

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: **«ЗАСТОСУВАННЯ WEB-ТЕХНОЛОГІЙ
В РЕАЛІЗАЦІЇ МАРКЕТИНГ-АНАЛІЗУ
КОМПАНІЇ»**

Виконав(ла): студент(ка) 2 курсу, групи 8.1218

спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)

освітньої програми інженерія програмного забезпечення
(назва освітньої програми)

В.Ю. Коротенко

(ініціали та прізвище)

Керівник

доцент кафедри програмної інженерії,

доцент, к.ф.-м.н. Горбенко В.І.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент

завідувач кафедри фундаментальної

математики, доцент, д.т.н. Гребенюк С.М.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.		
2.	Збір вихідних даних.		
3.	Обробка методичних та теоретичних джерел.		
4.	Розробка першого розділу.		
5.	Розробка другого розділу.		
6.	Оформлення та нормоконтроль кваліфікаційної роботи.	27.12.2019	
7.	Захист кваліфікаційної роботи.	15.01.2020 (заочне: 10.01.2020)	

Студент _____
(підпис)

В.Ю. Коротенко _____
(ініціали та прізвище)

Керівник роботи _____
(підпис)

В.І. Горбенко _____
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

О.В. Кудін _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Застосування WEB-технологій в реалізації маркетинг-аналізу компанії» містить 97 с., 15 рис., 4 таблиці, 54 джерела, 1 додаток.

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ, ІНТЕРНЕТ-ТЕХНОЛОГІЇ, МАРКЕТПЛЕЙС, PYTHON, WINDOWS PRESENTATION FOUNDATION, XAML.

Об'єктом дослідження є процес реалізації маркетинг-аналізу маркетинг-плейсних підприємств на основі WEB-технологій.

Метою роботи є розроблення методичних і практичних рекомендацій щодо застосування WEB-технологій в реалізації маркетинг-аналізу маркетинг-плейсних підприємств.

Методи дослідження. У роботі використано загальнонаукові методи пізнання економічних явищ і процесів, фундаментальні положення теорії менеджменту і маркетингу.

В роботі запропоновано два варіанти застосування WEB-технологій в реалізації маркетинг-аналізу підприємства: оцінювання ефективності продажів підприємства, розрахунок вебметричного рангу застосування системи бізнес-аналітики SAS Integrated Marketing Management.

SUMMARY

Master's Qualification Thesis "The using of Web-technologies in the Implementation of marketing analysis of the company" : 97 pages, 15 figures, 4 tables, 54 references, 1 supplement.

INFORMATION TECHNOLOGIES, INTERNET TECHNOLOGIES, MARKETPLAYS, PYTHON, WINDOWS PRESENTATION FOUNDATION, XAML.

The object of the research is the process of marketing analysis of marketplace companies based on WEB technologies.

The aim of the work is to develop methodological and practical recommendations on the application of WEB technologies in marketing analysis of marketplace companies.

Research methods. The general scientific methods of knowledge of the economic phenomena and processes, fundamental provisions of the theory of management and marketing are used in the work.

The paper proposes two variants of application of WEB-technologies in realization of marketing-analysis of the enterprise: evaluation of efficiency of marketing communications of the enterprise, calculation of webometric rank and application of business analytics system SAS Integrated Marketing Management.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary.....	5
Вступ.....	7
1 Теоретичні положення застосування web-технологій в реалізації маркетинг-аналізу компанії.....	9
1.1 Теоретичні положення управління маркетинговими WEB-технологіями.....	9
1.2 Впровадження WEB-технологій як метод підвищення ефективності маркетингової діяльності на підприємстві.....	20
1.3 Постановка основних задач впровадження WEB-технологій в реалізації маркетинг-аналізу компанії	23
2 Основні web-технології, що використано в реалізації маркетинг-аналізу компанії.....	33
2.1 Оцінювання ефективності маркетингових комунікацій підприємства, розрахунок вебметричного рангу	33
2.2 Структура економіко-математичної моделі.....	34
2.3 Апробація економіко-математичної моделі.....	35
2.4 Програма статистичного аналізу продажів інтернет магазинів.....	37
3 Розробка функціоналу програми.....	42
3.1 Реалізація програми в середовищі розробки.....	42
Висновки.....	55
Перелік посилань	56
Додатки.....	61

ВСТУП

Актуальність теми. Процеси глобалізації економіки пов'язані з комп'ютеризацією майже всіх сфер діяльності людини. Нині світові інвестиції в інформаційні технології у півтора рази перевищують інвестиції в основні фонди виробництва. Це посприяло тому, що інформаційні технології сформували повноцінний ринок із власною інфраструктурою та обумовили появу такого виду економічної діяльності, як інформаційна економіка. Її головне завдання полягає у вивченні тенденцій розвитку, умов і факторів ефективного використання інформації та інформаційних технологій як ресурсу.

Актуальність теми зумовлена необхідністю впровадження у ринкову діяльність маркетингових підприємств нової концепції управління маркетинговими комунікаціями на основі WEB-технологій. Це ставить перед завдання визначення факторів та показників, які обумовлюють ефективність заходів маркетингових комунікацій у мережі Інтернет, збільшують віддачу від вкладених у просування коштів. Обґрунтована стратегія маркетингових комунікацій на основі WEB-технологій стає головним чинником у досягненні успіху компаніями на ринку.

Значий внесок у теорію, пов'язану з інформаційними та WEB-технологіями у маркетингу, зробили зарубіжні економісти Н. Бакман, Т. Бокар'юв, В. Бордюже, Т. Буреніна, А. Калініна, Д. Козьє, А. Кутлалієв, Ю. Піменов, У. Хенсон та багато інших.

Мета і завдання дослідження. Метою роботи є розроблення теоретичних підходів, методичних і практичних рекомендацій щодо застосування WEB-технологій в реалізації маркетинг-аналізу маркетингових підприємств.

Для досягнення мети сформульовано такі **завдання**:

- дослідити теоретичні положення управління маркетинговими WEB-технологіями
- оцінити впровадження WEB-технологій як метод підвищення ефективності маркетингової діяльності на підприємстві
- охарактеризувати постановку основних задач впровадження WEB-технологій в реалізації маркетинг-аналізу компанії
- провести оцінювання ефективності маркетингових комунікацій підприємства, розрахунок вебометричного рангу
- дослідити застосування системи бізнес-аналітики SAS Integrated Marketing Management
- дослідити реалізацію Застосування системи бізнес-аналітики SAS Integrated Marketing Management
- провести аналіз проведення вебометричного аналізу веб-сторінки

Об'єктом дослідження є процес реалізації маркетинг-аналізу маркетинг-аналізу підприємств на основі WEB-технологій.

Предметом дослідження є теоретико-методичні та науково-практичні основи застосування WEB-технологій в реалізації маркетинг-аналізу маркетинг-аналізу підприємств.

Методи дослідження. У роботі використано загальнонаукові методи пізнання об'єктивної природи економічних явищ і процесів, фундаментальні положення теорії менеджменту і маркетингу.

Структурно робота складається зі вступу, трьох основних розділів, висновків, списку використаної літератури, додатків.

1 ТЕОРЕТИЧНІ ПОЛОЖЕННЯ ЗАСТОСУВАННЯ WEB-ТЕХНОЛОГІЙ В РЕАЛІЗАЦІЇ МАРКЕТИНГ-АНАЛІЗУ КОМПАНІЇ

1.1 Теоретичні положення управління маркетинговими WEB-технологіями

На сучасному етапі розвитку ринкових умов та посилення конкурентної боротьби в Україні діяльність сучасних торговельних підприємств неможлива без використання та вдосконалення маркетингової політики комунікацій. В останні роки маркетингові комунікації набувають все більшого значення, що обумовлено об'єктивними причинами.

В сучасній науковій та методичній літературі відсутній єдиний підхід до визначення поняття „комплексу маркетингових комунікацій”, або „комунікацій-мікс”. Аналіз існуючих підходів дозволив поточнити визначення цього поняття. Кожний з елементів комплексу маркетингових комунікацій має свої специфічні характеристики.

Комплексний підхід до сутності прямого маркетингу: „Прямий маркетинг – це комунікації із зворотнім зв'язком з відібраним певним покупцем з метою отримання негайного відгуку або здійснення операції”.

Використання сучасних технологій дозволило поточнити склад і надати характеристику всім каналам прямого маркетингу. Це як повністю нові канали прямого маркетингу (телемаркетинг прямого відгуку та інтерактивний маркетинг), так і традиційні, але оновлені (індивідуальний продаж згідно комп'ютерних баз даних, прямий маркетинг поштою (факсимільна, електронна та голосова пошта, мобільний маркетинг – SMS-повідомлення), маркетинг за каталогом (електронні каталоги, торгівля цифровими товарами), телефонний маркетинг (мобільний маркетинг)).

В процесі перетворення стандартних маркетингових комунікацій в інтерактивні визначальну роль грають сучасні Інтернет-технології. Але на даний момент відсутнє чітке визначення такого поняття, як інтерактивний маркетинг, яке б повністю розкривало його сутність, ціль та засоби. Інтерактивний маркетинг доцільно визначати як комплекс інтерактивних маркетингових комунікацій, що охоплює маркетингові дослідження в Інтернеті, рекламу прямого відгуку, рекламні засоби Інтернет-комунікацій, телемаркетинг, який орієнтований на задоволення потреб споживачів та здійснюється за допомогою засобів Інтернет-маркетингу та комерційних інтерактивних служб.

Основним каналом інтерактивного маркетингу залишається Інтернет, який розглядається як один із засобів просування в комплексі маркетингових комунікацій. Однак проблемі впровадження сучасних Інтернет-технологій для здійснення ефективних маркетингових комунікацій на підприємствах приділяється недостатньо уваги.

Виявлено, що Інтернет-користувачі мають свою специфіку та пріоритети. Не можна порівнювати онлайн-торгівлю з традиційною, тому що це два зовсім різних види торгівлі.

К засобам Інтернет-маркетингу доцільно віднести: канали зв'язку в режимі реального часу, засоби електронної торгівлі, Інтернет-вітрини.

Ефективність інтерактивних маркетингових комунікацій визначається через обсяги продажу, що зумовлені Інтернет-рекламою, інформованістю про товар та інформованістю про міру переваги певної марки товару.

Інтернет-інтеграція ринкової діяльності підприємств сприяє оптимізації процесів управління, виробництва, маркетингу, документообігу й організації віддалених робочих місць. Застосування Інтернет-технологій дає можливість розширити, спростити і водночас прискорити комунікаційні можливості, знизити витрати та розвивати бізнес більш ефективно (див. рис. 1.1).



Рисунок 1.1 – Комунікаційні переваги інформаційних та Інтернет-технологій

Поняття «електронного бізнесу» полягає у реалізації бізнес-процесів із використанням інформаційно-телекомунікаційних технологій для проведення маркетингових досліджень; обміну інформацією; встановлення контактів; підтримки до та після продажу; продажу товарів і послуг; електронної оплати із використанням електронних платіжних систем; дистрибуції продуктів, включаючи управління доставками; можливості організації віртуальних підприємств.

Аналіз мережі Інтернет як середовища для маркетингової діяльності, узагальнення світових і вітчизняних наукових напрацювань у сфері комунікацій дозволяє визначити маркетингові комунікації на основі Інтернет-технологій як засоби і заходи просування (реклама, PR, стимулювання збуту, директ-маркетинг) на основі глобальної телекомунікаційної мережі інформаційних та обчислювальних ресурсів, які

підприємства використовують для реалізації комунікаційної і комерційної діяльності.

Структурно-логічна схема управління комунікаціями в Інтернеті представлено в Додатку 1.

На відміну від традиційних засобів просування, Інтернет переважає за низкою характеристик: багатофункціональною моделлю передавання повідомлень; наявністю контролю над пошуком інформації з боку споживачів; ширшим діапазоном видів подання інформації; нелінійним пошуком інформації, який обумовлений гіпермедійним характером її подання; можливістю реалізації інтерактивного спілкування.

Маркетингові комунікації в електронному середовищі характеризуються двоступеневою взаємодією, де першою ланкою є зовнішні засоби просування (контекстна реклама, банери, колажі, дошки оголошень тощо), а другою та головною ланкою – електронний ресурс підприємства. Завдяки Інтернету комунікації є більш цілеспрямованими, що нівелює необхідність дорогих та неточних методів оцінювання ефективності. Властивості мережі Інтернет як носія інформації істотно змінюють традиційні погляди на процес комунікації і вимагають вирішення конкретних завдань у сфері інтерактивних інструментів просування. На сьогодні існує необхідність системного підходу до використання Інтернету як носія маркетингових комунікацій: від формулювання цілей, визначення інструментів і засобів просування до програми реалізації і способів оцінювання ефективності.

Аналіз параметрів для визначення ефективності просування дав змогу виявити, що вони є універсальними для оцінювання будь-яких носіїв інформації та пов'язані з аналізом впливу комунікаційного повідомлення, ефективності медіапланування та економічною ефективністю. Водночас обґрунтовано необхідність виявлення додаткових показників, які описують ефективність електронного ресурсу окремо від решти засобів просування, а

також дослідження наявних методів оцінювання маркетингових комунікацій та можливість їх застосування відносно Інтернету.

Маркетингова комунікативна політика має такі специфічні інструменти в Інтернет:

Корпоративний сайт – це набір інформаційних блоків і інструментів для взаємодії з одним або декількома сегментами цільової аудиторії. Яка інформація буде представлена на ньому, які інструменти будуть задіяні, як вони взаємодіятимуть між собою – все це залежить від вибраної моделі бізнесу, короткострокових і довгострокових задач, а також від типу сегментів цільової аудиторії і можливості контактувати з нею тим або іншим способом [5].

Переваги корпоративного сайту як інструменту комунікативної політики в Інтернет [11]:

- збільшення продажів, розширення ринків збуту і утримання існуючих клієнтів;
- повна всебічна презентація продукції і послуг підприємства без географічних обмежень, на будь-якій мові, в будь-який зручний для споживачів час;
- оптимізація комунікацій з споживачами, партнерам;
- забезпечення ефективного зв'язку з контрагентами з будь-яких країн світу і можливість оперативного реагування на них, здійснення консультацій та забезпечення безпосереднього зв'язку з кінцевими споживачами продукції;
- посилення ринкової позиції та підвищення рівня довіри до підприємства

Контекстна реклама – принцип розміщення реклами, коли реклама орієнтується на зміст Інтернет-сторінки вручну або автоматично. Контекстна реклама може бути у вигляді текстових блоків, графічних банерів та відеороликів, може розміщуватися як на сторінках результатів пошуку, так і на сайтах, що встановили блоки контекстної реклами на своїх сторінках [12].

Переваги контекстної реклами:

- звертання тільки до цільової аудиторії;
- миттєва видимість у видачі пошукової системи;
- можливість оперативного аналізу та коригування рекламної кампанії;
- не потрібно оптимізувати сайт;
- керованість;
- «зручний» бюджет (клієнт сам визначає, скільки грошей готовий витратити).

Медійна реклама – це розміщення текстово-графічних рекламних матеріалів на сайтах, що представляють собою рекламний майданчик. Може використовуватися в якості носія для контекстної реклами. Наявність у банера гіперпосилання, можливості анімованого зображення і можливості дзвінка з банера на мобільний телефон (WOW-call) значно розширюють вплив медійної реклами.

Переваги медійної реклами:

- дозволяє генерувати попит (залежить від вибору запитів);
- впливає на емоції та підсвідомі мотиви;
- добре запам'ятовується;
- працює навіть якщо не було переходу на сайт;
- дозволяє охопити велику аудиторію.

Реклама на віджетах (від англ. widget) – контент-модуль, що вбудовується у веб-сторінку або у браузер, який заснований на веб-технологіях, що працюють через браузер: HTML, Flash тощо. Віджет являє собою маленьку програмку, яку можна без зусиль вбудувати в сайт. Він розробляється з розрахунком на те, що блогери, власники сайтів, учасники соціальних мереж будуть вбудовувати віджет в свої сторінки, тим самим забезпечуючи його творця додатковою рекламою.

Переваги віджетів:

- не сприймаються як безпосередньо реклама;

- мають здатність поширюватися;
- припускають інтерактивність;
- представляють інтерес і користь для користувачів;
- постійно оновлюються.

Продакт-плейсмент в он-лайн іграх – передбачає інтеграцію рекламованого продукту або бренду в ігровий процес. Це прийом неявної (прихованої) реклами, що полягає в тому, що реквізит, яким користуються герої має реальний комерційний аналог. Зазвичай демонструється продукт чи його логотип, або згадуються його хороші якості. В іграх найбільш підходящими є економічні он-лайн ігри та бізнес-симулятори, де товари, послуги і бренди є основними елементами он-лайн гри.

Переваги продакт-плейсмент в он-лайн іграх:

- забезпечує лояльність аудиторії до брендів, які рекламуються таким чином;
- дозволяє встановити високий рівень контакту рекламодавця з кожним користувачем.

Крос-брендінг (ко-брендінг) – представляє об'єднання двох брендів з метою підвищення продажів кожного з них, підвищення пізнаваності серед аудиторії іншого бренду, та за рахунок цього – розширення власної аудиторії [13]. Цей інструмент застосовується як в он-лайн так і в офф-лайн маркетингу.

Переваги крос-брендінгу [14]:

- скорочення рекламних витрат завдяки спільному проведенню акцій;
- встановлення довгострокового співробітництва з компаніями з інших сфер бізнесу;
- активний відгук у споживачів, тому що вони можуть отримати подвійну користь;
- ефект синергії – результат від спільних крос-акцій вдвічі більше, ніж від акцій окремо;

- розширення сфери збуту послуг або товарів;
- збільшення популярності компаній, що проводять крос-маркетинг.

Афілійований маркетинг. Мережі, побудовані за принципом афілійованого маркетингу, виплачують партнерам винагороди за залучення нових клієнтів, реєстрацію користувачів або продажі, що відбулися завдяки цьому залученню. Ця форма Інтернет-маркетингу перегукується з іншими, її основною метою є отримання нової аудиторії завдяки вже існуючій.

Переваги афілійованого маркетингу:

- можливість отримання додаткових клієнтів;
- можливість збільшення доходу компанії.

SMM (social media marketing – просування в соціальних мережах) та SMO (social media optimization – оптимізація сайту під соціальні мережі) – включає створення своєї тематичної групи та її подальше просування; залучення нових учасників до групи; інформування учасників про події компанії; підтримка інтересу до групи через розміщення контенту, запрошення до різних подій учасників мережі та інше.

Переваги реклами в соціальних мережах [15]:

- можливість вибору цільової аудиторії з точним таргетингом;
- постійна підтримка контакту з потенційними клієнтами;
- можливість інформування клієнтів про нові пропозиції компанії;
- збільшення лояльності до компанії.

SEM (search engine marketing – пошуковий маркетинг): внутрішній SEO (search engine optimization – пошукова оптимізація) та зовнішній. Пошуковий маркетинг (від англ. search engine marketing, SEM) – комплекс заходів, спрямований на збільшення відвідуваності сайту його цільовою аудиторією з пошукових машин. Зовнішня пошукова оптимізація передбачає залучення платного трафіку на сайт із зовнішніх джерел (наприклад, за допомогою розміщення контекстної реклами в Google Adwords та покупки вхідних посилань на інших сайтах). Внутрішня оптимізація передбачає залучення

органічного трафіку на сайт за допомогою його внутрішньої оптимізації для пошукових машин. По суті, пошукової маркетинг займається перерозподілом трафіку в Інтернеті з місць, менш релевантних запиту в місця з більшою релевантністю.

Переваги пошукового маркетингу:

- можливість залучити цільову аудиторію, тобто саме ту аудиторію, яка сама зацікавлена в придбанні товарів і послуг замовника;
- контакт з користувачем пошукової системи відбувається у той момент, коли він дійсно цікавиться рекламованим продуктом;
- користувач не підозрює про те, що йому показують рекламу – він сприймає пошукову систему як експерта, який завжди прийде на допомогу з цінною порадою;
- порівняно невисока вартість послуги просування сайту.

Сайти рейтингу й порівняння товарів – сайти, що спеціалізуються на оцінці товарів експертами або користувачами. Тут розміщуються замітки редакторів, відео-демо, рейтинги користувачів, а також думки експертів. Відгуки, як правило, бувають як позитивними, так і негативними.

Переваги:

- позитивний вплив на репутацію компанії
- досягнення ефекту «сарафанного радіо», що буде впливати на підвищення рівня продажів.

Подкастинг (від англ. Podcasting, від iPod і англ. Broadcasting – повсюдне мовлення, широкомовлення) – процес створення і поширення звукових або відеофайлів (подкастів) у стилі радіо-і телепередач в Інтернеті (мовлення в Інтернеті) [16].

Переваги подкастингу:

- дозволяє істотно збільшити показник відвідуваності ресурсу;
- можливість представити інформацію у більш наглядному вигляді;
- нові можливості спілкування з цільовими аудиторіями.

Блоггинг. Блог це різновид динамічного тематичного сайту, на якому відбувається спілкування людей у вигляді обговорення тих чи інших питань, які підіймає автор блогу або ж його відвідувачі .

За даними дослідження Факультету інженерної справи и прикладних наук Колумбійського університету відомо, що близько 30% компаній розглядають блоги як засобу комунікації з потенційними клієнтами. Всі блоги передбачають зворотний зв'язок з читачами, які можуть залишати свої коментарі під публікацією у блозі.

Переваги блоггингу:

- форма блогу більш зручна для взаємодії групи, ніж електронна пошта, миттєві повідомлення і т.д.;

- можливість створити віртуальну спільноту з визначеної тематики. Унікальний, цікавий контент, який розміщено на блозі, здатний залучити досить велику аудиторію і створити його автору репутацію фахівця в тій чи іншій галузі. Що в свою чергу поповнить ряди його лояльних покупців і потенційних клієнтів.

Вірусний маркетинг – поширення інформації в прогресії, близької до геометричної, де головним розповсюджувачем інформації є самі одержувачі інформації, шляхом формування змісту, здатного залучити нових одержувачів інформації за рахунок яскравої, творчої, незвичайної ідеї або з використанням природного або довірчого послання. Особливу ефективність метод придбав завдяки розповсюдження контенту в соціальних мережах за допомогою таких функцій як «поділитися» (share) або «мені подобається» (like). Причому суспільство сприймає це як розвага, а не як рекламу.

Переваги вірусного маркетингу:

- швидкість створення і легкість розміщення інформації;
- викликає велику ступінь довіри потенційних споживачів до рекламованого таким способом товару чи послуги.

Прямий (директ-) маркетинг – відноситься до зовнішньої активності компанії в Інтернеті (outbound marketing – активність поза сайтом), тобто поширення маркетингового повідомлення компанії поза сайтом. В цифровому середовищі він набуває своїх особливостей. В основному він реалізується у вигляді персональної email-, Skype-, RSS-, ICQ- та інших прямих розсилок.

Переваги директ-маркетингу:

- одержувачі листів власноруч підписуються на розсилку, що забезпечує цілеспрямованість інформаційного впливу;
- можливість налагодження постійних ділових стосунків зі споживачами, вимірювання їхньої реакції на відповідні звернення;
- можливість отримання швидкої реакції ринку на зроблені комерційні пропозиції;
- висока оперативність створення інформаційних повідомлень та можливість їх коригування залежно від реакції споживачів.

e-CRM-системи (Customer Relationship Management) – корпоративні Інтернет-системи сервісного типу, за допомогою яких в Інтернет виносяться системи комунікацій компанії з клієнтами. Основна задача – організації служб маркетингу, продажів і сервісу «клієнтоорієнтованим» чином. Інтернет, як нове комунікаційне середовище, дозволив розширити область вживання e-CRM систем і привнести в них нову якість.

Переваги e-CRM-систем:

- забезпечує збір через Інтернет спілкування та збереження інформації про клієнтів;
- дозволяє робити аналітичні висновки на базі достовірної інформації та на цій основі приймати ефективні рішення щодо взаємодії з клієнтами;
- підтримує адресну розсилку, створення списків за критеріями;
- забезпечує індивідуальний підхід до обслуговування клієнта з мінімальними затратами.

1.2 Впровадження WEB-технологій як метод підвищення ефективності маркетингової діяльності на підприємстві

Розвиток інформаційних технологій, серед яких одне з ключових місць зайняв Інтернет, поява і бурхливе зростання електронної комерції стали основою для появи нового напрямку в сучасній концепції маркетингу взаємодії - інтернет-маркетингу.

Під терміном інтернет-маркетинг розуміється теорія і методологія організації маркетингу в гіпермедійному середовищі Інтернету.

Інтернет володіє унікальними характеристиками, що значно відрізняються від характеристик традиційних інструментів маркетингу. Однією з основних властивостей середовища Інтернету є її гіпермедійна природа, що характеризується високою ефективністю у представленні і засвоєнні інформації, що значно підвищує можливості маркетингу в посиленні взаємозв'язку підприємств і споживачів.

Крім того, роль, що виконується інтернетом, не обмежуються лише комунікативними функціями, а також включає можливість висновку операцій, здійснення покупок і проведення платежів, додаючи йому межі глобального електронного ринку.

Використання Інтернету привносить нові особливості і переваги в порівнянні з маркетингом, заснованому на традиційних технологіях. Ось деякі з них:

Перехід ключової ролі від виробників до споживачів.

Однією з найбільш фундаментальних якостей, привнесених Інтернетом в світ сучасної комерції, є перехід ключової ролі від виробників до споживачів. Інтернет зробив реальністю для компаній можливість привернути увагу нового клієнта всього за десятки секунд, проведених їм перед екраном комп'ютера. Проте в той же час він дав можливість тому ж користувачеві за декілька клацань миші перейти до будь-якого з конкурентів.

У такій ситуації увага покупців стає найбільшою цінністю, а встановлені взаємини з клієнтами головним капіталом компаній.

Глобалізація діяльності і зниження трансакційних витрат.

Інтернет значно змінює просторовий і часовий масштаби ведення комерції. Він є глобальним засобом комунікації, що не має яких-небудь територіальних обмежень, при цьому вартість доступу до інформації не залежить від віддаленості від неї, в протилежність традиційним засобам, де ця залежність прямопропорційна. Таким чином, електронна комерція дозволяє навіть найдрібнішим постачальникам досягати глобальної присутності і займатися бізнесом у світовому масштабі. Відповідно, замовники також дістають можливість глобального вибору зі всіх потенційних постачальників, що пропонують необхідні товари або послуги незалежно від географічного розташування. Відстань між продавцем і покупцем грає роль лише з точки зору транспортних витрат вже на етапі доставки товарів.

Часовий масштаб в середовищі інтернету також значно відрізняється від звичайного. Висока ефективність комунікативних властивостей інтернету забезпечує можливість скорочення часу на пошук партнерів, ухвалення рішень, здійснення операцій, розробку нової продукції, і так далі. Інформація і послуги в інтернеті доступні цілодобово. Крім того, його комунікативні характеристики володіють високою гнучкістю, що дозволяє легко робити зміни в представленій інформації, і, тим самим, підтримувати її актуальність без тимчасової затримки і витрат на поширення.

Названі ефекти також приводять до значного скорочення трансакційних витрат, тобто витрат, пов'язаних з налагодженням і підтримкою взаємодії між компанією, її замовниками і постачальниками. При цьому вартість комунікацій, в порівнянні з традиційними засобами, стає мінімальною, а їх функціональність і масштабованість значно зростають.

Персоналізація взаємодії і перехід до маркетингу «один-одного».

Використовуючи засоби електронної взаємодії, компанії можуть отримувати детальну інформацію про запити кожного індивідуального замовника і автоматично надавати продукти і послуги, відповідні індивідуальним вимогам. Одним з простих прикладів може служити персональне представлення web-сайту для кожного з клієнтів або партнерів компанії.

Зниження трансформаційних витрат.

Зниження трансформаційних витрат може досягатися за рахунок оптимального вибору структури товарного асортименту, скорочення часу на розробку і впровадження нової продукції, обґрунтованої політики ціноутворення, зниження числа посередників, витрат на збут і так далі.

Розглянемо деякі обмеження і особливості використання Інтернету в маркетинговій діяльності:

1) стійка конкуруюча перевага не може бути отримане виключно з доступу до Інтернету або розробки web-сайту. Оскільки число фірм, що постійно збільшується, набуває необхідних знань, кваліфікації і технології, щоб з'єднатися з WWW, конкуруюча перевага не може бути досягнута лише за допомогою стандартних способів, в яких ця технологія розвивається;

2) попит на продукцію підприємства виникає не на підставі розробки web-сайту, незалежно від проникнення в Інтернет. На рівень витрат споживача інколи неможливо впливати. Це означає, що навіть при тривалому використанні WWW або інших пов'язаних з Інтернетом застосувань не можна чекати збільшення річного доходу всіх фірм, хоча деякі з них отримують певний додатковий прибуток;

3) маркетингові проблеми підприємства не можуть бути вирішені на підставі доступу в Інтернет. Наприклад, фірми, що не займаються експортом, не можуть стати експортерами раптово, на підставі розробки і збереження web-сайтів. Розробка специфічної для експорту інфраструктури в межах фірми є досить дорогою. Крім того, маркетинг експорту включає безліч

макро- і мікрообмежень, пов'язаних з плануванням і управлінням, включаючи стандарти для конкретних виробів (програм, продуктів), цільової тарифікації ринку і чинників конкуренції, експортної валюти і проблем платежу, підтримки замовника і сервісних вимог, юридичних і регулюючих обмежень і так далі. Жодна з цих проблем не може бути вирішена на підставі наявності web-сторінки або доступу в Інтернет;

4) у Інтернеті не можна «зробити те ж саме, що раніше робилося поза Інтернетом, але набагато дешевше»;

5) велике значення має проблема захисту безпеки зв'язку через інтернет. Цього обмеження набуває величезне значення у випадках, коли Інтернет повинен послужити серйозним поштовхом в розвитку бізнесу. Хоча вчені пропонували деякі варіанти вирішення цієї проблеми, захист (безпека) даних залишиться одним з основних питань для будь-якої фірми, яка хоче інтегрувати Інтернет в бізнес. Як тільки з'являється вирішення проблеми безпеки, відразу ж розробляються нові способи проникнення і декодування переданих даних і захист останніх залишається невирішеною проблемою для підприємства.

1.3 Постановка основних задач впровадження WEB-технологій в реалізації маркетинг-аналізу компанії

Аналіз основних суб'єктів ринку в українському секторі мережі Інтернет виявив п'ять можливих форм реалізації ринкової діяльності вітчизняними підприємствами на основі Інтернет-технологій: виробничу, збутову, рекламну, суспільну та посередницьку (див. рис. 1.2.).

Детальний розгляд кожної з виявлених форм засвідчив, що вони керуються різними цілями, мають різну структуру організації бізнесу та різні шляхи отримання прибутку. Застосування Інтернету дає можливість

розширити традиційну модель підприємницької діяльності, автоматизувати й об'єднати функції передачі інформації, покращити якість послуг, підвищити ефективність виробництва, знизити вартість оброблення замовлень. Інтернет є джерелом додаткового прибутку від продажів фізичних і електронних товарів, дає можливість скоротити витрати і час на транзакції з дистрибуції товарів, управління відносинами зі споживачами, створення нових продуктів, поширення інформації.

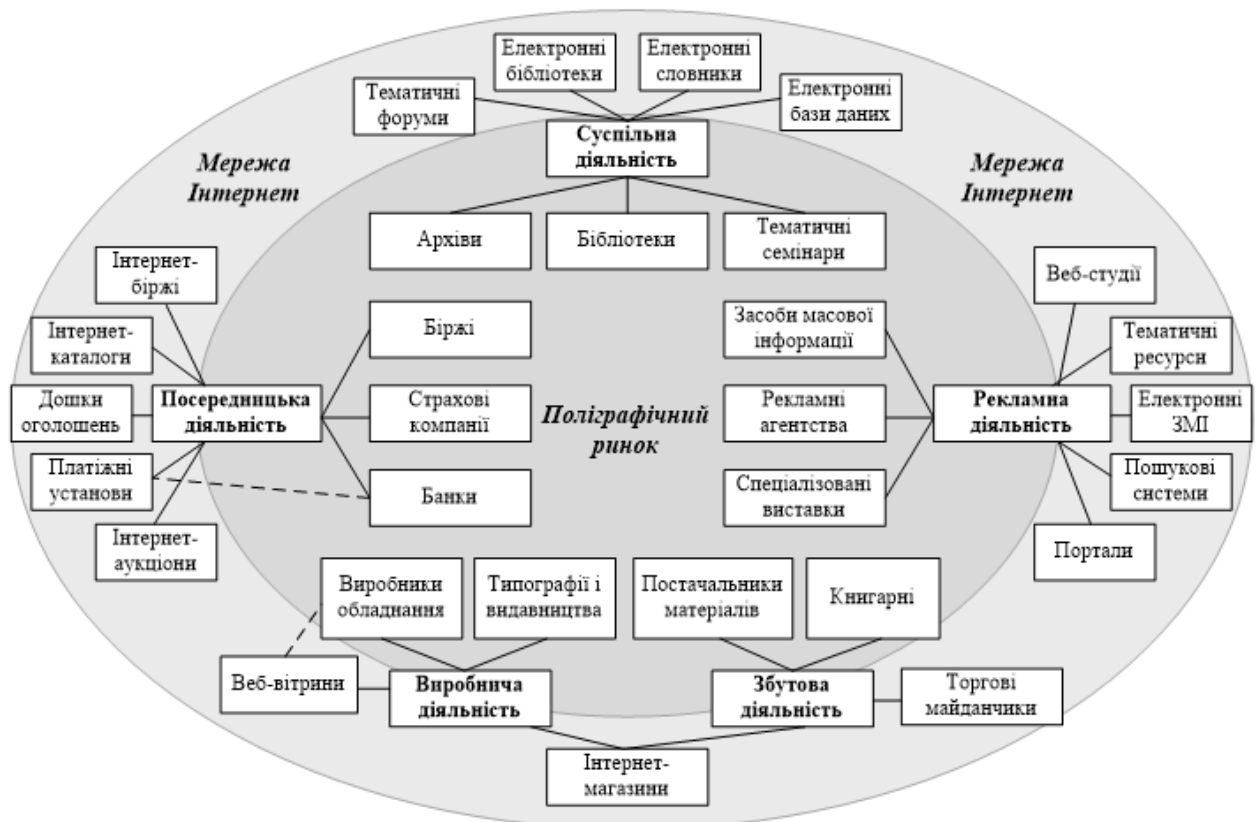


Рисунок 1.2 – Форми ринкової діяльності підприємств в Інтернеті

Дослідження підходів до визначення ефективності засобів і заходів маркетингових комунікацій засвідчив різноманіття показників, які було зведено до двох основних груп: показників комерційної ефективності та показників комунікаційної ефективності (див. табл. 1.1).

Таблиця 1.1 – Критерії та показники ефективності маркетингових комунікацій

Критерій	Показник	Одиниця виміру
Комунікаційна ефективність		
Ефективність комунікаційного звернення	Рівень відомості бренду	Співвідношення тих, хто знає бренд, до тих, хто не знає, %
	Розуміння змісту комунікаційного звернення	Кількість споживачів, здатних відновити у пам'яті характер та зміст повідомлення, осіб
	Рівень лояльності до товару, марки, компанії	Співвідношення лояльності до бренду за шкалою, %
	Рівень сприйняття атрибутів іміджу	Рівень знання атрибутів бренду, %
Комерційна ефективність		
Рентабельність	Рентабельність просування	Співвідношення прибутку від просування до витрат, %
	Рентабельність активів	Співвідношення прибутку за період до величини активів за період, %
Рівень продажів	Приріст середньомісячного товарообігу	Співвідношення поточного рівня продажів до попереднього за періоди, %
	Обсяг покупок у грошовому вираженні	Співвідношення поточного рівня продажів до попереднього за період, грн
	Частота і кількість покупок	Співвідношення частоти і кількості покупок на одного клієнта за періоди, шт. / од.
Прибуток	Динаміка виручки	Зростання прибутку від реалізації продукції, %
	Рівень продажів	Зростання прибутку від реалізації товару, грн.
Частка ринку	Частка ринку за кількістю продажів товару	Співвідношення обсягів продажів на ринку до кількості проданих товарів, грн
	Частка постійних клієнтів	Співвідношення постійних клієнтів до загальної кількості клієнтів, %
Дохід	Розмір валового доходу	Зростання валового доходу від реалізації готової продукції, грн.

Суттєвим недоліком виявлених показників є те, що майже всі вони розраховані на традиційні носії інформації, не беруть до уваги властивості Інтернету, де зношення і неефективна частота можуть бути ліквідовані електронними засобами просування. Окрім цього, не взято до уваги ефективність електронного ресурсу як невід'ємного інструменту комунікаційної взаємодії в мережі, не оцінено його вплив на досягнення поставлених цілей просування. Виявлено додаткові показники, що визначають ефективність електронного ресурсу підприємства в мережі Інтернет, які було систематизовано та зведено до таких п'яти основних груп критеріїв: технологічність, інформаційне наповнення, дизайн, навігація та інтерактивність електронного ресурсу (див. табл. 1.2).

Таблиця 1.2 – Загальні показники комунікаційної ефективності електронного ресурсу

№ п/п	Критерій	Показник
1	Технологічність	<ul style="list-style-type: none"> – відповідність імені сайту його тематичному спрямуванню; – легке знаходження ресурсу в мережі; – швидкість доступу до ресурсу; – цілодобова робота сайту (цілодобове адміністрування); – справність завантаження та відтворення сторінок сайту
2	Інформаційне наповнення	<ul style="list-style-type: none"> – відповідність наданої інформації інтересам цільової аудиторії та призначенню сайту; – відсутність граматичних та стилістичних помилок; – повнота й достовірність поданої інформації; – своєчасне періодичне оновлення даних (актуальність); – наявність додаткової інформації
3	Навігація	<ul style="list-style-type: none"> – логічна структура організації сайту; – правильність посилань та їх відповідність змісту; – зручність та простота користування навігацією (відсутність незрозумілої термінології, ясність призначення сторінок тощо)
4	Дизайн	<ul style="list-style-type: none"> – якість графіки та її завантаження; – доречність та прийнятність стилю і кольорів, їх відповідність маркетинговим цілям підприємства; – зручність макету сторінок, ефективне використання простору сторінки; – легкість сприйняття інтерфейсу та тексту (читабельність)

Продовження таблиці 1.2

5	Інтерактивність	<ul style="list-style-type: none"> – наявність контактної інформації, різні варіанти зв'язку; – наявність додаткових сервісів (форуми, чати, конференції, гостьові книги, опитування тощо); – допомога користувачам (віртуальні менеджери / продавці, дані та рекомендації щодо вибору товару); – швидкість реагування на запити клієнтів (прийом замовлень, відповіді на можливі запитання); – різні форми електронної комерції (варіанти оплати товару, його доставки)
---	-----------------	---

Зазначені критерії є універсальними для всіх галузей економіки, проте ступінь впливу кожного з них на ефективність електронного ресурсу для різних форм бізнесу та галузей економіки будуть різними. Відмінності Інтернету від інших носіїв комунікації у ряді випадків унеможливають застосування існуючих методів для оцінювання просування в електронному середовищі. Визначення залежності та впливу критеріїв і показників ефективності електронного ресурсу на досягнення комунікаційних цілей підприємства стало основою для розроблення нового методу, який враховує специфіку Інтернету та властиві йому засоби просування, дає можливість зосередитися на вузьких аспектах ефективності електронного ресурсу підприємства і підприємницької діяльності загалом. Інтеграція такого методу з уже існуючими дасть змогу використовувати його для більш якісного і реалістичного оцінювання ефективності просування.

Детально розглянемо основні задачі, щодо впровадження WEB-технологій в реалізації маркетинг-аналізу компанії

1) у межах розроблення методу оцінювання ефективності маркетингових комунікацій на основі Інтернет-технологій існує наявність взаємозв'язку між кількістю комунікацій, здійснюваних через електронний ресурс, та прибутком підприємства. Економічний ефект від застосування електронного ресурсу буде тим більшим, чим більше відвідувачів він матиме.

Для підприємства сутність такої залежності полягає у нарощуванні доходів за рахунок покращення видимості електронного ресурсу в мережі Інтернет, підвищення його конкурентоспроможності та залучення нових відвідувачів.

Важливою складовою оцінювання ефективності електронного ресурсу є розрахунок його вебметричного рангу (WR) – рейтингу на основі статистичних даних пошукових систем. Це обумовлено тим, що більшість користувачів (90–95 %) потрапляють на ресурси підприємств через пошукові системи, тому від того, на якому місці у результатах цільових запитів перебуває електронний ресурс, залежить кількість споживачів, які його відвідають.

Обчислення WR за даними найбільш популярних пошукових систем в українському секторі мережі описано формулою:

$$WR = P_S S_{norm}^i + P_V V_{norm}^i, \quad (1.1)$$

де P_S , P_V – ймовірність потрапляння на електронний ресурс через i -ту пошукову систему або посилання на сторінках i -го ресурсу,

S – індикатор розміру електронного ресурсу,

I – індикатор видимості електронного ресурсу на сторінках інших ресурсів у мережі Інтернет.

Оскільки кількість відвідувачів і ймовірність потрапляння їх на електронний ресурс підприємства на декількох пошукових системах буде різною, до уваги доцільно брати ті пошукові системи, які користуються найбільшою популярністю у користувачів. Результати обчислень показників популярності 11 відомих в Україні пошукових систем засвідчили, що найбільші значення інтегрального показника відвідуваності мають лише дві з них – Google (0,725) та Яндекс (0,174). Тому подальші розрахунки вебметричних рангів здійснено на основі статистичних даних цих ресурсів.

Для детального дослідження популярності електронного ресурсу

компанії нами було розроблено парсинг контенту. Парсер контенту - це не що інше, як скрипт, здатний сортувати інформацію, виділяючи найважливішу і обробляючи її згідно з алгоритмом, створеному для вирішення того чи іншого завдання. Це програма для проведення візуального або програмно-автоматизованого синтаксичного і лексичного аналізу сайту.

Необхідним є розробка економіко-математичної моделі ефективності електронного ресурсу як носія маркетингових комунікацій підприємств в Інтернеті, використання якої допоможе у прийнятті конкретних управлінських рішень щодо підвищення конкурентоспроможності ресурсу, ефективності просування в Інтернеті та успішної реалізації електронного бізнесу в майбутньому.

Процедуру побудови моделі ефективності веб-ресурсу підприємства подано на рисунку 1.3.

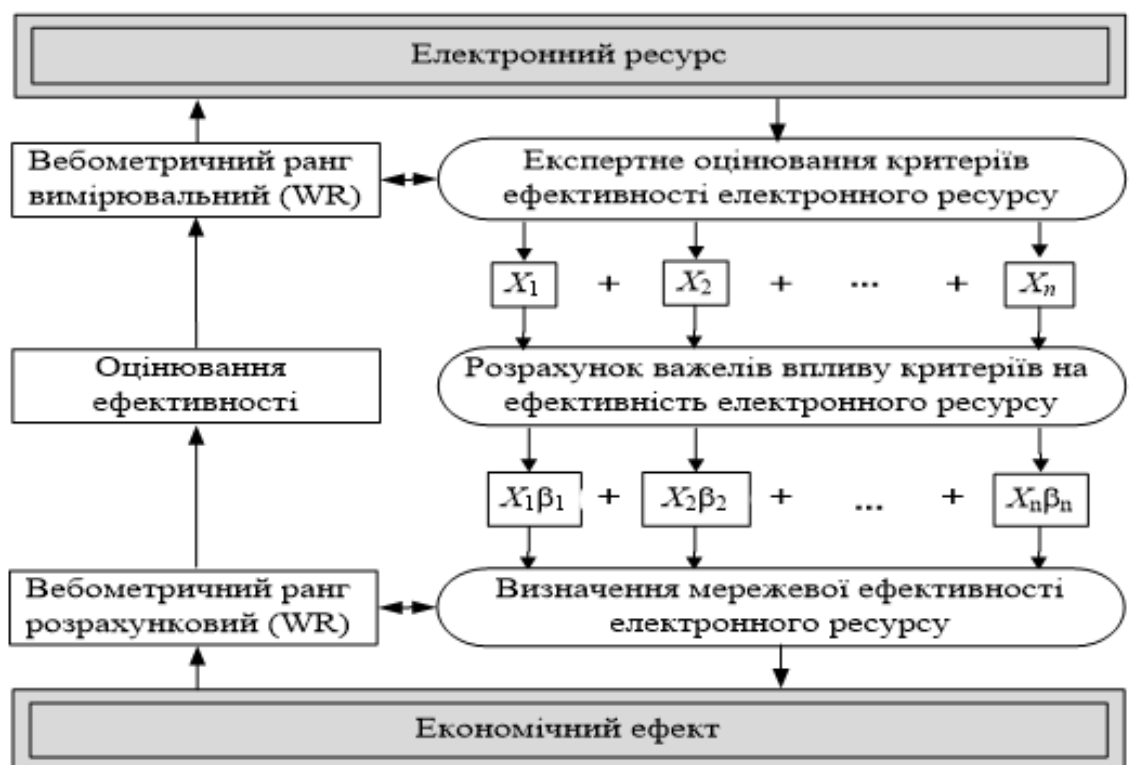


Рисунок 1.3 – Процедура побудови моделі ефективності електронного ресурсу

Для розрахунку залежності та впливу критеріїв і показників ефективності електронного ресурсу на досягнення комунікаційних цілей підприємства в мережі Інтернет запропоновану модель можна подати у вигляді:

$$WR = \beta_0 + \sum_{j=1}^N \beta_j \sum_{i_j=1}^{n_1} \frac{X_{ij}}{m_j X_j^{max}}; \quad (1.2)$$

$$WR = \frac{1}{N} (\beta_0 + \sum_{j=1}^N \beta_j \sum_{i_j=1}^{n_1} \frac{X_{ij}}{m_j X_j^{max}}), \quad (1.3)$$

де N – кількість груп показників,

X_{ij} – оцінка i -го показника j -ої групи,

β_j – вплив показника j -ої групи,

X_j^{max} – діапазон бальної шкали показників j -ої групи,

m_j – кількість показників j -ої групи.

Кількісне обчислення критеріїв здійснюється на основі розрахунку непараметричних показників зв'язку за допомогою експертних оцінок, у результаті чого модель набуває такого вигляду:

$$P_S S_{norm}^i + P_V V_{norm}^i = \frac{1}{N} (\beta_0 + \sum_{j=1}^N \beta_j \sum_{i_j=1}^{n_1} \frac{X_{ij}}{m_j X_j^{max}}). \quad (1.4)$$

Для апробації моделі було досліджено ресурси підприємства. З них було відібрано 56, які мали подібні цілі та форми реалізації підприємницької діяльності в Інтернеті.

2) основним етапом в розв'язку поставленої задачі є її постановка. Під час постановки завдання програміст повинен визначитися з предметною областю та необхідними у роботі вхідними даними, побудувати математичну модель задачі, вирішити основні проблеми проектування.

В нашому випадку нам необхідно розробити систему статистичного аналізу даних продажів маркетплейсу. Для початку нам необхідно розібратися з вхідними даними. Наша система повинна вміщувати в собі повну інформацію про кожний продаж товарів маркетплейсу, а саме:

- назва товару;
- актуальна ціна;
- категорія товару.

Значення наступних полів будуть згенеровані за допомогою псевдовипадкових чисел:

- дата продажу;
- покупець;
- область продажу;
- кількість;
- рейтинг;
- загальна сума.

Розібравшись зі вхідними даними нам потрібно визначитись, які функції виконуватиме наша система. Так як нам необхідно розробити програму аналізу продажів. Тобто функції будуть наступними:

- парсинг інформації з провідних веб-маркетплейсів;
- збереження даних у файл;
- зчитування та відображення даних користувачеві;
- можливість перегляду статистики за критеріями;
- зручний інтерактивний інтерфейс.

Парсери інформації з веб-ресурсів буде найбільш ефективно розробити на мові програмування Python з використанням фреймворку Scrapy.

Визначивши основні функції перейдемо до вирішення проблеми зберігання вихідних даних. Через те, що проект планується розділити на декілька частин, створених на різних мовах програмування, буде доцільно

використовувати спільних формат файлів для передачі інформації. Таким форматом є JSON.

Для реалізації віконного інтерфейсу, нам потрібно вибрати мову програмування. Використано нативну Windows технологію розробки віконних додатків WPF (Windows Presentation Foundation). Тож для проекту обрано мови розробки C# та XAML.

Система також дає можливість отримувати графіки продажів товарів на підставі наданої користувачем інформації.

Це програмне забезпечення призначене для перегляду статистики успішності підприємств.

2 ОСНОВНІ WEB-ТЕХНОЛОГІЇ, ЩО ВИКОРИСТАНО В РЕАЛІЗАЦІЇ МАРКЕТИНГ-АНАЛІЗУ КОМПАНІЇ

2.1 Оцінювання ефективності маркетингових комунікацій підприємств та розрахунок вебметричного рангу

Важливою складовою оцінювання ефективності електронного ресурсу є розрахунок його вебметричного рангу (WR) – рейтингу на основі статистичних даних пошукових систем.

Вибір цього критерію обумовлено тим, що більшість користувачів (90-95%) потрапляють на ресурси через пошукові системи та реферальні посилання, тому від того, на якому місці у результатах цільових запитів по заданій категорії перебуває електронний ресурс, залежить кількість споживачів, які його відвідають.

Обчислення WR за даними найбільш популярних пошукових систем, тобто Yandex та Google в мережі Інтернет описано формулою:

$$WR = P_n S_{norm} + P_i V_{norm}, \quad (2.1)$$

де P_n , P_i – ймовірність потрапляння на електронний ресурс через i -ту пошукову систему або реферальні посилання на сторінках i -го ресурсу,

S – індикатор розміру електронного ресурсу,

V – індикатор видимості електронного ресурсу на сторінках інших ресурсів у мережі Інтернет.

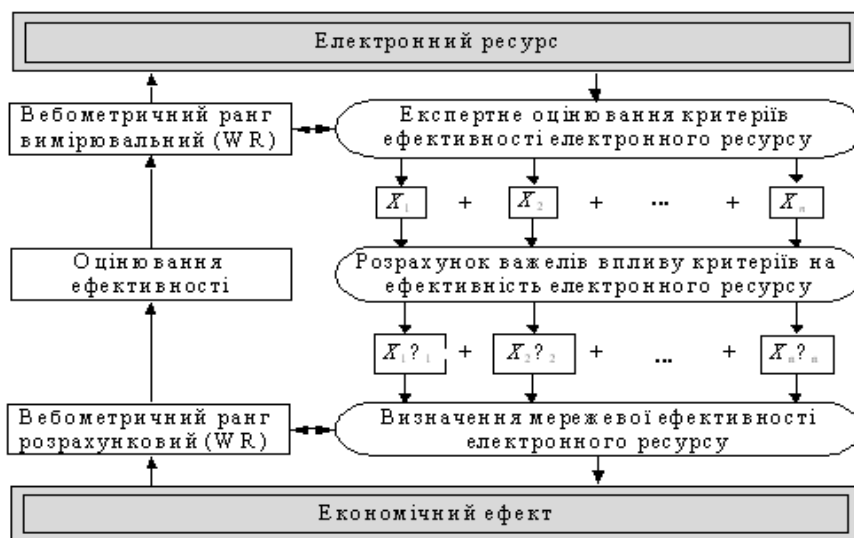
Оскільки кількість відвідувачів і ймовірність потрапляння їх на електронний ресурс підприємства на декількох пошукових системах буде різною, до уваги доцільно брати ті пошукові системи, які користуються найбільшою популярністю у користувачів.

Результати обчислень показників популярності 5 відомих в Україні пошукових систем, а саме Yandex, Google, DuckDuckGo, Bing, Rambler, засвідчили, що найбільші значення інтегрального показника відвідуваності з мають лише дві з них - Google (0,725) та Яндекс (0,174). Тому подальші розрахунки вебметричних рангів здійснено на основі статистичних даних цих ресурсів.

2.2 Структура економіко-математичної моделі

Економіко-математична модель ефективності електронного ресурсу – відображення маркетингових комунікацій підприємства в Інтернеті, мета якого виявити математичні залежності з метою отримання оптимального підходу у прийнятті конкретних управлінських рішень щодо підвищення конкурентоспроможності ресурсу, ефективності просування в Інтернеті та успішної реалізації електронного бізнесу в майбутньому.

Запропонована процедура побудови моделі ефективності веб-ресурсу підприємства схематично подана на рисунку 2.1.



Рисунк 2.1 – Процедура побудови моделі ефективності електронного ресурсу

Для розрахунку залежності та впливу критеріїв і показників ефективності електронного ресурсу на досягнення комунікаційних цілей підприємства в мережі Інтернет утверджену модель можна подати у вигляді математичної залежності:

$$WR = \frac{1}{N} (\beta_0 + \frac{\beta_j(X_i+\dots+X_m)}{m_j X_j^{max}} + \dots + \frac{\beta_m(X_i+\dots+X_m)}{m_j X_j^{max}}), \quad (2.2)$$

де N – кількість груп показників,

X_{ij} – оцінка і-го показника j-ої групи,

β_j, β_m – вплив показника j-ої групи,

m_i, m_n – кількість показників j-ої групи.

Кількісне обчислення критеріїв здійснюється на основі розрахунку непараметричних показників зв'язку за допомогою експертних оцінок, у результаті чого модель набуває такого вигляду:

$$P_S S_{norm} + P_V V_{norm} = \frac{1}{N} (\beta_0 + \sum_{j=1}^N \beta_j \sum_{i=1}^{m_i} \frac{X_{ij}}{m_j X_j^{max}}), \quad (2.3)$$

де N – кількість груп показників,

X_{ij} – оцінка і-го показника j-ої групи,

β_j – вплив показника j-ої групи,

m_i, m_j – кількість показників j-ої групи.

2.3 Апробація економіко-математичної моделі

Для апробації моделі було досліджено 30 інтернет ресурсів маркетплейсингового напрямку, в основі яких є моделі В2В(Бізнес для бізнесу) та В2С (Бізнес для споживача).

Серед претендентів було відібрано 7 інтернет ресурсів, які мали подібні цілі та форми реалізації підприємницької діяльності в Інтернеті.

Відбір сайтів для порівняння було здійснено шляхом аналізу актуальних на 2019р. рейтингів популярності на веб-сторінках 5 спеціалізованих галузевих каталогів і порталів, які містять розділи типу “ТОП 10 лучших интернет магазинов Украины 2018-2019 года” та “Top Websites Ranking/E-commerce And Shopping >E-commerce And Shopping in Ukraine” в яких дана статистика відвідувань конкретного інтернет ресурсу та значення вебметричного рангу відповідної сфери діяльності.

В таблиці 2.1 представлений загальний рейтинг відвідувань інтернет ресурсів станом на 2019р згідно з даними Інтернет ресурсу[7].

Таблиця 2.1 – Рейтинг відвідувань інтернет ресурсів

№	Назва веб-ресурсу	Загальна кількість відвідувань, млн
1	rozetka.com.ua	44,73
2	allo.ua	9,01
3	moyo.ua	2,57
4	brain.com.ua	0,787
5	mobilluck.com.ua	0,772
6	repka.ua	0,098
7	foxtrot.ua	0,08

Усі сайти вибірки ранжовано за показником виміряного рангу (WR). На основі цього було обрано лише ті ресурси, які мали найвищий ранг та на які припадає не менше 80 % відвідувань у галузевому розрізі маркетплейсингу.

Статистичні дані кожного інтернет-підприємства були отримані при самостійному порівнянні готової аналітики зі спеціалізованих веб-ресурсів, що призначені для аналізу ринку, серед яких американський сайт[7], що на

основі даних з різних джерел(реферальні програми, ранжування) генерує глибинну аналітику сайтів.

Результати аналізу вибірки з веб-ресурсу [7] було зведено та представлено у таблиці 2.2

Таблиця 2.2 – Аналітика ранжування веб-ресурсів

№	Назва веб-ресурсу	Глобальний ранг	Ранг по країні	Ранг по категорії	Кількість сторінок за сесію	Середнє перебування, хв	Показник відказів, %	Загальна кількість відвідувань, млн
1	rozetka.com.ua	670	10	54	7.3	6.46	33.51	44.73
2	allo.ua	5652	62	184	4.25	4.21	46.26	9.01
3	moyo.ua	26201	432	53	2.26	2.22	67.84	2.57
4	brain.com.ua	53768	764	472	6.48	8.25	38.94	0.787
5	repka.ua	386995	8077	3760	2.28	1.4	73.23	0.772
6	mobilluck.com.ua	65085	1435	398	3.25	1.47	60.91	0.098
7	foxtrot.ua	926948	19596	11507	1.23	1.2	70.3	0.08

Значення рангу пропорційне рівню відвідувань інтернет ресурсу через різні джерела (прямий перехід на сайт, перехід через пошук та соціальні мережі). Чим менше значення рангу, тим більше сайт є популярним в данній галузевій категорії.

2.4 Програма статистичного аналізу продажів інтернет магазинів

Для побудови раціональної об'єктної моделі даної системи необхідно виконати наступні етапи:

- виокремлення необхідних класів з предметної області;
- визначення залежностей між класами;
- визначення зв'язків між класами;
- побудова діаграми ієрархії визначених класів.

Після аналізу поставленого завдання і предметної області, можна виділити набір необхідних класів. Він буде вміщати в себе як бібліотечні

класи обраної мови програмування, так і користувачькі класи, визначені користувачем. Окрім цих класів, зазвичай створюються так звані «класи реалізації», без котрих не можливе виконання програми (наприклад, клас вікна).

У програмі будуть використані наступні стандартні класи бібліотеки .NET Framework:

- вікно Window
- абстрактний клас-слухач стану NotifyPropertyChanged
- контейнери об'єктів графічного інтерфейсу (StackPanel та Grid)
- кнопки (Button)
- вибір дати (DatePicker)
- текстові блоки (TextBox)
- вкладки (TabItem) та контейнери вкладок (TabControl)
- підписи (Label)
- таблиці (DataGrid)
- серіалізатор JSON (DataContractJsonSerializer)
- файловий потік (FileStream)
- списки (List)
- дія (Action)

Через специфіку розробки з застосуванням технології WPF, були використані класи роботи з графіками з бібліотек Windows Forms.

Для цього у розмітці використовувався клас WindowsFormsHost, котрий містить у собі Chart (з іменного простору «System.Windows.Forms.DataVisualization.Charting»), що доданий до проекту в DLL файлі «System.Windows.Controls.DataVisualization.Toolkit».

Замість примітивного використання подій натискання кнопок, був використаний інтерфейс ICommand, котрий дозволяє використовувати шаблон «Команда». Цей шаблон імплементований у класі DelegateCommand.

Це надає змогу використовувати прив'язування кнопок до полів команд, котрі виконують деякі дії.

У програмі будуть використані наступні самостійно розроблені класи:

- клас `ChartItem` для зберігання для відображення інформації, необхідної у діаграмі;
- клас `JsonFileService` для огортання десериалізації JSON файлу;
- клас `JsonTradeNoteItem` для зберігання інформації про товар у вигляді рядків і інтерфейсом, котрий містять об'єкти вихідного файлу JSON;
- клас `StoreItem` для зберігання списку товарів;
- клас `TradeNoteItem` для зберігання інформації про товар з коректними типами даних;
- клас `DefaultDialogService` для огортання використання діалогу відкриття файлу.

Діаграму створених класів можна переглянути на рисунках 2.3 та 2.4.

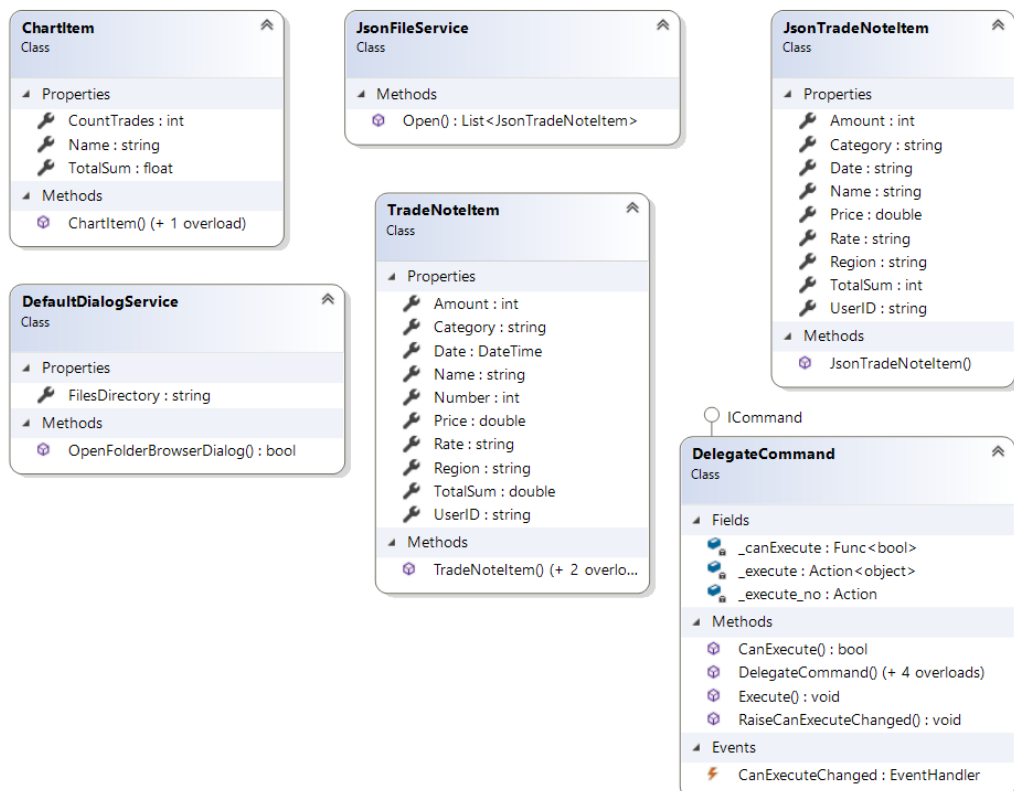


Рисунок 2.3 – Діаграма класів моделі даних та класів реалізації

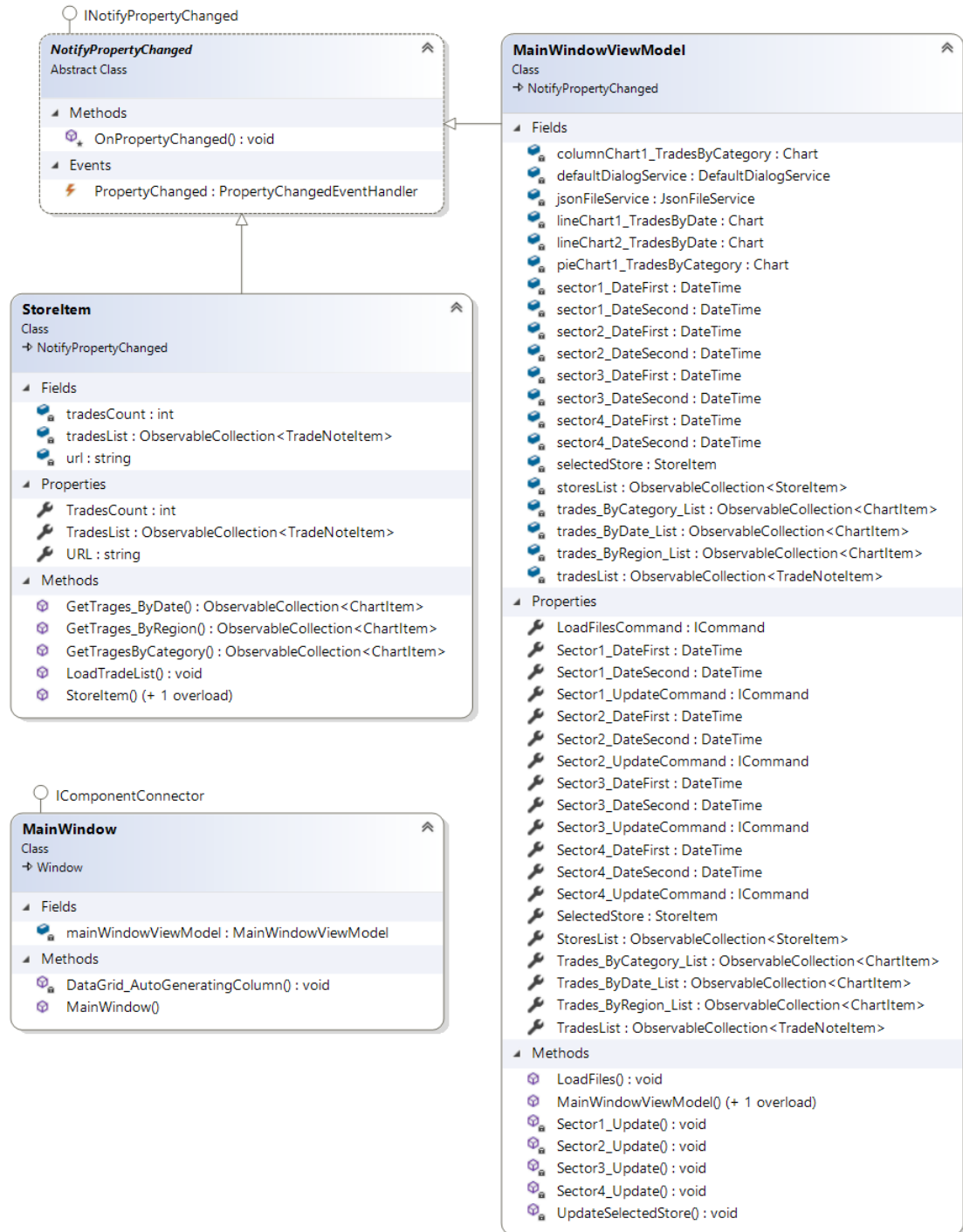


Рисунок 2.4 – Діаграма класів вікна, магазину та моделі відображення

Після того як ми визначилися з цілями існування класів, нам потрібно визначити зв'язки між ними.

Клас `JsonFileService` використовує класи `JsonTradeNoteItem` та `FileStream`.

`DefaultDialogService` використовує `FolderBrowserDialog`, містить у собі `String`.

`MainWindow` містить у собі `MainWindowViewModel`.

`DelegateCommand` імплементує `ICommand`, містить `Func` та 2 об'єкти `Action`.

`StoreItem` містить узагальнений `ObservableCollection` зі значеннями типу `TradeNoteItem`, `List` зі значеннями типу `JsonTradeNoteItem`, `ObservableCollection` зі значеннями `ChartItem`.

`MainWindowViewModel` містить у собі `DefaultDialogService`, `JsonFileService`, `ObservableCollection`, `StoreItem`, `Chart`, `DateTime`, використовує `StoreItem`, `Series`, `ChartArea`, `Legend`.

3 РОЗРОБКА ФУНКЦІОНАЛУ ПРОГРАМИ

3.1 Реалізація програми в середовищі розробки

Після завершення дослідження предметної області, постановки завдання та побудови моделі класів, можна почати програмну реалізацію продукту на мові програмування. Для отримання даних буде використана мова Python[8], вікно з аналітикою буде створене з використанням технології WPF[13], на мовах XAML[14] та C#[12].

Для отримання даних з веб-сторінок, інтерфейс не передбачений. Скрипти(набір інструкцій) Python працюють у режимі командного рядка.

Для парсингу даних напряму з сайтів продажу товарів був використаний фреймворк Scrapy[11], котрий надає можливість швидко та надійно отримувати довільну інформацію з веб-сторінок. Було розроблено користувацький клас-наслідник Spider (див. рис. 3.1), котрий отримує всю необхідну інформацію та віддає її на обробку.

```
regions = [
    'Вінницька',
    *Текст опущений*,
    'Чернігівська']
shops = ['rozetka.ua', 'moyo.ua', 'allo.ua',
        'brain.com.ua', 'foxtrot.ua', 'repka.ua']
class ProductInfoSpider(scrapy.Spider):
    name = 'prod_info'
    start_urls = [
        'https://ek.ua/list/149/pr-2480/',
        *Текст опущений*,
        'https://ek.ua/list/163/']
    allowed_domains = ['ek.ua']
    def start_requests(self):
        for url in self.start_urls:
            yield scrapy.Request(url, self.parse)
    def getSafeValue(self, value):
        return value if not (value is None) else None
    def parse(self, response):
        urls = self.getSafeValue(response.xpath(
            '//td/a[@class="model-short-title no-u"]/@href').extract())
        names = self.getSafeValue(response.xpath(
            '//td/a[@class="model-short-title no-u"]/@title').extract())
```

```

list_of_prices = self.getSafeValue(response.xpath(
    '//div[@class="model-price-range"]/a/span/text()').extract())
category = response.xpath(
    '//div[@class="catalog-path s-width ff-roboto"]/a/text()').extract()
    d1 = date(2016, 1, 13)
    d2 = date(2019, 12, 13)
    delta = d2 - d1
    price_shift = 0
for i in range(len(urls)-1) :
    try:
        item = {}
        item['url'] = 'https://ek.ua' + urls[i]
        item['name'] = names[i]
        item['price_low'] = list_of_prices[i +
            price_shift].replace('\xa0', '')
        item['price_top'] = list_of_prices[i + 1 +
            price_shift].replace('\xa0', '')
        item['price'] = item['price_low']
        item['category'] = category
        item['amount'] = random.randint(1, 10)
        item['date'] = str(
            d1 + timedelta(random.randint(1, delta.days)))
        item['region'] = regions[random.randint(0,
            len(regions) - 1)]
        item['shop'] = random.choice(shops)
        price_shift += 1
        yield item
    except IndexError as ie:
        self.logger.info(ie)

```

Рисунок 3.1– Код створеного Spider для Scrapy

Для тимчасового зберігання об'єктів та серіалізації їх списку у файл, був використаний клас-розширення (див. рис. 3.2) для фреймворку. Він підключений до події отримання нового об'єкту зі сторінки та додає його у список усіх об'єктів. В кінці роботи ProductInfoSpider, розширення серіалізує список у файл.

```

class ItemListExtension(object):
    items = []
    def __init__(self):
        pass
    @classmethod
    def from_crawler(cls, crawler):
        ext = cls()
        crawler.signals.connect(
            ext.spider_closed, signal=signals.spider_closed)
        crawler.signals.connect(
            ext.item_scraped, signal=signals.item_scraped)
        • return ext
    def spider_closed(self, spider) :
        spider.logger.info('Spider closed: %s', spider.name)
        j = str(json.dumps(self.items))
        with open('parsed_items.json', 'w') as output:

```

```

        output.write(j)
    def item_scraped(self, item, response, spider) :
        self.items.append(item)

```

Рисунок 3.2 – Розширення для збереження значень з парсингу

Окрім цього, був розроблений парсер локально збереженого HTML коду сторінок веб-ресурсу `mobilluck.ua` (див. рис. 3.3).

```

html_files = [x for x in os.listdir('./Saved pages') if('.html' in x)]
output_trades_list = []

regions = [
    'Вінницька',
    '*Текст опущений*',
    'Чернігівська']
shops = ['mobilluck.com.ua', 'rozetka.ua', 'moyo.ua', 'allo.ua',
         'brain.com.ua', 'foxtrot.ua', 'repka.ua']

d1 = date(2016, 1, 13)
d2 = date(2019, 12, 13)
delta = d2 - d1

for shop in shops:
    for file in html_files:
        with open('./Saved pages/' + file) as f:
            html_lines = [l.strip() for l in f.readlines() \
                if '<p class="cci2_newprice" style="color: #000">' in l \
                or '<p class="cci2_md1">' in l \
                or '<a itemprop="url" href="https://www.mobilluck.com.ua/katalog/'
in l]

            catalog_tags = [c for c in html_lines if '<a itemprop="url"
href="https://www.mobilluck.com.ua/katalog/' in c]

            c_tag = catalog_tags[0]
            current_category = c_tag[c_tag.find(
                "<span itemprop=\"title\">") + 23 : c_tag.find("</span>")]
            for i in range(2, len(html_lines) - 1, 2) :
                new_trade_note = {}
                new_trade_note['Date'] = str(d1 + timedelta(random.randint(1,
delta.days)))
                new_trade_note['UserID'] = random.randint(0, 10000)
                name_tag = html_lines[i]
                new_trade_note['Name'] = name_tag[name_tag.rfind("\>") + \
                    2:name_tag.find("</a></p>")]
                new_trade_note['Category'] = current_category
                price_tag = html_lines[i + 1]
                new_trade_note['Price'] = float (price_tag[price_tag.find(">") + 1 :
                    price_tag.find("грн")].replace(' ', ''))
                new_trade_note['Amount'] = random.randint(1, 10)
                new_trade_note['TotalSum'] = new_trade_note['Amount'] *
new_trade_note['Price']

                new_trade_note['Region'] = regions[random.randint(0, len(regions) - 1)]
                new_trade_note['Rate'] = random.randint(0, 5)
                output_trades_list.append(new_trade_note)
                random.shuffle(output_trades_list)

```

```
with open('./Output data/' + shop + ".json", "w") as write_file:
    json.dump(output_trades_list, write_file)
```

Рисунок 3.3 – Код парсеру локальних файлів розмітки mobilluck.ua

Отримані дані формуються у список і серіалізуються в формат JSON[10]. Це один з популярних форматів збереження інформації у розробці програм. Він надає можливість розробнику передавати об'єкти між різними мовами програмування, що значно спрощує використання різних мов для різних задач проекту. Приклад записаної у файли інформації наведений на рисунку 3.4. З нього видно, що файл являє собою список об'єктів з полями "Date", "UserID", "Name", "Category", "Price", "Amount", "TotalSum", "Region", "Rate".

```
[
  {
    "Date": "2019-04-26",
    "UserID": 7593,
    "Name": "Смартфон Samsung Galaxy J3 2017 Gold (SM-J330FZDD)",
    "Category": "Телефоны и гаджеты",
    "Price": 4739.0,
    "Amount": 1,
    "TotalSum": 4739.0,
    "Region": "Житомирська",
    "Rate": 3
  },
  *Дані опущені*,
  {
    "Date": "2016-12-22",
    "UserID": 4501,
    "Name": "Сетевой шуруповерт Элпром ЭШС-860",
    "Category": "Все для дачи и ремонту",
    "Price": 466.0,
    "Amount": 1,
    "TotalSum": 466.0,
    "Region": "Вінницька",
    "Rate": 5
  }
]
```

Рисунок 3.4 – Приклад JSON файлу з інформацією

Програма повинна бути функціонально повноцінною і одночасно простою у використанні. Для цього графічний інтерфейс можна спроектувати, використовуючи загальновідомі методики проектування зручних користувацьких інтерфейсів.

Для розмітки інтерфейсу (див. рис. 3.5) була використана мова XAML. Також, для більшості елементів були створені власні стилі відображення.

```

<Window>
  <Grid>
    <Grid>
      <StackPanel>
        <Button/>
        <ListBox>
          <ListBox.ItemTemplate>
            <DataTemplate>
              *Текст опущений*
            </DataTemplate>
          </ListBox.ItemTemplate>
        </ListBox>
      </StackPanel>
    </Grid>
    <Grid>
      <TabControl>
        <TabItem>
          <TabItem.Content>
            <Grid>
              <TabControl>
                <TabItem>
                  <TabItem.Content>
                    <Grid>
                      <StackPanel>
                        <DatePicker/>
                        <Label/>
                        <DatePicker/>
                        <Button/>
                      </StackPanel>
                      <StackPanel>
                        <Grid>
                          <Label/>
                          <WindowsFormsHost>
                            <DVC:Chart/>
                          </WindowsFormsHost>
                          <DataGrid/>
                        </Grid>
                      </StackPanel>
                    </Grid>
                  </TabItem.Content>
                </TabItem>
              </TabControl>
            </Grid>
          </TabItem>
          *Текст опущений*
        </TabItem>
      </TabControl>
    </Grid>
  </Grid>
</Window>

```

Рисунок 3.5 – Приклад розмітки головного вікна, атрибути опущені

Після створення розмітки головного вікна була створена ViewModel. Це клас, необхідний для реалізації шаблону проектування MVVM (Model-

View-ViewModel)[19]. Клас `MainWindow` у цій ієрархії виступає в ролі відображення (`View`), `MainWindowViewModel` виступає у ролі моделі відображення (`ViewModel`), а класи `StoreItem` та `TradeNoteItem` відповідно моделі даних (`Model`).

Однак, пов'язування існуючих даних з їх відображенням користувачеві, не проста задача. Класи `ViewModel` зазвичай найбільш трудомісткі у розробці та складні у пошуку помилок (див. рис. 3.6).

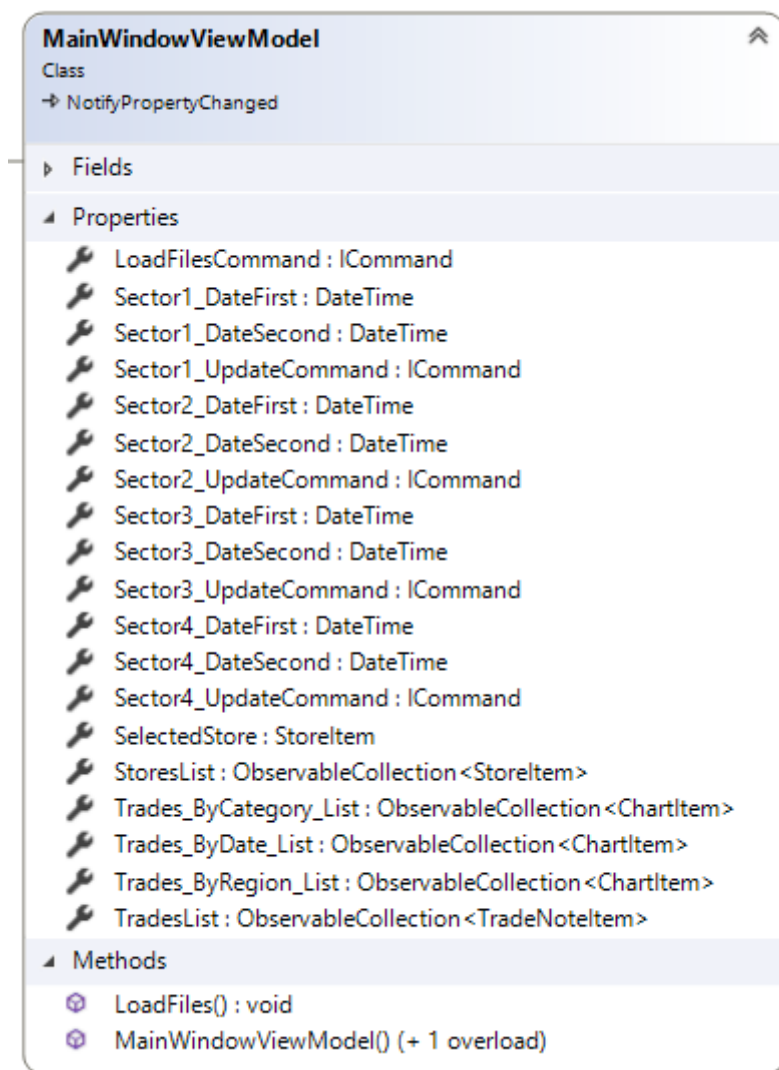


Рисунок 3.6 – Інтерфейсна частина класу `MainWindowViewModel`

Також, особливістю використання паттерну `MVVM` є використання прив'язування (див. рис. 3.7) відображення безпосередньо до змінних їх

контексту даних. Таким чином, програміст отримує можливість передачі змінених у моделі відображення даних, або передачі списків з даними, використовуючи лише одну строку коду. Це значно покращує швидкість роботи програми та її подальше тестування.

```
<DataGridTextColumn Header="Місяць" Binding="{Binding Path=Name}"/>
<DataGridTextColumn Header="Кількість продаж" Binding="{Binding Path=CountTrades}"/>
<DataGridTextColumn Header="Загальна вартість" Binding="{Binding Path=TotalSum}"/>
```

Рисунок 3.7 – Приклад коду прив'язки значень елементів вікна

Також треба отримати вихідні дані, сформовані парсером. Для цього користувач повинен натиснути на кнопку «Завантажити дані». У програмі будуть виконані наступні дії:

- 1) у MainWindowViewModel спрацює команда LoadFilesCommand, що викличе метод LoadFiles поточного об'єкту;
- 2) буде створено новий об'єкт класу StoreItem та додано до списку StoresList;
- 3) за допомогою методу Open(класу JsonFileService) з переданим до нього шляхом файлу з даними сформується список типу JsonTradeNoteItem;
- 4) для StoreItem буде викликано метод LoadTradeList зі списком з попереднього пункту у параметрі. Тепер магазин буде мати інформацію про продажі;
- 5) попередні дії будуть повторені для кожного магазину з обраної папки «.json» файлів.

Потім спрацює «магія» прив'язування списків до елементів інтерфейсу. У моделі відображення є атрибут SelectedStore, котрий буде являти собою обраний користувачем маркетплейс. Як тільки це поле отримає значення, воно буде одразу відображене на екран.

Зразу після цього, користувач побачить весь список продаж маркетплейсу.

Потім можна обрати різноманітні способи отримання аналітичних графіків, котрі формуються з наданої інформації.

Програма має можливість формування графіків наступних типів:

- 1) для обраного магазину:
 - статистика за категоріями;
 - статистика за областями;
 - статистика за місяцями.

- 2) загальна статистика для всіх магазинів.

Для отримання даних статистики, була використана бібліотека фреймворку .NET під назвою LINQ (Language Integrated Query)[17]. Це мова запитів, інтегрована у мову C#. Її синтаксис дуже схожий на мову SQL, однак ці запити працюватимуть значно швидше та спрощують роботу з об'єктами. За способом використання це схоже на роботу з об'єктною моделлю бази даних, але в цій програмі можна обійтись без останньої.

Статистика за категоріями формується за допомогою методу на рисунку 3.8. Результат виконання наведений на рисунку 3.9.

```
public ObservableCollection<ChartItem> GetTragesByCategory(DateTime
dataFirst, DateTime dataSecond)
{
    ObservableCollection<ChartItem> chartItems = new
ObservableCollection<ChartItem> ();

    var WrapByCategory = from trade in TradesList
        where trade.Date >= dataFirst && trade.Date <= dataSecond
        group trade by trade.Category into g
        select new { Name = g.Key, CountTrades = g.Sum(i =>
i.Amount) , TotalSum = g.Sum(i=> i.Amount * i.Price) };

    foreach (var item in WrapByCategory)
    {
        chartItems.Add(new ChartItem(item.Name,
(int)item.CountTrades, (float)item.TotalSum));
    }

    return chartItems;
}
```

Рисунок 3.8 – Код методу GetTragesByCategory класу StoreItem

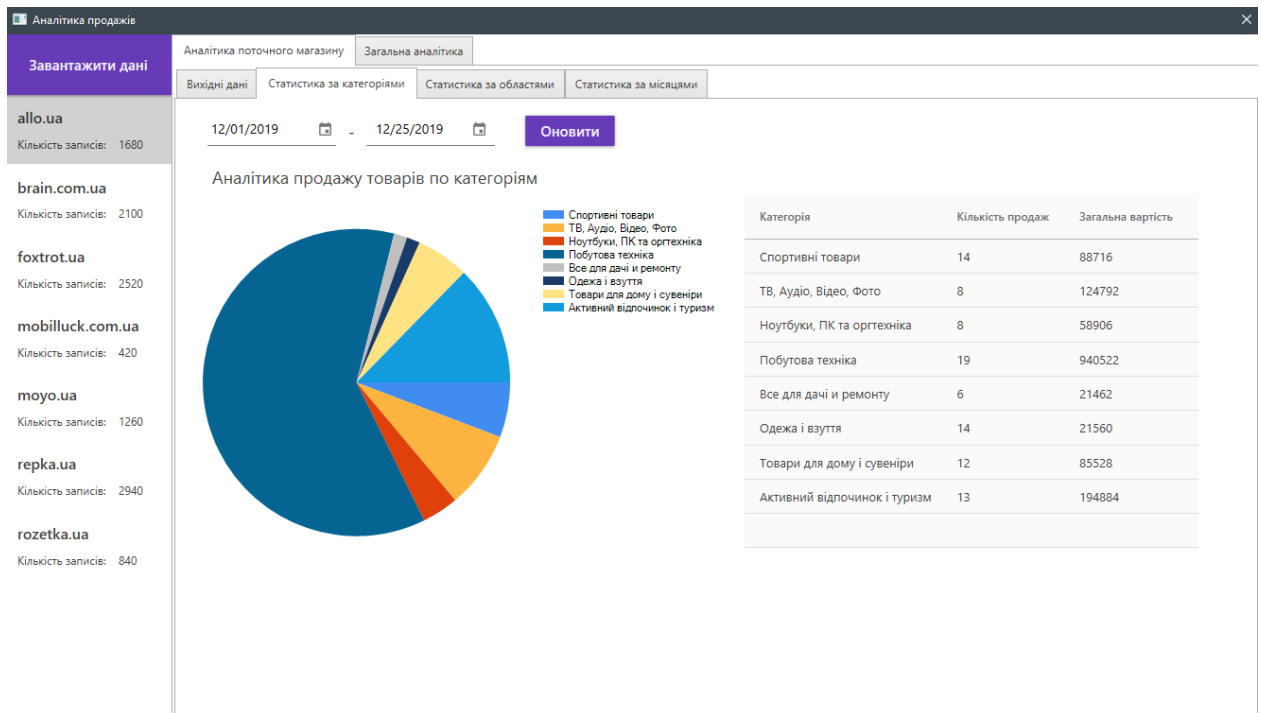


Рисунок 3.9 – Приклад графіку продажів товарів за категоріями

Код графіку товарів за областями наведений на рисунку 3.10. Результат виконання наведений на рисунку 3.11.

```
public ObservableCollection<ChartItem> GetTrages_ByRegion(DateTime dataFirst,
DateTime dataSecond)
{
    ObservableCollection<ChartItem> chartItems = new
ObservableCollection<ChartItem> ();

    var Wrap_ByRegion = from trade in TradesList
        where trade.Date >= dataFirst && trade.Date <= dataSecond
        group trade by trade.Region into g
        select new { Region = g.Key, CountTrades = g.Sum(i =>
i.Amount), TotalSum = g.Sum(i => i.Amount * i.Price) };

    foreach (var item in Wrap_ByRegion)
    {
        chartItems.Add(new ChartItem(item.Region,
(int)item.CountTrades, (float)item.TotalSum));
    }

    return chartItems;
}
```

Рисунок 3.10 – Код методу GetTrages_ByRegion класу StoreItem

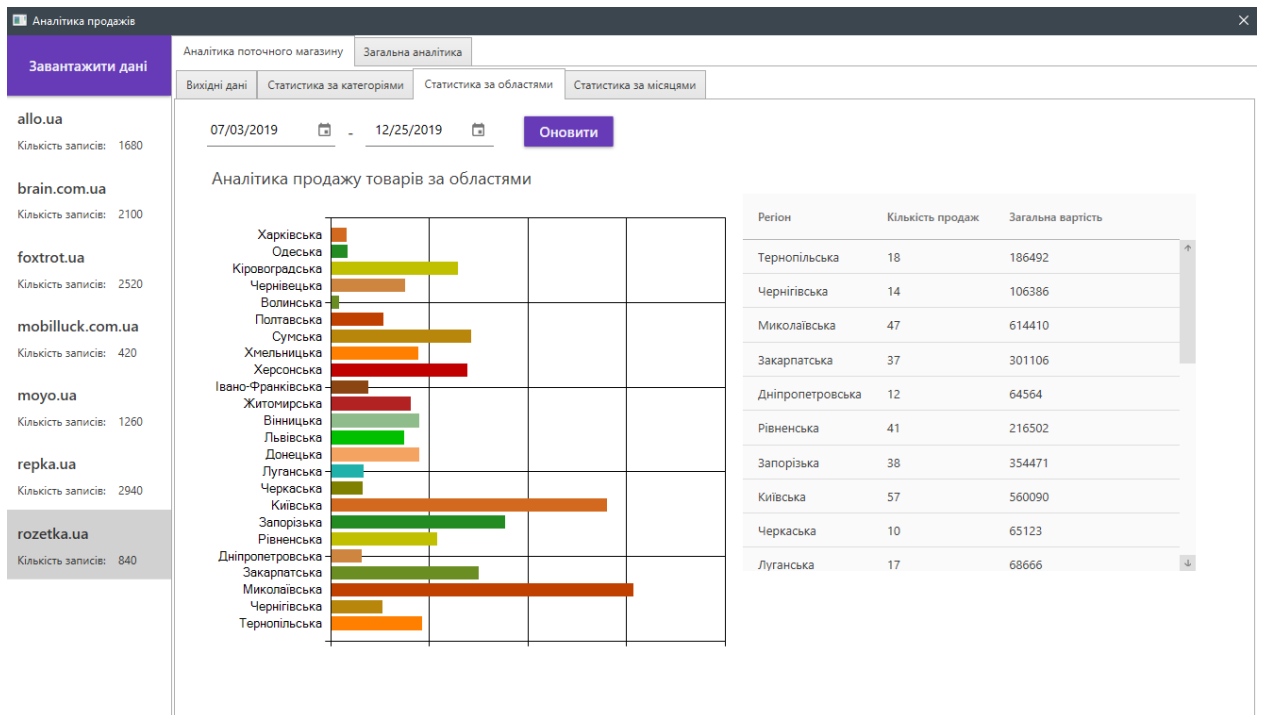


Рисунок 3.11 – Приклад графіку товарів за областями

Код графіку товарів за місяцями наведений на рисунку 3.12. Результат виконання наведений на рисунку 3.13.

```
public ObservableCollection<ChartItem> GetTrages_ByDate(DateTime dataFirst,
DateTime dataSecond)
{
    ObservableCollection<ChartItem> chartItems = new
ObservableCollection<ChartItem>();

    var Wrap_ByDate = from trade in TradesList
        where trade.Date >= dataFirst && trade.Date <= dataSecond
        orderby trade.Date ascending
        group trade by trade.Date.ToString("MM.yyyy") into g
        select new { Date = g.Key, CountTrades = g.Sum(i =>
i.Amount), TotalSum = g.Sum(i => i.Amount * i.Price) };

    foreach (var item in Wrap_ByDate)
    {
        var readable_month = item.Date.ToString().Split(' ')[0];
        chartItems.Add(new ChartItem(readable_month,
(int)item.CountTrades, (float)item.TotalSum));
    }

    return chartItems;
}
```

Рисунок 3.12 – Код методу GetTrages_ByRegion класу StoreItem

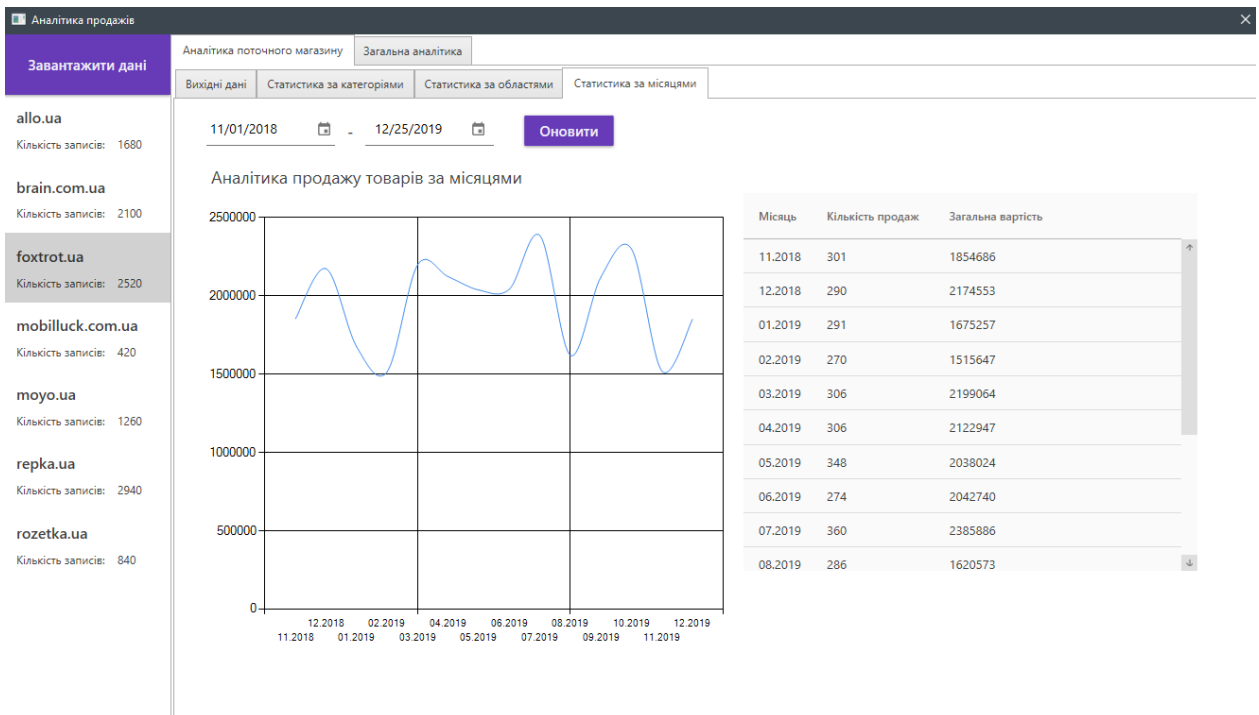


Рисунок 3.13 – Приклад графіку товарів за місяцями

Для формування статистики між різними магазинами, використовується метод `Sector4_Update` у класі моделі-відображення. Приклад коду наведений на рисунку 3.14. Результат виконання наведений на рисунку 3.15.

```
private void Sector4_Update()
{
    if (SelectedStore == null)
        return;

    lineChart2_TradesByDate.Series.Clear();
    foreach (var site in StoresList)
    {
        lineChart2_TradesByDate.Series.Add(new Series(site.URL));

        lineChart2_TradesByDate.Series[site.URL].ChartArea =
            "Default";
        lineChart2_TradesByDate.Series[site.URL].ChartType =
            SeriesChartType.Spline;
        lineChart2_TradesByDate.Series[site.URL].IsValueShownAsLabel
            = false;
        lineChart2_TradesByDate.Series[site.URL]["ColumnLabelStyle"]
            = "Disabled";
        Trades_ByDate_List=site.GetTrages_ByDate(Sector4_DateFirst,
            Sector4_DateSecond);
        foreach (var item in Trades_ByDate_List)
        {
```

```

        lineChart2_TradesByDate.Series[site.URL].Points.AddXY(item.Name,
item.TotalSum);
    }
}

```

Рисунок 3.14 – Код методу Sector4_Update класу MainWindowViewModel

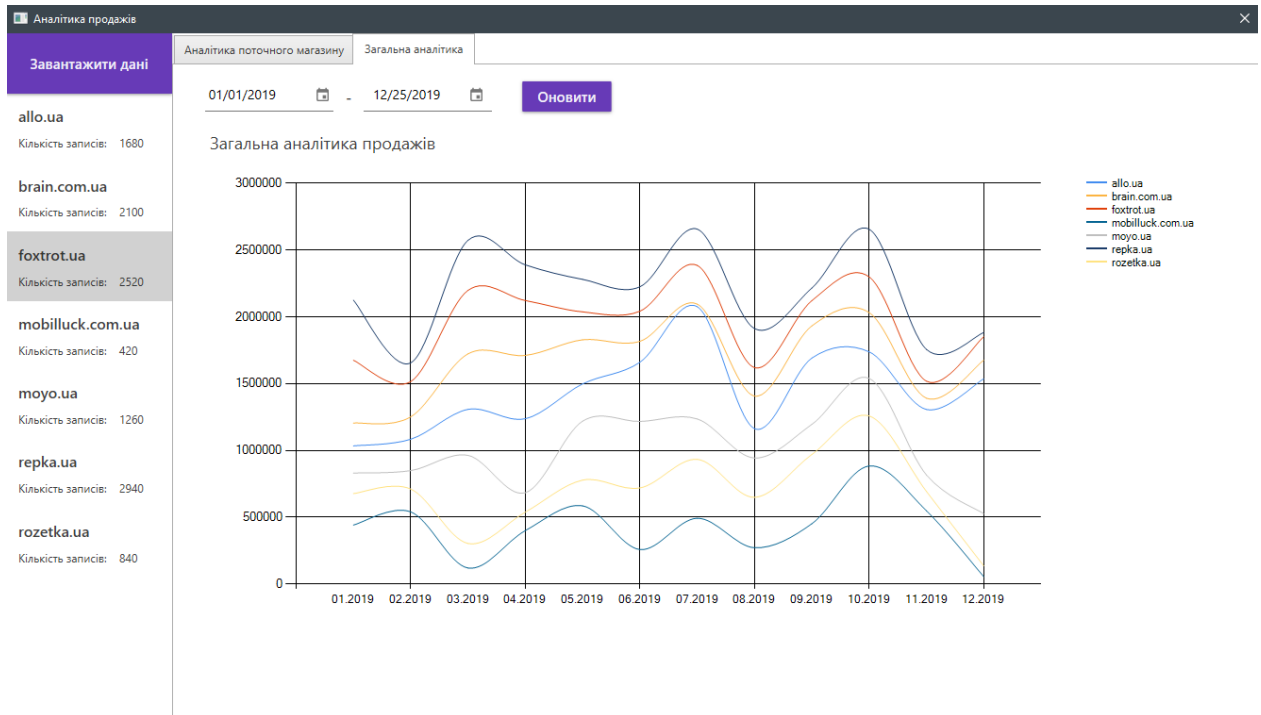


Рисунок 3.15 – Приклад графіку загальної статистики з усіх магазинів

Також, користувач може власноруч обирати період (дата початку та дата кінця) відбору даних, та натискає кнопку «Оновити» (див. рис. 3.16).

Окрім цього, всі таблиці у програмі мають можливість сортування за довільним стовпчиком (див. рис. 3.17).

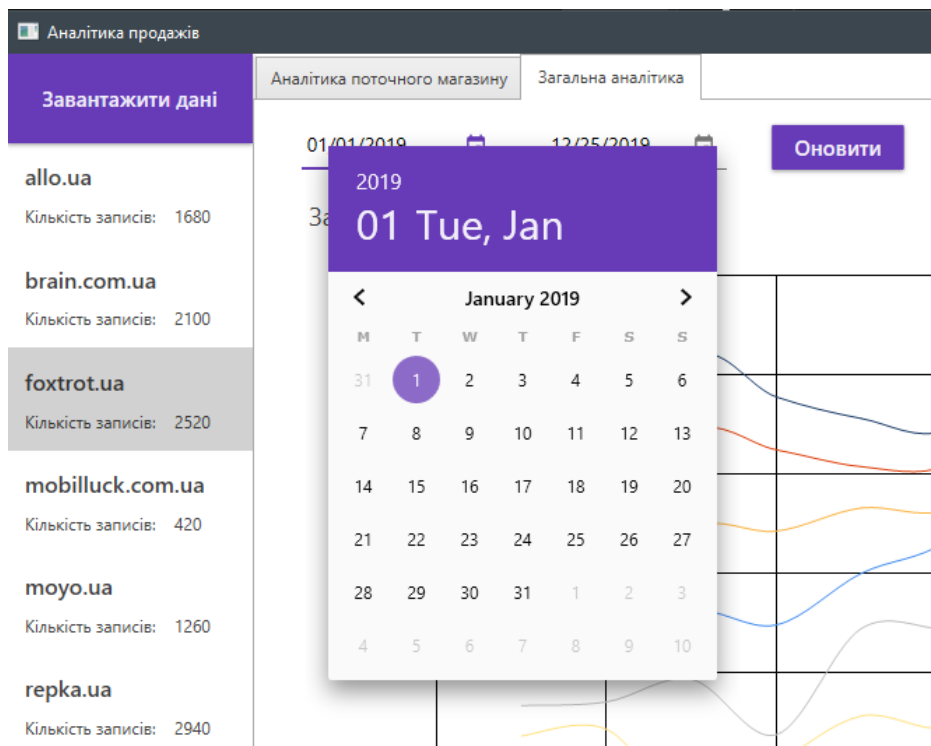


Рисунок 3.16 – Приклад діалогового вікна вибору дати

Вихідні дані	Аналітика поточного магазину							
	Статистика за категоріями	Статистика за областями	Статистика за місяцями					
Номер	Назва	Область	Дата продажу	Покупець	Категорія	Ціна, грн	Загальна сума, грн	Рейтинг
897	Шурупверт аккумуляторный Intertec	Житомирська	1/14/2016 12:00	5365	Все для дачі і ремонту	709	5672	4
1186	Парфюмированная вода Tiziana Terer	Черкаська	1/14/2016 12:00	1184	Красота і здоров'я	2491	12455	3
489	Палатка Kilimanjaro SS-061-091	Київська	1/14/2016 12:00	9065	Активний відпочинок і туризм	1399	12591	0
249	Парфюмированная вода Olfactive Stu	Львівська	1/15/2016 12:00	8149	Красота і здоров'я	1707	11949	1
1497	Джинсы Lee L301BCCN W29/L33 голубі	Донецька	1/15/2016 12:00	7617	Одежа і взуття	3600	14400	2
339	GPS трекер для животных GoGPS Z2 ч	Одеська	1/16/2016 12:00	8577	Товари для дому і сувеніри	2999	20993	5
136	Смартфон Google Pixel 2 64Gb Cleraly	Запорізька	1/16/2016 12:00	3567	Телефони і гаджети	8030	32120	0
308	Шурупверт сетевой Элпром ЭШС-9С	Дніпропетр	1/16/2016 12:00	9542	Все для дачі і ремонту	449	898	1
1349	Смартфон Xiaomi Redmi Note 8 Pro 6/	Житомирська	1/17/2016 12:00	3415	Телефони і гаджети	5999	29995	2
800	Палатка 4-х местная Norfin Malmo 4 П	Дніпропетр	1/18/2016 12:00	1885	Активний відпочинок і туризм	9208	18416	1
438	Палатка High Peak Ascoli 3 Dark grey/л	Донецька	1/20/2016 12:00	9753	Активний відпочинок і туризм	3954	23724	3
593	Палатка Totem Trek	Черкаська	1/20/2016 12:00	3009	Активний відпочинок і туризм	1651	3302	0
1516	Холодильник Liebherr CN 4015	Харківська	1/20/2016 12:00	387	Побутова техніка	16199	145791	4
1120	Детектор валют Dors 50 black	Київська	1/21/2016 12:00	3256	Ноутбуки, ПК та оргтехніка	377	2639	4
1360	Двухкамерный холодильник Atlant XI	Житомирська	1/21/2016 12:00	7227	Побутова техніка	9999	49995	1
1428	Ошейник GPS Muoh MPT-03CB для ко	Миколаївська	1/21/2016 12:00	2099	Товари для дому і сувеніри	1599	3198	5
1028	Джинсы Wrangler W2523556 W28/L34	Волинська	1/21/2016 12:00	880	Одежа і взуття	3000	6000	0

Рисунок 3.17 – Приклад сортування за стовпчиком «Дата продажу»

ВИСНОВКИ

Отримані науково-практичні результати дають можливість зробити такі висновки:

- 1) маркетинг-аналіз є невід'ємною складовою успішної підприємницької діяльності будь-якої компанії. Завдяки широкому вибору інструментарію, що значною мірою зобов'язаний активному розвитку веб-технологій, виникає можливість для проведення глибокої аналітики з використання сучасних методів обробки та аналізу статистичних даних;
- 2) застосування веб-технологій та правильний підхід у виборі оптимального інструменту для реалізації маркетинг-аналізу гарантують швидке отримання змістовних статистичних даних для проведення бізнес-заходів з підвищення ефективності підприємницької діяльності та активності споживачів;
- 3) запропонований спосіб застосування комплексу веб- та інтерактивних технологій реалізує один з напрямків маркетинг-аналізу. В його основі закладений набір кроків зі зчитування та збереження даних з відповідних веб-сторінок, обробки та відображенню користувачеві у графічному вигляді;
- 4) гнучкість використаних технологій та представлений системний підхід з вирішення поставлених задач маркетинг-аналізу компаній дають змогу швидко адаптувати їх для вирішення широкого ряду задач з оцінки якості та прогнозування підприємницької діяльності як в окремих сферах так і в цілому;
- 5) розроблений програмний продукт надає фундаментальні можливості аналізу та відкритий для розширення як функціональних можливостей, так і методів маркетинг-аналізу компаній.

ПЕРЕЛІК ПОСИЛАНЬ

1. Голик В.С. Эффективность интернет-маркетинга в бизнесе. Дикта, 2008. 196 с.
2. Голубков Е.П. Использование Интернета в маркетинге // Маркетинг в России и за рубежом №3 (29), 2002 г. Москва: сайт URL: <https://dis.ru/market/arhiv/2002/3/9.html>
3. Голубков Е. П. Маркетинговые исследования: теория, методология и практика: учебник. 4-е изд., перераб. и доп. / Москва : Финпресс, 2008. 496 с.
4. Басюк О. В. Експертний аналіз інтернет-сайтів турфірм як метод покращення їх економічної діяльності // Науковий вісник Ужгородського національного університету. Серія: Міжнародні економічні відносини та світове господарство. Ужгород : УНУ, 2016. Вип. 7.
5. Длигач А., Писаренко Н. Маркетинговые исследования в разработке стратегий // Маркетинговые исследования в Украине, 2007. № 2. 67 с.
6. Діброва Т.Г. Реалії ринку Інтернет-реклами в Україні / Т.Г. Діброва, М.С. Лебеденко // Маркетинг в Україні, 2007. № 2 (42). 21–24 с.
7. Статистичний аналіз сайтів та їх ранжування: сайт URL: <https://www.similarweb.com>
8. Мова програмування Python 3, сайт URL: <https://pythonworld.ru/>
9. Парсинг сайтів на Python: сайт URL: https://pikabu.ru/story/vsyo_o_parsinge_saytov_na_python_5384166
10. JSON в Python: сайт URL: <https://python-scripts.com/json>
11. Фреймворк Scrapy. Документація: сайт URL: <https://scrapy.org/>
12. Мова програмування C#. Посібник: сайт URL: <https://metanit.com/sharp/tutorial/>
13. Руководство по WPF: сайт URL: <https://metanit.com/sharp/wpf/>
14. WPF та XAML: сайт URL: <https://metanit.com/sharp/wpf/2.php>
15. Работа с JSON C#: сайт URL: <https://metanit.com/sharp/tutorial/6.5.php>

16. Построение графиков в среде WPF: сайт URL: <https://www.c-sharpcorner.com/UploadFile/mahesh/charting-in-wpf/>
17. Язык запросов LINQ: сайт URL: <https://metanit.com/sharp/tutorial/15.1.php>
18. WPF Bindings: URL: <http://kindelephant.blogspot.com/2011/03/binding-wpf.html>
19. Шаблон проектирования MVVM: сайт URL: <https://metanit.com/sharp/wpf/22.1.php>
20. Діброва Т.Г. Особливості розвитку Е-комерції в Україні / Т.Г. Діброва, М.С. Лебеденко // Економіка: проблеми теорії та практики. Вип. 226. Т. 2. Дніпропетровськ : ДНУ, 2007. 321–331 с.
21. Евдокимов Ф. И., Гавва В. М. Азбука маркетинга. Учеб. пособие. Москва : Сталкер, 1998. 432 с.
22. Єжова Л. Ф. Інформаційний маркетинг: Навч. посібник. Київ : КНЕУ, 2002. 560 с.
23. Іванова Л. Особливості маркетингових комунікацій у сфері послуг // Маркетинг в Україні, 2007. №6. 15 с.
24. Ілляшенко С. М. Маркетингові дослідження: Навч. посіб. Для студ. вищ. навч. закл. / С. М. Ілляшенко, М. Ю. Баскакова: Сумський держ. ун-т. Київ : Центр навчальної літератури, 2006. 192 с.
25. Інформаційні системи і технології в економіці: Навч. посібник. / За ред. Пономаренко В. С. Київ : Академія, 2002. 542 с.
26. Карповская-Скорик Е. Е. Маркетинговые прогнозы на уровне предприятия. О : Астропринт, 2000. 112 с.
27. Кракович Д. Новые методики в маркетинговых исследованиях // Маркетинговые исследования в Украине, 2007. № 3. 33 с.
28. Крылова Г. Д., Соколова М. И. Маркетинг. Теория и 86 ситуаций: Учеб пособие для студ. вузов, обуч. по эконом. спец. Москва : ЮНИТИ, 1999. 520 с.

29. Курс выживания интернет-компаний/К.Барроу. Пер. с англ. Москва : Альпина Паблишер, 2001. 341 с.
30. Лебеденко М.С. Интернет-магазины в українській дійсності // Вісник Хмельницького національного університету. Хмельницький, ХНУ, 2007. № 5. Т. 3 (100). 244–248 с.
31. Діброва Т. Г. Е-бізнес для друкованих ЗМІ // Формування ринкової економіки: зб. наук. праць. Київ, КНЕУ, 2008. Вип. 20. 199–208 с.
32. Лебеденко М. С. Вебметричний ранг як показник ефективності електронного ресурсу підприємства // Економічний вісник Національного технічного університету України "Київський політехнічний інститут", 2014. № 11. 401-408 с.
33. Лебеденко М.С. Маркетингові комунікації в мережі Інтернет // Економічний вісник НТУУ «КПІ». Київ, ВПІ ВПК «Політехніка», 2008. № 5. 332–338 с.
34. Лебеденко М.С. Стратегії маркетингових комунікацій в Інтернеті // Вісник Національного університету «Львівська політехніка». Львів, 2009. № 633. 394–402 с.
35. Лебеденко М.С. Стратегічне планування маркетингової діяльності в мережі Інтернет // Економічний вісник НТУУ «КПІ». К.: ВПІ ВПК «Політехніка», 2009. № 6. 285–289 с.
36. Лебеденко М.С. Особливості розвитку поліграфічного ринку України // Економічний простір. Дніпропетровськ, 2009. № 30. 77–88 с.
37. Мамыкин А.А. Стратегия и тактика маркетинга в Интернет // Маркетинг в России и за рубежом, 2000. № 3 - 50 с.
38. Манн И. Маркетинг на 100%. 2-е изд. Спб.: Питер, 2004. 320 с.:ил. (Серия «Деловой бестселлер»)
39. Михайлова Е.А. Проблемы и перспективы взаиморазвития Интернета и международного маркетинга // Маркетинг в России и за рубежом, 1999. № 6. 23 с.

40. Мозкова Г. В. Регулювання інформаційного середовища маркетингу: Автореф. Дис.. канд. екон. наук: 08. 02. 03. / Харківський національний ун-т ім. В. Н. Каразіна. Харків : 2006. 20 с.
41. Новикова І. В. Маркетинг сфери послуг: Навч. посібник. — Київ : Вид-во Європ. ун-ту, 2004. 72 с.
42. Норіцина Н. І. Маркетингова політика комунікацій: Курс лекцій / Міжрегіональна академія управління персоналом (МАУП). Київ : МАУП, 2003. 120 с.
43. МЕТОДЫ ПАРСИНГА САЙТОВ: сайт URL: <http://ponka.vnukov.ru/content/metody-parsinga-saytov>
44. Вирин Ф. Интернет-маркетинг. Полный сборник практических инструментов / Федор Вирин. Москва : Эскмо, 2009. 224 с.
45. Петрик Е.А. Интернет-маркетинг / Е.А. Петрик. Москва : Московская финансово-промышленная академия, 2004. 299 с.
46. Успенский И.В. Интернет-маркетинг: Учебник / И.В. Успенский. СПб.: Изд-во СПГУЭиФ, 2003 : сайт URL: www.aup.ru/books/m80/
47. Семеняк И. В. Стратегический маркетинг : учеб. пособ. / И. В. Семеняк. Х. : ХНУ имени В. Н. Каразина, 2007. 304 с.
48. Новомлинский Л. Интернет-стратегии каждый выбирает по себе / Л. Новомлинский. Информационно-консалтинговый центр по электронному бизнесу: сайт URL: <http://www.e-commerce.ru/analytics/analytics-part/analytics13.html>
49. Павленко А. Ф. Маркетинг: Підручник / А. Ф. Павленко, А. В. Войчак. Київ : КНЕУ, 2003. 246 с.
50. Павленко А. Ф. Маркетингові комунікації: сучасна теорія і практика / А. Ф. Павленко, А. В. Войчак, Т. О. Примак: Київський національний економічний ун-т ім. Вадима Гетьмана. Київ : КНЕУ, 2005. 408 с.
51. Панкрухин А.П. Маркетинг в компьютерных сетях// Маркетинг в России и за рубежом, 1999. № 4. 35 с.

52. Петров В. Н. Информационные системы. Спб.: Питер : 2002. 688 с.
53. Пінчук Н. С., Галузинський Г. П., Орленко Н. С. Інформаційні системи і технології в маркетингу: Навч. посібник. Київ : КНЕУ, 1999. 328 с.
54. Каймін В. А., Інформатика. Підручник для студентів. 6-е видання. Москва : ИНФРА-М, 1998–2009. 300 с.

ДОДАТОК А

Код програми

Файл ItemListExtension.py

```
class ItemListExtension(object):
    items = []

    def __init__(self):
        pass

    @classmethod
    def from_crawler(cls, crawler):
        ext = cls()
        crawler.signals.connect(
            ext.spider_closed, signal=signals.spider_closed)
        crawler.signals.connect(
            ext.item_scraped, signal=signals.item_scraped)
        return ext

    def spider_closed(self, spider):
        spider.logger.info('Spider closed: %s', spider.name)
        j = str(json.dumps(self.items))
        with open('parsed_items.json', 'w') as output:
            output.write(j)

    def item_scraped(self, item, response, spider):
        self.items.append(item)
```

Файл ProductInfoSpider.py

```
import scrapy
import settings as settings
import json
from items.ProductItem import ProductItem
import codecs
import random

from datetime import date, timedelta
```

```
regions = [  
    'Вінницька',  
    'Волинська',  
    'Дніпропетровська',  
    'Донецька',  
    'Житомирська',  
    'Закарпатська',  
    'Запорізька',  
    'Івано-Франківська',  
    'Київська',  
    'Кіровоградська',  
    'Луганська',  
    'Львівська',  
    'Миколаївська',  
    'Одеська',  
    'Полтавська',  
    'Рівненська',  
    'Сумська',  
    'Тернопільська',  
    'Харківська',  
    'Херсонська',  
    'Хмельницька',  
    'Черкаська',  
    'Чернівецька',  
    'Чернігівська'  
]  
  
shops = ['rozetka.ua', 'moyo.ua', 'allo.ua',  
        'brain.com.ua', 'foxtrot.ua', 'repka.ua']  
  
class ProductInfoSpider(scrapy.Spider):  
    name = 'prod_info'  
  
    start_urls = [  
        'https://ek.ua/list/149/pr-2480/',  
        'https://ek.ua/list/284/',  
        'https://ek.ua/list/217/pr-5525/',  
        'https://ek.ua/list/26/',  
        'https://ek.ua/list/41/',  
        'https://ek.ua/list/76/',  
        'https://ek.ua/list/365/',  
        'https://ek.ua/list/879/',  
        'https://ek.ua/list/190/',
```

```
'https://ek.ua/list/187/',
'https://ek.ua/list/239/pr-2374/',
'https://ek.ua/list/238/',
'https://ek.ua/list/127/',
'https://ek.ua/list/1136/',
'https://ek.ua/ek-list.php?katalog_=74&list_view_type_=3',
'https://ek.ua/list/163/']
```

```
allowed_domains = ['ek.ua']
```

```
def start_requests(self):
    for url in self.start_urls:
        yield scrapy.Request(url, self.parse)
```

```
def getSafeValue(self, value):
    return value if not(value is None) else None
```

```
def parse(self, response):
```

```
    urls = self.getSafeValue(response.xpath(
        '//td/a[@class="model-short-title no-u"]/@href').extract())
```

```
    names = self.getSafeValue(response.xpath(
        '//td/a[@class="model-short-title no-u"]/@title').extract())
```

```
    list_of_prices = self.getSafeValue(response.xpath(
        '//div[@class="model-price-range"]/a/span/text()').extract())
```

```
    category = response.xpath(
        '//div[@class="catalog-path s-width ff-roboto"]/a/text()').extract()
```

```
d1 = date(2016, 1, 13)
d2 = date(2019, 12, 13)
delta = d2 - d1
```

```
price_shift = 0
for i in range(len(urls)-1):
    try:
        item = {}
        item['url'] = 'https://ek.ua' + urls[i]
        item['name'] = names[i]

        item['price_low'] = list_of_prices[i +
```

```

        price_shift].replace('\xa0',
                               ")
    item['price_top'] = list_of_prices[i + 1 +
        price_shift].replace('\xa0',
                               ")

    item['price'] = item['price_low']
    item['category'] = category
    item['amount'] = random.randint(1, 10)
    item['date'] = str(
        d1 + timedelta(random.randint(1, delta.days)))
    item['region'] = regions[random.randint(0, len(regions)-1)]
    item['shop'] = random.choice(shops)
    price_shift += 1
    yield item
except IndexError as ie:
    self.logger.info(ie)

```

Файл Entry.py

```

import scrapy
from scrapy.crawler import CrawlerProcess
from scrapy.utils.project import get_project_settings

import logging
import json
import unicodedata
import codecs
import unidecode

LOG_FORMAT = ('%(levelname) -10s %(asctime)s %(name) -30s %(funcName) '
              '-35s %(lineno) -5d: %(message)s')
logger = logging.getLogger(__name__)

process = CrawlerProcess(get_project_settings())

process.crawl('prod_info')

process.start()

```

Файл ProxiesMiddleware.py


```

import settings as settings
import logging

LOG_FORMAT = ('%(levelname) -10s %(asctime)s %(name) -30s %(funcName) '
              '-35s %(lineno) -5d: %(message)s')
logger = logging.getLogger(__name__)

class ProxiesMiddleware(object):
    def __init__(self, settings):
        pass

    @classmethod
    def from_crawler(cls, crawler):
        return cls(crawler.settings)

    def process_request(self, request, spider):
        if settings.USE_PROXY_MID:
            logger.info("http://%s:%s" % (
                settings.PROXY_IP_OR_URL, settings.PROXY_PORT))
            request.meta["proxy"] = "http://%s:%s" % (
                settings.PROXY_IP_OR_URL, settings.PROXY_PORT)

```

Файл My parser.py

```

import os
import json
import random

from datetime import date, timedelta
from decimal import Decimal

html_files = [x for x in os.listdir('./Saved pages') if('.html' in x)]

output_trades_list = []

regions = [
    'Вінницька',
    'Волинська',
    'Дніпропетровська',
    'Донецька',

```

```

'Житомирська',
'Закарпатська',
'Запорізька',
'Івано-Франківська',
'Київська',
'Кіровоградська',
'Луганська',
'Львівська',
'Миколаївська',
'Одеська',
'Полтавська',
'Рівненська',
'Сумська',
'Тернопільська',
'Харківська',
'Херсонська',
'Хмельницька',
'Черкаська',
'Чернівецька',
'Чернігівська'
]

shops = ['mobilluck.com.ua', 'rozetka.ua', 'moyo.ua', 'allo.ua',
         'brain.com.ua', 'foxtrot.ua', 'repka.ua']

#Діапазон дат
d1 = date(2016, 1, 13)
d2 = date(2019, 12, 13)
delta = d2 - d1

for shop in shops:
    for file in html_files:
        with open('./Saved pages/' + file) as f:
            html_lines = [l.strip() for l in f.readlines() \
                          if '<p class="cci2_newprice" style="color: #000">' in l \
                          or '<p class="cci2_md1">' in l \
                          or '<a itemprop="url" href="https://www.mobilluck.com.ua/katalog/' in
            ]

            catalog_tags = [c for c in html_lines if '<a itemprop="url"
            href="https://www.mobilluck.com.ua/katalog/' in c]

            #Нужная строка с категорией

```

```

c_tag = catalog_tags[0]

current_category = c_tag[c_tag.find("<span itemprop=\"title\">") + 23 :
c_tag.find("</span>")]

for i in range(2, len(html_lines) - 1, 2):

    new_trade_note = {}

    #Дата продажи
    new_trade_note['Date'] = str(d1 + timedelta(random.randint(1,
delta.days)))

    #Генерируем ID покупателя
    new_trade_note['UserID'] = random.randint(0, 10000)

    #Нужная строка с названием товара
    name_tag = html_lines[i]
    new_trade_note['Name'] = name_tag[name_tag.rfind(">") +
2:name_tag.find("</a></p>")]

    #Категория
    new_trade_note['Category'] = current_category

    #Нужная строка с ценой
    price_tag = html_lines[i + 1]
    new_trade_note['Price'] = float(price_tag[price_tag.find(">") + 1 :
price_tag.find("грн")].replace(' ', ''))

    #Кол-во товара
    new_trade_note['Amount'] = random.randint(1, 10)

    #Кол-во товара
    new_trade_note['TotalSum'] = new_trade_note['Amount'] *
new_trade_note['Price']

    #Регион
    new_trade_note['Region'] = regions[random.randint(0, len(regions)-1)]

    #Рейтинг товара
    new_trade_note['Rate'] = random.randint(0, 5)

    output_trades_list.append(new_trade_note)

```

```
random.shuffle(output_trades_list)
```

```
with open('./Output data/' + shop + ".json", "w") as write_file:
    json.dump(output_trades_list, write_file)
```

Файл MainWindow.xaml.cs

```
using System.Windows;
using TradeAnalytics.ViewModels;

namespace TradeAnalytics
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        MainWindowViewModel mainWindowViewModel;

        public MainWindow()
        {
            InitializeComponent();

            mainWindowViewModel=new
            MainWindowViewModel(PieChart1_TradesByCategory,
            ColumnChart1_TradesByRegion, LineChart1_TradesByDate,
            LineChart2_TradesByDate);

            DataContext=mainWindowViewModel;
        }

        private void DataGrid_AutoGeneratingColumn(object sender,
        System.Windows.Controls.DataGridAutoGeneratingColumnEventArgs e)
        {
            if (e.PropertyType == typeof(System.DateTime))
            {
                e.Column.ClipboardContentBinding.StringFormat="d";
            }
        }
    }
}
```

Файл StoreItem.cs

```
public class StoreItem : NotifyPropertyChanged
{
    #region TradesList
    private ObservableCollection<TradeNoteItem> tradesList = new
ObservableCollection<TradeNoteItem>();

    public ObservableCollection<TradeNoteItem> TradesList
    {
        set { tradesList = value; }
        get { return tradesList; }
    }

    #endregion

    /// <summary>
    /// Адрес сайта
    /// </summary>
    private string url;

    public string URL
    {
        get { return url; }
        set
        {
            url = value;
            OnPropertyChanged("URL");
        }
    }

    /// <summary>
    /// Количество продаж
    /// </summary>
    private int tradesCount;

    public int TradesCount
    {
        get { return tradesCount; }
        set
        {
            tradesCount = value;
            OnPropertyChanged("TradesCount");
        }
    }
}
```

```

    }
}

public StoreItem(string url)
{
    this.url = url;
}

public StoreItem()
{
    this.url = "None";
}

public void LoadTradeList(List<JsonTradeNoteItem> jsonTradeNoteItems)
{
    TradesCount = jsonTradeNoteItems.Count;

    int number = 1;

    foreach (var tradeNoteItem in jsonTradeNoteItems)
    {
        tradesList.Add(new TradeNoteItem(number, tradeNoteItem));
        number++;
    }
}

//Получить группировку по категориям
public ObservableCollection<ChartItem> GetTragesByCategory(DateTime
dataFirst, DateTime dataSecond)
{
    ObservableCollection<ChartItem> chartItems = new
ObservableCollection<ChartItem>();

    var WrapByCategory = from trade in TradesList
        where trade.Date >= dataFirst && trade.Date <= dataSecond
        group trade by trade.Category into g
        select new { Name = g.Key, CountTrades = g.Sum(i =>
i.Amount) , TotalSum = g.Sum(i=> i.Amount * i.Price) };

    foreach (var item in WrapByCategory)
    {
        chartItems.Add(new ChartItem(item.Name, (int)item.CountTrades,
(float)item.TotalSum));
    }
}

```

```

    }

    return chartItems;
}

//Получить группировку по региону
public ObservableCollection<ChartItem> GetTrages_ByRegion(DateTime
dataFirst, DateTime dataSecond)
{
    ObservableCollection<ChartItem> chartItems = new
ObservableCollection<ChartItem>();

    var Wrap_ByRegion = from trade in TradesList
        where trade.Date >= dataFirst && trade.Date <= dataSecond
        group trade by trade.Region into g
        select new { Region = g.Key, CountTrades = g.Sum(i =>
i.Amount), TotalSum = g.Sum(i => i.Amount * i.Price) };

    foreach (var item in Wrap_ByRegion)
    {
        chartItems.Add(new ChartItem(item.Region, (int)item.CountTrades,
(float)item.TotalSum));
    }

    return chartItems;
}

//Получить группировку по дате
public ObservableCollection<ChartItem> GetTrages_ByDate(DateTime
dataFirst, DateTime dataSecond)
{
    ObservableCollection<ChartItem> chartItems = new
ObservableCollection<ChartItem>();

    var Wrap_ByDate = from trade in TradesList
        where trade.Date >= dataFirst && trade.Date <= dataSecond
        orderby trade.Date ascending
        group trade by trade.Date.ToString("MM.yyyy") into g
        select new { Date = g.Key, CountTrades = g.Sum(i =>
i.Amount), TotalSum = g.Sum(i => i.Amount * i.Price) };

    foreach (var item in Wrap_ByDate)
    {

```

```

        var readable_month = item.Date.ToString().Split(' ')[0];
        chartItems.Add(new ChartItem(readable_month, (int)item.CountTrades,
(float)item.TotalSum));
    }

    return chartItems;
}

```

Файл TradeNoteItem.cs

```

using System;
using TradeAnalytics.JsonItems;

namespace TradeAnalytics.Models
{
    public class TradeNoteItem
    {
        //Номер в списке
        public int Number { get; set; }

        //Дата заказа
        public DateTime Date { get; set; }

        //ID покупателя
        public string UserID { get; set; }

        //Наименование товара
        public string Name { get; set; }

        //Категория товара
        public string Category { get; set; }

        //Стоимость товара
        public double Price { get; set; }

        //Количество
        public int Amount { get; set; }

        //Общая сумма
        public double TotalSum { get; set; }

        //Регион потребителя
        public string Region { get; set; }
    }
}

```



```
//Рейтинг товара
```

```
public string Rate { get; set; }
```

```
public TradeNoteItem()
```

```
{
```

```
}
```

```
public TradeNoteItem(int number, JsonTradeNoteItem tradeNoteItem)
```

```
{
```

```
    Number = number;
```

```
    Date = DateTime.Parse(tradeNoteItem.Date);
```

```
    UserID = tradeNoteItem.UserID;
```

```
    Name = tradeNoteItem.Name;
```

```
    Category = tradeNoteItem.Category;
```

```
    Amount = (int)tradeNoteItem.Amount;
```

```
    Price = (double)tradeNoteItem.Price;
```

```
    TotalSum = (double)tradeNoteItem.TotalSum;
```

```
    Region = tradeNoteItem.Region;
```

```
    Rate = tradeNoteItem.Rate;
```

```
}
```

```
public TradeNoteItem(int number, DateTime date, string name, string
category, int amount, double price, string region)
```

```
{
```

```
    Number = number;
```

```
    Date = date;
```

```
    Name = name;
```

```
    Category = category;
```

```
    Amount = amount;
```

```
    Price = price;
```

```
    Region = region;
```

```
}
```

```
}
```

```
}
```

```
Файл DefaultDialogService.cs
```

```
public class DefaultDialogService
```

```
{
```

```
    public string FilesDirectory { get; set; }
```

```
    public bool OpenFolderBrowserDialog()
```

```

{
    FolderBrowserDialog folderBrowserDialog = new FolderBrowserDialog();

    if(folderBrowserDialog.ShowDialog() == DialogResult.OK)
    {
        FilesDirectory = folderBrowserDialog.SelectedPath;
        return true;
    }

    return false;
}
}

```

Файл JsonFileService.cs

```

public class JsonFileService
{
    public List<JsonTradeNoteItem> Open(string filename)
    {
        List<JsonTradeNoteItem> tradeNoteItems = new
List<JsonTradeNoteItem>();
        DataContractJsonSerializer jsonFormatter = new
DataContractJsonSerializer(typeof(List<JsonTradeNoteItem>));
        using (FileStream fs = new FileStream(filename, FileMode.Open))
        {
            tradeNoteItems = jsonFormatter.ReadObject(fs) as
List<JsonTradeNoteItem>;
        }

        return tradeNoteItems;
    }
}

```

Файл JsonTradeNoteItem.cs

```

public class JsonTradeNoteItem
{
    //Дата заказа
    public string Date { get; set; }

    //ID покупателя

```

```

public string UserID { get; set; }

//Наименование товара
public string Name { get; set; }

//Категория товара
public string Category { get; set; }

//Стоимость товара
public double Price { get; set; }

//Количество
public int Amount { get; set; }

//Общая сумма
public int TotalSum{ get; set; }

//Регион потребителя
public string Region { get; set; }

//Рейтинг товара
public string Rate { get; set; }

public JsonTradeNoteItem()
{
}
}

```

Файл NotifyPropertyChanged.cs

```

public abstract class NotifyPropertyChanged : INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    protected virtual void OnPropertyChanged(string propertyName)
    {
        PropertyChanged?.Invoke(this, new
        PropertyChangedEventArgs(propertyName));
    }
}

```

Файл DelegateCommand.cs

```

public class DelegateCommand : ICommand
{
    private readonly Action<object> _execute;
    private readonly Action _execute_no;

    private readonly Func<bool> _canExecute;

    /// <summary>
    /// С ПАРАМЕТРАМИ
    /// </summary>

    public DelegateCommand(Action<object> execute) : this(execute, null)
    {
    }

    public DelegateCommand(Action<object> execute, Func<bool> canExecute)
    {
        if (execute == null)
        {
            throw new ArgumentNullException("execute");
        }

        _execute = execute;
        _canExecute = canExecute;
    }

    /// <summary>
    /// БЕЗ ПАРАМЕТРОВ
    /// </summary>

    public DelegateCommand(Action execute_no) : this(execute_no, null)
    {
    }

    public DelegateCommand(Action execute, Func<bool> canExecute)
    {
        if (execute == null)
        {
            throw new ArgumentNullException("execute");
        }
    }
}

```

```

    _execute_no = execute;
    _canExecute = canExecute;
}

public DelegateCommand()
{
}

/// <summary>
/// Occurs when changes occur that affect whether the command should
execute.
/// </summary>
public event EventHandler CanExecuteChanged;

/// <summary>
/// Raises the <see cref="CanExecuteChanged" /> event.
/// </summary>
public void RaiseCanExecuteChanged()
{
    var handler = CanExecuteChanged;
    if (handler != null)
    {
        handler(this, EventArgs.Empty);
    }
}

/// <summary>
/// Defines the method that determines whether the command can execute in
its current state.
/// </summary>
/// <param name="parameter">This parameter will always be
ignored.</param>
/// <returns>true if this command can be executed; otherwise, false.</returns>
[DebuggerStepThrough]
public bool CanExecute(object parameter)
{
    return _canExecute == null ? true : _canExecute();
}

/// <summary>
/// Defines the method to be called when the command is invoked.
/// </summary>

```

```

    /// <param name="parameter">This parameter will always be
    ignored.</param>
    public void Execute(object parameter)
    {
        if (CanExecute(parameter))
        {
            if (_execute_no == null)
                _execute(parameter);
            else
                _execute_no();
        }
    }
}

```

Файл ChartItem.cs

```

public class ChartItem
{
    public string Name { get; set; }

    public int CountTrades { get; set; }

    public float TotalSum { get; set; }

    public ChartItem()
    {
        CountTrades = 0;
        TotalSum = 0;
    }

    public ChartItem(string name, int countTrades, float totalSum)
    {
        Name = name;
        CountTrades = countTrades;
        TotalSum = totalSum;
    }
}

```

Файл MainWindowViewModel.cs

```

public class MainWindowViewModel : NotifyPropertyChanged
{
    DefaultDialogService defaultDialogService = new DefaultDialogService();
}

```

```
JsonFileService jsonFileService = new JsonFileService();

#region Trades data

private ObservableCollection<TradeNoteItem> tradesList;

public ObservableCollection<TradeNoteItem> TradesList
{
    get { return tradesList; }
    set
    {
        tradesList = value; OnPropertyChanged("TradesList");
    }
}

#endregion

#region Shops data

private ObservableCollection<StoreItem> storesList;

public ObservableCollection<StoreItem> StoresList
{
    get { return storesList; }

}

private StoreItem selectedStore;

public StoreItem SelectedStore
{
    get { return selectedStore; }
    set
    {
        selectedStore = value;
        OnPropertyChanged("SelectedStore");
        UpdateSelectedStore(value);
    }
}

private void UpdateSelectedStore(StoreItem storeItem)
```

```

{
    TradesList = storeItem.TradesList;
}

#endregion

#region Sector1

Chart pieChart1_TradesByCategory;

private DateTime sector1_DateFirst = DateTime.Now;

public DateTime Sector1_DateFirst
{
    get { return sector1_DateFirst; }
    set
    {
        sector1_DateFirst = value;
        OnPropertyChanged("Sector1_DateFirst");
    }
}

private DateTime sector1_DateSecond = DateTime.Now;

public DateTime Sector1_DateSecond
{
    get { return sector1_DateSecond; }
    set
    {
        sector1_DateSecond = value;
        OnPropertyChanged("Sector1_DateSecond");
    }
}

public ICommand Sector1_UpdateCommand
{
    get { return new DelegateCommand(Sector1_Update); }
}

private ObservableCollection<ChartItem> trades_ByCategory_List;

public ObservableCollection<ChartItem> Trades_ByCategory_List
{

```



```

get { return trades_ByCategory_List; }
set
{
    trades_ByCategory_List = value;
    OnPropertyChanged("Trades_ByCategory_List");
}
}

#endregion

#region Sector2

Chart columnChart1_TradesByCategory;

private DateTime sector2_DateFirst = DateTime.Now;

public DateTime Sector2_DateFirst
{
    get { return sector2_DateFirst; }
    set
    {
        sector2_DateFirst = value;
        OnPropertyChanged("Sector2_DateFirst");
    }
}

private DateTime sector2_DateSecond = DateTime.Now;

public DateTime Sector2_DateSecond
{
    get { return sector2_DateSecond; }
    set
    {
        sector1_DateSecond = value;
        OnPropertyChanged("Sector2_DateSecond");
    }
}

public ICommand Sector2_UpdateCommand
{
    get { return new DelegateCommand(Sector2_Update); }
}

```

```

}

private ObservableCollection<ChartItem> trades_ByRegion_List;

public ObservableCollection<ChartItem> Trades_ByRegion_List
{
    get { return trades_ByRegion_List; }
    set
    {
        trades_ByRegion_List = value;
        OnPropertyChanged("Trades_ByRegion_List");
    }
}

}

#endregion

#region Sector3

Chart lineChart1_TradesByDate;

private DateTime sector3_DateFirst = DateTime.Now;

public DateTime Sector3_DateFirst
{
    get { return sector3_DateFirst; }
    set
    {
        sector3_DateFirst = value;
        OnPropertyChanged("Sector3_DateFirst");
    }
}

private DateTime sector3_DateSecond = DateTime.Now;

public DateTime Sector3_DateSecond
{
    get { return sector3_DateSecond; }
    set
    {
        sector3_DateSecond = value;
        OnPropertyChanged("Sector3_DateSecond");
    }
}

```

```

    }
}

public ICommand Sector3_UpdateCommand
{
    get { return new DelegateCommand(Sector3_Update); }
}

private ObservableCollection<ChartItem> trades_ByDate_List;

public ObservableCollection<ChartItem> Trades_ByDate_List
{
    get { return trades_ByDate_List; }
    set
    {
        trades_ByDate_List = value;
        OnPropertyChanged("Trades_ByDate_List");
    }
}

#endregion

#region Sector4

Chart lineChart2_TradesByDate;

private DateTime sector4_DateFirst = DateTime.Now;

public DateTime Sector4_DateFirst
{
    get { return sector4_DateFirst; }
    set
    {
        sector4_DateFirst = value;
        OnPropertyChanged("Sector4_DateFirst");
    }
}

private DateTime sector4_DateSecond = DateTime.Now;

public DateTime Sector4_DateSecond
{
    get { return sector4_DateSecond; }
}

```

```

    set
    {
        sector3_DateSecond = value;
        OnPropertyChanged("Sector4_DateSecond");
    }
}

public ICommand Sector4_UpdateCommand
{
    get { return new DelegateCommand(Sector4_Update); }
}

#endregion

public ICommand LoadFilesCommand
{
    get { return new DelegateCommand(LoadFiles); }
}

public void LoadFiles()
{
    StoresList.Clear();

    if (defaultDialogService.OpenFolderBrowserDialog())
    {
        foreach (var itemPath in
Directory.GetFiles(defaultDialogService.FilesDirectory, "*.json"))
        {
            StoreItem storeItem = new
StoreItem(Path.GetFileNameWithoutExtension(itemPath));

            StoresList.Add(storeItem);

            storeItem.LoadTradeList(jsonFileService.Open(itemPath));

            SelectedStore = StoresList[0];
        }
    }
}

```

```

private void Sector1_Update()
{
    if (SelectedStore == null)
        return;

    //Проверка на null

    pieChart1_TradesByCategory.Series.Clear();

    pieChart1_TradesByCategory.Series.Add(new Series("Series1"));

    pieChart1_TradesByCategory.Series["Series1"].ChartArea = "Default";
    pieChart1_TradesByCategory.Series["Series1"].ChartType =
SeriesChartType.Pie;
    pieChart1_TradesByCategory.Series["Series1"].IsValueShownAsLabel =
true;

    pieChart1_TradesByCategory.Series["Series1"]["PieLabelStyle"] =
"Disabled";

    Trades_ByCategory_List =
SelectedStore.GetTragesByCategory(Sector1_DateFirst, Sector1_DateSecond);

    foreach (var item in Trades_ByCategory_List)
    {

pieChart1_TradesByCategory.Series["Series1"].Points.AddXY(item.Name,
item.TotalSum);
        //chart.Series["Series1"].Points[0].LegendText = "wo";
    }
}

private void Sector2_Update()
{
    if (SelectedStore == null)
        return;

    columnChart1_TradesByCategory.Series.Clear();

    columnChart1_TradesByCategory.Series.Add(new Series("Series1"));

    columnChart1_TradesByCategory.Series["Series1"].Palette =
ChartColorPalette.EarthTones;

```

```

        columnChart1_TradesByCategory.Series["Series1"].ChartArea =
"Default";
        columnChart1_TradesByCategory.Series["Series1"].ChartType =
SeriesChartType.Bar;

columnChart1_TradesByCategory.Series["Series1"].IsValueShownAsLabel =
false;

        columnChart1_TradesByCategory.Series["Series1"]["BarLabelStyle"] =
"Disabled";

        Trades_ByRegion_List =
SelectedStore.GetTrages_ByRegion(Sector2_DateFirst, Sector2_DateSecond);

        foreach (var item in Trades_ByRegion_List)
        {

columnChart1_TradesByCategory.Series["Series1"].Points.AddXY(item.Name,
item.TotalSum);
        }
    }

private void Sector3_Update()
{
    if (SelectedStore == null)
        return;

    lineChart1_TradesByDate.Series.Clear();

    lineChart1_TradesByDate.Series.Add(new Series("Series1"));

    //lineChart1_TradesByDate.Series["Series1"].Palette =
ChartColorPalette.EarthTones;
    lineChart1_TradesByDate.Series["Series1"].ChartArea = "Default";
    lineChart1_TradesByDate.Series["Series1"].ChartType =
SeriesChartType.Spline;
    lineChart1_TradesByDate.Series["Series1"].IsValueShownAsLabel = false;

    lineChart1_TradesByDate.Series["Series1"]["ColumnLabelStyle"] =
"Disabled";

    Trades_ByDate_List =
SelectedStore.GetTrages_ByDate(Sector3_DateFirst, Sector3_DateSecond);

```

```

        foreach (var item in Trades_ByDate_List)
        {
            lineChart1_TradesByDate.Series["Series1"].Points.AddXY(item.Name,
item.TotalSum);
        }
    }

private void Sector4_Update()
{
    if (SelectedStore == null)
        return;

    lineChart2_TradesByDate.Series.Clear();

    foreach (var site in StoresList)
    {
        lineChart2_TradesByDate.Series.Add(new Series(site.URL));

        //lineChart1_TradesByDate.Series["Series1"].Palette =
ChartColorPalette.EarthTones;
        lineChart2_TradesByDate.Series[site.URL].ChartArea = "Default";
        lineChart2_TradesByDate.Series[site.URL].ChartType =
SeriesChartType.Spline;
        lineChart2_TradesByDate.Series[site.URL].IsValueShownAsLabel =
false;

        lineChart2_TradesByDate.Series[site.URL]["ColumnLabelStyle"] =
"Disabled";

        Trades_ByDate_List = site.GetTrages_ByDate(Sector4_DateFirst,
Sector4_DateSecond);

        foreach (var item in Trades_ByDate_List)
        {
            lineChart2_TradesByDate.Series[site.URL].Points.AddXY(item.Name,
item.TotalSum);
        }
    }
}

```

```

public MainWindowViewModel(Chart _pieChart1_TradesByCategory, Chart
_columnChart1_TradesByCategory, Chart _lineChart1_TradesByDate, Chart
_lineChart2_TradesByDate) : this()
{
    pieChart1_TradesByCategory = _pieChart1_TradesByCategory;
    pieChart1_TradesByCategory.ChartAreas.Add(new ChartArea("Default"));

    pieChart1_TradesByCategory.Legends.Add(new Legend("Main"));
    pieChart1_TradesByCategory.Legends["Main"].Docking = Docking.Right;

    columnChart1_TradesByCategory = _columnChart1_TradesByCategory;
    columnChart1_TradesByCategory.ChartAreas.Add(new
ChartArea("Default"));

columnChart1_TradesByCategory.ChartAreas[0].AxisY.LabelStyle.Interval = 1;

columnChart1_TradesByCategory.ChartAreas[0].AxisX.LabelStyle.Interval = 1;

    lineChart1_TradesByDate = _lineChart1_TradesByDate;
    lineChart1_TradesByDate.ChartAreas.Add(new ChartArea("Default"));

    lineChart1_TradesByDate.ChartAreas[0].AxisX.LabelStyle.Interval = 1;

    lineChart2_TradesByDate = _lineChart2_TradesByDate;
    lineChart2_TradesByDate.ChartAreas.Add(new ChartArea("Default"));
    lineChart2_TradesByDate.Legends.Add(new Legend("Main"));
    lineChart2_TradesByDate.Legends["Main"].Docking = Docking.Right;
    lineChart2_TradesByDate.ChartAreas[0].AxisX.LabelStyle.Interval = 1;
}
public MainWindowViewModel()
{
    storesList = new ObservableCollection<StoreItem>();
    tradesList = new ObservableCollection<TradeNoteItem>();

    trades_ByCategory_List = new ObservableCollection<ChartItem>();
    trades_ByRegion_List = new ObservableCollection<ChartItem>();
    trades_ByDate_List = new ObservableCollection<ChartItem>();
}
}

```



```

<Window x:Class="TradeAnalytics.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:wpf="clr-
namespace:MaterialDesignThemes.Wpf;assembly=MaterialDesignThemes.Wpf"
  xmlns:local="clr-namespace:TradeAnalytics"
  mc:Ignorable="d"
  xmlns:wfi="clr-
namespace:System.Windows.Forms.Integration;assembly=WindowsFormsIntegrat
ion"
  xmlns:DVC="clr-
namespace:System.Windows.Forms.DataVisualization.Charting;assembly=System
.Windows.Forms.DataVisualization"
  xmlns:DV="clr-
namespace:System.Windows.Controls.DataVisualization;assembly=System.Windo
ws.Controls.DataVisualization.Toolkit"
  Title="Аналітика продажів"
  WindowStartupLocation="CenterScreen"
  Height="768"
  ResizeMode="NoResize"
  Width="1344">

<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="Auto"/>
    <ColumnDefinition Width="*/>
  </Grid.ColumnDefinitions>

  <Grid Width="176">
    <StackPanel>
      <Button Height="64" Command="{Binding LoadFilesCommand}"
Content="Завантажити дані"/>
      <ListBox x:Name="StoresList" Style="{StaticResource
MaterialDesignListBox}" ItemsSource="{Binding StoresList}"
SelectedItem="{Binding SelectedStore}">
        <ListBox.ItemTemplate>
          <DataTemplate>
            <Grid>
              <StackPanel>
                <Label FontWeight="Medium" FontSize="16"
Content="{Binding URL}"/>

```

```

        <StackPanel Orientation="Horizontal">
            <Label Content="Кількість записів: "/>
            <Label Content="{Binding TradesCount}"/>
        </StackPanel>

    </StackPanel>
</Grid>
</DataTemplate>
</ListBox.ItemTemplate>
</ListBox>
</StackPanel>
</Grid>
<Grid Grid.Column="1">

    <TabControl>
        <!--АНАЛИТИКА ВЫБРАННОГО МАГАЗИНА-->
        <TabItem>
            <TabItem.Header>
                <Label Content="Аналітика поточного магазину"/>
            </TabItem.Header>
            <TabItem.Content>
                <Grid>
                    <!--АНАЛИТИКА ВЫБРАННОГО МАГАЗИНА-->
                    <TabControl>
                        <!--ИСХОДНЫЕ ДАННЫЕ-->
                        <TabItem>
                            <TabItem.Header>
                                <Label Content="Вихідні дані"/>
                            </TabItem.Header>
                            <TabItem.Content>
                                <Grid>
                                    <DataGrid Style="{StaticResource
MaterialDesignDataGrid}" AutoGenerateColumns="False"
ItemsSource="{Binding TradesList, UpdateSourceTrigger=PropertyChanged}"
AutoGeneratingColumn="DataGrid_AutoGeneratingColumn">
                                        <DataGrid.Columns>
                                            <DataGridTextColumn Header="Номер"
Binding="{Binding Path=Number}" />
                                            <DataGridTextColumn Header="Назва"
Binding="{Binding Path=Name}" />
                                            <DataGridTextColumn Header="Область"
Binding="{Binding Path=Region}" />

```

```

                <DataGridTextColumn Header="Дата продажу"
Binding="{ Binding Path=Date}" />
                <DataGridTextColumn Header="Покупець"
Binding="{ Binding Path=UserID}" />
                <DataGridTextColumn Header="Категорія"
Binding="{ Binding Path=Category}" />
                <DataGridTextColumn Header="Ціна, грн"
Binding="{ Binding Path=Price}" />
                <DataGridTextColumn Header="Загальна сума,
грн" Binding="{ Binding Path=TotalSum}" />
                <DataGridTextColumn Header="Рейтинг"
Binding="{ Binding Path=Rate}" />
            </DataGrid.Columns>
        </DataGrid>
    </Grid>
</TabItem.Content>
</TabItem>

<!--ЗА КАТЕГОРИЯМИ-->
    <TabItem>
        <TabItem.Header>
            <Label Content="Статистика за категоріями"/>
        </TabItem.Header>
        <TabItem.Content>
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="64"/>
                    <RowDefinition Height="Auto"/>
                </Grid.RowDefinitions>
                <StackPanel Orientation="Horizontal">
                    <DatePicker SelectedDate="{ Binding
Sector1_DateFirst, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
HorizontalContentAlignment="Center" FontSize="14" Margin="32 0 8 0"
Width="136" Height="32" wpf:TextFieldAssist.Hint="Початок періоду"/>
                    <Label Content="-" FontSize="16"
VerticalAlignment="Center" FontWeight="Medium"/>
                    <DatePicker SelectedDate="{ Binding
Sector1_DateSecond, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
HorizontalContentAlignment="Center" FontSize="14" Margin="8 0 0 0"
Width="136" Height="32" wpf:TextFieldAssist.Hint="Кінець періоду"/>
                    <Button Margin="32 0 0 0" Content="Оновити"
Command="{ Binding Sector1_UpdateCommand}" />
                </StackPanel>
            </Grid>
        </TabItem.Content>
    </TabItem>

```

```

<StackPanel Grid.Row="1">
  <Grid>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="Auto"/>
      <ColumnDefinition Width="*/>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
      <RowDefinition Height="Auto"/>
      <RowDefinition/>
    </Grid.RowDefinitions>
    <Label Margin="32 0 0 0" Content="Аналітика
продажу товарів по категоріям" FontSize="18"/>
    <WindowsFormsHost Grid.Row="1"
Background="Honeydew" Width="600" Height="400">
      <DVC:Chart
x:Name="PieChart1_TradesByCategory" />
    </WindowsFormsHost>
    <DataGrid Width="480"
VerticalAlignment="Top" HorizontalAlignment="Left" Grid.Column="1"
Grid.Row="1" Style="{StaticResource MaterialDesignDataGrid}"
AutoGenerateColumns="False" ItemsSource="{Binding Trades_ByCategory_List,
UpdateSourceTrigger=PropertyChanged}"
AutoGeneratingColumn="DataGrid_AutoGeneratingColumn">
      <DataGrid.Columns>
        <DataGridTextColumn Header="Категорія"
Binding="{Binding Path = Name}" />
        <DataGridTextColumn Header="Кількість
продаж" Binding="{Binding Path = CountTrades}" />
        <DataGridTextColumn Header="Загальна
вартість" Binding="{Binding Path = TotalSum}" />
      </DataGrid.Columns>
    </DataGrid>
  </Grid>
</StackPanel>
</Grid>
</TabItem.Content>
</TabItem>

<!--ЗА ГОРОДАМИ-->
<TabItem>
  <TabItem.Header>
    <Label Content="Статистика за об'лятями"/>
  </TabItem.Header>

```

```

<TabItem.Content>
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition Height="64"/>
      <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>
    <StackPanel Orientation="Horizontal">
      <DatePicker SelectedDate="{ Binding
Sector2_DateFirst, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
HorizontalContentAlignment="Center" FontSize="14" Margin="32 0 8 0"
Width="136" Height="32" wpf:TextFieldAssist.Hint="Початок періоду"/>
      <Label Content="-" FontSize="16"
VerticalAlignment="Center" FontWeight="Medium"/>
      <DatePicker SelectedDate="{ Binding
Sector2_DateSecond, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
HorizontalContentAlignment="Center" FontSize="14" Margin="8 0 0 0"
Width="136" Height="32" wpf:TextFieldAssist.Hint="Кінець періоду"/>
      <Button Margin="32 0 0 0" Content="Оновити"
Command="{ Binding Sector2_UpdateCommand }"/>
    </StackPanel>
    <StackPanel Grid.Row="1">
      <Grid>
        <Grid.ColumnDefinitions>
          <ColumnDefinition Width="Auto"/>
          <ColumnDefinition Width="*/>
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
          <RowDefinition Height="Auto"/>
          <RowDefinition/>
        </Grid.RowDefinitions>

        <Label Margin="32 0 0 0" Content="Аналітика
продажу товарів за об'єктами" FontSize="18"/>

        <WindowsFormsHost Grid.Row="1"
Background="Honeydew" Width="600" Height="500" VerticalAlignment="Top">
          <DVC:Chart
x:Name="ColumnChart1_TradesByRegion" />
        </WindowsFormsHost>

        <DataGrid Width="480" Height="400"
VerticalAlignment="Top" HorizontalAlignment="Left" Grid.Column="1"
Grid.Row="1" Style="{ StaticResource MaterialDesignDataGrid}"

```

```

AutoGenerateColumns="False" ItemsSource="{Binding Trades_ByRegion_List,
UpdateSourceTrigger=PropertyChanged}"
AutoGeneratingColumn="DataGrid_AutoGeneratingColumn">
    <DataGrid.Columns>
        <DataGridTextColumn Header="Region"
Binding="{Binding Path = Name}" />
        <DataGridTextColumn Header="Кількість
продаж" Binding="{Binding Path = CountTrades}" />
        <DataGridTextColumn Header="Загальна
вартість" Binding="{Binding Path = TotalSum}" />
    </DataGrid.Columns>
</DataGrid>
</Grid>
</StackPanel>
</Grid>
</TabItem.Content>
</TabItem>
<!--ПО МЕСЯЧНО-->
<TabItem>
    <TabItem.Header>
        <Label Content="Статистика за місяцями"/>
    </TabItem.Header>
    <TabItem.Content>
        <Grid>
            <Grid.RowDefinitions>
                <RowDefinition Height="64"/>
                <RowDefinition Height="Auto"/>
            </Grid.RowDefinitions>

            <StackPanel Orientation="Horizontal">
                <DatePicker SelectedDate="{Binding
Sector3_DateFirst, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
HorizontalContentAlignment="Center" FontSize="14" Margin="32 0 8 0"
Width="136" Height="32" wpf:TextFieldAssist.Hint="Початок періоду"/>
                <Label Content="-" FontSize="16"
VerticalAlignment="Center" FontWeight="Medium"/>
                <DatePicker SelectedDate="{Binding
Sector3_DateSecond, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
HorizontalContentAlignment="Center" FontSize="14" Margin="8 0 0 0"
Width="136" Height="32" wpf:TextFieldAssist.Hint="Кінець періоду"/>
                <Button Margin="32 0 0 0" Content="Оновити"
Command="{Binding Sector3_UpdateCommand}"/>

```

```

</StackPanel>
<StackPanel Grid.Row="1">
  <Grid>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="Auto"/>
      <ColumnDefinition Width="*/>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
      <RowDefinition Height="Auto"/>
      <RowDefinition/>
    </Grid.RowDefinitions>
    <Label Margin="32 0 0 0" Content="Аналітика
продажу товарів за місяцями" FontSize="18"/>
    <WindowsFormsHost Grid.Row="1"
Background="Honeydew" Width="600" Height="500" VerticalAlignment="Top">
      <DVC:Chart
x:Name="LineChart1_TradesByDate" />
    </WindowsFormsHost>

    <DataGrid Width="480" Height="400"
VerticalAlignment="Top" HorizontalAlignment="Left" Grid.Column="1"
Grid.Row="1" Style="{StaticResource MaterialDesignDataGrid}"
AutoGenerateColumns="False" ItemsSource="{Binding Trades_ByDate_List,
UpdateSourceTrigger=PropertyChanged}"
AutoGeneratingColumn="DataGrid_AutoGeneratingColumn" >
      <DataGrid.Columns>
        <DataGridTextColumn Header="Місяць"
Binding="{Binding Path = Name}" />
        <DataGridTextColumn Header="Кількість
продаж" Binding="{Binding Path = CountTrades}" />
        <DataGridTextColumn Header="Загальна
вартість" Binding="{Binding Path = TotalSum}" />
      </DataGrid.Columns>
    </DataGrid>
  </Grid>
</StackPanel>
</Grid>
</TabItem.Content>
</TabItem>
</TabControl>
</Grid>
</TabItem.Content>
</TabItem>

```

```
<!--ГЛОБАЛЬНА АНАЛІТИКА-->
```

```

    <TabItem>
        <TabItem.Header>
            <Label Content="Загальна аналітика"/>
        </TabItem.Header>
        <TabItem.Content>
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="64"/>
                    <RowDefinition Height="Auto"/>
                </Grid.RowDefinitions>

                <StackPanel Orientation="Horizontal">
                    <DatePicker SelectedDate="{ Binding Sector4_DateFirst,
Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
HorizontalContentAlignment="Center" FontSize="14" Margin="32 0 8 0"
Width="136" Height="32" wpf:TextFieldAssist.Hint="Початок періоду"/>
                    <Label Content="-" FontSize="16"
VerticalAlignment="Center" FontWeight="Medium"/>
                    <DatePicker SelectedDate="{ Binding Sector4_DateSecond,
Mode=TwoWay, UpdateSourceTrigger=PropertyChanged}"
HorizontalContentAlignment="Center" FontSize="14" Margin="8 0 0 0"
Width="136" Height="32" wpf:TextFieldAssist.Hint="Кінець періоду"/>
                    <Button Margin="32 0 0 0" Content="Оновити"
Command="{ Binding Sector4_UpdateCommand}"/>
                </StackPanel>

                <StackPanel Grid.Row="1">
                    <Grid>
                        <Grid.ColumnDefinitions>
                            <ColumnDefinition Width="Auto"/>
                            <ColumnDefinition Width="*/>
                        </Grid.ColumnDefinitions>
                        <Grid.RowDefinitions>
                            <RowDefinition Height="Auto"/>
                            <RowDefinition/>
                        </Grid.RowDefinitions>

                        <Label Margin="32 0 0 0" Content="Загальна аналітика
продажів" FontSize="18"/>

```



```
        <WindowsFormsHost Grid.Row="1"
Background="Honeydew" Width="1128" Height="500" VerticalAlignment="Top"
HorizontalAlignment="Center">
            <DVC:Chart x:Name="LineChart2_TradesByDate" />
        </WindowsFormsHost>

    </Grid>
</StackPanel>
</Grid>

</TabItem.Content>
</TabItem>
</TabControl>
</Grid>
</Grid>
</Window>
```