбитися на результатах дослідження, тому важливо працювати тільки з досвідченими модераторами, а такі завжди вимагають значних витрат</p>

Лабораторне модерowane тестування проводиться за аналогією з попереднім методом: поточний формат передбачає спілкування з тестером один на один, але при цьому відбувається наживо. Даний метод підходить для вивчення потреб користувачів. Жива зустріч полегшує можливість збору невербального фідбеку, тестування фізичних продуктів, але при цьому вимагає високої кваліфікованості модератора. Переваги та недоліки даного методу представлені в таблиці 4.3.

Таблиця 4.3 – Переваги і недоліки лабораторного модерowanego юзабіліті-тестування

Переваги	Недоліки
Під час інтерв'ю можна задати уточнюючі питання або по міміці респондента зрозуміти його стан, відношення до продукту або його складових	Якщо ви орендували лабораторію і найняли модератора, то неявка когось із тестерів витратить вкладені ресурси даремно і може поставити під сумнів результати дослідження
Перебуваючи в лабораторії, ніякі сторонні речі не відвернуть вас і респондента від тестування	Наявність локальних тестерів може привести до похибки

Фокус-група – це метод дослідження, при якому група користувачів запрошується для прийняття участі в тестуванні призначеного для користувача інтерфейсу. Як правило, користувачі, яким пропонується оцінити зручність інтерфейсу, є потенційними клієнтами. Проведення інтерв'ю з користувачами ведеться за підготовленим сценарієм. Користувачеві ставляться питання або завдання, які йому необхідно виконати при роботі з інтерфейсом програми. Всі дії користувача, реакція на певні елементи інтерфейсу, його репліки і емоції записуються на аудіо та відео носії для подальшого аналізу. Головне завдання проведення фокус-групи полягає в збільшенні показника юзабіліті сайту. Метод фокус-груп може застосовуватися на етапі тестування прототипу інтерфейсу додатку для збору і аналізу початкових думок про інтерфейс і доопрацювання елементів, які викликають питання. А також на етапі тестування інтерфейсу робочого додатку, за результатами якого пропонується список рекомендацій щодо його поліпшення. Проведення

фокус-груп дозволяє краще зрозуміти особливості поведінки кінцевих користувачів, що неможливо досягти іншими методами. Переваги та недоліки даного методу представлені в таблиці 4.4.

Таблиця 4.4 – Переваги і недоліки юзабіліті-тестування за допомогою фокус-груп

Переваги	Недоліки
Зробити фокус-групи з 8-12 учасниками і одним модератором щодо дешевше, ніж організувати окремі інтерв'ю	Людський фактор під час спілкування респондента і модератора тестування завжди може відбитися на результатах дослідження
Перебуваючи в лабораторії, ніякі сторонні речі не відвернуть вас і респондента від тестування	Наявність домінантної думки може привести до похибних результати дослідження

Для тестування розробленого прототипу системи рекомендаційного типу був використаний метод фокус-груп.

Висновки

В ході даної дослідницької роботи були вирішені такі задачі:

- проаналізовано основні підходи до побудови інтелектуальних веб-систем рекомендаційного типу. розглянути основні визначення, що стосуються систем рекомендаційного типу;
- проведено порівняльний аналіз існуючих підходів до розробки РС;
- розглянуті методи вироблення рекомендацій і алгоритми;
- проведено аналіз області застосування РС;
- провести аналіз предметної області, а саме поняття UX-дизайн і роботи фахівців даної сфери;
- виявлені основні проблеми, з якими стикається користувач при створенні прототипів;
- розглянуті існуючі системи прототипування інтерфейсів;
- проведено аналіз цільової аудиторії;
- розроблено прототип системи, яка підтримує покрокове проектування структури призначеного для користувача інтерфейсу у вигляді веб-ресурсу;
- проведено аналіз методик тестування usability;
- протестовано розроблений прототип системи рекомендаційного типу.

Розроблена система може допомогти UX-дизайнеру, а також звичайному користувачеві, в пошуку натхнення і розробці прототипів інтерфейсів. Алгоритм покрокового проектування і доступний опис кожного елемента дозволить зробити його правильним і якісним. Вироблення рекомендацій дозволяє підлаштовувати стрічку контенту і конструктор під запити користувача.

За допомогою методу фокус-груп було проведено тестування. Усі помилки виправлено. Створений прототип можна передати UI-дизайнеру для подальшої обробки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Пятикоп Е.Е. Исследование метода коллаборативной фильтрации на основе сходства элементов. URL : http://nbuv.gov.ua/UJRN/Npdntu_inf_2013_2_18 (дата звернення: 12.11.2019).
2. ООО «КасперСистемс» Что такое UX и UI дизайн – особенности и отличия. URL : <https://www.kasper.by/blog/что-такое-ux-i-ui-dizain/> (дата звернення: 13.11.2019).
3. UX PUB Перевернутый UX-айсберг на основе Instagram. URL : <https://ux.pub/perevernutyj-ux-ajsberg-na-osnove-instagram/> (дата звернення: 16.11.2019).
4. Jonathan P.R. URL : www.jonathanradford.com (дата звернення: 11.12.2019).
5. Яндекс.Метрика. URL : <https://metrika.yandex.ru/> (дата звернення: 12.12.2019).
6. Интернет-магазин товаров Comfy. URL : <https://comfy.ua/harkov/> (дата звернення: 11.12.2019).
7. Ex.Lpmotor Формула AIDA на лендинге. URL : lpmotor.ru/aida-landing (дата звернення: 11.12.2019).
8. Форум для дизайнеров. URL : <http://designforum.ru/> (дата звернення: 11.12.2019).
9. Gary Sheng. URL : <http://www.garysheng.com/> (дата звернення: 11.12.2019).
10. Serpstat Многофункциональная SEO-платформа. URL : <https://serpstat.com/ru/> (дата звернення: 11.12.2019).
11. Axure Powerful Prototyping and Developer Handoff. URL : <https://www.axure.com/> (дата звернення: 11.12.2019).
12. Design Prototyping with Origami Studio. URL : <https://origami.design/> (дата звернення: 11.12.2019).

13. Proto.io Prototypes that feel real. URL : <https://proto.io/> (дата звернення: 11.12.2019).
14. Vectr Free Vector Graphics Software. URL : <https://vectr.com/> (дата звернення: 11.12.2019).
15. InVision App Design better, faster, together. URL : <https://www.invisionapp.com/> (дата звернення: 11.12.2019).
16. Usetest.net Какой метод юзабилити-тестирования выбрать? URL : <https://usetest.net/methods-pros-cons/> (дата звернення: 11.12.2019).
17. Rosenfeld L., Morville P. Information Architecture for the World Wide Web. 2018. 109 p.
18. Sharp H., Rogers Y., Preece J. Interaction Design: Beyond Human-Computer Interaction. 2019. 205 p.
19. Grus J. Data Science from Scratch 2016. 187 p.
20. Shneiderman B., Plaisant C., Cohen M., Jacobs S., Elmqvist N., Diakopoulos N. Designing the User Interface: Strategies for Effective Human-Computer Interaction. 2018. 256 p.
21. Алена Салбер Как открыть магазин. 2017. 124 стр.
22. Унгер Р., Чендлер К. UX-дизайн: проектирование опыта взаимодействия. 2016. 105 стр.
23. Уэйншенк С. 100 главных принципов дизайна. 2015. 96 стр.